

**SISTEM INFORMASI AKADEMIK BERBASIS WEB
MENGUNAKAN METODE *SCRUM* PADA
RAUDHATUL ATHFAL AL-ISLAM
GUNUNGCUPU**

TUGAS AKHIR

Diajukan untuk Memenuhi Gelar Sarjana
Pada Program Studi Sistem Informasi Fakultas Teknik
Universitas Galuh



**Disusun oleh:
Rizky Maulana
7020220041**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS GALUH
CIAMIS
2026**

HALAMAN PENGESAHAN

SISTEM INFORMASI AKADEMIK BERBASIS WEB MENGUNAKAN METODE *SCRUM* PADA RAUDHATUL ATHFAL AL-ISLAM GUNUNGCUPU

Disusun Oleh

Rizky Maulana
7020220041

Disetujui dan Disahkan
Ciamis, 27 Agustus 2026

Pembimbing Utama

Pembimbing Pendamping


Dadan Mulyana, S.Kom., M.Kom.
NIK. 3112770807


H. Nana Yudi Permana, S.Kom., M.Kom.
NIK. 3112770718

Menyetujui,
Dekan Fakultas Teknik

Ir. Yanti Deliana, S.T., M.T.
NIK. 3112770447

PERNYATAAN

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah diajukan untuk memperoleh gelar sarjana di perguruan tinggi, dan sepanjang pengetahuan penulis juga tidak ada karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain kecuali yang tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Ciamis, 27 Agustus 2026



Rizky Maulana

ABSTRAK

Pengelolaan data akademik di RA Al-Islam Gunungcupu masih dilakukan secara manual menggunakan buku induk, buku absensi, dan buku penghubung guru dengan orang tua. Kondisi tersebut menyebabkan proses pengolahan, pencarian, penyimpanan, dan penyampaian informasi akademik membutuhkan waktu lebih lama serta belum terintegrasi dalam satu sistem. Berdasarkan kondisi tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem informasi akademik berbasis web untuk mengatasi kendala pengelolaan data manual, menerapkan metode Scrum dalam proses pengembangan sistem, serta mengimplementasikan sistem yang dibangun untuk membantu pengelolaan data akademik di RA Al-Islam Gunungcupu.

Metode penelitian diawali dengan pengumpulan data melalui observasi, wawancara, dan dokumentasi. Hasil pengumpulan data digunakan untuk memahami sistem manual yang sedang berjalan, mengidentifikasi permasalahan, serta menentukan kebutuhan pengguna dan kebutuhan sistem. Kebutuhan tersebut kemudian disusun ke dalam *Product Backlog* dan dikembangkan menggunakan metode Scrum melalui tahap *Sprint Planning* dan tiga *Sprint*. Pada setiap *Sprint* dilakukan perancangan menggunakan *Unified Modeling Language (UML)*, implementasi program, *Black Box Testing*, *Sprint Review*, dan *Sprint Retrospective*. Setelah seluruh *Sprint* selesai, sistem *dideploy* pada server produksi dan diimplementasikan dalam kegiatan operasional sekolah oleh Admin, Guru, dan Orang Tua Siswa sesuai dengan hak akses masing-masing.

Hasil penelitian menunjukkan bahwa Sistem Informasi Akademik berbasis web berhasil dirancang, dibangun, dan diimplementasikan sesuai dengan kebutuhan pengguna. Seluruh fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan fungsional serta mendukung proses pengelolaan dan penyampaian informasi akademik secara lebih terstruktur. Sistem juga berhasil mengintegrasikan data Penilaian Anak yang bersumber dari database Rapor Digital Madrasah (RDM). Evaluasi menggunakan *System Usability Scale (SUS)* terhadap 6 responden menghasilkan skor rata-rata sebesar 83,33 yang termasuk kategori *Acceptable*, *Grade A*, dan *Excellent*. Hasil tersebut menunjukkan bahwa sistem dapat diterima dan memiliki tingkat kemudahan penggunaan yang baik bagi pengguna.

Kata Kunci: Sistem Informasi Akademik, Scrum, Sistem Berbasis Web, *Black Box Testing*, *System Usability Scale*.

ABSTRACT

Academic data management at RA Al-Islam Gunungcupu is still carried out manually using student master books, attendance books, and teacher-parent communication books. This condition causes the processing, searching, storing, and delivering of academic information to require more time and prevents the data from being integrated into a single system. Based on these conditions, this study aims to design and develop a web-based academic information system to overcome the problems of manual data management, apply the Scrum method in the system development process, and implement the developed system to support academic data management at RA Al-Islam Gunungcupu.

The research method began with data collection through observation, interviews, and documentation. The collected data were used to understand the existing manual system, identify problems, and determine user and system requirements. These requirements were then organized into a Product Backlog and developed using the Scrum method through Sprint Planning and three Sprints. Each Sprint included system design using Unified Modeling Language (UML), program implementation, Black Box Testing, Sprint Review, and Sprint Retrospective. After all Sprints were completed, the system was deployed to a production server and implemented in school operational activities by Admins, Teachers, and Parents according to their respective access rights.

The results show that the web-based Academic Information System was successfully designed, developed, and implemented according to user requirements. All developed features functioned according to the required functionality and supported academic data management and information delivery in a more structured manner. The system also successfully integrated Student Assessment data sourced from the Rapor Digital Madrasah (RDM) database. Evaluation using the System Usability Scale (SUS) involving 6 respondents produced an average score of 83.33, which falls into the Acceptable, Grade A, and Excellent categories. These results indicate that the system is acceptable and provides a good level of usability for users.

Keywords: *Academic Information System, Scrum, Web-Based System, Black Box Testing, System Usability Scale.*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya, maka Tugas Akhir ini dapat diselesaikan dengan baik. Salawat serta salam semoga selalu tercurah limpahkan pada baginda Rasulullah Muhammad SAW.

Tugas Akhir yang berjudul “Sistem Informasi Akademik Berbasis Web Menggunakan Metode *Scrum* Pada Raudhatul Athfal Al-Islam Gunungcupu” penulis susun untuk memenuhi persyaratan memperoleh gelar Sarjana Teknik Pada Program Studi Sistem Informasi Fakultas Teknik Universitas Galuh.

Penulis mengucapkan terima kasih yang sebesar-besarnya atas semua bantuan yang telah diberikan, baik secara langsung mau pun tidak langsung selama penyusunan Tugas Akhir ini hingga selesai. Terima kasih penulis sampaikan kepada:

1. Ibu Dr. Tita Rohita, Drs., S.Kep., Ners., M.M., M.Kep., selaku Rektor Universitas Galuh Ciamis.
2. Bapak Dr. Gilang Hamzah Akbar, S.T., M.Sc., selaku Dekan Fakultas Teknik Universitas Galuh Ciamis.
3. Bapak Rian Dwicahya Supriatman, S.T., M.Kom., selaku Ketua Prodi Sistem Informasi Fakultas Teknik Universitas Galuh.
4. Bapak Dadan Mulyana, S.Kom., M.Kom., selaku dosen pembimbing Utama.
5. Bapak H. Nana Yudi Permana, S.Kom., M.Kom., selaku dosen pembimbing Pendamping.
6. Seluruh dosen dan Staf Tata Usaha Fakultas Teknik, Universitas Galuh.
7. Seluruh keluarga penulis, yang telah membesarkan dan mendidik, serta memberikan dukungan dan doa kepada penulis.
8. Rekan-rekan di Jurusan Sistem Informasi Universitas Galuh Ciamis yang juga telah banyak membantu penulis.

Penulis menyadari bahwa skripsi ini jauh dari sempurna, baik dari segi materi maupun penyajiannya. Untuk itu kritik dan saran yang membangun sangat diharapkan dalam penyempurnaan skripsi ini. Penulis berharap, semoga skripsi ini bermanfaat dan menambah wawasan bagi penulis khususnya dan pembaca pada umumnya.

Ciamis, Juni 2026

Penulis

DAFTAR ISI

HALAMAN PENGESAHAN	i
PERNYATAAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan Penelitian.....	3
1.4 Batasan Masalah.....	3
1.5 Manfaat Penelitian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Perkembangan Studi Terdahulu	5
2.2 Urgensi Penelitian	6
2.2.1 Menjembatani kesenjangan digitalisasi di Tingkat PAUD/RA	7
2.2.2 Penerapan solusi teknologi yang berdampak nyata	7
BAB III LANDASAN TEORI.....	9
3.1 Implementasi	9
3.2 Sistem Informasi	9
3.3 Sistem Informasi Akademik.....	10
3.4 Website.....	10
3.5 Metode <i>Scrum</i>	11
3.6 Raudhatul Athfal	13
3.7 <i>Flowchart</i>	14
3.8 <i>Database</i>	15

3.9 <i>Unified Modelling Language (UML)</i>	15
3.9.1 <i>Use case diagram</i>	16
3.9.2 <i>Activity diagram</i>	18
3.9.3 <i>Class diagram</i>	20
3.9.4 <i>Sequence diagram</i>	21
3.10 Laragon	22
3.10.1 MySQL.....	23
3.11 <i>React JS</i>	24
3.12 <i>Framework</i>	25
3.13 Laravel.....	25
3.14 <i>Visual Studio Code (VS Code)</i>	26
3.15 <i>Black Box Testing</i>	27
3.16 <i>System Usability Scale (SUS)</i>	28
BAB IV METODE PENELITIAN.....	29
4.1 Subjek dan Objek Penelitian	29
4.1.1 Subjek Penelitian.....	29
4.1.2 Objek Penelitian.....	29
4.2 Bahan dan Alat	30
4.2.1 Bahan Penelitian.....	30
4.2.2 Alat Penelitian	30
4.3 Bagan Alir Penelitian	31
4.4 Jadwal Penelitian.....	37
BAB V HASIL DAN PEMBAHASAN	38
5.1 Pengumpulan Data	38
5.1.1 Hasil observasi	38
5.1.1.1 Sistem manual yang sedang berjalan	38
5.1.1.2 <i>Flowchart</i> sistem manual yang sedang berjalan	39
5.1.1.3 Permasalahan sistem yang berjalan.....	42
5.1.2 Hasil wawancara	43
5.1.2.1 Hasil wawancara dengan Kepala RA.....	43
5.1.2.2 Hasil wawancara dengan Guru.....	44

5.1.2.3 Kebutuhan pengguna terhadap sistem.....	45
5.1.3 Hasil dokumentasi.....	46
5.1.3.1 Buku induk siswa	46
5.1.3.2 Buku absensi siswa	47
5.1.3.3 Buku penghubung guru dan orang tua	48
5.2 Menyusun Product Backlog	50
5.2.1 Identifikasi kebutuhan sistem.....	50
5.2.2 Use case diagram sistem usulan.....	52
5.2.2.1 Deskripsi aktor	53
5.2.2.2 Deskripsi per-use case.....	54
5.2.3 Product backlog.....	56
5.3 Sprint Planning.....	63
5.4 Pelaksanaan Sprint	64
5.4.1 <i>Sprint 1</i>	65
5.4.1.1 Login	65
5.4.1.2 Dashboard	72
5.4.1.3 Pengguna (admin, Guru, Orang Tua Siswa).....	79
5.4.1.4 Tahun Pelajaran	112
5.4.1.5 Kelas.....	124
5.4.1.6 Kelas (kenaikan kelas)	136
5.4.1.7 Kalender pendidikan (melihat).....	146
5.4.1.8 <i>Black box testing sprint 1</i>	158
5.4.1.9 <i>Sprint review sprint 1</i>	161
5.4.1.10 <i>Sprint retrospective sprint 1</i>	163
5.4.2 Sprint 2	164
5.4.2.1 Siswa (kelola siswa).....	165
5.4.2.2 Absensi (Input guru).....	179
5.4.2.3 Kalender pendidikan (mengelola)	187
5.4.2.4 Kalender pendidikan (<i>upload</i> PDF dokumen kalender pendidikan)	199
5.4.2.5 Siswa (cetak biodata PDF).....	209

5.4.2.6 <i>Black box testing sprint 2</i>	217
5.4.2.7 <i>Sprint review sprint 2</i>	220
5.4.2.8 <i>Sprint retrospective sprint 2</i>	222
5.4.3 Sprint 3	223
5.4.3.1 Daftar siswa.....	224
5.4.3.2 Profil siswa.....	232
5.4.3.3 Absensi (rekap admin).....	240
5.4.3.4 Riwayat kehadiran.....	248
5.4.3.5 Penilaian anak (admin, guru, orang tua)	257
5.4.3.6 <i>Black box testing sprint 3</i>	265
5.4.3.7 <i>Sprint review sprint 3</i>	268
5.4.3.8 <i>Sprint retrospective sprint 3</i>	270
5.5 Implementasi Sistem	271
5.5.1 <i>Deployment dan hosting</i>	272
5.5.1.1 Proses deployment ke hosting.....	273
5.5.1.2 Konfigurasi sistem pada <i>server</i> hosting	274
5.5.1.3 Hasil deployment sistem	275
5.5.2 Hasil implementasi sistem.....	276
5.5.2.1 Ringkasan hasil implementasi.....	277
5.6 Evaluasi Menggunakan <i>System Usability Scale</i> (SUS).....	278
5.6.1 Instrumen kuesioner dan penentuan jumlah responden	278
5.6.2 Aturan perhitungan dan hasil skor SUS	280
5.6.3 Analisis dan pembahasan kelayakan usabilitas	281
BAB VI PENUTUP	284
6.1 Simpulan	284
6.2 Saran.....	284
DAFTAR PUSTAKA.....	286
LAMPIRAN.....	291

DAFTAR TABEL

Tabel 3.1 <i>Flowchart</i>	14
Tabel 3.2 <i>Use Case Diagram</i>	17
Tabel 3.3 <i>Activity Diagram</i>	18
Tabel 3.4 <i>Class Diagram</i>	20
Tabel 3.5 <i>Sequence Diagram</i>	22
Tabel 4.1 Jadwal Penelitian.....	37
Tabel 5.1 Deskripsi Aktor	54
Tabel 5.2 Deskripsi Per-Use Case	54
Tabel 5.3 <i>Product Backlog</i>	56
Tabel 5.4 <i>Sprint Planning</i>	63
Tabel 5.5 Hasil <i>Black Box Testing Sprint 1</i>	159
Tabel 5.6 Hasil <i>Sprint Review Sprint 1</i>	161
Tabel 5.7 Hasil <i>Sprint Retrospective Sprint 1</i>	164
Tabel 5.8 Hasil <i>Black Box Testing Sprint 2</i>	218
Tabel 5.9 Hasil <i>Sprint Review Sprint 2</i>	220
Tabel 5.10 Hasil <i>Sprint Resrotpective Sprint 2</i>	222
Tabel 5.11 Hasil <i>Blackbox Testing Sprint 3</i>	266
Tabel 5.12 Hasil <i>Sprint Review Sprint 3</i>	268
Tabel 5.13 Hasil <i>Sprint Retrospective Sprint 3</i>	270
Tabel 5.14 Ringkasan Hasil Implementasi Sistem	277
Tabel 5.15 Daftar Pernyataan Kuesioner <i>System Usability Scale (SUS)</i>	279
Tabel 5.16 Hasil Perhitungan Skor SUS Seluruh Responden	281

DAFTAR GAMBAR

Gambar 3.1 Proses <i>Scrum</i> (Sumber: (Magdalena et al., 2023)	11
Gambar 4.1 Bagan Alir Penelitian (Sumber: Data Pribadi)	32
Gambar 5.1 <i>Flowchart</i> Sistem Manual yang sedang berjalan	40
Gambar 5.2 Buku induk siswa	47
Gambar 5.3 Buku absensi siswa.....	48
Gambar 5.4 Buku penghubung guru dan orang tua	49
Gambar 5.5 <i>Use Case Diagram</i>	53
Gambar 5.6 <i>Activity diagram</i> login.....	66
Gambar 5.7 <i>Class Diagram</i> Login.....	67
Gambar 5.8 <i>Sequence Diagram</i> Login.....	68
Gambar 5.9 <i>Code Backend</i> Login	69
Gambar 5.10 <i>Code Frontend</i> Login	70
Gambar 5.11 Halaman Login	71
Gambar 5.12 <i>Code Backend</i> Dashboard	73
Gambar 5.13 <i>Code Frontend</i> Dashboard Admin	74
Gambar 5.14 <i>Code Frontend</i> Dashboard Guru	75
Gambar 5.15 <i>Code Frontend</i> Dashboard Orang Tua Siswa.....	76
Gambar 5.16 Halaman Dashboard Admin	77
Gambar 5.17 Halaman Dashboard Guru.....	78
Gambar 5.18 Halaman Dashboard Orang Tua Siswa	79
Gambar 5.19 <i>Activity Diagram</i> Pengelolaan Data Admin	81
Gambar 5.20 <i>Class Diagram</i> Pengelolaan Data Admin	83
Gambar 5.21 <i>Sequence Diagram</i> Pengelolaan Data Admin	85
Gambar 5.22 <i>Code Backend</i> Pengelolaan Data Admin.....	87
Gambar 5.23 <i>Code Frontend</i> Pengelolaan Data Admin.....	88
Gambar 5.24 Halaman Pengelolaan Data Admin	89
Gambar 5.25 Halaman Tambah Data Admin	90
Gambar 5.26 Halaman Edit Data Admin	91
Gambar 5.27 Halaman Hapus Data Admin.....	91
Gambar 5.28 <i>Activity Diagram</i> Pengelolaan Data Guru.....	93
Gambar 5.29 <i>Sequence Diagram</i> Pengelolaan Data Guru	95
Gambar 5.30 <i>Code Backend</i> Pengelolaan Data Guru	97
Gambar 5.31 <i>Code Frontend</i> Pengelolaan Data Guru	98
Gambar 5.32 Halaman Pengelolaan Data Guru	99
Gambar 5.33 Halaman Tambah Data Guru	100
Gambar 5.34 Halaman Edit Data Guru	101
Gambar 5.35 Halaman Hapus Data Guru	102

Gambar 5.36 <i>Activity Diagram</i> Pengelolaan Data Orang Tua Siswa.....	103
Gambar 5.37 <i>Sequence Diagram</i> Pengelolaan Data Orang Tua Siswa.....	105
Gambar 5.38 <i>Code Backend</i> Pengelolaan Data Orang Tua Siswa	107
Gambar 5.39 <i>Code Frontend</i> Pengelolaan Data Orang Tua Siswa	108
Gambar 5.40 Halaman Pengelolaan Data Orang Tua Siswa.....	109
Gambar 5.41 Halaman Tambah Data Orang Tua Siswa.....	110
Gambar 5.42 Halaman Edit Data Orang Tua Siswa.....	111
Gambar 5.43 Halaman Hapus Data Orang Tua Siswa	112
Gambar 5.44 <i>Activity Diagram</i> Tahun Pelajaran	114
Gambar 5.45 <i>Class Diagram</i> Tahun Pelajaran.....	115
Gambar 5.46 <i>Sequence Diagram</i> Tahun Pelajaran.....	117
Gambar 5.47 <i>Code Backend</i> Tahun Pelajaran	118
Gambar 5.48 <i>Code Frontend</i> Tahun Pelajaran.....	120
Gambar 5.49 Halaman Tahun Pelajaran.....	121
Gambar 5.50 Halaman Tambah Tahun Pelajaran	122
Gambar 5.51 Halaman Edit Tahun Pelajaran	123
Gambar 5.52 Halaman Hapus Tahun Pelajaran	124
Gambar 5.53 <i>Activity Diagram</i> Kelas	126
Gambar 5.54 <i>Class Diagram</i> Kelas	127
Gambar 5.55 <i>Sequence Diagram</i> Kelas	129
Gambar 5.56 <i>Code Backend</i> Kelas.....	130
Gambar 5.57 <i>Code Frontend</i> Kelas.....	132
Gambar 5.58 Halaman Kelas	133
Gambar 5.59 Halaman Tambah Kelas.....	134
Gambar 5.60 Halaman Edit Kelas.....	135
Gambar 5.61 Halaman Hapus Kelas	136
Gambar 5.62 <i>Activity Diagram</i> Kelas (Kenaikan Kelas).....	138
Gambar 5.63 <i>Class Diagram</i> Kelas (Kenaikan Kelas)	139
Gambar 5.64 <i>Sequence Diagram</i> Kelas (Kenaikan Kelas)	141
Gambar 5.65 <i>Code Backend</i> Kelas (Kenaikan Kelas)	142
Gambar 5.66 <i>Code Frontend</i> Kelas (Kenaikan Kelas)	144
Gambar 5.67 Halaman Kelas (Kenaikan Kelas)	145
Gambar 5.68 <i>Activity Diagram</i> Melihat Kalender Pendidikan Guru.....	147
Gambar 5.69 <i>Class Diagram</i> Kalender Pendidikan	148
Gambar 5.70 <i>Sequence Diagram</i> Melihat Kalender Pendidikan	149
Gambar 5.71 <i>Code Backend</i> Kalender Pendidikan (melihat) Guru	150
Gambar 5.72 <i>Code Frontend</i> Kalender Pendidikan (melihat) Guru	151
Gambar 5.73 Halaman Melihat Kalender Pendidikan Guru	152
Gambar 5.74 <i>Activity Diagram</i> Melihat Kalender Pendidikan Orang Tua Siswa.....	153

Gambar 5.75 <i>Sequence Diagram</i> Melihat Kalender Pendidikan Orang Tua Siswa	154
Gambar 5.76 <i>Code Backend</i> Kalender Pendidikan (melihat) Orang Tua Siswa.	155
Gambar 5.77 <i>Code Frontend</i> Kalender Pendidikan (melihat) Orang Tua Siswa	156
Gambar 5.78 Halaman Kalender Pendidikan (melihat) Orang Tua Siswa.....	157
Gambar 5.79 <i>Activity Diagram</i> Siswa (Kelola siswa)	167
Gambar 5.80 <i>Class Diagram</i> Siswa (Kelola siswa).....	169
Gambar 5.81 <i>Sequence Diagram</i> Siswa (Kelola siswa).....	171
Gambar 5.82 <i>Code Backend</i> Siswa (Kelola siswa).....	173
Gambar 5.83 <i>Code Frontend</i> Siswa (Kelola siswa).....	175
Gambar 5.84 Halaman Siswa (Kelola siswa).....	176
Gambar 5.85 Halaman Tambah Siswa	177
Gambar 5.86 Halaman Edit Siswa	178
Gambar 5.87 Halaman Hapus Siswa.....	179
Gambar 5.88 <i>Activity Diagram</i> Absensi (Input guru).....	181
Gambar 5.89 <i>Class Diagram</i> Absensi (Input guru)	182
Gambar 5.90 <i>Sequence Diagram</i> Absensi (Input guru)	183
Gambar 5.91 <i>Code Backend</i> Absensi (Input guru).....	184
Gambar 5.92 <i>Code Frontend</i> Absensi (Input guru).....	185
Gambar 5.93 Halaman Absensi (Input guru)	186
Gambar 5.94 <i>Activity Diagram</i> Kalender Pendidikan (Mengelola).....	189
Gambar 5.95 <i>Class Diagram</i> Kalender Pendidikan (Mengelola)	191
Gambar 5.96 <i>Sequence Diagram</i> Kalender Pendidikan (Mengelola).....	192
Gambar 5.97 <i>Code Backend</i> Kalender Pendidikan (Mengelola)	193
Gambar 5.98 <i>Code Frontend</i> Kalender Pendidikan (Mengelola)	195
Gambar 5.99 Halaman Kalender Pendidikan (Mengelola)	196
Gambar 5.100 Halaman Tambah Kalender Pendidikan	197
Gambar 5.101 Halaman Edit Kalender Pendidikan	198
Gambar 5.102 Halaman Hapus Kalender Pendidikan.....	199
Gambar 5.103 <i>Activity Diagram Upload</i> PDF Dokumen Kalender Pendidikan	201
Gambar 5.104 <i>Class Diagram Upload</i> PDF Dokumen Kalender Pendidikan....	202
Gambar 5.105 <i>Sequence Diagram Upload</i> PDF Dokumen Kalender Pendidikan	203
Gambar 5.106 <i>Code Backend Upload</i> PDF Dokumen Kalender Pendidikan.....	205
Gambar 5.107 <i>Code Frontend Upload</i> PDF Dokumen Kalender Pendidikan....	207
Gambar 5.108 Halaman <i>Upload</i> PDF Dokumen Kalender Pendidikan.....	208
Gambar 5.109 <i>Activity Diagram</i> Siswa (Cetak biodata PDF).....	210
Gambar 5.110 <i>Sequence Diagram</i> Siswa (Cetak Biodata PDF)	212
Gambar 5.111 <i>Code Backend</i> Siswa (Cetak Biodata PDF).....	213
Gambar 5.112 <i>Code Frontend</i> Siswa (Cetak Biodata PDF)	215

Gambar 5.113 Halaman Siswa (Cetak Biodata PDF)	216
Gambar 5.114 <i>Activity Diagram</i> Melihat Daftar Siswa	225
Gambar 5.115 <i>Class Diagram</i> Daftar Siswa	226
Gambar 5.116 <i>Sequence diagram</i> daftar siswa	227
Gambar 5.117 <i>Code Backend</i> Daftar Siswa	229
Gambar 5.118 <i>Code Frontend</i> Daftar Siswa	230
Gambar 5.119 Halaman Daftar Siswa	231
Gambar 5.120 <i>Activity Diagram</i> Melihat Profil Siswa	233
Gambar 5.121 <i>Class Diagram</i> Profil Siswa	234
Gambar 5.122 <i>Sequence Diagram</i> Melihat Profil Siswa	235
Gambar 5.123 <i>Code Backend</i> Profil Siswa	236
Gambar 5.124 <i>Code Frontend</i> Profil Siswa	238
Gambar 5.125 Halaman Profil Siswa	239
Gambar 5.126 <i>Activity Diagram</i> Absensi (Rekap Admin)	241
Gambar 5.127 <i>Class Diagram</i> Absensi (Rekap Admin)	242
Gambar 5.128 <i>Sequence Diagram</i> Absensi (Rekap Admin)	243
Gambar 5.129 <i>Code Backend</i> Absensi (Rekap Admin)	245
Gambar 5.130 <i>Code Frontend</i> Absensi (Rekap Admin)	246
Gambar 5.131 Halaman Absensi (Rekap Admin)	247
Gambar 5.132 <i>Activity Diagram</i> Riwayat Kehadiran	249
Gambar 5.133 <i>Class Diagram</i> Riwayat Kehadiran	251
Gambar 5.134 <i>Sequence Diagram</i> Riwayat Kehadiran	252
Gambar 5.135 <i>Code Backend</i> Riwayat Kehadiran	253
Gambar 5.136 <i>Code Frontend</i> Riwayat Kehadiran	255
Gambar 5.137 Halaman Riwayat Kehadiran	256
Gambar 5.138 <i>Activity Diagram</i> Penilaian Anak (Admin dan Guru)	258
Gambar 5.139 <i>Class Diagram</i> Penilaian Anak (Admin dan Guru)	259
Gambar 5.140 <i>Sequence Diagram</i> Penilaian Anak (Admin dan Guru)	260
Gambar 5.141 <i>Code Backend</i> Penilaian Anak (Admin dan Guru)	261
Gambar 5.142 <i>Code Frontend</i> Penilaian Anak (Admin dan Guru)	262
Gambar 5.143 Halaman Penilaian Anak Admin	263
Gambar 5.144 Halaman Penilaian Anak Guru	264
Gambar 5.145 Direktori aplikasi pada File Manager HestiaCP	273
Gambar 5.146 Konfigurasi berkas .env pada server production	274
Gambar 5.147 Sistem Informasi Akademik RA Al-Islam yang telah berhasil diimplementasikan pada server	275

DAFTAR LAMPIRAN

Lampiran 1. Surat Bukti Implementasi Sistem.....	291
---	------------

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Sistem informasi modern semakin meluas pemanfaatannya di berbagai bidang, khususnya pendidikan. Penerapan teknologi informasi mendorong terjadinya transformasi dalam pengelolaan kegiatan akademik di lembaga pendidikan. Informasi menjadi komponen penting dalam mendukung aktivitas akademik, sehingga lembaga pendidikan membutuhkan sistem yang mampu menyajikan dan mengelola informasi secara efektif dan efisien (Marlina Ariansyah & Wijaya, 2021). Keberadaan sistem informasi memberikan kemudahan bagi pengguna dalam mengolah data dan mengakses informasi tanpa dibatasi oleh waktu dan tempat. Penerapan sistem informasi di lingkungan sekolah sangat dibutuhkan untuk membantu pengelolaan data akademik, seperti data peserta didik, hasil evaluasi pembelajaran, serta kehadiran. Selain mempercepat proses pekerjaan, sistem informasi juga mampu menyimpan data secara lebih terstruktur sehingga informasi yang dihasilkan menjadi lebih rapi dan sesuai dengan kebutuhan pengguna (Zulfa & Arifudin, 2025).

Meskipun demikian, masih terdapat lembaga pendidikan yang melakukan pengelolaan data akademik secara manual. Pengelolaan data secara manual sering kali menimbulkan berbagai kendala, seperti proses pengolahan data yang membutuhkan waktu lebih lama, kesulitan dalam melakukan pencarian data, serta risiko terjadinya kesalahan dalam pencatatan. Kondisi tersebut menyebabkan penyajian informasi menjadi kurang efektif dan efisien sehingga dapat mempengaruhi kelancaran kegiatan akademik (Santoso & Amanullah, 2022).

Raudhatul Athfal (RA) Al-Islam Gunungcupu adalah lembaga pendidikan anak usia dini yang masih menggunakan pengelolaan data akademik manual. Pencatatan kegiatan akademik disimpan dalam buku induk, sehingga data belum terintegrasi dalam sistem. Akibatnya, pengolahan dan pencarian data kurang efisien, terutama

saat dibutuhkan secara cepat. Penyampaian informasi akademik pun kurang optimal karena kurangnya dukungan sistem terstruktur.

Berdasarkan permasalahan tersebut, diperlukan suatu sistem yang dapat membantu pihak sekolah dalam mengelola data akademik secara lebih efektif dan terintegrasi. Salah satu solusi yang dapat diterapkan adalah dengan membangun sistem informasi akademik berbasis web sehingga data dapat dikelola dan diakses dengan lebih mudah. Dalam penelitian tersebut, digunakan metode *Scrum* sebagai metode pengembangan sistem karena memiliki pendekatan iteratif dan adaptif yang memungkinkan proses pengembangan dilakukan secara bertahap serta dapat menyesuaikan dengan kebutuhan pengguna. *Scrum* merupakan salah satu bagian dari metodologi *Agile* yang umum diaplikasikan dalam proses pengembangan produk di bidang teknologi. Selain itu, *Scrum* juga dapat diartikan sebagai suatu kerangka kerja yang dirancang agar sebuah tim mampu menyelesaikan berbagai permasalahan yang kompleks, serta mendorong tim tersebut untuk menghasilkan produk dengan nilai dan kualitas tinggi secara produktif maupun kreatif (Maulana & Prianto, 2023).

Dengan demikian, penelitian ini bertujuan untuk merancang dan membangun sistem informasi akademik berbasis web menggunakan metode *Scrum* pada RA Al-Islam Gunungcupu yang diharapkan dapat membantu meningkatkan efektivitas dalam pengelolaan data akademik serta memudahkan penyajian informasi yang berkaitan dengan kegiatan akademik di lingkungan sekolah.

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem informasi akademik berbasis web untuk mengatasi kendala pengelolaan data manual pada RA Al-Islam Gunungcupu?
2. Bagaimana penerapan metode *Scrum* dalam merancang dan membangun sistem informasi akademik berbasis web pada RA Al-Islam Gunungcupu?

3. Bagaimana implementasi sistem informasi akademik berbasis web yang dibangun menggunakan metode *Scrum* untuk membantu pengelolaan data akademik di RA Al-Islam Gunungcupu?

1.3 Tujuan Penelitian

Berdasarkan perumusan masalah, maka penelitian ini mempunyai tujuan sebagai berikut:

1. Merancang dan membangun sistem informasi akademik berbasis web untuk mengatasi kendala pengelolaan data manual pada RA Al-Islam Gunungcupu.
2. Menerapkan metode *Scrum* dalam merancang dan membangun sistem informasi akademik berbasis web pada RA Al-Islam Gunungcupu.
3. Mengimplementasikan sistem informasi akademik berbasis web yang dibangun menggunakan metode *Scrum* untuk membantu pengelolaan data akademik di RA Al-Islam Gunungcupu.

1.4 Batasan Masalah

Agar penyusunan penelitian ini tidak keluar dari pokok permasalahan yang dirumuskan, maka ruang lingkup pembahasan dibatasi pada:

1. Penelitian ini dilakukan pada RA Al-Islam Gunungcupu, Sindangkasih, Ciamis, Jawa Barat.
2. Sistem yang dibangun tidak mencakup pengelolaan administrasi keuangan sekolah.
3. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Scrum*.
4. Sistem yang dibangun merupakan sistem informasi akademik berbasis web.
5. Fitur yang dibahas dalam sistem ini terbatas pada manajemen pengguna, tahun pelajaran, kalender pendidikan, kelas, siswa, absensi. Khusus untuk fitur penilaian anak, sistem ini mengintegrasikan database eksternal dari Rapor Digital Madrasah (RDM).
6. Sistem ini dibangun dengan bahasa pemrograman PHP (menggunakan framework Laravel), *TypeScript/JavaScript (React JS)*, serta menggunakan database MySQL.

7. Pengujian kelayakan sistem difokuskan pada fungsionalitas menggunakan *Black Box Testing*. sedangkan evaluasi tingkat kemudahan dan penerimaan pengguna akhir diukur menggunakan metode *System Usability Scale* (SUS).

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

1. Manfaat Teoritis

Secara teoritis, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan ilmu pengetahuan di bidang sistem informasi, khususnya yang berkaitan dengan perancangan dan pembangunan sistem informasi akademik berbasis web serta penerapan metode *Scrum* dalam pengembangan sistem.

2. Manfaat Praktis

- a) Bagi Penulis

Penelitian ini diharapkan dapat menjadi sarana bagi peneliti dalam menerapkan serta mengembangkan pengetahuan yang telah diperoleh selama perkuliahan, khususnya dalam bidang perancangan dan pembangunan sistem informasi berbasis web menggunakan metode *Scrum*.

- b) Bagi RA Al-Islam Gunungcupu

Penelitian ini diharapkan dapat membantu pihak RA Al-Islam Gunungcupu dalam mengelola data yang berkaitan dengan kegiatan akademik secara lebih efektif melalui sistem informasi akademik berbasis web sehingga proses pengolahan, penyimpanan, dan pencarian data dapat dilakukan dengan lebih mudah dan terstruktur.

- c) Bagi Umum

Hasil penelitian ini diharapkan dapat menjadi referensi bagi pihak lain maupun penelitian selanjutnya yang berkaitan dengan pembangunan sistem informasi akademik berbasis web serta penerapan metode *Scrum* dalam pengembangan sistem.

BAB II

TINJAUAN PUSTAKA

2.1 Perkembangan Studi Terdahulu

Perkembangan studi pada bidang sistem informasi pendidikan menunjukkan bahwa fokus bahasan telah bergerak melampaui sekadar digitalisasi pencatatan menuju pembuatan sistem yang mudah menyesuaikan diri dengan kebutuhan pengguna. Dalam dunia pendidikan, kebutuhan ini menjadi sangat penting mengingat tingginya beban tenaga pendidik dalam mengelola data akademik secara rinci.

Penelitian (Erawati et al., 2023) berupaya menjawab kebutuhan ini dengan merancang sistem informasi akademik menggunakan pendekatan SDLC model *waterfall*. Sistem tersebut dinilai berhasil menghubungkan informasi perkembangan anak kepada orang tua, yang dibuktikan dengan nilai pengujian statistik yang menunjukkan tingginya akurasi dan kecepatan pengolahan data. Meskipun tingkat keberhasilannya tinggi, sistem tersebut masih memiliki kelemahan mendasar pada aspek fleksibilitas. Pendekatan *waterfall* yang kaku menunjukkan perlunya cara kerja yang lebih dinamis jika sewaktu-waktu terjadi perubahan standar operasional maupun kurikulum di sekolah.

Dari sudut pandang manajemen, (Ariansyah & Wijaya, 2021) mengembangkan sistem informasi akademik di Sekolah Dasar (SD) menggunakan metode *prototype*. Sistem berbasis web tersebut dirancang untuk mengubah data akademik menjadi bentuk digital secara menyeluruh, mulai dari data siswa, guru, jadwal pelajaran, hingga pengolahan nilai. Walaupun sistem tersebut secara fungsi sangat efektif untuk menggantikan pencatatan buku fisik, struktur tampilan dan basis datanya rentan rusak jika harus dipelihara dalam jangka waktu panjang tanpa pengawasan yang ketat. Penggunaan tampilan aplikasi yang rumit juga kerap menciptakan kebingungan bagi pengguna, sebagaimana ditunjukkan oleh (Zulfa & Arifudin, 2025) yang mengevaluasi aplikasi SALAM di tingkat perguruan tinggi. Mereka menemukan bahwa kendala teknis dan tampilan sistem yang terlalu padat dapat memicu kesulitan penyesuaian diri serta hambatan pemahaman digital, yang

menegaskan bahwa sistem harus dirancang ringkas dan seramah mungkin bagi pengguna.

Mengenai cara pengembangan perangkat lunak yang lebih mudah menyesuaikan pengguna, (Subagyo & Jannah, 2025) meneliti penerapan metode *Scrum* dalam pengembangan sistem akademik. Studi tersebut menemukan bahwa siklus pengerjaan bertahap (*sprint*) dalam *Scrum* berhasil meningkatkan efisiensi dan mampu menampung kebutuhan penilaian kurikulum *Outcome-Based Education* (OBE) yang sangat mendetail. Hal ini sejalan dengan penelitian (Nadhira et al., 2022) yang membuktikan bahwa perancangan aplikasi akademik menggunakan *Agile Scrum* yang dipadukan dengan pengujian *White Box Testing* mampu menghasilkan sistem yang sangat stabil dan mempermudah proses pengelolaan nilai secara daring. Pendekatan secara bertahap ini telah terbukti secara nyata mampu melibatkan pengguna secara langsung dalam proses pembuatan aplikasi, sehingga dapat mengurangi rasa enggan atau penolakan dari guru saat sistem tersebut diterapkan.

Berdasarkan kajian tersebut, penelitian-penelitian sebelumnya telah memberikan dasar yang sangat penting, yaitu perlunya otomatisasi data akademik, pentingnya tampilan yang mudah digunakan agar tidak membebani guru, serta keunggulan metode *Agile Scrum* untuk mengatasi sistem yang kaku. Namun, menggabungkan seluruh dasar penting tersebut ke dalam satu perangkat lunak menggunakan metode *Scrum* untuk memecahkan masalah pencatatan manual secara khusus di RA Al-Islam Gunungcupu belum pernah dilakukan oleh penelitian mana pun.

2.2 Urgensi Penelitian

Posisi strategis penelitian ini didasarkan pada hal-hal yang belum dibahas pada penelitian sebelumnya, serta didorong oleh kondisi nyata operasional pendidikan di lapangan. Arah dan keluaran dari penelitian ini secara langsung dibentuk oleh dua urgensi utama.

2.2.1 Menjembatani kesenjangan digitalisasi di Tingkat PAUD/RA

Raudhatul Athfal Al-Islam Gunungcupu dipilih sebagai objek penelitian karena pengelolaan data akademik masih dilakukan secara manual menggunakan dokumen fisik. Kegiatan tersebut meliputi pendataan siswa yang dicatat pada buku induk, pencatatan kehadiran siswa pada buku absensi, serta penyampaian informasi akademik kepada orang tua melalui buku penghubung guru dan orang tua. Pengelolaan data secara manual dan berulang tersebut menyebabkan proses pengolahan, pencarian, serta penyampaian informasi membutuhkan waktu yang relatif lebih lama, terutama ketika data dibutuhkan dalam waktu cepat. Kondisi tersebut menunjukkan adanya kesenjangan dalam pemanfaatan teknologi untuk mendukung pengelolaan data akademik pada tingkat PAUD/RA.

Oleh karena itu, pembuatan sistem informasi akademik berbasis web sangat mendesak untuk segera diterapkan guna menggantikan pencatatan manual tersebut. Melalui sistem yang terkomputerisasi ini, seluruh data akademik dapat dikelola dan disimpan di dalam satu pusat *database*. Penerapan sistem ini akan membantu sekolah dalam mempercepat proses pengelolaan data, mempermudah pencarian informasi, serta memastikan data yang dihasilkan tersimpan dengan lebih rapi, terstruktur, dan aman.

2.2.2 Penerapan solusi teknologi yang berdampak nyata

Urgensi penting selanjutnya adalah untuk menghasilkan perangkat lunak yang berfungsi dengan baik dan memberikan manfaat langsung bagi sekolah. Penelitian ini secara sadar difokuskan pada pembuatan sistem yang benar-benar diimplementasikan secara utuh di lapangan. Hal ini bertujuan untuk memastikan agar hasil inovasi tidak hanya berhenti pada tahap perancangan desain atau tahap uji coba model awal *prototype* saja. Melalui pembuatan sistem informasi akademik berbasis web yang menggunakan teknologi *Laravel* dan *React JS*, serta dihubungkan dengan data dari Rapor Digital Madrasah (RDM), kecanggihan teknologi dapat disederhanakan menjadi alat bantu yang tepat sasaran.

Perancangan sistem ini memungkinkan semua kerumitan pemrosesan data dikelola dengan rapi secara otomatis oleh sistem di balik layar. Pada akhirnya, hal ini akan memudahkan guru dan staf untuk cukup menggunakan tampilan layar yang bersih dan ramah pengguna. Aplikasi yang dibuat ini akan digunakan secara langsung sebagai pusat data akademik di RA Al-Islam Gunungcupu untuk menyelesaikan kendala pencatatan kegiatan akademik secara manual. Penelitian ini memberikan contoh nyata tentang bagaimana metode *Scrum* dapat diterapkan sepenuhnya menjadi produk yang siap beroperasi di sekolah.

BAB III

LANDASAN TEORI

3.1 Implementasi

Implementasi berasal dari bahasa Inggris “*to implement*” yang artinya mengimplementasikan. Implementasi bukan hanya suatu aktivitas, tetapi implementasi juga merupakan suatu kegiatan yang direncanakan serta dilaksanakan dengan serius dan mengacu pada norma-norma tertentu, guna mencapai tujuan kegiatan (Wahidin et al., 2021). Implementasi adalah sebuah perangkat aktivitas baru yang di dalamnya terdapat pengharapan mengenai perubahan terhadap objek-objek yang bersangkutan. Dalam pelaksanaannya tersebut, ada pula harapan agar apa yang telah tersusun dalam rencana yang sedemikian matang dapat diterima oleh seluruh pihak dari aspek yang dikenainya. Sehingga perubahan yang terjadi akan bersifat menyeluruh (Wibowo, 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa implementasi merupakan serangkaian tindakan terencana dan sistematis yang bertujuan untuk menerapkan suatu rancangan secara nyata, guna mewujudkan perubahan positif yang menyeluruh dan dapat diterima oleh seluruh pihak yang bersangkutan.

3.2 Sistem Informasi

Sistem informasi merupakan suatu kesatuan komponen yang saling berkaitan dengan tujuan untuk mengumpulkan, memproses, menyimpan, serta menyebarkan informasi demi menunjang proses pengambilan keputusan dan pengawasan di dalam sebuah organisasi. Secara umum, istilah 'sistem informasi' merujuk pada serangkaian elemen seperti sumber daya manusia, data, perangkat lunak, perangkat keras, serta prosedur yang berkolaborasi untuk menyajikan fakta dan informasi yang akurat kepada pihak yang membutuhkan di waktu yang sesuai (Asih Della Saftini, 2024). Sistem informasi dapat didefinisikan sebagai suatu sistem buatan manusia yang mencakup berbagai komponen dalam sebuah organisasi untuk mencapai tujuan utama, yakni menghasilkan suatu informasi (Arifin et al., 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa sistem informasi adalah sebuah instrumen terpadu yang memadukan teknologi dan sumber daya manusia untuk mengelola data mentah menjadi informasi yang bernilai guna bagi kelancaran operasional sebuah instansi.

3.3 Sistem Informasi Akademik

Sistem informasi akademik merupakan sebuah sistem yang berfungsi untuk mengolah data-data akademik pada suatu lembaga pendidikan, baik formal maupun informal, mulai dari tingkat dasar hingga perguruan tinggi. Secara umum, ruang lingkup data yang dikelola dalam sistem ini mencakup data tenaga pendidikan, peserta didik, mata pelajaran, jadwal pembelajaran, serta data-data pendukung lainnya yang sesuai dengan kebutuhan operasional masing-masing (Zulfa & Arifudin, 2025). Sistem informasi akademik dikembangkan untuk membantu menyelesaikan berbagai problematika yang terjadi di lembaga pendidikan terkait proses pengelolaan data akademik secara efektif dan efisien (Halim & Rahmayanti, 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa sistem informasi akademik merupakan solusi teknologi yang dirancang untuk mengintegrasikan seluruh elemen administrasi Pendidikan agar proses pengelolaan dan pencarian data menjadi lebih terstruktur, cepat, dan akurat.

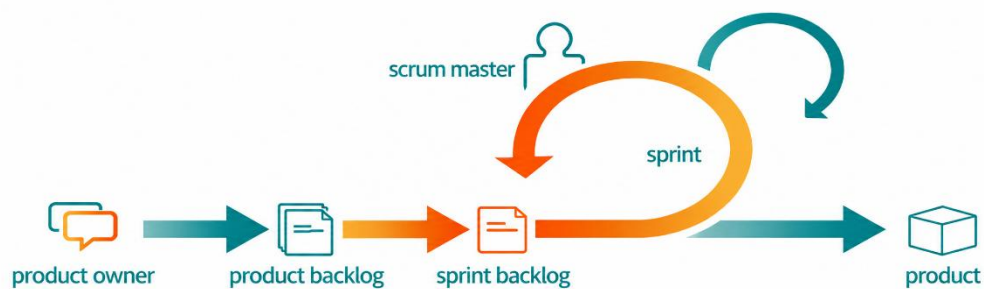
3.4 Website

Situs web (*website*) merupakan kumpulan halaman dalam suatu domain yang memuat berbagai informasi agar dapat diakses dan dilihat oleh pengguna internet melalui sebuah mesin pencari. Informasi yang disajikan di dalam sebuah situs web pada umumnya berupa konten teks, gambar, ilustrasi, maupun video untuk berbagai macam kepentingan. Secara umum, tampilan awal dari sebuah situs web dapat diakses melalui halaman utama (*homepage*) menggunakan peramban (*browser*) dengan memasukkan alamat URL yang tepat. Pada halaman utama tersebut, memuat pula beberapa halaman web turunan yang saling terintegrasi satu sama lain (Syahputra et al., 2021). Website merupakan kumpulan halaman web yang dapat

diakses oleh publik dan saling terkait yang berada di bawah satu nama domain. Website dapat dibangun dan dikelola oleh individu, kelompok, entitas bisnis, maupun organisasi untuk melayani berbagai tujuan (Nuh, 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa website adalah sebuah media informasi digital berbasis internet yang mengintegrasikan berbagai pihak dalam menyebarkan konten secara luas dan mudah diakses oleh publik.

3.5 Metode *Scrum*



Gambar 3.1 Proses *Scrum*
(Sumber: (Magdalena et al., 2023))

Scrum merupakan salah satu bagian dari metodologi *Agile* yang umum diaplikasikan dalam proses pengembangan produk di bidang teknologi. Selain itu, *Scrum* juga dapat diartikan sebagai suatu kerangka kerja yang dirancang agar sebuah tim mampu menyelesaikan berbagai permasalahan yang kompleks, serta mendorong tim tersebut untuk menghasilkan produk dengan nilai dan kualitas tinggi secara produktif maupun kreatif (Maulana & Prianto, 2023). Metode *Scrum* merupakan metode turunan dari metodologi *Agile*. Metode *Scrum* ini pada dasarnya dirancang sebagai sebuah solusi pemecahan masalah. Cara kerja *Scrum* dilakukan melalui penetapan target-target berskala kecil yang wajib diselesaikan dalam kurun waktu singkat guna mencapai sebuah tujuan akhir berskala besar (Giandari Maulani et al., 2024).

Metode *scrum* adalah teknik yang dibentuk dari beberapa tahapan proses mulai dari *product backlog* hingga *sprint retrospective*. Berikut adalah langkah-langkah yang umum dalam siklus *Scrum* (Magdalena et al., 2023).

1. *Product Backlog*

Tahapan yang pertama ini menjadi tanggung jawab *product owner* atau manajer. Secara sederhana, tahapan ini berisi daftar apa saja yang harus tim lakukan sesuai dengan skala prioritas Perusahaan.

Perwakilan Perusahaan bersama representasi *stakeholder* bertugas untuk terus menata ulang *product backlog*. Hal ini karena kondisi pasar akan selalu berubah, sehingga apa yang tidak esensial dalam *backlog* juga harus dihilangkan.

2. *Sprint Planning*

Dalam tahapan ini *sprint planning*, produk atau proyek teratas dalam *product backlog* kemudian disusun kembali menjadi *sprint backlog*. Tugas lain yang dilakukan tim *scrum* adalah menentukan bagaimana mereka akan menyelesaikan apa yang ada dalam *sprint backlog*.

3. *Sprint*

Setelah cara dan batas waktu pengerjaan ditentukan dalam *sprint planning*, tahapan metode *scrum* selanjutnya adalah melakukan *sprint*. Proses ini dilakukan melalui kegiatan *daily scrum*.

Daily scrum adalah saat dimana tim berkumpul dan bekerja untuk memastikan perkembangan produk terus berjalan. Dalam proses ini juga dikenal adanya peran *scrum master*.

Umumnya, tim yang terlibat dalam proses *scrum* adalah sekelompok orang. Di antara orang-orang tersebut, ada perwakilan yang berperan sebagai *scrum master*. Tugas *scrum master* adalah untuk membantu tim agar tetap fokus.

4. *Sprint Review*

Sprint review adalah tahapan *sprint* yang terakhir. Dalam proses ini, produk atau proyek yang dikerjakan harus sudah selesai dan siap digunakan. Kemudian produk tersebut akan direview kembali.

5. *Retrospective Process*

Scrum adalah metode yang sifatnya berulang. Proses yang dilakukan Perusahaan mulai dari pembuatan *backlog* hingga review diingat dan dilakukan kembali dalam proses *scrum* selanjutnya.

Setelah retrospektif *sprint* selesai, siklus *Scrum* dimulai kembali dengan perencanaan *sprint* berikutnya. Tim *Scrum* terus mengulang siklus ini hingga mencapai tujuan proyek yang ditetapkan atau hingga proyek selesai. Dengan demikian, siklus *Scrum* membantu dalam mengatur pekerjaan, meningkatkan transparansi, memungkinkan umpan balik yang cepat, dan memastikan proyek berjalan secara terus-menerus. *Sprint retrospective* dilakukan oleh *scrum master* bersama *product owner* dan *development team* dengan melakukan evaluasi keseluruhan hasil. *Product owner* akan memberikan masukan terkait kesesuaian hasil dengan kebutuhan sistem informasi yang telah ditetapkan (Nurmasani et al., 2024).

3.6 Raudhatul Athfal

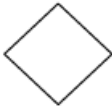
Raudhatul Athfal atau yang disingkat dengan sebutan RA, merupakan sebuah wujud institusi pendidikan anak usia dini pada ruang lingkup formal yang mengimplementasikan program pembelajaran dengan mengedepankan karakteristik nilai-nilai agama Islam bagi peserta didik yang berada dalam rentang usia 4 (empat) hingga 6 (enam) tahun (Sururiyah, 2023). RA merupakan lembaga pendidikan formal untuk anak usia 4-6 tahun yang berada di bawah naungan Departemen Agama, yang kini berubah nama menjadi Kementrian Agama. Lembaga ini dikelola secara profesional oleh guru-guru RA dalam wadah IGRA (Ikatan Guru Raudhatul Athfal). RA dan TK sama-sama merupakan pendidikan anak usia 4-6 tahun pada jalur formal yang diakui pemerintah dan tercantum dalam UU RI No 20 tahun 2003. Perbedaannya terletak pada pengelolaannya, di mana TK dikelola oleh DikNas, sedangkan RA dikelola oleh Departemen Agama. Setelah lulus dari RA atau TK, murid kemudian melanjutkan ke jenjang pendidikan lebih tinggi di atasnya, yaitu Sekolah Dasar atau yang sederajat (Amirudin et al., 2024).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa Raudhatul Athfal (RA) adalah lembaga pendidikan formal anak usia dini yang kedudukannya setara dengan Taman Kanak-Kanak (TK) sebagai fondasi persiapan menuju sekolah dasar, dengan karakteristik pembeda utama pada integrasi kurikulum berbasis nilai-nilai keislaman di bawah kewenangan Kementerian Agama.

3.7 Flowchart

Diagram alir (*flowchart*) merupakan instrumen penting dalam upaya penyempurnaan suatu proses kerja. Melalui penyajian representasi visual berupa grafik, *flowchart* memfasilitasi tim pengembang untuk memetakan berbagai elemen operasional sekaligus menganalisis korelasi yang terjalin antar setiap tahapan (Nurhadi et al., 2023).

Tabel 3.1 *Flowchart*

Simbol	Keterangan
	<i>Flow Symbol</i> yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain.
	<i>Terminator</i> Simbol yang menyatakan awal atau akhir suatu program.
	<i>Decision</i> Simbol yang digunakan untuk memilih proses yang akan dilakukan berdasarkan kondisi tertentu.
	<i>Manual Operation</i> Simbol yang digunakan untuk menunjukkan pengolahan yang tidak dilakukan oleh komputer.

Simbol	Keterangan
	<i>Document</i> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik atau <i>output</i> yang perlu dicetak.

3.8 Database

Basis data (*database*) merupakan sekumpulan data terstruktur yang disimpan, dikelola, dan diakses secara elektronik untuk memfasilitasi efisiensi manajemen informasi pada berbagai *platform*, termasuk aplikasi berbasis web. Secara struktural, basis data tersusun atas tabel-tabel yang memuat baris sebagai representasi suatu entitas, serta kolom yang mendeskripsikan atribut dari entitas tersebut. Keunggulannya terletak pada kemampuan dalam mengelola data yang besar dan kompleks secara terpusat, sehingga memungkinkan proses pencarian, penyaringan, dan manipulasi data secara cepat melalui penggunaan bahasa kueri terstruktur seperti SQL (*Structured Query Language*) (Prayoga et al., 2023). Sistem basis data pada umumnya memanfaatkan perangkat lunak manajemen basis data (Database Management System atau DBMS) guna mengatur, menyimpan, serta memanipulasi ketersediaan data. Beberapa contoh perangkat lunak DBMS yang lazim diimplementasikan dalam pengembangan sistem antara lain MySQL, PostgreSQL, Oracle Database, dan Microsoft SQL Server (Jurnaidi et al., 2025).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa database beserta perangkat lunak pengelolanya (DBMS) merupakan pondasi utama dalam perancangan sebuah sistem informasi, yang berperan krusial untuk memastikan seluruh data operasional instansi dapat tersimpan secara aman, terintegrasi, dan mudah diproses kembali.

3.9 Unified Modelling Language (UML)

Unified Modeling Language (UML) merupakan suatu bahasa standar yang difungsikan untuk merinci, menampilkan secara visual, mengonstruksi, serta mendokumentasikan berbagai *artifact* di dalam sebuah sistem perangkat lunak.

Artifact tersebut adalah sekumpulan informasi vital yang dapat berupa rancangan model, deskripsi tekstual, maupun wujud perangkat lunak itu sendiri yang dihasilkan selama tahapan pengembangan, di mana pemanfaatannya juga dapat diterapkan pada pemodelan bisnis maupun sistem di luar ruang lingkup perangkat lunak (Destriana et al., 2022).

Selain itu, UML bertindak sebagai bahasa pemodelan utama dalam siklus pengembangan sistem informasi yang mampu menggambarkan secara komprehensif berbagai dimensi proyek, mulai dari penjabaran kebutuhan, perancangan tata letak, tahapan implementasi, hingga proses pengujian. Untuk merepresentasikan ragam dimensi tersebut, UML menyediakan serangkaian diagram spesifik, di antaranya *Use Case Diagram* guna memetakan kebutuhan interaksi bisnis, *Class Diagram* untuk merancang struktur objek dan relasi antartabel, serta *Sequence Diagram* dan *Activity Diagram* yang dirancang untuk menjabarkan alur kerja atau proses bisnis secara rinci (Nur Fitrianingsih Hasan et al., 2023).

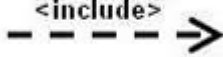
Berdasarkan uraian tersebut, dapat disimpulkan bahwa UML adalah bahasa pemodelan visual yang terstandarisasi dan menyeluruh, yang berperan krusial sebagai cetak biru (*blueprint*) untuk merancang, memvisualisasikan, serta mendokumentasikan seluruh elemen dan alur kerja perangkat lunak melalui berbagai jenis diagram spesifik.

3.9.1 *Use case diagram*

Use case diagram merupakan bagian dari beragam model diagram UML (*Unified Modelling Language*) yang memvisualisasikan korelasi interaksi di antara sistem dengan aktor. *Use Case* mampu menjabarkan kategori interaksi yang terjadi di antara pemakai sistem terhadap sistem yang digunakan (Siregar, 2025). Diagram *Use Case* adalah salah satu variasi dari sekian banyak bentuk diagram UML yang memaparkan atau menjelaskan keterkaitan hubungan antara sebuah sistem dengan aktornya (Ahmad et al., 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Use Case Diagram* merupakan representasi visual fungsionalitas sistem yang menitikberatkan pada pola interaksi antara pengguna (aktor) dan sistem guna mendefinisikan kebutuhan sistem secara jelas.

Tabel 3.2 *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menspesifikasi himpunan kepada pemakai untuk memainkan pada saat interaksi dengan <i>Use Case</i> .
	<i>Generalization</i>	Hubungan yang dimana letak antara objek kecil atau disebut anak yang berbagi perilaku dan struktur data dari objek yang ada diatasnya objek induk.
	<i>Use Case</i>	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja.
	<i>Include</i>	Include merupakan di dalam <i>Use Case</i> lain (<i>required</i>) atau pemanggilan <i>Use Case</i> oleh <i>Use Case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.
	<i>Association</i>	Asosiasi antara aktor dan <i>Use Case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.

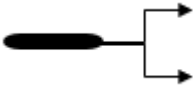
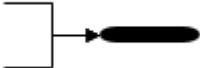
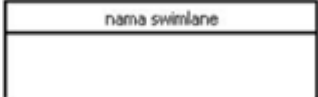
3.9.2 Activity diagram

Activity Diagram merupakan proyeksi visual dari sebuah alur kerja yang mendeskripsikan runtutan kegiatan di dalam sebuah sistem. Diagram tersebut mengilustrasikan tahapan kegiatan mulai dari titik mula sampai titik akhir, yang juga mencakup berbagai rute pengambilan keputusan yang berpotensi muncul di sepanjang proses tersebut. Pemodelan aktivitas ini merupakan bentuk pengembangan lanjutan dari konsep diagram alir konvensional, yang telah dilengkapi dengan kapasitas untuk memvisualisasikan jalannya proses paralel (Suwanda et al., 2025). *Activity Diagram* merupakan model yang memetakan beragam aliran kegiatan di dalam sistem yang tengah dibangun, mencakup bagaimana setiap aliran tersebut dipicu, *decision* yang berpeluang terbentuk, serta bagaimana rangkaian proses tersebut diselesaikan. Selain itu, *Activity Diagram* juga memiliki kemampuan untuk mengilustrasikan proses paralel yang berpotensi terjadi ketika beberapa eksekusi dijalankan secara bersamaan (Indra Gunawan, 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Activity Diagram* adalah pemodelan visual dari alur kerja sistem yang secara komprehensif memetakan urutan proses, titik pengambilan keputusan, hingga penanganan eksekusi yang berjalan paralel dari tahap awal hingga tahap penyelesaian.

Tabel 3.3 *Activity Diagram*

Simbol	Nama	Keterangan
	Status Awal	Menandakan titik awal atau dimulainya suatu alur aktivitas. Setiap diagram aktivitas harus memiliki satu titik awal.
	<i>Activity</i>	Mewakili suatu langkah, tindakan, atau tugas yang harus diselesaikan dalam proses. Nama dari aktivitas

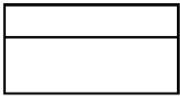
Simbol	Nama	Keterangan
		biasanya ditulis di dalam simbol ini.
	<i>Decision</i>	Digunakan untuk menunjukkan sebuah titik di mana ada pilihan atau keputusan yang harus dibuat. Biasanya diikuti oleh dua atau lebih aliran keluar, masing-masing dengan kondisi yang berbeda (disebut guard).
	<i>End Point</i>	Menandakan berakhirnya alur aktivitas. Suatu diagram bisa memiliki lebih dari satu titik akhir.
	<i>Control Flow</i>	Menghubungkan satu aktivitas ke aktivitas lain, menunjukkan urutan atau alur langkah-langkah dalam proses.
	<i>Fork Node</i>	Digunakan untuk memisahkan satu aliran aktivitas menjadi beberapa aliran yang berjalan secara paralel (bersamaan)
	<i>Join Node</i>	Digunakan untuk menggabungkan kembali beberapa aliran paralel menjadi satu aliran tunggal. Aktivitas setelah <i>join</i> hanya bisa dimulai setelah semua aliran yang masuk telah selesai.
	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

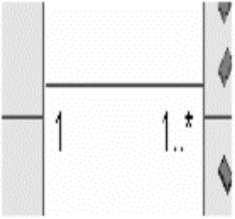
3.9.3 Class diagram

Diagram ini merupakan visualisasi dari struktur beserta deskripsi mengenai *class*, *package*, maupun objek yang memiliki keterkaitan satu sama lain, seperti halnya relasi pewarisan, asosiasi, serta jenis hubungan lainnya. Penggunaan *Class Diagram* berfungsi untuk merancang pemodelan dari berbagai kelas yang ada di dalam sebuah sistem, mencakup penjabaran atribut, metode operasional, hingga interaksi yang terjadi antar kelas tersebut. Pengaplikasian *Class Diagram* pada perancangan sebuah sistem informasi perpustakaan ditujukan untuk membentuk kerangka sistem yang terdefinisi dengan jelas, sistematis, serta mudah untuk dipahami. Pemodelan sistem yang dirancang dengan baik nantinya akan berfungsi sebagai fondasi yang kokoh dalam mendukung proses pengembangan sekaligus implementasi sistem informasi yang berjalan secara efektif (Ramdany et al., 2024).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Class Diagram* adalah pemodelan struktural yang memetakan elemen-elemen sistem beserta atribut dan korelasi antar objeknya, yang berperan penting sebagai kerangka dasar untuk membangun arsitektur perangkat lunak yang terarah dan tepat guna.

Tabel 3.4 *Class Diagram*

Simbol	Nama	Keterangan
	<i>Class</i>	<i>Class</i> menggambarkan keadaan (atribut/properti) dari suatu objek. Nama menggambarkan nama dari <i>Class</i> /objek. Atribut menggambarkan batasan dari nilai yang dapat dimiliki oleh <i>property</i> tersebut. <i>Method</i> menggambarkan implementasi dari layanan yang dapat diminta dari beberapa <i>object</i> dari <i>Class</i> , yang mempengaruhi <i>behaviour</i> .

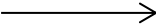
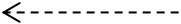
Simbol	Nama	Keterangan
	<i>Association</i>	Menggambarkan mekanisme komunikasi suatu objek dengan objek lainnya. Atau dapat juga menggambarkan ketergantungan antar kelas.
	<i>Multiplicity</i>	Menggambarkan banyaknya object yang terhubung satu dengan yang lainnya. Contoh : 1 Tepat satu 0..* Nol atau lebih 1..* Satu atau lebih 0..1 Nol atau satu

3.9.4 Sequence diagram

Menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang berupa message yang digambarkan terhadap waktu (Ramdany et al., 2024). Sequence diagram adalah diagram UML paling umum kedua yang merepresentasikan bagaimana objek berinteraksi dan bertukar pesan dari waktu ke waktu (over time). Sequence diagram menggambarkan bagaimana peristiwa (aktivitas) dalam use case yang dipetakan ke dalam operasi kelas objek dalam class diagram (Wayahdi & Ruziq, 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa Sequence diagram adalah pemodelan dinamis yang merepresentasikan alur komunikasi dan pertukaran pesan antar objek secara sekuensial, yang berfungsi penting untuk menerjemahkan skenario aktivitas pengguna ke dalam langkah operasional sistem yang lebih spesifik.

Tabel 3.5 *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Object</i>	Menggambarkan <i>object</i> apa saja yang terlibat dan berpartisipasi secara berurutan dengan mengirim dan/ atau menerima pesan.
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari system. Berpartisipasi secara berurutan dan/ atau menerima pesan.
	<i>Activation</i>	Menggambarkan hubungan antara <i>object</i> dengan <i>message</i> .
	<i>Message (call)</i>	Menggambarkan alur <i>message</i> yang merupakan kejadian objek pengirim <i>lifeline</i> ke objek penerima <i>lifeline</i> .
	<i>Message (return)</i>	Menggambarkan alur pengambilan <i>message</i> ke objek pemanggil dan tanda bahwa objek penerima telah menyelesaikan prosesnya.
	<i>Lifeline</i>	Menyatakan kehidupan suatu objek.

3.10 Laragon

Laragon merupakan perangkat lunak bersumber terbuka (*open source*) yang kompatibel dengan beragam sistem operasi, di mana fungsionalitasnya berperan sebagai server virtual atau yang lazim dikenal dengan istilah *localhost* (Andarsyah et al., 2022). Laragon merupakan perangkat lunak bebas biaya yang dapat

beroperasi pada berbagai sistem operasi sebagai peladen mandiri atau localhost. Platform ini menyajikan beragam layanan, perkakas, serta fitur terintegrasi yang mencakup *Apache*, *PHP Server*, *PhpMyAdmin*, *MySQL*, *Memcached*, *Redis*, *Composer*, *Xdebug*, *Cmder*, hingga *Laravel* (Lukman et al., 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa penggunaan laragon merupakan lingkungan pengembangan lokal (local environment) utama dalam fase perancangan sebuah sistem informasi, yang berperan krusial untuk menyediakan ekosistem peladen terpadu guna memastikan seluruh proses pengkodean, pengelolaan basis data, dan pengujian aplikasi dapat berjalan lancar sebelum diimplementasikan ke server publik.

3.10.1 MySQL

MySQL merupakan sistem manajemen basis data relasional (RDBMS) yang beroperasi sebagai server guna memfasilitasi akses multi-pengguna terhadap berbagai basis data. Seluruh data pada MySQL tersebut diorganisasikan ke dalam objek basis data yang diklasifikasikan sebagai tabel (Sastradipraja & Munawar, 2022). MySQL merupakan basis data yang lazim diimplementasikan oleh para pengembang web, dikarenakan sistem ini dinilai lebih stabil dan sangat tangguh sebagai media penyimpanan data dibandingkan dengan basis data alternatif lainnya (Putri et al., 2023). MySQL merupakan sistem manajemen basis data (*database management system*) terkemuka yang dioperasikan menggunakan perintah dasar SQL (*Structured Query Language*). Sebagai sebuah DBMS bersumber terbuka (open-source), MySQL didistribusikan di bawah dua jenis lisensi utama, yakni perangkat lunak bebas (*free software*) serta perangkat lunak berpemilik dengan batasan penggunaan (*shareware*). Dengan demikian, MySQL bertindak sebagai peladen basis data tak berbayar di bawah naungan lisensi GNU *General Public License* (GPL), yang mengizinkan pemanfaatannya untuk kebutuhan personal maupun komersial tanpa adanya kewajiban pembayaran lisensi (Namruddin et al., 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa MySQL merupakan sistem manajemen basis data relasional yang sangat ideal dalam perancangan aplikasi ini,

yang berperan krusial untuk memastikan seluruh data akademik instansi dapat terkelola secara terstruktur, stabil, dan aman tanpa membebani biaya lisensi perangkat lunak.

3.11 *React JS*

React adalah *open-source library JavaScript* deklaratif, efisien dan fleksibel untuk membangun antarmuka pengguna. *React* memungkinkan untuk membuat *user interface* yang kompleks dengan set kode kecil yang terisolasi yang disebut "komponen". *React JS* ini digunakan untuk menangani lapisan tampilan dalam aplikasi satu halaman dan pengembangan *mobile application*. *React JS* dikelola oleh facebook, instagram, komunitas pengembangan korporasi. *React* berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur yang paling mencolok adalah *JSX*, Komponen *Stateful*, Model Objek Dokumen Virtual (Iswari, 2025). *React JS* adalah *library Javascript* yang bersifat *open-source* yang mayoritas digunakan untuk membangun *User Interface* (UI) secara spesifik untuk satu page dalam aplikasi. *React JS* digunakan sebagai salah satu *framework* untuk membuat bagian *front-end* dari sebuah aplikasi. *React JS* berfungsi untuk mengatur Layer pada tampilan untuk *Desktop* maupun versi *mobile* Aplikasi. Dengan *React*, *user* bisa membuat *Reusable Component* atau jenis komponen yang bisa digunakan kembali tanpa harus membangunnya dari awal lagi. *React JS* dapat mendesain tampilan sederhana untuk setiap level dalam aplikasi, sehingga dapat digunakan untuk membuat dan mengembangkan pembuatan aplikasi berbasis web (Sulistyorini et al., 2022).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *React JS* adalah pustaka (*library*) *JavaScript* sumber terbuka yang dikhususkan untuk membangun lapisan antarmuka pengguna (*front-end*) yang cepat, efisien, dan interaktif melalui penerapan arsitektur komponen yang dapat digunakan kembali (*reusable component*) pada pengembangan aplikasi web maupun perangkat bergerak.

3.12 Framework

Framework merupakan sebuah elemen pemrograman yang bersifat *re-use* atau siap untuk diaplikasikan kembali. Hal ini memungkinkan seorang programmer agar tidak perlu menuliskan baris skrip yang serupa untuk mengerjakan instruksi yang berulang. Sebagai contoh, apabila seorang programmer hendak merancang antarmuka web yang menampilkan sekumpulan data melalui fitur paginasi (*paging*), maka *framework* sudah melengkapi fungsi tersebut secara bawaan. Dalam praktiknya, programmer hanya perlu menerapkan fungsi tersebut dengan memanfaatkan kaidah coding yang telah dibakukan oleh setiap framework (Sinlae et al., 2024). Di sisi lain, *framework* juga bisa didefinisikan sebagai sekumpulan script (khususnya berupa *class* maupun *function*) yang memfasilitasi developer atau programmer dalam merespons berbagai persoalan pengkodean, seperti proses koneksi menuju *database*, eksekusi variabel, pengelolaan file, serta kebutuhan teknis lainnya, yang mana hal ini membuat fokus kerja developer menjadi lebih terarah dan proses pengembangan aplikasi berjalan lebih cepat (Bardadi, 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *framework* adalah kerangka kerja berupa koleksi kode dan fungsi siap pakai yang dirancang untuk mempermudah, mempercepat, serta meningkatkan efisiensi programmer dalam merancang dan membangun sebuah aplikasi.

3.13 Laravel

Laravel merupakan *framework* PHP terkemuka yang dirancang oleh Taylor Otwell. *Framework* web berbasis PHP ini bersifat *open source* serta dapat diakses secara gratis, yang secara khusus ditujukan untuk membangun aplikasi web dengan mengadopsi arsitektur *MVC* (Sinlae et al., 2024). Laravel merupakan kerangka kerja dari bahasa pemrograman *Hypertext Preprocessor* (PHP) yang difokuskan pada proses perancangan aplikasi berbasis web dengan mengimplementasikan pola *Model View Controller* (MVC). Kerangka kerja ini diciptakan oleh Taylor Otwell dan diluncurkan untuk pertama kalinya pada tanggal 9 Juni 2011. Laravel didistribusikan di bawah lisensi *open source*, yang berarti pengguna dapat memanfaatkannya secara bebas tanpa dikenakan kewajiban pembayaran. Portal

situs resmi untuk mengakses framework Laravel ini dapat ditemukan pada tautan <https://laravel.com>. Berbagai instrumen modern pada Laravel yang berperan besar dalam memudahkan developer selama proses pembuatan aplikasi mencakup *Bundles*, *Eloquent ORM (Object-Relational Mapping)*, *Query Builder*, *Application Logic*, *Reverse Routing*, *Resource Controller*, *Class Auto Loading*, *View Composers*, *Blade*, *IoC Containers*, *Migration*, *Database Seeding*, *Unit Testing*, *Automatic Pagination*, *Form Request*, serta *Middleware* (Ardiansyah, 2023).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa Laravel adalah *framework* PHP bersifat *open source* dengan standar arsitektur *MVC* yang dilengkapi oleh serangkaian fitur modern yang komprehensif, guna mempermudah developer dalam merancang dan mengembangkan aplikasi web secara efisien dan terstruktur.

3.14 Visual Studio Code (VS Code)

Visual Studio Code merupakan sebuah teks editor yang ringan sekaligus tangguh yang dikembangkan oleh Microsoft untuk kebutuhan sistem operasi *multiplatform*, yang berarti perangkat lunak ini dapat diakses pada lingkungan *Linux*, *Mac*, maupun *Windows*. Teks editor ini memiliki dukungan bawaan secara langsung terhadap bahasa pemrograman *Javascript*, *Typescript*, maupun *Node Js*, serta mampu memfasilitasi berbagai bahasa pemrograman lain melalui penambahan *plugin* yang dapat diunduh melalui *marketplace Visual Studio Code*, seperti halnya *C++*, *C#*, *Python*, *Go*, *Java*, hingga *PHP* (Ummy Gusti Salamah, 2021). Terdapat ragam fitur fungsional yang disematkan di dalam *Visual Studio Code*, di antaranya mencakup *Intellisense*, *Git Integration*, *Debugging*, serta dukungan fitur ekstensi yang berfungsi untuk meningkatkan kapasitas kinerja dari teks editor tersebut. Kelengkapan fitur tersebut akan mengalami penambahan secara berkelanjutan sejalan dengan pembaruan versi dari *Visual Studio Code*. Siklus pembaruan versi *Visual Studio Code* ini secara rutin dilaksanakan setiap bulan, dan aspek inilah yang menjadi daya pembeda utama antara VS Code dibandingkan dengan teks editor jenis lainnya. Perangkat teks editor VS Code ini juga mengusung lisensi *open source*, di mana akses kode sumbernya dapat ditinjau secara publik sehingga memungkinkan para pengguna untuk turut berkontribusi dalam proses

pengembangannya. Basis kode sumber dari VS Code tersebut juga dapat diakses secara langsung melalui *platform Github* (Aldi & Aji, 2025).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Visual Studio Code* adalah teks editor *multiplatform* bersifat *open source* buatan Microsoft yang ringan, tangguh, dan selalu diperbarui secara rutin, serta didukung oleh ekosistem ekstensi yang kaya untuk memfasilitasi penulisan berbagai jenis bahasa pemrograman.

3.15 Black Box Testing

Black box testing merupakan pendekatan evaluasi perangkat lunak yang dieksekusi tanpa perlu meninjau struktur bagian dalam maupun logika operasional dari sistem yang sedang diuji. Proses pengujian ini secara khusus memusatkan perhatian pada aspek fungsionalitas sistem, sehingga sama sekali tidak membutuhkan pemahaman mendetail mengenai bagaimana mekanisme sistem tersebut beroperasi di balik layer (Mardiati & Saputra, 2025). *Black Box Testing* merupakan sebuah teknik pengujian perangkat lunak yang berhubungan langsung dengan spesifikasi fungsi tanpa perlu melakukan pemeriksaan terhadap baris kode atau kerangka internal dari program yang bersangkutan. Pada penerapan *Black Box Testing*, evaluasi hanya ditujukan pada sisi fungsionalitas, antarmuka pengguna, serta alur kerja sistem tanpa mempedulikan rincian dari implementasi kode maupun skrip yang tertanam di dalam perangkat lunak tersebut. Pelaksanaan pengujian ini difokuskan pada spesifikasi fungsi perangkat lunak melalui penetapan berbagai kondisi input yang dimanfaatkan untuk meninjau respons fungsionalitasnya secara langsung. Rangkaian proses uji ini meliputi tahapan eksekusi data pengujian sekaligus tahap evaluasi terhadap performa operasional dari perangkat lunak tersebut. Oleh karena itu, di dalam metode *Black Box Testing*, fokus evaluasi secara dominan diarahkan pada bagian antarmuka atau proyeksi tampilan luar perangkat lunak, yakni dengan menguji mekanisme input beserta output tanpa harus menganalisis rincian dari struktur internalnya (Banjarnahor & Sinuraya, 2025).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *Black Box Testing* adalah metode pengujian sistem yang berfokus penuh pada validasi fungsionalitas,

antarmuka, dan alur input-output aplikasi dari sudut pandang pengguna, tanpa perlu membedah atau memahami struktur kode pemrograman di dalamnya.

3.16 *System Usability Scale (SUS)*

Metode *System Usability Scale (SUS)* yang digagas oleh John Brooke pada tahun 1986 merupakan instrumen evaluasi Usabilitas yang dikenal andal, praktis, dan banyak digunakan untuk menilai berbagai platform digital. Pengukuran dengan instrumen ini dilakukan melalui penyajian 10 pernyataan standar yang disusun secara bergantian antara kalimat bernada positif dan negatif. Responden kemudian memberikan penilaian terhadap pernyataan tersebut menggunakan acuan skala Likert 5 poin. Keunggulan utama pendekatan ini terletak pada kemampuannya merangkum hasil evaluasi menjadi satu nilai akhir dengan rentang 0 sampai 100, yang selanjutnya dikelompokkan berdasarkan tingkat kelayakan sistem melalui kriteria *Adjective Rating* dan *Acceptability*. Dalam penerapannya, nilai 68.0 ditetapkan sebagai standar batas minimal untuk menentukan apakah sistem yang dievaluasi sudah masuk dalam kategori acceptable (layak pakai) atau justru masih membutuhkan perbaikan lebih lanjut (Oktavidiyanto et al., 2026). Metode SUS memiliki tiga karakteristik utama yang mendasarinya. Pertama, instrumen ini hanya terdiri dari sepuluh pertanyaan ringkas, sehingga memudahkan responden untuk memahami dan menyelesaikannya dalam waktu singkat. Kedua, metode ini bersifat teknologi agnostic, yang berarti sangat fleksibel dan dapat diterapkan secara luas untuk mengevaluasi berbagai jenis antarmuka, baik itu website, aplikasi smartphone, maupun platform digital lainnya. Ketiga, hasil pengukuran dalam metode ini dikonversi ke dalam rentang skor 0 hingga 100, sebuah standar nilai yang sangat umum dan mudah dipahami secara relatif (Ginting et al., 2025).

Berdasarkan uraian tersebut, dapat disimpulkan bahwa *System Usability Scale (SUS)* adalah instrumen evaluasi yang praktis, cepat, dan fleksibel untuk mengukur tingkat kemudahan dan kelayakan suatu sistem digital dari sudut pandang pengguna, melalui perhitungan skor akhir yang objektif tanpa membebani responden dengan pertanyaan yang rumit.

BAB IV

METODE PENELITIAN

4.1 Subjek dan Objek Penelitian

4.1.1 Subjek Penelitian

Berdasarkan kondisi operasional saat ini, RA Al-Islam Gunungcupu membutuhkan sebuah sistem yang dapat menghubungkan seluruh pihak yang terlibat dalam kegiatan sekolah sehari-hari. Oleh karena itu, subjek dalam penelitian ini ditentukan berdasarkan orang-orang yang secara langsung mengelola dan membutuhkan data akademik di sekolah tersebut. Pemilihan pihak-pihak ini sangat penting karena keluhan dan kebutuhan merekalah yang nantinya akan disusun menjadi daftar tugas pembuatan sistem (*Product Backlog*) dalam metode *Scrum*. Untuk memenuhi kebutuhan saling terhubung tersebut, pengguna aplikasi dibagi menjadi tiga hak akses dengan tugas masing-masing, yaitu:

- a. Admin bertindak sebagai pengelola utama sistem yang memiliki wewenang penuh terhadap manajemen data master, seperti pengaturan tahun pelajaran, kalender pendidikan, pendataan akun pengguna, hingga alokasi kelas.
- b. Guru berperan sebagai subjek operasional harian yang menggunakan sistem untuk memasukkan data presensi siswa, mengelola informasi kelas, serta memantau perkembangan akademik anak didiknya secara berkala.
- c. Orang Tua Siswa berkedudukan sebagai penerima informasi akhir yang membutuhkan akses transparansi terhadap data absensi dan perkembangan akademik anak mereka di sekolah tanpa harus datang langsung ke institusi.

4.1.2 Objek Penelitian

Untuk menyelesaikan masalah pencatatan yang masih menggunakan buku induk fisik, RA Al-Islam Gunungcupu sangat membutuhkan sebuah wadah digital yang bisa menyimpan seluruh data sekolah agar tidak mudah hilang dan mudah dicari. Oleh karena itu, objek utama dalam penelitian ini adalah pembuatan sistem

informasi akademik berbasis web. Sejalan dengan tujuan menghasilkan produk yang benar-benar bisa dipakai, pembuatan aplikasi ini difokuskan pada fitur-fitur yang langsung berguna bagi sekolah, seperti pengelolaan data pengguna, tahun pelajaran, kalender pendidikan, kelas, siswa, dan absensi. Selain kelengkapan data, pihak sekolah juga membutuhkan solusi agar guru tidak perlu bekerja dua kali dalam memasukkan nilai siswa. Untuk mencegah terjadinya pendataan yang berulang, sistem yang dibangun ini dirancang agar dapat terhubung dengan data dari Rapor Digital Madrasah (RDM) yang sudah ada sebelumnya. Keputusan untuk menggabungkan database dari luar ini diambil agar proses penilaian anak di sekolah menjadi lebih cepat, akurat, dan datanya terpusat di satu tempat.

4.2 Bahan dan Alat

4.2.1 Bahan Penelitian

Bahan penelitian yang diproses dan dianalisis dalam pengembangan sistem ini dikategorikan menjadi dua jenis sumber, yaitu:

- a. Data Primer yang diperoleh melalui observasi langsung terhadap alur kerja akademik di RA Al-Islam Gunungcupu serta wawancara mendalam dengan kepala sekolah dan tenaga pendidik. Data ini digunakan untuk mengidentifikasi akar permasalahan operasional dan merumuskan spesifikasi fungsional sistem.
- b. Data Sekunder yang meliputi dokumen fisik sekolah seperti format buku induk siswa, kurikulum yang berlaku, serta literatur pendukung berupa jurnal penelitian terbitan lima tahun terakhir yang relevan dengan metode *Scrum* dan sistem informasi akademik.

4.2.2 Alat Penelitian

Berikut adalah bahan dan alat (*hardware* dan *software*) yang digunakan peneliti selama berlangsungnya penelitian.

- a. Perangkat keras (*hardware*)

Device name: LAPTOP-5TLL6CKP

Processor: 12th Gen Intel(R) Core(TM) i3-1215U (1.20 GHz)

Installed RAM: 8,00 GB (7,73 GB usable)

Device ID: 11EBAB65-BBEA-40A3-BD06-351E87EE8A73

Product ID: 00356-24781-39052-AAOEM

System type: 64-bit operating system, x64-based processor

b. Perangkat lunak (*software*)

1. Sistem operasi: windows 11

Edition: Windows 11 Home Single Language

Version: 25H2

Installed on: 01/12/2024

OS build: 26200.8039

Serial number: PF577V67

Experience: Windows Feature Experience Pack 1000.26100.300.0

2. Bahasa pemrograman: PHP.

3. *Framework*: Laravel 12.

4. Library frontend: React JS.

5. Komponen UI: *Shadcn UI*.

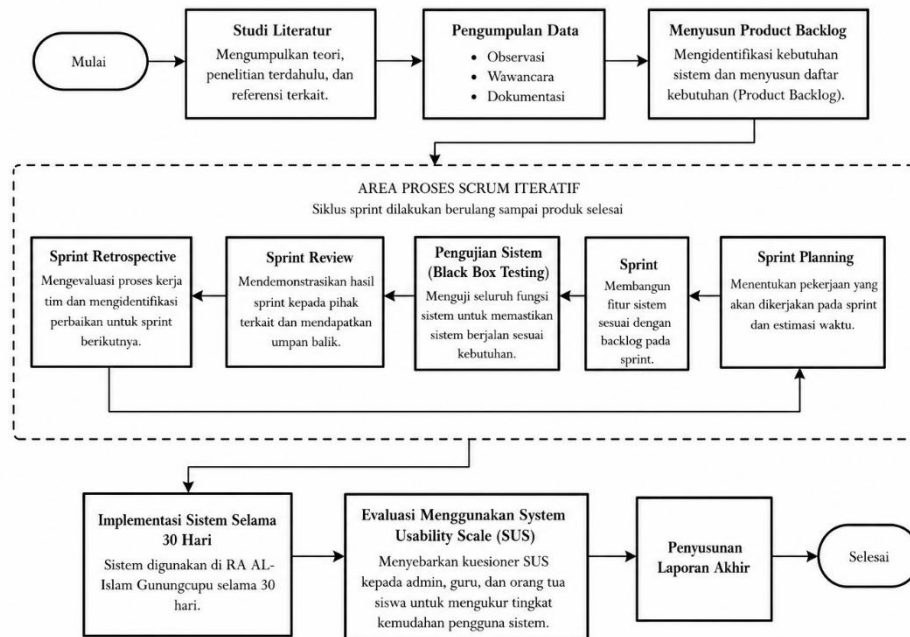
6. *Database*: mysql.

7. *Web server*: Laragon.

8. *Tools*: Visual Studio Code, Draw.io.

4.3 Bagan Alir Penelitian

Dalam perancangan dan pembuatan sistem informasi akademik berbasis web di RA Al-Islam Gunungcupu, diperlukan sebuah tahapan kerja yang sistematis agar proses pengembangan dapat berjalan terarah sesuai dengan metode *Scrum* yang digunakan. Rangkaian tahapan tersebut mencakup proses studi literatur, pengumpulan data kebutuhan dari pihak sekolah, pengelolaan daftar tugas, pelaksanaan siklus pengerjaan melalui tahapan *Sprint*, penulisan kode program, hingga pengujian akhir dan penerapan sistem. Keseluruhan langkah-langkah penelitian tersebut divisualisasikan ke dalam sebuah alur kerja yang digambarkan pada Gambar 4.1.



Gambar 4.1 Bagan Alir Penelitian
(Sumber: Data Pribadi)

1. Mulai

Tahap ini merupakan awal pelaksanaan penelitian yang diawali dengan identifikasi permasalahan pada proses pengelolaan akademik di RA Al-Islam Gunungcupu. Peneliti mengidentifikasi berbagai kendala yang terjadi pada sistem yang sedang berjalan, khususnya dalam pengelolaan data akademik yang masih dilakukan secara manual menggunakan dokumen fisik. Hasil identifikasi tersebut menjadi dasar dalam menentukan tujuan penelitian serta kebutuhan pengembangan sistem informasi akademik berbasis web.

2. Studi Literatur

Tahap studi literatur dilakukan untuk memperoleh landasan teori yang mendukung penelitian. Peneliti mempelajari berbagai referensi berupa buku, jurnal ilmiah, dan penelitian terdahulu yang berkaitan dengan sistem informasi akademik, metode *Scrum*, *Unified Modeling Language* (UML), *Laravel*, *React JS*, *Black Box Testing*, serta *System Usability Scale* (SUS). Kajian literatur digunakan sebagai dasar dalam

menentukan metode pengembangan sistem dan mendukung proses analisis selama penelitian.

3. Pengumpulan Data

Tahap pengumpulan data dilakukan untuk memperoleh informasi mengenai kondisi nyata di RA Al-Islam Gunungcupu sehingga sistem yang dikembangkan sesuai dengan kebutuhan pengguna. Pengumpulan data dilakukan melalui observasi, wawancara, dan dokumentasi. Ketiga teknik tersebut digunakan secara saling melengkapi untuk memahami proses bisnis yang sedang berjalan, mengidentifikasi permasalahan, serta menentukan kebutuhan sistem yang akan dikembangkan.

a. Observasi

Observasi dilakukan untuk memperoleh informasi mengenai kondisi dan proses pengelolaan data akademik yang sedang berjalan di RA Al-Islam Gunungcupu. Hasil observasi digunakan untuk memahami alur kegiatan akademik secara langsung, mulai dari pendataan siswa, pencatatan absensi, hingga penyampaian informasi akademik kepada orang tua siswa. Berdasarkan hasil observasi tersebut, proses sistem manual yang sedang berjalan kemudian divisualisasikan dalam bentuk *flowchart* sistem manual yang sedang berjalan untuk menggambarkan urutan aktivitas dan hubungan antar pihak yang terlibat secara lebih sistematis. Hasil *flowchart* tersebut selanjutnya digunakan sebagai dasar dalam mengidentifikasi permasalahan sistem yang berjalan dan menentukan kebutuhan sistem yang akan dikembangkan.

b. Wawancara

Wawancara dilakukan kepada Kepala RA dan guru untuk memperoleh informasi mengenai kendala yang dihadapi dalam pengelolaan data akademik serta kebutuhan pengguna terhadap sistem informasi akademik yang akan dikembangkan.

c. Dokumentasi

Dokumentasi dilakukan dengan mengumpulkan dokumen pendukung, seperti buku induk siswa, buku absensi, dan buku penghubung guru dengan orang tua. Dokumen

tersebut digunakan sebagai bahan pendukung dalam menganalisis proses pengelolaan data akademik, struktur data, dan kebutuhan fungsional sistem.

4. Menyusun *Product Backlog*

Berdasarkan hasil observasi, wawancara, dan dokumentasi, seluruh kebutuhan sistem dianalisis kemudian disusun ke dalam *Product Backlog* sebagai daftar kebutuhan fungsional yang akan dikembangkan menggunakan metode *Scrum*. Penyusunan *Product Backlog* diawali dengan identifikasi kebutuhan sistem, kemudian divisualisasikan menggunakan *Use Case Diagram* untuk menggambarkan interaksi antara aktor dengan sistem. Selanjutnya setiap kebutuhan dituangkan dalam bentuk *user story* yang dilengkapi dengan aktor, tingkat prioritas, serta estimasi waktu pengerjaan sebagai dasar penyusunan *Sprint Planning*.

5. *Sprint Planning*

Sprint Planning merupakan tahap perencanaan pengembangan sistem berdasarkan *Product Backlog* yang telah disusun sebelumnya. Pada tahap ini setiap kebutuhan dipilih dan dikelompokkan ke dalam beberapa *Sprint* sesuai tingkat prioritas serta estimasi waktu pengerjaan. Pembagian *Sprint* dilakukan agar proses pengembangan berlangsung secara bertahap sehingga setiap *Sprint* memiliki target pekerjaan yang jelas dan dapat diselesaikan dalam waktu yang telah ditentukan.

6. *Sprint*

Sprint merupakan tahap pelaksanaan pengembangan sistem berdasarkan *Sprint Backlog* yang telah direncanakan. Pada setiap *Sprint* dilakukan proses perancangan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai acuan implementasi. Selanjutnya dilakukan pembangunan fitur melalui penulisan kode program menggunakan Laravel sebagai *framework backend* dan React JS sebagai *library frontend*. Setiap *Sprint* menghasilkan sekumpulan fitur yang siap diuji sebelum dipresentasikan kepada pengguna.

7. Pengujian Sistem (*Black Box Testing*)

Setelah proses pengembangan pada setiap *Sprint* selesai dilakukan, fitur-fitur yang dihasilkan diuji menggunakan metode *Black Box Testing*. Pengujian bertujuan untuk memastikan bahwa seluruh fungsi pada *Sprint* tersebut telah berjalan sesuai dengan kebutuhan fungsional tanpa memperhatikan struktur kode program. Hasil pengujian menjadi dasar untuk menentukan apakah fitur yang dikembangkan telah layak dilanjutkan ke tahap *Sprint Review* atau masih memerlukan perbaikan.

8. *Sprint Review*

Sprint Review dilakukan dengan mendemonstrasikan hasil pengembangan *Sprint* yang telah melalui proses pengujian kepada pihak RA Al-Islam Gunungcupu. Pada tahap ini pengguna memberikan masukan mengenai kesesuaian fungsi sistem dengan kebutuhan operasional sekolah. Umpan balik yang diperoleh digunakan sebagai bahan evaluasi untuk penyempurnaan pengembangan pada *Sprint* berikutnya.

9. *Sprint Retrospective*

Sprint Retrospective merupakan tahap evaluasi terhadap proses pelaksanaan *Sprint* yang telah selesai. Peneliti mengevaluasi proses pengembangan, pembagian pekerjaan, manajemen waktu, serta hambatan teknis yang ditemui selama *Sprint* berlangsung. Hasil evaluasi tersebut menjadi dasar untuk meningkatkan kualitas proses pengembangan pada *Sprint* berikutnya. Siklus *Sprint* akan terus berulang hingga seluruh *Product Backlog* berhasil dikembangkan menjadi sistem informasi akademik yang utuh.

10. Implementasi Sistem 30 Hari

Setelah sistem informasi akademik dinyatakan bebas dari kesalahan fungsi (*bug*) melalui pengujian lokal, tahapan selanjutnya adalah melakukan publikasi sistem ke *server* daring (*hosting*). Migrasi ke server publik tersebut mutlak dilakukan agar aplikasi web dapat diakses secara luas melalui jaringan internet menggunakan perangkat masing-masing pengguna. Setelah sistem berstatus aktif secara *online*, sistem tersebut mulai diterapkan secara nyata di RA Al-Islam Gunungcupu untuk menggantikan fungsi pencatatan buku induk manual. Proses penerapan tersebut

dijalankan secara langsung selama 30 hari operasional. Keputusan pemberian waktu satu bulan dirancang secara spesifik untuk memberikan ruang adaptasi yang memadai bagi para pengguna utama sistem, yaitu admin, guru, dan orang tua siswa, agar mereka benar-benar terbiasa menggunakan dan mengakses informasi dari aplikasi tersebut secara mandiri.

11. Evaluasi Menggunakan *System Usability Scale* (SUS)

Setelah sistem digunakan selama masa implementasi, dilakukan evaluasi menggunakan metode *System Usability Scale* (SUS). Evaluasi dilakukan dengan menyebarkan kuesioner kepada admin, guru, dan orang tua siswa sebagai pengguna sistem untuk mengukur tingkat kemudahan penggunaan aplikasi. Hasil pengukuran kemudian dianalisis sebagai dasar dalam menilai tingkat *usability* sistem yang telah dikembangkan

12. Penyusunan Laporan Akhir

Tahap penyusunan laporan akhir dilakukan sebagai bentuk dokumentasi seluruh rangkaian penelitian yang telah dilaksanakan, mulai dari identifikasi masalah, pengumpulan data, penyusunan *Product Backlog*, pelaksanaan *Sprint*, pengujian pada setiap *Sprint*, implementasi sistem, hingga evaluasi menggunakan metode SUS. Seluruh hasil penelitian kemudian disusun secara sistematis ke dalam laporan skripsi sesuai dengan kaidah penulisan ilmiah.

11. Selesai

Tahap ini menandai berakhirnya seluruh proses penelitian, mulai dari perancangan, pembangunan, implementasi, hingga evaluasi Sistem Informasi Akademik berbasis web di RA Al-Islam Gunungcupu. Hasil penelitian diharapkan mampu memberikan solusi terhadap permasalahan pengelolaan data akademik yang sebelumnya masih dilakukan secara manual.

BAB V

HASIL DAN PEMBAHASAN

5.1 Pengumpulan Data

Pengumpulan data merupakan tahapan awal yang dilakukan untuk memperoleh informasi mengenai kondisi nyata proses pengelolaan data akademik di RA Al-Islam Gunungcupu. Data yang diperoleh menjadi dasar dalam proses analisis kebutuhan sistem sehingga fitur-fitur yang dikembangkan dapat disesuaikan dengan kebutuhan pengguna. Sesuai dengan metode penelitian yang telah dijelaskan pada Bab IV, pengumpulan data dilakukan melalui observasi, wawancara, dan dokumentasi. Hasil dari ketiga teknik pengumpulan data tersebut digunakan sebagai acuan dalam penyusunan *Product Backlog* pada metode *Scrum*.

5.1.1 Hasil observasi

Observasi dilakukan secara langsung di RA Al-Islam Gunungcupu untuk mengetahui proses pengelolaan data akademik yang sedang berjalan. Kegiatan observasi difokuskan pada aktivitas pencatatan data siswa, pengelolaan absensi, penyimpanan data akademik, serta penyampaian informasi kepada pihak yang berkepentingan. Melalui observasi ini, peneliti memperoleh gambaran mengenai alur kerja yang diterapkan oleh sekolah beserta kendala yang muncul selama proses pengelolaan data berlangsung.

5.1.1.1 Sistem manual yang sedang berjalan

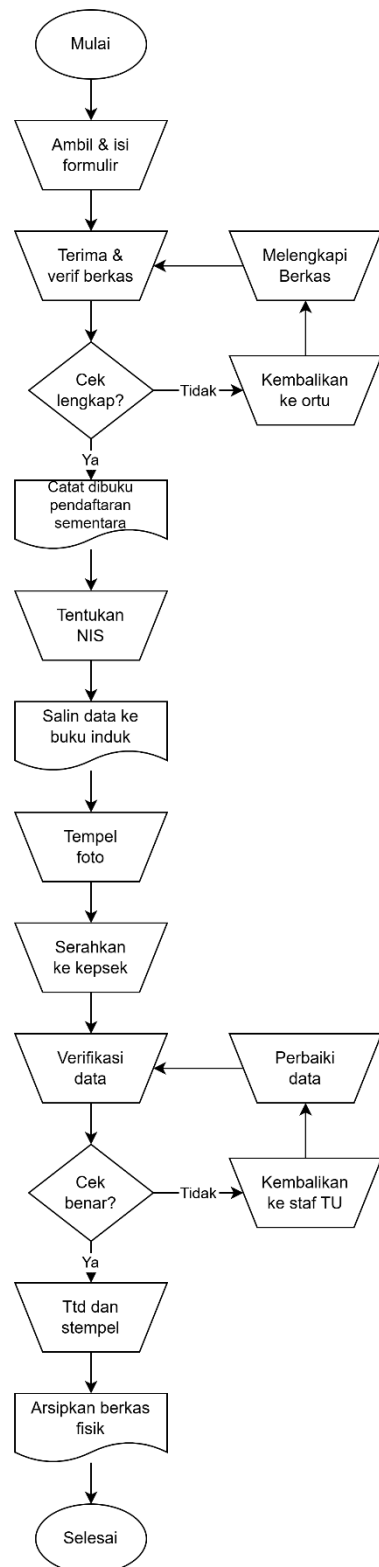
Berdasarkan hasil observasi, pengelolaan data akademik di RA Al-Islam Gunungcupu masih mengandalkan pencatatan secara manual. Data identitas siswa dicatat ke dalam buku induk sebagai arsip utama sekolah, sedangkan data kehadiran siswa dicatat setiap hari pada buku absensi oleh guru. Selain itu, penyampaian informasi akademik kepada orang tua masih dilakukan melalui komunikasi langsung dengan guru apabila orang tua membutuhkan informasi mengenai penilaian anak.

Pola kerja tersebut telah digunakan sebagai prosedur operasional sekolah selama ini. Namun, seluruh data masih tersimpan dalam bentuk dokumen fisik sehingga proses pengelolaan informasi sangat bergantung pada pencatatan manual. Akibatnya, pencarian data membutuhkan waktu yang relatif lebih lama dan penyampaian informasi kepada orang tua belum dapat dilakukan secara cepat. Untuk memberikan gambaran mengenai proses bisnis yang sedang berjalan, hasil observasi divisualisasikan ke dalam satu *flowchart* sistem manual yang sedang berjalan. *Flowchart* tersebut menggambarkan keterkaitan antara proses pendataan siswa, pencatatan absensi, serta penyampaian informasi akademik kepada orang tua yang masih dilakukan secara manual di RA Al-Islam Gunungcupu. Penyusunan *flowchart* sistem yang sedang berjalan bertujuan untuk memvisualisasikan proses bisnis sehingga permasalahan pada sistem manual dapat diidentifikasi sebelum dilakukan perancangan sistem usulan. Pendekatan ini sejalan dengan penelitian (Ruswa Dwipa, 2021), yang menyatakan bahwa analisis sistem berjalan dilakukan sebagai dasar dalam menentukan kebutuhan sistem.

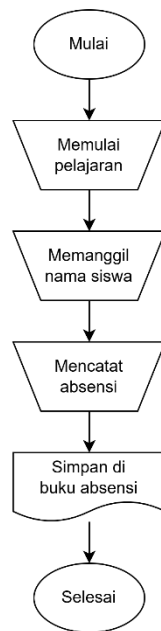
5.1.1.2 *Flowchart* sistem manual yang sedang berjalan

Flowchart sistem yang sedang berjalan disusun berdasarkan hasil observasi terhadap proses akademik di RA Al-Islam Gunungcupu. *Flowchart* ini menggambarkan alur kerja yang masih dilakukan secara manual mulai dari proses pendataan siswa, pencatatan absensi, hingga penyampaian informasi akademik kepada orang tua siswa. Seluruh proses tersebut melibatkan beberapa pihak, yaitu Orang Tua, Staf Tata Usaha, Guru, dan Kepala Sekolah. *Flowchart* sistem yang sedang berjalan dapat dilihat pada Gambar 5.1.

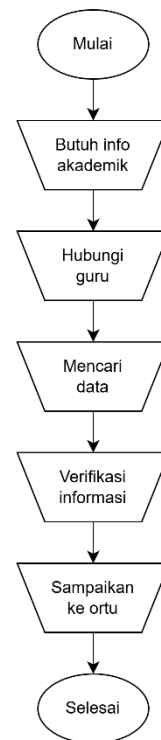
PROSES 1: PENDAFTARAN



PROSES 2: ABSENSI



PROSES 3: INFO AKADEMIK

Gambar 5.1 *Flowchart* Sistem Manual yang sedang berjalan

Berdasarkan flowchart pada Gambar 5.1, proses akademik yang sedang berjalan di RA Al-Islam Gunungcupu terdiri atas tiga proses utama yang saling berkaitan, yaitu proses pendataan siswa, proses absensi, dan proses penyampaian informasi akademik kepada orang tua siswa. Ketiga proses tersebut dijelaskan sebagai berikut. Proses pertama diawali dengan pendataan siswa baru. Orang tua mengambil dan mengisi formulir pendaftaran kemudian menyerahkan berkas persyaratan kepada Staf Tata Usaha untuk dilakukan pemeriksaan kelengkapan dokumen. Apabila terdapat berkas yang belum lengkap, dokumen dikembalikan kepada orang tua untuk dilengkapi kembali. Sebaliknya, apabila seluruh persyaratan telah sesuai, Staf Tata Usaha mencatat data siswa ke dalam buku pendaftaran sementara, menentukan Nomor Induk Siswa (NIS), kemudian menyalin seluruh data ke dalam buku induk dan menempelkan foto siswa. Selanjutnya data diserahkan kepada Kepala Sekolah untuk dilakukan verifikasi. Apabila masih terdapat kesalahan, berkas dikembalikan kepada Staf Tata Usaha untuk diperbaiki. Setelah data dinyatakan benar, Kepala Sekolah memberikan tanda tangan dan stempel sebagai bentuk pengesahan, kemudian berkas diarsipkan sebagai dokumen sekolah.

Proses berikutnya adalah pencatatan absensi siswa yang dilakukan oleh Guru pada setiap kegiatan belajar mengajar. Guru memulai pembelajaran dengan memanggil nama siswa satu per satu, kemudian mencatat status kehadiran siswa ke dalam buku absensi. Setelah seluruh data kehadiran selesai dicatat, buku absensi disimpan sebagai arsip yang akan digunakan kembali apabila diperlukan pada proses rekapitulasi kehadiran siswa.

Selain mendukung proses administrasi dan absensi, kegiatan akademik juga mencakup penyampaian informasi kepada orang tua siswa. Ketika orang tua membutuhkan informasi mengenai perkembangan akademik anak, mereka terlebih dahulu menghubungi guru. Guru kemudian mencari data akademik yang diperlukan, melakukan verifikasi terhadap informasi tersebut, kemudian menyampaikan hasilnya kepada orang tua. Apabila informasi yang diberikan

memerlukan penyesuaian atau terdapat kesalahan data, proses verifikasi dilakukan kembali hingga informasi dinyatakan sesuai.

Berdasarkan hasil observasi terhadap keseluruhan proses tersebut, dapat diketahui bahwa seluruh aktivitas administrasi akademik masih bergantung pada dokumen fisik dan komunikasi langsung antar pengguna. Kondisi tersebut menyebabkan proses pengelolaan data membutuhkan waktu yang relatif lebih lama, terutama ketika dilakukan pencarian arsip, penyusunan laporan, maupun penyampaian informasi akademik kepada orang tua. Oleh karena itu, diperlukan suatu sistem informasi akademik berbasis web yang mampu mengintegrasikan seluruh proses tersebut sehingga pengelolaan data menjadi lebih efektif dan efisien.

5.1.1.3 Permasalahan sistem yang berjalan

Berdasarkan hasil observasi terhadap keseluruhan proses sistem manual yang sedang berjalan di RA Al-Islam Gunungcupu, dapat disimpulkan bahwa pengelolaan data akademik masih menghadapi beberapa kendala yang memengaruhi efektivitas pelayanan administrasi sekolah. Seluruh data masih dikelola menggunakan dokumen fisik, baik dalam proses pendataan siswa, pencatatan absensi, maupun penyampaian informasi akademik kepada orang tua. Kondisi ini menyebabkan proses pengelolaan data menjadi kurang efisien karena setiap aktivitas administrasi memerlukan pencatatan dan pencarian data secara manual.

Selain itu, penggunaan buku induk dan buku absensi sebagai media penyimpanan utama meningkatkan risiko kehilangan, kerusakan, maupun penumpukan arsip seiring bertambahnya jumlah siswa setiap tahun. Proses penyampaian informasi kepada orang tua juga belum berjalan secara optimal karena masih bergantung pada komunikasi langsung dengan guru. Akibatnya, penyampaian informasi tidak dapat dilakukan secara cepat apabila guru sedang tidak berada di sekolah atau membutuhkan waktu untuk mencari data yang diperlukan.

Berdasarkan permasalahan tersebut, diperlukan sebuah sistem informasi akademik berbasis web yang mampu mengintegrasikan proses pendataan siswa, pengelolaan

absensi, serta penyampaian informasi akademik kepada orang tua dalam satu sistem. Hasil analisis observasi ini selanjutnya menjadi dasar dalam penyusunan kebutuhan sistem yang dituangkan ke dalam *Product Backlog* pada tahapan berikutnya menggunakan metode *Scrum*.

5.1.2 Hasil wawancara

Selain melakukan observasi, penelitian ini juga menggunakan metode wawancara untuk memperoleh informasi secara langsung mengenai kebutuhan sistem dari pihak yang terlibat dalam kegiatan akademik di RA Al-Islam Gunungcupu. Wawancara dilakukan secara semi terstruktur kepada Kepala RA dan guru sebagai pengguna utama sistem. Tujuan wawancara adalah untuk memperkuat hasil observasi, mengidentifikasi permasalahan yang dihadapi selama proses administrasi akademik, serta mengetahui kebutuhan pengguna terhadap sistem informasi yang akan dikembangkan.

Hasil wawancara menunjukkan bahwa proses administrasi akademik yang masih dilakukan secara manual menimbulkan beberapa kendala dalam pengelolaan data, pencarian informasi, dan penyampaian informasi kepada orang tua. Informasi yang diperoleh dari wawancara kemudian dianalisis sebagai dasar dalam menentukan kebutuhan sistem yang akan disusun pada tahap *Product Backlog*.

5.1.2.1 Hasil wawancara dengan Kepala RA

Wawancara dengan Kepala RA Al-Islam Gunungcupu dilakukan untuk memperoleh gambaran mengenai kondisi pengelolaan administrasi akademik dari sudut pandang pengelola sekolah. Berdasarkan hasil wawancara, diketahui bahwa seluruh data akademik masih dikelola secara manual menggunakan buku induk siswa dan buku absensi. Selama ini belum tersedia sistem informasi yang mampu mengintegrasikan seluruh data akademik dalam satu media sehingga setiap proses administrasi masih bergantung pada dokumen fisik.

Kepala RA menjelaskan bahwa penggunaan dokumen fisik menyebabkan proses administrasi menjadi kurang efisien, terutama ketika diperlukan pencarian data siswa maupun penyusunan laporan akademik. Semakin banyak jumlah siswa yang

dikelola, semakin besar pula arsip yang harus disimpan sehingga proses pencarian informasi membutuhkan waktu lebih lama. Selain itu, penyimpanan data dalam bentuk buku memiliki risiko kerusakan maupun kehilangan apabila arsip tidak tersimpan dengan baik.

Penyampaian informasi akademik kepada orang tua juga masih dilakukan secara langsung melalui guru. Kondisi tersebut menyebabkan proses penyampaian informasi belum dapat dilakukan secara cepat karena guru harus terlebih dahulu mencari data yang diperlukan sebelum informasi dapat diberikan kepada orang tua.

Berdasarkan kondisi tersebut, Kepala RA mengharapkan adanya sistem informasi akademik berbasis web yang mampu mengelola seluruh data akademik secara terpusat, mempercepat proses administrasi, mempermudah pencarian data, serta meningkatkan efektivitas penyampaian informasi akademik kepada orang tua.

5.1.2.2 Hasil wawancara dengan Guru

Wawancara juga dilakukan kepada guru sebagai pengguna yang akan berinteraksi langsung dengan sistem dalam kegiatan operasional sehari-hari. Berdasarkan hasil wawancara, guru menyampaikan bahwa proses pencatatan absensi siswa dan pengelolaan data akademik masih dilakukan secara manual sehingga membutuhkan waktu yang relatif lama. Setiap kegiatan administrasi harus dicatat pada dokumen fisik sehingga pekerjaan menjadi lebih banyak dan berpotensi menimbulkan kesalahan pencatatan.

Guru juga menjelaskan bahwa ketika orang tua membutuhkan informasi mengenai perkembangan akademik siswa, guru harus mencari terlebih dahulu data yang tersimpan pada buku induk maupun arsip administrasi lainnya. Proses tersebut memerlukan waktu, terutama ketika data yang dicari berasal dari periode pembelajaran sebelumnya atau jumlah arsip yang dikelola semakin banyak.

Selain itu, guru menyampaikan bahwa proses pengelolaan nilai masih memanfaatkan aplikasi Rapor Digital Madrasah (RDM). Namun, data pada aplikasi tersebut belum terhubung dengan administrasi akademik sekolah sehingga sebagian data masih harus dikelola secara terpisah. Kondisi tersebut menyebabkan pekerjaan

administrasi menjadi kurang efisien karena data belum terintegrasi dalam satu sistem.

Oleh karena itu, guru mengharapkan adanya sistem yang mampu membantu pengelolaan data siswa, pencatatan absensi, serta penyampaian informasi akademik kepada orang tua secara lebih cepat. Guru juga mengharapkan sistem yang dikembangkan dapat memanfaatkan data nilai dari RDM sehingga proses pengelolaan data akademik menjadi lebih efisien dan tidak memerlukan penginputan data secara berulang.

5.1.2.3 Kebutuhan pengguna terhadap sistem

Berdasarkan hasil observasi dan wawancara yang telah dilakukan, dapat disimpulkan bahwa pengguna memiliki sejumlah kebutuhan yang harus dipenuhi oleh sistem informasi akademik yang akan dikembangkan. Kebutuhan tersebut merupakan hasil analisis terhadap permasalahan yang ditemukan selama proses administrasi akademik dan menjadi dasar dalam penyusunan *Product Backlog* menggunakan metode *Scrum*.

Secara umum, pihak sekolah membutuhkan sistem yang mampu mengintegrasikan seluruh data akademik ke dalam satu basis data sehingga proses pengelolaan informasi dapat dilakukan secara lebih efektif dan efisien. Sistem diharapkan mampu membantu admin dalam mengelola data master sekolah, membantu guru dalam mengelola data siswa serta pencatatan absensi, dan memberikan kemudahan kepada orang tua untuk memperoleh informasi akademik tanpa harus datang langsung ke sekolah.

Selain kebutuhan tersebut, hasil wawancara juga menunjukkan bahwa sistem harus memiliki antarmuka yang sederhana dan mudah dipahami oleh seluruh pengguna. Hal ini disebabkan kemampuan penggunaan teknologi di lingkungan sekolah masih beragam sehingga sistem harus dirancang agar dapat digunakan tanpa memerlukan proses pembelajaran yang rumit.

Berdasarkan hasil analisis tersebut, kebutuhan pengguna kemudian diterjemahkan menjadi daftar kebutuhan sistem yang selanjutnya disusun dalam bentuk *Product*

Backlog. Daftar kebutuhan tersebut menjadi acuan dalam penyusunan *Sprint Planning* dan proses pengembangan sistem menggunakan metode *Scrum* pada tahapan berikutnya.

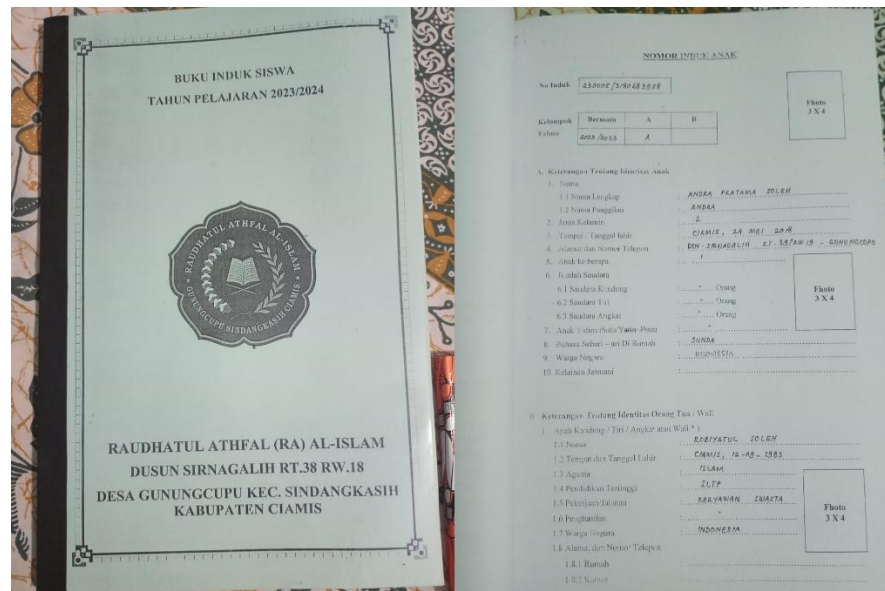
5.1.3 Hasil dokumentasi

Selain observasi dan wawancara, penelitian ini juga didukung oleh kegiatan dokumentasi. Dokumentasi dilakukan untuk memperoleh bukti fisik mengenai proses pengelolaan data akademik yang sedang berjalan di RA Al-Islam Gunungcupu. Dokumen yang dikumpulkan digunakan sebagai sumber informasi dalam memahami proses administrasi sekolah sekaligus sebagai acuan dalam merancang sistem informasi akademik berbasis web yang sesuai dengan kebutuhan pengguna.

Hasil dokumentasi menunjukkan bahwa sebagian besar data akademik masih dikelola menggunakan dokumen fisik. Dokumen tersebut terdiri atas buku induk siswa dan buku absensi yang digunakan dalam kegiatan administrasi sehari-hari. Kedua dokumen tersebut menjadi dasar dalam menganalisis kebutuhan data yang akan diimplementasikan ke dalam sistem informasi akademik.

5.1.3.1 Buku induk siswa

Buku induk siswa merupakan dokumen utama yang digunakan oleh RA Al-Islam Gunungcupu untuk menyimpan data identitas setiap peserta didik. Berdasarkan hasil dokumentasi, buku induk memuat informasi seperti nomor induk siswa, identitas pribadi, data orang tua, alamat, serta informasi administrasi lainnya yang berkaitan dengan peserta didik.



Gambar 5.2 Buku induk siswa

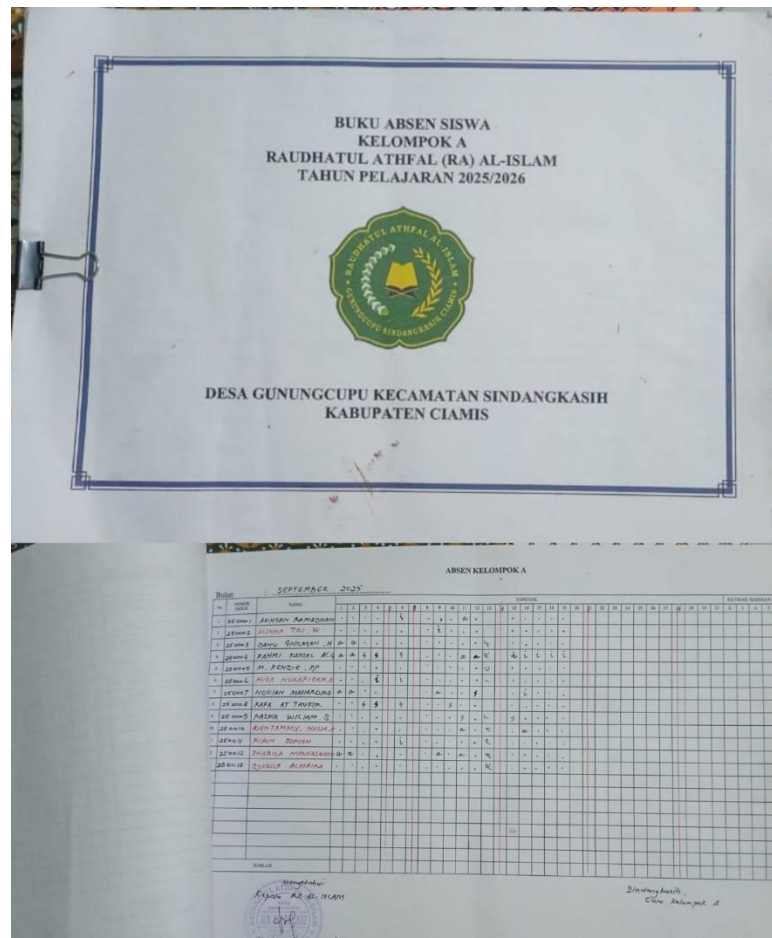
Berdasarkan dokumentasi tersebut, diketahui bahwa seluruh data siswa masih dicatat secara manual menggunakan media tulis. Meskipun cara tersebut telah mampu memenuhi kebutuhan administrasi sekolah, proses pencarian, pembaruan, maupun pengelolaan data masih memerlukan waktu yang relatif lama karena petugas harus membuka arsip secara langsung. Selain itu, penyimpanan data dalam bentuk buku fisik memiliki risiko kerusakan maupun kehilangan apabila tidak dilakukan pengelolaan arsip dengan baik.

Hasil dokumentasi buku induk menjadi salah satu dasar dalam menentukan kebutuhan fitur pengelolaan data siswa pada sistem informasi akademik yang akan dikembangkan. Seluruh informasi yang sebelumnya dicatat secara manual dirancang untuk disimpan ke dalam basis data sehingga proses pengelolaan data dapat dilakukan secara lebih efektif, terstruktur, dan mudah diakses oleh pengguna sesuai dengan hak akses masing-masing.

5.1.3.2 Buku absensi siswa

Buku absensi merupakan dokumen yang digunakan guru untuk mencatat kehadiran siswa pada setiap kegiatan pembelajaran. Berdasarkan hasil dokumentasi, pencatatan absensi dilakukan secara manual dengan memberikan keterangan kehadiran setiap siswa pada tanggal pembelajaran yang bersangkutan. Selanjutnya

buku absensi disimpan sebagai arsip administrasi sekolah dan digunakan sebagai sumber informasi apabila diperlukan laporan kehadiran siswa.



Gambar 5.3 Buku absensi siswa

Hasil dokumentasi menunjukkan bahwa proses pencatatan absensi masih bergantung pada dokumen fisik sehingga guru harus melakukan pencatatan secara langsung setiap hari. Ketika diperlukan rekapitulasi kehadiran maupun informasi mengenai absensi siswa, guru harus mencari kembali data pada buku absensi. Kondisi tersebut menyebabkan proses pengelolaan informasi menjadi kurang efisien, terutama ketika jumlah data yang harus diperiksa semakin banyak.

5.1.3.3 Buku penghubung guru dan orang tua

Berdasarkan hasil dokumentasi, RA Al-Islam Gunungcupu menggunakan buku penghubung sebagai media komunikasi antara guru dan orang tua siswa. Buku

tersebut dimanfaatkan untuk menyampaikan pengumuman dari pihak sekolah, catatan guru, serta pesan atau tanggapan dari orang tua mengenai perkembangan maupun kegiatan siswa. Seluruh informasi masih dicatat secara tertulis sehingga komunikasi dilakukan melalui media fisik yang dibawa oleh siswa.



Gambar 5.4 Buku penghubung guru dan orang tua

Penggunaan buku penghubung menunjukkan bahwa proses penyampaian informasi masih dilakukan secara manual. Meskipun dapat membantu komunikasi antara guru dan orang tua, cara tersebut memiliki beberapa keterbatasan, seperti informasi yang tidak dapat diterima secara langsung, bergantung pada siswa untuk membawa buku, serta berisiko mengalami kerusakan atau kehilangan. Kondisi ini menyebabkan penyampaian informasi menjadi kurang efektif apabila diperlukan dalam waktu yang cepat.

Hasil dokumentasi tersebut memperkuat temuan pada tahap observasi dan wawancara bahwa proses komunikasi akademik di RA Al-Islam Gunungcupu belum didukung oleh sistem informasi yang terintegrasi. Oleh karena itu, sistem informasi akademik yang diusulkan menyediakan akses bagi orang tua untuk memperoleh informasi akademik siswa secara langsung melalui aplikasi berbasis web sehingga penyampaian informasi menjadi lebih cepat, terdokumentasi, dan mudah diakses.

Berdasarkan hasil dokumentasi yang telah dilakukan, diketahui bahwa pengelolaan data siswa, pencatatan kehadiran, dan penyampaian informasi kepada orang tua masih dilakukan menggunakan dokumen fisik berupa buku induk, buku absensi, dan buku penghubung. Kondisi tersebut menunjukkan bahwa proses administrasi akademik belum terintegrasi dalam satu sistem sehingga pengelolaan maupun penyampaian informasi masih kurang efisien. Oleh karena itu, sistem informasi akademik yang dikembangkan dirancang untuk mengelola data siswa, absensi, serta penyampaian informasi akademik kepada orang tua secara terintegrasi sehingga proses pencatatan, penyimpanan, pencarian, dan penyampaian informasi dapat dilakukan dengan lebih cepat, terstruktur, dan mudah diakses sesuai dengan hak akses masing-masing pengguna.

5.2 Menyusun Product Backlog

Tahap penyusunan Product Backlog merupakan tahapan awal dalam metode *Scrum* yang bertujuan untuk mengidentifikasi dan menyusun seluruh kebutuhan sistem berdasarkan tingkat prioritas. *Product Backlog* berisi daftar kebutuhan atau fitur yang harus dikembangkan agar sistem mampu memenuhi kebutuhan pengguna. Penyusunan *Product Backlog* pada penelitian ini dilakukan berdasarkan hasil observasi, wawancara, dan dokumentasi yang telah dilaksanakan pada tahap pengumpulan data. Informasi yang diperoleh dari ketiga teknik pengumpulan data tersebut dianalisis untuk mengidentifikasi permasalahan pada sistem yang sedang berjalan serta menentukan solusi yang dapat diterapkan melalui sistem informasi akademik berbasis web. Seluruh kebutuhan yang telah diidentifikasi kemudian disusun menjadi daftar fitur yang akan dikembangkan secara bertahap menggunakan metode *Scrum*.

5.2.1 Identifikasi kebutuhan sistem

Tahap identifikasi kebutuhan sistem dilakukan untuk menentukan kebutuhan fungsional yang harus tersedia pada sistem informasi akademik berbasis web. Identifikasi kebutuhan dilakukan dengan menganalisis hasil observasi terhadap proses administrasi sekolah, hasil wawancara dengan Kepala RA dan guru sebagai calon pengguna sistem, serta hasil dokumentasi berupa buku induk siswa dan buku

absensi. Analisis tersebut bertujuan agar sistem yang dikembangkan mampu memberikan solusi terhadap permasalahan yang ditemukan selama proses penelitian.

Berdasarkan hasil observasi, diketahui bahwa pengelolaan data akademik di RA Al-Islam Gunungcupu masih dilakukan secara manual menggunakan dokumen fisik. Proses pendataan siswa, pencatatan absensi, serta penyampaian informasi akademik kepada orang tua belum terintegrasi dalam satu sistem sehingga proses administrasi memerlukan waktu yang relatif lama. Selain itu, pencarian data masih dilakukan dengan membuka arsip secara langsung sehingga kurang efektif ketika informasi dibutuhkan dalam waktu yang singkat.

Hasil wawancara memperkuat temuan tersebut. Kepala RA mengharapkan adanya sistem yang mampu mengelola seluruh data akademik secara terpusat sehingga proses administrasi menjadi lebih mudah dan penyimpanan data menjadi lebih aman. Sementara itu, guru mengharapkan sistem yang dapat membantu proses pengelolaan data siswa, pencatatan absensi, serta penyampaian informasi akademik kepada orang tua. Guru juga mengharapkan agar sistem dapat memanfaatkan data nilai dari Rapor Digital Madrasah (RDM) sehingga proses pengelolaan data tidak dilakukan secara berulang.

Selain itu, hasil dokumentasi menunjukkan bahwa buku induk siswa dan buku absensi masih menjadi media utama dalam penyimpanan data akademik. Kondisi tersebut mengakibatkan proses pencarian, pembaruan, dan penyusunan laporan memerlukan waktu yang cukup lama karena seluruh data masih tersimpan dalam bentuk arsip fisik. Oleh karena itu, sistem yang dikembangkan diharapkan mampu menggantikan proses pencatatan manual menjadi pengelolaan data secara digital yang lebih terstruktur dan mudah diakses sesuai hak akses masing-masing pengguna.

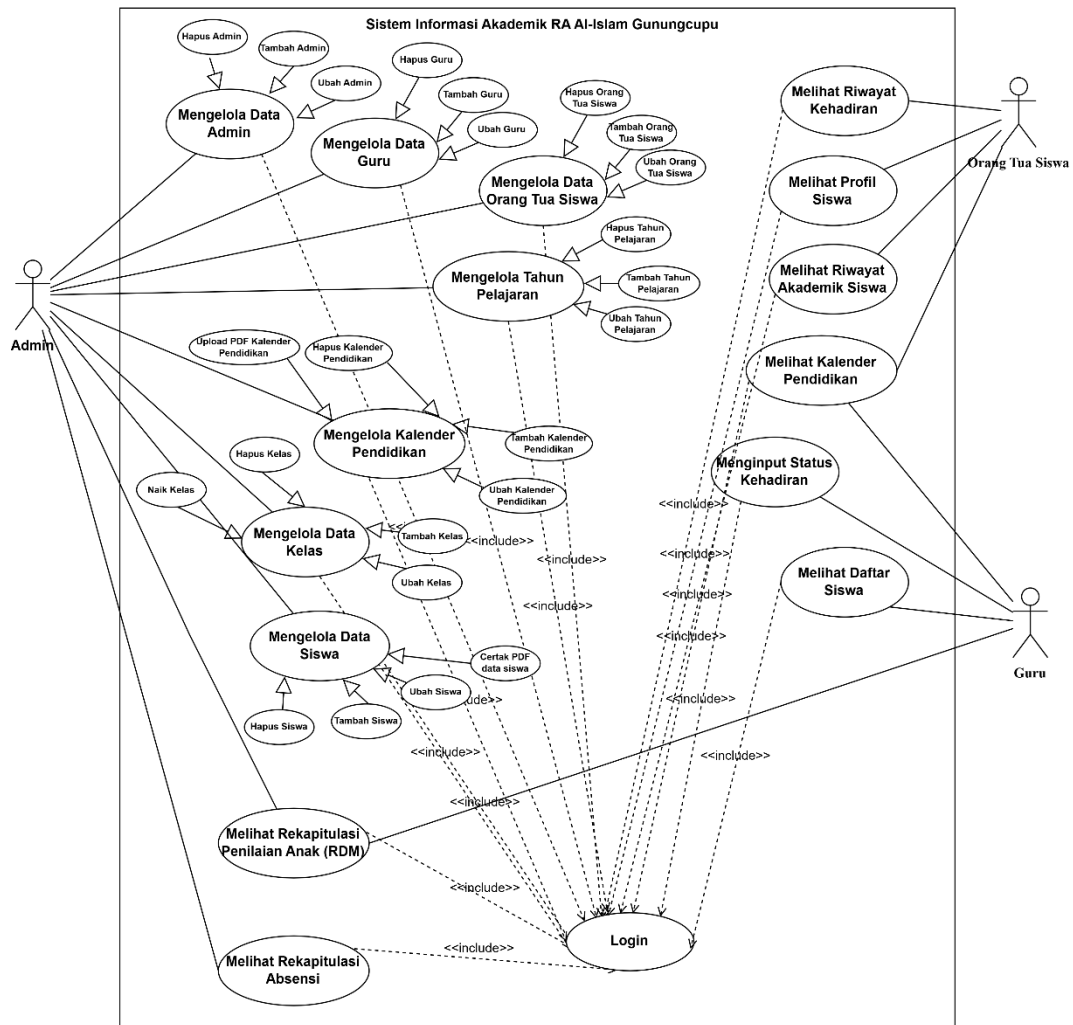
Berdasarkan hasil analisis tersebut, diperoleh sejumlah kebutuhan fungsional yang harus dipenuhi oleh sistem informasi akademik berbasis web. Kebutuhan tersebut meliputi pengelolaan data pengguna, pengelolaan tahun pelajaran, kalender

pendidikan, data kelas, data siswa, absensi siswa, integrasi data penilaian dari Rapor Digital Madrasah (RDM), penyajian informasi akademik kepada orang tua, serta pengelolaan laporan akademik. Seluruh kebutuhan tersebut selanjutnya divisualisasikan melalui *Use Case Diagram* dan diterjemahkan menjadi *Product Backlog* sebagai dasar pelaksanaan *Sprint* pada metode *Scrum*.

5.2.2 Use case diagram sistem usulan

Berdasarkan hasil identifikasi kebutuhan sistem yang telah diperoleh dari proses observasi, wawancara, dan dokumentasi, selanjutnya disusun rancangan sistem usulan menggunakan *Use Case Diagram*. Diagram ini digunakan untuk menggambarkan hubungan antara aktor dengan fungsi-fungsi utama yang tersedia pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui *Use Case Diagram*, batasan hak akses setiap pengguna dapat dijelaskan secara lebih jelas sehingga proses pengembangan sistem pada metode *Scrum* memiliki acuan yang terstruktur.

Pada sistem yang diusulkan terdapat tiga aktor utama, yaitu Admin, Guru, dan Orang Tua Siswa. Setiap aktor memiliki hak akses yang berbeda sesuai dengan tugas dan tanggung jawabnya dalam kegiatan akademik di RA Al-Islam Gunungcupu. Admin memiliki hak akses penuh terhadap pengelolaan data master sistem. Guru berperan sebagai pengguna yang mengelola kegiatan akademik sehari-hari, sedangkan Orang Tua Siswa berperan sebagai penerima informasi akademik yang disediakan oleh sistem.



Gambar 5.5 Use Case Diagram

5.2.2.1 Deskripsi aktor

Untuk memberikan penjelasan mengenai peran masing-masing aktor pada sistem usulan, deskripsi aktor disajikan pada Tabel 5.1.

Tabel 5.1 Deskripsi Aktor

No	Aktor	Deskripsi
1	Admin	Admin merupakan pengguna yang memiliki hak akses penuh terhadap sistem. Admin bertugas mengelola data admin, data guru, data orang tua siswa, tahun pelajaran, kalender pendidikan, data kelas, data siswa, serta melihat rekapitulasi absensi dan rekapitulasi penilaian siswa dari RDM.
2	Guru	Guru menggunakan sistem untuk melihat daftar siswa sesuai kelas, menginput status kehadiran siswa, serta melihat kalender pendidikan dan rekapitulasi penilaian anak.
3	Orang Tua Siswa	Orang tua siswa menggunakan sistem untuk memperoleh informasi akademik anaknya, meliputi profil siswa, riwayat kehadiran, riwayat akademik, serta kalender pendidikan tanpa harus datang langsung ke sekolah.

5.2.2.2 Deskripsi per-use case

Berdasarkan *Use Case Diagram* yang telah dirancang, setiap fungsi utama pada sistem memiliki tujuan dan proses yang berbeda sesuai kebutuhan pengguna. Deskripsi masing-masing use case disajikan pada Tabel 5.2.

Tabel 5.2 Deskripsi Per-Use Case

No	Use Case	Deskripsi
1	Login	Proses autentikasi pengguna untuk memverifikasi username dan password sehingga pengguna dapat mengakses sistem sesuai hak aksesnya.
2	Mengelola Data Admin	Mengelola data admin yang meliputi penambahan, perubahan, dan penghapusan data admin.
3	Mengelola Data Guru	Mengelola data guru yang meliputi penambahan, perubahan, dan penghapusan data guru.

No	Use Case	Deskripsi
4	Mengelola Data Orang Tua Siswa	Mengelola data orang tua siswa yang meliputi penambahan, perubahan, dan penghapusan data orang tua siswa.
5	Mengelola Tahun Pelajaran	Mengelola data tahun pelajaran yang meliputi penambahan, perubahan, dan penghapusan tahun pelajaran yang digunakan dalam sistem.
6	Mengelola Kalender Pendidikan	Mengelola kalender pendidikan yang meliputi penambahan, perubahan, penghapusan jadwal kegiatan akademik sekolah dan upload PDF dokumen Kalender Pendidikan.
7	Mengelola Data Kelas	Mengelola data kelas yang meliputi penambahan, perubahan, penghapusan data kelas dan kenaikan kelas.
8	Mengelola Data Siswa	Mengelola data siswa yang meliputi penambahan, perubahan, penghapusan, serta pencetakan data siswa dalam format PDF.
9	Melihat Rekapitulasi Penilaian Anak (RDM)	Menampilkan rekapitulasi hasil penilaian siswa yang berasal dari sistem Rapor Digital Madrasah (RDM).
10	Melihat Rekapitulasi Absensi	Menampilkan laporan rekapitulasi kehadiran siswa berdasarkan data absensi yang telah tersimpan pada sistem.
11	Melihat Daftar Siswa	Menampilkan daftar siswa sesuai kelas yang diampu oleh guru.
12	Menginput Status Kehadiran	Digunakan guru untuk mencatat status kehadiran siswa pada setiap kegiatan pembelajaran.

No	Use Case	Deskripsi
13	Melihat Kalender Pendidikan	Menampilkan kalender pendidikan yang berisi jadwal kegiatan akademik sekolah dan dapat diakses oleh guru maupun orang tua siswa.
14	Melihat Profil Siswa	Menampilkan informasi identitas lengkap siswa kepada orang tua sesuai akun yang digunakan.
15	Melihat Riwayat Kehadiran	Menampilkan riwayat kehadiran siswa sehingga orang tua dapat memantau absensi anaknya.
16	Melihat Riwayat Akademik Siswa	Menampilkan informasi perkembangan akademik siswa berdasarkan data yang tersedia pada sistem.

5.2.3 Product backlog

Berdasarkan hasil identifikasi kebutuhan sistem melalui observasi, wawancara, dan dokumentasi, disusunlah *Product Backlog* untuk pengembangan Sistem Informasi Akademik RA Al-Islam Gunungcupu menggunakan metode *Scrum*. Daftar kebutuhan fungsional tersebut memuat fitur berdasarkan kebutuhan pengguna (*user story*), aktor, tingkat prioritas, serta estimasi waktu pengerjaan guna memberikan gambaran ruang lingkup pengembangan sekaligus menjadi dasar pelaksanaan *Sprint Planning*. Rincian *Product Backlog* tersebut disajikan secara lengkap pada Tabel 5.3.

Tabel 5.3 *Product Backlog*

No	Nama Menu	<i>User Story</i> (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
1	Login / Autentikasi	Sebagai pengguna, saya ingin login menggunakan email dan password agar dapat mengakses	Semua Pengguna	Sedang	1 Hari

No	Nama Menu	User Story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
		sistem sesuai dengan hak akses yang dimiliki.			
2	Dashboard	Sebagai pengguna, saya ingin melihat informasi ringkas pada dashboard setelah berhasil login agar dapat mengetahui kondisi sistem secara cepat.	Semua Pengguna	Sedang	1 Hari
3	Pengguna	Sebagai Admin, saya ingin mengelola data Admin (tambah, ubah, dan hapus) agar akun administrator dapat dikelola dengan baik.	Admin	Tinggi	2 Hari
4	Pengguna	Sebagai Admin, saya ingin mengelola data Guru agar guru dapat menggunakan sistem sesuai dengan hak aksesnya.	Admin	Tinggi	2 Hari
5	Pengguna	Sebagai Admin, saya ingin mengelola data	Admin	Tinggi	2 Hari

No	Nama Menu	User Story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
		Orang Tua agar setiap wali siswa memiliki akun untuk mengakses sistem.			
6	Lupa Sandi	Sebagai Orang Tua, saya ingin menggunakan fitur lupa sandi agar saya bisa mengakses kembali akun saat saya melupakan kata sandi.	Orang Tua Siswa	Tinggi	1 Hari
7	Tahun Pelajaran	Sebagai Admin, saya ingin mengelola data Tahun Pelajaran (tambah, ubah, dan hapus) agar sistem selalu menggunakan periode akademik yang aktif.	Admin	Tinggi	2 Hari
8	Kalender Pendidikan	Sebagai Admin, saya ingin mengelola Kalender Pendidikan (tambah, ubah, dan hapus) agar seluruh kegiatan akademik sekolah dapat	Admin	Sedang	2 Hari

No	Nama Menu	User Story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
		dijadwalkan dengan baik.			
9	Kalender Pendidikan	Sebagai Admin, saya ingin upload PDF dokumen Kalender Pendidikan.	Admin	Sedang	1 Hari
10	Kalender Pendidikan	Sebagai Guru dan Orang Tua, saya ingin melihat Kalender Pendidikan agar mengetahui jadwal kegiatan dan hari libur sekolah.	Guru, Orang Tua	Sedang	1 Hari
11	Kelas	Sebagai Admin, saya ingin mengelola data Kelas (tambah, ubah, dan hapus) agar pembagian kelas siswa dapat diatur dengan baik.	Admin	Tinggi	2 Hari
12	Kelas	Sebagai Admin, saya ingin menaikkan kelas siswa ke tingkat kelas selanjutnya agar pendataan tahun ajaran baru terorganisir.	Admin	Sedang	1 Hari

No	Nama Menu	<i>User Story</i> (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
13	Siswa	Sebagai Admin, saya ingin mengelola data siswa (tambah, ubah, dan hapus) agar seluruh data siswa tersimpan secara digital.	Admin	Sangat Tinggi	4 Hari
14	Siswa	Sebagai Admin, saya ingin mencetak Biodata Siswa dalam format PDF sebagai arsip administrasi sekolah.	Admin	Sedang	2 Hari
15	Daftar Siswa	Sebagai Guru, saya ingin melihat daftar siswa beserta profilnya agar dapat mengenal dan mengelola data siswa pada kelas yang diampu.	Guru	Tinggi	2 Hari
16	Profil Siswa	Sebagai Orang Tua, saya ingin melihat profil anak saya agar dapat memperoleh informasi identitas siswa secara mandiri.	Orang Tua	Sedang	1 Hari

No	Nama Menu	<i>User Story</i> (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
17	Absensi	Sebagai Guru, saya ingin menginput status kehadiran siswa (Hadir, Izin, Sakit, atau Alpa) agar data absensi tercatat dengan baik setiap hari.	Guru	Sangat Tinggi	4 Hari
18	Absensi	Sebagai Admin, saya ingin melihat rekapitulasi absensi seluruh siswa agar dapat memantau tingkat kehadiran siswa.	Admin	Tinggi	2 Hari
19	Riwayat Kehadiran	Sebagai Orang Tua, saya ingin melihat riwayat kehadiran anak saya agar dapat memantau kehadiran siswa di sekolah.	Orang Tua	Tinggi	2 Hari
20	Penilaian Anak	Sebagai Admin, saya ingin melihat rekapitulasi menu Penilaian Anak dari database RDM agar	Admin	Tinggi	3 Hari

No	Nama Menu	User Story (Kebutuhan Fitur)	Aktor	Prioritas	Estimasi Waktu (Hari)
		data nilai tersinkronisasi.			
21	Penilaian Anak	Sebagai Guru, saya ingin melihat rekapitulasi Penilaian Anak dari database RDM agar saya dapat memantau capaian nilai siswa di kelas yang saya ampu.	Guru	Tinggi	2 Hari
22	Penilaian Anak	Sebagai Orang Tua Siswa, saya ingin melihat capaian akademik anak saya pada menu akademik agar bisa memantaunya dari rumah.	Orang Tua	Tinggi	2 Hari

Berdasarkan *Product Backlog* yang telah disusun, diperoleh 22 kebutuhan fungsional yang akan dikembangkan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Setiap *backlog* disusun berdasarkan hasil analisis kebutuhan pengguna, kemudian diberikan tingkat prioritas dan estimasi waktu pengerjaan sebagai acuan dalam menentukan urutan pengembangan sistem. *Backlog* dengan prioritas Sangat Tinggi diposisikan sebagai fitur inti yang harus dikembangkan terlebih dahulu, sedangkan *backlog* dengan prioritas Tinggi dan Sedang dikembangkan sesuai urutan kebutuhan pada setiap *Sprint*. Selanjutnya, seluruh

Product Backlog tersebut disusun ke dalam *Sprint Backlog* pada tahap *Sprint Planning*.

5.3 Sprint Planning

Sprint Planning merupakan tahapan untuk menentukan pekerjaan yang akan dikerjakan pada setiap *Sprint* berdasarkan *Product Backlog* yang telah disusun pada tahap sebelumnya. Pada tahap ini, seluruh kebutuhan sistem dianalisis kembali untuk menentukan fitur yang menjadi prioritas pengembangan, target yang ingin dicapai pada setiap *Sprint*, serta estimasi waktu penyelesaiannya. Pembagian *Sprint* dilakukan agar proses pengembangan sistem berlangsung secara bertahap, terarah, dan lebih mudah dievaluasi pada akhir setiap *Sprint*.

Dalam penelitian ini, seluruh *Product Backlog* dibagi ke dalam tiga *Sprint* dengan durasi masing-masing selama 14 hari. Pembagian tersebut disusun berdasarkan tingkat keterkaitan antar fitur serta urutan implementasi sistem. Fitur-fitur dasar yang menjadi fondasi sistem dikembangkan pada *Sprint* pertama, kemudian dilanjutkan dengan fitur operasional utama pada *Sprint* kedua, dan diakhiri dengan fitur pelaporan serta penyajian informasi kepada pengguna pada *Sprint* ketiga. Dengan pembagian tersebut, setiap *Sprint* menghasilkan peningkatan fungsionalitas sistem yang dapat digunakan sebagai dasar pengembangan pada *Sprint* berikutnya.

Rincian pembagian *Sprint Planning* yang digunakan dalam penelitian ini disajikan pada Tabel 5.4.

Tabel 5.4 Sprint Planning

Sprint	Target Sprint	Product Backlog yang Dipilih	Estimasi Waktu
Sprint 1	Mengembangkan fitur Login, Dashboard, manajemen Pengguna (Admin, Guru, Orang Tua, dan Lupa Sandi), serta fitur Tahun Pelajaran, Kelas, dan melihat Kalender Pendidikan.	Login, Dashboard, Pengguna (Admin, Guru, Ortu, Lupa Sandi), Tahun Pelajaran, Kelas, Kelas	14 Hari

		(Kenaikan Kelas), Kalender (Melihat).	
Sprint 2	Mengembangkan fitur pengelolaan Siswa (tambah, ubah, hapus), pencatatan kehadiran (Absensi) harian oleh guru, membuat Kalender Pendidikan, dan cetak Biodata PDF.	Siswa (Kelola Siswa), Absensi (Input Guru), Kalender (Mengelola), Kalender (Upload PDF Dokumen Kalender Pendidikan, Siswa (Cetak PDF).	14 Hari
Sprint 3	Mengembangkan fitur pelaporan seperti melihat Daftar Siswa, Profil Siswa, rekapitulasi kehadiran, serta menu melihat Capaian Akademik (Rapor) untuk semua aktor.	Daftar Siswa, Profil Siswa, Absensi (Rekap Admin), Riwayat Kehadiran, Penilaian Anak (Admin, Guru, Orang Tua).	14 Hari

Setelah *Sprint Planning* selesai disusun, setiap *Sprint* menjadi acuan dalam proses pengembangan sistem. Daftar pekerjaan yang telah ditentukan pada masing-masing *Sprint* kemudian diimplementasikan secara bertahap pada tahap *Sprint*. Dengan pendekatan tersebut, proses pengembangan tidak dilakukan secara sekaligus, melainkan melalui beberapa iterasi yang menghasilkan peningkatan fungsionalitas sistem secara bertahap hingga seluruh kebutuhan pengguna yang terdapat pada *Product Backlog* dapat direalisasikan.

5.4 Pelaksanaan Sprint

Sprint merupakan tahap pelaksanaan pengembangan sistem berdasarkan *Sprint Planning* yang telah disusun pada tahap sebelumnya. Pada tahap ini, setiap *Product Backlog* yang telah dipilih pada masing-masing *Sprint* diimplementasikan menjadi fitur yang dapat digunakan oleh pengguna. Proses pengembangan dilakukan secara bertahap sesuai dengan target *Sprint* sehingga setiap fitur dapat diselesaikan, diuji, dan dievaluasi sebelum dilanjutkan ke *Sprint* berikutnya.

Pada penelitian ini, pengembangan sistem dibagi menjadi tiga *Sprint*. Setiap *Sprint* menghasilkan peningkatan fungsionalitas sistem sesuai dengan kebutuhan pengguna yang telah diidentifikasi pada tahap *Product Backlog*. Pembahasan pada setiap *Sprint* meliputi proses perancangan menggunakan *Unified Modeling Language* (UML), implementasi program, serta hasil tampilan antarmuka sistem yang telah berhasil dibangun.

5.4.1 *Sprint* 1

Sprint 1 difokuskan pada pembangunan fitur-fitur dasar yang menjadi fondasi Sistem Informasi Akademik. Fitur yang dikembangkan pada *Sprint* ini meliputi Login, Dashboard, Pengguna, Tahun Pelajaran, Kelas, serta Kalender Pendidikan (melihat). Fitur-fitur tersebut diprioritaskan karena merupakan komponen utama yang harus tersedia sebelum pengembangan fitur operasional lainnya dilakukan.

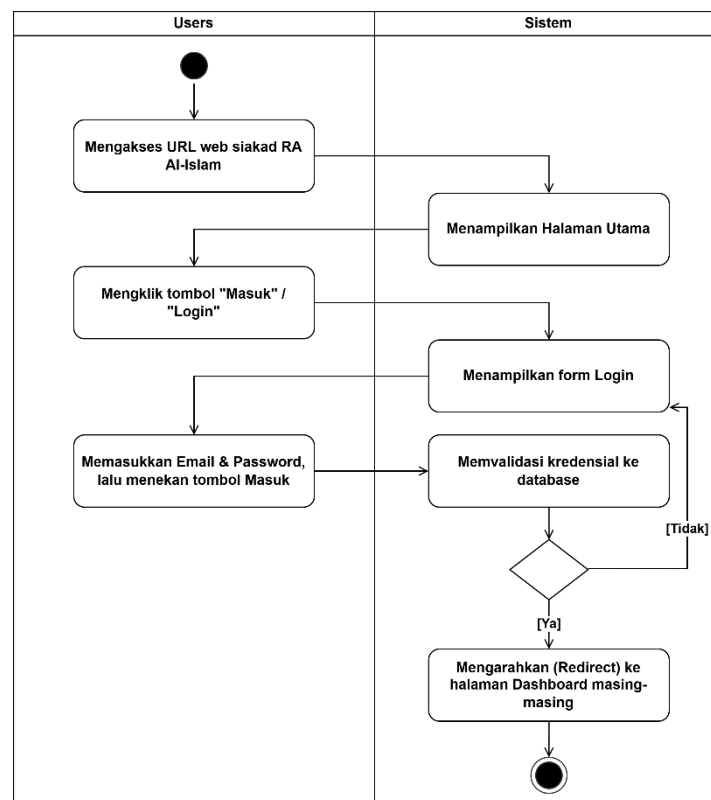
Pada setiap fitur yang dikembangkan, proses diawali dengan penyusunan rancangan menggunakan *Unified Modeling Language* (UML) sebagai acuan implementasi. Selanjutnya dilakukan proses pengembangan menggunakan Laravel 12 sebagai backend, React JS sebagai frontend, Shadcn UI sebagai komponen antarmuka, serta MySQL sebagai basis data. Hasil implementasi kemudian ditampilkan dalam bentuk antarmuka sistem untuk menunjukkan bahwa setiap kebutuhan fungsional telah berhasil direalisasikan.

5.4.1.1 Login

Fitur Login merupakan fitur awal yang dikembangkan pada *Sprint* 1 karena berfungsi sebagai gerbang utama bagi seluruh pengguna untuk mengakses sistem. Melalui fitur ini, pengguna dapat melakukan autentikasi menggunakan alamat email dan kata sandi yang telah terdaftar sehingga sistem dapat memberikan hak akses sesuai dengan peran masing-masing pengguna, yaitu Admin, Guru, maupun Orang Tua Siswa. Dengan adanya mekanisme autentikasi tersebut, keamanan data dapat lebih terjaga karena setiap pengguna hanya dapat mengakses fitur sesuai dengan kewenangannya.

Perancangan fitur Login dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses autentikasi sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas pengguna, struktur kelas yang terlibat, serta urutan komunikasi antara pengguna dengan sistem selama proses login berlangsung.

1. *Activity diagram login*



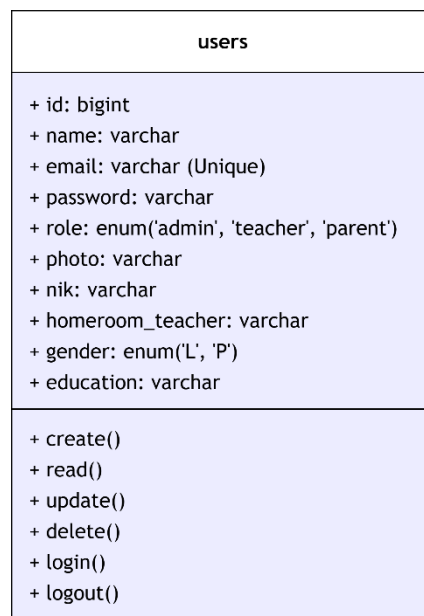
Gambar 5.6 *Activity diagram login*

Activity Diagram Login pada Gambar 5.6 menggambarkan alur proses autentikasi yang dilakukan oleh pengguna ketika mengakses sistem. Proses dimulai ketika pengguna membuka halaman Login kemudian memasukkan alamat email dan kata sandi. Selanjutnya sistem melakukan validasi terhadap data yang diberikan dengan mencocokkannya pada basis data pengguna.

Apabila data yang dimasukkan sesuai, sistem akan membuat sesi pengguna dan mengarahkan pengguna menuju halaman Dashboard sesuai dengan hak akses yang

dimiliki. Sebaliknya, apabila data tidak sesuai, sistem akan menampilkan pesan kesalahan sehingga pengguna dapat melakukan proses login kembali. Diagram tersebut menjadi acuan dalam proses implementasi agar mekanisme autentikasi berjalan sesuai dengan kebutuhan sistem.

2. *Class diagram login*

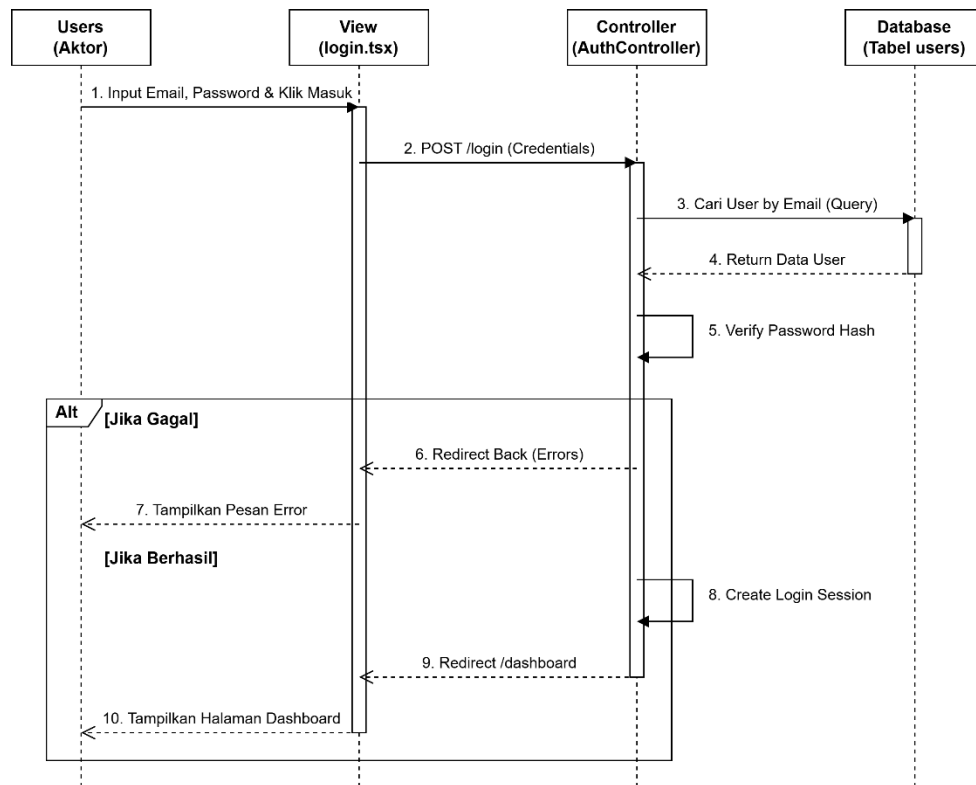


Gambar 5.7 *Class Diagram Login*

Class Diagram Login pada Gambar 5.7 digunakan untuk menggambarkan struktur kelas yang berperan dalam proses autentikasi pengguna. Diagram ini menunjukkan hubungan antar kelas beserta atribut dan operasi yang digunakan selama proses login berlangsung.

Melalui *Class Diagram*, struktur data yang digunakan dalam proses autentikasi dapat dirancang secara lebih sistematis sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan antara data pengguna dengan mekanisme autentikasi yang diterapkan pada sistem.

3. Sequence diagram login



Gambar 5.8 Sequence Diagram Login

Sequence Diagram Login pada Gambar 5.8 menggambarkan urutan komunikasi antara pengguna, antarmuka sistem, proses autentikasi, dan basis data selama proses login berlangsung. Diagram ini menunjukkan bagaimana permintaan login dikirimkan oleh pengguna, diproses oleh sistem, kemudian dilakukan validasi terhadap data yang terdapat pada basis data sebelum menghasilkan respons kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses login dapat dipahami secara lebih rinci. Diagram ini menjadi acuan dalam implementasi logika autentikasi sehingga proses pertukaran data antar komponen sistem dapat berjalan sesuai dengan kebutuhan yang telah dirancang.

4. Implementasi login

```

1   public function store(LoginRequest $request): RedirectResponse
2   {
3       // 1. Memvalidasi data email dan kata sandi yang dimasukkan pengguna
4       $user = $request->validateCredentials();
5
6       // (Optional) Pengecekan Autentikasi Dua Langkah (2FA) jika diaktifkan
7       if (Features::enabled(Features::twoFactorAuthentication()) && $user-
8   >hasEnabledTwoFactorAuthentication()) {
9           $request->session()->put([
10              'login.id' => $user->getKey(),
11              'login.remember' => $request->boolean('remember'),
12          ]);
13          return to_route('two-factor.login');
14      }
15
16      // 2. Membuat sesi pengguna (Login session) pada sistem
17      Auth::login($user, $request->boolean('remember'));
18
19      // Memperbarui session ID untuk keamanan
20      $request->session()->regenerate();
21
22      // 3. Mengarahkan pengguna menuju halaman Dashboard
23      return redirect()->intended(route('dashboard', absolute: false));
24  }

```

Gambar 5.9 Code Backend Login

Implementasi *backend* pada fitur Login dilakukan menggunakan framework Laravel 12. Potongan kode pada Gambar 5.9 menunjukkan proses autentikasi pengguna melalui mekanisme validasi email dan kata sandi yang dimasukkan. Apabila data autentikasi dinyatakan valid, sistem akan membuat sesi pengguna (*session*) kemudian mengarahkan pengguna menuju halaman *Dashboard* sesuai dengan hak akses masing-masing. Sebaliknya, apabila data tidak valid, sistem akan menampilkan pesan kesalahan sehingga pengguna dapat melakukan autentikasi kembali.

```

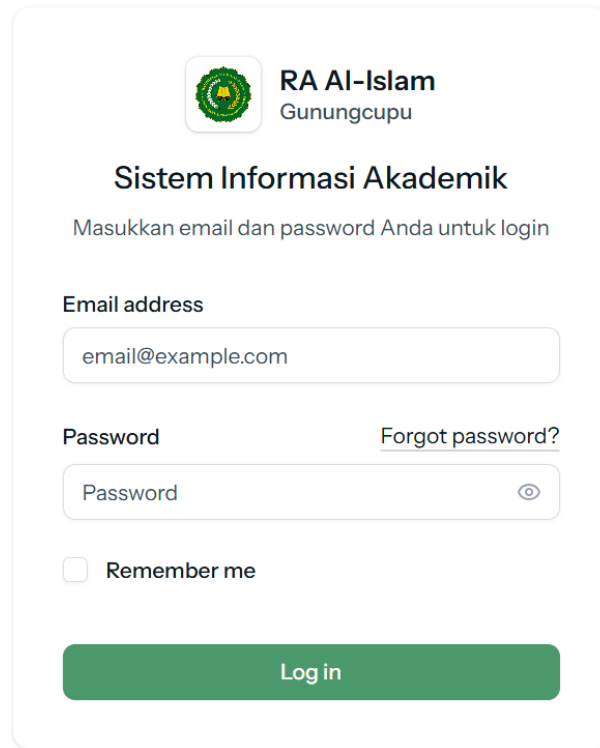
1 <Form
2   {...AuthenticatedSessionController.store.form()}
3   resetOnSuccess={['password']}
4   className="flex flex-col gap-6"
5 >
6   {{ { processing, errors } }} => (
7     <div className="grid gap-6">
8
9       {/* Kolom Email */}
10      <div className="grid gap-2">
11        <Label htmlFor="email">Email address</Label>
12        <Input
13          id="email" type="email" name="email" required autoFocus
14          placeholder="email@example.com"
15        />
16        <InputError message={errors.email} />
17      </div>
18
19      {/* Kolom Kata Sandi */}
20      <div className="grid gap-2">
21        <Label htmlFor="password">Password</Label>
22        <Input
23          id="password" type={showPassword ? "text" : "password"}
24          name="password" required
25          placeholder="Password"
26        />
27        <InputError message={errors.password} />
28      </div>
29
30      {/* Tombol Masuk */}
31      <Button type="submit" className="mt-4 w-full" disabled=
32        {processing}>
33        {processing && <LoaderCircle className="h-4 w-4 animate-spin"
34        />}
35        Log in
36      </Button>
37    </div>
38  )}
39 </Form>

```

Gambar 5.10 Code Frontend Login

Potongan kode frontend pada Gambar 5.10 digunakan untuk membangun antarmuka Login menggunakan React JS. Formulir Login terdiri atas kolom email, kolom kata sandi, dan tombol masuk yang berfungsi untuk mengirimkan data autentikasi ke backend. Dengan implementasi tersebut, proses komunikasi antara antarmuka dan server dapat berjalan dengan baik sehingga pengguna dapat melakukan login secara aman.

5. Tampilan Halaman Login



The image shows a login form for the 'Sistem Informasi Akademik' (Academic Information System) of 'RA Al-Islam Gunungcupu'. At the top left is a green circular logo. To its right, the text 'RA Al-Islam' is in bold, with 'Gunungcupu' below it. The title 'Sistem Informasi Akademik' is centered below the logo. Underneath the title is the instruction 'Masukkan email dan password Anda untuk login'. The form contains two input fields: 'Email address' with the placeholder 'email@example.com' and 'Password' with the placeholder 'Password'. To the right of the password field is a link 'Forgot password?'. Below the password field is a checkbox labeled 'Remember me'. At the bottom is a green 'Log in' button.

Gambar 5.11 Halaman *Login*

Gambar 5.11 menunjukkan hasil implementasi halaman *Login* pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menyediakan formulir autentikasi yang terdiri atas kolom email, kolom kata sandi, dan tombol Masuk sebagai media bagi pengguna untuk mengakses sistem.

Tampilan dirancang dengan antarmuka yang sederhana sehingga mudah dipahami oleh Admin, Guru, maupun Orang Tua. Selain itu, sistem juga memberikan informasi apabila data autentikasi yang dimasukkan tidak sesuai sehingga pengguna dapat melakukan perbaikan sebelum mencoba kembali. Dengan demikian, halaman Login tidak hanya berfungsi sebagai mekanisme keamanan sistem, tetapi juga memberikan kemudahan bagi pengguna dalam mengakses seluruh layanan sesuai dengan hak akses yang dimiliki.

5.4.1.2 Dashboard

Fitur Dashboard merupakan halaman utama yang ditampilkan setelah pengguna berhasil melakukan proses login ke dalam sistem. Pada *Sprint 1*, fitur ini dikembangkan setelah proses autentikasi selesai karena berfungsi sebagai pusat informasi awal bagi seluruh pengguna. Dashboard dirancang untuk menyajikan informasi akademik secara ringkas sesuai dengan hak akses masing-masing pengguna sehingga Admin, Guru, maupun Orang Tua dapat langsung memperoleh informasi yang dibutuhkan tanpa harus membuka setiap menu secara satu per satu.

Implementasi Dashboard menerapkan mekanisme *role-based access control* sehingga informasi yang ditampilkan berbeda sesuai dengan peran pengguna yang sedang mengakses sistem. Admin memperoleh informasi yang berkaitan dengan pengelolaan data akademik secara keseluruhan, Guru memperoleh informasi yang mendukung kegiatan pembelajaran dan pengelolaan siswa, sedangkan Orang Tua memperoleh informasi mengenai perkembangan akademik anaknya. Dengan penyajian informasi yang disesuaikan berdasarkan hak akses tersebut, Dashboard diharapkan dapat meningkatkan efisiensi pengguna dalam memperoleh informasi sejak pertama kali masuk ke dalam sistem.

1. Implementasi dashboard

```

1  public function index()
2  {
3      $user = auth()->user();
4
5      // 1. Logika untuk Dashboard Orang Tua
6      if ($user->role === 'parent') {
7          // Mengambil data anak yang berelasi dengan akun orang tua ini
8          $children = Student::with('studentClass.schoolYear')
9              ->where('user_id', $user->id)
10             ->get();
11          // Mengarahkan ke tampilan dashboard khusus orang tua
12          return Inertia::render('parent/dashboard', ['children' =>
13              $children]);
14      }
15
16      // 2. Logika untuk Dashboard Guru
17      if ($user->role === 'teacher') {
18          // Mengarahkan ke rute dashboard khusus ruang lingkup guru
19          return redirect()->route('teacher.dashboard');
20      }
21
22      // 3. Logika untuk Dashboard Admin (Pusat Kendali)
23      // Menghitung seluruh metrik akademik sekolah
24      $total_students = Student::count();
25      $active_students = Student::where('status', 'aktif')->count();
26
27      $total_classes = StudentClass::count();
28      $total_teachers = User::where('role', 'teacher')->count();
29
30      // Mengirimkan hasil kalkulasi metrik ke antarmuka React
31      return Inertia::render('dashboard', [
32          'total_students' => $total_students,
33          'active_students' => $active_students,
34          'total_classes' => $total_classes,
35          'total_teachers' => $total_teachers,
36      ]);
37  }

```

Gambar 5.12 Code Backend Dashboard

Implementasi backend pada fitur Dashboard dilakukan menggunakan framework Laravel 12. Potongan kode pada Gambar 5.12 menunjukkan proses pengambilan data dari basis data yang kemudian diolah sesuai dengan hak akses pengguna yang sedang login. Data tersebut meliputi informasi akademik yang dibutuhkan sebagai ringkasan pada halaman Dashboard sehingga setiap pengguna hanya memperoleh data yang sesuai dengan kewenangannya.

Proses tersebut memanfaatkan mekanisme autentikasi dan manajemen hak akses yang telah dibangun pada fitur Login. Setelah data berhasil diperoleh, sistem akan

mengirimkan informasi tersebut ke sisi frontend melalui proses pertukaran data sehingga Dashboard dapat menampilkan informasi secara dinamis. Dengan demikian, penyajian informasi pada Dashboard dapat dilakukan secara cepat, akurat, dan sesuai dengan kebutuhan masing-masing pengguna.

```

1  export default function Dashboard({
2    total_students, total_classes, total_teachers
3  }: DashboardProps) {
4    return (
5      <AppLayout breadcrumbs={breadcrumbs}>
6        <Head title="Dashboard Pusat" />
7
8        <div className="flex flex-col gap-6 p-6">
9          <div className="bg-card p-8 rounded-2xl shadow-sm">
10             <h1 className="text-3xl font-extrabold">Selamat Datang di
11             Pusat Kendali</h1>
12             <p className="text-muted-foreground">Sistem Informasi
13             Akademik berjalan stabil.</p>
14           </div>
15
16           { /* Grid Kartu Metrik Keseluruhan */ }
17           <div className="grid gap-4 sm:grid-cols-3">
18             <div className="bg-card p-6 border rounded-2xl">
19               <p>Total Peserta Didik</p>
20               <h3 className="text-4xl font-black">{total_students}
21             </h3>
22             </div>
23             <div className="bg-card p-6 border rounded-2xl">
24               <p>Rombongan Belajar</p>
25               <h3 className="text-4xl font-black">{total_classes}
26             </h3>
27             </div>
28             <div className="bg-card p-6 border rounded-2xl">
29               <p>Tenaga Pendidik</p>
30               <h3 className="text-4xl font-black">{total_teachers}
31             </h3>
32             </div>
33           </div>
34         </div>
35       </AppLayout>
36     );
37   }

```

Gambar 5.13 *Code Frontend Dashboard Admin*

Potongan kode pada Gambar 5.13 menunjukkan implementasi antarmuka Dashboard untuk Admin menggunakan React JS. Halaman ini dirancang untuk menampilkan informasi yang berkaitan dengan pengelolaan data akademik secara keseluruhan sehingga Admin dapat memantau kondisi sistem dan mengakses menu administrasi sesuai dengan hak akses yang dimiliki.

```

1  export default function TeacherDashboard({
2    totalClasses, upcomingAgendas
3  }: TeacherDashboardProps) {
4    return (
5      <AppLayout breadcrumbs={breadcrumbs}>
6        <Head title="Dashboard Guru" />
7        <div className="flex flex-col gap-6 p-6">
8          <div>
9            <h1 className="text-2xl font-bold">Selamat Datang, Guru!
10          </h1>
11          <p className="text-muted-foreground">Kelola data absensi
12            harian dari portal ini.</p>
13          <div>
14            <div className="grid gap-6 sm:grid-cols-3">
15              <Card>
16                <CardTitle>Total Kelas Anda</CardTitle>
17                <CardContent>
18                  <div className="text-3xl font-bold">{totalClasses}
19                </div>
20              </CardContent>
21            </Card>
22            <Card>
23              <CardTitle>Absensi Harian</CardTitle>
24              <CardContent>
25                <Link href="/teacher/attendances" className="bg-
26                  secondary text-secondary-foreground">
27                  Isi Absensi Sekarang
28                </Link>
29              </CardContent>
30            </Card>
31          </div>
32        </div>
33      </AppLayout>
34    );
35  }

```

Gambar 5.14 *Code Frontend Dashboard Guru*

Potongan kode pada Gambar 5.14 merupakan implementasi antarmuka Dashboard Guru. Halaman ini menyajikan informasi yang mendukung aktivitas guru, seperti akses menuju data siswa, absensi, maupun informasi akademik lainnya. Penyajian informasi disesuaikan dengan kebutuhan operasional guru sehingga penggunaan sistem menjadi lebih efektif.

```

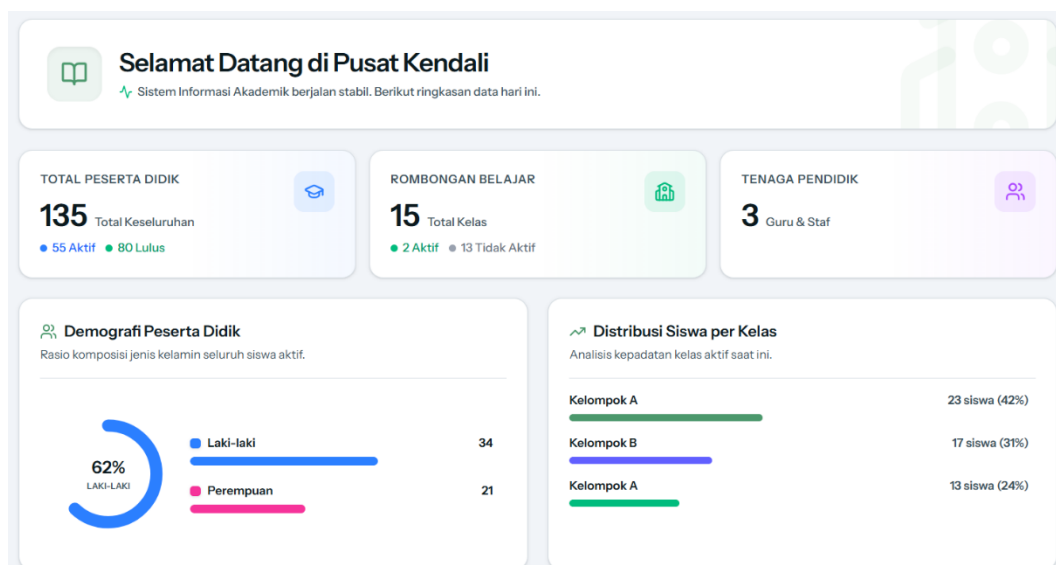
1  export default function ParentDashboard({
2    children, upcoming_events
3  }: ParentDashboardProps) {
4    const { auth } = usePage<SharedData>().props;
5    return (
6      <AppLayout breadcrumbs={breadcrumbs}>
7        <Head title="Beranda Orang Tua" />
8        <div className="flex flex-col gap-6 p-4">
9          <div className="bg-blue-600 p-8 text-white rounded-2xl">
10             <h1 className="text-3xl font-extrabold">Selamat Datang,
11             {auth.user.name}</h1>
12           </div>
13           <h2 className="text-xl font-bold mt-4">Informasi Anak Asuh</h2>
14
15           <div className="grid gap-6 sm:grid-cols-2">
16             {children.map((child) => (
17               <div key={child.id} className="border rounded-2xl
18               overflow-hidden bg-white">
19                 <div className="p-6">
20                   <h3 className="text-lg font-bold">
21                     {child.full_name}</h3>
22                   <p>NIS: {child.nis || '-'}</p>
23                   <div className="grid grid-cols-2 border-t">
24                     <Link href={`~/parent/students/${child.id}`}>
25                       Profil Lengkap
26                     </Link>
27                     <Link href=
28                       {`~/parent/students/${child.id}/academic-records`}>
29                       Riwayat Akademik
30                     </Link>
31                   </div>
32                 </div>
33               </div>
34             ))}
35           </div>
36         </div>
37       </AppLayout>
38     );
39   }

```

Gambar 5.15 *Code Frontend Dashboard Orang Tua Siswa*

Potongan kode pada Gambar 5.15 menunjukkan implementasi antarmuka Dashboard Orang Tua. Halaman ini dirancang untuk memberikan akses kepada orang tua terhadap informasi akademik anaknya, seperti kehadiran, profil siswa, dan perkembangan akademik. Informasi yang ditampilkan dibatasi sesuai hak akses sehingga keamanan data tetap terjaga.

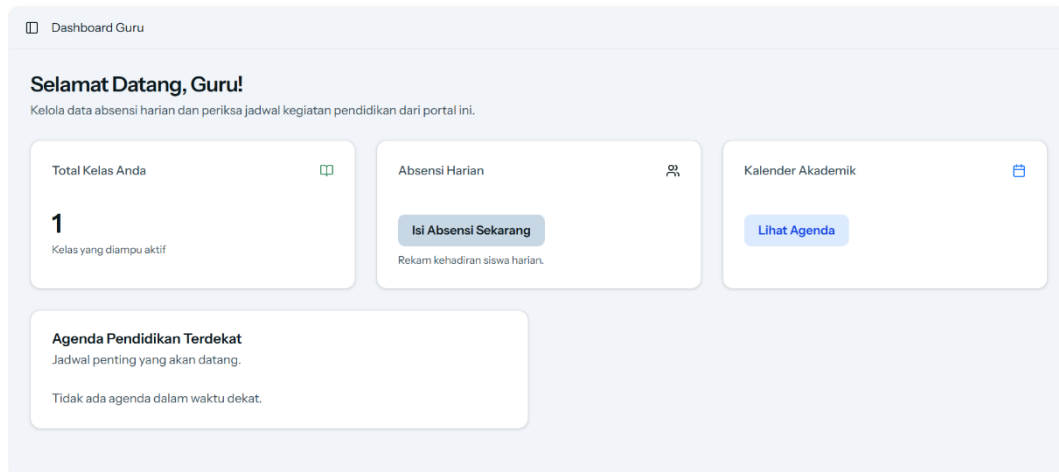
2. Tampilan halaman dashboard



Gambar 5.16 Halaman Dashboard Admin

Gambar 5.16 menunjukkan hasil implementasi halaman Dashboard Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini ditampilkan setelah Admin berhasil melakukan proses login dan berfungsi sebagai halaman utama yang menyajikan informasi akademik secara ringkas. Dashboard Admin menampilkan berbagai informasi yang berkaitan dengan pengelolaan data pengguna, siswa, kelas, tahun pelajaran, absensi, serta penilaian sebagai bahan pemantauan aktivitas akademik di sekolah.

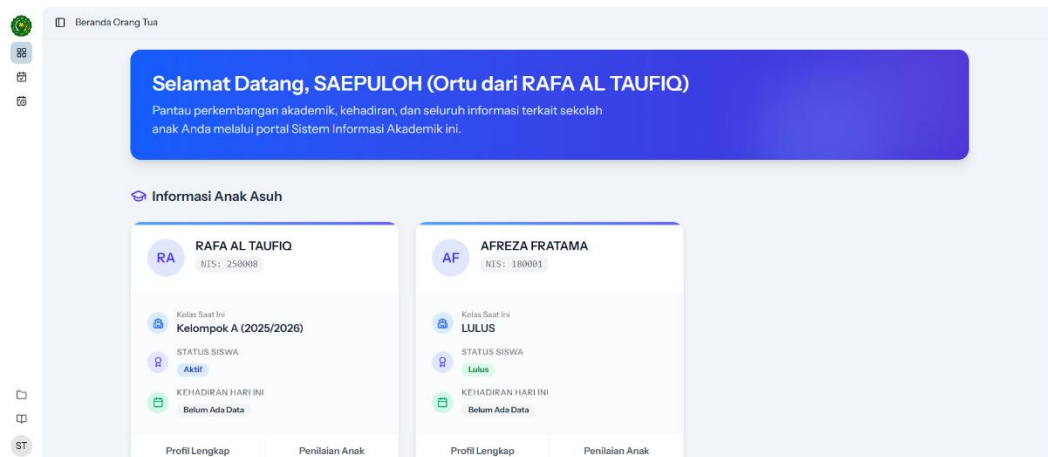
Tampilan Dashboard Admin dirancang dengan antarmuka yang sederhana, terstruktur, dan mudah dipahami sehingga memudahkan Admin dalam mengakses setiap menu yang tersedia. Selain menyajikan informasi secara ringkas, halaman ini juga berfungsi sebagai pusat navigasi menuju seluruh fitur pengelolaan akademik sesuai dengan hak akses Admin sehingga proses pengelolaan data dapat dilakukan secara lebih efektif dan efisien.



Gambar 5.17 Halaman Dashboard Guru

Gambar 5.17 menunjukkan hasil implementasi halaman Dashboard Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini ditampilkan setelah Guru berhasil melakukan proses login dan berfungsi sebagai halaman utama yang menyajikan informasi sesuai dengan kebutuhan proses pembelajaran. Dashboard Guru memberikan akses terhadap menu yang berkaitan dengan pengelolaan data siswa, pencatatan absensi, serta informasi akademik lainnya yang menjadi tanggung jawab guru.

Tampilan Dashboard Guru dirancang dengan antarmuka yang sederhana sehingga mudah digunakan dalam mendukung kegiatan akademik sehari-hari. Penyusunan menu dan informasi disesuaikan dengan hak akses Guru sehingga pengguna dapat mengakses fitur yang diperlukan secara lebih cepat tanpa harus menampilkan informasi yang tidak berkaitan dengan tugasnya.



Gambar 5.18 Halaman Dashboard Orang Tua Siswa

Gambar 5.18 menunjukkan hasil implementasi halaman Dashboard Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini ditampilkan setelah Orang Tua berhasil melakukan proses login dan berfungsi sebagai media untuk memperoleh informasi akademik anak secara mandiri. Dashboard menampilkan informasi yang berkaitan dengan profil siswa, kehadiran, kalender pendidikan, serta perkembangan akademik sesuai dengan data yang dimiliki oleh anak.

Tampilan Dashboard Orang Tua dirancang dengan antarmuka yang sederhana sehingga memudahkan pengguna dalam memperoleh informasi akademik tanpa harus datang langsung ke sekolah. Dengan adanya halaman ini, orang tua dapat memantau perkembangan akademik anak secara lebih cepat, sedangkan sekolah dapat menyampaikan informasi kepada orang tua melalui sistem sesuai dengan hak akses yang telah ditentukan.

5.4.1.3 Pengguna (admin, Guru, Orang Tua Siswa)

Fitur Pengguna merupakan salah satu fitur yang dikembangkan pada *Sprint 1* sebagai bagian dari proses pengelolaan hak akses dalam Sistem Informasi Akademik RA Al-Islam Gunungcupu. Fitur ini digunakan oleh Admin untuk mengelola data akun pengguna yang terdiri atas Admin, Guru, dan Orang Tua

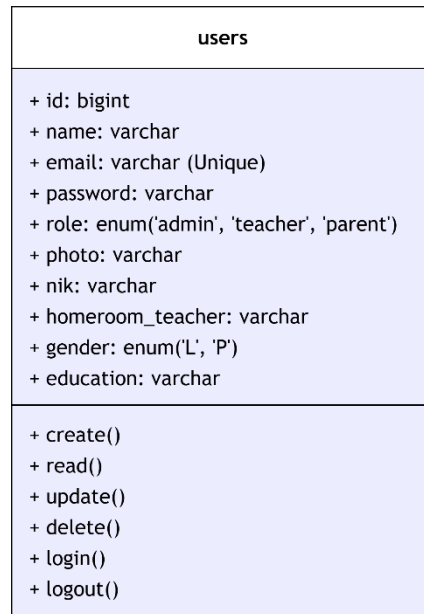
Siswa. Melalui fitur ini, Admin dapat melakukan penambahan, perubahan, penghapusan.

Pengembangan fitur Pengguna bertujuan untuk memastikan bahwa proses pengelolaan akun dapat dilakukan secara terpusat dan lebih mudah dibandingkan dengan pencatatan secara manual. Selain mendukung keamanan sistem melalui pengaturan hak akses, fitur ini juga memberikan kemudahan dalam proses administrasi apabila terjadi perubahan data pengguna atau pengguna mengalami kendala saat melakukan proses login. Sebelum proses implementasi dilakukan, fitur Pengguna terlebih dahulu dirancang menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai acuan dalam menggambarkan proses hubungan antar komponen sistem.

Activity Diagram Pengelolaan Data Admin pada Gambar 5.19 menggambarkan alur proses yang dilakukan oleh Admin dalam mengelola data akun administrator pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin mengakses halaman Dashboard kemudian memilih menu Pengguna dan submenu Admin sehingga sistem menampilkan halaman daftar data Admin. Selanjutnya Admin dapat menentukan tindakan yang akan dilakukan, yaitu menambahkan data baru, mengubah data yang sudah ada, atau menghapus data Admin sesuai dengan kebutuhan pengelolaan pengguna.

Apabila Admin memilih menambahkan data, sistem akan menampilkan formulir penambahan data untuk diisi sesuai informasi yang diperlukan. Setelah tombol Simpan dipilih, sistem melakukan proses validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Pada proses perubahan data, sistem terlebih dahulu menampilkan formulir yang berisi data Admin yang dipilih, kemudian Admin dapat memperbarui informasi tersebut dan menyimpannya kembali. Sementara itu, pada proses penghapusan, sistem menampilkan konfirmasi sebelum data dihapus dari basis data untuk menghindari kesalahan penghapusan. Dengan adanya mekanisme validasi dan konfirmasi tersebut, proses pengelolaan data Admin dapat dilakukan secara lebih terstruktur, aman, dan sesuai dengan kebutuhan sistem.

b. *Class diagram*

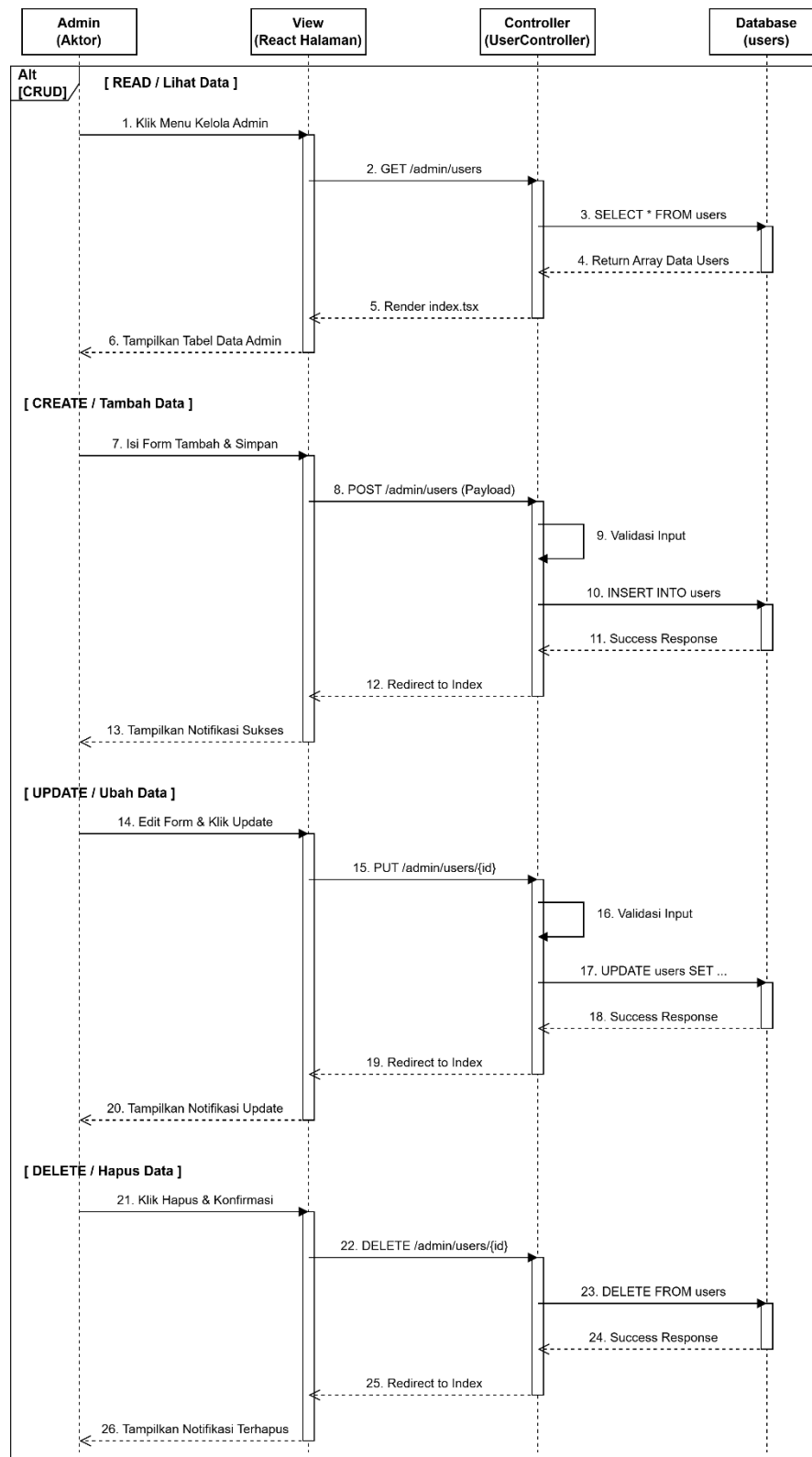


Gambar 5.20 *Class Diagram* Pengelolaan Data Admin

Class Diagram Pengelolaan Data Admin pada Gambar 5.20 digunakan untuk menggambarkan struktur kelas yang berperan dalam pengelolaan data pengguna pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Diagram ini menampilkan satu kelas utama, yaitu *Users*, yang berfungsi sebagai representasi data seluruh pengguna sistem, meliputi Admin, Guru, dan Orang Tua Siswa. Setiap pengguna dibedakan berdasarkan atribut *role*, sehingga meskipun menggunakan struktur data yang sama, sistem tetap dapat memberikan hak akses sesuai dengan peran masing-masing pengguna.

Kelas *Users* memiliki sejumlah atribut yang digunakan untuk menyimpan informasi pengguna, seperti *id*, *name*, *email*, *password*, *role*, *photo*, *nik*, *homeroom_teacher*, *gender*, dan *education*. Selain atribut tersebut, kelas ini juga memiliki beberapa operasi, yaitu *create()*, *read()*, *update()*, *delete()*, *login()*, dan *logout()* yang mendukung proses pengelolaan data serta autentikasi pengguna. Dengan perancangan tersebut, struktur data pengguna menjadi lebih terpusat dan konsisten sehingga implementasi proses *CRUD* maupun pengelolaan hak akses dapat

dilakukan secara lebih terarah, mudah dipelihara, serta mendukung kebutuhan pengembangan sistem pada fitur-fitur berikutnya.

c. *Sequence diagram*Gambar 5.21 *Sequence Diagram* Pengelolaan Data Admin

Sequence Diagram Pengelolaan Data Admin pada Gambar 5.21 menggambarkan urutan interaksi antara aktor Admin, antarmuka sistem (*React JS*), *controller* pada Laravel, dan basis data selama proses pengelolaan data admin berlangsung. Diagram tersebut menunjukkan bahwa seluruh proses *Create*, *Read*, *Update*, dan *Delete (CRUD)* dilakukan melalui komunikasi yang terstruktur antara setiap komponen sistem sehingga setiap permintaan dari pengguna dapat diproses secara sistematis.

Proses diawali ketika Admin membuka halaman pengelolaan data admin sehingga sistem mengambil data dari basis data dan menampilkannya pada halaman utama. Selanjutnya Admin dapat menambahkan, mengubah, maupun menghapus data sesuai kebutuhan. Setiap permintaan yang dikirimkan akan divalidasi terlebih dahulu oleh *controller* sebelum dilakukan proses penyimpanan, perubahan, atau penghapusan data pada basis data. Setelah proses berhasil dilakukan, sistem mengirimkan respons kepada antarmuka untuk memperbarui tampilan data dan menampilkan notifikasi keberhasilan kepada pengguna. *Sequence Diagram* ini menjadi acuan implementasi agar komunikasi antara *frontend*, *backend*, dan basis data berjalan secara konsisten sesuai dengan rancangan sistem.

d. Implementasi

```

1  public function index(Request $request)
2  {
3      $search = $request->input('search');
4      $entries = $request->input('entries', 10);
5
6      // Membaca jenis pengguna (role) dari parameter URL rute
7      $role = $request->route('role') ?? $request->query('role');
8
9      // Kueri pengambilan data pengguna secara dinamis
10     $users = User::query()
11         ->when($role, function ($query, $role) {
12             // Memfilter data sesuai dengan hak akses (admin/teacher/parent)
13             $query->where('role', $role);
14         })
15         ->when($search, function ($query, $search) {
16             // Menerapkan logika pencarian berdasarkan nama atau email
17             $query->where(function($q) use ($search) {
18                 $q->where('name', 'like', '%' . $search . '%')
19                 ->orWhere('email', 'like', '%' . $search . '%');
20             });
21         })
22         ->orderBy('name', 'asc')
23         ->paginate($entries)
24         ->withQueryString();
25
26     // Mengirimkan data yang sudah diproses ke antarmuka React JS
27     return Inertia::render('admin/users/index', [
28         'users' => $users,
29         'role' => $role,
30         'search' => $search,
31     ]);
32 }

```

Gambar 5.22 Code Backend Pengelolaan Data Admin

Potongan kode pada Gambar 5.22 menunjukkan implementasi logika *Backend* untuk fitur Pengelolaan Data Pengguna pada kelas *UserController* menggunakan kerangka kerja Laravel. Fungsi *index()* ini dirancang secara dinamis untuk menangani pengambilan data semua tipe pengguna (Admin, Guru, atau Orang Tua) hanya dari satu kontroler melalui pengecekan parameter rute. Kode ini juga menerapkan fitur pencarian (*search*) dan pembagian halaman (*pagination*) secara langsung pada tingkatan kueri basis data (*database query*), sehingga menghasilkan performa pemuatan data yang ringan dan optimal sebelum dikirimkan ke antarmuka aktor.

```

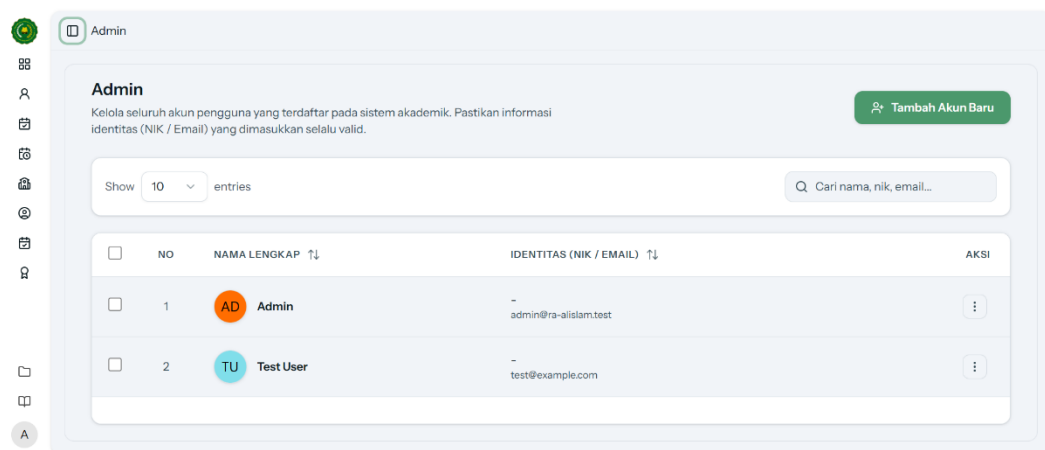
1 export default function UserIndex({ users, search, role }: UserIndexProps) {
2   // Menyesuaikan judul halaman secara dinamis berdasarkan parameter role
3   const pageTitle = role === 'admin' ? 'Admin' : role === 'teacher' ? 'Guru' :
  'Orang Tua';
4
5   return (
6     <AppLayout breadcrumbs={[{ title: pageTitle, href: '#' }]}>
7       <Head title={pageTitle} />
8
9       <div className="p-4 sm:p-6">
10        {/* Bagian Header dan Tombol Tambah */}
11        <div className="mb-6 flex flex-col sm:flex-row justify-between gap-4">
12          <h2 className="text-xl font-bold">{pageTitle}</h2>
13          <Link href={route('admin.users.create')} className="bg-primary
  text-white px-4 py-2 rounded-lg font-semibold">
14            Tambah Akun Baru
15          </Link>
16        </div>
17
18        {/* Bagian Tabel Data Pengguna */}
19        <div className="overflow-x-auto bg-card rounded-xl border">
20          <table className="w-full text-left text-sm">
21            <thead className="bg-muted text-muted-foreground uppercase
  text-xs">
22              <tr>
23                <th className="px-6 py-4">NO</th>
24                <th className="px-6 py-4">NAMA LENGKAP</th>
25                <th className="px-6 py-4">IDENTITAS (NIK / EMAIL)</th>
26                <th className="px-6 py-4 text-right">AKSI</th>
27              </tr>
28            </thead>
29            <tbody className="divide-y divide-border">
30
31              {/* Proses perulangan/mapping data ke baris tabel */}
32              {users.data.map((user, index) => (
33                <tr key={user.id} className="hover:bg-muted/40
  transition-colors">
34                <td className="px-6 py-4">{index + 1}</td>
35                <td className="px-6 py-4 font-bold">{user.name}</td>
36                <td className="px-6 py-4">
37                  <div className="font-medium text-foreground">
38                    {user.nik || '-'}</div>
39                  <div className="text-xs text-muted-
  foreground">{user.email}</div>
40                </td>
41                <td className="px-6 py-4 text-right">
42                  <Link href={route('admin.users.edit',
  user.id)} className="text-primary hover:underline">
43                    Edit Data
44                  </Link>
45                </td>
46              </tr>
47            </tbody>
48          </table>
49        </div>
50      </div>
51    </AppLayout>
52  );
53 }

```

Gambar 5.23 Code Frontend Pengelolaan Data Admin

Potongan kode pada Gambar 5.23 menunjukkan implementasi antarmuka fitur Pengelolaan Data Pengguna menggunakan *React JS*. Halaman ini dirancang untuk menampilkan daftar pengguna di dalam tabel data interaktif (*Data Table*). Pada potongan kode tersebut, sistem menggunakan fungsi *mapping* (*users.data.map*) untuk merender baris-baris data dari basis data secara otomatis ke dalam tabel *HTML*. Antarmuka ini juga dilengkapi dengan elemen navigasi tindakan (*action button*) seperti fitur Edit dan Hapus, sehingga Admin dapat mengelola seluruh entitas akun pengguna secara terpusat dari satu halaman.

e. Tampilan halaman



Gambar 5.24 Halaman Pengelolaan Data Admin

Gambar 5.24 menunjukkan hasil implementasi halaman Pengelolaan Data Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar akun admin yang telah terdaftar dalam bentuk tabel yang memuat informasi identitas pengguna, seperti nama lengkap dan email. Selain itu, tersedia tombol Tambah Akun Baru, fasilitas pencarian data, serta menu aksi yang digunakan untuk melakukan proses ubah maupun hapus data admin.

Tampilan dirancang dengan antarmuka yang sederhana dan terstruktur sehingga memudahkan administrator dalam mengelola akun pengguna. Pengelompokan fungsi berdasarkan tombol aksi dan penyajian data dalam bentuk tabel membantu proses pengelolaan data menjadi lebih cepat dan mudah dipahami. Dengan demikian, halaman Pengelolaan Data Admin tidak hanya mendukung proses

administrasi akun secara efisien, tetapi juga membantu menjaga keakuratan dan konsistensi data pengguna yang digunakan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu.

Gambar 5.25 Halaman Tambah Data Admin

Gambar 5.25 menampilkan antarmuka formulir penambahan akun administrator baru yang dirancang dengan alur pengisian data runtut meliputi pengunggahan foto profil, nama lengkap, alamat email, kata sandi, konfirmasi kata sandi, dan penentuan peran. Secara analitis, struktur masukan ini dibuat untuk meminimalkan beban kognitif pengguna sekaligus menerapkan mekanisme validasi ganda (*double-layer validation*) pada sisi *frontend* dan *backend* guna menjamin keunikan alamat email serta ketaatan format data. Dari aspek keamanan siber, data kata sandi yang diinput dilindungi oleh token *CSRF* dan dienkripsi menggunakan algoritma *Bcrypt* sebelum disimpan ke basis data, sehingga memastikan proses pendaftaran pengguna baru beroperasi secara aman, terstruktur, dan terhindar dari potensi data anomali.

Gambar 5.26 Halaman Edit Data Admin

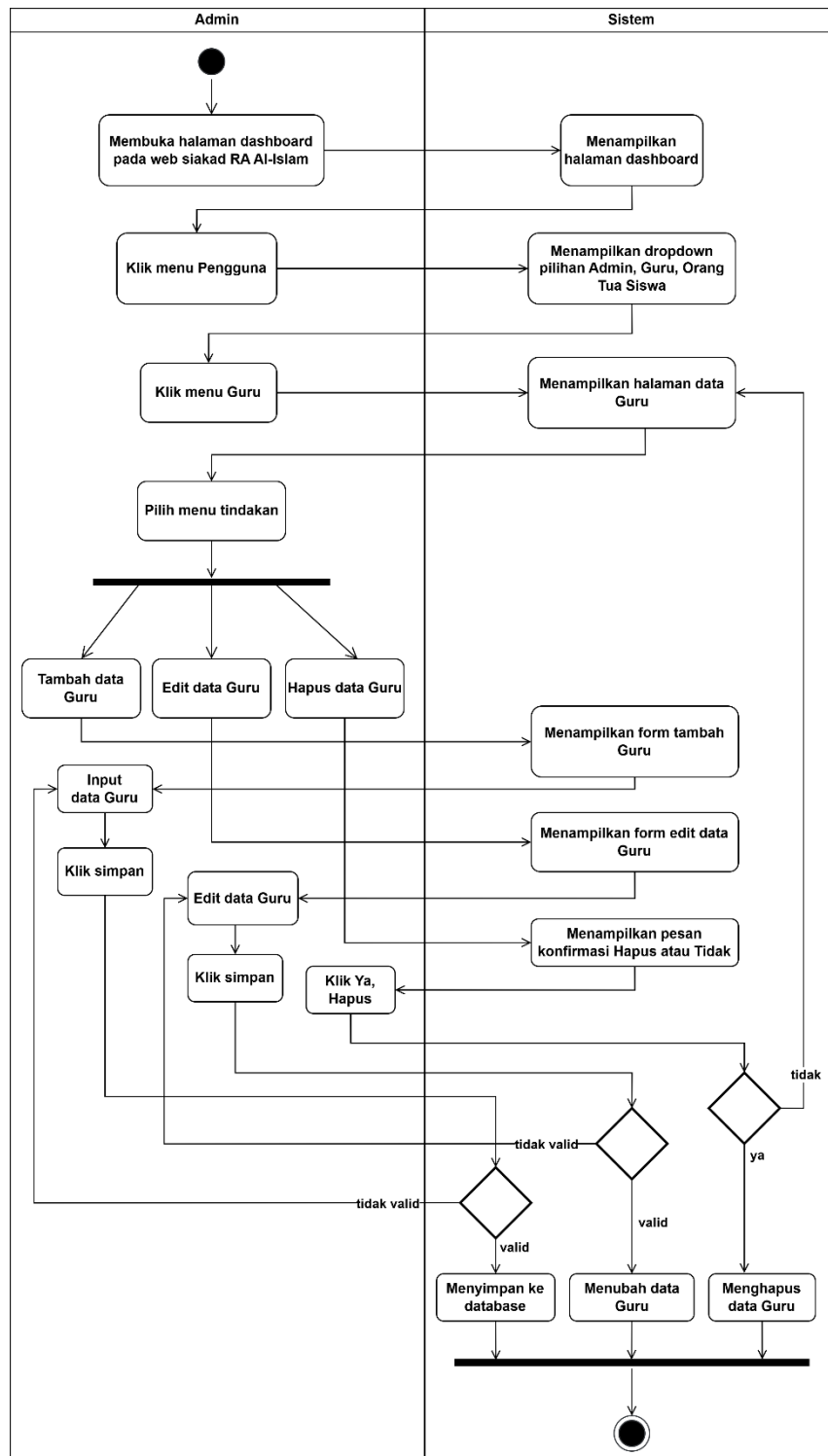
Gambar 5.26 memperlihatkan halaman pembaruan (*Update*) data administrator yang secara otomatis memuat data lama (*pre-filled form*) berdasarkan parameter ID yang dikirimkan via *route*. Ditinjau dari sudut pandang *User Experience (UX)* dan keamanan, pemisahan logika pembaruan profil melalui petunjuk "*password (kosongkan jika tidak diubah)*" berfungsi secara efisien untuk mencegah perombakan *hash* kata sandi lama apabila pengguna hanya ingin memperbarui identitas non-kredensial. Pembaruan ini diimbangi dengan proteksi *Insecure Direct Object Reference (IDOR)* di sisi *backend* untuk memverifikasi otorisasi pengguna, sehingga pemeliharaan data akun dapat berjalan secara fleksibel dan adaptif tanpa mengorbankan integritas autentikasi sistem.

Gambar 5.27 Halaman Hapus Data Admin

Gambar 5.27 mengilustrasikan mekanisme penghapusan data (*Delete*) akun administrator menggunakan komponen jembatan berupa *Modal Overlay Confirmation*. Secara analitis, penerapan konfirmasi modal ini merupakan wujud prinsip *Fail-Safe Design* untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) akibat penekanan tombol yang tidak disengaja. Menghapus akun administrator merupakan proses sensitif yang dapat memutus hak akses operasional, sehingga penyediaan dialog konfirmasi eksplisit dengan pilihan tombol *Batal* dan *Hapus Permanen* ini menjamin bahwa seluruh manipulasi data berisiko tinggi dilakukan secara sadar, terverifikasi, dan akuntabel.

2. Pengelolaan data guru

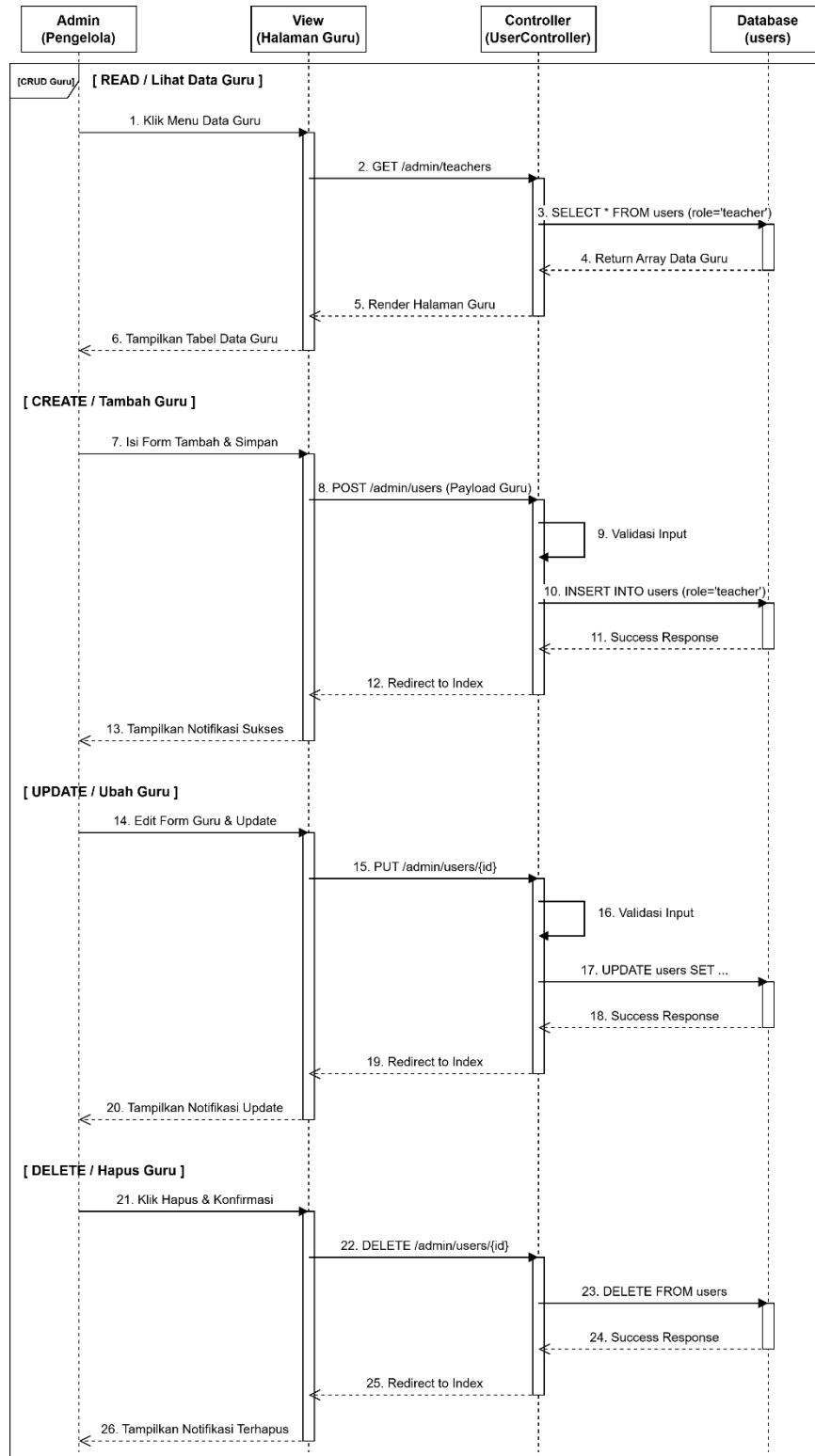
a. Activity diagram



Gambar 5.28 Activity Diagram Pengelolaan Data Guru

Activity Diagram Pengelolaan Data Guru pada Gambar 5.28 menggambarkan alur aktivitas yang dilakukan oleh Admin dalam mengelola data guru pada sistem. Proses dimulai ketika Admin membuka menu Pengguna, kemudian memilih submenu Guru sehingga sistem menampilkan halaman data guru yang berisi daftar akun guru yang telah terdaftar. Selanjutnya Admin dapat memilih tindakan untuk menambahkan, mengubah, maupun menghapus data guru sesuai dengan kebutuhan pengelolaan sistem.

Pada proses penambahan dan perubahan data, sistem terlebih dahulu menampilkan formulir yang harus diisi oleh Admin, kemudian melakukan validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, pada proses penghapusan data, sistem akan menampilkan konfirmasi sebagai bentuk validasi sebelum data benar-benar dihapus. *Activity Diagram* ini menjadi acuan dalam proses implementasi sehingga setiap alur pengelolaan data guru dapat berjalan secara sistematis, sesuai dengan kebutuhan fungsional yang telah dirancang.

b. *Sequence diagram*Gambar 5.29 *Sequence Diagram* Pengelolaan Data Guru

Sequence Diagram Pengelolaan Data Guru pada Gambar 5.29 menggambarkan urutan interaksi antara aktor Admin, antarmuka sistem (*React JS*), *controller* pada Laravel, dan basis data selama proses pengelolaan data guru berlangsung. Diagram tersebut menunjukkan bahwa seluruh proses *Create, Read, Update, dan Delete (CRUD)* dilakukan melalui komunikasi yang terstruktur antara setiap komponen sistem sehingga setiap permintaan dari pengguna dapat diproses secara sistematis.

Proses diawali ketika Admin membuka halaman pengelolaan data guru sehingga sistem mengambil data guru dari basis data dan menampilkannya pada halaman utama. Selanjutnya Admin dapat menambahkan, mengubah, maupun menghapus data guru sesuai kebutuhan. Setiap permintaan yang dikirimkan akan divalidasi terlebih dahulu oleh *controller* sebelum dilakukan proses penyimpanan, perubahan, atau penghapusan data pada basis data. Setelah proses berhasil dilakukan, sistem mengirimkan respons kepada antarmuka untuk memperbarui tampilan data dan menampilkan notifikasi keberhasilan kepada pengguna. *Sequence Diagram* ini menjadi acuan implementasi agar komunikasi antara *frontend, backend*, dan basis data berjalan secara konsisten sesuai dengan rancangan sistem.

c. Implementasi

```

1  public function store(Request $request)
2  {
3      // Melakukan validasi data yang dikirimkan dari formulir frontend
4      $validated = $request->validate([
5          'name' => ['required', 'string', 'max:255'],
6          'email' => ['required', 'email', 'max:255', 'unique:users,email'],
7          'password' => ['required', 'string', 'min:8', 'confirmed'],
8          'role' => ['required', 'in:admin,teacher,parent'],
9
10         // Validasi atribut spesifik khusus untuk entitas Guru (Teacher)
11         'nik' => ['nullable', 'string', 'max:255'],
12         'gender' => ['nullable', 'in:L,P'],
13         'education' => ['nullable', 'string', 'max:255'],
14         'homeroom_teacher' => ['nullable', 'string', 'max:255'],
15     ]);
16
17     // Menyimpan data yang telah divalidasi ke dalam tabel users
18     User::create([
19         'name' => $validated['name'],
20         'email' => $validated['email'],
21         'password' => Hash::make($validated['password']),
22         'role' => $validated['role'],
23         // Jika atribut kosong, sistem akan mengisinya dengan null
24         'nik' => $validated['nik'] ?? null,
25         'gender' => $validated['gender'] ?? null,
26         'education' => $validated['education'] ?? null,
27         'homeroom_teacher' => $validated['homeroom_teacher'] ?? null,
28     ]);
29
30     return to_route('admin.teachers.index')->with('success', 'Data Guru berhasil
    ditambahkan.');
```

Gambar 5.30 Code Backend Pengelolaan Data Guru

Potongan kode pada Gambar 5.30 menunjukkan implementasi logika *Backend* untuk fitur penambahan Data Guru. Pada kontroler ini, logika penyimpanan data dirancang menggunakan fungsi validasi bawaan kerangka kerja Laravel (*\$request->validate()*). Secara spesifik, karena entitas Guru digabung dalam satu tabel Pengguna, sistem diinstruksikan untuk melakukan validasi tambahan pada atribut yang hanya dimiliki oleh tenaga pendidik, seperti NIK, Jenis Kelamin, Pendidikan Terakhir, dan penempatan Wali Kelas. Pendekatan validasi yang ketat di sisi server ini diimplementasikan untuk mencegah masuknya data anomali atau tidak valid ke dalam basis data.

```

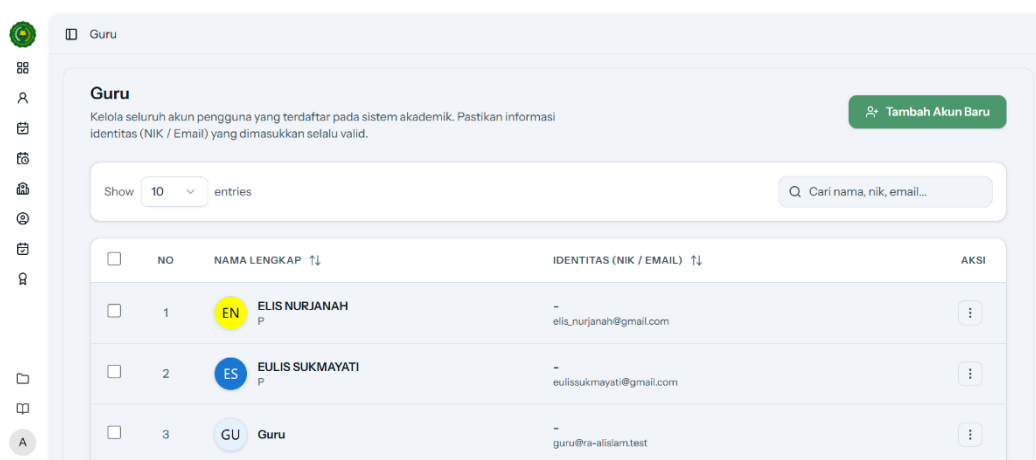
1  export default function CreateUser() {
2    // Menginisialisasi state formulir menggunakan hook dari Inertia
3    const { data, setData, post, processing } = useForm({
4      name: '', email: '', password: '', role: 'teacher',
5      nik: '', homeroom_teacher: '', gender: '', education: ''
6    });
7
8    const handleSubmit = (e: FormEvent) => {
9      e.preventDefault();
10     post(route('admin.users.store')); // Mengirim payload data ke backend
11   };
12
13   return (
14     <form onSubmit={handleSubmit}>
15       <div className="mb-4">
16         <Label htmlFor="name">Nama Lengkap</Label>
17         <Input value={data.name} onChange={(e) => setData('name',
18           e.target.value)} required />
19       </div>
20       <div className="mb-4">
21         <Label htmlFor="role">Peran Pengguna</Label>
22         <Select value={data.role} onChange={(value) => setData('role',
23           value)}>
24           <SelectItem value="admin">Admin</SelectItem>
25           <SelectItem value="teacher">Guru</SelectItem>
26           <SelectItem value="parent">Orang Tua</SelectItem>
27         </Select>
28       </div>
29       {/* Conditional Rendering: Kolom muncul jika role yang dipilih adalah
30        Guru/Orang Tua */}
31       {(data.role === 'teacher' || data.role === 'parent') && (
32         <div className="mb-4">
33           <Label htmlFor="nik">Nomor Induk Kependudukan (NIK)</Label>
34           <Input value={data.nik} onChange={(e) => setData('nik',
35             e.target.value)} />
36         </div>
37         <div className="mb-4">
38           <Label htmlFor="education">Pendidikan Terakhir</Label>
39           <Input value={data.education} onChange={(e) =>
40             setData('education', e.target.value)} />
41         </div>
42         {/* Conditional Rendering Spesifik: Kolom ini HANYA muncul untuk
43         Guru */}
44         {data.role === 'teacher' && (
45           <div className="mb-4">
46             <Label htmlFor="homeroom_teacher">Status Wali
47             Kelas</Label>
48             <Input value={data.homeroom_teacher} onChange={(e) =>
49               setData('homeroom_teacher', e.target.value)} />
50           </div>
51         )}
52       </>
53     </form>
54   );
55 }

```

Gambar 5.31 Code Frontend Pengelolaan Data Guru

Potongan kode pada Gambar 5.31 menunjukkan implementasi antarmuka *Frontend* untuk formulir penambahan Data Guru menggunakan *React JS*. Halaman ini memanfaatkan *hook useForm* dari pustaka *Inertia* untuk mengelola status (*state*) input formulir secara waktu nyata (*real-time*). Secara teknis, antarmuka ini mengimplementasikan metode *conditional rendering* (perenderan bersyarat). Kolom isian spesifik seperti NIK, Pendidikan, dan Wali Kelas hanya akan dimunculkan di layar apabila nilai variabel *data.role* bernilai "teacher" (Guru). Teknik ini dipilih untuk menjaga antarmuka formulir tetap bersih dan meminimalisir potensi kesalahan penginputan oleh administrator.

d. Tampilan Halaman



Gambar 5.32 Halaman Pengelolaan Data Guru

Gambar 5.32 menunjukkan hasil implementasi halaman Pengelolaan Data Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar akun guru yang telah terdaftar dalam bentuk tabel yang memuat informasi identitas pengguna, seperti nama lengkap dan email. Selain itu, tersedia tombol Tambah Akun Baru, fasilitas pencarian data, serta menu aksi yang digunakan untuk melakukan proses ubah maupun hapus data guru.

Tampilan dirancang dengan antarmuka yang sederhana dan terstruktur sehingga memudahkan administrator dalam mengelola akun guru. Pengelompokan fungsi berdasarkan tombol aksi dan penyajian data dalam bentuk tabel membantu proses

pengelolaan data menjadi lebih cepat dan mudah dipahami. Dengan demikian, halaman Pengelolaan Data Guru tidak hanya mendukung proses administrasi akun guru secara efisien, tetapi juga membantu menjaga keakuratan dan konsistensi data pengguna yang digunakan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu.

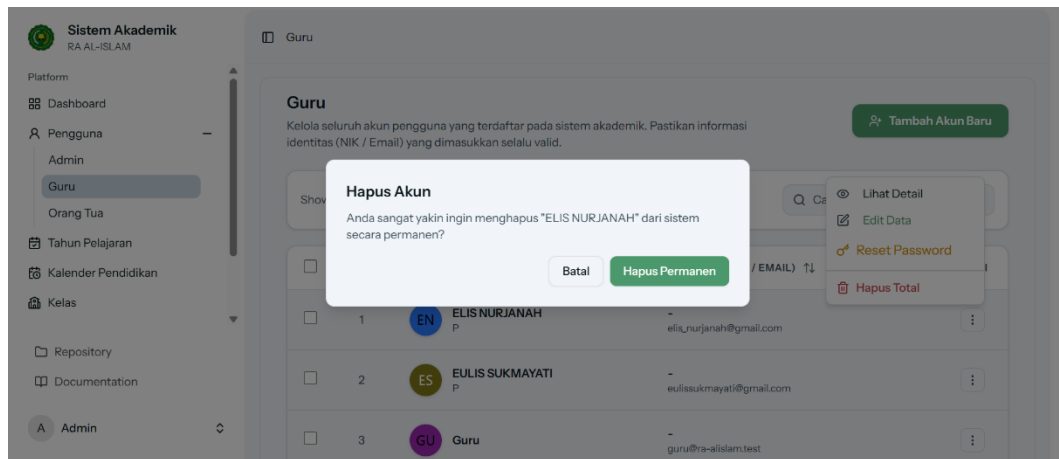
Gambar 5.33 Halaman Tambah Data Guru

Gambar 5.33 menampilkan antarmuka formulir pendaftaran akun guru baru yang dirancang untuk mengumpulkan informasi kredensial dan data kependidikan tenaga pendidik secara terstruktur. Secara analitis, formulir ini memanfaatkan teknik *conditional rendering* pada *frontend* React JS yang secara otomatis memunculkan atribut spesifik guru (seperti NIK, jenis kelamin, pendidikan terakhir, dan penetapan wali kelas) ketika opsi peran *teacher* dipilih. Di balik layar, *backend* Laravel mengeksekusi validasi ganda untuk memastikan keunikan format email serta mencocokkan konfirmasi kata sandi sebelum menyimpannya ke dalam basis data. Selain itu, keamanan akun dijamin melalui proteksi token *CSRF* dan enkripsi kata sandi menggunakan fungsi *hash Bcrypt*, sehingga proses penambahan data guru ini tidak hanya memberikan kemudahan pengisian bagi administrator, tetapi juga menjamin keandalan data dan keamanan autentikasi sejak akun dibuat.

The screenshot displays the 'Edit Pengguna' (Edit User) interface within the 'Sistem Akademik RA AL-ISLAM' application. On the left, a sidebar lists various system modules, with 'Admin' currently selected. The main panel shows the user management section, specifically the 'Edit Pengguna' form. The form contains several input fields: a file upload for 'Foto', a text field for 'Nama' (filled with 'ELIS NURJANAH'), an email field (filled with 'elis_nurjanah@gmail.com'), a password field with a hint '(kosongkan jika tidak diubah)', a confirmation password field, and a role selection field labeled 'Peran'.

Gambar 5.34 Halaman Edit Data Guru

Gambar 5.34 memperlihatkan antarmuka pembaruan (*Update*) data profil guru yang secara otomatis memuat informasi lama (*pre-filled form*) berdasarkan parameter ID pengguna yang dikirimkan melalui *route*. Ditinjau dari aspek rekayasa perangkat lunak, formulir ini menerapkan penanganan kata sandi yang fleksibel dengan instruksi "*password (kosongkan jika tidak diubah)*", yang secara kritis memisahkan logika pembaruan identitas operasional dari logika perombakan enkripsi kata sandi. Apabila kolom kata sandi tidak diisi, *controller backend* secara adaptif mempertahankan nilai *hash* lama di dalam tabel *Users*, sehingga mencegah terjadinya kegagalan autentikasi akibat perubahan yang tidak disengaja. Diimbangi dengan proteksi *Insecure Direct Object Reference (IDOR)* pada rantai otorisasi *backend*, pembaruan data guru ini memastikan pemeliharaan akun pendidik dapat dilakukan secara dinamis tanpa mengganggu stabilitas hak akses pengguna.



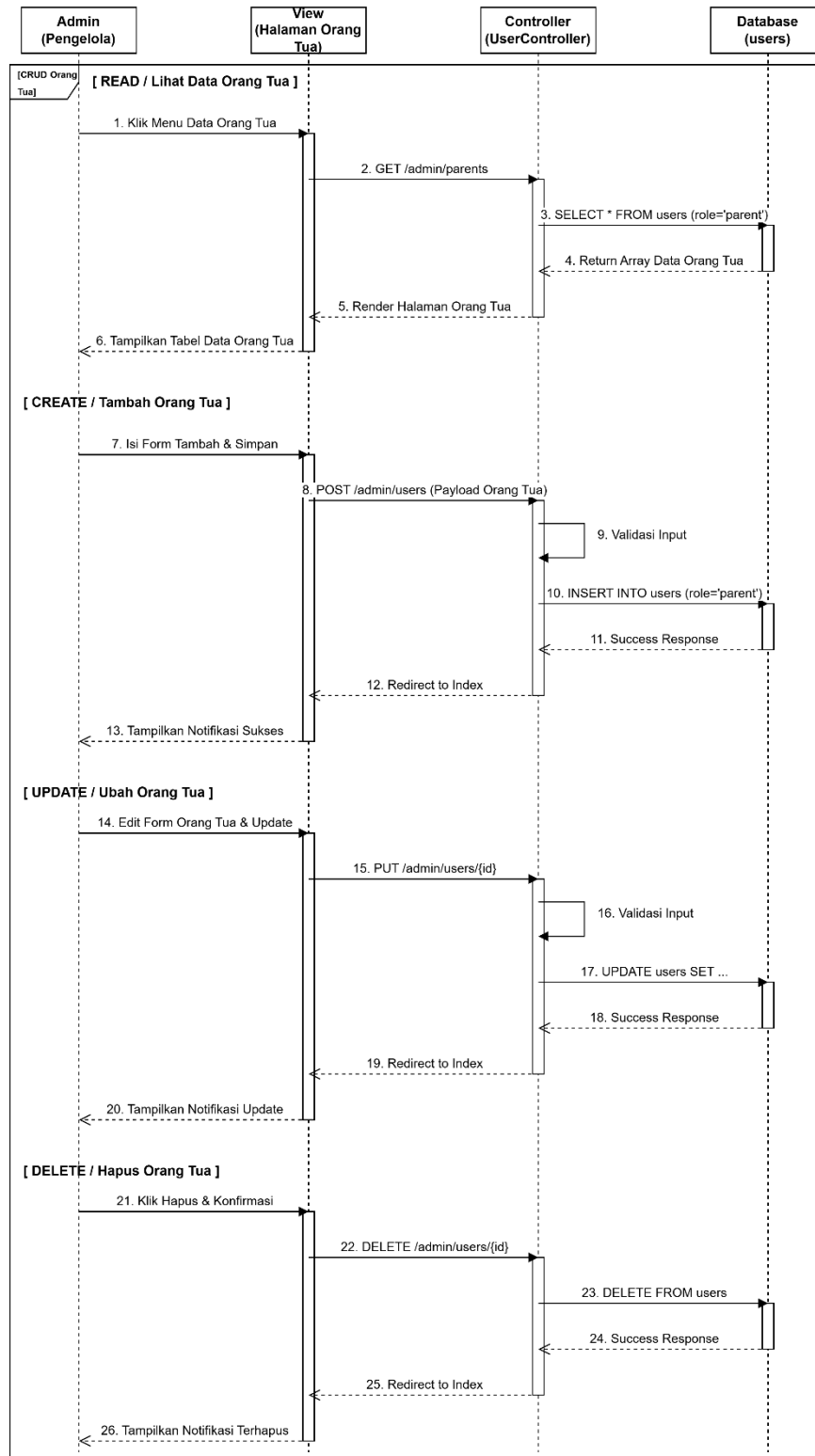
Gambar 5.35 Halaman Hapus Data Guru

Gambar 5.35 mengilustrasikan mekanisme penghapusan data (*Delete*) akun guru menggunakan komponen interaktif berupa *Modal Overlay Confirmation*. Secara analitis, penerapan jendela konfirmasi melayang ini merupakan bentuk implementasi prinsip *Fail-Safe Design* yang bertujuan untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) seperti penekanan tombol tanpa sengaja. Mengingat entitas guru memiliki keterkaitan relasional yang erat dengan tabel *student_classes* (sebagai wali kelas) dan *daily_attendances* (sebagai pencatat presensi), penghapusan akun secara permanen memerlukan verifikasi tingkat tinggi. Kehadiran *modal* dengan pilihan tombol *Batal* dan *Hapus Permanen* ini memaksa administrator untuk mengonfirmasi ulang keputusannya secara sadar, sehingga menjamin bahwa pembersihan data pengguna berisiko tinggi tetap terkendali, akuntabel, dan tidak merusak integritas relasional basis data.

Gambar 5.36 *Activity Diagram* Pengelolaan Data Orang Tua Siswa

Activity Diagram Pengelolaan Data Orang Tua Siswa pada Gambar 5.36 menggambarkan alur aktivitas yang dilakukan oleh Admin dalam mengelola data orang tua siswa pada sistem. Proses dimulai ketika Admin membuka menu Pengguna, kemudian memilih submenu Orang Tua Siswa sehingga sistem menampilkan halaman data orang tua siswa yang berisi daftar akun orang tua siswa yang telah terdaftar. Selanjutnya Admin dapat memilih tindakan untuk menambahkan, mengubah, maupun menghapus data orang tua siswa sesuai dengan kebutuhan pengelolaan sistem.

Pada proses penambahan dan perubahan data, sistem terlebih dahulu menampilkan formulir yang harus diisi oleh Admin, kemudian melakukan validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, pada proses penghapusan data, sistem akan menampilkan konfirmasi sebagai bentuk validasi sebelum data benar-benar dihapus. *Activity Diagram* ini menjadi acuan dalam proses implementasi sehingga setiap alur pengelolaan data orang tua siswa dapat berjalan secara sistematis, sesuai dengan kebutuhan fungsional yang telah dirancang.

b. *Sequence diagram*Gambar 5.37 *Sequence Diagram* Pengelolaan Data Orang Tua Siswa

Sequence Diagram Pengelolaan Data Orang Tua Siswa pada Gambar 5.37 menggambarkan urutan interaksi antara aktor Admin, antarmuka sistem (*React JS*), *controller* pada *Laravel*, dan basis data selama proses pengelolaan data orang tua siswa berlangsung. Diagram tersebut menunjukkan bahwa seluruh proses *Create*, *Read*, *Update*, dan *Delete (CRUD)* dilakukan melalui komunikasi yang terstruktur antara setiap komponen sistem sehingga setiap permintaan dari pengguna dapat diproses secara sistematis.

Proses diawali ketika Admin membuka halaman pengelolaan data orang tua siswa sehingga sistem mengambil data orang tua siswa dari basis data dan menampilkannya pada halaman utama. Selanjutnya Admin dapat menambahkan, mengubah, maupun menghapus data orang tua siswa sesuai kebutuhan. Setiap permintaan yang dikirimkan akan divalidasi terlebih dahulu oleh *controller* sebelum dilakukan proses penyimpanan, perubahan, atau penghapusan data pada basis data. Setelah proses berhasil dilakukan, sistem mengirimkan respons kepada antarmuka untuk memperbarui tampilan data dan menampilkan notifikasi keberhasilan kepada pengguna. *Sequence Diagram* ini menjadi acuan implementasi agar komunikasi antara *frontend*, *backend*, dan basis data berjalan secara konsisten sesuai dengan rancangan sistem.

c. Implementasi

```

1  public function show(Request $request, User $user)
2  {
3      // Membaca parameter peran (role) pengguna dari rute URL
4      $role = $request->query('role') ?? $request->route('role');
5
6      // Kondisi khusus untuk memeriksa entitas Orang Tua (Parent)
7      if ($user->role === 'parent') {
8          // Mengimplementasikan Eager Loading untuk memuat relasi data anak asuh
9          // secara bersamaan demi optimasi kecepatan kueri basis data
10         $user->load('children.studentClass');
11     }
12
13     // Mengirimkan objek data pengguna beserta relasinya ke antarmuka React JS
14     return Inertia::render('admin/users/show', [
15         'user' => $user,
16         'role' => $role
17     ]);
18 }

```

Gambar 5.38 *Code Backend* Pengelolaan Data Orang Tua Siswa

Potongan kode pada Gambar 5.38 menunjukkan implementasi logika *Backend* untuk melihat detail Pengelolaan Data Orang Tua Siswa. Pada fungsi *show()* di dalam *UserController*, sistem mengimplementasikan teknik *Eager Loading* yang direpresentasikan melalui sintaks *load('children.studentClass')*. Teknik ini dieksekusi secara dinamis hanya jika entitas pengguna yang sedang diakses memiliki peran sebagai Orang Tua Siswa. Pendekatan optimasi ini sangat krusial dalam rekayasa perangkat lunak karena mampu mencegah anomali N+1 Query Problem saat sistem harus menarik data relasional antara akun orang tua dengan banyak data anak asuhnya beserta kelas masing-masing dari basis data.

```

1 export default function ShowUser({ user }: { user: any }) {
2   return (
3     <AppLayout breadcrumbs={breadcrumbs}>
4       <Head title={`Profil: ${user.name}`} />
5
6       <div className="p-4 sm:p-6">
7         <div className="mb-8 rounded-xl border bg-card shadow-sm">
8           {/* Header Profil Utama */}
9           <div className="px-6 pb-6">
10            <h1 className="text-2xl font-bold">{user.name}</h1>
11            <span className="rounded-full bg-primary/10 px-2.5 py-0.5
text-xs font-semibold text-primary capitalize">
12              Orang Tua Wali
13            </span>
14          </div>
15        </div>
16
17        <div className="grid gap-6 md:grid-cols-2">
18
19          {/* Pemanggilan Komponen Modular untuk Informasi Dasar */}
20          <SectionCard title="Informasi Personal Dasar" icon={Users}>
21            <div className="grid gap-6 sm:grid-cols-2">
22              <InfoRow label="Nama Lengkap" value={user.name} />
23              <InfoRow label="NIK / Nomor Unik" value={user.nik} />
24              <InfoRow label="Email Kontak" value={user.email} />
25            </div>
26          </SectionCard>
27
28          {/* Conditional Rendering: Spesifik Khusus Entitas Orang Tua */}
29          {user.role === 'parent' && (
30            <SectionCard title="Pekerjaan & Data Lain" icon={Calendar}>
31              <div className="grid gap-6 sm:grid-cols-2">
32                <InfoRow label="Pekerjaan" value={user.occupation} />
33                <div className="sm:col-span-2">
34                  <p className="text-sm italic text-muted-
35                    foreground">
36                      (Profil wali murid ini akan tertaut langsung
37                      secara
38                      relasional terhadap data absensi dan akademik
39                      peserta didik.)
40                  </p>
41                </div>
42              </div>
43            </SectionCard>
44          )}
45        </div>
46      </AppLayout>
47    );
48  }

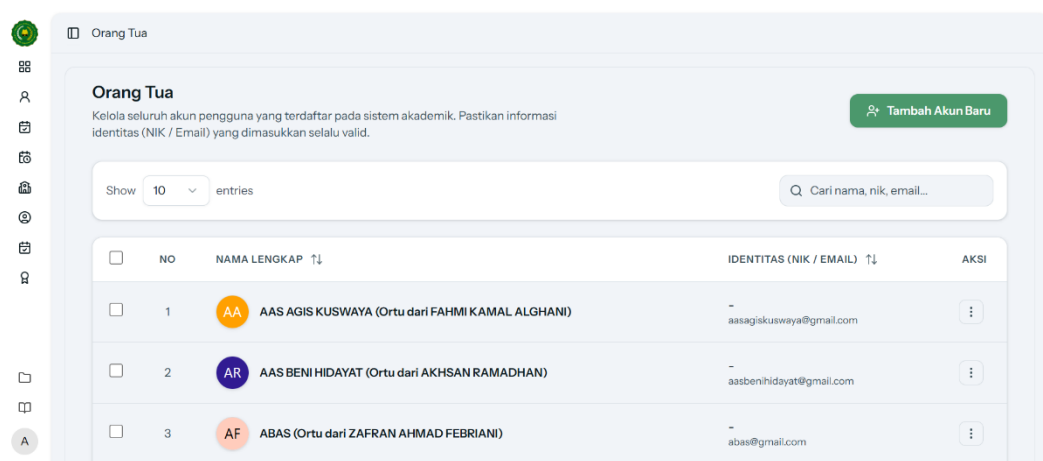
```

Gambar 5.39 Code Frontend Pengelolaan Data Orang Tua Siswa

Potongan kode pada Gambar 5.39 menunjukkan implementasi antarmuka *Frontend* untuk halaman Detail Profil Orang Tua Siswa menggunakan *React JS*. Halaman ini dirancang menggunakan arsitektur antarmuka berbasis komponen (*component-based architecture*) dengan membuat sub-komponen fungsional terpisah secara

kustom, seperti *SectionCard* dan *InfoRow*, untuk menyusun tata letak profil. Secara struktural, sistem mengandalkan perenderan bersyarat (*conditional rendering*) di mana blok informasi "Pekerjaan & Data Lain" hanya akan diproses dan ditampilkan di layar apabila variabel profil membawa nilai hak akses sebagai Orang Tua. Struktur penulisan komponen modular seperti ini menghasilkan kode yang sangat bersih, mudah dirawat (*maintainable*), dan minim duplikasi.

d. Tampilan halaman



Gambar 5.40 Halaman Pengelolaan Data Orang Tua Siswa

Gambar 5.40 menunjukkan hasil implementasi halaman Pengelolaan Data Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar akun orang tua siswa yang telah terdaftar dalam bentuk tabel yang memuat informasi identitas pengguna, seperti nama lengkap dan email. Selain itu, tersedia tombol Tambah Akun Baru, fasilitas pencarian data, serta menu aksi yang digunakan untuk melakukan proses ubah maupun hapus data orang tua siswa.

Tampilan dirancang dengan antarmuka yang sederhana dan terstruktur sehingga memudahkan administrator dalam mengelola akun orang tua siswa. Pengelompokan fungsi berdasarkan tombol aksi dan penyajian data dalam bentuk tabel membantu proses pengelolaan data menjadi lebih cepat dan mudah dipahami. Dengan demikian, halaman Pengelolaan Data Orang Tua Siswa tidak hanya

mendukung proses administrasi akun orang tua siswa secara efisien, tetapi juga membantu menjaga keakuratan dan konsistensi data pengguna yang digunakan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu.

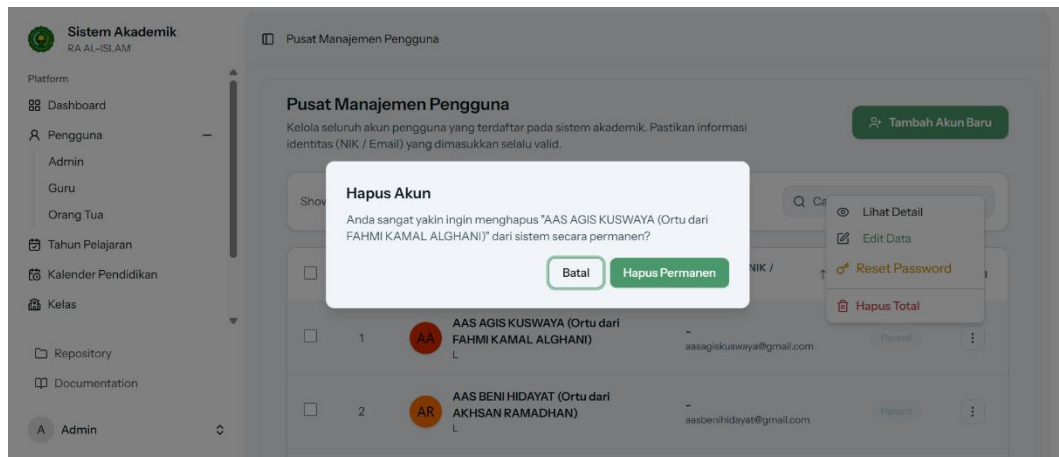
Gambar 5.41 Halaman Tambah Data Orang Tua Siswa

Gambar 5.41 menampilkan antarmuka formulir registrasi akun orang tua siswa baru yang dirancang untuk mengumpulkan data kredensial serta atribut identitas wali murid secara terstruktur. Secara analitis, formulir ini memfasilitasi pembuatan entitas pengguna berkewenangan *parent* yang nantinya akan ditautkan secara relasional (*foreign key*) dengan data siswa di basis data guna mendukung akses pemantauan perkembangan anak. Di balik layar, *backend* Laravel mengeksekusi proses validasi ganda untuk memastikan format email yang diinput bersifat unik (*unique constraint*) dan memverifikasi kesesuaian kata sandi melalui kolom konfirmasi. Selain itu, seluruh kredensial sensitif dilindungi oleh token *CSRF* dan dienkripsi menggunakan algoritma *Bcrypt* sebelum disimpan ke dalam tabel *Users*, sehingga memastikan pembuatan akun orang tua siswa berjalan secara terorganisir, transparan, dan terjamin keamanannya dari potensi manipulasi data.

The screenshot displays the 'Edit Pengguna' (Edit User) interface. On the left is a sidebar for the 'Sistem Akademik RA AL-ISLAM' with a 'Platform' section containing links to Dashboard, Pengguna (with sub-links for Admin, Guru, and Orang Tua), Tahun Pelajaran, Kalender Pendidikan, Kelas, Repository, Documentation, and a selected 'Admin' role. The main content area is titled 'Manajemen Pengguna > Edit Pengguna' and contains a form with the following fields: 'Foto' (with a 'Choose File' button), 'Nama' (pre-filled with 'AAS AGIS KUSWAYA (Ortu dari FAHMI KAMAL ALGHANI)'), 'Email' (pre-filled with 'aasegiskuswaya@gmail.com'), 'Password' (with the instruction '(kosongkan jika tidak diubah)'), 'Konfirmasi Password', and 'Peran' (partially visible as 'Guru Tua').

Gambar 5.42 Halaman Edit Data Orang Tua Siswa

Gambar 5.42 memperlihatkan antarmuka pembaruan (*Update*) data profil orang tua siswa yang secara otomatis menyajikan data lama (*pre-filled form*) berdasarkan parameter ID pengguna yang dikirimkan melalui *route*. Ditinjau dari aspek rekayasa perangkat lunak, formulir ini menerapkan penanganan kata sandi yang fleksibel melalui instruksi "*password (kosongkan jika tidak diubah)*", yang secara kritis memisahkan pembaruan identitas rutin dari perombakan enkripsi kredensial. Apabila kolom kata sandi dibiarkan kosong, *controller backend* secara adaptif mempertahankan nilai *hash* lama pada basis data, sehingga mencegah kegagalan autentikasi akibat penginputan yang tidak disengaja. Diimbangi dengan proteksi *Insecure Direct Object Reference (IDOR)* pada rantai otorisasi *backend*, pemeliharaan akun orang tua ini memastikan pembaruan data wali murid dapat dilakukan secara cepat dan aman tanpa merusak keterkaitan relasional terhadap akun anak.



Gambar 5.43 Halaman Hapus Data Orang Tua Siswa

Gambar 5.43 mengilustrasikan mekanisme penghapusan data (*Delete*) akun orang tua siswa menggunakan komponen interaktif berupa *Modal Overlay Confirmation*. Secara analitis, penerapan jendela konfirmasi melayang ini merupakan wujud prinsip *Fail-Safe Design* yang bertujuan untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) seperti penekanan opsi hapus secara tidak sengaja. Mengingat akun orang tua memiliki keterkaitan relasional (*foreign key*) terhadap tabel *students* untuk penyajian informasi presensi dan rapor akademik harian, penghapusan entitas akun ini memerlukan tingkat kehati-hatian ekstra. Kehadiran *modal* dengan pilihan tombol *Batal* dan *Hapus Permanen* ini memaksa administrator untuk memverifikasi ulang keputusannya secara sadar, sehingga menjamin bahwa pembersihan data pengguna berisiko tinggi tetap terkendali, terverifikasi, dan tidak menimbulkan anomali data liar (*orphaned records*) pada basis data.

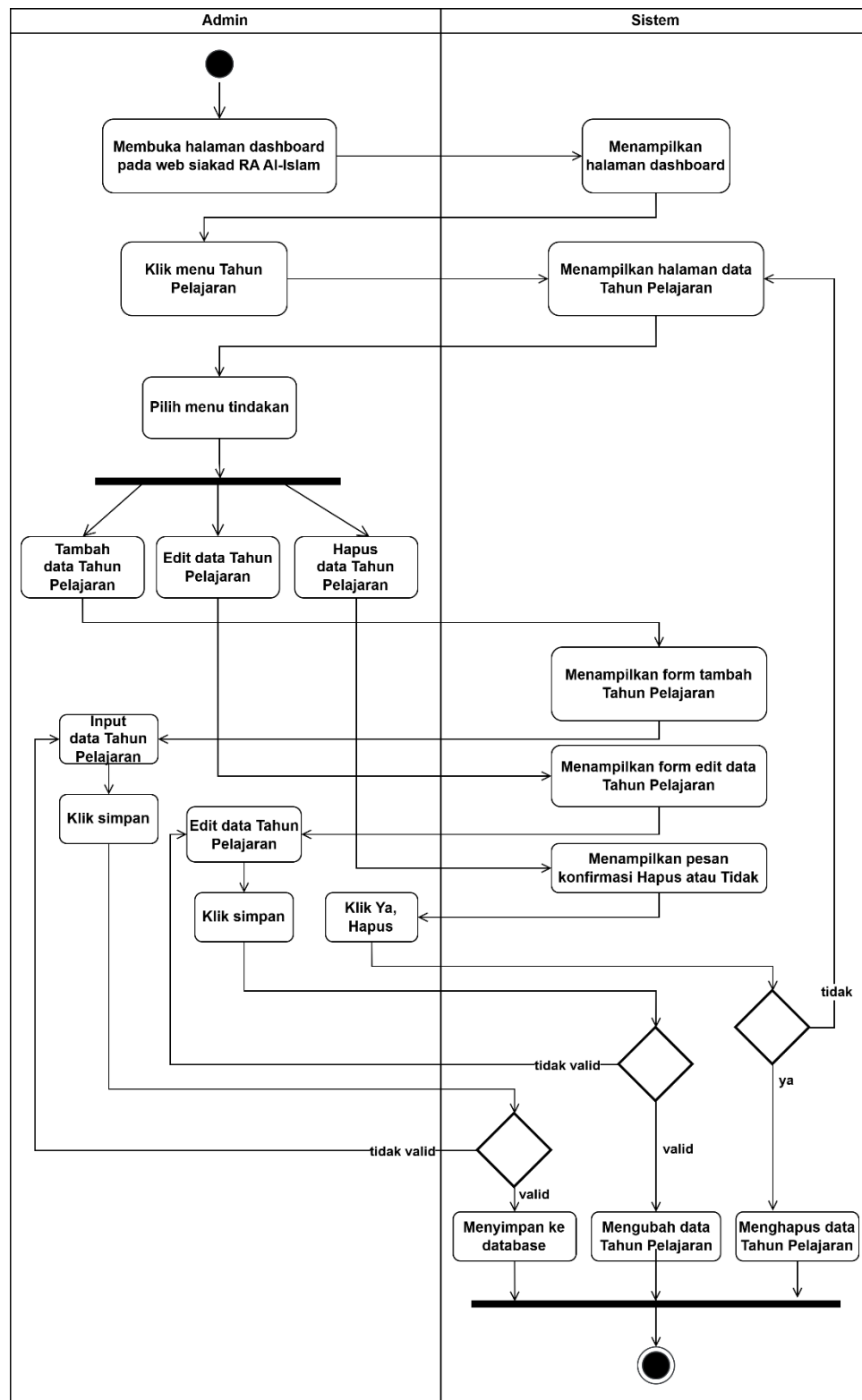
5.4.1.4 Tahun Pelajaran

Fitur Tahun Pelajaran merupakan salah satu fitur yang dikembangkan pada *Sprint* 1 untuk mendukung pengelolaan periode akademik pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Admin dapat melakukan pengelolaan data tahun pelajaran yang akan digunakan sebagai acuan dalam berbagai proses akademik, seperti pengelolaan kelas, absensi, maupun kegiatan akademik lainnya. Dengan adanya pengelolaan tahun pelajaran secara terpusat,

setiap data akademik dapat dikelompokkan berdasarkan periode yang sedang berjalan sehingga informasi yang tersimpan menjadi lebih terstruktur dan mudah dikelola.

Perancangan fitur Tahun Pelajaran dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur pengelolaan data sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas pengguna, struktur data yang digunakan, serta urutan komunikasi antara pengguna dengan sistem selama proses pengelolaan data Tahun Pelajaran berlangsung. Dengan adanya perancangan tersebut, proses implementasi dapat dilakukan secara lebih sistematis sehingga fungsi yang dikembangkan sesuai dengan kebutuhan pengguna.

1. *Activity diagram* tahun Pelajaran

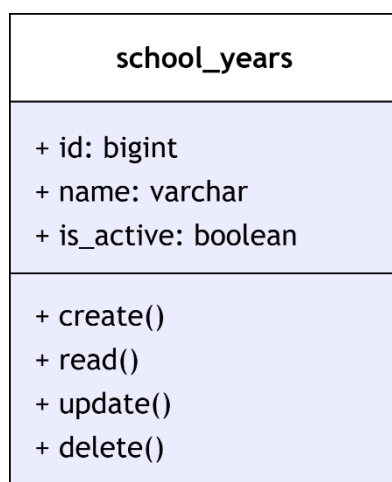


Gambar 5.44 *Activity Diagram* Tahun Pelajaran

Activity Diagram Tahun Pelajaran pada Gambar 5.44 menggambarkan alur proses pengelolaan data tahun pelajaran yang dilakukan oleh Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Tahun Pelajaran sehingga sistem menampilkan halaman data tahun pelajaran yang berisi seluruh data yang telah tersimpan pada basis data. Selanjutnya Admin dapat memilih tindakan berupa menambah data, mengubah data, maupun menghapus data tahun pelajaran sesuai dengan kebutuhan pengelolaan akademik.

Apabila Admin memilih tambah atau ubah data, sistem akan menampilkan formulir yang harus diisi kemudian melakukan proses validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, apabila Admin memilih menghapus data, sistem akan terlebih dahulu menampilkan konfirmasi penghapusan untuk menghindari kesalahan tindakan. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pengelolaan data tahun pelajaran dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram tahun pelajaran*



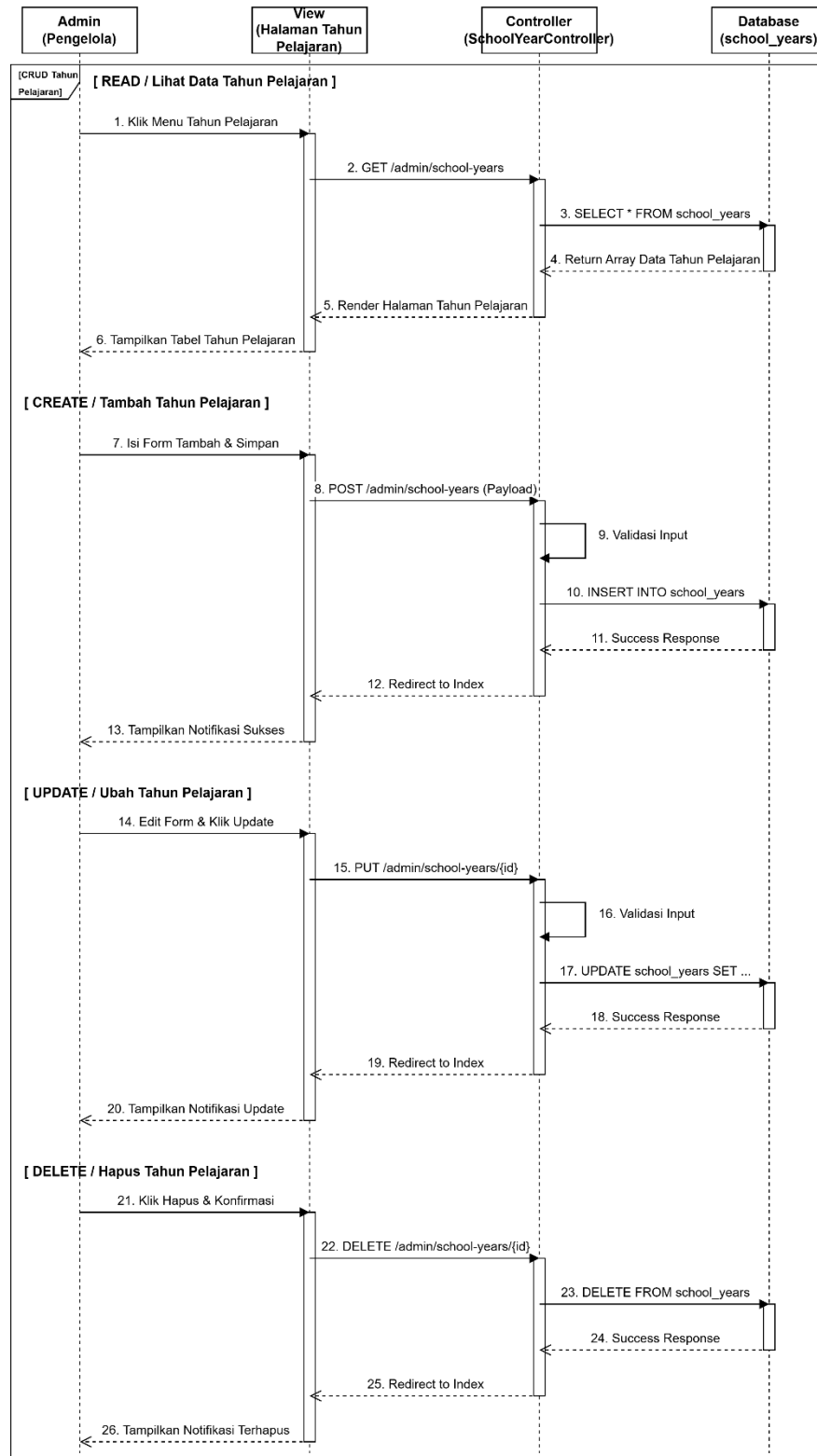
Gambar 5.45 *Class Diagram* Tahun Pelajaran

Class Diagram Tahun Pelajaran pada Gambar 5.45 digunakan untuk menggambarkan struktur kelas yang berperan dalam proses pengelolaan data tahun

pelajaran. Diagram ini menunjukkan atribut yang dimiliki oleh kelas *school_years*, yaitu *id*, *name*, dan *is_active*, beserta operasi yang digunakan untuk melakukan proses *Create*, *Read*, *Update*, dan *Delete* (*CRUD*) terhadap data tahun pelajaran.

Melalui *Class Diagram*, struktur data yang digunakan dalam proses pengelolaan tahun pelajaran dapat dirancang secara lebih sistematis sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan antara data tahun pelajaran dengan mekanisme pengelolaan yang diterapkan pada sistem sehingga setiap proses penambahan, perubahan, maupun penghapusan data dapat dilakukan secara konsisten sesuai dengan kebutuhan sistem.

3. Sequence diagram tahun Pelajaran

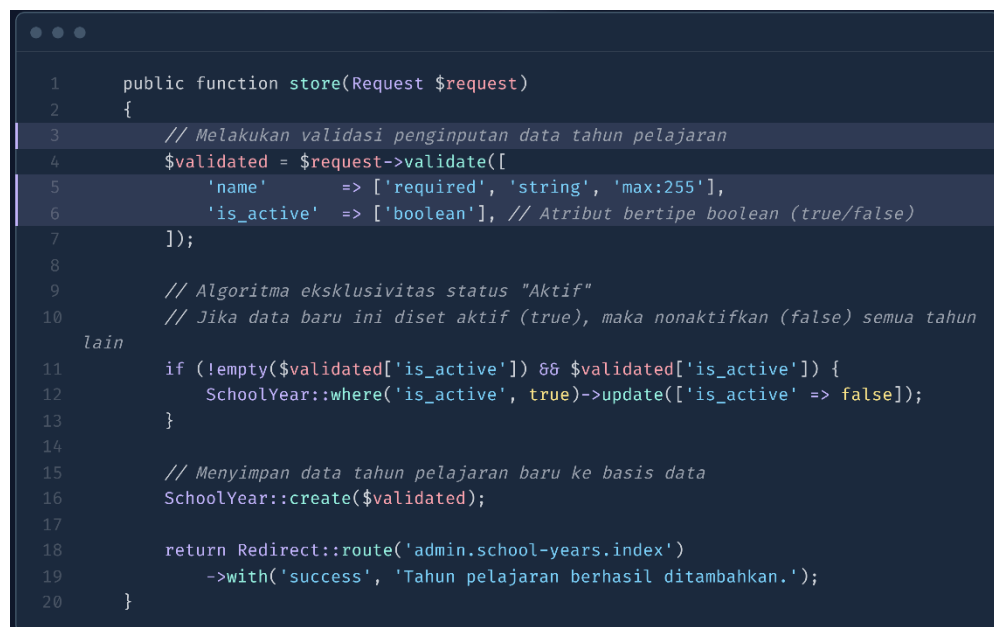


Gambar 5.46 Sequence Diagram Tahun Pelajaran

Sequence Diagram Tahun Pelajaran pada Gambar 5.46 menggambarkan urutan komunikasi antara Admin, antarmuka sistem, controller, dan basis data selama proses pengelolaan data tahun pelajaran berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data, menambahkan data baru, mengubah data, maupun menghapus data dikirimkan oleh Admin, kemudian diproses oleh *controller* melalui proses validasi sebelum diteruskan ke basis data hingga menghasilkan respons yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengelolaan data tahun pelajaran dapat dipahami secara lebih rinci. Diagram ini menjadi acuan dalam implementasi logika sistem sehingga proses pertukaran data antara antarmuka, *controller*, dan basis data dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi tahun Pelajaran



```

1  public function store(Request $request)
2  {
3      // Melakukan validasi penginputan data tahun pelajaran
4      $validated = $request->validate([
5          'name' => ['required', 'string', 'max:255'],
6          'is_active' => ['boolean'], // Atribut bertipe boolean (true/false)
7      ]);
8
9      // Algoritma eksklusivitas status "Aktif"
10     // Jika data baru ini diset aktif (true), maka nonaktifkan (false) semua tahun
11     lain
12     if (!empty($validated['is_active']) && $validated['is_active']) {
13         SchoolYear::where('is_active', true)->update(['is_active' => false]);
14     }
15
16     // Menyimpan data tahun pelajaran baru ke basis data
17     SchoolYear::create($validated);
18
19     return Redirect::route('admin.school-years.index')
20         ->with('success', 'Tahun pelajaran berhasil ditambahkan.');
```

Gambar 5.47 Code Backend Tahun Pelajaran

Potongan kode pada Gambar 5.47 menunjukkan implementasi logika *Backend* untuk fitur Pengelolaan Tahun Pelajaran pada *SchoolYearController*. Secara

fungsional, entitas ini sangat krusial karena tahun pelajaran yang berstatus "aktif" akan menjadi acuan variabel *global state* bagi seluruh operasional sistem SIAKAD (seperti penyaringan data nilai dan absensi). Pada potongan fungsi penyimpanan data (*store*), sistem diinstruksikan untuk menjalankan algoritma eksklusivitas status; apabila administrator menyimpan atau mengubah sebuah tahun pelajaran menjadi aktif (*is_active = true*), maka secara otomatis kerangka kerja Laravel akan melakukan kueri pembaruan terpusat (*bulk update*) untuk merubah nilai seluruh tahun pelajaran lain di basis data menjadi "*false*" (tidak aktif). Mekanisme operasional ini dieksekusi secara ketat guna menjamin integritas relasional data agar tidak terjadi tabrakan referensi waktu akademik.

```

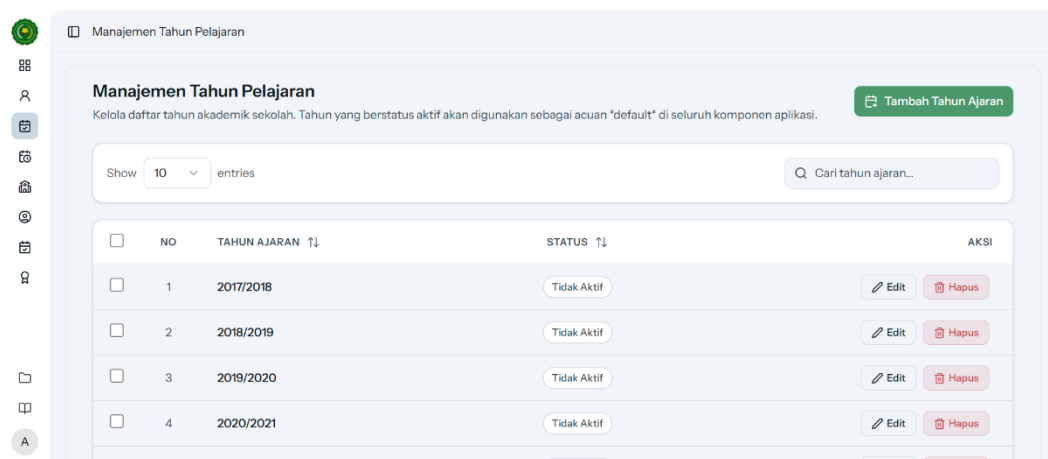
1  export default function SchoolYearIndex({ schoolYears }: SchoolYearIndexProps) {
2    return (
3      <AppLayout breadcrumbs={breadcrumbs}>
4        <Head title="Manajemen Tahun Ajaran" />
5
6        <div className="p-4 sm:p-6">
7
8          {/* Header Utama Antarmuka */}
9          <div className="mb-6 flex justify-between">
10             <div>
11               <h2 className="text-xl font-bold">Manajemen Tahun
12               Pelajaran</h2>
13             </div>
14             {/* Pemanggilan Komponen Kustom Modal secara Modular */}
15             <div className="flex shrink-0">
16               <AddSchoolYearModal />
17             </div>
18
19             {/* Struktur Data Table */}
20             <div className="overflow-x-auto bg-card rounded-xl border">
21               <table className="w-full text-left text-sm">
22                 <thead className="bg-muted text-muted-foreground uppercase
23                 text-xs">
24                   <tr>
25                     <th className="px-6 py-4">TAHUN AJARAN</th>
26                     <th className="px-6 py-4 text-center">STATUS</th>
27                     <th className="px-6 py-4 text-right">AKSI</th>
28                   </tr>
29                 </thead>
30                 <tbody className="divide-y divide-border">
31
32                   {/* Pemetaan (Mapping) Objek ke dalam HTML */}
33                   {schoolYears.data.map((item) => (
34                     <tr key={item.id} className="hover:bg-muted/40">
35                       <td className="px-6 py-4 font-medium">{item.name}
36
37                       <td className="px-6 py-4 text-center">
38                         {/* Ternary Operator untuk Perenderan Visual
39                         Status Aktif */}
40                         {item.is_active ? (
41                           <span className="rounded-full bg-green-
42                           500/10 px-2.5 py-1 text-green-600 font-semibold">
43                             Tahun Aktif
44                           </span>
45                         ) : (
46                           <span className="rounded-full bg-muted px-
47                           2.5 py-1 text-muted-foreground">
48                             Tidak Aktif
49                           </span>
50                         )}
51                       </td>
52                       <td className="px-6 py-4 text-right">
53                         {/* Pemanggilan Komponen Modal untuk Edit Data
54                         */}
55                         <EditSchoolYearModal schoolYear={item} />
56                       </td>
57                     </tr>
58                   ))}
59                 </tbody>
60               </table>
61             </div>
62           </div>
63         </AppLayout>
64       );
65     }

```

Gambar 5.48 Code Frontend Tahun Pelajaran

Potongan kode pada Gambar 5.48 menunjukkan implementasi antarmuka *Frontend* untuk Pengelolaan Data Tahun Pelajaran menggunakan pustaka *React JS*. Secara struktural, halaman ini mengimplementasikan konsep arsitektur modular, terbukti dengan penggunaan komponen *pop-up* eksternal (seperti `<AddSchoolYearModal />`) untuk mengeksekusi fungsi penambahan data tanpa harus melakukan navigasi perpindahan *URL routing*. Di dalam area tabel, sistem melakukan perenderan bersyarat (*conditional rendering*) menggunakan operasi logika *Ternary Operator* (`? :`) bawaan *JavaScript*. Logika ini berfungsi secara instan untuk merender representasi visual atau badge berwarna hijau apabila tahun pelajaran berstatus "Aktif", dan warna abu-abu apabila berstatus "Tidak Aktif", sehingga meminimalisir kesalahan persepsi visual bagi administrator.

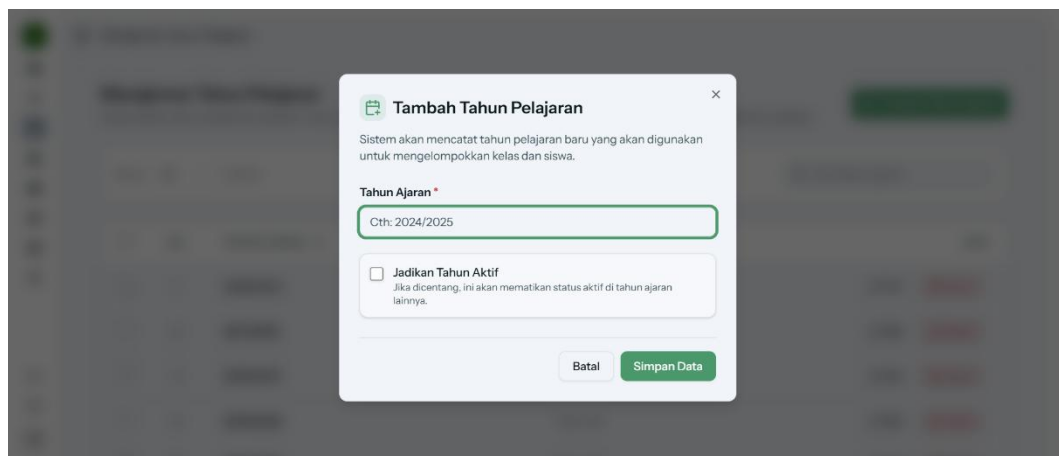
5. Tampilan halaman tahun Pelajaran



Gambar 5.49 Halaman Tahun Pelajaran

Gambar 5.49 menunjukkan hasil implementasi halaman Tahun Pelajaran pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar tahun pelajaran yang telah tersimpan di dalam sistem beserta informasi status aktif setiap tahun pelajaran. Selain itu, tersedia tombol Tambah Tahun Ajaran, fasilitas pencarian, serta tombol Edit dan Hapus sebagai media bagi Admin untuk melakukan pengelolaan data tahun pelajaran sesuai dengan kebutuhan.

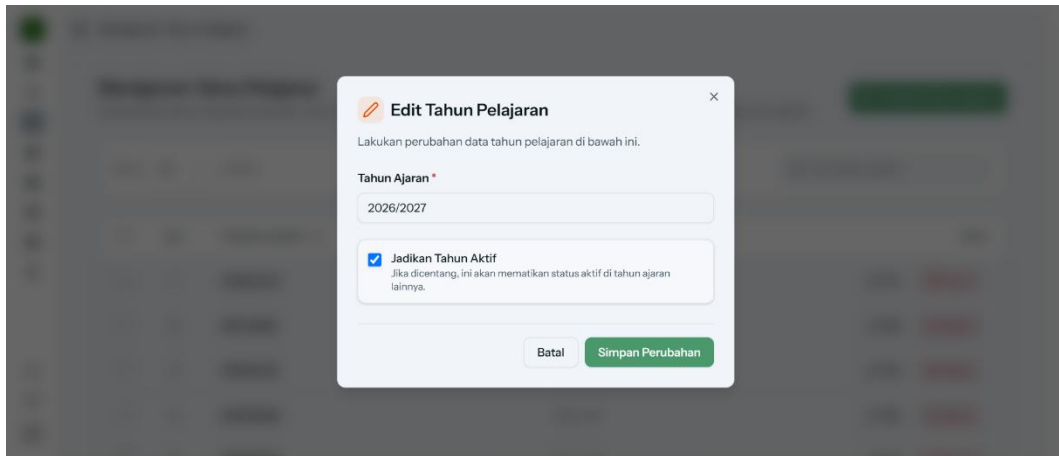
Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Admin dalam mengelola data tahun pelajaran secara lebih efektif. Selain itu, sistem juga menyediakan informasi status tahun pelajaran yang aktif sehingga dapat digunakan sebagai acuan dalam proses akademik yang berlangsung. Dengan demikian, halaman Tahun Pelajaran tidak hanya berfungsi sebagai media pengelolaan data, tetapi juga membantu menjaga konsistensi penggunaan tahun pelajaran pada seluruh fitur yang berkaitan dengan kegiatan akademik di dalam sistem.



Gambar 5.50 Halaman Tambah Tahun Pelajaran

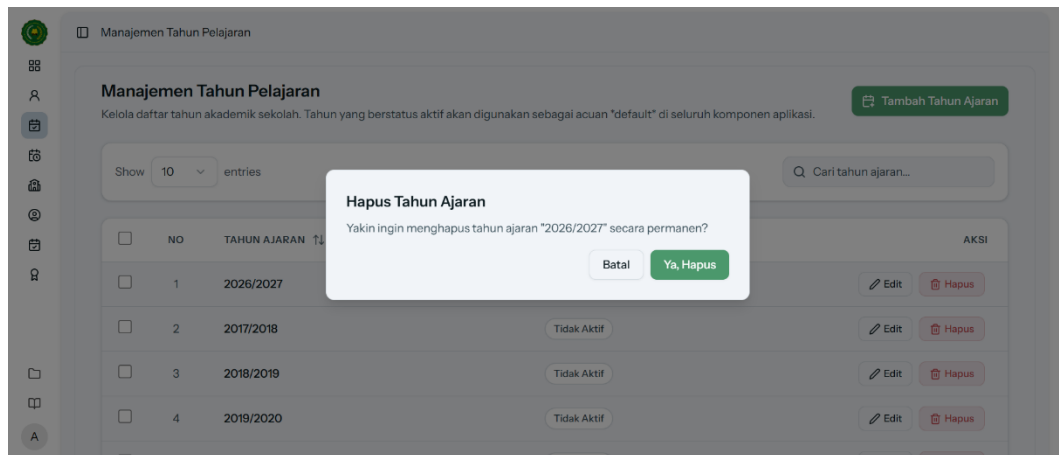
Gambar 5.50 menampilkan antarmuka modal penambahan data tahun pelajaran baru yang dirancang menggunakan komponen penampil melayang (*modal popup*) untuk mengoptimalkan efisiensi interaksi tanpa perlu melakukan navigasi halaman baru. Secara analitis, formulir ini menyediakan masukan nama rentang periode akademik dan pilihan kotak centang "*Jadikan Tahun Aktif*". Di balik layar, *backend* Laravel mengeksekusi algoritma pembaruan eksklusif (*exclusivity status logic*); apabila administrator mencentang status aktif, sistem secara otomatis akan mematikan status aktif pada seluruh tahun pelajaran lain di basis data via kueri *bulk update*. Mekanisme penanganan logika di tingkat peladen (*server-side logic*) ini sangat krusial dalam rekayasa sistem informasi akademik untuk menjamin integritas data relasional, sehingga mencegah terjadinya konflik acuan waktu *global*

state pada proses pencatatan presensi, pengelompokan kelas, dan transaksi nilai siswa.



Gambar 5.51 Halaman Edit Tahun Pelajaran

Gambar 5.51 memperlihatkan antarmuka modal pembaruan (*Update*) data tahun pelajaran yang secara otomatis menyajikan data lama (*pre-filled form*) sesuai entitas yang dipilih administrator pada tabel utama. Ditinjau dari aspek *User Experience (UX)* dan rekayasa perangkat lunak, penggunaan komponen *modal* interaktif ini memungkinkan perubahan nama ajaran maupun pergeseran status aktif dilakukan secara instan tanpa memicu pemuatan ulang halaman secara penuh (*full page reload*). Apabila kotak centang "*Jadikan Tahun Aktif*" diperbarui menjadi aktif, sistem akan kembali mengeksekusi algoritma otomatisasi penyesuaian status tunggal di basis data untuk memastikan kontinuitas referensi periode berjalan. Melalui penanganan pengkodean yang adaptif ini, administrator dapat melakukan koreksi data atau pengalihan periode ajaran baru secara fleksibel, tepat guna, serta tetap menjaga konsistensi pangkalan data utama.



Gambar 5.52 Halaman Hapus Tahun Pelajaran

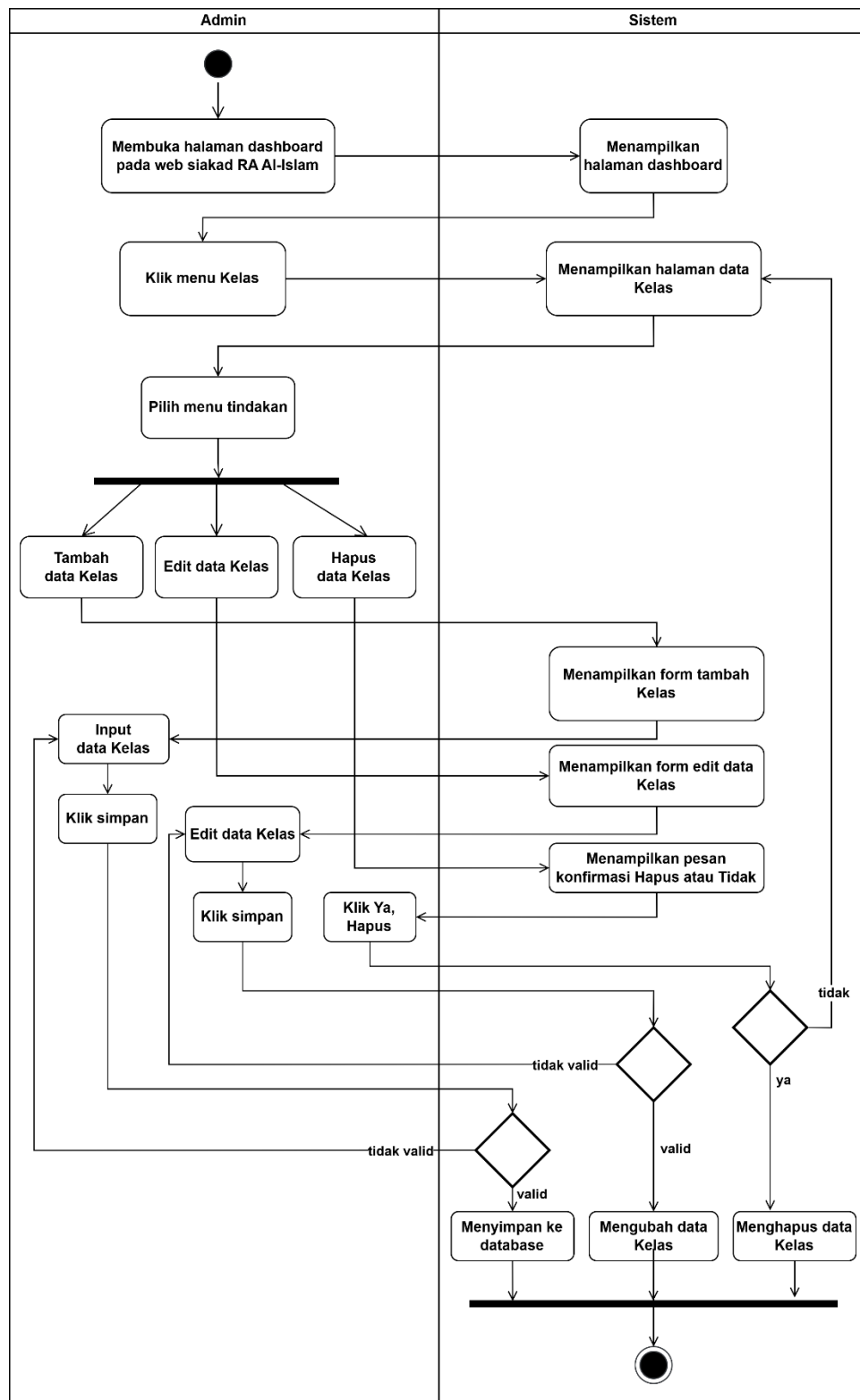
Gambar 5.52 mengilustrasikan mekanisme konfirmasi penghapusan (*Delete*) data tahun pelajaran melalui komponen *Modal Overlay Confirmation*. Secara analitis, penyediaan dialog peringatan eksplisit ini menerapkan prinsip *Fail-Safe Design* untuk meminimalkan risiko kesalahan manusia (*human error*) akibat penekanan tombol hapus yang tidak disengaja. Mengingat entitas tahun pelajaran merupakan acuan tabel referensi utama (*primary lookup table*) yang memiliki keterikatan kunci tamu (*foreign key*) terhadap tabel *student_classes*, *students*, dan *daily_attendances*, tindakan penghapusan data bersifat sangat sensitif dan berisiko memicu anomali data liar (*orphaned records*). Oleh karena itu, adanya jeda konfirmasi dengan pilihan tombol *Batal* dan *Ya, Hapus* ini memaksa pengguna melakukan verifikasi ulang secara sadar, sehingga memastikan seluruh proses manipulasi data berisiko tinggi tereksekusi secara terukur, aman, dan akuntabel.

5.4.1.5 Kelas

Fitur Kelas merupakan salah satu fitur yang dikembangkan pada *Sprint* 1 untuk mendukung pengelolaan data kelas pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Admin dapat melakukan pengelolaan data kelas yang meliputi penambahan, perubahan, penghapusan, dan melihat data kelas yang telah tersimpan pada sistem. Pengelolaan data kelas dilakukan secara terpusat sehingga informasi mengenai setiap kelas dapat tersimpan dengan lebih rapi, mudah diperbarui, dan digunakan sebagai acuan pada proses akademik berikutnya, seperti

pengelompokan siswa maupun pembagian wali kelas. Dengan adanya fitur ini, proses pengelolaan data kelas menjadi lebih efektif dibandingkan pencatatan secara manual karena seluruh data dapat diakses dan dikelola melalui sistem.

Perancangan fitur Kelas dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pengelolaan data sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas Admin selama mengelola data kelas, struktur data yang digunakan dalam penyimpanan informasi kelas, serta urutan komunikasi antara pengguna dengan sistem ketika menjalankan setiap proses *CRUD* (*Create*, *Read*, *Update*, *Delete*). Dengan adanya perancangan tersebut, proses implementasi dapat dilakukan secara lebih terarah sehingga fungsi pengelolaan data kelas dapat berjalan sesuai dengan kebutuhan sistem yang telah direncanakan.

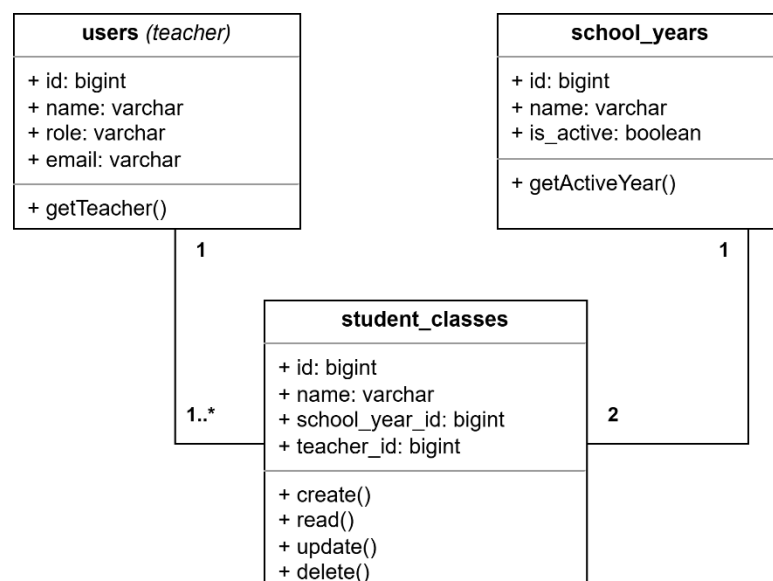
1. *Activity diagram kelas*

Gambar 5.53 Activity Diagram Kelas

Activity Diagram Kelas pada Gambar 5.53 menggambarkan alur proses pengelolaan data kelas yang dilakukan oleh Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Kelas sehingga sistem menampilkan halaman data kelas yang berisi seluruh data yang telah tersimpan pada basis data. Selanjutnya Admin dapat memilih tindakan berupa menambah data, mengubah data, maupun menghapus data kelas sesuai dengan kebutuhan pengelolaan akademik.

Apabila Admin memilih tambah atau ubah data, sistem akan menampilkan formulir yang harus diisi kemudian melakukan proses validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, apabila Admin memilih menghapus data, sistem akan terlebih dahulu menampilkan konfirmasi penghapusan untuk menghindari kesalahan tindakan. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pengelolaan data kelas dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* kelas

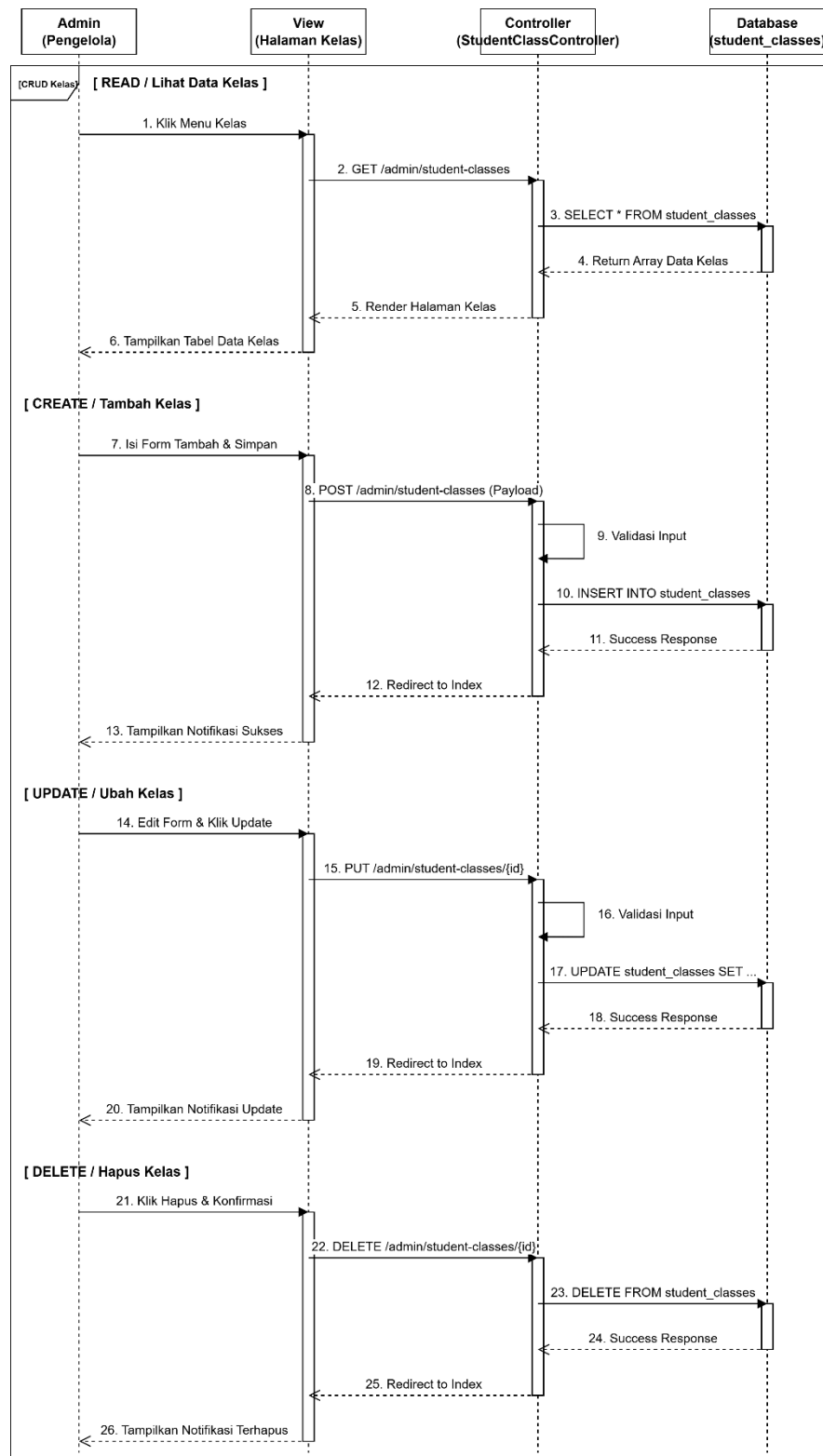


Gambar 5.54 *Class Diagram* Kelas

Class Diagram Kelas pada Gambar 5.54 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses pengelolaan data kelas pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *student_classes* yang memiliki atribut *id*, *name*, *school_year_id*, dan *teacher_id* beserta operasi *CRUD* (*create()*, *read()*, *update()*, *delete()*), dengan kelas *users (teacher)* yang memuat atribut *id*, *name*, *role*, *email* serta operasi *getTeacher()*, dan kelas *school_years* dengan atribut *id*, *name*, *is_active* serta operasi *getActiveYear()*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengelolaan rombongan belajar dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* di mana satu data guru (*users*) dapat memiliki relasi satu atau banyak (1..*) terhadap kelas setiap tahunnya (*student_classes*) sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan struktural antar-tabel melalui pemetaan kunci tamu (*foreign key*) yang dikelola oleh sistem sehingga setiap proses penambahan, perubahan, maupun penghapusan data kelas dapat dilakukan secara konsisten sesuai dengan kebutuhan integritas basis data pada sistem.

3. Sequence diagram kelas



Gambar 5.55 Sequence Diagram Kelas

Sequence Diagram Kelas pada Gambar 5.55 menggambarkan urutan komunikasi antara Admin, antarmuka sistem, *controller*, dan basis data selama proses pengelolaan data kelas berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data, menambahkan data baru, mengubah data, maupun menghapus data dikirimkan oleh Admin, kemudian diproses oleh *controller* melalui proses validasi sebelum diteruskan ke basis data hingga menghasilkan respons yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengelolaan data kelas dapat dipahami secara lebih rinci. Diagram ini menjadi acuan dalam implementasi logika sistem sehingga proses pertukaran data antara antarmuka, *controller*, dan basis data dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi kelas

```

1      public function index(Request $request)
2      {
3          $search = $request->input('search');
4          $entries = $request->input('entries', 10);
5
6          // Mengambil data kelas dengan teknik Eager Loading (memuat relasi secara
           bersamaan)
7          // untuk mencegah N+1 Query Problem saat merender nama wali kelas dan tahun
           ajaran
8          $studentClasses = StudentClass::with(['schoolYear', 'teacher'])
9              ->when($search, function ($query, $search) {
10                  $query->where('name', 'like', '%' . $search . '%');
11              })
12              ->orderBy('name', 'asc')
13              ->paginate($entries)
14              ->withQueryString();
15
16          // Mengirimkan objek Data Kelas beserta Supplementary Data (Data Tambahan)
17          // ke antarmuka klien React JS (Inertia) untuk keperluan Dropdown Form
18          return Inertia::render('admin/student-classes/index', [
19              'studentClasses' => $studentClasses,
20              // Supplementary Data: Mengambil semua tahun pelajaran dan nama guru
           (teacher)
21              'schoolYears' => SchoolYear::orderBy('name', 'desc')->get(['id',
           'name', 'is_active']),
22              'teachers' => User::where('role', 'teacher')->orderBy('name')-
           >get(['id', 'name']),
23              'search' => $search,
24          ]);
25      }

```

Gambar 5.56 Code Backend Kelas

Potongan kode pada Gambar 5.56 menunjukkan implementasi logika *Backend* untuk fitur Pengelolaan Kelas pada kontroler *StudentClassController*. Secara arsitektural basis data, entitas Kelas memiliki ketergantungan relasional (*relational dependency*) terhadap dua entitas referensi utama, yaitu Tahun Pelajaran dan tabel Pengguna (khusus untuk profil Wali Kelas). Oleh karena itu, pada fungsi *index()*, sistem mengimplementasikan teknik Eager Loading bertingkat (direpresentasikan dengan *with(['schoolYear', 'teacher'])*) guna memuat data-data relasional tersebut secara paralel. Selain itu, *Backend* juga menginisialisasi *Supplementary Data* (data tambahan pelengkap) berupa pemanggilan senarai (*array*) guru dan senarai tahun pelajaran untuk dikirimkan bersamaan ke antarmuka klien. Data tambahan ini ditujukan untuk memfasilitasi integrasi elemen *dropdown selection* pada komponen *Frontend*.

```

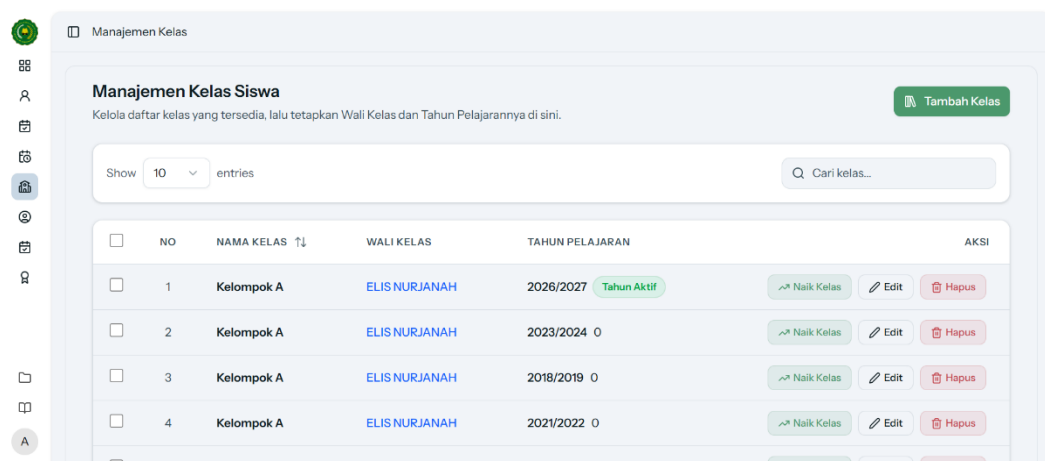
1  export default function StudentClassIndex({ studentClasses, schoolYears, teachers }:
  Props) {
2    return (
3      <AppLayout breadcrumbs={breadcrumbs}>
4        <Head title="Manajemen Kelas" />
5
6        <div className="p-4 sm:p-6">
7
8          {/* Area Header Manajemen Kelas */}
9          <div className="mb-6 flex justify-between">
10            <div>
11              <h2 className="text-xl font-bold">Manajemen Kelas Siswa</h2>
12            </div>
13            <div className="flex shrink-0">
14              {/*
15               Mengoper Supplementary Data (schoolYears dan teachers)
16               ke sub-komponen Modal melalui pola arsitektur Prop
17               Drilling
18               */}
19              <AddStudentClassModal schoolYears={schoolYears} teachers=
20                {teachers} />
21            </div>
22          </div>
23
24          <div className="overflow-x-auto bg-card rounded-xl border">
25            <table className="w-full text-left text-sm">
26              <thead className="bg-muted text-muted-foreground uppercase
27                text-xs">
28                <tr>
29                  <th className="px-6 py-4">NAMA KELAS</th>
30                  <th className="px-6 py-4">WALI KELAS</th>
31                  <th className="px-6 py-4">TAHUN PELAJARAN</th>
32                  <th className="px-6 py-4 text-right">AKSI</th>
33                </tr>
34              </thead>
35              <tbody className="divide-y divide-border">
36                {studentClasses.data.map((item) => (
37                  <tr key={item.id} className="hover:bg-muted/40">
38                    <td className="px-6 py-4 font-bold">{item.name}
39
40                    <td className="px-6 py-4">
41                      {/* Render nama Wali Kelas dari hasil Eager
42                       Loading */}
43                      {item.teacher ? (
44                        <span className="font-medium text-blue-
45                          600">
46                          {item.teacher.name}
47                        </span>
48                      ) : (
49                        <span className="text-muted-foreground
50                          italic">Belum Diatur</span>
51                      )}
52                    </td>
53
54                    <td className="px-6 py-4">
55                      {/* Render Tahun Ajaran dan Indikator LED
56                       Aktif (hijau) */}
57                      {item.school_year ? (
58                        <span className="inline-flex items-center
59                          gap-1.5 rounded-md border bg-muted/50 px-2.5 py-1 text-xs font-semibold">
60                          {item.school_year.name}
61                          {item.school_year.is_active && (
62                            <span className="h-1.5 w-1.5
63                              rounded-full bg-green-500"></span>
64                          )}
65                        </span>
66                      ) : '-'}
67                    </td>
68
69                    <td className="px-6 py-4 text-right">
70                      <EditStudentClassModal studentClass={item}
71                        schoolYears={schoolYears} teachers={teachers} />
72                    </td>
73                  </tr>
74                </tbody>
75              </table>
76            </div>
77          </div>
78        </AppLayout>
79      );
80    }
81  }

```

Gambar 5.57 Code Frontend Kelas

Potongan kode pada Gambar 5.57 menunjukkan implementasi antarmuka *Frontend* untuk Pengelolaan Data Kelas menggunakan pustaka *React JS*. Berdasarkan objek data relasional yang telah disediakan dari sisi *Backend*, antarmuka tabel dirancang untuk dapat mengekstraksi dan merender secara langsung atribut nama Wali Kelas dan Tahun Pelajaran (melalui properti objek *item.teacher.name* dan *item.school_year.name*). Lebih lanjut, *Supplementary Data* yang diterima dari *Backend* (variabel *schoolYears* dan *teachers*) dioperasikan menggunakan teknik *Prop Drilling*. Variabel tersebut dilempar secara hierarkis (*pass-through*) ke dalam sub-komponen eksternal `<AddStudentClassModal />` dan `<EditStudentClassModal />`. Metode ini menjamin ketersediaan referensi data yang akurat dan *real-time* saat administrator mengoperasikan formulir penambahan atau pengubahan kelas.

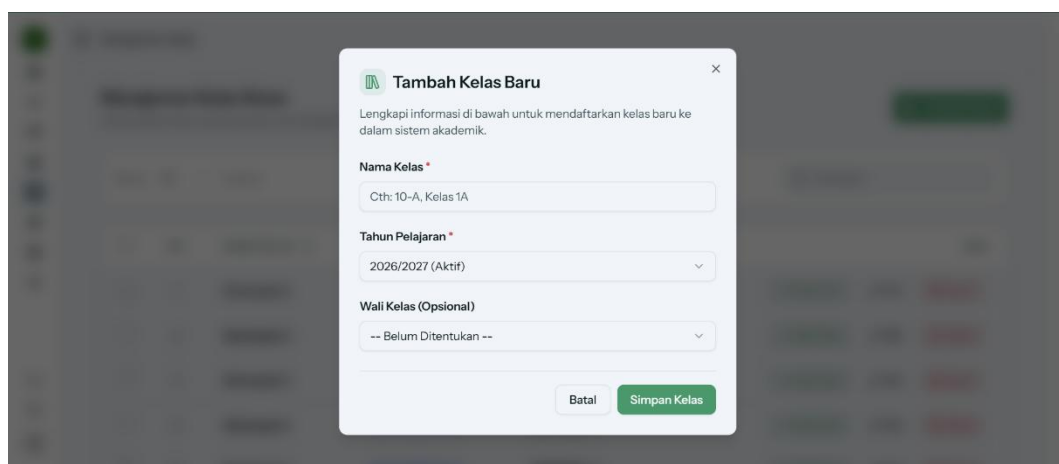
5. Tampilan halaman kelas



Gambar 5.58 Halaman Kelas

Gambar 5.58 menunjukkan hasil implementasi halaman Kelas pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar kelas yang telah tersimpan di dalam sistem beserta informasi wali kelas dan tahun pelajaran aktif. Selain itu, tersedia tombol Tambah Kelas, fasilitas pencarian, serta tombol Edit dan Hapus sebagai media bagi Admin untuk melakukan pengelolaan data kelas sesuai dengan kebutuhan.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Admin dalam mengelola data kelas secara lebih efektif. Selain itu, sistem juga menyediakan informasi wali kelas dan status aktif tahun pelajaran sehingga dapat digunakan sebagai acuan dalam proses akademik yang berlangsung. Dengan demikian, halaman Kelas tidak hanya berfungsi sebagai media pengelolaan data, tetapi juga membantu menjaga konsistensi penggunaan kelas pada seluruh fitur yang berkaitan dengan kegiatan akademik di dalam sistem.



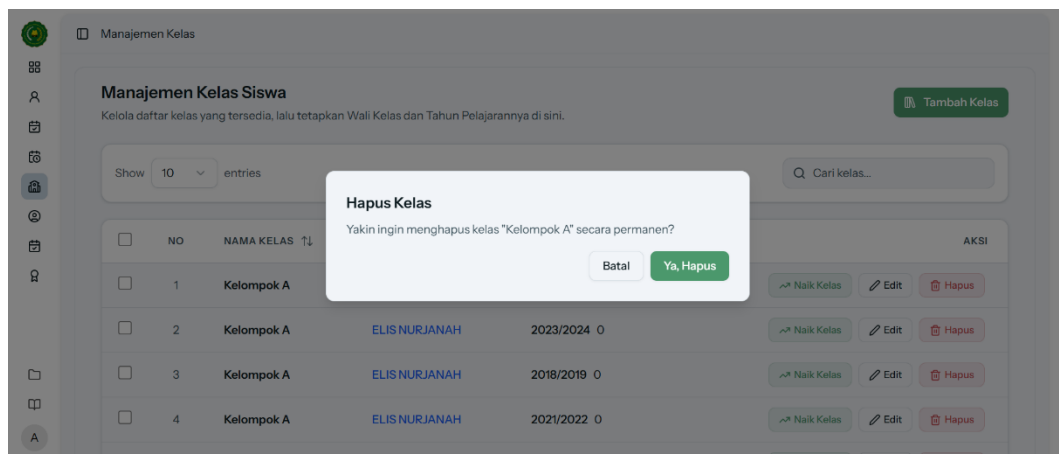
Gambar 5.59 Halaman Tambah Kelas

Gambar 5.59 menampilkan antarmuka modal penambahan data kelas baru yang dirancang menggunakan komponen penampil melayang (*modal popup*) berbasis *Single Page Application* (SPA) untuk meningkatkan efisiensi navigasi tanpa memicu pemuatan ulang halaman secara penuh. Secara analitis, formulir ini meminta masukan nama kelas, pemetaan tahun pelajaran wajib, serta alokasi wali kelas yang bersifat opsional (*nullable*). Di balik layar, *backend* Laravel memanfaatkan teknik *supplementary data loading* untuk menyajikan daftar pilihan tahun ajaran aktif dan tenaga pendidik secara dinamis ke dalam elemen *dropdown selection*. Proses penyimpanan data ini mengeksekusi validasi kueri untuk memastikan keterikatan kunci tamu (*foreign key*) antara tabel *student_classes*, *school_years*, dan *users* terjalin secara presisi. Pendekatan perancangan ini tidak hanya memudahkan alokasi rombongan belajar bagi administrator, tetapi juga

menjamin konsistensi arsitektur relasional pangkalan data utama dalam mendukung pengelolaan kegiatan akademik sekolah.

Gambar 5.60 Halaman Edit Kelas

Gambar 5.60 memperlihatkan antarmuka modal pembaruan (*Update*) data kelas yang secara otomatis menyajikan data lama (*pre-filled form*) berdasarkan entitas kelas yang dipilih administrator pada tabel utama. Ditinjau dari aspek rekayasa perangkat lunak, penggunaan komponen *modal* interaktif ini memfasilitasi penyesuaian nama kelas, pengalihan periode tahun pelajaran, maupun penetapan ulang wali kelas secara *real-time* tanpa mengganggu kestabilan antarmuka *client*. Di sisi *backend*, *controller* memproses pembaruan masukan melalui penguncian validasi entitas untuk memastikan bahwa perubahan wali kelas (*teacher_id*) hanya dapat ditautkan kepada akun pengguna yang memiliki peran (*role*) sebagai guru. Melalui alur pemrosesan data yang terstruktur ini, pemeliharaan atribut rombongan belajar dapat dieksekusi secara adaptif dan tepat guna, sekaligus menjaga integritas relasional terhadap data siswa yang terdaftar di dalam kelas tersebut.



Gambar 5.61 Halaman Hapus Kelas

Gambar 5.46c mengilustrasikan mekanisme konfirmasi penghapusan (*Delete*) data kelas menggunakan komponen *Modal Overlay Confirmation*. Secara analitis, penerapan dialog peringatan eksplisit ini mengadopsi prinsip *Fail-Safe Design* untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) akibat penekanan tombol aksi yang tidak disengaja. Mengingat entitas kelas merupakan induk relasional yang menampung keterikatan data peserta didik (*students*) serta rekaman presensi harian, tindakan penghapusan entitas ini bersifat sangat krusial dan berpotensi memicu anomali keterikatan data (*orphaned records*) apabila tidak dikontrol secara ketat. Oleh karena itu, ketersediaan jeda konfirmasi dengan pilihan opsi *Batal* dan *Ya, Hapus* ini memaksa administrator melakukan verifikasi ulang secara sadar, sehingga seluruh proses manipulasi pangkalan data yang berisiko tinggi dapat berjalan secara aman, dan terukur.

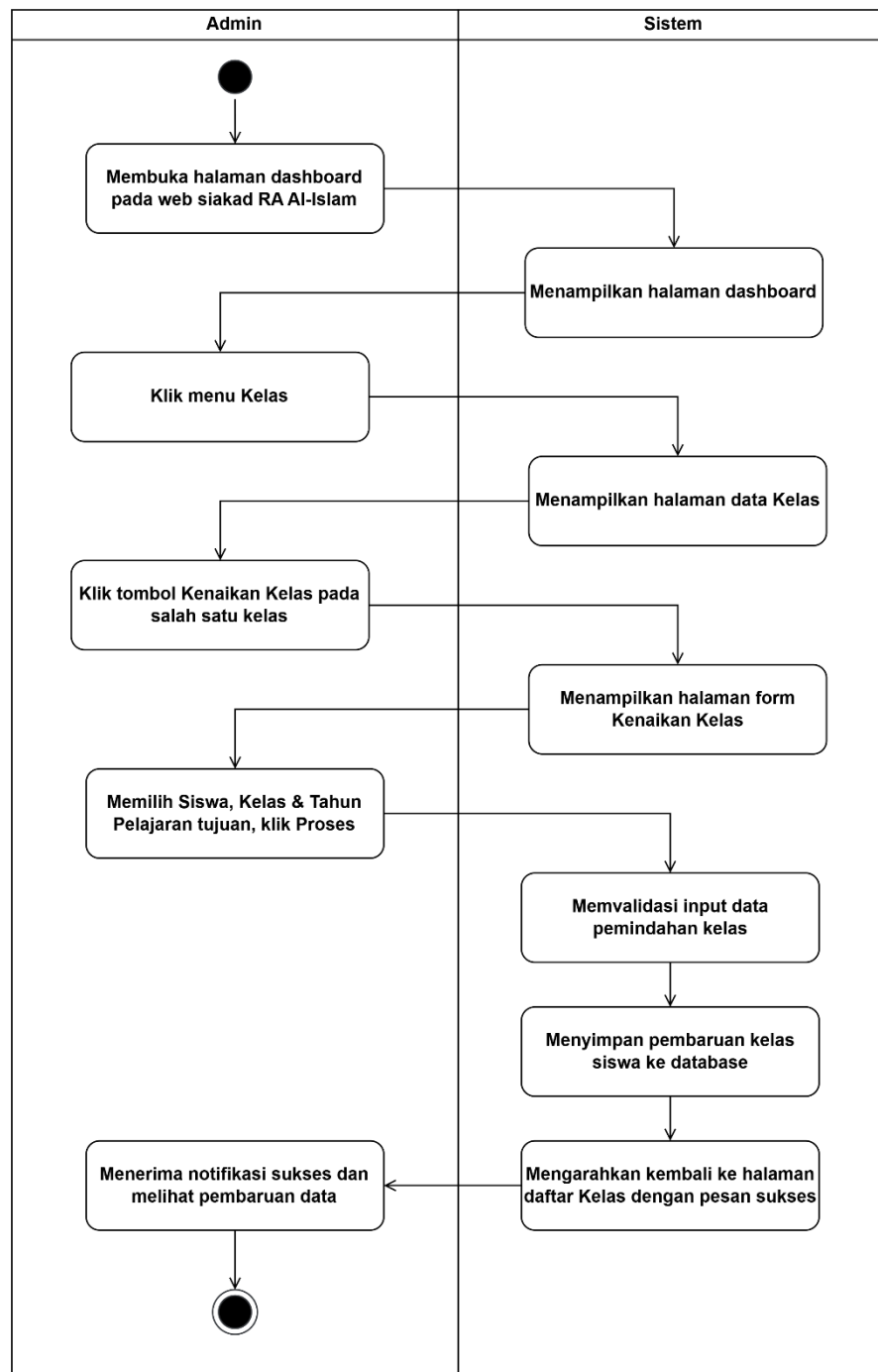
5.4.1.6 Kelas (kenaikan kelas)

Fitur Kenaikan Kelas merupakan salah satu fitur yang dikembangkan pada *Sprint 1* untuk mendukung efisiensi pembaruan tingkat akademik siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Admin dapat melakukan proses promosi jenjang kelas secara masal atau kolektif, yaitu memindahkan seluruh siswa dalam satu rombongan belajar dari Kelompok A ke Kelompok B, maupun dari Kelompok B ke status Lulus hanya dengan satu kali

tindakan eksekusi per kelas. Pengelolaan kenaikan kelas dilakukan secara terotomatisasi sehingga proses transisi siswa antar-tahun pelajaran dapat berjalan secara lebih cepat, terstruktur, dan terhindar dari potensi kesalahan pembaruan data individual. Dengan adanya fitur ini, proses administrasi kelulusan dan pemindahan jenjang peserta didik menjadi jauh lebih efektif dibandingkan prosedur manual yang mengharuskan pembaruan status siswa satu per satu.

Perancangan fitur Kenaikan Kelas dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pemrosesan data secara masal sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas Admin saat memicu instruksi promosi kelas, struktur data serta keterkaitan relasional antar-tabel yang terlibat dalam transaksi pembaruan status, serta urutan komunikasi antara pengguna dengan sistem ketika menjalankan eksekusi pembaruan data secara kolektif (*bulk update*). Dengan adanya perancangan tersebut, proses implementasi dapat dilakukan secara lebih terarah sehingga fungsi kenaikan kelas masal dapat berjalan secara aman, stabil, dan sesuai dengan kebutuhan operasional sekolah yang telah direncanakan.

1. *Activity diagram kelas (kenaikan kelas)*



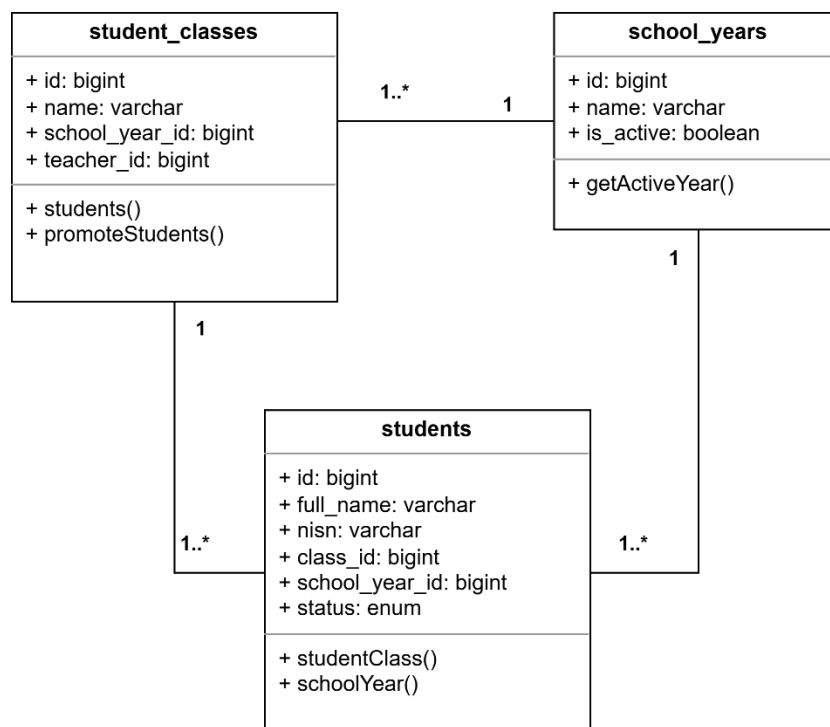
Gambar 5.62 *Activity Diagram Kelas (Kenaikan Kelas)*

Activity Diagram Kenaikan Kelas pada Gambar 5.62 menggambarkan alur aktivitas operasional yang dilakukan oleh Admin dalam mengelola proses promosi jenjang

peserta didik pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka *Dashboard*, memilih menu Kelas, lalu menekan tombol aksi Kenaikan Kelas pada salah satu rombongan belajar. Selanjutnya, sistem merespons dengan menampilkan antarmuka formulir Kenaikan Kelas, di mana Admin dapat memilih siswa yang akan dipromosikan serta menentukan kelas dan tahun pelajaran tujuan sebelum menekan tombol proses.

Apabila Admin mengirimkan masukan data pemindahan kelas, sistem akan melakukan validasi masukan untuk memastikan kelayakan relasi antara kelas asal dan kelas tujuan. Setelah data dinyatakan valid, sistem menyimpan pembaruan kelas siswa ke dalam basis data secara serentak, lalu mengarahkan Admin kembali ke halaman daftar Kelas disertai notifikasi keberhasilan. Diagram ini menjadi acuan dalam proses implementasi agar seluruh mekanisme promosi siswa secara masal dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan operasional sekolah.

2. *Class diagram* kelas (kenaikan kelas)

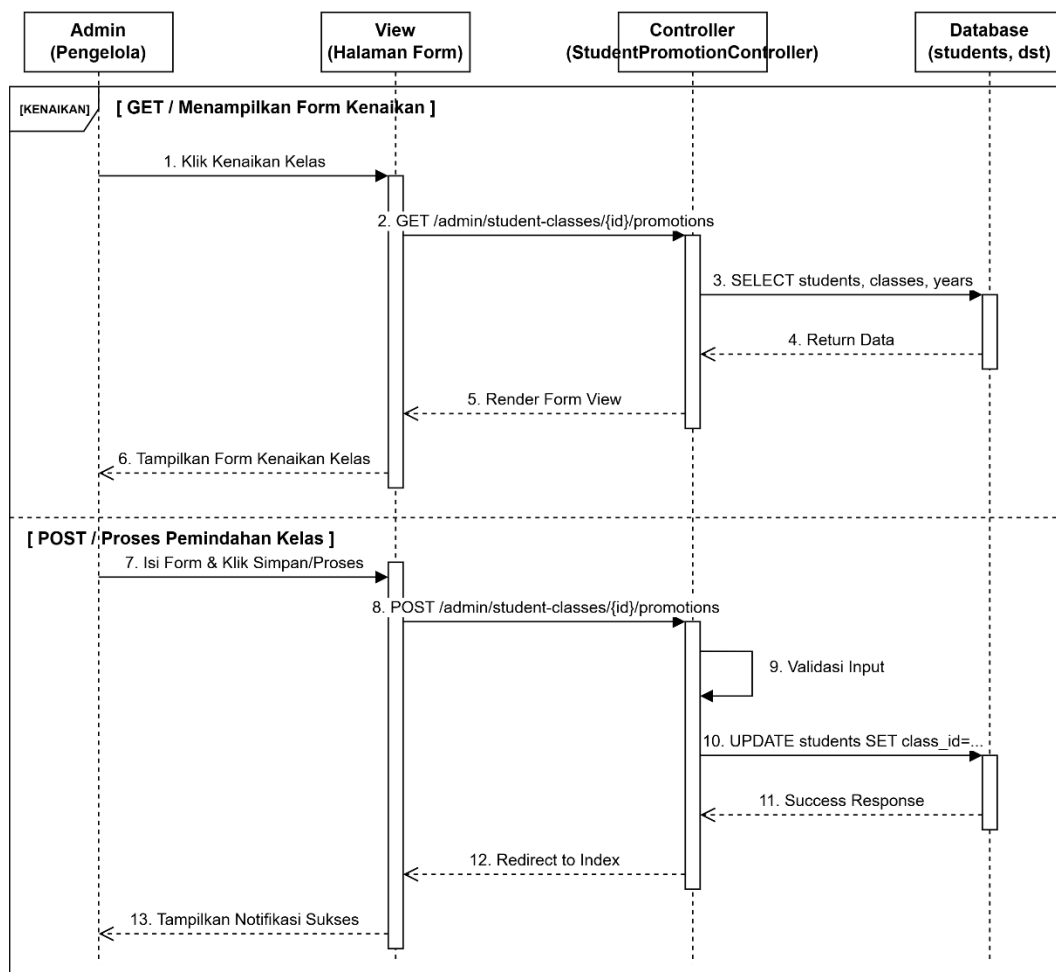


Gambar 5.63 *Class Diagram* Kelas (Kenaikan Kelas)

Class Diagram Kenaikan Kelas pada Gambar 5.63 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses promosi jenjang peserta didik pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *student_classes* yang memuat atribut *id*, *name*, *school_year_id*, *teacher_id* serta fungsi *promoteStudents()*, kelas *students* dengan atribut *id*, *full_name*, *nisn*, *class_id*, *school_year_id*, *status*, dan kelas *school_years* yang memuat atribut *id*, *name*, *is_active* serta fungsi *getActiveYear()*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengorganisasian mutasi kelas dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* di mana entitas *student_classes* dan *school_years* masing-masing memiliki relasi satu ke banyak (1 ke 1..*) terhadap tabel *students*. Selain itu, kehadiran metode *promoteStudents()* bertindak sebagai enkapsulasi logika bisnis untuk memperbarui kunci tamu (*class_id* dan *school_year_id*) secara masal, sehingga memastikan setiap proses promosi kelas dapat dijalankan secara konsisten sesuai aturan integritas basis data.

3. Sequence diagram kelas (kenaikan kelas)



Gambar 5.64 Sequence Diagram Kelas (Kenaikan Kelas)

Sequence Diagram Kenaikan Kelas pada Gambar 5.64 menggambarkan urutan komunikasi antara Admin, antarmuka sistem (*View*), *controller* (*StudentPromotionController*), dan basis data selama proses pemindahan jenjang siswa berlangsung. Diagram ini menunjukkan bagaimana permintaan awal dikirimkan oleh Admin melalui antarmuka dengan metode HTTP GET `GET /admin/student-classes/{id}/promotions`, yang kemudian diproses oleh *controller* dengan melakukan eksekusi kueri SELECT ke basis data untuk mengambil data siswa, kelas, dan tahun ajaran sebelum merender tampilan form.

Pada tahap eksekusi pemindahan data, Admin mengirimkan formulir melalui metode HTTP POST `POST /admin/student-classes/{id}/promotions`. *Controller*

terlebih dahulu menjalankan fungsi validasi masukan di sisi *server*, kemudian meneruskan instruksi pembaruan kueri UPDATE students SET class_id=... ke basis data. Setelah basis data mengembalikan respons sukses, *controller* melakukan pengalihan halaman (*redirect*) kembali ke halaman utama disertai pesan notifikasi keberhasilan, sehingga menjamin proses pertukaran data berjalan secara terstruktur, aman, dan efisien.

4. Implementasi kelas (kenaikan kelas)

```

1  public function store(Request $request, StudentClass $studentClass)
2  {
3      // Memvalidasi data payload yang dikirim dari antarmuka React JS (Inertia)
4      $validated = $request->validate([
5          'student_ids'          => ['required', 'array', 'min:1'],
6          'action'               => ['required', 'in:promote,graduate'],
7          'destination_school_year_id' => ['required_if:action,promote',
8          'nullable'],
9          'destination_class_id'  => ['required_if:action,promote',
10         'nullable'],
11         ]);
12
13     try {
14         // Memulai transaksi Database untuk mencegah data korup jika terjadi
15         // kegagalan sistem
16         DB::beginTransaction();
17
18         // Mengecek apakah aksi yang dipilih adalah Kelulusan atau Kenaikan Kelas
19         if ($validated['action'] === 'graduate') {
20             // Proses Kelulusan: Mengubah status kumpulan siswa menjadi 'lulus'
21             Student::whereIn('id', $validated['student_ids'])->update([
22                 'status' => 'lulus'
23             ]);
24         } else {
25             // Proses Kenaikan Kelas: Memperbarui tahun ajaran dan kelas tujuan
26             // kumpulan siswa
27             Student::whereIn('id', $validated['student_ids'])->update([
28                 'school_year_id' => $validated['destination_school_year_id'],
29                 'class_id'       => $validated['destination_class_id'],
30             ]);
31         }
32
33         // Menyimpan seluruh perubahan di atas ke dalam Database secara permanen
34         DB::commit();
35
36         // Mengarahkan kembali (Redirect) ke halaman Daftar Kelas beserta pesan
37         // sukses
38         return Redirect::route('admin.student-classes.index')
39             ->with('success', 'Data siswa terpilih berhasil diproses.');
```

```

35     } catch (\Exception $e) {
36         // Membatalkan semua perubahan jika terjadi masalah (Rollback)
37         DB::rollBack();
38         return back()->with('error', 'Terjadi kesalahan sistem: ' . $e-
39             >getMessage());
40     }

```

Gambar 5.65 Code Backend Kelas (Kenaikan Kelas)

Potongan kode pada Gambar 5.65 menunjukkan implementasi logika *backend* untuk pemrosesan Kenaikan Kelas dan Kelulusan masal pada *controller* menggunakan *framework* Laravel. Logika ini diawali dengan validasi masukan *payload* (`student_ids`, `action`, `destination_school_year_id`, `destination_class_id`) untuk memastikan integritas skenario tindakan. Secara analitis, keunggulan utama dari penulisan kode ini terletak pada penerapan *Database Transaction* (`DB::beginTransaction()`, `DB::commit()`, dan `DB::rollBack()`). Penggunaan transaksi basis data ini menjamin prinsip *Atomicity* (ACID), di mana eksekusi pembaruan masal (`Student::whereIn()->update()`) terhadap puluhan data siswa akan diproses secara utuh; apabila terjadi kesalahan sistem di tengah jalan, seluruh perubahan akan dibatalkan (*rollback*) secara otomatis. Pendekatan rekayasa ini sangat krusial untuk mencegah terjadinya korupsi data (*data corruption*) atau pembaruan sebagian (*partial update*) yang berpotensi menimbulkan ketidakcocokan status akademik antar-tabel di pangkalan data.

```

1      const handleConfirm = () => {
2          setIsConfirmOpen(false);
3          setProcessing(true);
4
5          // Mengirimkan objek Data Kenaikan Kelas (Payload) ke Backend melalui route
6          POST router.post(
7              route('admin.student-classes.promotions.store', sourceClass.id),
8              {
9                  student_ids: selectedStudents,
10                 action: action,
11                 destination_school_year_id: destinationSchoolYearId,
12                 destination_class_id: destinationClassId,
13             },
14             {
15                 // Menonaktifkan status loading ketika proses eksekusi selesai
16                 onFinish: () => setProcessing(false),
17             }
18         );
19     };
20
21     return (
22         /* ... Kode Layout & Header diringkas ... */
23
24         <div className="lg:col-span-2 rounded-xl border border-border bg-card shadow-sm p-4">
25             <h3 className="font-semibold text-lg">Daftar Siswa</h3>
26
27             /* Merender data siswa dari Controller ke dalam bentuk Tabel */
28             <table className="w-full text-left text-sm mt-4">
29                 <thead className="bg-muted text-muted-foreground uppercase text-xs">
30                     <tr>
31                         <th className="px-5 py-3">
32                             <input
33                                 type="checkbox"
34                                 onChange={toggleSelectAll}
35                                 checked={selectedStudents.length === students.length}
36                             />
37                         </th>
38                         <th className="px-6 py-3">NAMA SISWA</th>
39                         <th className="px-6 py-3">NISN</th>
40                     </tr>
41                 </thead>
42                 <tbody className="divide-y divide-border">
43                     /* Melakukan perulangan (Mapping) data siswa untuk ditampilkan
44                     per baris */
45                     {students.map((student) => (
46                         <tr key={student.id} className="hover:bg-muted/40">
47                             <td className="px-5 py-2">
48                                 <input
49                                     type="checkbox"
50                                     value={student.id}
51                                     onChange={() => toggleSelection(student.id)}
52                                     checked={selectedStudents.includes(student.id)}
53                                 />
54                             <td className="px-6 py-2 font-medium">{student.full_name}
55                             <td className="px-6 py-2 text-muted-foreground">
56                                 {student.nisn || '-'}</td>
57                         </tr>
58                     ))}
59                 </tbody>
60             </table>
61
62             /* ... Sisa form opsi disembunyikan untuk keperluan snippet ... */
63         </div>
64     );

```

Gambar 5.66 Code Frontend Kelas (Kenaikan Kelas)

Potongan kode pada Gambar 5.66 memperlihatkan implementasi antarmuka *frontend* untuk penanganan interaksi Kenaikan Kelas menggunakan React JS dan Inertia.js. Pada komponen ini, sistem mengelola *state* pilihan siswa (*selectedStudents*) secara reaktif, yang memungkinkan eksekusi centang masal (*select-all*) melalui fungsi *toggleSelectAll* maupun pemilihan individual per baris melalui *toggleSelection*. Saat konfirmasi dipicu (*handleConfirm*), fungsi mengirimkan berkas *payload* JSON secara asinkron ke *route backend* memanfaatkan modul *router.post()*. Ditinjau dari sudut pandang *User Experience (UX)* dan kinerja aplikasi, pengumpulan *state* array ID siswa di sisi klien (*client-side*) sebelum dikirimkan sekaligus ke peladen ini secara signifikan mengurangi beban lalu lintas jaringan (*network overhead*). Teknik *Client-Side State Management* dan *Inertia Dispatching* ini memastikan proses seleksi peserta didik berjalan secara responsif, interaktif, serta bebas dari pemuatan ulang halaman secara penuh (*full page reload*).

5. Tampilan halaman kelas (kenaikan kelas)

Kenaikan Kelas & Kelulusan
Atur kenaikan kelas atau kelulusan siswa dari kelas **Kelompok A**.

Daftar Siswa
Kelas Asal: **Kelompok A (2026/2027)** 23 Terpilih

☒	NO	NAMA SISWA	NIS/NISN
☒	1	ADNAN ELFATHAN GANI	/ -
☒	2	BARA ATHARRAZKA NUGRAHA	/ -
☒	3	DADIN SAEPULOH	/ -
☒	4	DAFFA ARYA NUGRAHA	/ -
☒	5	DANIYAL RAMADHAN	/ -
☒	6	DEVAN DIKA PRATAMA	/ -
☒	7	DIAS AGARA	/ -

Pengaturan Tujuan

Aksi:

Tahun Pelajaran Tujuan:

Kelas Tujuan:

Gambar 5.67 Halaman Kelas (Kenaikan Kelas)

Gambar 5.67 menunjukkan hasil implementasi halaman Kenaikan Kelas & Kelulusan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan antarmuka berbasis dua panel (*split-card layout*) yang memisahkan tabel daftar siswa dari kelas asal di sisi kiri dan panel pengaturan tujuan di sisi kanan. Panel daftar siswa dilengkapi dengan penanda indikator

centang masal (*select-all checkbox*) serta rekapitulasi jumlah siswa terpilih, sedangkan panel sebelah kanan memuat tombol navigasi mode ("Naik Kelas" atau "Lulus"), *dropdown* pilihan Tahun Pelajaran Tujuan, *dropdown* Kelas Tujuan, serta tombol eksekusi utama berlabel "Proses Eksekusi".

Tampilan dirancang dengan antarmuka yang bersih dan terstruktur sehingga memudahkan Admin dalam mengelola pemindahan jenjang siswa secara kolektif tanpa perlu melakukan pembaruan satu per satu. Penggabungan fungsi seleksi masal dan konfigurator tujuan dalam satu layar memberikan pengalaman penggunaan (*User Experience*) yang intuitif dan cepat. Dengan demikian, halaman Kenaikan Kelas & Kelulusan tidak hanya meningkatkan efisiensi operasional administrasi sekolah pada setiap pergantian tahun ajaran, tetapi juga meminimalisir potensi kesalahan manusia (*human error*) dalam pembaruan status peserta didik.

5.4.1.7 Kalender pendidikan (melihat)

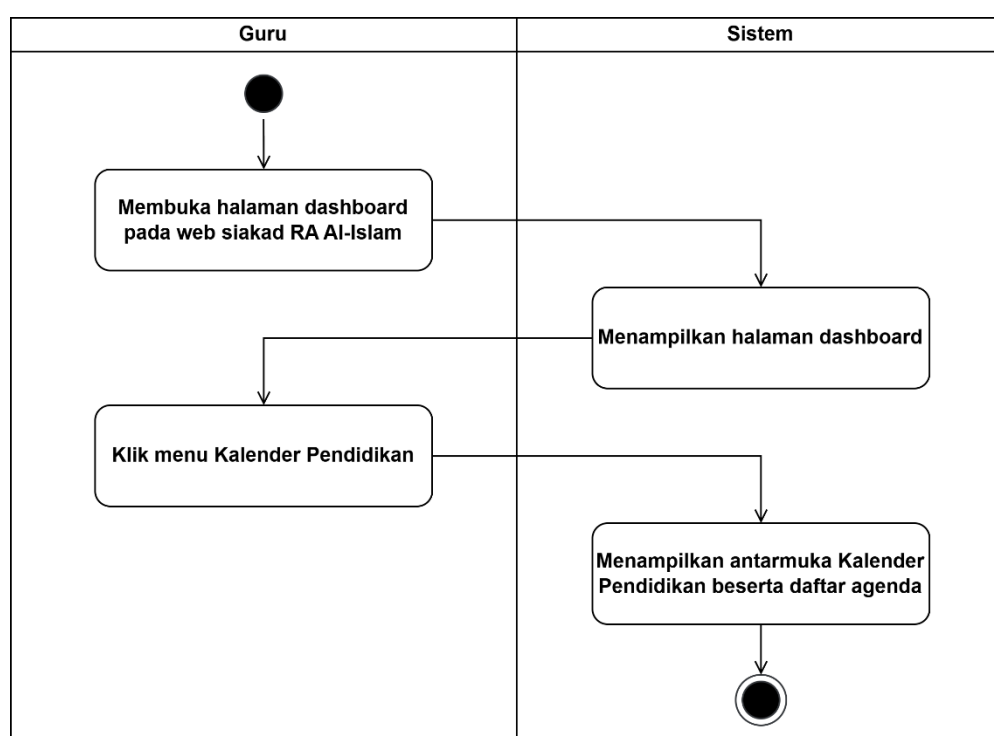
Fitur Kalender Pendidikan (Melihat) merupakan salah satu fitur yang dikembangkan pada *Sprint* 1 untuk mendukung penyajian jadwal kegiatan akademik pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Guru dan Orang Tua Siswa dapat melakukan peninjauan jadwal penting yang meliputi agenda kegiatan harian, hari libur sekolah, serta jadwal pelaksanaan evaluasi belajar yang telah tersimpan pada sistem. Penyajian data kalender dilakukan secara terstruktur sehingga informasi mengenai setiap agenda dapat disajikan dengan lebih rapi, mudah dipantau, dan digunakan sebagai acuan dalam menyelaraskan aktivitas belajar baik di lingkungan sekolah maupun di rumah. Dengan adanya fitur ini, proses penyampaian agenda akademik menjadi lebih efektif dibandingkan buku pengumuman manual karena seluruh informasi dapat diakses secara langsung dan real-time melalui sistem.

Perancangan fitur Kalender Pendidikan (Melihat) dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pemuatan data sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas pengguna saat mengakses informasi

jadwal, struktur data yang digunakan dalam pemetaan agenda tahun ajaran, serta urutan komunikasi antara komponen antarmuka dengan basis data ketika menjalankan proses pembacaan data (*read data*). Dengan adanya perancangan tersebut, proses implementasi dapat dilakukan secara lebih terarah sehingga fungsi melihat kalender pendidikan dapat berjalan sesuai dengan kebutuhan sistem serta kenyamanan pengguna yang telah direncanakan.

1. Kalender pendidikan guru

a. *Activity diagram*



Gambar 5.68 *Activity Diagram* Melihat Kalender Pendidikan Guru

Activity Diagram Kalender Pendidikan (Melihat) pada Gambar 5.68 menggambarkan alur proses melihat jadwal atau agenda akademik yang dilakukan oleh Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Guru membuka Dashboard kemudian memilih menu Kalender Pendidikan sehingga sistem menampilkan halaman data kalender pendidikan yang berisi seluruh data agenda yang telah tersimpan pada basis data. Selanjutnya Guru dapat melihat visualisasi kalender beserta daftar agenda kegiatan akademik secara

langsung sesuai dengan kebutuhan pemantauan waktu pembelajaran di lingkungan sekolah.

Apabila Guru mengakses menu Kalender Pendidikan, sistem secara otomatis akan menarik seluruh rekaman data jadwal yang aktif dari database untuk diproyeksikan ke dalam antarmuka kalender bulanan. Proses pemuatan informasi agenda ini dirancang secara terstruktur dan efisien agar visualisasi jadwal kegiatan penting sekolah dapat langsung dibaca dengan jelas. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme menampilkan informasi kalender pendidikan dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

b. *Class diagram*

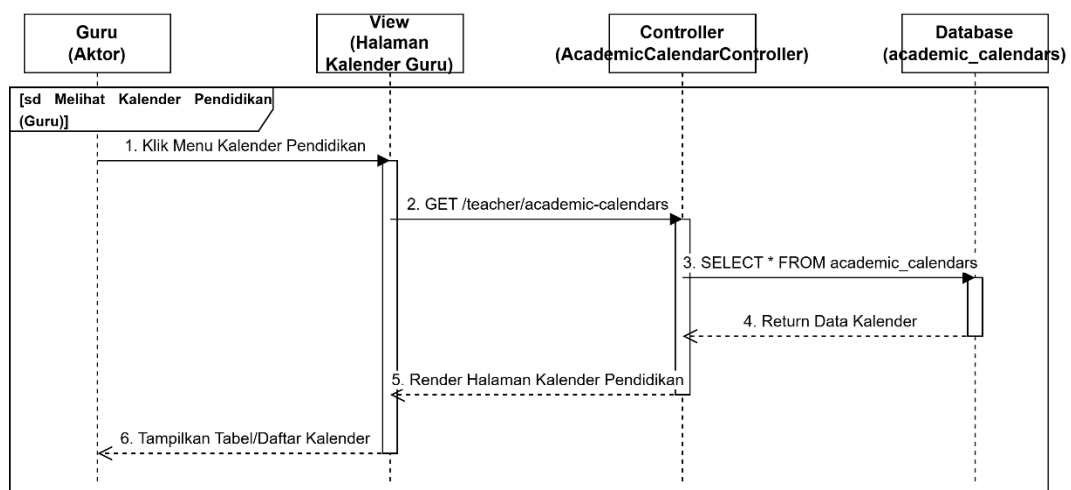


Gambar 5.69 *Class Diagram* Kalender Pendidikan

Class Diagram Kalender Pendidikan (Melihat) pada Gambar 5.69 digunakan untuk menggambarkan struktur kelas yang berperan dalam proses pengelolaan data kalender pendidikan. Diagram ini menunjukkan atribut yang dimiliki oleh kelas *academic_calendars*, yaitu *id*, *title*, *start_date*, *end_date*, *type*, dan *description*, beserta operasi yang digunakan untuk melakukan proses *Create*, *Read*, *Update*, dan *Delete* (CRUD) terhadap data kalender pendidikan.

Melalui *Class Diagram*, struktur data yang digunakan dalam proses pengelolaan agenda kegiatan akademik dapat dirancang secara lebih sistematis sehingga implementasi pada tahap pengkodean komponen basis data menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan karakteristik properti dari setiap objek jadwal seperti penentuan tipe agenda dan deskripsi kegiatan sehingga setiap proses penambahan, pembaruan, maupun pembacaan data kalender oleh sistem dapat berjalan secara konsisten dan aman.

c. *Sequence diagram*

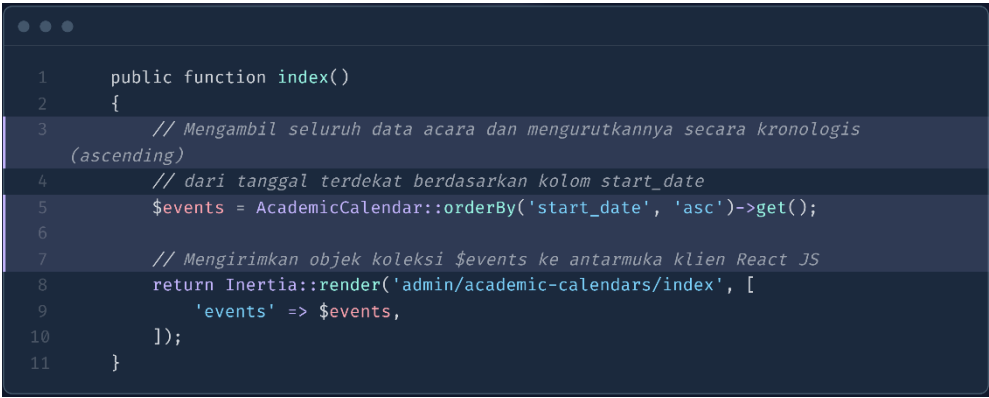


Gambar 5.70 *Sequence Diagram* Melihat Kalender Pendidikan

Sequence Diagram Kalender Pendidikan (Melihat) pada Gambar 5.70 menggambarkan urutan komunikasi antara Guru, antarmuka sistem, *controller*, dan basis data selama proses melihat data kalender pendidikan berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data kalender dikirimkan oleh Guru melalui antarmuka halaman kalender, kemudian diproses oleh *controller* (*AcademicCalendarController*) melalui metode *HTTP GET* sebelum diteruskan ke database (tabel *academic_calendars*) menggunakan kueri *SQL* untuk menghasilkan respons berupa kembalian data kalender yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pembacaan data kalender akademik bulanan dapat dipahami secara lebih rinci dan berurutan. Diagram ini menjadi acuan dalam implementasi logika penarikan data di sisi server sehingga proses pertukaran data antara antarmuka *front-end React JS*, *controller* Laravel, dan basis data *MySQL* dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan performa aplikasi yang telah dirancang sebelumnya.

d. Implementasi



```

1   public function index()
2   {
3       // Mengambil seluruh data acara dan mengurutkannya secara kronologis
      (ascending)
4       // dari tanggal terdekat berdasarkan kolom start_date
5       $events = AcademicCalendar::orderBy('start_date', 'asc')->get();
6
7       // Mengirimkan objek koleksi $events ke antarmuka klien React JS
8       return Inertia::render('admin/academic-calendars/index', [
9         'events' => $events,
10      ]);
11  }

```

Gambar 5.71 *Code Backend* Kalender Pendidikan (melihat) Guru

Potongan kode pada Gambar 5.71 menunjukkan implementasi antarmuka *Backend* untuk fitur Kalender Pendidikan pada kontroler *AcademicCalendarController*. Pada metode *index()*, sistem difokuskan untuk melakukan pengambilan seluruh entitas agenda dari basis data. Teknik utama yang diimplementasikan pada kueri ini adalah pengurutan di tingkat basis data (*database-level sorting*) melalui klausa *orderBy('start_date', 'asc')*. Pengurutan secara menaik (kronologis) ini secara sengaja diproses di sisi peladen (*server-side*) untuk menjamin bahwa antarmuka klien menerima muatan data (*payload*) secara terstruktur dari waktu terdekat, sehingga beban memori (*memory overhead*) pada saat me-render visualisasi kalender di sisi antarmuka (*Frontend*) menjadi lebih ringan dan cepat.

```

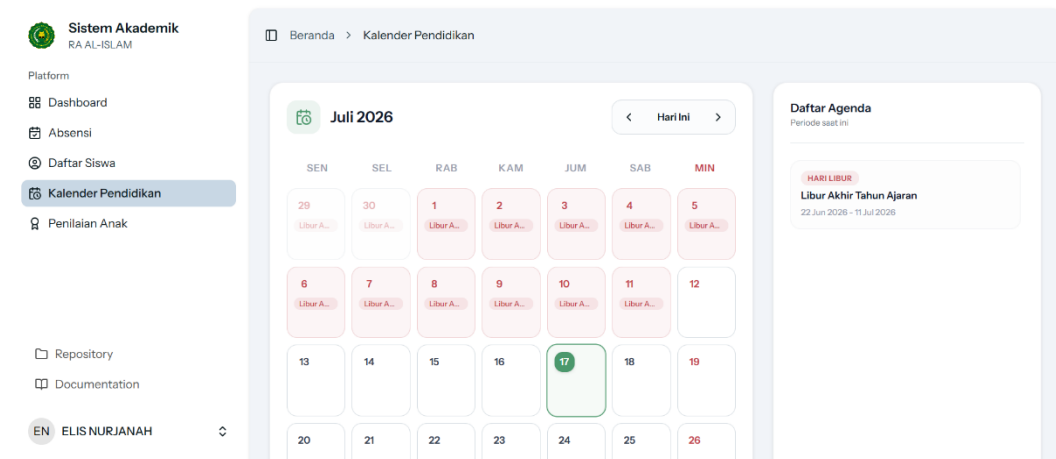
1 // — Menginisialisasi Perhitungan Logika Kalender —
2 const monthStart = startOfMonth(currentMonth);
3 const monthEnd = endOfMonth(monthStart);
4 // Mengatur agar kalender dimulai pada hari Senin (weekStartsOn: 1)
5 const startDate = startOfWeek(monthStart, { weekStartsOn: 1 });
6 const endDate = endOfWeek(monthEnd, { weekStartsOn: 1 });
7 // Menghasilkan array (senarai) seluruh hari yang akan digambar di layar
8 const days = eachDayOfInterval({ start: startDate, end: endDate });
9
10 /**
11  * Algoritma penyaringan (filtering) acara untuk merender kotak tanggal.
12  * Fungsi ini akan dipanggil untuk setiap kotak tanggal di layar.
13  */
14 const getDayEvents = (day: Date) => {
15   return events.filter((event) => {
16     // Mengkonversi string ISO tanggal dari Backend menjadi objek Date JS
17     const start = parseISO(event.start_date.substring(0, 10));
18     const end = parseISO(event.end_date.substring(0, 10));
19
20     // Memeriksa irisan waktu: apakah 'day' (hari yang sedang di-render)
21     // berada di antara rentang waktu (interval) start_date dan end_date
22     return (
23       isWithinInterval(day, { start, end }) || isSameDay(day, start)
24     );
25   });
26 };
27

```

Gambar 5.72 Code Frontend Kalender Pendidikan (melihat) Guru

Potongan kode pada Gambar 5.72 menunjukkan algoritma logika pada *Frontend* untuk memproses dan menampilkan data Kalender Pendidikan. Berbeda dengan komponen data tabular biasa, tampilan kalender direpresentasikan dalam bentuk kisi-kisi tanggal (*date grid*). Untuk menangani kompleksitas perhitungan hari, sistem memanfaatkan pustaka *date-fns*. Jantung utama dari visualisasi kalender ini berada pada algoritma fungsional *getDayEvents()*. Fungsi ini akan melakukan iterasi berulang pada setiap blok tanggal di layar. Fungsi tersebut menerapkan metode *isWithinInterval* untuk menghitung secara presisi apakah sebuah kotak tanggal berada tepat di dalam irisan rentang waktu (*time span*) antara variabel *start_date* dan *end_date* milik sebuah agenda. Jika kondisi logika tersebut terpenuhi, maka sistem akan secara otomatis merender agenda/hari libur tersebut ke dalam antarmuka blok tanggal yang bersangkutan.

e. Tampilan halaman



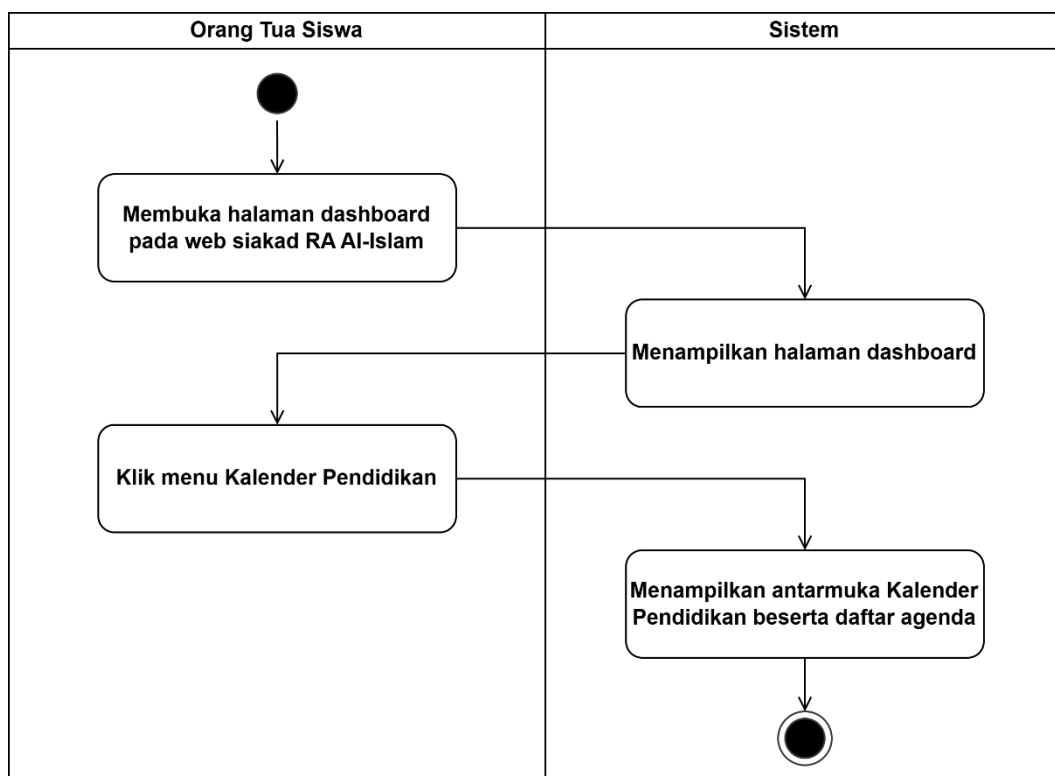
Gambar 5.73 Halaman Melihat Kalender Pendidikan Guru

Gambar 5.73 menunjukkan hasil implementasi halaman Kalender Pendidikan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan visualisasi utama berupa *grid* tanggal bulanan interaktif beserta panel "Daftar Agenda" di sisi kanan yang merangkum rincian pelaksanaan kegiatan penting pada periode berjalan. Selain itu, tersedia menu *navigasi platform* di bilah sisi kiri, fasilitas perpindahan bulan, serta penanda visual khusus pada tanggal terpilih sebagai media bagi Guru maupun Orang Tua untuk melihat jadwal kegiatan akademik sesuai dengan kebutuhan.

Tampilan dirancang dengan antarmuka yang sederhana dan bersih sehingga memudahkan pengguna dari berbagai latar belakang dalam memantau agenda sekolah secara lebih efektif. Selain itu, sistem juga menyediakan informasi detail rentang waktu pelaksanaan agenda (seperti Libur Akhir Tahun Ajaran) serta pengelompokan jenis kegiatan yang terstruktur sehingga dapat digunakan sebagai acuan bersama dalam pelaksanaan proses pembelajaran. Dengan demikian, halaman Kalender Pendidikan tidak hanya berfungsi sebagai media penyajian informasi operasional, tetapi juga membantu menjaga konsistensi transparansi waktu pada seluruh fitur akademik di dalam sistem.

2. Kalender pendidikan orang tua siswa

a. Activity diagram



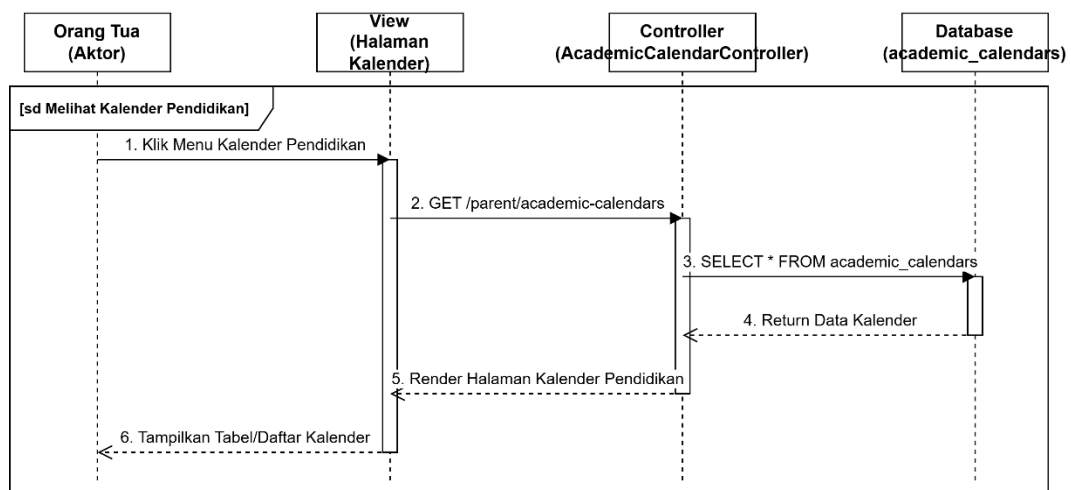
Gambar 5.74 Activity Diagram Melihat Kalender Pendidikan Orang Tua Siswa

Activity Diagram Kalender Pendidikan (Melihat) Orang Tua Siswa pada Gambar 5.74 menggambarkan alur proses melihat jadwal atau agenda akademik yang dilakukan oleh Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Orang Tua Siswa membuka Dashboard kemudian memilih menu Kalender Pendidikan sehingga sistem menampilkan halaman data kalender pendidikan yang berisi seluruh data agenda yang telah tersimpan pada basis data. Selanjutnya Orang Tua Siswa dapat melihat visualisasi kalender beserta daftar agenda kegiatan akademik secara langsung sesuai dengan kebutuhan pemantauan waktu kegiatan anak di lingkungan sekolah.

Apabila Orang Tua Siswa mengakses menu Kalender Pendidikan, sistem secara otomatis akan menarik seluruh rekaman data jadwal yang aktif dari database untuk diproyeksikan ke dalam antarmuka kalender bulanan. Proses pemuatan informasi

agenda ini dirancang secara terstruktur dan efisien agar visualisasi jadwal kegiatan penting sekolah dapat langsung dibaca dengan jelas. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme menampilkan informasi kalender pendidikan dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

b. *Sequence diagram*



Gambar 5.75 *Sequence Diagram* Melihat Kalender Pendidikan Orang Tua Siswa

Sequence Diagram Kalender Pendidikan (Melihat) Orang Tua Siswa pada Gambar 5.75 menggambarkan urutan komunikasi antara Orang Tua, antarmuka sistem, *controller*, dan basis data selama proses melihat data kalender pendidikan berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data kalender dikirimkan oleh Orang Tua melalui antarmuka halaman kalender, kemudian diproses oleh *controller* (*AcademicCalendarController*) melalui metode HTTP GET dengan rute */parent/academic-calendars* sebelum diteruskan ke *database* (tabel *academic_calendars*) menggunakan kueri SQL untuk menghasilkan respons berupa kembalian data kalender yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pembacaan data kalender akademik bulanan dapat dipahami secara lebih rinci dan

berurutan. Diagram ini menjadi acuan dalam implementasi logika penarikan data di sisi server sehingga proses pertukaran data antara antarmuka front-end React JS, controller Laravel, dan basis data MySQL dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan performa aplikasi yang telah dirancang sebelumnya.

c. Implementasi



```

1 // File: routes/web.php
2
3 // Portal khusus Orang Tua (Parent)
4 Route::middleware(['auth'])->prefix('parent')->name('parent.')->group(function () {
5
6     // Rute Kalender Pendidikan Orang Tua
7     // Menggunakan ulang (reuse) Controller yang sama dengan Admin untuk efisiensi
8     // kode
9     Route::get(
10         'academic-calendars',
11         [\App\Http\Controllers\Admin\AcademicCalendarController::class, 'index']
12     )->name('academic-calendars.index');
13 });

```

Gambar 5.76 *Code Backend* Kalender Pendidikan (melihat) Orang Tua Siswa

Potongan kode pada Gambar 5.76 menunjukkan konfigurasi rute (*Routing*) dan logika *Backend* untuk fitur Kalender Pendidikan khusus aktor Orang Tua. Alih-alih membuat kontroler terpisah, arsitektur sistem menerapkan prinsip DRY (*Don't Repeat Yourself*) dengan menggunakan ulang (*reuse*) *AcademicCalendarController* milik Admin. Meskipun menggunakan rute (*endpoint*) yang berbeda yaitu di bawah prefiks */parent*, sistem tetap mengarah ke metode *index()* yang sama untuk mengambil data database. Hal ini sangat efisien karena algoritma pengambilan data kalender berlaku universal untuk semua aktor, sehingga mengurangi duplikasi kode di tingkat peladen (*server-side redundancy*).

```

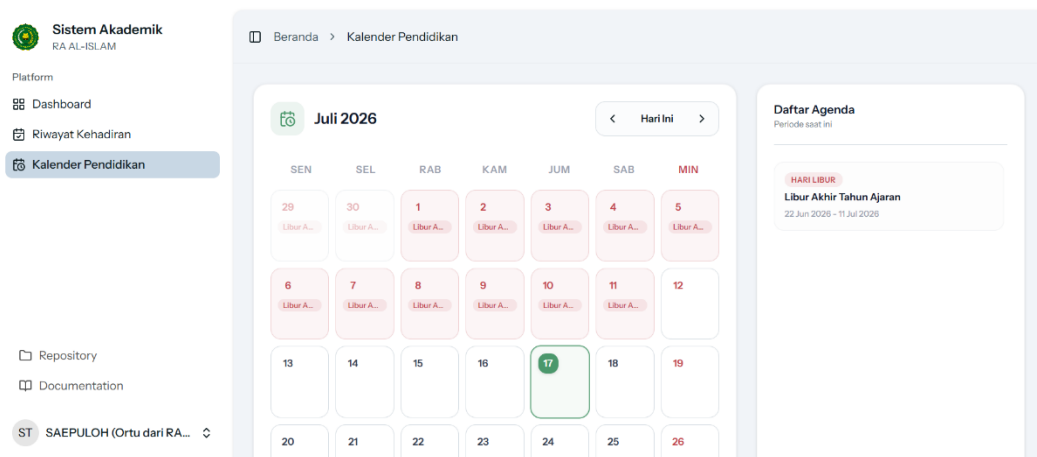
1 // File: resources/js/pages/admin/academic-calendars/index.tsx
2
3 export default function AcademicCalendarIndex({ events }: PageProps) {
4   const { auth } = usePage<PageProps>().props;
5
6   // Mengevaluasi peran (role) pengguna yang sedang aktif (Hanya bernilai true jika
   'admin')
7   const isAdmin = auth.user?.role === 'admin';
8
9   return (
10     <AppLayout breadcrumbs={breadcrumbs}>
11       {/* Bagian Grid Kalender */}
12       <div className="grid grid-cols-7 gap-2">
13         {days.map((day, idx) => {
14           return (
15             <div
16               key={idx}
17               // Mematikan visualisasi kursor tangan (pointer) jika
               bukan admin
18               className={`... ${isAdmin ? 'cursor-pointer' : ''}`}
19               // Memblokir akses klik untuk membuka Modal Edit/Tambah
               jika bukan admin
20               onClick={() => isAdmin && handleOpenModal(undefined, day)}
21             >
22               {/* Render UI Tanggal dan Acara... */}
23             </div>
24           );
25         })}
26       </div>
27
28       {/* Bagian Tombol Aksi */}
29       <div className="flex items-center justify-between">
30         <h3>Daftar Agenda</h3>
31
32         {/* Conditional Rendering: Tombol Tambah HANYA dirender jika pengguna
           adalah Admin */}
33         {isAdmin && (
34           <Button onClick={() => handleOpenModal()}>
35             <Plus /> Tambah
36           </Button>
37         )}
38       </div>
39
40       {/* Conditional Rendering pada elemen daftar agenda */}
41       {currentMonthEvents.map((e) => (
42         <div key={e.id}>
43           {/* Render informasi agenda ... */}
44
45           {/* Tombol Edit dan Hapus HANYA dirender jika pengguna adalah
           Admin */}
46           {isAdmin && (
47             <div className="flex flex-col gap-1">
48               <Button onClick={() => handleOpenModal(e)}><Edit />
49             </Button>
50             <DeleteDialog onConfirm={() => destroy(`/admin/academic-
           calendars/${e.id}`)} />
51             </div>
52           )}
53         </div>
54       ))}
55     </AppLayout>
56   );
57 }

```

Gambar 5.77 Code Frontend Kalender Pendidikan (melihat) Orang Tua Siswa

Potongan kode pada Gambar 5.77 menunjukkan implementasi antarmuka *Frontend* Kalender Pendidikan. Mengingat sistem menggunakan berkas tampilan (*View*) yang sama dengan Admin, pembatasan hak akses untuk Orang Tua dikendalikan sepenuhnya melalui teknik *Conditional Rendering* (Perenderan Bersyarat) berbasis Role. Sistem mendeteksi identitas pengguna melalui objek properti global *auth.user.role*. Apabila pengguna yang masuk bukan Admin (dalam hal ini Orang Tua), maka seluruh interaksi mutasi data seperti tombol "Tambah Agenda", tombol "Edit", tombol "Hapus", hingga fungsi klik pada sel tanggal akan disembunyikan dan dimatikan sepenuhnya. Hal ini secara efektif mengubah antarmuka kalender yang tadinya interaktif menjadi mode *Read-Only* (Hanya Lihat) khusus untuk Orang Tua.

d. Tampilan halaman



Gambar 5.78 Halaman Kalender Pendidikan (melihat) Orang Tua Siswa

Gambar 5.78 menunjukkan hasil implementasi halaman Kalender Pendidikan Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan visualisasi utama berupa *grid* tanggal bulanan interaktif beserta panel "Daftar Agenda" di sisi kanan yang merangkum rincian pelaksanaan kegiatan penting pada periode berjalan. Selain itu, tersedia menu navigasi *platform* di bilah sisi kiri (seperti menu Dashboard, Riwayat Kehadiran, dan Kalender Pendidikan), fasilitas perpindahan bulan, serta informasi penanda identitas akun

wali murid pada pojok bawah sebagai media bagi Orang Tua untuk memantau jadwal kegiatan akademik anak secara mandiri.

Tampilan dirancang dengan antarmuka yang sederhana dan bersih sehingga memudahkan pengguna dari berbagai latar belakang dalam memantau agenda sekolah secara lebih efektif. Selain itu, sistem juga menyediakan informasi detail rentang waktu pelaksanaan agenda (seperti Libur Akhir Tahun Ajaran) serta pengelompokan jenis kegiatan yang terstruktur sehingga dapat digunakan sebagai acuan pengawasan bagi orang tua dari rumah. Dengan demikian, halaman Kalender Pendidikan tidak hanya berfungsi sebagai media penyajian informasi operasional, tetapi juga membantu menjaga konsistensi transparansi waktu pada seluruh fitur akademik di dalam sistem.

5.4.1.8 *Black box testing sprint 1*

Setelah seluruh fitur pada *Sprint 1* berhasil diimplementasikan, tahapan selanjutnya adalah melakukan pengujian menggunakan metode *Black Box Testing*. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur yang telah dikembangkan dapat berfungsi sesuai dengan kebutuhan fungsional yang telah ditetapkan pada *Product Backlog*. Berbeda dengan pengujian yang berfokus pada struktur kode program, *Black Box Testing* dilakukan dengan mengamati kesesuaian antara masukan (*input*) yang diberikan pengguna dengan keluaran (*output*) yang dihasilkan oleh sistem. Dengan demikian, pengujian ini dapat membuktikan bahwa setiap fungsi yang tersedia telah bekerja sebagaimana mestinya sebelum hasil pengembangan dipresentasikan kepada pihak RA Al-Islam Gunungcupu melalui *Sprint Review*.

Pengujian pada *Sprint 1* dilakukan terhadap seluruh fitur yang berhasil diselesaikan, yaitu Login, Dashboard, Pengguna, Tahun Pelajaran, Kelas, dan Kalender Pendidikan (Melihat). Setiap fitur diuji berdasarkan skenario penggunaan yang mewakili aktivitas pengguna pada sistem, kemudian dibandingkan dengan hasil yang diharapkan untuk mengetahui kesesuaian fungsi yang dihasilkan. Hasil pengujian menunjukkan bahwa seluruh fitur pada *Sprint 1* dapat berjalan sesuai dengan kebutuhan fungsional tanpa ditemukan kesalahan yang memengaruhi

proses utama sistem. Oleh karena itu, seluruh fitur dinyatakan berhasil dan layak untuk dilanjutkan ke tahapan *Sprint Review* sebagai bahan evaluasi bersama stakeholder.

Tabel 5.5 Hasil *Black Box Testing Sprint 1*

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
1	Login Admin	Email dan password valid	Sistem menampilkan Dashboard Admin	Dashboard Admin berhasil ditampilkan	Berhasil
2	Login Guru	Email dan password valid	Sistem menampilkan Dashboard Guru	Dashboard Guru berhasil ditampilkan	Berhasil
3	Login Orang Tua	Email dan password valid	Sistem menampilkan Dashboard Orang Tua	Dashboard Orang Tua berhasil ditampilkan	Berhasil
4	Menambahkan, mengubah, dan menghapus data Admin	Data Admin	Data Admin berhasil dikelola	Data berhasil ditambah, diubah, dan dihapus	Berhasil
5	Menambahkan, mengubah, dan menghapus data Guru	Data Guru	Data Guru berhasil dikelola	Data berhasil ditambah, diubah, dan dihapus	Berhasil

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
6	Menambahkan, mengubah, dan menghapus data Orang Tua	Data Orang Tua	Data Orang Tua berhasil dikelola	Data berhasil ditambah, diubah, dan dihapus	Berhasil
7	Menambahkan, mengubah, dan menghapus Tahun Pelajaran	Data Tahun Pelajaran	Data berhasil dikelola	Data berhasil diproses	Berhasil
8	Menambahkan, mengubah, dan menghapus Kelas	Data Kelas	Data berhasil dikelola	Data berhasil diproses	Berhasil
9	Melihat Kalender Pendidikan	Memilih menu kalender	Data kalender ditampilkan	Kalender berhasil ditampilkan	Berhasil
10	Memproses Kenaikan Kelas Siswa	Data siswa, kelas tujuan, dan tahun pelajaran tujuan	Sistem berhasil memproses kenaikan kelas dan memperbarui data siswa	Kenaikan kelas berhasil diproses dan data siswa terbaru	Berhasil

Berdasarkan hasil *Black Box Testing* pada Tabel 5.5, seluruh skenario pengujian yang dilakukan pada fitur-fitur *Sprint 1* menunjukkan status berhasil sesuai dengan ekspektasi yang telah ditetapkan. Setiap fungsi mampu menghasilkan keluaran

yang sesuai terhadap masukan yang diberikan, baik pada proses autentikasi pengguna, pengelolaan data, maupun penyajian informasi kepada pengguna. Hasil tersebut menunjukkan bahwa seluruh fitur yang dikembangkan pada *Sprint* 1 telah memenuhi kebutuhan fungsional sistem sehingga layak untuk dilanjutkan ke tahap *Sprint Review* sebagai proses evaluasi bersama stakeholder sebelum memasuki *Sprint* berikutnya.

5.4.1.9 *Sprint review sprint 1*

Setelah seluruh fitur pada *Sprint* 1 berhasil diimplementasikan dan dinyatakan berfungsi dengan baik melalui *Black Box Testing*, tahapan selanjutnya adalah melaksanakan *Sprint Review*. Pada tahap ini, hasil pengembangan sistem dipresentasikan kepada stakeholder, yaitu pihak RA Al-Islam Gunungcupu, untuk memperoleh umpan balik terhadap fitur-fitur yang telah dikembangkan. Kegiatan ini bertujuan untuk memastikan bahwa setiap fungsi yang diimplementasikan telah sesuai dengan kebutuhan operasional sekolah serta memenuhi harapan pengguna sebelum proses pengembangan dilanjutkan ke *Sprint* berikutnya.

Sprint Review pada *Sprint* 1 dilakukan terhadap seluruh fitur yang berhasil dikembangkan, yaitu Login, Dashboard, Pengelolaan Data Pengguna (Admin, Guru, dan Orang Tua), Tahun Pelajaran, Kelas, Kelas (Kenaikan Kelas), serta Kalender Pendidikan (Melihat). Hasil evaluasi menunjukkan bahwa seluruh fitur telah berjalan dengan baik, sesuai dengan kebutuhan pengguna, dan tidak memerlukan adanya perbaikan maupun penyesuaian fungsi. Hasil presentasi dan pengujian bersama stakeholder menyatakan bahwa seluruh fitur dasar pada *Sprint* 1 telah memenuhi spesifikasi fungsional yang diharapkan oleh sekolah. Hasil *Sprint Review Sprint 1* dapat dilihat pada Tabel 5.6.

Tabel 5.6 Hasil *Sprint Review Sprint 1*

Pengguna	Fitur	Review
Admin	Login	Fitur login telah berjalan dengan baik sesuai kebutuhan. Proses autentikasi berhasil

		mengarahkan pengguna ke halaman Dashboard berdasarkan hak akses Admin.
	Dashboard	Dashboard berhasil menampilkan informasi sesuai kebutuhan Admin dan seluruh menu dapat diakses dengan baik.
	Pengelolaan Data Admin	Fitur pengelolaan data Admin telah berjalan sesuai kebutuhan dan seluruh fungsi pengelolaan data dapat digunakan dengan baik.
	Pengelolaan Data Guru	Fitur pengelolaan data Guru telah berjalan sesuai kebutuhan dan seluruh fungsi pengelolaan data dapat digunakan dengan baik.
	Pengelolaan Data Orang Tua	Fitur pengelolaan data Orang Tua telah berjalan sesuai kebutuhan dan seluruh fungsi pengelolaan data dapat digunakan dengan baik.
	Tahun Pelajaran	Fitur pengelolaan Tahun Pelajaran telah berjalan sesuai kebutuhan dan tidak ditemukan kendala selama proses penggunaan.
	Kelas	Fitur pengelolaan Kelas telah berjalan sesuai kebutuhan dan seluruh proses pengelolaan data berhasil dilakukan.
	Kenaikan Kelas	Fitur kenaikan kelas masal telah berjalan sesuai kebutuhan. Proses pemindahan jenjang rombongan belajar siswa secara kolektif berhasil dijalankan dengan baik.
Guru	Dashboard	Dashboard Guru telah menampilkan informasi sesuai dengan kebutuhan pengguna dan mudah dipahami.

	Kalender Pendidikan (Melihat)	Informasi kalender pendidikan berhasil ditampilkan sesuai dengan data yang tersedia pada sistem.
Orang Tua	Dashboard	Dashboard Orang Tua telah menampilkan informasi sesuai dengan kebutuhan pengguna dan dapat diakses dengan baik.
	Kalender Pendidikan (Melihat)	Informasi kalender pendidikan berhasil ditampilkan dengan baik sehingga dapat diakses oleh Orang Tua Siswa.

Berdasarkan hasil *Sprint Review* pada Tabel 5.6, seluruh fitur yang dikembangkan pada *Sprint 1* telah memenuhi kebutuhan operasional RA Al-Islam Gunungcupu dan dapat digunakan sesuai dengan tujuan pengembangannya. Masukan yang diperoleh dari stakeholder menjadi bahan evaluasi untuk meningkatkan kualitas sistem pada *Sprint* berikutnya, sedangkan fitur yang telah dinilai sesuai dipertahankan sebagai bagian dari *increment* sistem yang siap mendukung proses pengembangan selanjutnya melalui tahapan *Sprint Retrospective*.

5.4.1.10 *Sprint retrospective sprint 1*

Setelah *Sprint Review* selesai dilaksanakan, tahapan berikutnya adalah *Sprint Retrospective*. Pada tahap ini, peneliti melakukan evaluasi terhadap keseluruhan proses pelaksanaan *Sprint 1*, mulai dari perencanaan *backlog*, implementasi sistem, pengujian menggunakan *Black Box Testing*, hingga hasil *Sprint Review* yang telah dilakukan bersama *stakeholder*. Evaluasi ini bertujuan untuk mengidentifikasi hal-hal yang telah berjalan dengan baik selama *Sprint* berlangsung serta memastikan bahwa proses pengembangan telah sesuai dengan perencanaan yang telah ditetapkan sebelumnya. Hasil *Sprint Retrospective Sprint 1* dapat dilihat pada Tabel 5.7.

Tabel 5.7 Hasil *Sprint Retrospective Sprint 1*

Pengguna	Aspek Evaluasi	<i>Sprint Retrospective</i>
Peneliti	Pelaksanaan Sprint 1	Seluruh <i>backlog</i> pada <i>Sprint 1</i> berhasil diselesaikan sesuai dengan perencanaan. Proses implementasi sistem berjalan dengan baik, pengujian <i>Black Box Testing</i> menunjukkan seluruh fungsi berjalan sesuai kebutuhan, serta hasil <i>Sprint Review</i> menunjukkan bahwa seluruh fitur diterima oleh <i>stakeholder</i> tanpa revisi. Berdasarkan hasil evaluasi tersebut, proses pengembangan dapat dilanjutkan ke <i>Sprint</i> berikutnya.

Berdasarkan hasil *Sprint Retrospective* pada Tabel 5.7, dapat diketahui bahwa pelaksanaan *Sprint 1* telah berjalan sesuai dengan perencanaan yang telah disusun pada tahap *Sprint Planning*. Seluruh *backlog* yang menjadi target *Sprint* berhasil diselesaikan tepat waktu dan diimplementasikan sesuai dengan kebutuhan sistem. Selain itu, proses implementasi menggunakan Laravel 12 dan React JS berlangsung dengan baik, seluruh skenario *Black Box Testing* menunjukkan hasil sesuai dengan yang diharapkan, serta hasil *Sprint Review* menyatakan bahwa seluruh fitur yang dikembangkan telah diterima oleh *stakeholder* tanpa memerlukan revisi. Oleh karena itu, hasil evaluasi pada *Sprint Retrospective* menunjukkan bahwa proses pengembangan pada *Sprint 1* telah berjalan secara efektif dan dapat dijadikan dasar untuk melanjutkan pengembangan pada *Sprint* berikutnya dengan tetap mempertahankan proses kerja yang telah berjalan dengan baik.

5.4.2 Sprint 2

Sprint 2 merupakan tahapan lanjutan dalam proses pengembangan Sistem Informasi Akademik RA Al-Islam Gunungcupu setelah seluruh fitur dasar pada *Sprint 1* berhasil diimplementasikan. Fokus utama pada *Sprint* ini adalah mengembangkan

fitur-fitur yang berkaitan dengan pengelolaan data akademik utama, yaitu Siswa, Absensi, Kalender Pendidikan, serta Cetak Biodata Siswa dalam format PDF. Pengembangan fitur-fitur tersebut bertujuan untuk mendukung proses pengelolaan data akademik secara lebih lengkap sehingga aktivitas pendataan siswa, pencatatan kehadiran, penyampaian informasi kegiatan sekolah, hingga penyajian dokumen biodata dapat dilakukan secara terintegrasi melalui sistem berbasis web.

Sebelum proses implementasi dilakukan, setiap fitur pada *Sprint 2* terlebih dahulu dirancang menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam menggambarkan alur proses, struktur data, serta interaksi antar komponen sistem. Selanjutnya, hasil perancangan tersebut diimplementasikan menggunakan framework Laravel 12 sebagai backend dan React JS sebagai frontend. Pendekatan ini bertujuan agar setiap fitur yang dikembangkan memiliki alur kerja yang jelas, mudah dipelihara, serta mampu memenuhi kebutuhan pengguna sesuai dengan hasil analisis kebutuhan yang telah dilakukan pada tahap sebelumnya.

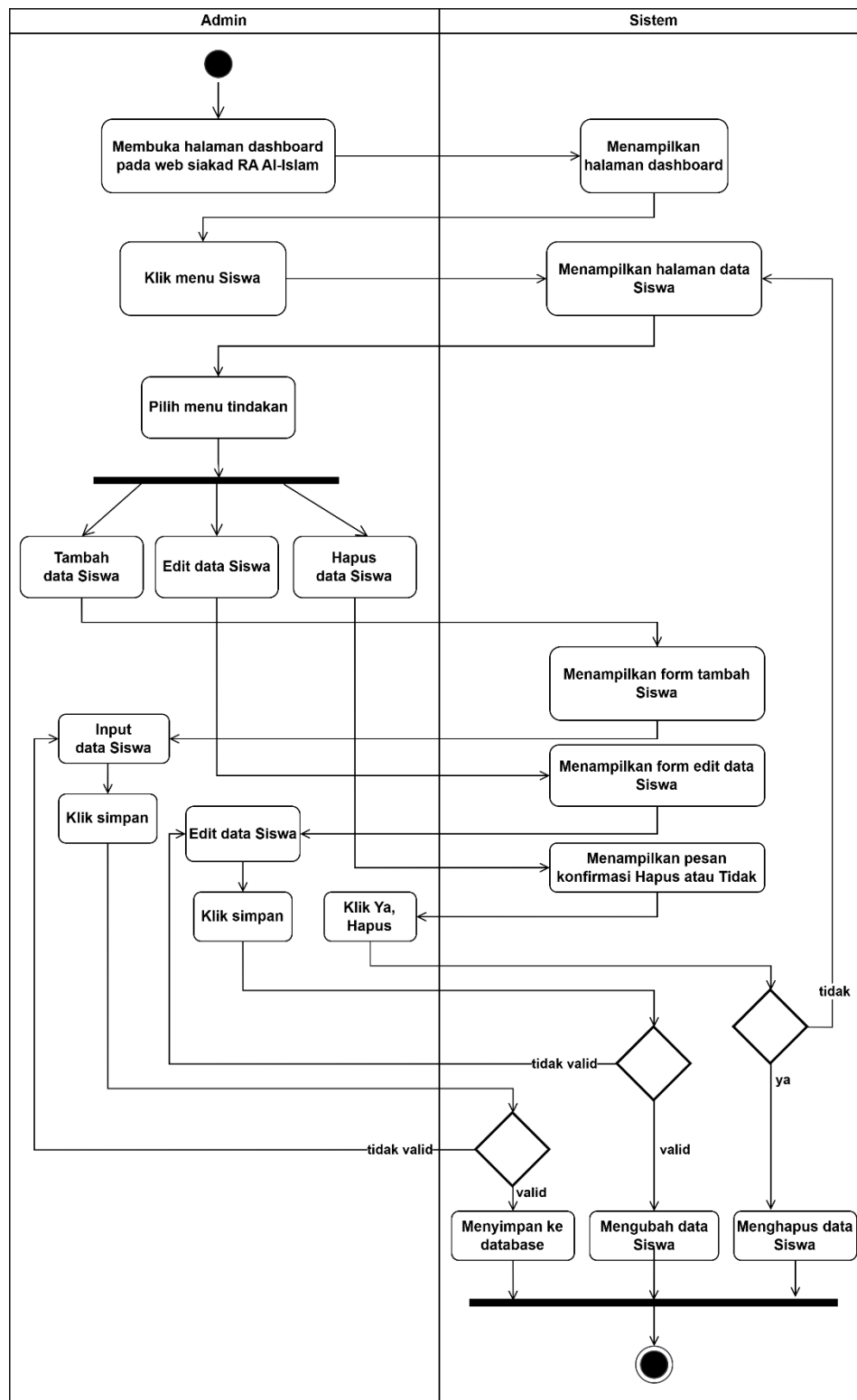
5.4.2.1 Siswa (kelola siswa)

Fitur Kelola Siswa merupakan salah satu fitur utama yang dikembangkan pada *Sprint 2* karena berfungsi sebagai pusat pengelolaan data peserta didik pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Admin dapat melakukan proses pengelolaan data siswa yang meliputi penambahan, perubahan, penghapusan, serta melihat data siswa yang telah tersimpan dalam sistem. Informasi yang dikelola mencakup identitas siswa, data orang tua, penempatan kelas, serta informasi akademik lainnya yang dibutuhkan selama proses pembelajaran. Dengan adanya pengelolaan data secara terpusat, proses administrasi siswa menjadi lebih terstruktur, meminimalkan kesalahan pencatatan, serta mempermudah pencarian dan pembaruan data apabila terjadi perubahan.

Perancangan fitur Kelola Siswa dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pengelolaan data sebelum implementasi dilakukan. Ketiga diagram tersebut

membantu menggambarkan aktivitas pengguna, struktur data siswa yang digunakan oleh sistem, serta urutan komunikasi antara pengguna dengan sistem selama proses pengelolaan data berlangsung. Setelah proses perancangan selesai, fitur diimplementasikan menggunakan framework Laravel 12 sebagai *backend* dan React JS sebagai *frontend* sehingga seluruh proses pengelolaan data siswa dapat berjalan secara terintegrasi, mudah dipelihara, serta sesuai dengan kebutuhan pengguna yang telah diidentifikasi pada tahap analisis kebutuhan.

1. *Activity diagram siswa (kelola siswa)*

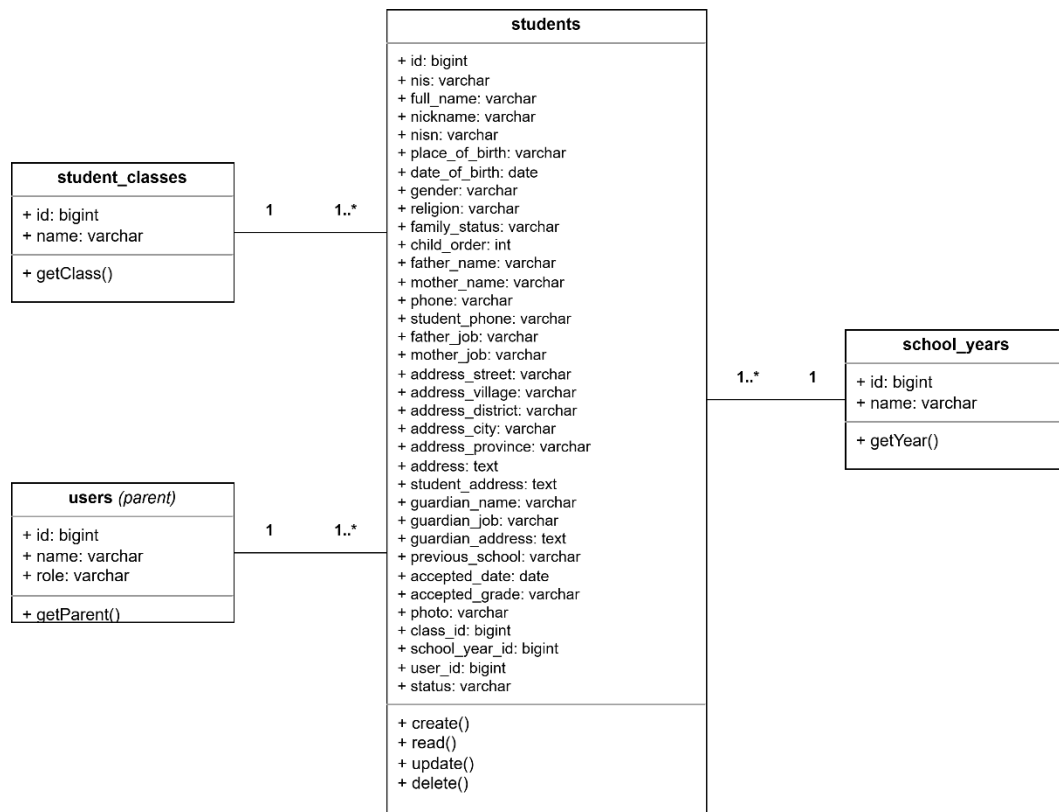


Gambar 5.79 Activity Diagram Siswa (Kelola siswa)

Activity Diagram Siswa pada Gambar 5.79 menggambarkan alur proses pengelolaan data siswa yang dilakukan oleh Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Siswa sehingga sistem menampilkan halaman data siswa yang berisi seluruh data yang telah tersimpan pada basis data. Selanjutnya Admin dapat memilih tindakan berupa menambah data, mengubah data, maupun menghapus data siswa sesuai dengan kebutuhan pengelolaan akademik.

Apabila Admin memilih tambah atau ubah data, sistem akan menampilkan formulir yang harus diisi kemudian melakukan proses validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, apabila Admin memilih menghapus data, sistem akan terlebih dahulu menampilkan konfirmasi penghapusan untuk menghindari kesalahan tindakan. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pengelolaan data siswa dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. Class diagram siswa (kelola siswa)

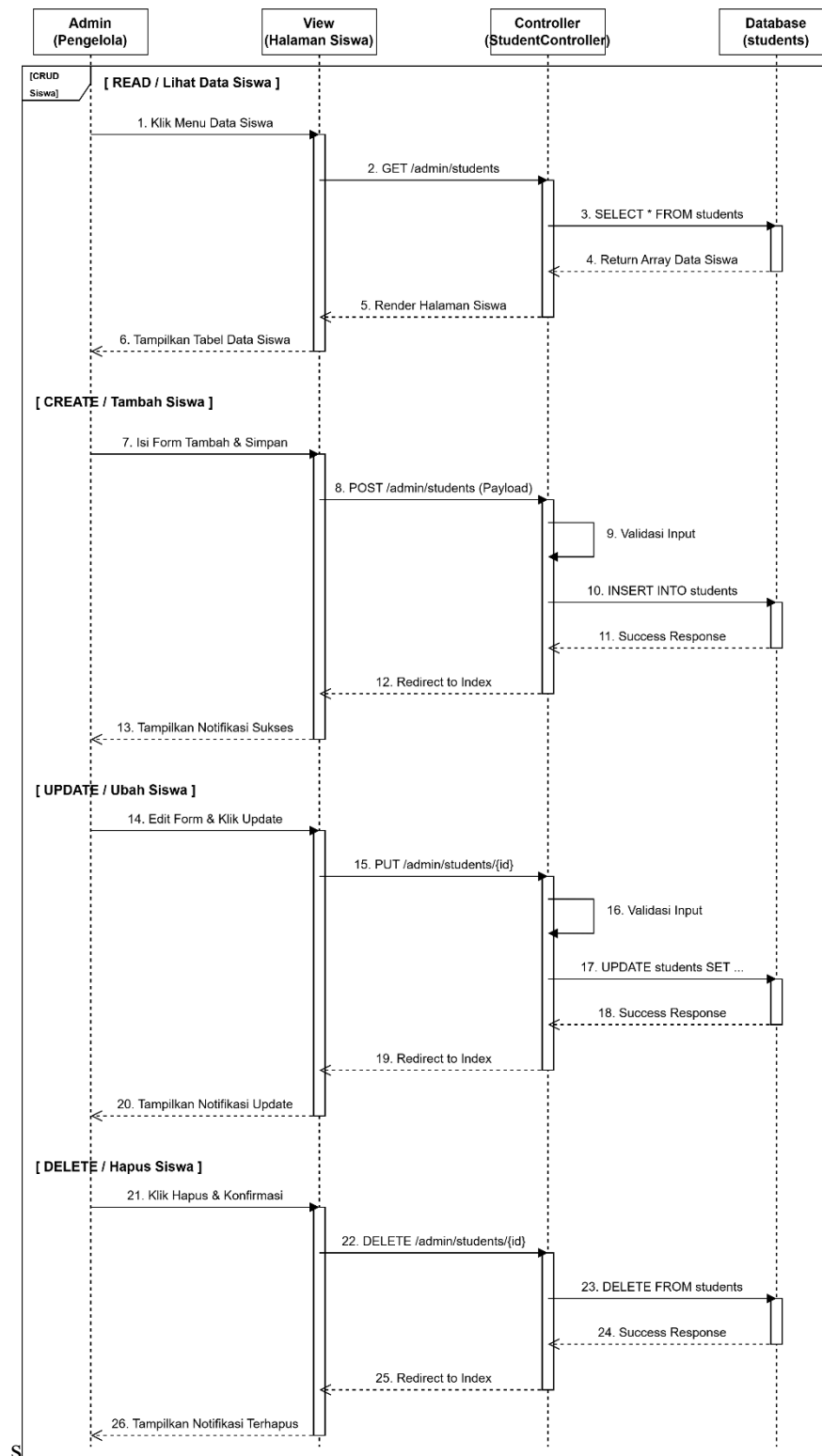


Gambar 5.80 Class Diagram Siswa (Kelola siswa)

Class Diagram Siswa pada Gambar 5.80 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses pengelolaan data siswa pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *students* yang memuat banyak atribut rinci (seperti *id*, *nis*, *full_name*, *nickname*, *nisn*, *place_of_birth*, *date_of_birth*, *gender*, *religion*, *family_status*, *child_order*, *father_name*, *mother_name*, *phone*, *student_phone*, *father_job*, *mother_job*, *address_street*, *address_village*, *address_district*, *address_city*, *address_province*, *address*, *student_address*, *guardian_name*, *guardian_job*, *guardian_address*, *previous_school*, *accepted_date*, *accepted_grade*, *photo*, *class_id*, *school_year_id*, *user_id*, dan *status*) beserta operasi CRUD (*create()*, *read()*, *update()*, *delete()*), dengan kelas *student_classes* melalui fungsi *getClass()*, kelas *users (parent)* melalui fungsi *getParent()*, dan kelas *school_years* melalui fungsi *getYear()*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengorganisasian data murid dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* di mana entitas students dapat terikat pada banyak data kelas (*student_classes*), banyak tahun ajaran (*school_years*), dan satu data orang tua (*users*), sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan struktural antar-tabel melalui pemetaan kunci tamu (*class_id*, *school_year_id*, *user_id*) sehingga setiap proses penambahan, perubahan, maupun penghapusan data siswa dapat dilakukan secara konsisten sesuai dengan kebutuhan integritas basis data pada sistem.

3. Sequence diagram siswa (kelola siswa)



Gambar 5.81 Sequence Diagram Siswa (Kelola siswa)

Sequence Diagram Siswa pada Gambar 5.81 menggambarkan urutan komunikasi antara Admin, antarmuka sistem, *controller*, dan basis data selama proses pengelolaan data siswa berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data, menambahkan data baru, mengubah data, maupun menghapus data dikirimkan oleh Admin dari *View* (Halaman Siswa), kemudian diproses oleh *StudentController* melalui proses validasi sebelum diteruskan ke Database (tabel *students*) hingga menghasilkan respons yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengelolaan data siswa dapat dipahami secara lebih rinci dan berurutan. Diagram ini menjadi acuan dalam implementasi logika sistem sehingga proses pertukaran data antara antarmuka, *controller*, dan basis data dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi siswa (kelola siswa)

```

1      // Auto-create parent user:
2      // Dieksekusi hanya jika Admin tidak memilih/menautkan akun Orang Tua yang
      sudah ada
3      if (empty($studentData['user_id'])) {
4          // Ekstraksi nama orang tua untuk dijadikan profil pengguna
5          $parentName = $request->input('father_name') ?: ($request-
>input('mother_name') ?: 'Orang Tua');
6          $accountName = $parentName . ' (Ortu dari ' . $request->input('full_name')
. ')';
7
8          // Generate email unik berbasis string nama, atau fallback menggunakan
      kombinasi NISN
9          $baseEmailStr = strtolower(preg_replace('/[^a-zA-Z0-9]/', '',
      $parentName));
10         if (empty($baseEmailStr)) {
11             $fallbackStr = $request->input('nisn') ?: ($request->input('nis') ?:
      rand(1000, 9999));
12             $baseEmailStr = 'ortu' . preg_replace('/[^a-zA-Z0-9]/', '',
      $fallbackStr);
13         }
14
15         // Loop rekursif (while) untuk memvalidasi dan memastikan email 100% belum
      terdaftar di database
16         $uniqueEmail = $baseEmailStr . '@gmail.com';
17         $counter = 1;
18         while (\App\Models\User::where('email', $uniqueEmail)->exists()) {
19             $uniqueEmail = $baseEmailStr . $counter . '@gmail.com';
20             $counter++;
21         }
22
23         // Injeksi otomatis entitas pengguna baru ber-role 'parent'
24         $newUser = \App\Models\User::create([
25             'name' => $accountName,
26             'email' => $uniqueEmail,
27             'password' => \Illuminate\Support\Facades\Hash::make('password'),
28             'role' => 'parent',
29         ]);
30
31         // Menautkan Primary Key (ID) akun wali yang baru dibuat ke entitas Siswa
32         $studentData['user_id'] = $newUser->id;
33     }
34
35     // Menyimpan data siswa ke database
36     Student::create(array_merge($studentData, ['photo' => $photoPath]));
37

```

Gambar 5.82 Code Backend Siswa (Kelola siswa)

Potongan kode pada Gambar 5.82 menunjukkan implementasi logika *Backend* untuk fitur Pengelolaan Data Siswa pada metode *store()* di *StudentController*. Salah satu nilai arsitektural yang menonjol pada modul ini adalah penerapan mekanisme *Automated Account Provisioning* (Otomatisasi Penyediaan Akun). Ketika administrator menambahkan biodata siswa namun membiarkan relasi kolom

user_id (Akun Wali) kosong, sistem secara mandiri akan mengintersepsi alur tersebut. Sistem mengekstrak nama orang tua/wali dari *form*, secara dinamis menggenerasi kombinasi alamat email yang bersifat unik (*collision-free*), dan menyuntikkan profil baru tersebut ke dalam tabel *users* dengan peran (*role*) sebagai *parent*. Setelah entitas akun wali tersebut berhasil direkam, ID akun akan otomatis ditautkan ke data anak.

```

1  /**
2   * Fungsi reaktif yang akan dipicu (trigger) saat pengguna memilih filter
3   * dropdown Tahun Ajaran atau Kelas.
4   */
5   const handleCombinedFilterChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
6       const val = e.target.value;
7       let newClassId = '';
8       let newSyId = '';
9
10      // Mengekstrak parameter kueri berdasarkan prefix 'class_' atau 'sy_'
11      if (val.startsWith('class_')) {
12          newClassId = val.replace('class_', '');
13      } else if (val.startsWith('sy_')) {
14          newSyId = val.replace('sy_', '');
15      }
16
17      // Inertia Router: Meminta Payload JSON asinkron dari server untuk memperbarui
tabel
18      // secara instan tanpa melakukan full page reload
19      router.get(
20          route('admin.students.index'),
21          { search: search, entries: entries, class_id: newClassId, school_year_id:
newSyId, status: status },
22          { preserveState: true, replace: true },
23      );
24  };
25
26  return (
27      /* Implementasi Dynamic Grouped Dropdown di View HTML (JSX) */
28      <select
29          className="w-full rounded-lg border text-sm text-foreground"
30          value={combinedFilterValue}
31          onChange={handleCombinedFilterChange}
32      >
33          <option value="">Semua Tahun Ajaran & Kelas</option>
34          {schoolYears.map((sy) => (
35              /* Pembungkus Induk Kategori (Tahun Ajaran) */
36              <optgroup key={`sy_${sy.id}`} label={`T.A. ${sy.name} ${sy.is_active ?
'(Aktif)' : ''}`}>
37                  <option value={`sy_${sy.id}`}>Semua Kelas di T.A. {sy.name}
</option>
38
39                  {/* Melakukan Nested Filter (Penyaringan Bersarang) untuk merender
anak kelas */}
40                  {studentClasses
41                      .filter((c) => c.school_year_id === sy.id)
42                      .map((c) => (
43                          <option key={`class_${c.id}`} value={`class_${c.id}`}>
44                              {c.name}
45                          </option>
46                      ))}
47                  </optgroup>
48              )}
49          </select>
50  );

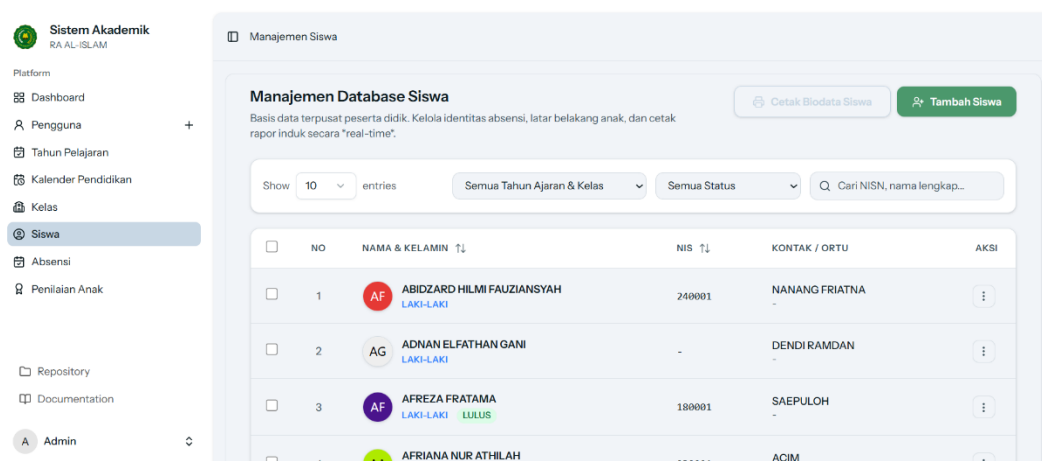
```

Gambar 5.83 Code Frontend Siswa (Kelola siswa)

Potongan kode pada Gambar 5.83 menunjukkan antarmuka *Frontend* Pengelolaan Siswa menggunakan kerangka kerja React JS. Mengingat basis data siswa memiliki

volume rekaman (baris) yang masif, antarmuka klien mengimplementasikan komponen *Dynamic Grouped Filtering* (Penyaringan Terkelompok Dinamis). Berbeda dengan kotak pilihan (*dropdown*) tunggal konvensional, elemen UI ini menyatukan dua parameter filter (Class dan School Year) dengan memanfaatkan markah HTML `<optgroup>`. Saat pengguna berinteraksi dengan elemen tersebut, fungsi reaktif *handleCombinedFilterChange* akan membedah nilai referensi (*prefix parse*), lalu mengirimkan state perubahan secara asinkron (*asynchronous state dispatch*) ke *Backend* via modul *Inertia*. Metode *Client-Side Routing* ini memungkinkan tabel siswa merender ulang baris data secara instan (*rehydration*) tanpa perlu mengalami proses siklus muat ulang halaman secara penuh (*full page reload*).

5. Tampilan halaman siswa (kelola siswa)



Gambar 5.84 Halaman Siswa (Kelola siswa)

Gambar 5.84 menunjukkan hasil implementasi halaman Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan daftar *database* peserta didik terpusat yang dikemas dalam bentuk tabel interaktif yang memuat informasi nama, jenis kelamin, NIS, serta data kontak orang tua. Selain itu, tersedia tombol Tambah Siswa, Cetak Biodata Siswa, pilihan filter tahun ajaran/kelas, serta tombol aksi untuk melakukan pengelolaan data siswa sesuai dengan kebutuhan administrasi sekolah.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Admin dalam mengelola data siswa secara lebih efektif. Selain itu, sistem juga menyediakan informasi status aktif siswa serta filter data yang dinamis sehingga dapat digunakan sebagai acuan valid dalam proses penataan akademik yang berlangsung. Dengan demikian, halaman Siswa tidak hanya berfungsi sebagai media penyimpanan data digital, tetapi juga membantu menjaga konsistensi integrasi informasi peserta didik pada seluruh fitur transaksional lainnya di dalam sistem.

The screenshot shows a web application interface for 'Manajemen Siswa' (Student Management) with a 'Tambah Baru' (Add New) button. The main form is titled 'Data Siswa' and contains the following fields:

- NIS**: Text input field.
- NISN**: Text input field.
- Nama Lengkap**: Text input field.
- Nama Panggilan**: Text input field.
- Jenis Kelamin**: Dropdown menu with 'Pilih Jenis Kelamin'.
- Tempat Lahir**: Text input field.
- Tanggal Lahir**: Date picker with 'dd/mm/yyyy' format.
- Agama**: Dropdown menu with 'Pilih Agama'.
- Status Keluarga**: Dropdown menu with 'Pilih Status Keluarga'.
- Anak Ke**: Text input field.
- No. Telp / HP Siswa**: Text input field.

At the bottom of the form, there is a label 'Alamat Lengkap Siswa'.

Gambar 5.85 Halaman Tambah Siswa

Gambar 5.85 menampilkan antarmuka formulir penambahan data siswa baru yang dirancang secara komprehensif untuk mendokumentasikan identitas pribadi, informasi kependudukan, serta latar belakang keluarga peserta didik. Secara analitis, struktur masukan berkolom ganda ini dirancang untuk memisahkan bidang data pokok seperti NIS, NISN, nama lengkap, tempat tanggal lahir, hingga kontak secara sistematis guna meminimalkan kesalahan penginputan data master. Di balik layar, *backend* Laravel mengeksekusi validasi kueri untuk memastikan keunikan NIS/NISN (*unique constraint*) sekaligus memicu fitur *Automated Account Provisioning*; apabila entitas orang tua belum ditautkan, sistem secara mandiri menggenerasi akun wali murid baru ber-role *parent* dan menghubungkannya via *foreign key* `user_id`. Pendekatan rekayasa ini tidak hanya menyederhanakan alur

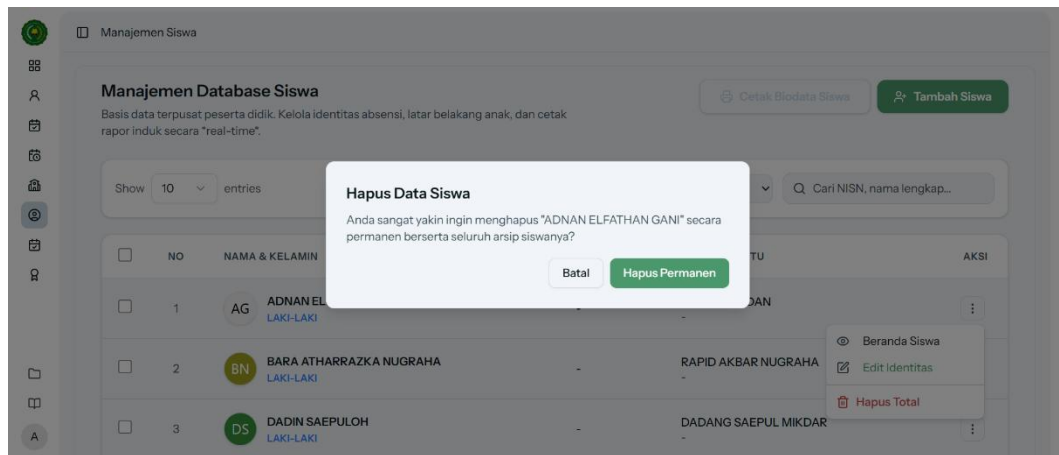
pendataan bagi administrator, tetapi juga menjamin terciptanya relasi pangkalan data yang utuh dan terintegrasi sejak awal pendaftaran murid.

The screenshot shows a web application interface for editing student data. The page title is 'Manajemen Siswa > Edit Siswa'. The main form is titled 'Data Siswa' and contains the following fields:

- NIS (Nomor Induk Siswa): Text input field.
- NISN: Text input field.
- Nama Lengkap: Text input field with the value 'ADNAN ELFATHAN GANI'.
- Nama Panggilan: Text input field.
- Jenis Kelamin: Dropdown menu with 'Laki-laki' selected.
- Tempat Lahir: Text input field with the value 'CIAMIS'.
- Tanggal Lahir: Text input field with the value '09/01/2022' and a calendar icon.
- Agama: Dropdown menu with 'Pilih Agama' selected.
- Status Keluarga: Dropdown menu with 'Pilih Status Keluarga' selected.
- Anak Ke: Text input field.
- No. Telp / HP Siswa: Text input field.
- Alamat Lengkap Siswa: Label at the bottom of the form.

Gambar 5.86 Halaman Edit Siswa

Gambar 5.86 memperlihatkan antarmuka pembaruan (*Update*) data identitas siswa yang secara otomatis memuat informasi lama (*pre-filled form*) berdasarkan parameter ID peserta didik yang dikirimkan melalui *route*. Ditinjau dari aspek rekayasa perangkat lunak, formulir ini memfasilitasi penyesuaian data diri maupun pembaruan status akademik secara fleksibel tanpa merusak keterkaitan relasional terhadap tabel riwayat presensi (*daily_attendances*) dan rekapitulasi nilai RDM. Di sisi *backend*, proses pembaruan dilindungi oleh otorisasi bertingkat dan proteksi *Insecure Direct Object Reference (IDOR)* untuk memastikan bahwa modifikasi data hanya dapat dieksekusi oleh administrator yang sah. Melalui pemrosesan kueri yang terisolasi dan validasi *server-side* yang ketat, pemeliharaan data induk murid dapat berjalan secara adaptif, akurat, serta tetap menjaga konsistensi pangkalan data utama sekolah.



Gambar 5.87 Halaman Hapus Siswa

Gambar 5.87 mengilustrasikan mekanisme penghapusan data (*Delete*) siswa menggunakan komponen interaktif berupa *Modal Overlay Confirmation* dengan opsi tombol *Batal* dan *Hapus Permanen*. Secara analitis, penerapan dialog konfirmasi ini mengadopsi prinsip *Fail-Safe Design* untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) akibat penekanan tombol aksi yang tidak sengaja. Mengingat entitas siswa memiliki keterikatan kunci tamu (*foreign key*) yang sangat krusial terhadap tabel presensi harian, riwayat akademik, dan akun wali murid, tindakan penghapusan data bersifat sangat sensitif karena berpotensi memicu penghapusan berantai (*cascade deletion*) atau anomali data. Oleh karena itu, ketersediaan peringatan eksplisit mengenai pembersihan arsip siswa ini memaksa administrator memverifikasi keputusan secara sadar, sehingga seluruh manipulasi data destruktif dapat berjalan secara aman, terukur, dan akuntabel.

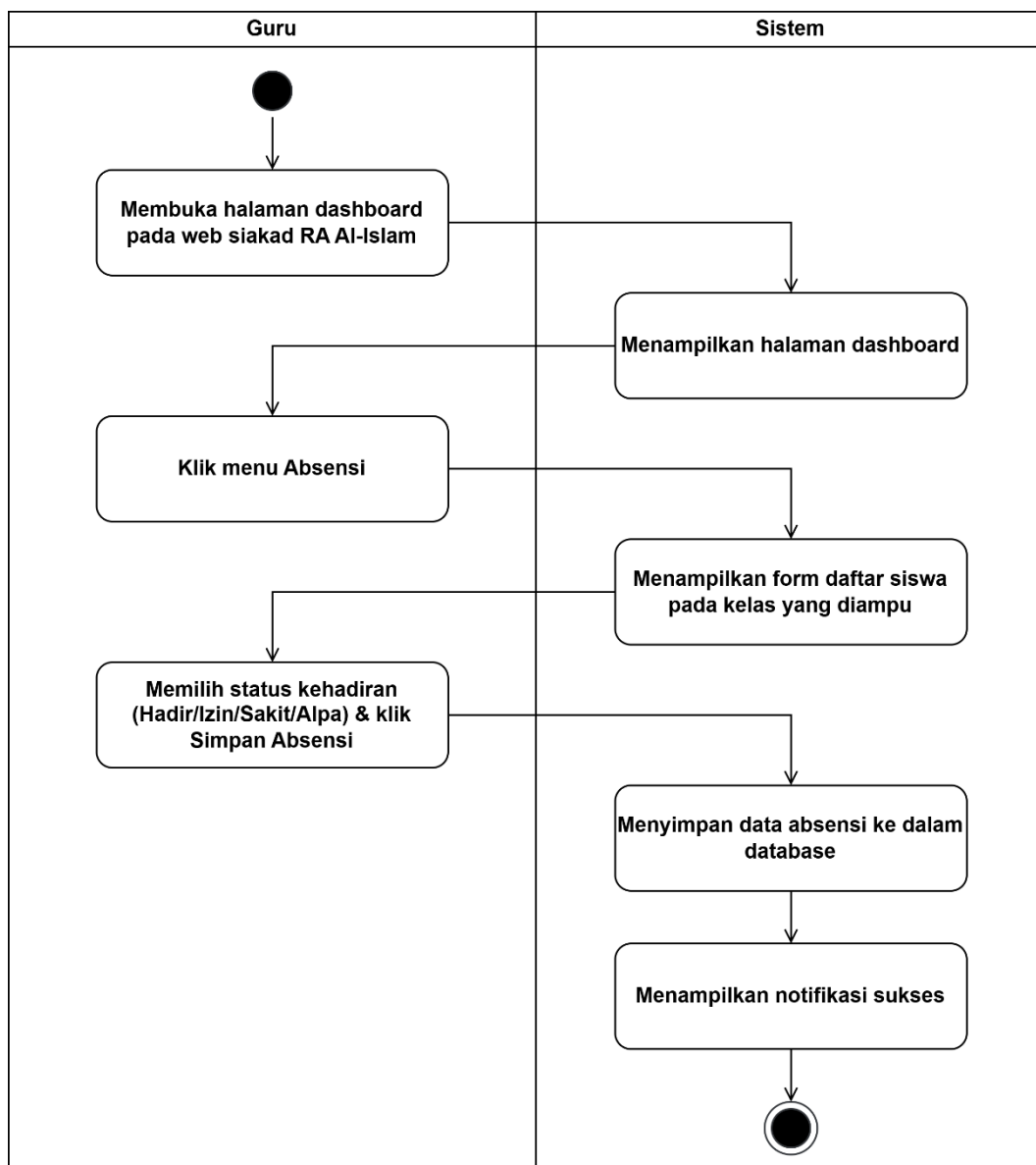
5.4.2.2 Absensi (Input guru)

Fitur Absensi (Input Guru) merupakan salah satu fitur utama yang dikembangkan pada *Sprint 2* untuk mendukung proses pencatatan kehadiran siswa secara digital pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Guru dapat melakukan pencatatan kehadiran siswa berdasarkan kelas dan tanggal pembelajaran yang sedang berlangsung. Data kehadiran yang dikelola meliputi status hadir, izin, sakit, maupun alpa sehingga informasi kehadiran siswa dapat

terdokumentasi secara lebih sistematis. Dengan adanya fitur ini, proses pencatatan absensi tidak lagi bergantung pada media tulis sehingga risiko kehilangan data maupun kesalahan pencatatan dapat diminimalkan, sekaligus mempermudah proses pemantauan kehadiran siswa sebagai bagian dari kegiatan akademik.

Perancangan fitur Absensi (Input Guru) dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pencatatan kehadiran sebelum implementasi dilakukan. Ketiga diagram tersebut digunakan untuk menggambarkan aktivitas Guru dalam melakukan input absensi, struktur data yang digunakan untuk menyimpan informasi kehadiran siswa, serta urutan komunikasi antara pengguna dengan sistem selama proses pencatatan berlangsung. Setelah proses perancangan selesai, fitur diimplementasikan menggunakan framework Laravel 12 sebagai backend dan React JS sebagai frontend sehingga proses pencatatan absensi dapat dilakukan secara terintegrasi, menghasilkan data yang lebih akurat, serta mendukung penyajian informasi kehadiran yang dibutuhkan pada fitur-fitur akademik lainnya.

1. *Activity diagram* absensi (input guru)



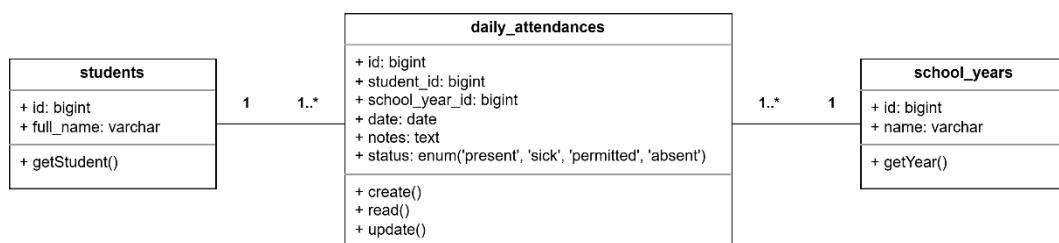
Gambar 5.88 *Activity Diagram* Absensi (Input guru)

Activity Diagram Absensi (Input Guru) pada Gambar 5.88 menggambarkan alur proses pengelolaan presensi harian siswa yang dilakukan oleh Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Guru membuka Dashboard kemudian memilih menu Absensi sehingga sistem menampilkan halaman data absensi yang menyediakan opsi filter tahun ajaran,

kelas, dan tanggal akademik. Selanjutnya Guru dapat menentukan status kehadiran masing-masing siswa dan menekan tombol simpan untuk mengirim data ke sistem.

Apabila Guru menyimpan data absensi tersebut, sistem akan memproses permintaan dan melakukan validasi kelayakan input data ke dalam basis data. Sementara itu, jika terdapat data yang tidak sesuai atau belum terisi lengkap, sistem akan memberikan peringatan agar Guru melengkapi pemilihan status kehadiran sebelum proses penyimpanan dieksekusi. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pencatatan presensi harian dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* absensi (input guru)



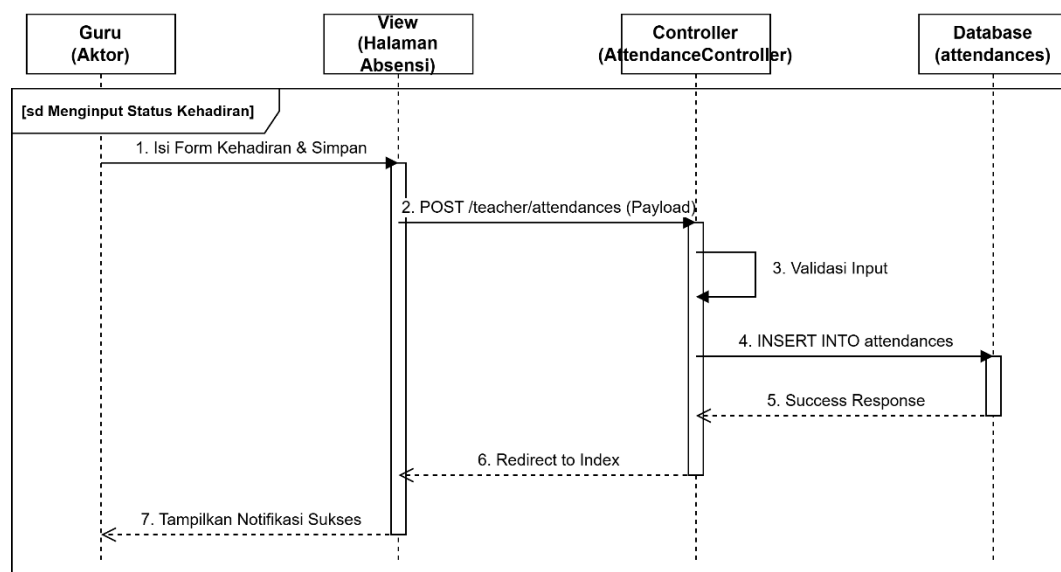
Gambar 5.89 *Class Diagram* Absensi (Input guru)

Class Diagram Absensi (input guru) pada Gambar 5.89 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses pengelolaan data kehadiran harian siswa pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas utama *daily_attendances* yang memiliki atribut *id*, *student_id*, *school_year_id*, *date*, *notes*, dan *status* berjenis data *enum*('present', 'sick', 'permitted', 'absent') beserta operasi *create()*, *read()*, dan *update()*, dengan kelas *students* yang memuat atribut *id*, *full_name* serta operasi *getStudent()*, dan kelas *school_years* dengan atribut *id*, *name* serta operasi *getYear()*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengorganisasian data presensi harian dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* di mana entitas *students* dan *school_years*

masing-masing memiliki relasi satu ke banyak (1 ke 1..*) terhadap tabel transaksional *daily_attendances* sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan struktural antar-tabel melalui pemetaan kunci tamu (*student_id*, *school_year_id*) sehingga setiap proses penambahan, pembacaan, maupun pembaruan data absensi dapat dilakukan secara konsisten sesuai dengan kebutuhan integritas basis data pada sistem.

3. *Sequence diagram* absensi (input guru)



Gambar 5.90 *Sequence Diagram* Absensi (Input guru)

Sequence Diagram Absensi (Input Guru) pada Gambar 5.90 menggambarkan urutan komunikasi antara Guru, antarmuka sistem, *controller*, dan basis data selama proses menginput status kehadiran berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menyimpan data presensi dikirimkan oleh Guru dari *View* (Halaman Absensi), kemudian diproses oleh *AttendanceController* melalui metode *HTTP POST* dengan rute */teacher/attendances* untuk menjalankan fungsi internal validasi input sebelum diteruskan ke Database (tabel *attendances*) menggunakan kueri *SQL INSERT INTO attendances* hingga menghasilkan respons sukses yang akan dikembalikan kepada komponen antarmuka.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses penginputan data kehadiran harian peserta didik dapat dipahami secara lebih rinci dan berurutan. Diagram ini menjadi acuan dalam implementasi logika sistem sehingga proses pertukaran data antara antarmuka, *controller*, dan basis data dapat berjalan secara terstruktur, memicu pengalihan halaman (*redirect to index*), menjaga konsistensi data, serta menampilkan notifikasi keberhasilan sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi absensi (input guru)

```

1      // 1. Simpan Absensi Harian (Tabel: daily_attendances)
2      foreach ($request->attendances as $row) {
3          \App\Models\DailyAttendance::updateOrCreate(
4              ['student_id' => $row['student_id'], 'date' => $date],
5              ['school_year_id' => $schoolYearId, 'status' => $row['status']]
6          );
7
8      // 2. Aggregate Calculation & Denormalization
9      // Hitung total dari sebulan tersebut menggunakan Raw SQL Aggregate
10     $monthlyData = \App\Models\DailyAttendance::selectRaw('
11         COUNT(CASE WHEN status = "present" THEN 1 END) as present,
12         COUNT(CASE WHEN status = "sick" THEN 1 END) as sick,
13         COUNT(CASE WHEN status = "permitted" THEN 1 END) as permitted,
14         COUNT(CASE WHEN status = "absent" THEN 1 END) as absent
15     ')
16     ->where('student_id', $row['student_id'])
17     ->whereMonth('date', $month)
18     ->whereYear('date', date('Y', strtotime($date)))
19     ->first();
20
21     // 3. Simpan otomatis ke tabel rekap bulanan (Tabel: attendances)
22     \App\Models\Attendance::updateOrCreate(
23         ['student_id' => $row['student_id'], 'school_year_id' =>
24             $schoolYearId, 'month' => $month],
25         [
26             'present' => $monthlyData->present ?? 0,
27             'sick' => $monthlyData->sick ?? 0,
28             'permitted' => $monthlyData->permitted ?? 0,
29             'absent' => $monthlyData->absent ?? 0,
30         ]
31     );

```

Gambar 5.91 *Code Backend* Absensi (Input guru)

Potongan kode pada Gambar 5.91 menunjukkan metode *store()* di dalam *AttendanceController* khusus Guru. Saat guru menyimpan absen harian, sistem tidak hanya sekadar merekam data ke tabel *daily_attendances*. Sistem

mengimplementasikan teknik Data *Denormalization* (Denormalisasi Data) untuk mendongkrak performa aplikasi (*read performance*). Setelah data harian disimpan, *Backend* secara otomatis mengeksekusi *Aggregate Query* menggunakan *SQL* murni (*COUNT(CASE WHEN...)*) untuk menghitung total kehadiran (Hadir, Sakit, Izin, Alpa) siswa pada bulan berjalan. Hasil perhitungan tersebut kemudian di-*(updateOrCreate)* ke tabel rekap bulanan (*attendances*). Sistem tidak perlu lagi menjumlahkan ribuan baris data absen harian dari awal, melainkan cukup membaca satu baris data rekap yang sudah matang.

```

1  // State form berbasis Object Record (Map student_id ke status absensi)
2  const [formData, setFormData] = useState<Record<number, { status: string; notes?:
   string }>>({});
3
4  // Fungsi pembaruan state reaktif saat Radio Button diklik
5  const handleStatusChange = (studentId: number, status: string) => {
6    setFormData((prev) => ({
7      ...prev,
8      [studentId]: { ...prev[studentId], status },
9    }));
10 };
11
12 const handleSave = () => {
13   setErrorMessage(null);
14
15   // Algoritma Validasi Integritas Data (Data Integrity Validation)
16   // Menyisir jika ada siswa yang state 'status'-nya masih kosong
17   const unselectedStudents = students.filter(student =>
   !formData[student.id]?.status);
18
19   // Memblokir pengiriman ke server (API) jika data tidak lengkap
20   if (unselectedStudents.length > 0) {
21     setErrorMessage(`Harap isi status kehadiran untuk semua siswa! (Ada
   ${unselectedStudents.length} siswa yang belum diisi)`);
22     return;
23   }
24
25   // Mapping ulang struktur object agar sesuai dengan format array API Laravel
26   const attendancesToSave = Object.entries(formData).map(([studentId, data]) =>
   ({
27     student_id: parseInt(studentId),
28     status: data.status,
29     notes: data.notes,
30   }));
31
32   // Melakukan Dispatch Request ke Backend
33   router.post(route('teacher.attendances.store'), {
34     school_year_id: filterSchoolYearId,
35     date: filterDate,
36     attendances: attendancesToSave,
37   });
38 };

```

Gambar 5.92 Code Frontend Absensi (Input guru)

Potongan kode pada Gambar 5.92 menampilkan antarmuka *Frontend* Pengisian Absensi menggunakan React JS. Karena absen harus dipastikan akurat, sistem mengimplementasikan *Strict Client-Side Validation* (Validasi Ketat di Sisi Klien). Sistem memetakan status kehadiran setiap siswa (melalui *Radio Button*) ke dalam sebuah *Controlled State React* berstruktur *Object Record* (*Record<number, {status: string}>*). Sebelum data dikirim ke server, sistem (melalui fungsi *handleSave*) akan menyisir struktur data tersebut. Apabila terdeteksi ada satu saja siswa yang status kehadirannya lupa diisi oleh guru, sistem akan langsung mengintersepsi pengiriman ((*block submission*)) dan memunculkan notifikasi *error* secara *real-time*. Hal ini memastikan integritas data (*data integrity*), di mana mustahil ada data siswa yang "bolong" atau berstatus *null* pada hari tersebut.

5. Tampilan halaman absensi (input guru)

No	NIS	Nama Lengkap	L/P	Atur Kehadiran
1	-	ADNAN ELFATHAN GANI	L	☐ Hadir ☐ Sakit ☐ Izin ☐ Alpa
2	-	BARA ATHARRAZKA NUGRAHA	L	☐ Hadir ☐ Sakit ☐ Izin ☐ Alpa
3	-	DADIN SAEPULOH	L	☐ Hadir ☐ Sakit ☐ Izin ☐ Alpa

Gambar 5.93 Halaman Absensi (Input guru)

Gambar 5.93 menunjukkan hasil implementasi halaman Absensi pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan menu pencatatan presensi harian siswa yang dikemas dalam bentuk tabel data terstruktur yang memuat kolom nomor, NISN, nama lengkap, jenis kelamin (L/P), serta panel tombol opsi pilihan penentuan status kehadiran (Hadir, Sakit, Izin, Alpa) untuk masing-masing peserta didik. Selain itu, tersedia filter pilihan tahun ajaran/kelas, penentuan tanggal absensi, kolom pencarian nama siswa, serta tombol utama

berwarna hijau "Simpan Data Absensi" sebagai media bagi Guru untuk melakukan perekaman kehadiran secara waktu nyata.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Guru dalam mengelola pencatatan presensi harian secara lebih efektif tanpa memerlukan alur penggunaan yang rumit. Selain itu, penataan layout kolom yang bersih serta penempatan tombol navigasi aksi yang jelas sangat membantu meminimalkan potensi kesalahan penginputan data kehadiran harian peserta didik. Dengan demikian, halaman Absensi tidak hanya berfungsi sebagai media pencatatan presensi digital, tetapi juga membantu menjaga konsistensi keandalan rekapitulasi data kehadiran siswa pada seluruh fitur transaksional lainnya di dalam sistem.

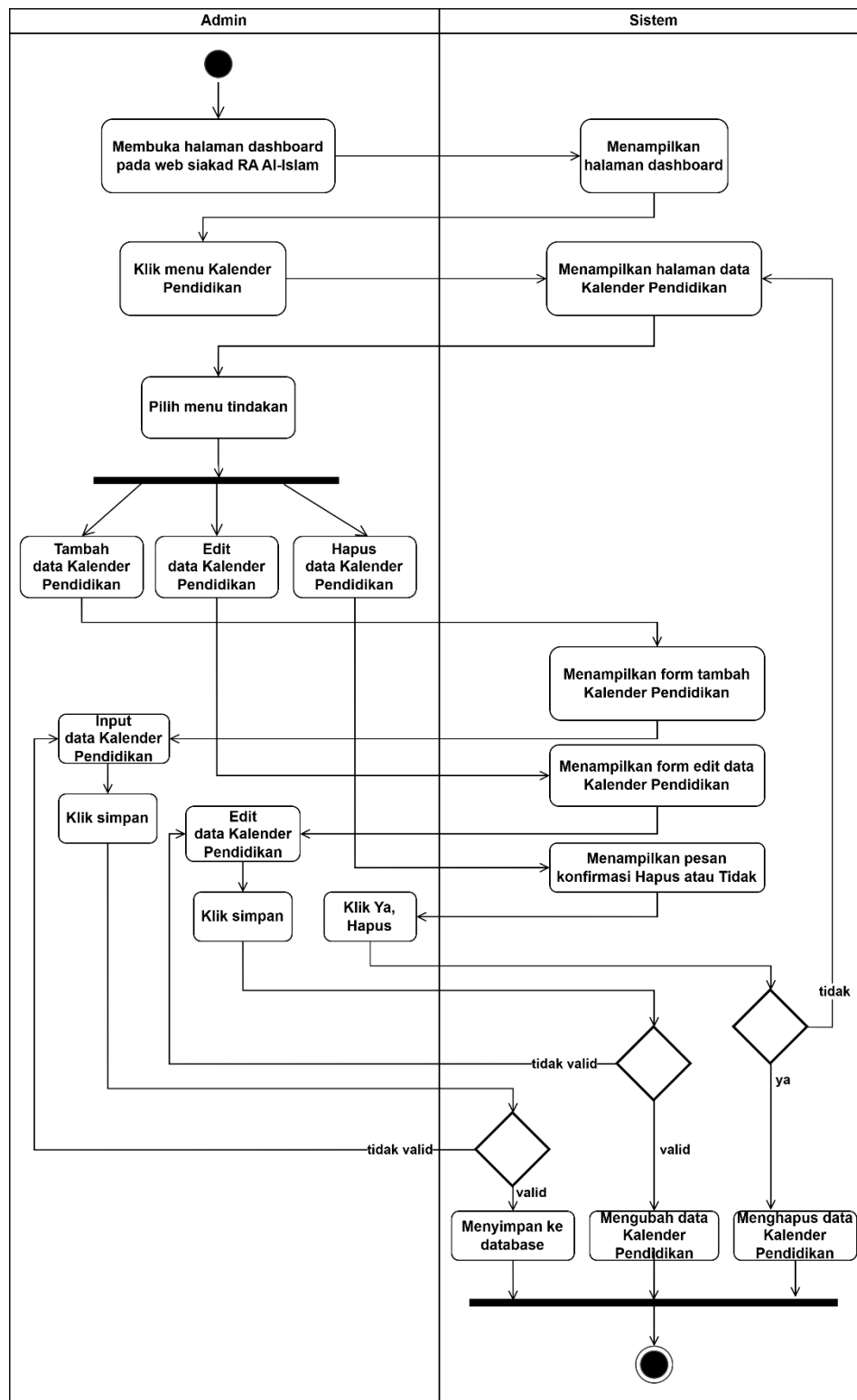
5.4.2.3 Kalender pendidikan (mengelola)

Fitur Kalender Pendidikan (Mengelola) merupakan salah satu fitur yang dikembangkan pada *Sprint 2* untuk mendukung pengelolaan jadwal kegiatan akademik pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Berbeda dengan fitur melihat kalender yang dikembangkan pada *Sprint 1*, pada tahap ini Admin diberikan hak akses untuk menambahkan, mengubah, maupun menghapus informasi kegiatan akademik yang akan ditampilkan kepada Guru dan Orang Tua Siswa. Informasi yang dikelola meliputi nama kegiatan, tanggal pelaksanaan, serta keterangan yang berkaitan dengan aktivitas akademik sekolah. Dengan adanya fitur ini, penyampaian informasi mengenai agenda sekolah dapat dilakukan secara lebih terstruktur, mudah diperbarui, dan dapat diakses oleh seluruh pengguna sesuai dengan hak akses yang dimiliki.

Perancangan fitur Kalender Pendidikan (Mengelola) dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pengelolaan data sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas Admin dalam mengelola data kalender pendidikan, struktur data yang digunakan untuk menyimpan informasi kegiatan akademik, serta urutan komunikasi antara pengguna dengan

sistem selama proses pengelolaan data berlangsung. Setelah proses perancangan selesai, fitur diimplementasikan menggunakan framework Laravel 12 sebagai *backend* dan React JS sebagai *frontend* sehingga proses pengelolaan kalender pendidikan dapat berjalan secara terintegrasi, memudahkan penyampaian informasi kepada pengguna, serta mendukung kelancaran pelaksanaan kegiatan akademik di lingkungan RA Al-Islam Gunungcupu.

1. *Activity diagram* kalender pendidikan (mengelola)

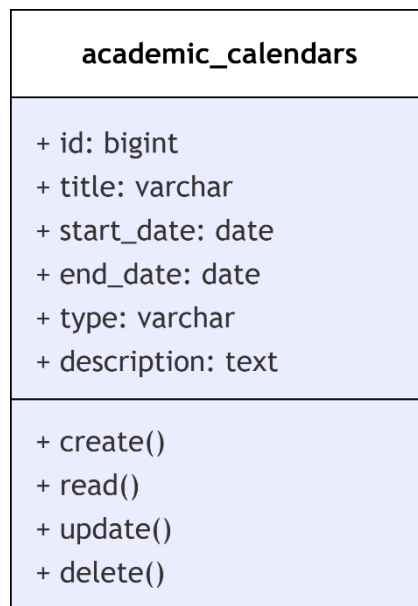


Gambar 5.94 *Activity Diagram* Kalender Pendidikan (Mengelola)

Activity Diagram Kalender Pendidikan (Mengelola) pada Gambar 5.94 menggambarkan alur proses pengelolaan agenda kegiatan akademik yang dilakukan oleh Admin pada Sistem Informasi Akademik RAAI-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Kalender Pendidikan sehingga sistem menampilkan halaman data kalender pendidikan yang berisi seluruh data agenda yang telah tersimpan pada basis data. Selanjutnya Admin dapat memilih tindakan berupa menambah data, mengubah data, maupun menghapus data agenda kegiatan sesuai dengan kebutuhan pengelolaan manajemen sekolah.

Apabila Admin memilih tambah atau ubah data kalender pendidikan, sistem akan menampilkan formulir yang harus diisi kemudian melakukan proses validasi terhadap data yang dimasukkan sebelum menyimpannya ke dalam basis data. Sementara itu, apabila Admin memilih menghapus data agenda, sistem akan terlebih dahulu menampilkan konfirmasi penghapusan untuk menghindari kesalahan tindakan. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pengelolaan operasional agenda dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* kalender pendidikan (mengelola)

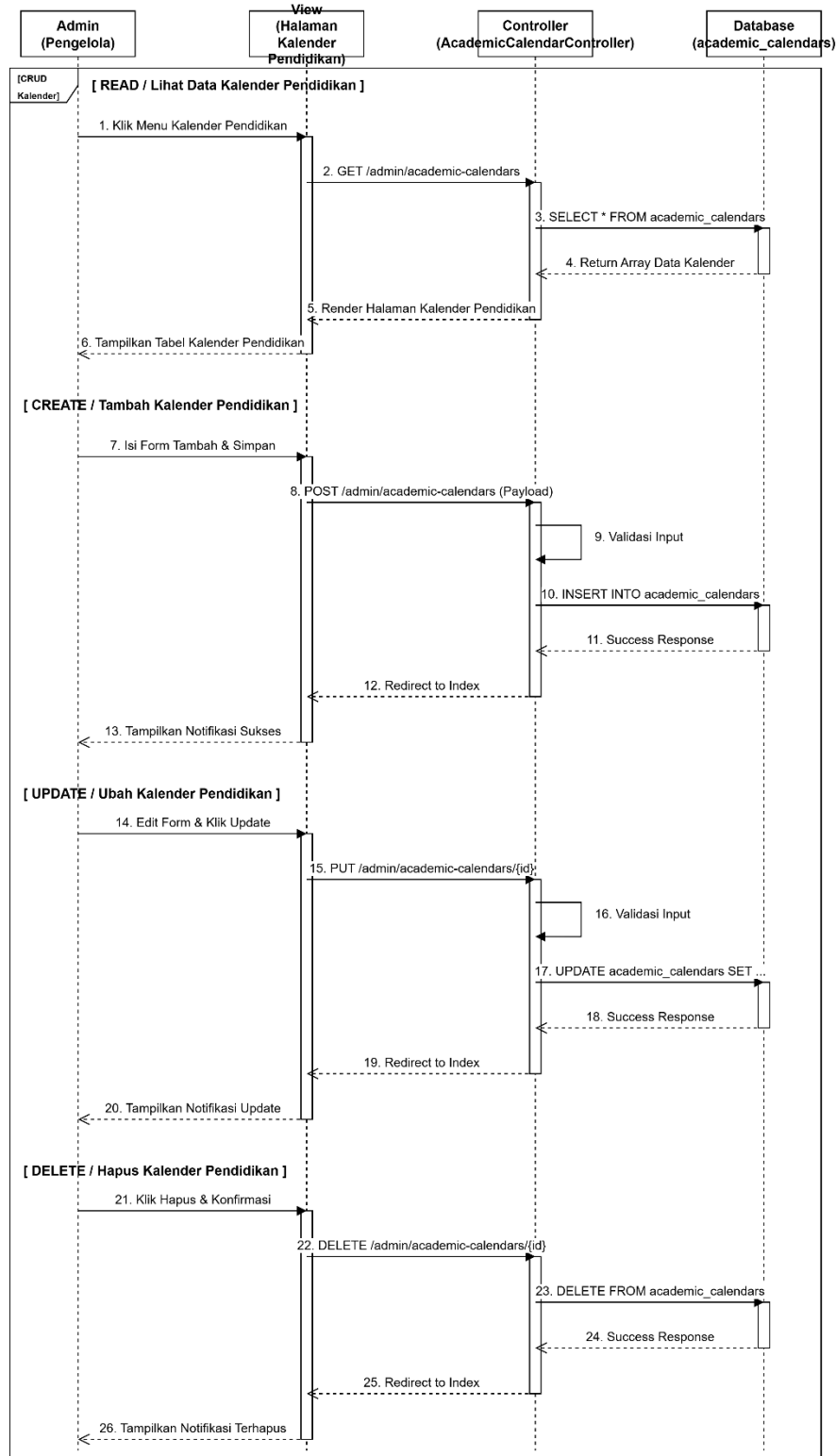


Gambar 5.95 *Class Diagram* Kalender Pendidikan (Mengelola)

Class Diagram Kalender Pendidikan (Mengelola) pada Gambar 5.95 digunakan untuk menggambarkan struktur kelas yang berperan dalam proses pengelolaan data kalender pendidikan. Diagram ini menunjukkan atribut yang dimiliki oleh kelas *academic_calendars*, yaitu *id*, *title*, *start_date*, *end_date*, *type*, dan *description*, beserta operasi yang digunakan untuk melakukan proses *Create*, *Read*, *Update*, dan *Delete* (CRUD) terhadap seluruh data kalender akademik sekolah.

Melalui *Class Diagram*, struktur data yang digunakan dalam proses pengelolaan data kalender pendidikan dapat dirancang secara lebih sistematis sehingga implementasi pada tahap pengkodean menjadi lebih terarah. Selain itu, diagram ini juga membantu dalam menggambarkan hubungan antara data agenda dengan mekanisme pengelolaan yang diterapkan pada sistem sehingga setiap proses penambahan, pengubahan, maupun penghapusan data dapat dilakukan secara konsisten sesuai dengan kebutuhan aturan sistem.

3. Sequence diagram kalender pendidikan (mengelola)



Gambar 5.96 Sequence Diagram Kalender Pendidikan (Mengelola)

Sequence Diagram Kalender Pendidikan (Mengelola) pada Gambar 5.96 menggambarkan urutan komunikasi antara Admin, antarmuka sistem, *controller*, dan basis data selama proses pengelolaan data kalender pendidikan berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data, menambahkan data baru, mengubah data, maupun menghapus data dikirimkan oleh Admin, kemudian diproses oleh *controller* (*AcademicCalendarController*) melalui proses validasi sebelum diteruskan ke *database* (tabel *academic_calendars*) hingga menghasilkan respons yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengelolaan data agenda kegiatan akademik sekolah dapat dipahami secara lebih rinci dan berurutan. Diagram ini menjadi acuan dalam implementasi logika sistem sehingga proses pertukaran data antara antarmuka, *controller*, dan basis data dapat berjalan secara terstruktur, menjaga konsistensi data, serta sesuai dengan kebutuhan operasional sistem yang telah dirancang sebelumnya.

4. Implementasi kalender pendidikan (mengelola)

```

1  /**
2   * Simpan acara baru ke kalender (Create)
3   * Mengimplementasikan Mass Assignment Protection via form validation
4   */
5  public function store(Request $request)
6  {
7      // 1. Validasi & Sanitasi Data (Menyaring payload berbahaya)
8      $validated = $request->validate([
9          'title'      => 'required|string|max:255',
10         'start_date' => 'required|date',
11         // Memastikan tanggal selesai tidak boleh mendahului tanggal mulai
12         'end_date'   => 'required|date|after_or_equal:start_date',
13         'type'       => 'required|in:holiday,event',
14         'description'=> 'nullable|string',
15     ]);
16
17     // 2. Injeksi data yang sudah aman (Validated Data) ke Database
18     AcademicCalendar::create($validated);
19
20     return redirect()->back()->with('success', 'Acara berhasil ditambahkan ke
    kalender.');
```

Gambar 5.97 *Code Backend* Kalender Pendidikan (Mengelola)

Potongan kode pada Gambar 5.97 menunjukkan struktur *AcademicCalendarController*. *Backend* dirancang sangat bersih dengan mengadopsi standar *RESTful Architecture* (*index, store, update, destroy*). Satu aspek krusial yang menonjol dari segi keamanan siber (Keamanan Perangkat Lunak) adalah penerapan *Mass Assignment Protection*. Pada metode *store()* dan *update()*, sistem tidak serta-merta menerima masukan bebas dari pengguna (*\$request->all()*). Sistem melakukan filtrasi ketat melalui *\$request->validate()* untuk memastikan bahwa hanya atribut *title, start_date, end_date, type*, dan *description* yang valid secara tipe data yang boleh berinteraksi dengan Model Basis Data. Pendekatan ini mencegah eksploitasi di mana peretas mencoba menyuntikkan kolom ilegal ke dalam *payload request*.

```

1 // 1. Client-Side Date Matrix Computation (Algoritma Penentuan Titik Kalender)
2 const monthStart = startOfMonth(currentMonth);
3 const monthEnd = endOfMonth(monthStart);
4 // Menarik mundur awal matriks ke hari Senin pertama di bulan tersebut
5 const startDate = startOfWeek(monthStart, { weekStartsOn: 1 });
6 const endDate = endOfWeek(monthEnd, { weekStartsOn: 1 });
7
8 // Menghasilkan array of dates (sel matriks kalender)
9 const days = eachDayOfInterval({ start: startDate, end: endDate });
10
11 // 2. Fungsi Komputasi Pencocokan Acara (Event Intersection)
12 const getDayEvents = (day: Date) => {
13   return events.filter((event) => {
14     const start = parseISO(event.start_date.substring(0, 10));
15     const end = parseISO(event.end_date.substring(0, 10));
16     // Mengembalikan acara jika tanggal yang sedang dirender berada di dalam
    rentang acara
17     return (
18       isWithinInterval(day, { start, end }) || isSameDay(day, start)
19     );
20   });
21 };
22
23 return (
24   /* Render Matriks Kalender (Grid) */
25   <div className="grid grid-cols-7 gap-2">
26     {days.map((day, idx) => {
27       const dayEvents = getDayEvents(day);
28       const isCurrentMonth = isSameMonth(day, currentMonth);
29       const isToday = isSameDay(day, new Date());
30
31       return (
32         <div
33           key={idx}
34           // Jika Admin (isAdmin === true), sel tanggal bisa diklik
    untuk memicu form tambah acara
35           className={`... ${isAdmin ? 'cursor-pointer' : ''}`}
36           onClick={() => isAdmin && handleOpenModal(undefined, day)}
37         >
38           <span className="text-xs font-bold">{format(day, 'd')}</span>
39
40           /* Rendering Event Badges (Label Acara) */
41           <div className="flex flex-col gap-1">
42             {dayEvents.slice(0, 2).map((e, index) => (
43               <div key={index} className="truncate rounded-md px-1
py-0.5 text-[9px]">
44                 {e.title}
45               </div>
46             ))}
47           </div>
48         </div>
49       );
50     })}
51   </div>
52 );

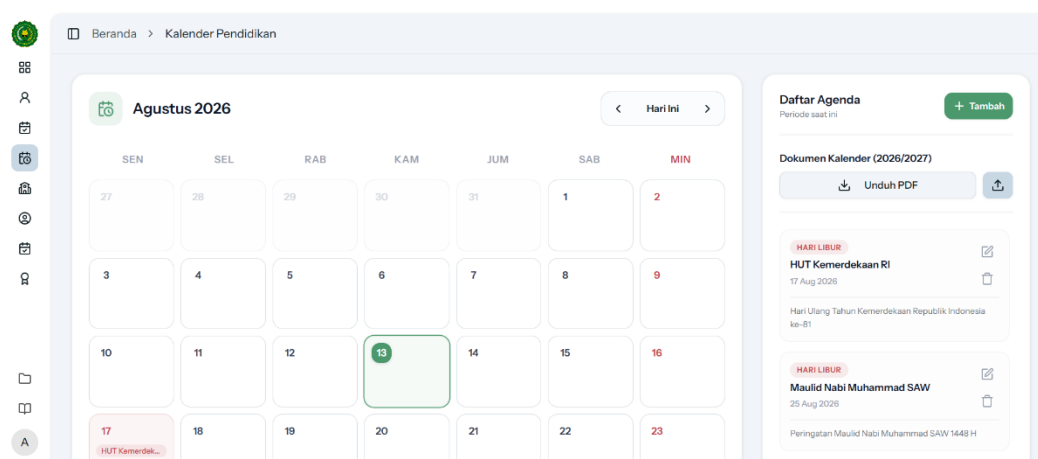
```

Gambar 5.98 *Code Frontend* Kalender Pendidikan (Mengelola)

Potongan kode pada Gambar 5.98 menunjukkan rekayasa tingkat lanjut di sisi klien (*Frontend*). Alih-alih membebani *server backend* untuk menyusun matriks tanggal

kalender (hari Senin s.d. Minggu beserta posisi acaranya), sistem mendelegasikan tugas komputasi komputasi tersebut ke browser pengguna (*Client-Side Rendering / CSR*). Dengan memanfaatkan pustaka utilitas waktu *date-fns*, komponen *React* mengekstrak batas awal bulan (*startOfMonth*) dan akhir pekan (*endOfWeek*) untuk menghasilkan rentang *interval array* harian (*eachDayOfInterval*). Selanjutnya, algoritma *getDayEvents()* melakukan pencocokan (*intersection*) antara matriks tanggal dengan daftar acara (*events*) untuk merender elemen *badge* acara secara dinamis tepat di atas sel tanggal yang bersesuaian.

5. Tampilan halaman kalender pendidikan (mengelola)



Gambar 5.99 Halaman Kalender Pendidikan (Mengelola)

Gambar 5.99 menunjukkan hasil implementasi halaman Kalender Pendidikan (Mengelola) pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan visualisasi utama berupa grid penanggalan bulanan yang interaktif serta panel "Daftar Agenda" di sisi kanan yang dilengkapi tombol operasional hijau "+ Tambah". Selain itu, tersedia menu navigasi manajemen di bilah sisi kiri, fasilitas filter navigasi penanggalan, serta tombol ikon edit dan hapus pada baris daftar kegiatan sebagai media bagi Admin untuk melakukan pengelolaan operasional jadwal sesuai dengan kebutuhan sekolah.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Admin dalam mengelola pembagian waktu kegiatan akademik secara lebih efektif.

Selain itu, sistem juga menyajikan ringkasan rentang tanggal pelaksanaan agenda secara jelas untuk mempermudah pengawasan jadwal penting sekolah yang sedang berjalan. Dengan demikian, halaman Kalender Pendidikan tidak hanya berfungsi sebagai media manipulasi data administratif, tetapi juga membantu menjaga konsistensi sinkronisasi penggunaan jadwal pada seluruh modul fitur yang berkaitan dengan kegiatan akademik di dalam sistem.

Gambar 5.100 Halaman Tambah Kalender Pendidikan

Gambar 5.100 menampilkan antarmuka modal penambahan agenda kegiatan sekolah baru yang dirancang menggunakan komponen penampil melayang (*modal popup*) berbasis *Single Page Application* (SPA) untuk menjaga kontinuitas visual tanpa memicu pemuatan ulang halaman secara penuh. Secara analitis, formulir ini menyediakan bidang masukan Judul Agenda, rentang Tanggal Mulai dan Tanggal Selesai, Tipe Agenda (kegiatan akademik atau hari libur), serta Deskripsi opsional. Di balik layar, *backend* Laravel mengeksekusi aturan validasi ketat (*strict validation rule*) `after_or_equal:start_date` untuk menjamin konsistensi kronologis; sistem akan menolak penyimpanan jika tanggal selesai diset mendahului tanggal mulai. Selain itu, seluruh masukan disaring melalui *validated payload* untuk mencegah eksploitasi *mass assignment*, sehingga pembuatan agenda kegiatan sekolah dapat dilakukan secara efisien, aman, dan menjaga keandalan kalender pendidikan akademik.

Edit Agenda

Isi informasi agenda kalender pendidikan dengan lengkap.

Judul Agenda *

HUT Kemerdekaan RI

Tanggal Mulai * **Tanggal Selesai ***

17/08/2026 17/08/2026

Tipe Agenda *

Hari Libur Resmi

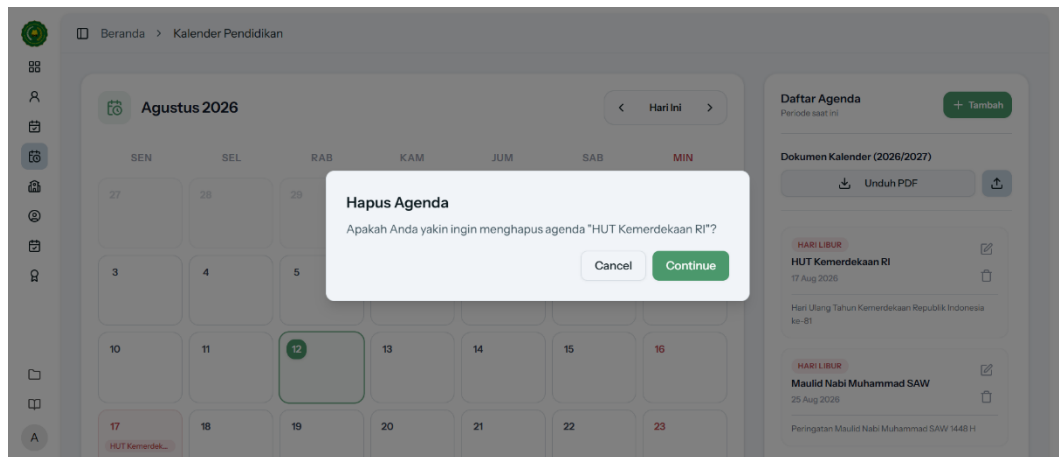
Deskripsi (Optional)

Hari Ulang Tahun Kemerdekaan Republik Indonesia ke-81

Batal Simpan

Gambar 5.101 Halaman Edit Kalender Pendidikan

Gambar 5.101 memperlihatkan antarmuka modal pembaruan (*Update*) data agenda kalender pendidikan yang secara otomatis menyajikan data lama (*pre-filled form*) berdasarkan entitas acara yang dipilih administrator pada panel utama. Ditinjau dari aspek rekayasa perangkat lunak dan *User Experience (UX)*, penggunaan komponen *modal* interaktif ini memfasilitasi penyesuaian judul, perubahan rentang tanggal, maupun pengalihan kategori agenda secara *real-time* tanpa mengganggu kestabilan antarmuka utama. Di sisi *frontend*, pembaruan data acara ini ditangkap oleh algoritma reaktif React JS (memanfaatkan pustaka *date-fns*) untuk me-*render* ulang sel tanggal dan *badge* warna kegiatan secara presisi di atas matriks kalender bulanan. Melalui alur pemrosesan data yang terisolasi dan validasi *server-side* yang adaptif, pemeliharaan jadwal kegiatan sekolah dapat dieksekusi secara dinamis, akurat, serta tetap menjaga konsistensi pangkalan data utama.



Gambar 5.102 Halaman Hapus Kalender Pendidikan

Gambar 5.102 mengilustrasikan mekanisme konfirmasi penghapusan (*Delete*) data agenda kegiatan sekolah menggunakan komponen interaktif berupa *Modal Overlay Confirmation*. Secara analitis, penerapan dialog peringatan eksplisit ini mengadopsi prinsip *Fail-Safe Design* untuk mengintersepsi tindakan destruktif dan meminimalkan risiko kesalahan manusia (*human error*) akibat penekanan tombol hapus yang tidak disengaja. Mengingat entitas agenda kalender pendidikan berfungsi sebagai acuan waktu operasional bersama yang diakses langsung oleh akun Guru dan Orang Tua Siswa, penghapusan suatu jadwal kegiatan atau hari libur secara permanen memerlukan verifikasi tingkat tinggi. Oleh karena itu, ketersediaan jeda konfirmasi dengan pilihan tombol *Cancel* dan *Continue* ini memaksa administrator untuk memverifikasi ulang keputusannya secara sadar, sehingga seluruh proses manipulasi pangkalan data berisiko tinggi dapat berjalan secara aman, terukur, dan akuntabel.

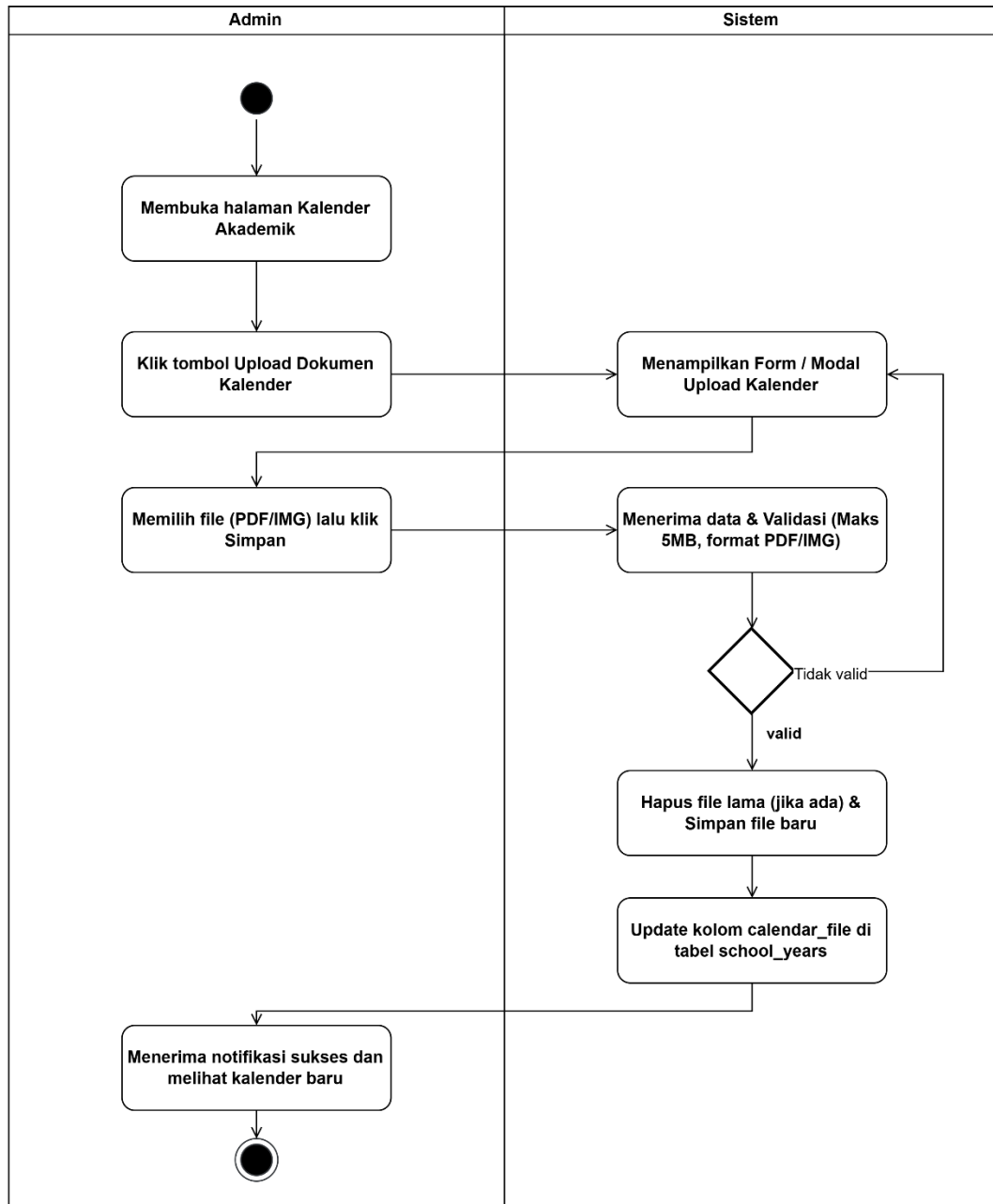
5.4.2.4 Kalender pendidikan (*upload* PDF dokumen kalender pendidikan)

Fitur Kalender Pendidikan (*Upload* Dokumen PDF) merupakan salah satu fitur yang dikembangkan pada Sprint 2 untuk mendukung pengelolaan data dan penyampaian agenda sekolah pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Melalui fitur ini, Admin dapat melakukan pengunggahan, pembaruan, penghapusan, serta penyajian berkas dokumen kalender pendidikan berformat PDF yang dapat diakses dan diunduh oleh Guru maupun Orang Tua

Siswa. Pengelolaan dokumen kalender pendidikan dilakukan secara terpusat sehingga informasi agenda akademik selama satu tahun pelajaran dapat tersimpan dengan lebih rapi, mudah diperbarui, dan dijadikan acuan operasional bersama. Dengan adanya fitur ini, proses penyampaian informasi kegiatan sekolah menjadi lebih efektif dibandingkan pengedaran lembaran fisik secara manual karena seluruh pihak dapat mengunduh dokumen resmi secara langsung melalui sistem.

Perancangan fitur Upload Kalender Pendidikan dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pengelolaan berkas sebelum implementasi dilakukan. Ketiga diagram tersebut membantu menggambarkan aktivitas Admin selama mengunggah dokumen kalender, struktur data serta penanganan penyimpanan berkas (*file storage*) pada sistem, serta urutan komunikasi antara pengguna dengan sistem ketika menjalankan proses pengunggahan, pemrosesan, dan pembacaan berkas. Dengan adanya perancangan tersebut, proses implementasi dapat dilakukan secara lebih terarah sehingga fungsi pengelolaan dokumen kalender pendidikan dapat berjalan secara stabil, aman, dan sesuai dengan kebutuhan sistem yang telah direncanakan.

1. *Activity diagram* kalender pendidikan (*upload* PDF dokumen kalender pendidikan)



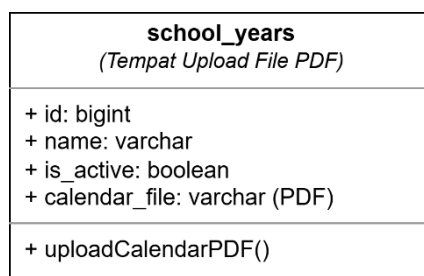
Gambar 5.103 *Activity Diagram Upload* PDF Dokumen Kalender Pendidikan

Activity Diagram Upload Kalender Pendidikan pada Gambar 5.103 menggambarkan alur aktivitas operasional yang dilakukan oleh Admin dalam mengunggah berkas kalender pendidikan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka halaman Kalender

Akademik, lalu menekan tombol Upload Dokumen Kalender sehingga sistem merespons dengan menampilkan formulir *modal pop-up* Upload Kalender. Selanjutnya, Admin memilih berkas dokumen berformat PDF atau gambar dari direktori lokal perangkat dan menekan tombol simpan untuk dikirimkan ke sistem.

Apabila berkas dikirimkan, sistem secara otomatis mengeksekusi proses penerimaan data dan validasi kelayakan berkas, seperti pemeriksaan batas ukuran maksimal 5MB serta kesesuaian format PDF/IMG. Jika berkas dinyatakan valid, sistem memicu mekanisme pembersihan dengan menghapus berkas lama di direktori *storage* (apabila ada), menyimpan berkas baru, serta memperbarui atribut *calendar_file* pada tabel *school_years* di basis data. Rangkaian aktivitas ini diakhiri dengan pengiriman notifikasi sukses ke antarmuka pengguna, yang membuktikan bahwa pemodelan alur pengunggahan dokumen telah dirancang secara sistematis untuk menjaga efisiensi penyimpanan dan keandalan arsip sekolah.

2. *Class diagram* kalender pendidikan (upload PDF dokumen kalender pendidikan)



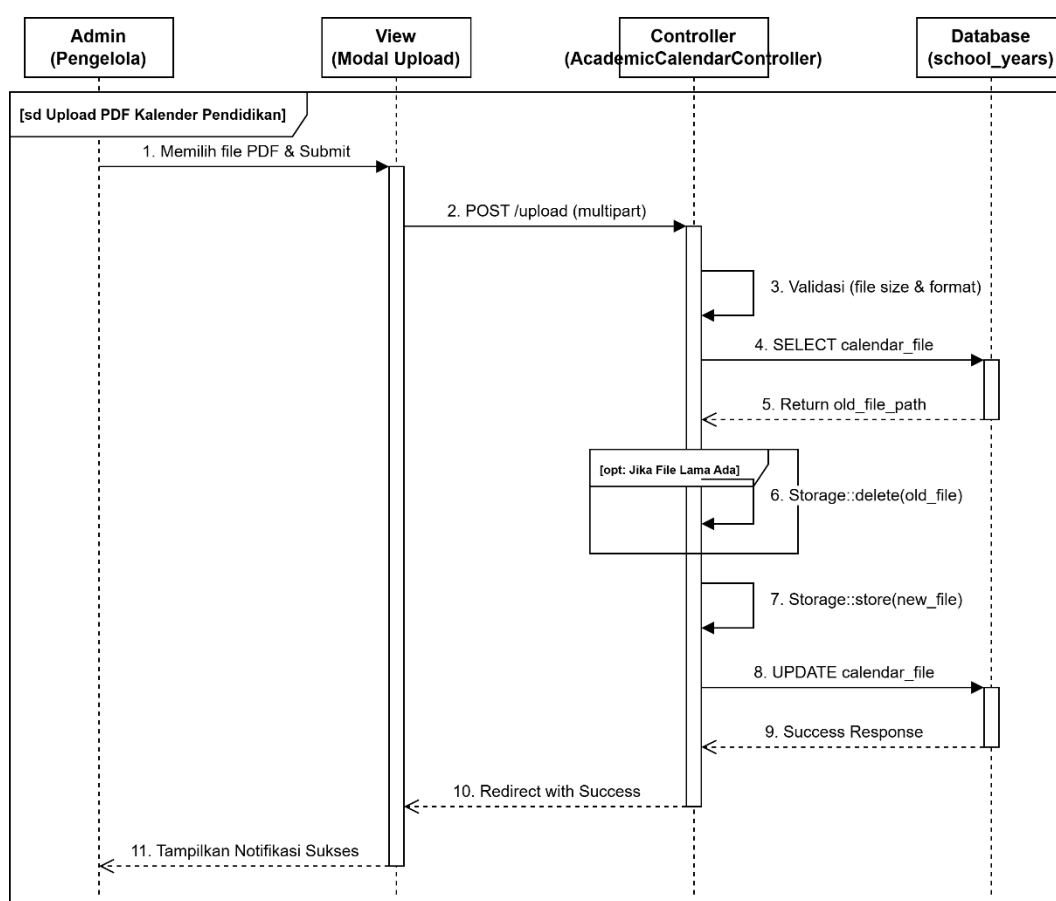
Gambar 5.104 *Class Diagram Upload PDF Dokumen Kalender Pendidikan*

Class Diagram Upload Kalender Pendidikan pada Gambar 5.104 digunakan untuk menggambarkan pemodelan struktural dan entitas objek yang menangani penyimpanan berkas kalender akademik pada sistem. Diagram ini menunjukkan penambahan atribut khusus *calendar_file: varchar (PDF)* serta fungsi operasional *uploadCalendarPDF()* pada kelas utama *school_years*, di samping atribut standar seperti *id*, *name*, dan *is_active*.

Melalui *Class Diagram* ini, perancangan struktur data ditujukan untuk menautkan berkas dokumen digital secara langsung dengan entitas periode tahun pelajaran

yang sedang aktif. Kehadiran fungsi `uploadCalendarPDF()` bertindak sebagai enkapsulasi logika bisnis yang mengelola pembaruan jalur (*path*) berkas di dalam basis data tanpa merusak struktur relasi tabel lainnya. Penataan properti yang ringkas namun terfokus ini memastikan bahwa mekanisme penyimpanan data acuan kalender sekolah dapat dikelola secara terpusat, konsisten, dan mudah dirujuk oleh aktor pengguna lain dalam sistem.

3. *Sequence diagram* kalender pendidikan (upload PDF dokumen kalender pendidikan)



Gambar 5.105 *Sequence Diagram Upload PDF Dokumen Kalender Pendidikan*

Sequence Diagram Upload Kalender Pendidikan pada Gambar 5.105 menggambarkan urutan komunikasi sekuensial berdasarkan garis waktu antara Admin, antarmuka *View (Modal Upload)*, pengontrol *AcademicCalendarController*, dan *Database (school_years)* selama proses pengunggahan berkas berlangsung. Proses diawali ketika Admin memilih berkas

PDF/gambar dan mengirimkan formulir via metode HTTP POST dengan format pengiriman *multipart/form-data* menuju *controller*.

Setelah menerima *payload*, *controller* melakukan validasi format dan ukuran berkas secara internal, lalu mengeksekusi kueri SELECT *calendar_file* ke basis data untuk mengecek keberadaan berkas lama. Jika berkas lama terdeteksi, *controller* menjalankan perintah *Storage::delete()* untuk menghapus berkas fisik lama sebelum mengeksekusi *Storage::store()* guna menyimpan berkas baru dan memperbarui jalur berkas di tabel *school_years*. Setelah transaksi basis data selesai, sistem mengembalikan respons sukses dan mengarahkan kembali antarmuka disertai penayangan notifikasi, sehingga menjamin proses pertukaran data berkas berjalan secara terisolasi, aman, dan efisien.

4. Implementasi kalender pendidikan (upload PDF dokumen kalender pendidikan)

```

1 // File: app/Http/Controllers/Admin/AcademicCalendarController.php (Backend)
2
3 <?php
4
5 namespace App\Http\Controllers\Admin;
6
7 use App\Http\Controllers\Controller;
8 use App\Models\AcademicCalendar;
9 use App\Models\SchoolYear;
10 use Illuminate\Http\Request;
11 use Illuminate\Support\Facades\Storage;
12 use Inertia\Inertia;
13
14 class AcademicCalendarController extends Controller
15 {
16     // ... method index, store, update, destroy ...
17
18     /**
19      * Upload dokumen kalender pendidikan untuk tahun ajaran tertentu.
20      */
21     public function uploadCalendar(Request $request, SchoolYear $schoolYear)
22     {
23         $request->validate([
24             'calendar_file' => 'required|file|mimes:pdf,jpg,jpeg,png|max:5120',
25         ]);
26
27         if ($request->hasFile('calendar_file')) {
28             // Hapus file lama jika ada
29             if ($schoolYear->calendar_file) {
30                 Storage::disk('public')->delete($schoolYear->calendar_file);
31             }
32
33             // Simpan file baru ke Storage public
34             $path = $request->file('calendar_file')->store('calendars', 'public');
35             $schoolYear->update(['calendar_file' => $path]);
36
37             return redirect()->back()->with('success', 'Dokumen kalender berhasil diunggah.');
```

Gambar 5.106 Code Backend Upload PDF Dokumen Kalender Pendidikan

Potongan kode pada Gambar 5.106 memperlihatkan implementasi logika *backend* pada metode `uploadCalendar()` di dalam `AcademicCalendarController` menggunakan *framework* Laravel. Logika diawali dengan validasi ketat terhadap berkas masukan (`calendar_file`) melalui aturan `required|file|mimes:pdf,jpg,jpeg,png|max:5120` untuk mengintersepsi potensi ancaman keamanan berupa *malicious file upload*. Selanjutnya, sistem

memanfaatkan fasad `Storage::disk('public')` untuk memeriksa dan menghapus berkas fisik lama dari direktori penyimpanan sebelum menyimpan berkas baru dan memperbarui jalur alamat (*file path*) pada model `SchoolYear`. Pendekatan rekayasa perangkat lunak ini secara kritis menerapkan pembersihan memori (*garbage cleanup*) di tingkat server untuk mencegah penumpukan berkas sampah yang tidak terpakai (*orphaned files*), sekaligus menjaga efisiensi ruang penyimpanan pada peladen.

```

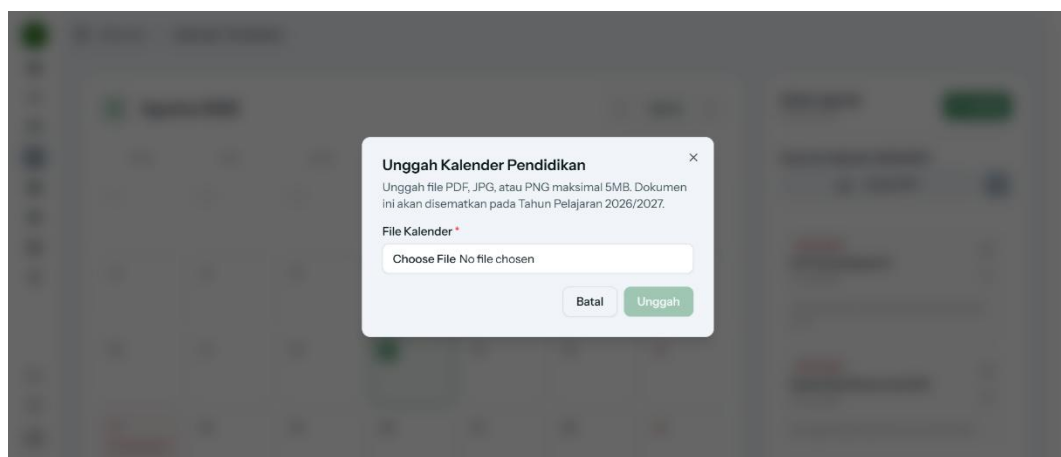
1 // File: resources/js/pages/admin/academic-calendars/index.tsx (Frontend)
2 import { useForm } from '@inertiajs/react';
3 import { useState } from 'react';
4 import { Button } from '@components/ui/button';
5 import { Dialog, DialogContent, DialogHeader, DialogTitle, DialogDescription,
6   DialogFooter } from '@components/ui/dialog';
7 import { Input } from '@components/ui/input';
8 import { Label } from '@components/ui/label';
9 import { Upload, Download } from 'lucide-react';
10 export default function AcademicCalendarIndex({ activeSchoolYear }) {
11   const [isUploadModalOpen, setIsUploadModalOpen] = useState(false);
12   // 1. Inisialisasi State Form Data khusus untuk unggah File
13   const {
14     data: uploadData,
15     setData: setUploadData,
16     post: postUpload,
17     processing: processingUpload,
18     errors: errorsUpload,
19     reset: resetUpload,
20   } = useForm({
21     calendar_file: null as File | null,
22   });
23   // 2. Fungsi Submit (Memanggil Endpoint Backend)
24   const handleUploadSubmit = (e: React.FormEvent) => {
25     e.preventDefault();
26     if (activeSchoolYear) {
27       postUpload(`/admin/school-years/${activeSchoolYear.id}/upload-calendar`, {
28         onSuccess: () => {
29           setIsUploadModalOpen(false);
30           resetUpload();
31         },
32       });
33     }
34   };
35   return (
36     <div>
37       {/* 3. Panel UI: Tombol Trigger Upload & Download PDF */}
38       <div className="flex flex-col gap-2">
39         <h3 className="text-xs font-bold">Dokumen Kalender</h3>
40         {activeSchoolYear?.calendar_file_url ? (
41           <div className="flex items-center gap-2">
42             <Button onClick={() =>
43               window.open(activeSchoolYear.calendar_file_url, '_blank')}>
44             <Download className="mr-2 h-3 w-3" /> Unduh PDF
45             </Button>
46             <Button variant="secondary" onClick={() =>
47               setIsUploadModalOpen(true)}>
48             <Upload className="h-3 w-3" /> Ganti Dokumen
49             </Button>
50           </div>
51         ) : (
52           <Button variant="outline" onClick={() =>
53             setIsUploadModalOpen(true)} disabled={!activeSchoolYear}>
54           <Upload className="mr-2 h-3 w-3" /> Unggah PDF Kalender
55           </Button>
56         )}
57       </div>
58       {/* 4. Modal Pop-up: Form Upload */}
59       <Dialog open={isUploadModalOpen} onOpenChange={setIsUploadModalOpen}>
60         <DialogContent className="sm:max-w-[425px]">
61           <form onSubmit={handleUploadSubmit}>
62             <DialogHeader>
63               <DialogTitle>Unggah Kalender Pendidikan</DialogTitle>
64               <DialogDescription>
65                 Unggah file PDF, JPG, atau PNG maksimal 5MB. Dokumen
66                 ini akan disematkan pada Tahun Pelajaran {activeSchoolYear?.name}.
67               </DialogDescription>
68             </DialogHeader>

```

Gambar 5.107 Code Frontend Upload PDF Dokumen Kalender Pendidikan

Potongan kode pada Gambar 5.107 menunjukkan implementasi antarmuka *frontend* pada komponen `AcademicCalendarIndex` menggunakan React JS dan kait (*hook*) `useForm` dari `Inertia.js`. Komponen ini mengelola status (*state*) formulir pengunggahan secara reaktif melalui variabel `calendar_file`, di mana penanganan pengiriman data (`handleUploadSubmit`) mengeksekusi metode `postUpload()` secara asinkron menuju *endpoint* peladen. Ditinjau dari sudut pandang *User Experience* (UX), kode ini menyajikan logika dinamis pada panel UI yang secara otomatis mengalihkan tombol dari "Unggah PDF Kalender" menjadi tombol "Unduh PDF" dan "Ganti Dokumen" apabila jalur berkas (`calendar_file_url`) telah tersedia pada tahun ajaran aktif. Teknik *Conditional Component Rendering* ini memastikan interaksi pengunggahan dokumen berlangsung secara responsif, interaktif, serta memberikan kejelasan status bagi administrator.

5. Tampilan Halaman kalender pendidikan (upload PDF dokumen kalender pendidikan)



Gambar 5.108 Halaman *Upload* PDF Dokumen Kalender Pendidikan

Gambar 5.108 menampilkan hasil implementasi antarmuka *modal* Unggah Kalender Pendidikan pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Antarmuka ini dirancang menggunakan komponen jendela timbul (*modal popup*) berbasis *Single Page Application* (SPA) yang muncul secara halus di atas tampilan utama tanpa memicu pemuatan ulang halaman secara penuh. Jendela *modal* ini memuat judul petunjuk, deskripsi batasan berkas (format PDF, JPG, atau PNG

dengan ukuran maksimal 5MB), bidang elemen masukan berkas (*file input*), serta tombol aksi *Batal* dan *Unggah*.

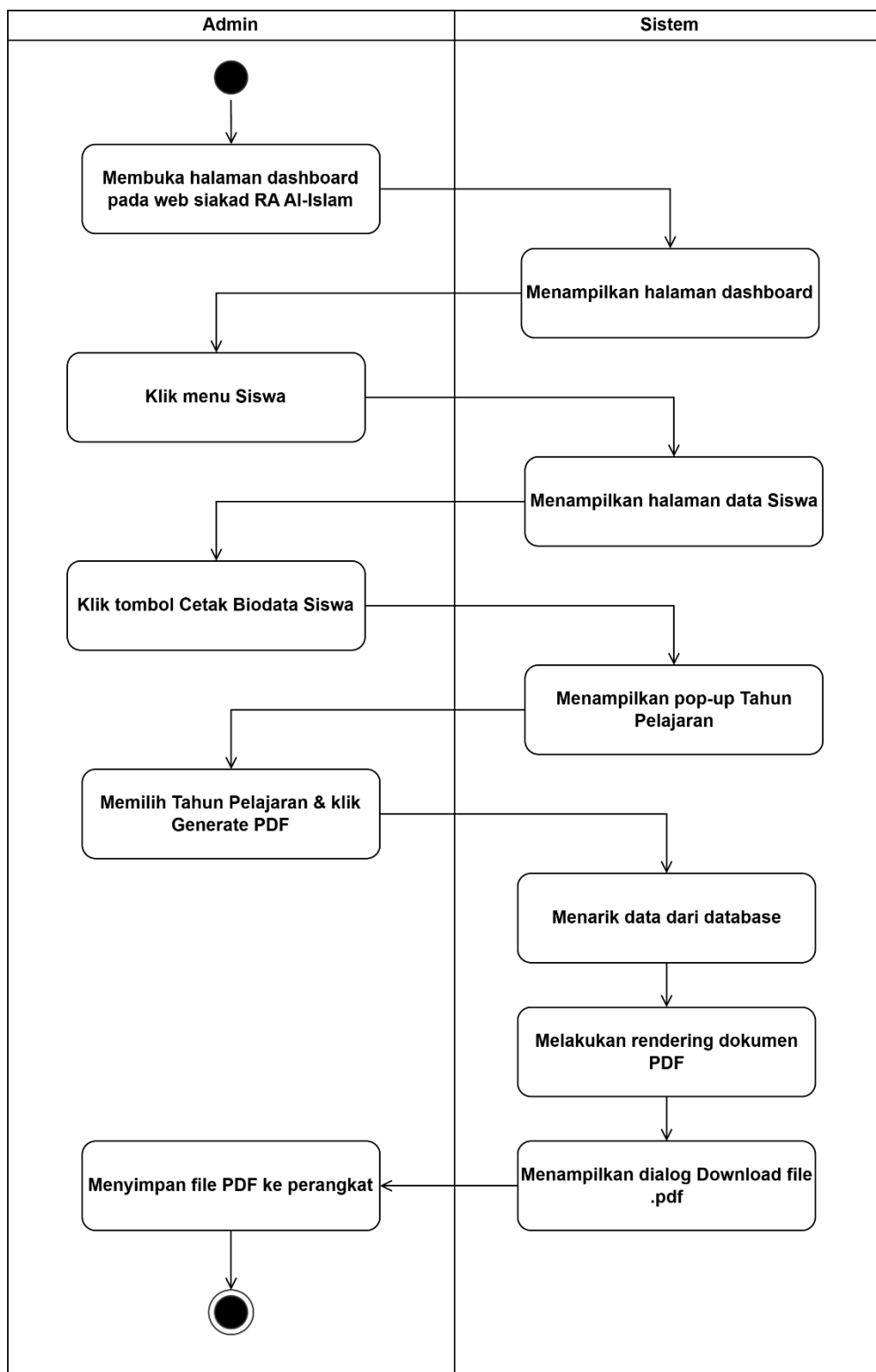
Tampilan dirancang dengan antarmuka yang bersih dan terstruktur sehingga memudahkan Admin dalam melakukan pembaruan dokumen resmi kalender akademik sekolah. Penggunaan komponen *modal overlay* memberikan fokus perhatian penuh kepada pengguna saat melakukan pemilihan berkas lokal dari perangkat. Dengan demikian, halaman *modal* Unggah Kalender Pendidikan ini tidak hanya meningkatkan efisiensi operasional administrasi sekolah dalam pengelolaan berkas publik, tetapi juga membantu meminimalkan potensi kesalahan pengunggahan melalui penyampaian instruksi batasan berkas yang jelas.

5.4.2.5 Siswa (cetak biodata PDF)

Fitur Cetak PDF Data Siswa merupakan pengembangan lanjutan pada menu Siswa yang dikerjakan pada *Sprint 2*. Fitur ini dikembangkan untuk mempermudah Admin dalam menghasilkan laporan data siswa dalam bentuk dokumen *Portable Document Format* (PDF) tanpa harus melakukan pencatatan atau penyusunan laporan secara manual. Dengan adanya fitur ini, proses penyajian data siswa menjadi lebih cepat, terstruktur, dan siap digunakan sebagai dokumen administrasi maupun arsip sekolah. Selain meningkatkan efisiensi kerja, fitur ini juga membantu menjaga konsistensi format laporan yang dihasilkan sehingga informasi yang disajikan lebih mudah dibaca dan didokumentasikan.

Perancangan fitur Cetak PDF Data Siswa dilakukan menggunakan *Activity Diagram* dan *Sequence Diagram* sebagai dasar dalam memahami alur proses pembuatan laporan sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan aktivitas pengguna ketika melakukan proses pencetakan data siswa, sedangkan *Sequence Diagram* menggambarkan urutan komunikasi antara pengguna, sistem, dan basis data selama proses menghasilkan dokumen PDF. Kedua diagram tersebut menjadi acuan dalam proses implementasi sehingga fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan sistem yang telah dirancang.

1. *Activity diagram siswa (cetak biodata PDF)*

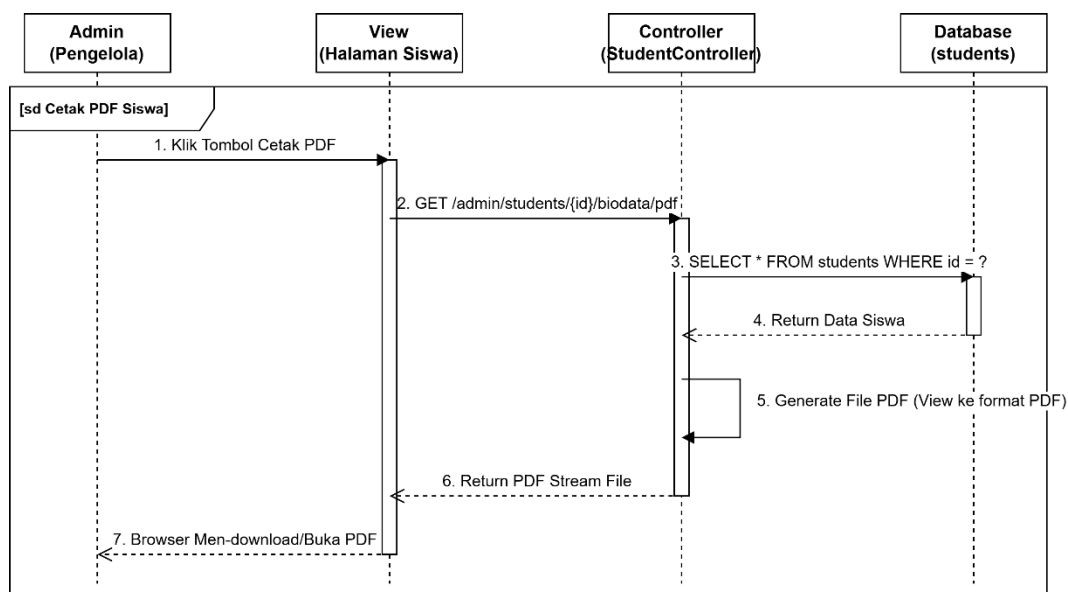


Gambar 5.109 *Activity Diagram Siswa (Cetak biodata PDF)*

Activity Diagram Cetak Biodata PDF Siswa pada Gambar 5.109 menggambarkan alur proses pembuatan dan pengunduhan dokumen rekapitulasi identitas murid yang dilakukan oleh Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Siswa sehingga sistem menampilkan halaman data siswa yang berisi seluruh daftar rekapitulasi digital. Selanjutnya Admin dapat menekan tombol Cetak Biodata Siswa yang akan memicu sistem untuk menampilkan jendela *pop-up* pemilihan periode akademik sebelum dokumen fisik digital digenerasikan.

Apabila Admin telah menentukan Tahun Pelajaran dan mengklik tombol *Generate PDF*, sistem secara otomatis akan menarik data relasional dari database, mengeksekusi proses rendering dokumen, hingga memunculkan dialog unduhan berkas format .pdf pada perangkat. Penanganan alur ini dirancang secara kritis untuk memastikan bahwa output dokumen yang dicetak telah tersaring secara valid dan rapi sesuai dengan tahun ajaran aktif peserta didik. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme pencetakan arsip buku induk berbasis PDF dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. Sequence diagram siswa (cetak biodata PDF)



Gambar 5.110 Sequence Diagram Siswa (Cetak Biodata PDF)

Sequence Diagram Cetak Biodata PDF Siswa pada Gambar 5.110 menggambarkan urutan komunikasi kronologis antara Admin, komponen antarmuka, kontroler, dan basis data selama proses pencetakan dokumen identitas berlangsung. Diagram ini menunjukkan bagaimana permintaan cetak diinisiasi oleh Admin melalui komponen *View* (Halaman Siswa), yang kemudian meneruskan permintaan *HTTP GET* dengan rute `/admin/students/{id}/biodata/pdf` menuju kontroler *backend* (*StudentController*). Kontroler tersebut kemudian mengeksekusi *query SQL* `SELECT * FROM students WHERE id = ?` ke *database* untuk menarik entitas data siswa secara spesifik berdasarkan parameter ID yang dikirimkan.

Setelah *database* memberikan kembalian berupa objek data siswa, kontroler akan menjalankan fungsi internal secara sekuensial untuk melakukan konversi tampilan HTML menjadi aliran berkas (*PDF Stream File*) sebelum dikirimkan kembali ke peramban client untuk diunduh secara otomatis. Melalui *Sequence Diagram* ini, proses pengolahan dokumen di balik layar dapat dipisahkan secara modular antara layer presentasi dengan mesin kompilasi PDF di sisi *server backend* sehingga tidak membebani memori perangkat pengguna. Diagram ini menjadi acuan utama dalam

implementasi logika pencetakan berkas digital terotomatisasi sehingga proses pertukaran data berjalan secara konsisten, aman, serta menjamin keakuratan isi dokumen cetak sesuai dengan rancangan sistem sebelumnya.

3. Implementasi siswa (cetak biodata PDF)

```

1  /**
2   * Cetak biodata form siswa PDF (Massal per Tahun Pelajaran)
3   */
4   public function bulkBiodataPdf(Request $request)
5   {
6       // Mengekstrak parameter kueri Tahun Pelajaran dari URL
7       $schoolYearId = $request->query('school_year_id');
8       $schoolYear = $schoolYearId ? SchoolYear::find($schoolYearId) :
        SchoolYear::where('is_active', true)->first();
9
10      // Menarik data siswa menggunakan teknik Eager Loading untuk efisiensi kueri
        basis data
11      $students = Student::with(['studentClass.schoolYear'])
12      ->when($schoolYear, function ($q) use ($schoolYear) {
13          $q->whereHas('studentClass', function ($q) use ($schoolYear) {
14              $q->where('school_year_id', $schoolYear->id);
15          });
16      })
17      ->orderBy('full_name')
18      ->get();
19
20      // Server-Side PDF Rendering: Mengubah View Template HTML (Blade) menjadi
        format PDF
21      $pdf = Pdf::loadView('bulk-biodata-pdf', [
22          'students' => $students,
23          'schoolYear' => $schoolYear,
24      ])
25      ->setPaper('A4', 'portrait') // Konfigurasi format kertas
26      ->setOption([
27          'isHtml5ParserEnabled' => true,
28          'isRemoteEnabled'      => true, // Mengizinkan render gambar eksternal
        (foto siswa)
29          'dpi'                    => 150,
30      ]);
31
32      // Streaming dokumen PDF ke peramban tanpa harus menyimpannya secara fisik di
        server
33      $filename = 'biodata-siswa-massal-' . str_replace('/', '-', ($schoolYear->
        >name ?? 'semua')) . '.pdf';
34      return $pdf->stream($filename);
35  }

```

Gambar 5.111 *Code Backend Siswa (Cetak Biodata PDF)*

Potongan kode pada Gambar 5.111 menunjukkan metode *bulkBiodataPdf()* di dalam *StudentController*. Logika *Backend* ini bertugas untuk mencetak biodata seluruh siswa secara massal berdasarkan filter Tahun Pelajaran. Dari segi *Software*

Engineering, sistem mengimplementasikan teknik *Eager Loading* (*with(['studentClass.schoolYear'])*) untuk menarik data relasional secara serentak guna mencegah masalah N+1 Query yang membebani basis data. Setelah sekumpulan data mentah (*Collection*) ditarik, sistem menggunakan utilitas *library Barryvdh\DomPDF* untuk melakukan *Server-Side PDF Rendering* (Perenderan Dokumen di Sisi Peladen). Mesin ini akan menyuntikkan data tersebut ke dalam *View template HTML*, memanipulasi *layout* kertas menjadi ukuran A4 potret, dan merespons (mengembalikan) data (*stream*) tersebut ke peramban (*browser*) pengguna secara instan (*real-time*).

```

1 // State untuk menyimpan ID Tahun Pelajaran yang dipilih oleh Admin
2 const [filterSchoolYearId, setFilterSchoolYearId] = useState<number>(
3   schoolYears.find((y) => y.is_active)?.id ?? (schoolYears[0]?.id || 0),
4 );
5
6 /**
7  * Handler untuk memicu (trigger) pengunduhan/pembukaan tab PDF baru
8  */
9 const handleCetakBiodataMassal = () => {
10   // Merakit URL akhir dengan menyuntikkan Parameter Kueri (Query String)
11   const url = route('admin.students.biodata.bulk-pdf') + `?
school_year_id=${filterSchoolYearId}`;
12
13   // Membuka URL di tab baru browser (_blank) untuk menampilkan Stream PDF
14   window.open(url, '_blank');
15
16   // Menutup jendela pop-up modal secara otomatis
17   setShowModal(false);
18 };
19
20 return (
21   /* Potongan kode di dalam Jendela Modal */
22   <div className="space-y-4">
23     <div>
24       <label className="mb-1.5 block text-sm font-medium">Pilih Tahun
Pelajaran Induk</label>
25       <select
26         value={filterSchoolYearId}
27         onChange={(e) => setFilterSchoolYearId(Number(e.target.value))}
28         className="w-full rounded-lg border bg-background px-3 py-2.5
text-sm"
29       >
30         {schoolYears.map((y) => (
31           <option key={y.id} value={y.id}>
32             {y.name} {y.is_active ? '(Aktif)' : ''}
33           </option>
34         ))}
35       </select>
36       <p className="mt-2 text-xs text-muted-foreground">
37         Dokumen ini akan mencetak form biodata seluruh siswa yang
tergabung di tahun ajaran yang dipilih.
38       </p>
39     </div>
40
41     /* Tombol Eksekusi Aksi */
42     <button
43       onClick={handleCetakBiodataMassal}
44       className="items-center gap-1.5 rounded-lg bg-primary px-5 py-2 text-
sm font-bold shadow-sm"
45     >
46       Cetak Biodata Massal
47     </button>
48   </div>
49 );

```

Gambar 5.112 *Code Frontend Siswa (Cetak Biodata PDF)*

Potongan kode pada Gambar 5.112 menunjukkan antarmuka *Frontend* berupa *Modal Component* (Jendela Timbul) di React JS. Berbeda dengan pengiriman

formulir pada umumnya yang menggunakan AJAX (*router.post*), pemicu pencetakan PDF ini menggunakan pendekatan *External URL Routing*. Ketika admin memilih Tahun Pelajaran dan menekan tombol cetak, *React* akan menangkap identitas *state ID* (yakni *filterSchoolYearId*), merakitnya menjadi sebuah parameter kueri (*URL Query Parameter*), lalu memanggil antarmuka *API* peramban asli (*window.open(url, '_blank')*) untuk membuka tab baru secara paralel. Teknik ini wajib dilakukan karena respons dari server *backend* bukanlah format *JSON* standar, melainkan format dokumen biner murni (*application/pdf*) yang harus di-(*stream*) dan dirender langsung oleh penampil PDF bawaan browser pengguna.

4. Tampilan halaman siswa (cetak biodata PDF)

The screenshot shows a web application interface for printing student biodata. The top bar includes a hamburger menu, the title 'Biodata Siswa', and document controls (page 1 of 3, 53% zoom, and icons for print, download, and share). The main content area is divided into a sidebar on the left showing document thumbnails (labeled 1 and 2) and a large central preview area. The preview area displays a form titled '*** SISWA AKTIF ***' and 'NOMOR INDUK ANAK'. The form includes fields for student ID, name, gender, date of birth, and parent information, along with checkboxes for various attributes.

*** SISWA AKTIF ***		NOMOR INDUK ANAK	
No Induk	000001		
Kelompok	Berkas	A	B
Tahun	2023/2024		
Foto 3 X 4			
A. Keterangan Tentang Identitas Anak			
1. Nama			
1.1 Nama Lengkap	ABDIFARID HIRMI FALZANISYAH		
1.2 Nama Panggilan			
2. Jenis Kelamin	LAKI-LAKI		
3. Tempat / Tanggal Lahir	CIAMIS, 27 July 2018		
4. Alamat dan Nomor Induk	DUSUN SIBENGALEH		
5. Anak ke berapa	2		
6. Jumlah Saudara			
6.1 Saudara Kandung	Orang		
6.2 Saudara Tiri	Orang		
6.3 Saudara Angkat	Orang		
7. Anak Yatim/Piada/Yatim Piada			
8. Bahasa Daerah - art Di Rantau			
9. Warga Negara	INDONESIA		
10. Keterangan lainnya			
B. Keterangan Tentang Identitas Orang Tua / Wali			
1. Ayah Kandung / Tiri / Angkat atau Wali			
2)			
1.1 Nama	NAINANG FRIATNA		
1.2 Tempat dan Tanggal Lahir			
1.3 Agama	ISLAM		
1.4 Pendidikan Terakhir			
Foto 3 X 4			

Gambar 5.113 Halaman Siswa (Cetak Biodata PDF)

Gambar 5.113 menunjukkan hasil implementasi halaman Cetak Biodata PDF Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan visualisasi pratinjau (*preview*) dokumen cetak dari data identitas anak yang memuat nomor induk anak, rombongan belajar kelompok tahun, komponen penempatan pasfoto, serta rincian lengkap mengenai keterangan identitas anak beserta orang tua/wali murid. Selain itu, tersedia menu navigasi peramban dokumen di bagian atas (seperti informasi halaman, kontrol tingkat perbesaran, tombol unduh, dan cetak langsung) serta panel laci pratinjau lembar (*thumbnail sidebar*) di

sisi kiri sebagai media bagi Admin untuk melakukan pemeriksaan cetak dokumen sesuai dengan kebutuhan administrasi sekolah.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Admin dalam melihat dan memeriksa berkas cetak fisik buku induk secara lebih efektif. Selain itu, sistem juga menyediakan format tata letak teks baku yang rapi dan terstruktur mengikuti standar formulir resmi pendidikan anak usia dini sehingga dapat digunakan sebagai acuan arsip yang sah dalam proses pelaporan akademik berjalan. Dengan demikian, halaman Cetak Biodata PDF Siswa tidak hanya berfungsi sebagai media penampil berkas digital, tetapi juga membantu menjaga konsistensi validitas konversi data peserta didik pada seluruh fitur pengelolaan arsip dokumen fisik di dalam sistem.

5.4.2.6 *Black box testing sprint 2*

Setelah seluruh fitur pada *Sprint 2* berhasil diimplementasikan, tahapan selanjutnya adalah melakukan pengujian menggunakan metode *Black Box Testing*. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur yang telah dikembangkan dapat berfungsi sesuai dengan kebutuhan fungsional yang telah ditetapkan pada *Product Backlog*. Berbeda dengan pengujian yang berfokus pada struktur kode program, *Black Box Testing* dilakukan dengan mengamati kesesuaian antara masukan (*input*) yang diberikan pengguna dengan keluaran (*output*) yang dihasilkan oleh sistem. Dengan demikian, pengujian ini dapat membuktikan bahwa setiap fungsi yang tersedia telah bekerja sebagaimana mestinya sebelum hasil pengembangan dipresentasikan kepada pihak RA Al-Islam Gunungcupu melalui *Sprint Review*.

Pengujian pada *Sprint 2* dilakukan terhadap seluruh fitur yang berhasil diselesaikan, yaitu Siswa (Kelola Siswa), Absensi (Input Guru), Kalender Pendidikan (Mengelola), dan Siswa (Cetak Biodata PDF). Setiap fitur diuji berdasarkan skenario penggunaan yang mewakili aktivitas pengguna pada sistem, kemudian dibandingkan dengan hasil yang diharapkan untuk mengetahui kesesuaian fungsi yang dihasilkan. Hasil pengujian menunjukkan bahwa seluruh fitur pada *Sprint 2* dapat berjalan sesuai dengan kebutuhan fungsional tanpa ditemukan kesalahan yang

memengaruhi proses utama sistem. Oleh karena itu, seluruh fitur dinyatakan berhasil dan layak untuk dilanjutkan ke tahapan *Sprint Review* sebagai bahan evaluasi bersama stakeholder.

Tabel 5.8 Hasil *Black Box Testing Sprint 2*

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
1	Menambahkan, mengubah, dan menghapus data Siswa	Data siswa	Data siswa berhasil disimpan, diperbarui, dan dihapus sesuai dengan proses yang dilakukan	Seluruh proses pengelolaan data siswa berhasil dijalankan	Berhasil
2	Menginput data absensi siswa	Data absensi siswa	Data absensi berhasil disimpan sesuai dengan data yang diinput	Data absensi berhasil disimpan	Berhasil
3	Menambahkan, mengubah, dan menghapus data Kalender Pendidikan	Data kalender pendidikan	Data kalender pendidikan berhasil disimpan, diperbarui, dan dihapus sesuai dengan proses yang dilakukan	Seluruh proses pengelolaan kalender pendidikan berhasil dijalankan	Berhasil

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
4	Mencetak biodata siswa dalam format PDF	Memilih data siswa	Sistem berhasil menampilkan dan menghasilkan biodata siswa dalam format PDF	Dokumen PDF berhasil ditampilkan	Berhasil
5	Mengunggah dokumen PDF Kalender Pendidikan	Berkas PDF kalender pendidikan (maksimal 5MB)	Sistem berhasil memvalidasi, mengunggah, dan menyimpan berkas dokumen kalender pendidikan	Berkas PDF kalender pendidikan berhasil diunggah dan tersimpan	Berhasil

Berdasarkan hasil *Black Box Testing* pada Tabel 5.8, seluruh skenario pengujian yang dilakukan pada fitur *Sprint 2* menunjukkan hasil sesuai dengan ekspektasi. Setiap fungsi yang diuji berhasil dijalankan dengan baik, mulai dari pengelolaan data siswa, input absensi, pengelolaan kalender pendidikan, unggah dokumen PDF kalender pendidikan, hingga pencetakan biodata siswa dalam format PDF. Hasil pengujian tersebut menunjukkan bahwa seluruh fitur telah memenuhi kebutuhan fungsional sistem sehingga proses pengembangan dapat dilanjutkan ke tahapan *Sprint Review* untuk memperoleh umpan balik dari stakeholder.

5.4.2.7 *Sprint review sprint 2*

Setelah seluruh fitur pada *Sprint 2* berhasil diimplementasikan dan dinyatakan berfungsi dengan baik melalui *Black Box Testing*, tahapan selanjutnya adalah melaksanakan *Sprint Review*. Pada tahap ini, hasil pengembangan sistem dipresentasikan kepada stakeholder, yaitu pihak RA Al-Islam Gunungcupu, untuk memperoleh umpan balik terhadap fitur-fitur yang telah dikembangkan. Kegiatan ini bertujuan untuk memastikan bahwa setiap fungsi yang diimplementasikan telah sesuai dengan kebutuhan operasional sekolah serta memenuhi harapan pengguna sebelum proses pengembangan dilanjutkan ke *Sprint* berikutnya.

Sprint Review pada *Sprint 2* dilakukan terhadap seluruh fitur yang berhasil dikembangkan, yaitu Kelola Siswa, Input Absensi, Kelola Kalender Pendidikan, Unggah Dokumen PDF Kalender Pendidikan, dan Cetak Biodata Siswa (PDF). Hasil evaluasi menunjukkan bahwa sebagian besar fitur telah sesuai dengan kebutuhan pengguna dan dapat digunakan dengan baik pada proses operasional sekolah. Namun, pada fitur Cetak Biodata Siswa (PDF) terdapat masukan agar tampilan hasil cetak disesuaikan dengan format Buku Induk yang digunakan di RA Al-Islam Gunungcupu dengan keterangan siswa yang sudah lulus dan aktif. Masukan tersebut diterima sebagai bahan penyempurnaan sehingga hasil implementasi dapat lebih sesuai dengan kebutuhan pengguna. Hasil *Sprint Review Sprint 2* dapat dilihat pada Tabel 5.9.

Tabel 5.9 Hasil *Sprint Review Sprint 2*

Pengguna	Fitur	<i>Review</i>
Admin	Kelola Siswa	Fitur pengelolaan data siswa telah berjalan sesuai kebutuhan dan seluruh fungsi pengelolaan data dapat digunakan dengan baik.

Pengguna	Fitur	<i>Review</i>
	Kelola Kalender Pendidikan	Fitur pengelolaan kalender pendidikan telah berjalan sesuai kebutuhan dan seluruh proses pengelolaan data berhasil dilakukan dengan baik.
	Upload PDF Kalender Pendidikan	Fitur pengunggahan dokumen PDF kalender pendidikan telah berjalan sesuai kebutuhan. Sistem berhasil memvalidasi format serta menyimpan berkas berukuran hingga 5MB dengan baik.
	Cetak Biodata Siswa (PDF)	Perlu dilakukan penyesuaian tampilan hasil cetak PDF agar menyerupai format buku induk yang digunakan di RA Al-Islam Gunungcupu dengan keterangan sudah lulus dan aktif sehingga informasi biodata siswa lebih mudah dibaca dan sesuai dengan kebutuhan sekolah.
Guru	Input Absensi	Fitur input absensi telah berjalan sesuai kebutuhan sehingga proses pencatatan kehadiran siswa dapat dilakukan dengan baik.

Berdasarkan hasil *Sprint Review* pada Tabel 5.9, sebagian besar fitur yang dikembangkan pada *Sprint 2* telah diterima dan dinyatakan sesuai dengan kebutuhan pengguna. Fitur Kelola Siswa, Input Absensi, Kelola Kalender Pendidikan dan Unggah Dokumen PDF Kalender Pendidikan telah berjalan sesuai dengan proses bisnis yang diterapkan di RA Al-Islam Gunungcupu. Sementara itu, pada fitur Cetak Biodata Siswa (PDF) diperoleh masukan berupa penyesuaian tampilan hasil cetak dan keterangan sudah lulus dan aktif agar menyerupai format Buku Induk yang digunakan oleh sekolah. Masukan tersebut kemudian menjadi dasar dalam melakukan penyempurnaan terhadap tampilan dokumen sehingga hasil

cetak tidak hanya berfungsi dengan baik, tetapi juga sesuai dengan kebutuhan administrasi yang berlaku di RA Al-Islam Gunungcupu.

5.4.2.8 *Sprint retrospective sprint 2*

Setelah *Sprint Review* selesai dilaksanakan, tahapan berikutnya adalah *Sprint Retrospective*. Pada tahap ini, peneliti melakukan evaluasi terhadap keseluruhan proses pelaksanaan *Sprint 2*, mulai dari perencanaan *backlog*, implementasi sistem, pengujian menggunakan *Black Box Testing*, hingga hasil *Sprint Review* yang telah dilakukan bersama pengguna. Evaluasi ini bertujuan untuk mengidentifikasi hal-hal yang telah berjalan dengan baik selama *Sprint* berlangsung serta mengetahui aspek yang masih perlu disempurnakan agar proses pengembangan pada *Sprint* berikutnya dapat berjalan secara lebih efektif dan sesuai dengan kebutuhan pengguna. Hasil *Sprint Retrospective Sprint 2* dapat dilihat pada Tabel 5.10.

Tabel 5.10 Hasil *Sprint Retrospective Sprint 2*

Pengguna	Aspek Evaluasi	<i>Sprint Retrospective</i>
Peneliti	Pelaksanaan <i>Sprint 2</i>	Seluruh <i>backlog</i> pada <i>Sprint 2</i> berhasil diselesaikan sesuai dengan perencanaan. Berdasarkan hasil <i>Sprint Review</i> , terdapat masukan terhadap fitur Cetak Biodata Siswa (PDF), yaitu penyesuaian tampilan dokumen dengan keterangan sudah lulus dan aktif agar menyerupai format Buku Induk yang digunakan di RA Al-Islam Gunungcupu. Masukan tersebut telah ditindaklanjuti melalui penyempurnaan tampilan hasil cetak sehingga implementasi yang dihasilkan telah sesuai dengan kebutuhan pengguna dan dapat digunakan sebagai

		dasar untuk melanjutkan proses pengembangan pada <i>Sprint</i> berikutnya.
--	--	--

Berdasarkan hasil *Sprint Retrospective* pada Tabel 5.10, dapat diketahui bahwa pelaksanaan *Sprint 2* telah berjalan sesuai dengan perencanaan yang telah disusun pada tahap *Sprint Planning*. Seluruh *backlog* yang menjadi target *Sprint* berhasil diselesaikan dan diimplementasikan sesuai dengan kebutuhan sistem. Selain itu, proses implementasi menggunakan Laravel 12 dan React JS berlangsung dengan baik, seluruh skenario *Black Box Testing* menunjukkan hasil sesuai dengan yang diharapkan, serta hasil *Sprint Review* menunjukkan bahwa sebagian besar fitur telah diterima oleh pengguna. Adapun masukan yang diberikan pada fitur Cetak Biodata Siswa (PDF) mengenai penyesuaian tampilan hasil cetak agar menyerupai format Buku Induk serta penambahan keterangan aktif dan sudah lulus telah ditindaklanjuti sebagai bagian dari penyempurnaan implementasi pada *Sprint 2*. Dengan demikian, hasil evaluasi menunjukkan bahwa proses pengembangan telah berjalan secara efektif dan mampu menghasilkan sistem yang semakin sesuai dengan kebutuhan operasional RA Al-Islam Gunungcupu.

5.4.3 Sprint 3

Sprint 3 merupakan tahap ketiga dalam proses pengembangan Sistem Informasi Akademik Berbasis Web pada RA Al-Islam Gunungcupu menggunakan metode *Scrum*. Setelah penyelesaian fitur-fitur pada *Sprint 2*, pengembangan dilanjutkan dengan memprioritaskan kebutuhan sistem yang berkaitan dengan pengelolaan data akademik siswa, khususnya pada proses penilaian hasil belajar. Penentuan *backlog* pada *Sprint 3* dilakukan berdasarkan prioritas kebutuhan pengguna yang masih belum terpenuhi pada *Sprint* sebelumnya serta hasil evaluasi yang diperoleh melalui *Sprint Review* dan *Sprint Retrospective*. Dengan demikian, pengembangan pada *Sprint 3* difokuskan pada penyelesaian fitur yang mendukung proses pengelolaan nilai siswa agar informasi akademik dapat dikelola secara lebih efektif, terstruktur, dan mudah diakses sesuai dengan hak akses masing-masing pengguna.

Pelaksanaan *Sprint* 3 diawali dengan penyusunan *Sprint Backlog* berdasarkan *Product Backlog* yang telah ditetapkan pada tahap perencanaan. Selanjutnya, setiap backlog diimplementasikan secara bertahap sesuai dengan prioritas pengembangan, kemudian dilakukan pengujian menggunakan metode *Black Box Testing* untuk memastikan seluruh fungsi berjalan sesuai dengan kebutuhan sistem. Setelah proses pengujian selesai, hasil implementasi dipresentasikan kepada pengguna melalui *Sprint Review* guna memperoleh masukan terhadap fitur yang telah dikembangkan. Tahapan *Sprint* 3 kemudian diakhiri dengan *Sprint Retrospective* sebagai bentuk evaluasi terhadap keseluruhan proses pengembangan sehingga hasil evaluasi tersebut dapat dijadikan dasar dalam meningkatkan kualitas proses pengembangan pada *Sprint* berikutnya.

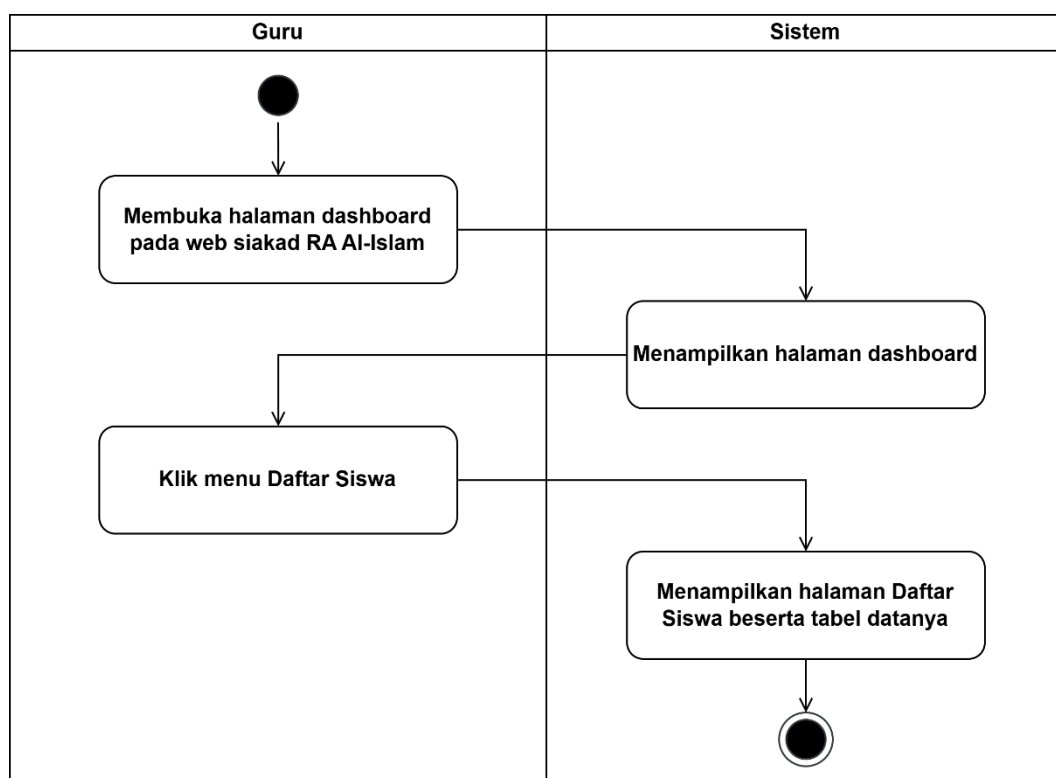
5.4.3.1 Daftar siswa

Fitur Daftar Siswa merupakan salah satu fitur yang dikembangkan pada *Sprint* 3 untuk mendukung proses pengelolaan data akademik oleh Guru. Fitur ini berfungsi menampilkan daftar seluruh siswa berdasarkan kelas yang diampu sehingga Guru dapat dengan mudah mengetahui data siswa yang menjadi tanggung jawabnya sebelum melakukan proses akademik lainnya, seperti pengisian nilai maupun penyusunan laporan perkembangan belajar. Dengan adanya fitur ini, proses pencarian data siswa menjadi lebih cepat dan terstruktur karena seluruh informasi yang diperlukan telah tersaji dalam satu halaman sesuai dengan hak akses pengguna. Selain meningkatkan kemudahan dalam mengakses data siswa, fitur ini juga membantu menjaga ketepatan proses pengelolaan data akademik karena Guru hanya dapat mengakses data siswa yang menjadi tanggung jawabnya.

Perancangan fitur Daftar Siswa dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan alur aktivitas Guru ketika mengakses daftar siswa, *Class Diagram* digunakan untuk menjelaskan struktur kelas yang berperan dalam pengelolaan data siswa beserta hubungan antar kelas yang terlibat, sedangkan *Sequence Diagram* digunakan untuk menggambarkan urutan komunikasi antara pengguna, sistem, dan

basis data selama proses menampilkan daftar siswa berlangsung. Ketiga diagram tersebut menjadi acuan dalam proses implementasi sehingga fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan sistem yang telah dirancang.

1. *Activity diagram* daftar siswa

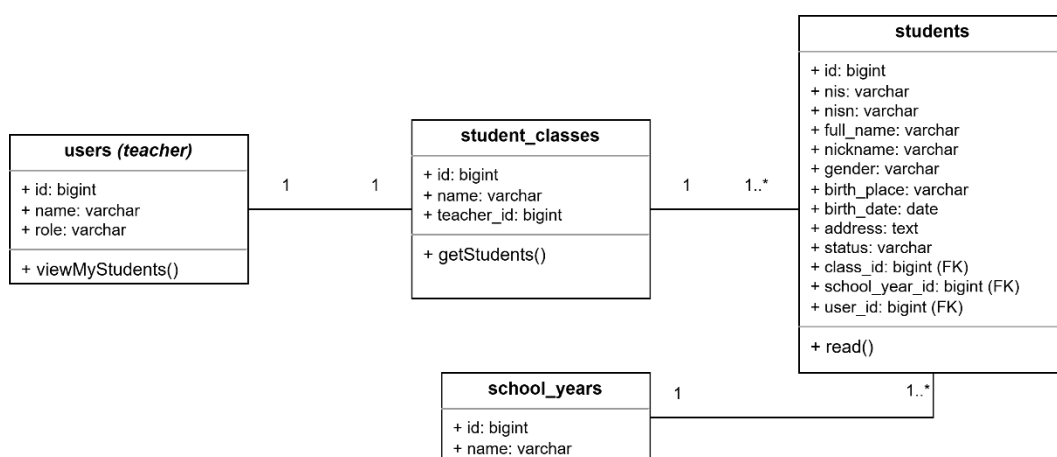


Gambar 5.114 *Activity Diagram* Melihat Daftar Siswa

Activity Diagram Daftar Siswa pada Gambar 5.114 menggambarkan alur proses pembacaan data peserta didik yang dilakukan oleh Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Guru membuka Dashboard kemudian memilih menu Daftar Siswa sehingga sistem menampilkan halaman Daftar Siswa yang berisi seluruh data siswa yang telah tersimpan pada basis data. Selanjutnya Guru dapat melihat representasi visual tabel data siswa yang diampu sesuai penempatan kelas masing-masing secara langsung.

Apabila Guru mengakses menu Daftar Siswa, sistem secara otomatis akan mengeksekusi kueri pembacaan data ke server untuk mengumpulkan daftar murid aktif. Proses pemuatan informasi ini dirancang berjalan di balik layar untuk merender keseluruhan *record* siswa secara cepat dan teratur ke dalam komponen tabel presentasi. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme penyajian daftar kelompok belajar siswa dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* daftar siswa



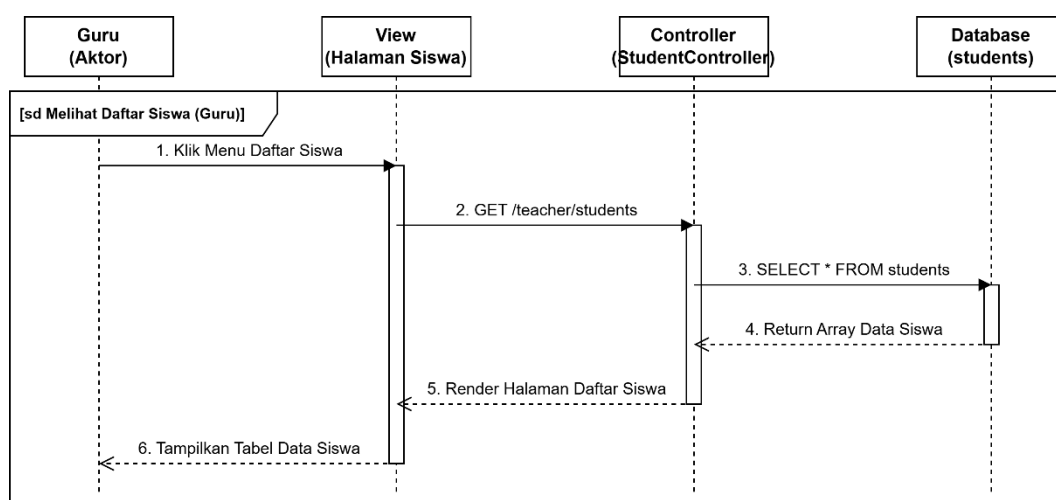
Gambar 5.115 *Class Diagram* Daftar Siswa

Class Diagram Daftar Siswa pada Gambar 5.115 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses penyajian daftar peserta didik pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *students* yang memuat atribut *id*, *nis*, *nispn*, *full_name*, *nickname*, *gender*, *birth_place*, *birth_date*, *address*, *status*, *class_id* (FK), *school_year_id* (FK), dan *user_id* (FK) serta operasi *read()*, dengan kelas *student_classes* yang memuat atribut *id*, *name*, *teacher_id* dan operasi *getStudents()*, kelas *users (teacher)* dengan atribut *id*, *name*, *role* dan operasi *viewMyStudents()*, serta kelas *school_years* yang memuat atribut *id* dan *name*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengelompokan daftar murid dapat dirancang secara lebih sistematis, seperti

kejelasan hubungan *multiplicity* satu ke satu (1 ke 1) antara kelas guru *users* (*teacher*) dengan rombongan belajar *student_classes*, serta hubungan satu ke banyak (1 ke 1..*) antara *student_classes* maupun *school_years* terhadap entitas *students*. Selain itu, diagram ini juga membantu dalam menggambarkan pemetaan batasan hak akses data transaksional melalui penempatan kunci tamu (*class_id*, *school_year_id*, *user_id*) sehingga setiap proses penarikan data *record* daftar siswa dapat dijalankan secara konsisten sesuai dengan kebutuhan aturan basis data pada sistem.

3. *Sequence diagram* daftar siswa



Gambar 5.116 *Sequence diagram* daftar siswa

Sequence Diagram Daftar Siswa pada Gambar 5.116 menggambarkan urutan komunikasi antara Guru, antarmuka sistem, *controller*, dan basis data selama proses melihat daftar siswa berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan informasi kelompok belajar dikirimkan oleh Guru melalui antarmuka halaman siswa, kemudian diproses oleh *controller* (*StudentController*) melalui metode *HTTP GET* dengan rute `/teacher/students` sebelum diteruskan ke *database* (tabel *students*) menggunakan kueri `SQL SELECT * FROM students` hingga menghasilkan respons berupa kembalian data array siswa yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada skenario pemuatan data murid kelas dapat dipahami secara lebih rinci dan berurutan sesuai lini waktu. Diagram ini menjadi acuan dalam implementasi logika penarikan data di sisi server sehingga proses pertukaran data antara komponen presentasi *front-end* React JS, *controller* Laravel, dan basis data MySQL dapat berjalan secara terstruktur, menjaga konsistensi data, serta menjamin efisiensi eksekusi aplikasi sesuai dengan kebutuhan penulisan kode program yang telah dirancang sebelumnya.

4. Implementasi daftar siswa

```

1  public function index(Request $request)
2  {
3      // 1. Ekstraksi Sesi Identitas Pengguna (Guru)
4      $teacherId = auth()->id();
5
6      // 2. Authorization Scoping: Melacak kelas yang menjadi hak asuh eksklusif
       guru tersebut
7      $student_classes = StudentClass::where('teacher_id', $teacherId)->get();
8      $classIds = $student_classes->pluck('id')->toArray();
9
10     // 3. Multi-Tenancy Data Isolation & IDOR Prevention
11     // Mengunci akses hanya pada siswa yang class_id-nya ada di dalam array hak
       asuh guru
12     $query = Student::with(['studentClass', 'schoolYear'])
13         ->whereIn('class_id', $classIds);
14
15     // Filter dinamis untuk Kelas dan Tahun Pelajaran (Hanya aktif dalam scope
       yangizinkan)
16     if ($request->has('class_id') && $request->class_id != '') {
17         $query->where('class_id', $request->class_id);
18     }
19
20     // Kueri pencarian teks (Full_name atau NISN)
21     if ($request->has('search') && $request->search != '') {
22         $search = $request->search;
23         $query->where(function($q) use ($search) {
24             $q->where('full_name', 'like', "%{$search}%")
25             ->orWhere('nisn', 'like', "%{$search}%");
26         });
27     }
28
29     $students = $query->orderBy($sort, $direction)->paginate(10)-
       >withQueryString();
30
31     return Inertia::render('teacher/students/index', [
32         'students' => $students,
33         'filters' => $request->only(['search', 'class_id', 'school_year_id']),
34     ]);
35 }

```

Gambar 5.117 Code Backend Daftar Siswa

Potongan kode pada Gambar 5.117 menunjukkan metode *index()* di dalam *Teacher/StudentController*. Berbeda dengan fitur kelola siswa milik Admin yang menampilkan seluruh isi pangkalan data, antarmuka khusus Guru ini mengimplementasikan teknik keamanan *Authorization Scoping* (Pembatasan Lingkup Otorisasi) atau *Multi-Tenancy Data Isolation*. Sistem mengekstrak sesi ID guru yang sedang masuk (*auth()->id()*), melacak ID kelas mana saja yang menjadi hak asuhnya di tabel *student_classes*, lalu merakitnya ke dalam array *\$classIds*. Array tersebut kemudian disuntikkan secara paksa ke dalam kueri *Eloquent* melalui

klausa *whereIn('class_id', \$classIds)*. Pendekatan ini sangat krusial di dunia nyata karena secara absolut menutup celah keamanan IDOR (*Insecure Direct Object Reference*). Dengan metode ini, secanggih apa pun seorang guru mencoba meretas URL atau memanipulasi Parameter *Request*, pangkalan data menolak (*reject*) mengembalikan data siswa dari kelas guru lain.

```

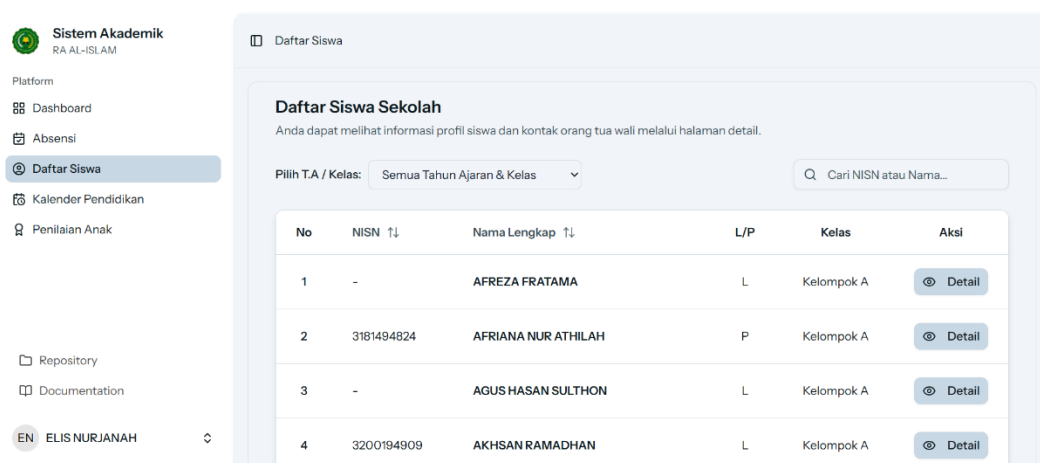
1 // State Preservation via Inertia Router
2 const handleSearchChange = (value: string) => {
3   router.get(
4     route('teacher.students.index'), // Rute API tujuan
5     { ...filters, search: value }, // Payload pencarian
6     {
7       preserveState: true, // Tidak menghapus state memori komponen (misal:
        tulisan yang sedang diketik)
8       replace: true // Mengganti riwayat URL tanpa merusak tombol
        'Back' di peramban
9     }
10  );
11 };
12
13 return (
14   /* Implementasi Read-Only Table View */
15   <div className="relative overflow-x-auto border border-border shadow-md
sm:rounded-lg">
16     <table className="w-full text-left text-sm text-muted-foreground">
17       <tbody>
18         {students.data.map((student, i) => (
19           <tr key={student.id} className="border-b bg-card">
20             /* Menampilkan hanya atribut profil siswa */
21             <td className="px-6 py-4">{student.nisn || '-'}</td>
22             <td className="px-6 py-4 font-semibold">
23               {student.full_name}</td>
24               <td className="px-6 py-4">{student.student_class?.name
25                 || '-'}</td>
26               /* Tidak ada akses Mutasi (Edit/Delete), hanya Read
27                 (Detail) */
28               <td className="px-6 py-4 text-center">
29                 <Link href={route('teacher.students.show',
30                   student.id)}>
31                   <Button variant="secondary" size="sm"
32                     className="h-8">
33                       <Eye className="mr-1.5 h-3.5 w-3.5" />
34                       Detail
35                     </Button>
36                   </Link>
37                 </td>
38             </tr>
39           </td>
40         ))}
41       </tbody>
42     </table>
43   </div>
44 );

```

Gambar 5.118 Code Frontend Daftar Siswa

Potongan kode pada Gambar 5.118 memperlihatkan antarmuka tabel *TeacherStudentsIndex* menggunakan React JS. Secara struktural, arsitektur *Frontend* ini dimodifikasi menjadi *Read-Only Component* tombol Tambah, Edit, dan Hapus dihilangkan sepenuhnya, digantikan dengan tombol Detail tunggal (*<Eye />*). Selain itu, implementasi penyaringan/pencarian datanya menerapkan fitur canggih dari pustaka *Inertia.js* yaitu SPA (*Single Page Application*) *State Preservation*. Saat guru mengetik nama siswa pada kotak pencarian (*handleSearchChange*), React mengirimkan request *asynchronous* ke peladen menggunakan parameter `{ preserveState: true, replace: true }`. Hal ini menginstruksikan peramban agar merekayasa URL (*history API*) dan me-*(render)* ulang isi tabel tabel tanpa mereset posisi scroll (guliran layar) dan tanpa melakukan *refresh* halaman (berkedip).

5. Tampilan halaman daftar siswa



Gambar 5.119 Halaman Daftar Siswa

Gambar 5.119 menunjukkan hasil implementasi halaman Daftar Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan menu Daftar Siswa Sekolah yang dikemas dalam bentuk tabel data dinamis yang memuat kolom nomor, NISN, nama lengkap, jenis kelamin (L/P), kelas, serta tombol aksi "Detail" untuk membuka rekapitulasi data profil mendalam masing-masing peserta didik. Selain itu, tersedia menu navigasi platform di bilah sisi kiri, filter *dropdown*

"Pilih T.A / Kelas", serta kolom isian pencarian pencocokan kata kunci nama atau NISN siswa secara langsung.

Tampilan dirancang dengan antarmuka yang sederhana sehingga memudahkan Guru dalam mengelola pengawasan informasi profil siswa secara lebih efektif. Selain itu, penataan layout kolom yang bersih serta penempatan tombol navigasi yang kontras sangat membantu mempercepat peninjauan latar belakang kontak orang tua atau wali murid. Dengan demikian, halaman Daftar Siswa tidak hanya berfungsi sebagai media penampil list kelompok belajar digital, tetapi juga membantu menjaga konsistensi keandalan akses informasi peserta didik pada seluruh rute transaksional lainnya di dalam sistem.

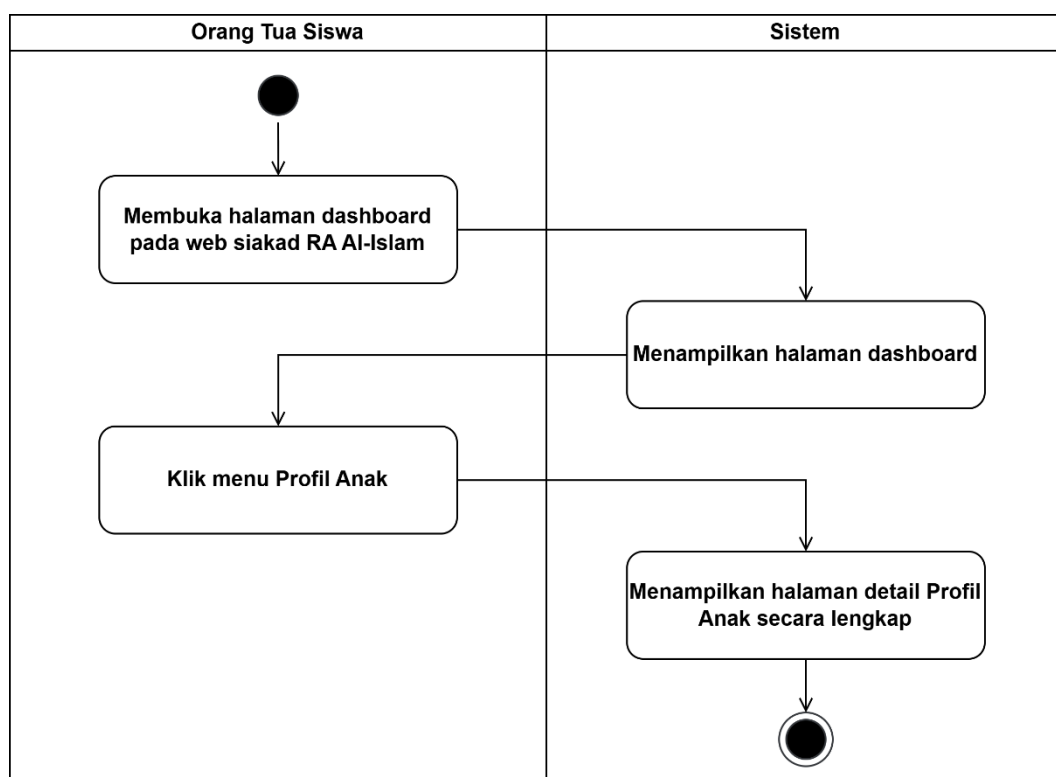
5.4.3.2 Profil siswa

Fitur Profil Siswa merupakan salah satu fitur yang dikembangkan pada *Sprint 3* untuk memberikan kemudahan kepada Orang Tua Siswa dalam mengakses informasi biodata anak secara langsung melalui sistem. Fitur ini berfungsi menampilkan identitas siswa beserta informasi pendukung yang telah tersimpan pada sistem sehingga Orang Tua Siswa dapat mengetahui data anak tanpa harus melihat dokumen administrasi sekolah secara langsung. Dengan adanya fitur ini, penyampaian informasi menjadi lebih mudah diakses, transparan, dan terintegrasi sehingga Orang Tua Siswa dapat memperoleh informasi yang dibutuhkan kapan saja sesuai dengan hak akses yang dimiliki.

Perancangan fitur Profil Siswa dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan aktivitas Orang Tua Siswa ketika mengakses halaman profil siswa, *Class Diagram* digunakan untuk menjelaskan struktur kelas yang berperan dalam pengelolaan data profil siswa beserta hubungan antar kelas yang terlibat, sedangkan *Sequence Diagram* digunakan untuk menggambarkan urutan komunikasi antara pengguna, sistem, dan basis data selama proses menampilkan informasi profil siswa berlangsung. Ketiga diagram tersebut menjadi acuan dalam proses implementasi

sehingga fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan sistem yang telah dirancang.

1. *Activity diagram* profil siswa



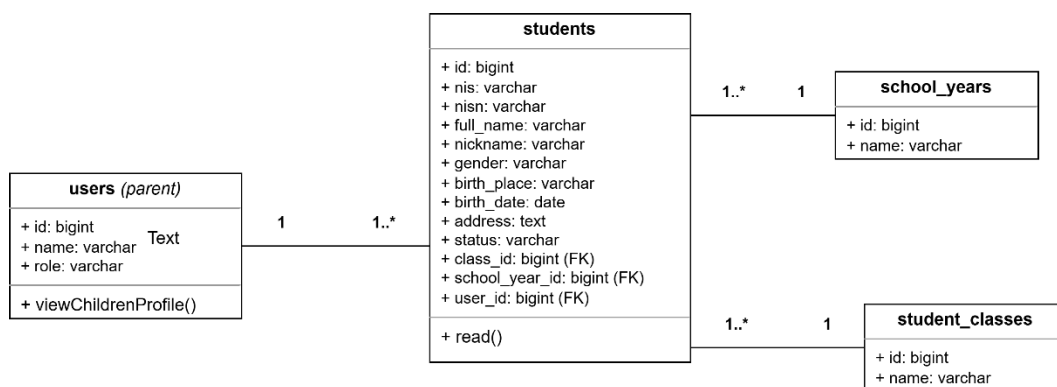
Gambar 5.120 *Activity Diagram* Melihat Profil Siswa

Activity Diagram Profil Siswa Orang Tua Siswa pada Gambar 5.120 menggambarkan alur proses peninjauan profil detail anak yang dilakukan oleh Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Orang Tua Siswa membuka Dashboard kemudian memilih menu Profil Anak sehingga sistem menerima instruksi permintaan penarikan data. Selanjutnya sistem memproses kueri pembacaan data dan menampilkan halaman detail Profil Anak secara lengkap pada layar antarmuka pengguna.

Apabila Orang Tua Siswa mengakses menu Profil Anak, sistem secara otomatis mengeksekusi validasi di sisi server untuk memastikan bahwa data yang ditarik hanya data siswa yang terhubung dengan ID akun orang tua yang sedang aktif.

Mekanisme pembacaan data ini dirancang secara terstruktur dan efisien agar penyajian biodata pribadi serta status akademik murid dapat langsung dibaca dengan jelas. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh alur penyajian detail profil anak dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* profil siswa



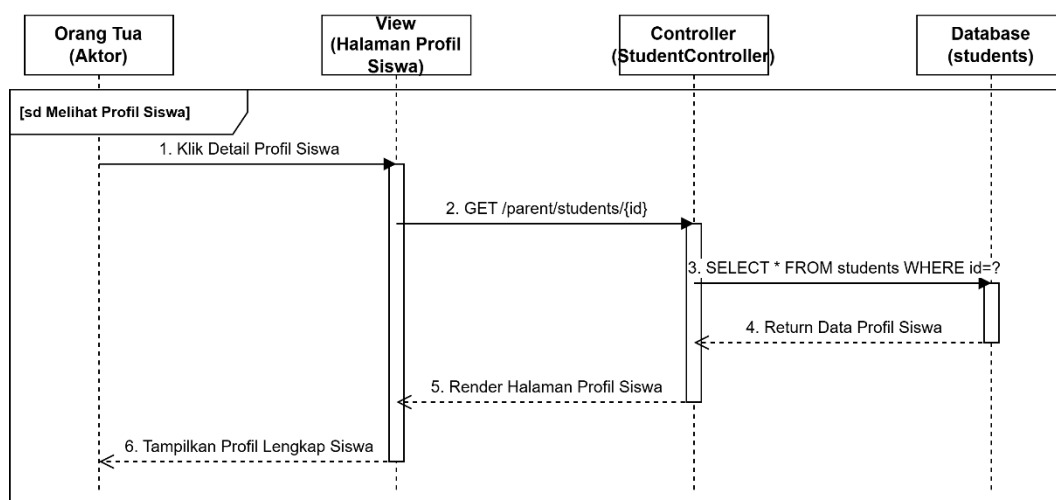
Gambar 5.121 *Class Diagram* Profil Siswa

Class Diagram Profil Siswa Orang Tua Siswa pada Gambar 5.121 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses penyajian profil siswa untuk aktor Orang Tua Siswa pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *students* yang memuat atribut *id*, *nis*, *nisn*, *full_name*, *nickname*, *gender*, *birth_place*, *birth_date*, *address*, *status*, *class_id* (FK), *school_year_id* (FK), dan *user_id* (FK) serta operasi *read()*, dengan kelas *users (parent)* yang memuat atribut *id*, *name*, *role* dan operasi *viewChildrenProfile()*, kelas *school_years* dengan atribut *id* dan *name*, serta kelas *student_classes* dengan atribut *id* dan *name*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam penyajian biodata anak dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* satu ke banyak (1 ke 1..*) dari kelas *users (parent)*, *school_years*, maupun *student_classes* terhadap entitas *students*. Selain itu, diagram ini juga membantu dalam menggambarkan pemetaan batasan hak akses melalui penguncian

foreign key user_id pada tabel *students*, sehingga setiap proses pembacaan data profil anak oleh akun orang tua dapat dijalankan secara konsisten dan aman sesuai aturan basis data pada sistem.

3. *Sequence diagram* profil siswa



Gambar 5.122 *Sequence Diagram* Melihat Profil Siswa

Sequence Diagram Profil Siswa Orang Tua Siswa pada Gambar 5.122 menggambarkan urutan komunikasi antara Orang Tua, antarmuka sistem, *controller*, dan basis data selama proses melihat detail profil siswa berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan detail profil dikirimkan oleh Orang Tua dari *View* (Halaman Profil Siswa), kemudian diproses oleh *StudentController* melalui metode *HTTP GET* dengan rute */parent/students/{id}* sebelum diteruskan ke *Database* (tabel *students*) menggunakan kueri *SQL SELECT * FROM students WHERE id=?* hingga menghasilkan respons berupa kembalian data profil siswa yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses penarikan data profil murid dapat dipahami secara lebih rinci dan berurutan sesuai lini waktu. Diagram ini menjadi acuan dalam implementasi logika penarikan data

di sisi server sehingga proses pertukaran data antara komponen presentasi *front-end* React JS, *controller* Laravel, dan basis data *MySQL* dapat berjalan secara terstruktur, menjaga keamanan relasi data, serta sesuai dengan kebutuhan performa aplikasi yang telah dirancang sebelumnya.

4. Implementasi profil siswa

```

1  /**
2   * Menampilkan detail profil siswa khusus untuk orang tua.
3   */
4  public function show(Student $student)
5  {
6      // Strict Ownership Authorization (Pencegahan IDOR)
7      // Memastikan orang tua HANYA bisa melihat data anak kandungnya sendiri
8      if ($student->user_id !== auth()->id()) {
9          abort(403, 'Anda tidak memiliki akses untuk melihat data siswa ini.');
```

```

10     }
11
12     // Eager Loading relasi untuk menghindari N+1 Query Problem
13     $student->load(['studentClass.schoolYear', 'parentUser']);
14
15     return Inertia::render('parent/students/show', [
16         'student' => $student,
17     ]);
18 }
19
20 /**
21 * Cetak kartu identitas siswa (PDF Streaming)
22 */
23 public function cardPdf(Student $student)
24 {
25     // Validasi kepemilikan juga diterapkan pada fitur cetak kartu
26     if ($student->user_id !== auth()->id()) {
27         abort(403, 'Anda tidak memiliki akses.');
```

```

28     }
29
30     $student->load(['studentClass', 'parentUser']);
31     $logoBase64 =
base64_encode(file_get_contents(public_path('images/logo_alislam.png')));
32
33     // Server-Side Rendering PDF
34     $pdf = Pdf::loadView('student-card-pdf', [
35         'student' => $student,
36         'logoBase64' => $logoBase64,
37     ]->setPaper([0, 0, 240.94, 153.07], 'landscape');
38
39     return $pdf->stream('kartu-siswa-' . $student->id . '.pdf');
```

```

40 }

```

Gambar 5.123 Code Backend Profil Siswa

Potongan kode pada Gambar 5.123 menunjukkan metode *show()* dan *cardPdf()* di dalam *Parent/StudentController*. Karena fitur ini berhubungan dengan data privasi

anak di bawah umur, sistem mengimplementasikan teknik *Strict Ownership Authorization* (Otorisasi Kepemilikan Ketat). Sebelum data dikirim ke layar, peladen (*server*) melakukan pengecekan logika `if ($student->user_id !== auth()->id()) { abort(403); }`. Pengecekan ini secara absolut menutup kerentanan keamanan IDOR (*Insecure Direct Object Reference*). Artinya, jika seorang wali murid mencoba meretas URL dengan mengganti ID Siswa milik anak orang lain (misal: `/parent/students/5` diganti menjadi `/parent/students/6`), sistem akan menolak akses tersebut dengan melemparkan kode *Error 403 Forbidden*.

```

1 // Dynamic Conditional Rendering: Kalkulasi umur secara on-the-fly di peramban
2 const calculateAge = (dob: string | null) => {
3   if (!dob) return '-';
4   const birthDate = new Date(dob);
5   const today = new Date();
6   let age = today.getFullYear() - birthDate.getFullYear();
7   const m = today.getMonth() - birthDate.getMonth();
8   if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
9     age--;
10  }
11  return `${age} tahun`;
12 };
13
14 return (
15   {/* Tampilan Visual Profil (Code Reusability) */}
16   <div className="flex-1 pb-2">
17     <h1 className="text-xl font-bold text-gray-900 sm:text-3xl dark:text-
18       white">
19       {student.full_name}
20     </h1>
21     <p className="mt-1 flex items-center gap-2 text-xs font-medium text-
22       gray-500">
23       <span>NISN: {student.nisn || '-'}</span>
24       <span className="text-gray-300">|</span>
25       <span>NIS: {student.nis || '-'}</span>
26     </p>
27   </div>
28
29   {/* Rendering Dinamis Berdasarkan Data Database */}
30   <div>
31     <dt className="text-xs font-medium text-gray-500">Tanggal Lahir (Umur)
32   </dt>
33
34   <dd className="mt-1 text-sm font-medium text-gray-900">
35     {student.date_of_birth
36       ? new Date(student.date_of_birth).toLocaleDateString('id-ID',
37         {
38           day: 'numeric', month: 'long', year: 'numeric',
39         })
40       : '-'}
41   </dd>
42
43   <span className="ml-1 text-gray-500">
44     {/* Memanggil helper kalkulasi umur */}
45     ({calculateAge(student.date_of_birth)})
46   </span>
47 </div>
48 );

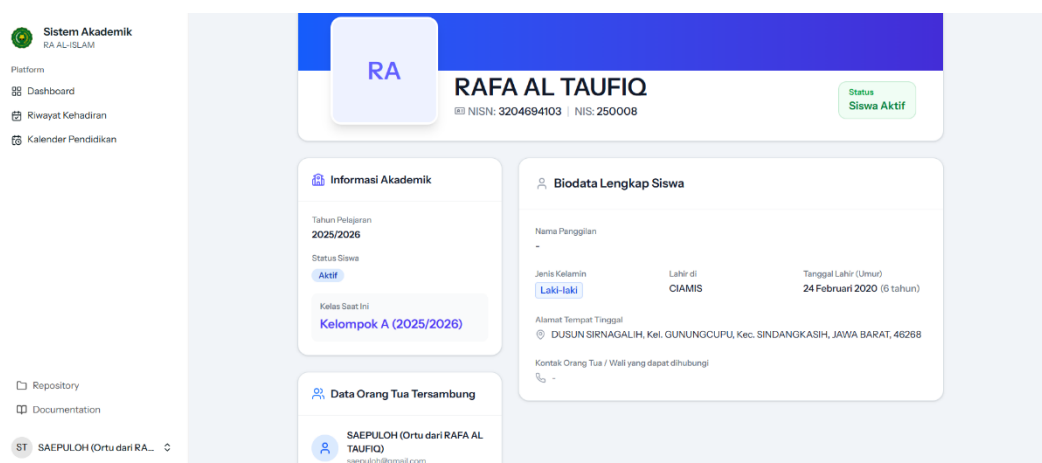
```

Gambar 5.124 Code Frontend Profil Siswa

Potongan kode pada Gambar 5.124 memperlihatkan implementasi *Frontend* menggunakan React JS (*parent/students/show.tsx*). Antarmuka ini dirancang dengan prinsip *Dynamic Conditional Rendering*, di mana sistem dapat mengolah data mentah menjadi informasi yang ramah pengguna. Contohnya adalah metode *calculateAge()*, yang secara cerdas (*on-the-fly*) menghitung umur anak saat ini berdasarkan kolom tanggal lahir, sehingga orang tua bisa melihat "(6 tahun)" di

samping tanggal lahir. Selain itu, tata letak *grid* dan kartu profil di halaman ini mendemonstrasikan prinsip *Code Reusability* (Penggunaan Ulang Kode). UI yang disajikan untuk Orang Tua memiliki struktur komponen visual yang identik dengan yang digunakan oleh Aktor Guru, sehingga menghemat ukuran beban aplikasi (*bundle size*) dan mempercepat proses pemuatan halaman (SPA).

5. Tampilan halaman profil siswa



Gambar 5.125 Halaman Profil Siswa

Gambar 5.125 menunjukkan hasil implementasi halaman Profil Siswa pada antarmuka akun Orang Tua Siswa di Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan kartu header utama berisi avatar identitas "RA", nama lengkap siswa (seperti "RAFA AL TAUFIQ"), nomor NISN/NIS, serta indikator *badge* hijau "Status Siswa Aktif". Selain itu, halaman ini dilengkapi dengan tiga panel kartu informasi terpisah, yaitu kartu "Informasi Akademik" (menampilkan tahun pelajaran, status siswa, dan kelas berjalan), kartu "Biodata Lengkap Siswa" (memuat nama panggilan, jenis kelamin, tempat/tanggal lahir, umur, alamat, dan nomor kontak), serta kartu "Data Orang Tua Tersambung" yang menampilkan nama dan email wali murid.

Tampilan dirancang dengan antarmuka yang bersih dan teratur sehingga memudahkan Orang Tua Siswa dalam membaca seluruh rincian biodata anak secara efektif. Penataan tata letak berbasis kartu (*card layout*) dan pemilihan warna kontras

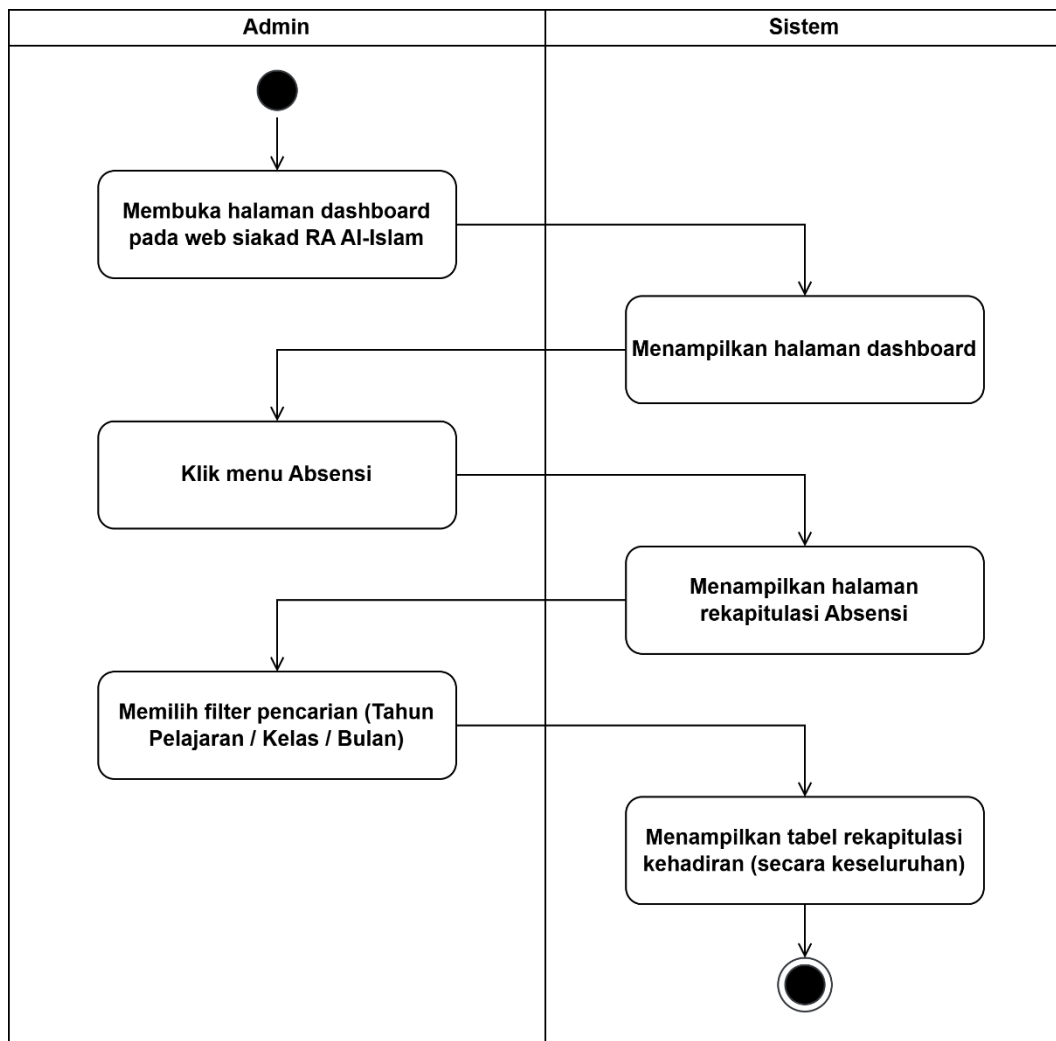
pada indikator status sangat membantu wali murid dalam melakukan pemantauan identitas administratif anak dari rumah. Dengan demikian, halaman Profil Siswa tidak hanya berfungsi sebagai media penyajian data identitas pribadi, tetapi juga mendukung keandalan transparansi informasi peserta didik di dalam sistem.

5.4.3.3 Absensi (rekap admin)

Fitur Absensi (Rekap Admin) merupakan salah satu fitur yang dikembangkan pada *Sprint* 3 untuk membantu Admin dalam melakukan rekapitulasi data kehadiran siswa yang telah diinput oleh Guru. Fitur ini berfungsi menampilkan seluruh data absensi siswa dalam bentuk rekap berdasarkan periode tertentu sehingga Admin dapat memantau tingkat kehadiran siswa secara lebih mudah dan terstruktur. Dengan adanya fitur ini, proses pengelolaan data absensi tidak lagi dilakukan secara manual karena seluruh data kehadiran telah tersimpan dalam basis data dan dapat ditampilkan kembali sesuai kebutuhan. Selain meningkatkan efisiensi pengelolaan data, fitur ini juga membantu menjaga ketepatan informasi yang digunakan sebagai dasar penyusunan laporan akademik maupun administrasi sekolah.

Perancangan fitur Absensi (Rekap Admin) dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan aktivitas Admin ketika mengakses rekapitulasi data absensi siswa, *Class Diagram* digunakan untuk menjelaskan struktur kelas yang berperan dalam pengelolaan data absensi beserta hubungan antar kelas yang terlibat, sedangkan *Sequence Diagram* digunakan untuk menggambarkan urutan komunikasi antara pengguna, sistem, dan basis data selama proses menampilkan rekap absensi berlangsung. Ketiga diagram tersebut menjadi acuan dalam proses implementasi sehingga fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan sistem yang telah dirancang.

1. *Activity diagram* absensi (rekap admin)

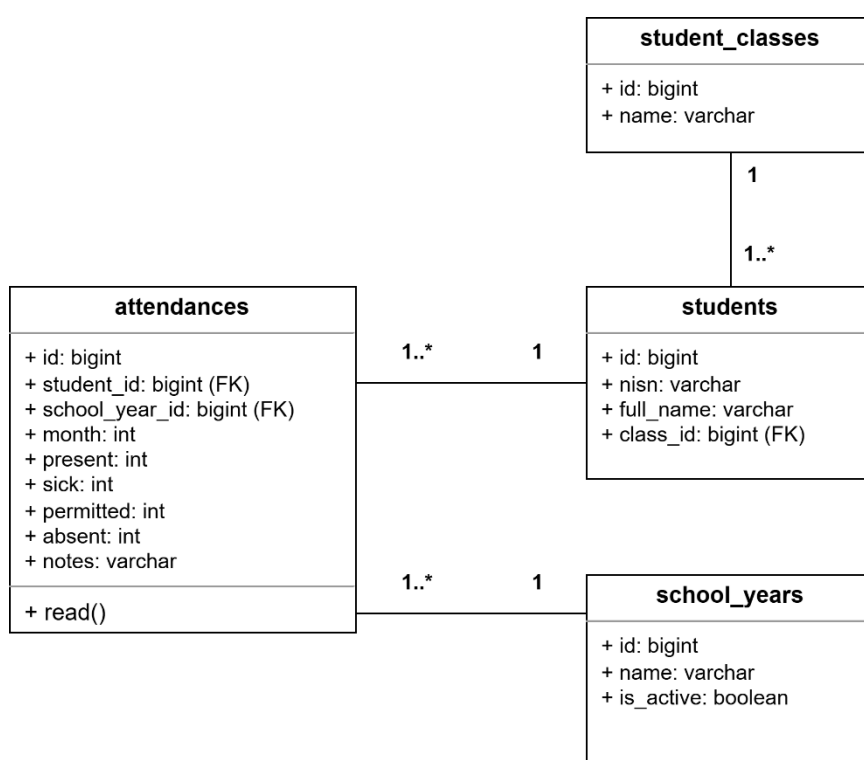


Gambar 5.126 *Activity Diagram* Absensi (Rekap Admin)

Activity Diagram Rekapitulasi Absensi pada Gambar 5.126 menggambarkan alur proses pemantauan rekapitulasi kehadiran siswa secara menyeluruh yang dilakukan oleh Admin pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Admin membuka Dashboard kemudian memilih menu Absensi sehingga sistem menampilkan halaman rekapitulasi Absensi. Selanjutnya Admin memilih filter pencarian (Tahun Pelajaran / Kelas / Bulan) untuk menyaring data yang dibutuhkan.

Setelah Admin memilih filter pencarian, sistem secara otomatis memproses parameter tersebut dan menampilkan tabel rekapitulasi kehadiran siswa secara keseluruhan. Proses ini memberikan kemudahan bagi pengelola untuk memperoleh rekap data presensi bulanan tanpa harus mengalkulasi akumulasi secara manual. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme penyajian rekapitulasi presensi dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* absensi (rekap admin)



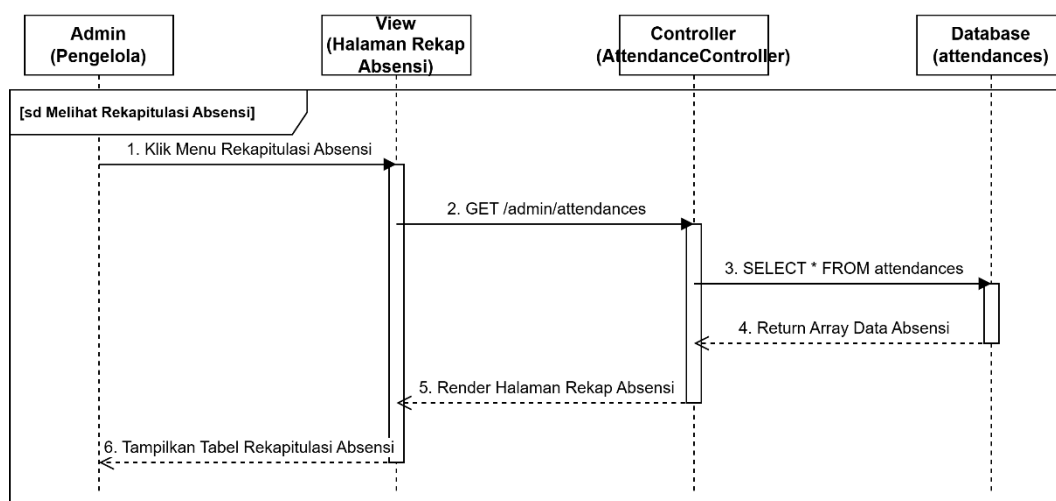
Gambar 5.127 *Class Diagram* Absensi (Rekap Admin)

Class Diagram Rekapitulasi Absensi pada Gambar 5.127 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses kalkulasi dan penyajian rekapitulasi presensi siswa pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *attendances* yang memiliki atribut *id*, *student_id* (FK), *school_year_id* (FK), *month*, *present*, *sick*, *permitted*, *absent*, dan *notes* serta operasi *read()*, dengan kelas *students* yang memuat atribut *id*, *nisn*, *full_name*,

class_id (FK), kelas *student_classes* dengan atribut *id*, *name*, serta kelas *school_years* dengan atribut *id*, *name*, dan *is_active*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam pengagregasian data presensi dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* di mana kelas *students* dan *school_years* masing-masing memiliki relasi satu ke banyak (1 ke 1..*) terhadap tabel *attendances*. Selain itu, diagram ini juga membantu dalam menggambarkan pemetaan kunci tamu (*student_id*, *school_year_id*, *class_id*) sehingga proses agregasi akumulasi nilai kehadiran per bulan dapat dieksekusi secara efisien dan konsisten sesuai kebutuhan integritas basis data pada sistem.

3. *Sequence diagram* absensi (rekap admin)



Gambar 5.128 *Sequence Diagram* Absensi (Rekap Admin)

Sequence Diagram Rekapitulasi Absensi pada Gambar 5.128 menggambarkan urutan komunikasi antara Admin, antarmuka sistem, *controller*, dan basis data selama proses melihat rekapitulasi absensi berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan rekap data kehadiran dikirimkan oleh Admin dari *View* (Halaman Rekap Absensi), kemudian diproses oleh *AttendanceController* melalui metode *HTTP GET* dengan rute */admin/attendances* sebelum diteruskan ke *Database* (tabel *attendances*) menggunakan kueri *SQL*

*SELECT * FROM attendances* hingga menghasilkan *respons* berupa kembalian data array absensi yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengolahan dan penyajian rekapitulasi kehadiran siswa dapat dipahami secara lebih rinci dan berurutan. Diagram ini menjadi acuan dalam implementasi logika penarikan data di sisi server sehingga proses pertukaran data antara antarmuka front-end React JS, *controller* Laravel, dan basis data *MySQL* dapat berjalan secara terstruktur, memicu pembaruan perenderan komponen tabel (Render Halaman Rekap Absensi), menjaga konsistensi data, serta menampilkan rekapitulasi secara akurat sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi absensi (rekap admin)

```

1  /**
2   * Menampilkan daftar rekapitulasi absensi bulanan.
3   */
4  public function index(Request $request)
5  {
6      $search = $request->input('search');
7      $filterClassId = $request->input('class_id');
8
9      // Penentuan Parameter Waktu Dinamis (Default ke Bulan Berjalan)
10     $monthStr = $request->input('month');
11     $activeMonth = $monthStr ? (int) $monthStr : (int) now()->format('n');
12
13     $activeSchoolYear = $request->input('school_year_id'); // [Disempurnakan]
14
15     // Eksekusi Kueri dengan Constrained Eager Loading
16     $students = Student::query()
17         ->when($search, function ($query, $search) {
18             // ... logika pencarian teks (NISN / Nama) ...
19         })
20         ->when($filterClassId, function ($query, $filterClassId) {
21             $query->where('class_id', $filterClassId);
22         })
23         // Memfilter Data Induk (Siswa) Berdasarkan Tahun Pelajaran
24         ->where(function ($query) use ($activeSchoolYear, $activeMonth) {
25             $query->whereHas('studentClass', function ($q) use ($activeSchoolYear)
26             {
27                 $q->where('school_year_id', $activeSchoolYear);
28             });
29         })
30         // Constrained Eager Loading: Membatasi relasi absensi HANYA pada bulan
31         terpilih
32         ->with(['attendances' => function ($query) use ($activeSchoolYear,
33             $activeMonth) {
34                 $query->where('school_year_id', $activeSchoolYear)
35                 ->where('month', $activeMonth);
36             }])
37         ->orderBy($request->input('sort', 'full_name'), $request-
38             >input('direction', 'asc'))
39         ->get();
40
41     return Inertia::render('admin/attendances/index', [
42         'students' => $students,
43         'activeMonth' => $activeMonth,
44         'search' => $search,
45     ]);
46 }

```

Gambar 5.129 Code Backend Absensi (Rekap Admin)

Potongan kode pada Gambar 5.129 menunjukkan metode *index()* di dalam *Admin/AttendanceController*. Karena fitur rekapitulasi absensi ini mengolah ratusan baris data siswa setiap bulannya, sistem mengimplementasikan teknik *Constrained Eager Loading* (Pemuatan Relasi Data Terbatas). Sebelum data

dikirim ke layar, peladen (server) membatasi pemanggilan data absensi melalui logika `with(['attendances' => function ($query) use ($activeMonth)...])`. Pembatasan ini secara absolut menutup kerentanan performa aplikasi yang dikenal sebagai N+1 Query Problem (Kelebihan Beban Kueri Basis Data). Artinya, jika seorang Admin memfilter rekapitulasi untuk bulan Agustus, sistem akan menolak pemanggilan data riwayat absensi dari bulan-bulan lain yang tidak relevan, sehingga menghemat pemakaian memori peladen secara signifikan dan halaman dapat dimuat dengan sangat cepat.

```

1 // SPA Combined Filtering Handler
2 const handleCombinedFilterChange = (e: React.ChangeEvent<HTMLSelectElement>) => {
3   const val = e.target.value;
4   // ... logika pengambilan nilai class_id dan school_year_id ...
5
6   router.get(
7     route('admin.attendances.index'),
8     { search, school_year_id: sy_id, class_id: c_id, month: filterMonth },
9     { preserveState: true, replace: true } // Mencegah reload halaman penuh
10  );
11 };
12
13 return (
14   <tbody>
15     {students.map((student, index) => (
16       <tr key={student.id} className="border-b bg-card">
17         <td className="px-6 py-3">{student.nisn || '-'}</td>
18         <td className="px-6 py-3">{student.full_name}</td>
19
20         { /* Optional Chaining & Nullish Coalescing mencegah error jika
absen kosong */ }
21         <td className="px-4 py-2 font-semibold">
22           {student.attendances?.[0]?.present ?? 0}
23         </td>
24         <td className="px-4 py-2 font-semibold">
25           {student.attendances?.[0]?.sick ?? 0}
26         </td>
27         <td className="px-4 py-2 font-semibold">
28           {student.attendances?.[0]?.permitted ?? 0}
29         </td>
30         <td className="px-4 py-2 font-semibold">
31           {student.attendances?.[0]?.absent ?? 0}
32         </td>
33       </tr>
34     ))}
35   </tbody>
36 );

```

Gambar 5.130 Code Frontend Absensi (Rekap Admin)

Potongan kode pada Gambar 5.130 memperlihatkan implementasi *Frontend* pada antarmuka `admin/attendances/index.tsx`. Antarmuka ini dirancang dengan komponen panel *Read-Only* (Hanya Baca) yang mengusung prinsip *SPA Combined Filtering* (Penyaringan Terintegrasi Berbasis Aplikasi Halaman Tunggal). Ketika Admin merubah pilihan filter pada menu tarik-turun (Tahun Pelajaran, Kelas, atau Bulan), sistem secara mulus menyuntikkan parameter baru ke rute peladen menggunakan fungsi `router.get(..., { preserveState: true })`. Konfigurasi ini memastikan aplikasi beroperasi merender ulang tabel kehadiran tanpa perlu memuat ulang keseluruhan kerangka halaman (page reload). Selain itu, pelekatan variabel menggunakan operator opsional `{student.attendances?.[0]?.present ?? 0}` sangat krusial untuk mencegah kerentanan layar putih (*fatal error screen*) apabila relasi *array* data absensi bulan tersebut belum pernah diinput oleh guru kelas bersangkutan.

5. Tampilan halaman absensi (rekap admin)

No	NISN	Nama Lengkap	L/P	Hadir	Sakit	Ijin	Alpa
1	3185384514	ABIDZARD HILMI FAUZIANSYAH	L	15	0	0	0
2	3200194909	AKHSAN RAMADHAN	L	15	0	0	0
3	3191564790	AL AYYUBY FAIZ	L	15	0	0	0
4	3186404777	ALESHA MAULIDA	P	15	0	0	0
5	3186404777	AI INNA TRI WULI ANDARY	P	15	0	0	0

Gambar 5.131 Halaman Absensi (Rekap Admin)

Gambar 5.131 menunjukkan hasil implementasi halaman Rekapitulasi Kehadiran Bulanan pada antarmuka Admin di Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan panel "Rekapitulasi Kehadiran Bulanan" yang memuat ringkasan data absensi siswa yang dikemas dalam bentuk tabel terstruktur dengan kolom nomor, NISN, nama lengkap, jenis kelamin (L/P), serta rekap jumlah akumulasi Hadir, Sakit, Ijin, dan Alpa. Selain itu, tersedia filter

pilihan kelas/tahun ajaran, filter bulan (seperti "Juni"), serta kolom pencarian nama atau NISN siswa sebagai fasilitas bagi Admin untuk memantau data presensi.

Tampilan dirancang dengan antarmuka yang bersih dan teratur sehingga memudahkan Admin dalam meninjau rekapitulasi kehadiran siswa secara waktu nyata (*real-time*). Penataan tata letak filter pencarian yang intuitif di atas tabel memudahkan pengelola dalam melakukan penyaringan data akademik berdasarkan periode tertentu secara lebih cepat. Dengan demikian, halaman Rekapitulasi Absensi tidak hanya berfungsi sebagai media penyajian laporan presensi bulanan, tetapi juga membantu menjaga keandalan pengawasan kedisiplinan siswa di dalam sistem.

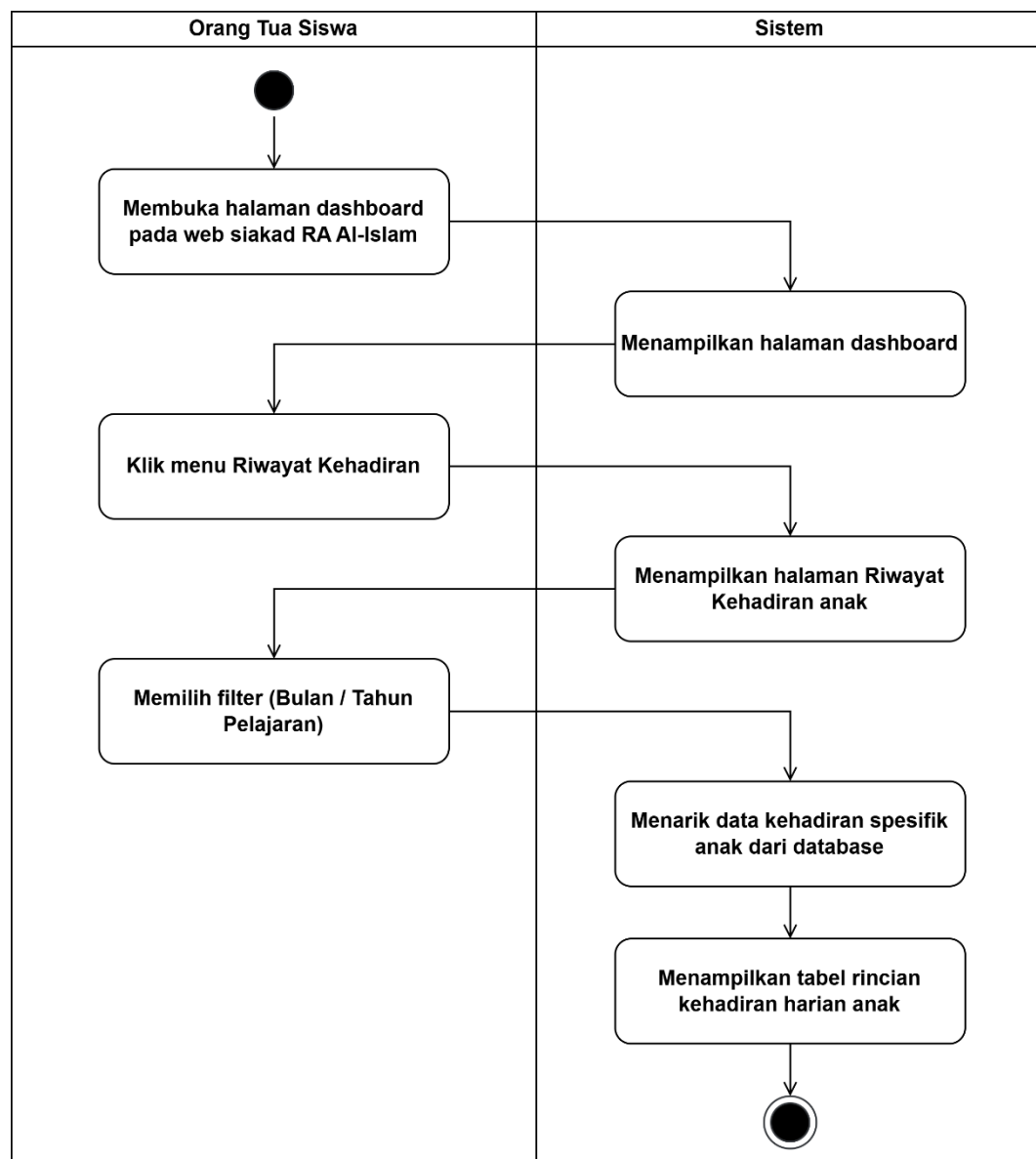
5.4.3.4 Riwayat kehadiran

Fitur Riwayat Kehadiran merupakan salah satu fitur yang dikembangkan pada *Sprint 3* untuk memberikan kemudahan kepada Orang Tua Siswa dalam memantau riwayat kehadiran anak secara langsung melalui sistem. Fitur ini berfungsi menampilkan informasi kehadiran siswa yang telah dicatat oleh Guru sehingga Orang Tua Siswa dapat mengetahui riwayat kehadiran anak secara lebih mudah, cepat, dan transparan tanpa harus menunggu penyampaian informasi dari pihak sekolah. Dengan adanya fitur ini, proses penyampaian informasi mengenai kehadiran siswa menjadi lebih efektif karena seluruh data yang ditampilkan berasal dari hasil pencatatan absensi yang telah tersimpan pada sistem. Selain itu, fitur ini juga mendukung peningkatan komunikasi antara sekolah dan Orang Tua Siswa melalui penyediaan informasi akademik yang dapat diakses sesuai dengan hak akses pengguna.

Perancangan fitur Riwayat Kehadiran dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan aktivitas Orang Tua Siswa ketika mengakses riwayat kehadiran anak, *Class Diagram* digunakan untuk menjelaskan struktur kelas yang berperan dalam pengelolaan data kehadiran beserta hubungan antar kelas yang terlibat,

sedangkan *Sequence Diagram* digunakan untuk menggambarkan urutan komunikasi antara pengguna, sistem, dan basis data selama proses menampilkan riwayat kehadiran berlangsung. Ketiga diagram tersebut menjadi acuan dalam proses implementasi sehingga fitur yang dikembangkan dapat berjalan sesuai dengan kebutuhan sistem yang telah dirancang.

1. *Activity diagram* riwayat kehadiran

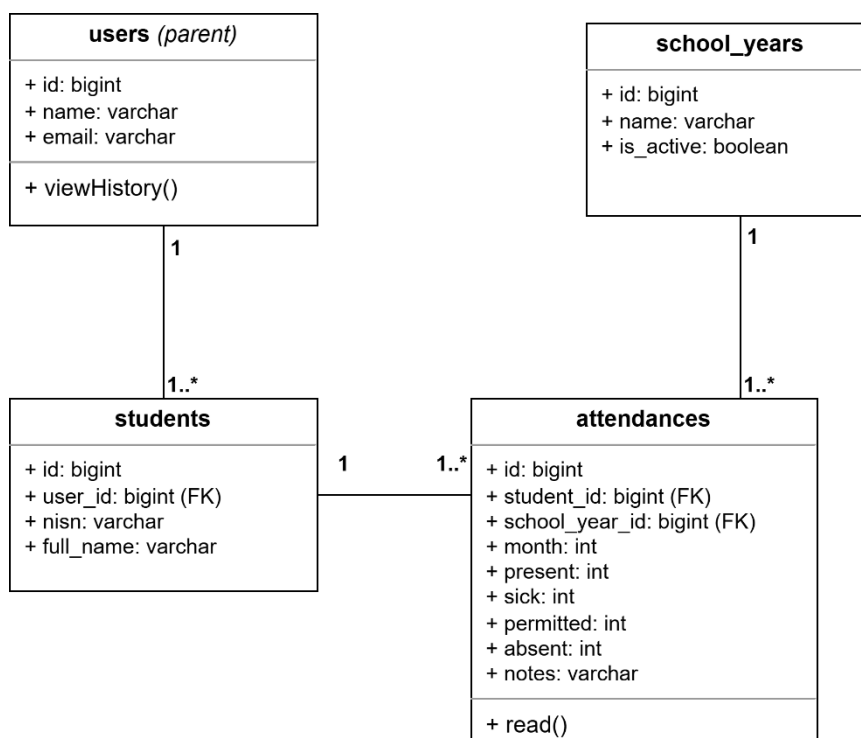


Gambar 5.132 *Activity Diagram* Riwayat Kehadiran

Activity Diagram Riwayat Kehadiran Orang Tua Siswa pada Gambar 5.132 menggambarkan alur proses pemantauan presensi anak yang dilakukan oleh Orang Tua Siswa pada Sistem Informasi Akademik RA Al-Islam Gunungcupu. Proses diawali ketika Orang Tua Siswa membuka Dashboard kemudian memilih menu Riwayat Kehadiran sehingga sistem menampilkan halaman Riwayat Kehadiran anak. Selanjutnya Orang Tua Siswa memilih filter (Bulan / Tahun Pelajaran) untuk menyaring informasi presensi yang ingin ditinjau.

Setelah Orang Tua Siswa memilih parameter filter, sistem secara otomatis mengeksekusi proses penarikan data kehadiran spesifik anak dari database berdasarkan relasi ID akun orang tua. Sistem kemudian merender data tersebut dan menampilkan tabel rincian kehadiran harian anak secara terstruktur pada layar antarmuka pengguna. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh mekanisme penyajian riwayat presensi dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

2. *Class diagram* riwayat kehadiran



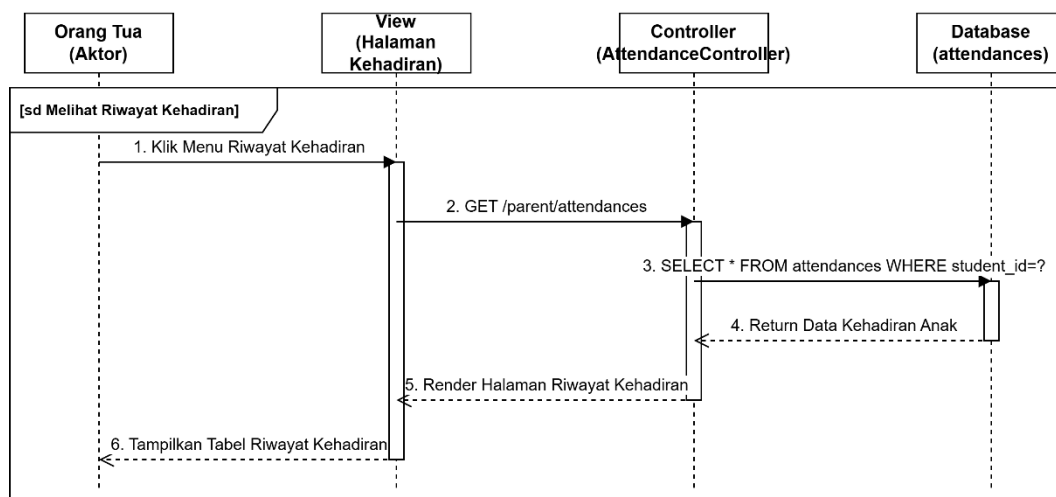
Gambar 5.133 *Class Diagram* Riwayat Kehadiran

Class Diagram Riwayat Kehadiran Orang Tua Siswa pada Gambar 5.133 digunakan untuk menggambarkan struktur relasional objek yang berperan dalam proses penyajian riwayat presensi anak pada sistem. Diagram ini menunjukkan interaksi terintegrasi antara kelas *users (parent)* yang memiliki atribut *id*, *name*, *email* dan operasi *viewHistory()*, kelas *students* yang memuat atribut *id*, *user_id (FK)*, *nisn*, *full_name*, kelas *school_years* dengan atribut *id*, *name*, *is_active*, serta kelas transaksional *attendances* yang memiliki atribut *id*, *student_id (FK)*, *school_year_id (FK)*, *month*, *present*, *sick*, *permitted*, *absent*, *notes* beserta operasi *read()*.

Melalui *Class Diagram*, struktur data dan korelasi antar-entitas dalam penyajian data presensi anak dapat dirancang secara lebih sistematis, seperti kejelasan hubungan *multiplicity* satu ke banyak (1 ke 1..*) antara kelas *users (parent)* terhadap *students*, serta dari *students* dan *school_years* terhadap kelas *attendances*. Selain itu, diagram ini membantu dalam menggambarkan pemetaan kunci tamu

(*user_id*, *student_id*, *school_year_id*) sehingga proses penarikan rekapitulasi kehadiran anak oleh akun orang tua dapat dieksekusi secara aman, konsisten, dan terisolasi sesuai aturan hak akses pada sistem.

3. *Sequence diagram* riwayat kehadiran



Gambar 5.134 *Sequence Diagram* Riwayat Kehadiran

Sequence Diagram Riwayat Kehadiran Orang Tua Siswa pada Gambar 5.134 menggambarkan urutan komunikasi antara Orang Tua, antarmuka sistem, *controller*, dan basis data selama proses melihat riwayat kehadiran berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data kehadiran dikirimkan oleh Orang Tua dari *View* (Halaman Kehadiran), kemudian diproses oleh *AttendanceController* melalui metode *HTTP GET* dengan rute */parent/attendances* sebelum *attendances* diteruskan ke *Database* (tabel) menggunakan kueri *SQL SELECT * FROM attendances WHERE student_id=?* hingga menghasilkan respons berupa kembalian data kehadiran anak yang akan ditampilkan kembali kepada pengguna.

Melalui *Sequence Diagram*, interaksi antarobjek yang terlibat pada proses pengolahan dan penyajian riwayat presensi siswa dapat dipahami secara lebih rinci dan berurutan sesuai alur waktu. Diagram ini menjadi acuan dalam implementasi logika penarikan data di sisi server sehingga proses pertukaran data antara

antarmuka front-end React JS, *controller* Laravel, dan basis data *MySQL* dapat berjalan secara terstruktur, memicu pembaruan perenderan komponen (*Render Halaman Riwayat Kehadiran*), menjaga keamanan data anak, serta menampilkan riwayat presensi secara akurat sesuai dengan kebutuhan yang telah dirancang sebelumnya.

4. Implementasi riwayat kehadiran

```

1  public function index(Request $request)
2  {
3      $user = auth()->user();
4
5      // Memastikan orang tua hanya bisa melihat anak kandungnya
6      $children = Student::where('user_id', $user->id)->get(['id', 'full_name',
7      'nispn']);
8      $schoolYears = SchoolYear::orderBy('name', 'desc')->get(['id', 'name',
9      'is_active']);
10
11      $selectedStudentId = $request->query('student_id', $children->first()->id);
12      $activeSchoolYear = $request->query('school_year_id');
13
14      $summary = ['present' => 0, 'sick' => 0, 'permission' => 0, 'absent' => 0];
15
16      if ($selectedStudentId) {
17          // Strict Ownership Authorization: Cek silang kepemilikan
18          $isValidChild = $children->contains('id', $selectedStudentId);
19
20          if ($isValidChild) {
21              $allRecordsThisYear = Attendance::where('student_id',
22              $selectedStudentId)
23              ->where('school_year_id', $activeSchoolYear)
24              ->get();
25
26              // Data Aggregation: Perhitungan otomatis sisi peladen
27              $summary['present'] = $allRecordsThisYear->sum('present');
28              $summary['sick'] = $allRecordsThisYear->sum('sick');
29              $summary['permission'] = $allRecordsThisYear->sum('permitted');
30              $summary['absent'] = $allRecordsThisYear->sum('absent');
31          } else {
32              abort(403); // Blokir paksa jika mencoba melihat absensi anak lain
33          }
34      }
35
36      return Inertia::render('parent/attendances/index', [
37          'children' => $children,
38          'summary' => $summary,
39          // ...
40      ]);
41  }

```

Gambar 5.135 *Code Backend Riwayat Kehadiran*

Potongan kode pada Gambar 5.135 menunjukkan metode *index()* di dalam *Parent/AttendanceController*. Karena fitur ini bersifat sangat rahasia bagi masing-masing keluarga, sistem mengimplementasikan validasi keamanan tingkat tinggi melalui teknik *Strict Ownership Authorization* (Otorisasi Kepemilikan Ketat). Sebelum peladen (*server*) melakukan kueri (*query*) riwayat kehadiran, sistem terlebih dahulu mendeteksi identitas pengguna yang sedang login, memetakan data anak kandungnya melalui fungsi *Student::where('user_id', \$user->id)*, dan melakukan verifikasi silang logika *if (\$isValidChild)*. Pengecekan ini secara absolut menutup kerentanan IDOR (*Insecure Direct Object Reference*), sehingga orang tua tidak akan pernah bisa mengakses atau melihat riwayat absensi anak lain meskipun mereka mencoba meretas parameter URL browser. Setelah lolos validasi, peladen akan melakukan pengelompokan data (*Data Aggregation*) untuk menghitung total kalkulasi jumlah kehadiran secara otomatis melalui fungsi agregat bawaan seperti *sum('present')*.

```

1 // SPA Route Injector: Memuat data tanpa merefresh layar
2 const handleFilterChange = (key: string, value: string) => {
3   router.get(
4     '/parent/attendances',
5     { ...filters, [key]: value },
6     { preserveState: true, preserveScroll: true, replace: true },
7   );
8 };
9
10 // Client-side Analytics: Menghitung metrik total hari belajar
11 const totalDays = summary.present + summary.sick + summary.permission +
    summary.absent;
12
13 return (
14   <div className="grid grid-cols-2 gap-4 lg:grid-cols-4">
15     {/* Kartu Statistik Kehadiran Visual */}
16     <div className="rounded-xl border border-gray-100 bg-white p-5 shadow-sm">
17       <div className="flex items-center justify-between">
18         <h3 className="text-sm font-medium text-gray-600">Total Hadir</h3>
19       </div>
20       <div className="mt-2 flex items-baseline gap-2">
21         <div className="text-3xl font-bold">{summary.present}</div>
22         <span className="text-sm font-medium text-gray-500">Hari</span>
23       </div>
24
25       {/* Kalkulasi Persentase Kehadiran Dinamis (Matematika) */}
26       {totalDays > 0 && (
27         <div className="mt-2.5 flex items-center text-xs font-medium text-
28           emerald-600">
29           <span className="bg-emerald-50 px-2 py-0.5 rounded-full">
30             {Math.round((summary.present / totalDays) * 100)}%
31           </span>
32         </div>
33       )}
34     </div>
35     {/* ... Kartu Sakit, Izin, Alpa ... */}
36   </div>
37 );

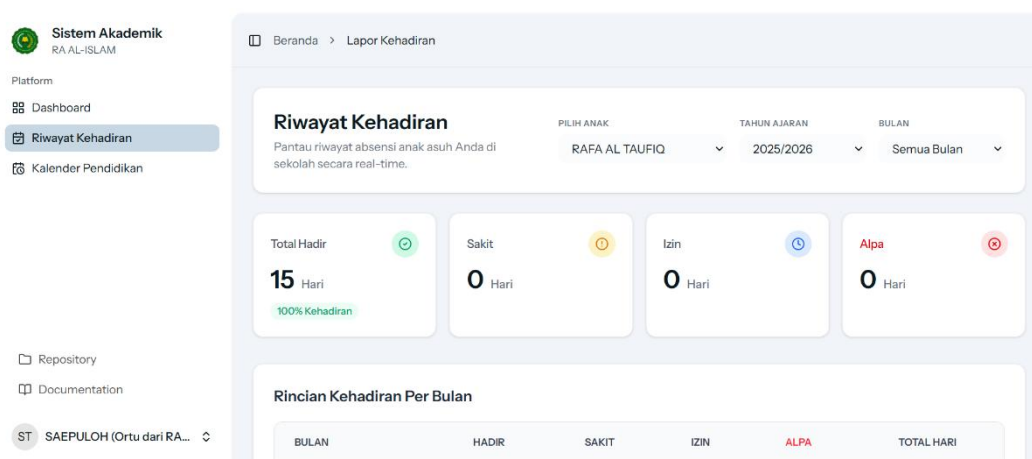
```

Gambar 5.136 Code Frontend Riwayat Kehadiran

potongan kode pada Gambar 5.136 memperlihatkan implementasi *Frontend* pada antarmuka *parent/attendances/index.tsx*. Antarmuka ini dirancang sebagai Dasbor Analitik Visual (*Visual Analytics Dashboard*) dengan mengusung prinsip reaktivitas Halaman Tunggal (*Single Page Application*). Ketika pengguna mengubah parameter pada menu tarik-turun, fungsi *handleFilterChange* langsung menyuntikkan data parameter terbaru tanpa memerlukan proses muat ulang halaman secara penuh. Selain menampilkan tabel rincian per bulan, sistem antarmuka ini juga melakukan kalkulasi matematika secara seketika (*on-the-fly calculation*) pada tingkat peramban (*client-side*) dengan sintaksis

$Math.round((summary.present / totalDays) * 100)$. Proses ini akan menghasilkan metrik persentase kehadiran anak yang sangat informatif (misal: 95% Kehadiran), sehingga mempermudah para orang tua dalam memantau secara proaktif rekam jejak kedisiplinan anak mereka di RA Al-Islam.

5. Tampilan halaman riwayat kehadiran



Gambar 5.137 Halaman Riwayat Kehadiran

Gambar 5.137 menunjukkan hasil implementasi halaman Riwayat Kehadiran pada antarmuka akun Orang Tua Siswa di Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan panel "Riwayat Kehadiran" yang dilengkapi tiga kontrol filter interaktif di bagian atas, yaitu dropdown "PILIH ANAK" (seperti "RAFA AL TAUFIQ"), filter "TAHUN AJARAN" (seperti "2025/2026"), dan filter "BULAN" (seperti "Semua Bulan"). Selain itu, halaman ini menyajikan empat kartu ringkasan metrik utama (kartu "Total Hadir" dengan persentase kehadiran 100%, kartu "Sakit", kartu "Izin", dan kartu "Alpa") serta panel tabel "Rincian Kehadiran Per Bulan" di bagian bawah.

Tampilan dirancang dengan antarmuka yang bersih dan ramah pengguna sehingga memudahkan Orang Tua Siswa dalam memantau rekapitulasi kehadiran anak secara mandiri. Penataan tata letak berbasis kartu indikator berwarna dan penanda ikon visual sangat membantu wali murid dalam mencerna ringkasan statistik kehadiran anak secara cepat. Dengan demikian, halaman Riwayat Kehadiran tidak hanya

berfungsi sebagai media penyajian laporan presensi digital, tetapi juga mendukung keterbukaan informasi akademik antara pihak sekolah dan orang tua siswa.

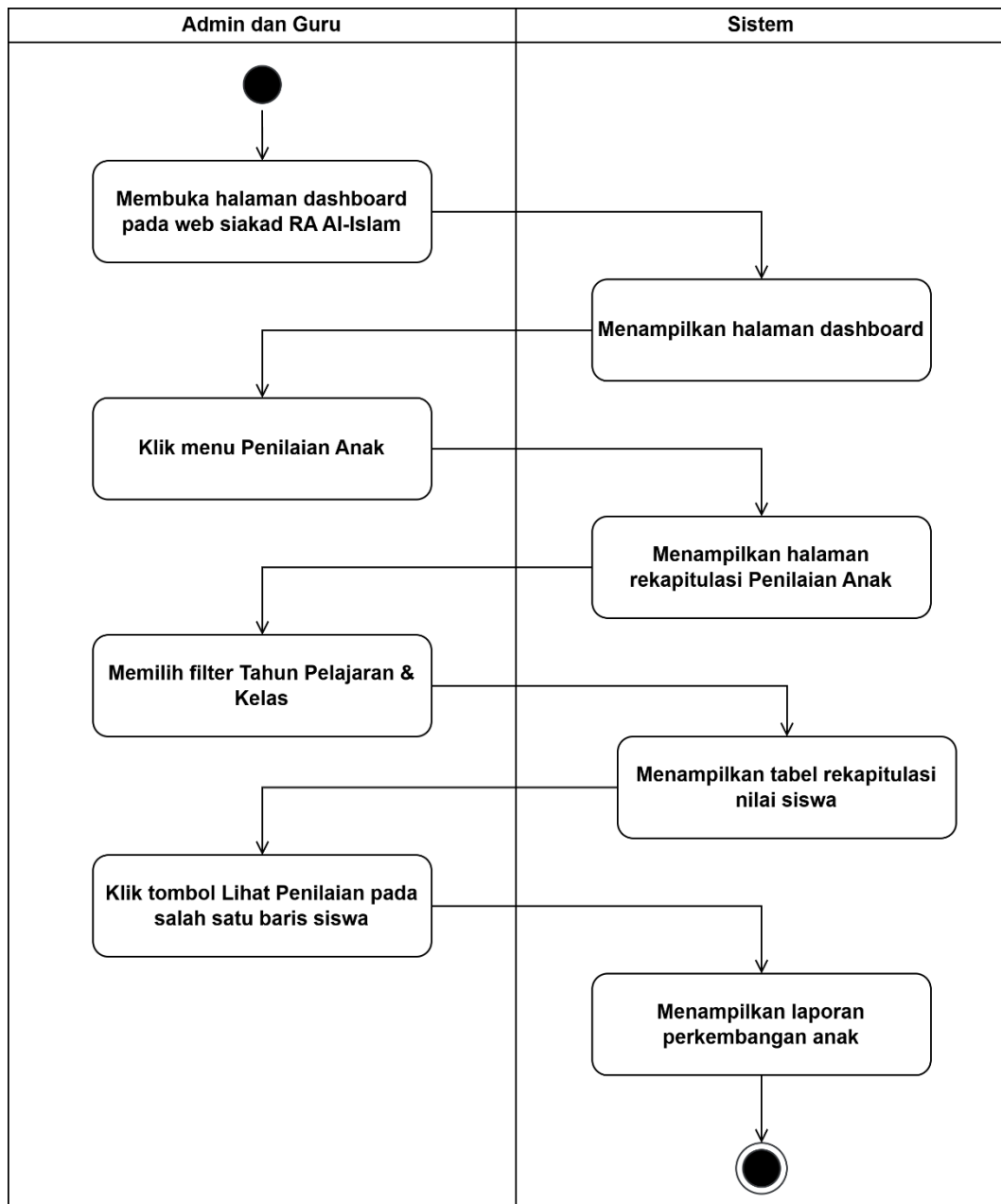
5.4.3.5 Penilaian anak (admin, guru, orang tua)

Fitur Penilaian Anak merupakan salah satu fitur yang dikembangkan pada *Sprint 3* untuk memberikan kemudahan kepada Admin, Guru, dan Orang Tua Siswa dalam melihat informasi hasil penilaian siswa yang bersumber dari Rapor Digital Madrasah (RDM). Pada fitur ini, Sistem Informasi Akademik tidak digunakan untuk melakukan proses penginputan maupun pengelolaan data penilaian, melainkan hanya menampilkan data yang telah tersedia pada database RDM. Dengan demikian, informasi penilaian dapat diakses secara lebih mudah melalui satu sistem tanpa mengubah mekanisme penilaian yang telah diterapkan oleh sekolah. Selain meningkatkan kemudahan dalam memperoleh informasi akademik, fitur ini juga membantu menjaga konsistensi data karena seluruh informasi penilaian yang ditampilkan berasal dari sumber data yang sama.

Perancangan fitur Penilaian Anak dilakukan menggunakan *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram* sebagai dasar dalam memahami alur proses sebelum implementasi dilakukan. *Activity Diagram* digunakan untuk menggambarkan aktivitas pengguna ketika mengakses informasi penilaian siswa, *Class Diagram* digunakan untuk menjelaskan struktur kelas yang berperan dalam proses pengambilan dan penyajian data penilaian dari database RDM beserta hubungan antar kelas yang terlibat, sedangkan *Sequence Diagram* digunakan untuk menggambarkan urutan komunikasi antara pengguna, sistem, dan basis data selama proses menampilkan informasi penilaian berlangsung. Ketiga diagram tersebut menjadi acuan dalam proses implementasi sehingga data penilaian dapat ditampilkan secara akurat sesuai dengan hak akses masing-masing pengguna.

1. Penilaian anak (admin dan guru)

a. Activity diagram



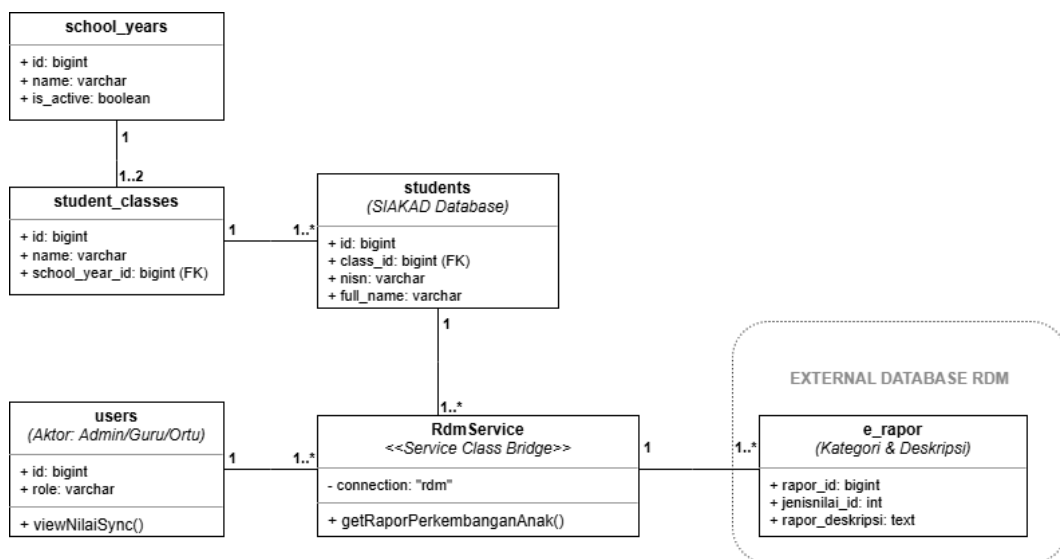
Gambar 5.138 Activity Diagram Penilaian Anak (Admin dan Guru)

Activity Diagram Penilaian Anak (Admin dan Guru) pada Gambar 5.138 menggambarkan alur proses peninjauan rekapitulasi nilai siswa yang dilakukan oleh Admin dan Guru pada Sistem Informasi Akademik RA Al-Islam Gunungcupu.

Proses diawali ketika Admin dan Guru membuka Dashboard kemudian memilih menu Penilaian Anak sehingga sistem menampilkan halaman rekapitulasi Penilaian Anak. Selanjutnya pengguna memilih filter Tahun Pelajaran & Kelas untuk menampilkan tabel rekapitulasi nilai siswa yang relevan.

Setelah tabel daftar siswa ditampilkan, pengguna dapat mengklik tombol Lihat Penilaian pada salah satu baris siswa yang dipilih. Sistem kemudian secara otomatis memproses kueri penarikan data perkembangan anak dari database eksternal dan menampilkan laporan perkembangan anak secara detail pada layar antarmuka. Diagram tersebut menjadi acuan dalam proses implementasi agar seluruh alur penyajian rekapitulasi dan detail nilai RDM dapat berjalan secara terstruktur, konsisten, serta sesuai dengan kebutuhan sistem yang telah dirancang.

b. *Class diagram*



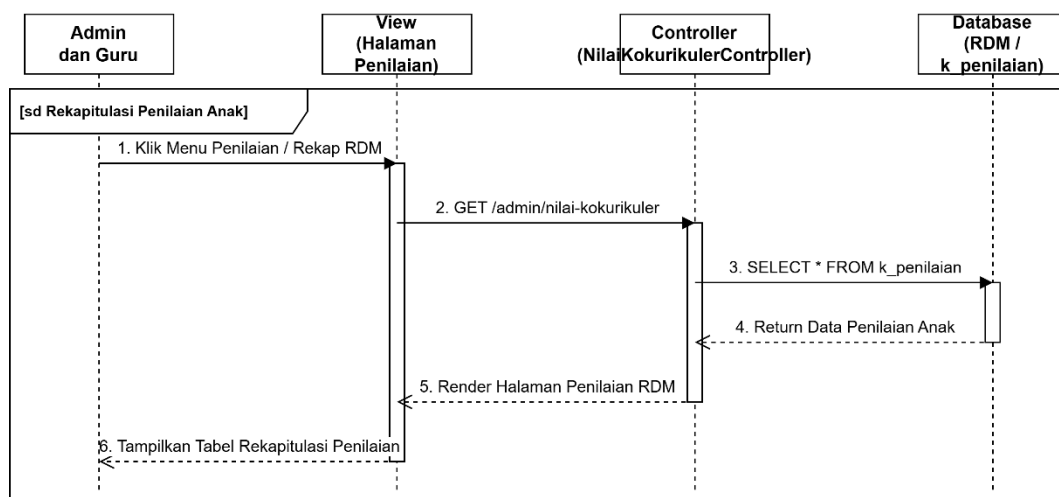
Gambar 5.139 *Class Diagram* Penilaian Anak (Admin dan Guru)

Class Diagram Penilaian Anak (Admin dan Guru) pada Gambar 5.139 digunakan untuk menggambarkan struktur relasional objek dan arsitektur integrasi basis data eksternal yang berperan dalam penyajian nilai siswa pada sistem. Diagram ini menunjukkan interaksi antara kelas lokal SIKAD (seperti *school_years*, *student_classes*, *students*, dan *users* dengan operasi *viewNilaiSync()*) terhadap kelas

jembatan *RdmService* (*<<Service Bridge Class>>*) yang mengelola properti connection: "rdm" dan fungsi *getRaporPerkembanganAnak()*, yang terhubung langsung ke entitas eksternal RDM yaitu kelas *e_rapor* dengan atribut *rapor_id*, *jenisnilai_id*, dan *rapor_deskripsi*.

Melalui *Class Diagram*, pemetaan arsitektur bridge pattern antara *database* internal SIAKAD dan *database* eksternal RDM dapat dirancang secara sistematis. Kejelasan hubungan *multiplicity* memperlihatkan bahwa satu entitas *students* dapat terhubung ke satu atau banyak *record* penilaian pada *RdmService* dan *e_rapor*. Pembagian kelas ini memastikan bahwa proses pembacaan data nilai dari *database* luar tidak mengganggu struktur tabel utama SIAKAD, sehingga kestabilan dan integritas basis data sistem tetap terjaga secara konsisten.

c. Sequence diagram



Gambar 5.140 *Sequence Diagram* Penilaian Anak (Admin dan Guru)

Sequence Diagram Penilaian Anak (Admin dan Guru) pada Gambar 5.140 menggambarkan urutan komunikasi antara Admin/Guru, antarmuka sistem, *controller*, dan basis data eksternal selama proses melihat rekapitulasi penilaian berlangsung. Diagram ini menunjukkan bagaimana permintaan untuk menampilkan data nilai dikirimkan oleh pengguna dari *View* (Halaman Penilaian), kemudian diproses oleh *NilaiKokurikulerController* melalui metode *HTTP GET* dengan rute

/admin/nilai-kokurikuler sebelum diteruskan ke Database RDM (tabel k_penilaian) menggunakan kueri *SQL SELECT * FROM k_penilaian* hingga menghasilkan respons data penilaian anak yang dikembalikan ke antarmuka.

Melalui *Sequence Diagram*, interaksi pertukaran pesan lintas basis data dapat dipahami secara rinci dan berurutan sesuai alur waktu. Diagram ini menjadi acuan dalam implementasi logika controller di sisi server agar proses penarikan data dari database RDM dapat dieksekusi secara cepat, memicu perenderan halaman (Render Halaman Penilaian RDM), serta menyajikan tabel rekapitulasi nilai secara akurat tanpa mengalami kegagalan koneksi (*connection timeout*).

d. Implementasi

```

1  public function penilaian(string $id)
2  {
3      $student = Student::with('studentClass')->findOrFail($id);
4
5      // Role-Based Access Control (RBAC): Validasi ketat khusus Guru
6      if (auth()->user()->role === 'teacher') {
7          if ($student->studentClass->teacher_id !== auth()->id()) {
8              abort(403, 'Unauthorized access to this student'); // Blokir akses
9          }
10     }
11
12     // Integrasi Sinkronisasi API ke Database RDM Eksternal via Service Layer
13     $rapor = $this->rdmService->getRaporPerkembanganAnak($student->nisn);
14
15     return Inertia::render('admin/nilai-kokurikuler/penilaian', [
16         'student' => $student,
17         'rapor'   => $rapor,
18     ]);
19 }

```

Gambar 5.141 *Code Backend* Penilaian Anak (Admin dan Guru)

Potongan kode pada Gambar 5.141 menunjukkan metode `penilaian()` di dalam kelas `Admin/NilaiKokurikulerController`. Karena fitur ini diakses bersamaan oleh dua aktor dengan wewenang berbeda (Admin dan Guru), sistem mengimplementasikan teknik *Role-Based Access Control* (Kendali Akses Berbasis Peran). Sebelum memanggil data rapor, peladen (*server*) melakukan validasi keamanan `if (auth()->user()->role === 'teacher')`. Pengecekan ini memastikan bahwa seorang Guru

hanya memiliki hak otorisasi untuk melihat nilai siswa yang terdaftar di kelas yang ia ampu. Jika Guru mencoba mengakses ID siswa dari kelas lain melalui manipulasi parameter URL, sistem akan secara otomatis memblokir koneksi dengan kode Error 403 (*Unauthorized Access*). Setelah proses validasi lolos, *Controller* menugaskan kelas perantara melalui fungsi `$this->rdmService->getRaporPerkembanganAnak($student->nisn)` untuk menjembatani komunikasi ke basis data eksternal RDM (Rapor Digital Madrasah). Pendekatan *Service Pattern* ini memastikan bahwa sistem SIAKAD tidak melakukan redudansi penyimpanan data, melainkan menyajikan data termutakhir (*real-time*) secara tersinkronisasi.

```

1  export default function PenilaianPerkembanganAnakDetail({ student, rapor }: Props) {
2    return (
3      

4        

5          

6            

7              

8                

9                  

10                     

11                       

12                         

13                           

14                             

15                               

16                                 

17                                   

18                                     

19                                       

20                                         

21                                           

22                                             

23                                               

24                                                 

25                                                   

26                                                     

27                                                       

28                                                         

29                                                           

30                                                             

31                                                               

32                                                                 

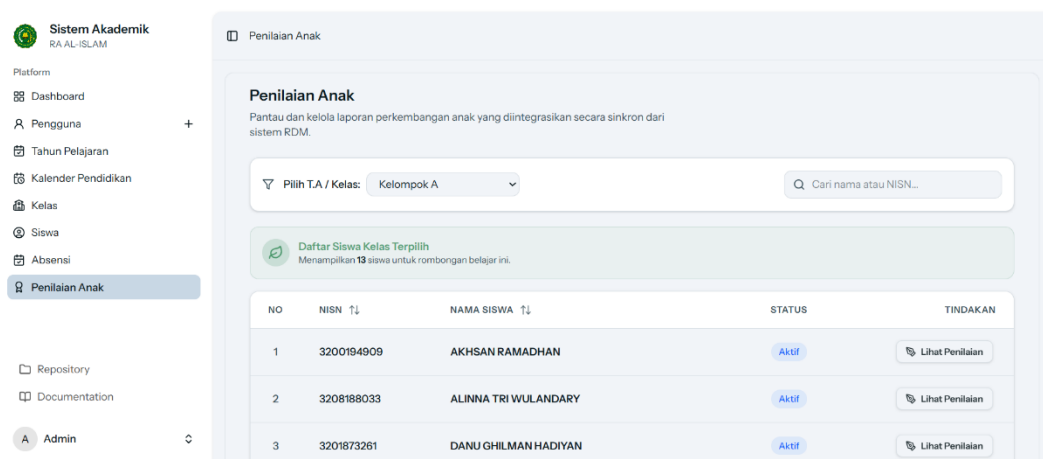
33


```

Gambar 5.142 Code Frontend Penilaian Anak (Admin dan Guru)

potongan kode pada Gambar 5.142 memperlihatkan implementasi *Frontend* pada antarmuka admin/nilai-kokurikuler/penilaian.tsx. Antarmuka ini dirancang dengan arsitektur visual yang responsif (*Responsive Visual Architecture*) untuk menyajikan rincian Laporan Perkembangan Anak secara sistematis. Alih-alih menulis kode antarmuka berulang kali untuk setiap kriteria penilaian (Agama, Jati Diri, Dasar Literasi), sistem memanfaatkan fungsi perulangan `rapor.map((item, index) => ...)` untuk merender blok-blok informasi secara dinamis (*Dynamic Rendering*) sesuai dengan struktur data *Array* yang dikembalikan oleh RDM. Selain itu, implementasi penyajian ini dibekali dengan ekspresi kondisional (*Conditional Rendering*) menggunakan sintaksis *Ternary Operator* `item.deskripsi ? (...) : (...)`. Logika ini sangat esensial agar sistem dapat mendeteksi kekosongan nilai; jika data deskripsi dari peladen RDM belum diinput oleh pihak kementerian/sekolah, sistem tidak akan menampilkan layar kerusakan (*crash/blank page*), melainkan menyisipkan pesan peringatan otomatis dengan sangat halus.

e. Tampilan halaman

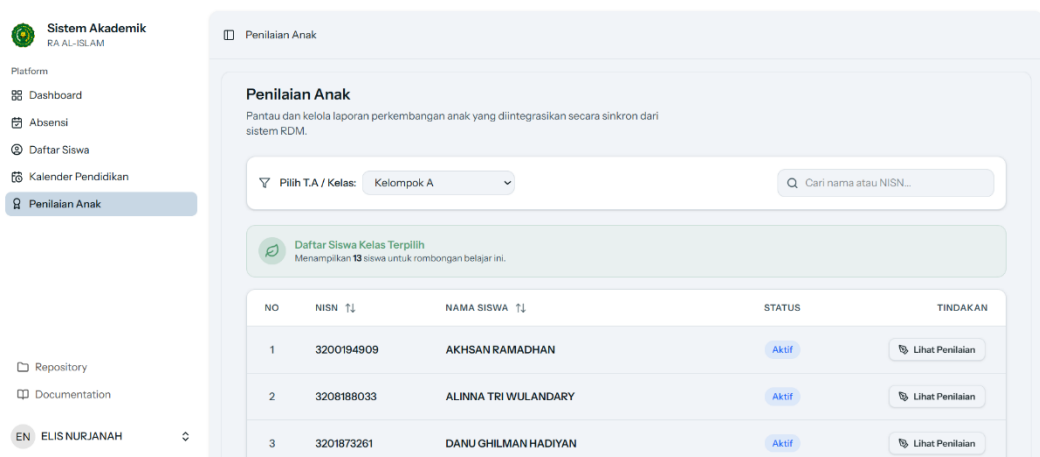


Gambar 5.143 Halaman Penilaian Anak Admin

Gambar 5.143 menunjukkan hasil implementasi halaman Penilaian Anak pada antarmuka Admin di Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan panel "Penilaian Anak" yang dilengkapi filter *dropdown* "Pilih T.A / Kelas" (seperti "Kelompok A"), kolom pencarian nama/NISN, serta

kotak informasi "Daftar Siswa Kelas Terpilih" yang merangkum jumlah siswa aktif dalam rombongan belajar. Pada bagian utama, disajikan tabel terstruktur dengan kolom nomor, NISN, nama siswa, status siswa (*badge* "Aktif"), serta kolom tindakan berupa tombol "Lihat Penilaian" untuk membuka detail laporan perkembangan anak dari RDM.

Tampilan dirancang dengan antarmuka yang bersih dan konsisten antara hak akses Admin sehingga memberikan pengalaman penggunaan yang intuitif. Penataan elemen filter dan tombol tindakan yang kontras memudahkan pengguna dalam bernavigasi serta mengakses laporan perkembangan murid secara efisien. Dengan demikian, halaman Penilaian Anak tidak hanya berfungsi sebagai media penyajian laporan hasil belajar, tetapi juga membuktikan keberhasilan integrasi sistem informasi akademik dengan basis data eksternal RDM secara tepat guna.



Gambar 5.144 Halaman Penilaian Anak Guru

Gambar 5.144 menunjukkan hasil implementasi halaman Penilaian Anak pada antarmuka Guru di Sistem Informasi Akademik RA Al-Islam Gunungcupu. Halaman ini menampilkan panel "Penilaian Anak" yang dilengkapi filter *dropdown* "Pilih T.A / Kelas" (seperti "Kelompok A"), kolom pencarian nama/NISN, serta kotak informasi "Daftar Siswa Kelas Terpilih" yang merangkum jumlah siswa aktif dalam rombongan belajar. Pada bagian utama, disajikan tabel terstruktur dengan kolom nomor, NISN, nama siswa, status siswa (*badge* "Aktif"), serta kolom

tindakan berupa tombol "Lihat Penilaian" untuk membuka detail laporan perkembangan anak dari RDM.

Tampilan dirancang dengan antarmuka yang bersih dan konsisten antara hak akses Admin dan Guru sehingga memberikan pengalaman penggunaan yang intuitif. Penataan elemen filter dan tombol tindakan yang kontras memudahkan pengguna dalam bernavigasi serta mengakses laporan perkembangan murid secara efisien. Dengan demikian, halaman Penilaian Anak tidak hanya berfungsi sebagai media penyajian laporan hasil belajar, tetapi juga membuktikan keberhasilan integrasi sistem informasi akademik dengan basis data eksternal RDM secara tepat guna.

5.4.3.6 *Black box testing sprint 3*

Setelah seluruh fitur pada *Sprint 3* berhasil diimplementasikan, tahapan berikutnya adalah melakukan pengujian menggunakan metode *Black Box Testing*. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur yang telah dikembangkan dapat berfungsi sesuai dengan kebutuhan fungsional yang telah ditetapkan pada *Product Backlog*. Metode *Black Box Testing* dilakukan dengan menguji setiap fungsi sistem berdasarkan masukan (*input*) yang diberikan oleh pengguna dan membandingkan hasil keluaran (*output*) yang dihasilkan dengan hasil yang diharapkan, tanpa memperhatikan struktur maupun logika kode program yang digunakan. Melalui pengujian ini, dapat diketahui apakah seluruh fitur yang dikembangkan telah berjalan sesuai dengan kebutuhan pengguna sebelum dilakukan evaluasi bersama stakeholder pada tahapan *Sprint Review*.

Pengujian pada *Sprint 3* dilakukan terhadap seluruh fitur yang berhasil diselesaikan, yaitu Daftar Siswa, Profil Siswa, Absensi (Rekap Admin), Riwayat Kehadiran, dan Penilaian Anak. Setiap fitur diuji berdasarkan skenario penggunaan yang mewakili aktivitas masing-masing pengguna sesuai dengan hak akses yang dimiliki, kemudian hasil yang diperoleh dibandingkan dengan keluaran yang diharapkan untuk memastikan kesesuaian fungsi sistem. Hasil pengujian menunjukkan bahwa seluruh fitur pada *Sprint 3* dapat berjalan sesuai dengan kebutuhan fungsional tanpa ditemukan kesalahan yang memengaruhi proses utama sistem. Dengan demikian,

seluruh fitur dinyatakan berhasil dan layak untuk dilanjutkan ke tahapan *Sprint Review* sebagai bahan evaluasi bersama stakeholder.

Tabel 5.11 Hasil *Blackbox Testing* Sprint 3

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
1	Menampilkan daftar siswa	Guru memilih menu Daftar Siswa	Sistem menampilkan daftar siswa sesuai kelas yang diampu	Daftar siswa berhasil ditampilkan sesuai data	Berhasil
2	Menampilkan profil siswa	Orang Tua memilih menu Profil Siswa	Sistem menampilkan profil anak sesuai akun yang digunakan	Profil siswa berhasil ditampilkan dengan benar	Berhasil
3	Menampilkan rekapitulasi absensi siswa	Admin memilih menu Absensi kemudian memilih Tahun Pelajaran dan Kelas	Sistem menampilkan rekapitulasi absensi siswa sesuai filter yang dipilih	Rekapitulasi absensi berhasil ditampilkan sesuai data	Berhasil

No	Skenario Uji	Input Data	Ekspektasi Output	Hasil Pengujian	Status
4	Menampilkan riwayat kehadiran siswa	Orang Tua memilih menu Riwayat Kehadiran	Sistem menampilkan riwayat kehadiran anak sesuai data yang tersedia	Riwayat kehadiran berhasil ditampilkan dengan benar	Berhasil
5	Menampilkan penilaian anak	Admin, Guru, atau Orang Tua memilih menu Penilaian Anak	Sistem menampilkan data penilaian anak sesuai hak akses pengguna	Data penilaian berhasil ditampilkan dengan benar	Berhasil

Berdasarkan hasil *Black Box Testing* pada Tabel 5.11, dapat diketahui bahwa seluruh skenario pengujian yang dilakukan pada fitur Daftar Siswa, Profil Siswa, Absensi (Rekap Admin), Riwayat Kehadiran, dan Penilaian Anak telah menghasilkan keluaran yang sesuai dengan kebutuhan fungsional sistem. Setiap fitur mampu menampilkan informasi yang dibutuhkan oleh pengguna sesuai dengan hak akses masing-masing tanpa ditemukan kesalahan yang memengaruhi proses utama sistem. Khusus pada fitur Penilaian Anak, sistem berhasil menampilkan data penilaian yang bersumber dari Rapor Digital Madrasah (RDM) sesuai dengan hak akses Admin, Guru, dan Orang Tua Siswa tanpa menyediakan proses penambahan, perubahan, maupun penghapusan data. Dengan demikian, hasil pengujian menunjukkan bahwa seluruh fitur pada *Sprint 3* telah berjalan dengan baik dan siap untuk dievaluasi lebih lanjut pada tahapan *Sprint Review*.

5.4.3.7 *Sprint review sprint 3*

Setelah seluruh fitur pada *Sprint 3* berhasil diimplementasikan dan dinyatakan berfungsi dengan baik melalui *Black Box Testing*, tahapan selanjutnya adalah melaksanakan *Sprint Review*. Pada tahap ini, hasil pengembangan sistem dipresentasikan kepada stakeholder, yaitu pihak RA Al-Islam Gunungcupu, untuk memperoleh umpan balik terhadap fitur-fitur yang telah dikembangkan. Kegiatan ini bertujuan untuk memastikan bahwa setiap fungsi yang diimplementasikan telah sesuai dengan kebutuhan operasional sekolah serta memenuhi harapan pengguna sebelum proses pengembangan dilanjutkan ke tahap implementasi penuh.

Sprint Review pada *Sprint 3* dilakukan terhadap seluruh fitur yang berhasil dikembangkan, yaitu Daftar Siswa, Profil Siswa, Rekapitulasi Absensi, Riwayat Kehadiran, dan Penilaian Anak (RDM). Hasil evaluasi menunjukkan bahwa seluruh fitur telah berfungsi secara optimal, sesuai dengan ekspektasi pengguna, dan tidak memerlukan masukan atau revisi dari pihak sekolah. Integrasi data penilaian eksternal RDM serta penyajian rekapitulasi kehadiran dinilai telah sangat rapi dan siap untuk diterapkan pada operasional sekolah. Hasil *Sprint Review Sprint 3* dapat dilihat pada Tabel 5.12.

Tabel 5.12 Hasil *Sprint Review Sprint 3*

Pengguna	Fitur	Review
Guru	Daftar Siswa	Fitur Daftar Siswa telah berjalan sesuai kebutuhan sehingga Guru dapat melihat daftar siswa beserta profil siswa pada kelas yang diampu dengan baik.
	Penilaian Anak	Fitur Penilaian Anak berhasil menampilkan data penilaian siswa yang bersumber dari database RDM sesuai dengan kelas yang diampu sehingga dapat

Pengguna	Fitur	Review
		digunakan untuk memantau capaian perkembangan siswa.
Orang Tua	Profil Siswa	Fitur Profil Siswa telah menampilkan informasi identitas anak sesuai dengan akun Orang Tua yang digunakan sehingga informasi siswa dapat diakses secara mandiri.
	Riwayat Kehadiran	Fitur Riwayat Kehadiran telah berjalan sesuai kebutuhan sehingga Orang Tua dapat memantau kehadiran anak berdasarkan data absensi yang tersedia pada sistem.
	Penilaian Anak	Fitur Penilaian Anak berhasil menampilkan informasi capaian perkembangan anak yang bersumber dari database RDM sehingga Orang Tua dapat memantau perkembangan akademik anak dengan mudah.
Admin	Rekap Absensi	Fitur Rekap Absensi telah berjalan sesuai kebutuhan sehingga Admin dapat melihat rekapitulasi kehadiran seluruh siswa berdasarkan Tahun Pelajaran dan Kelas.
	Penilaian Anak	Fitur Penilaian Anak berhasil menampilkan rekapitulasi penilaian siswa yang bersumber dari database RDM sesuai dengan kebutuhan sistem sehingga informasi dapat diakses dengan baik.

Berdasarkan hasil *Sprint Review* pada Tabel 5.12, seluruh fitur yang dikembangkan pada *Sprint 3* telah diterima sepenuhnya tanpa adanya catatan masukan atau revisi. Fitur Daftar Siswa, Profil Siswa, Rekapitulasi Absensi, Riwayat Kehadiran, dan Penilaian Anak (RDM) terbukti telah berjalan sesuai dengan alur administrasi akademik yang berlaku di RA Al-Islam Gunungcupu. Dengan diterimanya seluruh modul pada *Sprint 3* ini, keseluruhan tahap pengembangan sistem informasi berbasis *Scrum* telah dinyatakan selesai dan siap digunakan secara penuh.

5.4.3.8 *Sprint retrospective sprint 3*

Setelah *Sprint Review* selesai dilaksanakan, tahapan berikutnya adalah *Sprint Retrospective*. Pada tahap ini, peneliti melakukan evaluasi terhadap keseluruhan proses pelaksanaan *Sprint 3*, mulai dari perencanaan *backlog*, implementasi sistem, pengujian menggunakan *Black Box Testing*, hingga hasil *Sprint Review* yang telah dilakukan bersama stakeholder. Evaluasi ini bertujuan untuk mengidentifikasi hal-hal yang telah berjalan dengan baik selama *Sprint* berlangsung serta memastikan bahwa proses pengembangan telah sesuai dengan perencanaan yang telah ditetapkan sebelumnya. Hasil *Sprint Retrospective Sprint 3* dapat dilihat pada Tabel 5.13.

Tabel 5.13 Hasil *Sprint Retrospective Sprint 3*

Pengguna	Aspek Evaluasi	<i>Sprint Retrospective</i>
Peneliti	Pelaksanaan <i>Sprint 3</i>	Seluruh <i>backlog</i> pada <i>Sprint 3</i> berhasil diselesaikan sesuai dengan perencanaan. Proses implementasi sistem berjalan dengan baik, pengujian <i>Black Box Testing</i> menunjukkan seluruh fungsi berjalan sesuai kebutuhan, serta hasil <i>Sprint Review</i> menunjukkan bahwa seluruh fitur diterima oleh stakeholder tanpa revisi. Berdasarkan hasil evaluasi tersebut, proses pengembangan dinyatakan telah memenuhi target

Pengguna	Aspek Evaluasi	<i>Sprint Retrospective</i>
		Sprint yang direncanakan dan siap untuk dilanjutkan ke tahapan penyelesaian sistem.

Berdasarkan hasil *Sprint Retrospective* pada Tabel 5.13, dapat diketahui bahwa pelaksanaan *Sprint 3* telah berjalan sesuai dengan perencanaan yang telah disusun pada tahap *Sprint Planning*. Seluruh *backlog* yang menjadi target *Sprint* berhasil diselesaikan tepat waktu dan diimplementasikan sesuai dengan kebutuhan sistem. Selain itu, proses implementasi menggunakan Laravel 12 dan React JS berlangsung dengan baik, seluruh skenario *Black Box Testing* menunjukkan hasil sesuai dengan yang diharapkan, serta hasil *Sprint Review* menyatakan bahwa seluruh fitur yang dikembangkan telah diterima oleh stakeholder tanpa memerlukan revisi. Oleh karena itu, hasil evaluasi pada *Sprint Retrospective* menunjukkan bahwa proses pengembangan pada *Sprint 3* telah berjalan secara efektif dan seluruh target pengembangan sistem berhasil diselesaikan sesuai dengan tujuan penelitian yang telah ditetapkan.

5.5 Implementasi Sistem

Tahap implementasi sistem merupakan proses penerapan hasil pengembangan yang telah dilakukan pada tahapan sebelumnya sehingga sistem dapat digunakan pada lingkungan operasional yang sebenarnya. Pada tahap ini, Sistem Informasi Akademik RA Al-Islam yang sebelumnya hanya dijalankan dan diuji pada lingkungan lokal (*localhost*) dipindahkan ke sebuah peladen (*server*) yang terhubung ke internet sehingga dapat diakses secara daring oleh pengguna sesuai dengan hak akses yang dimiliki. Proses implementasi ini menjadi tahapan yang penting karena sistem informasi tidak hanya dinilai dari keberhasilan proses pengembangannya, tetapi juga dari kemampuannya untuk diterapkan dan digunakan secara langsung oleh pengguna, yaitu Admin, Guru, dan Orang Tua Siswa. Melalui tahap implementasi ini, kesiapan sistem untuk beroperasi pada

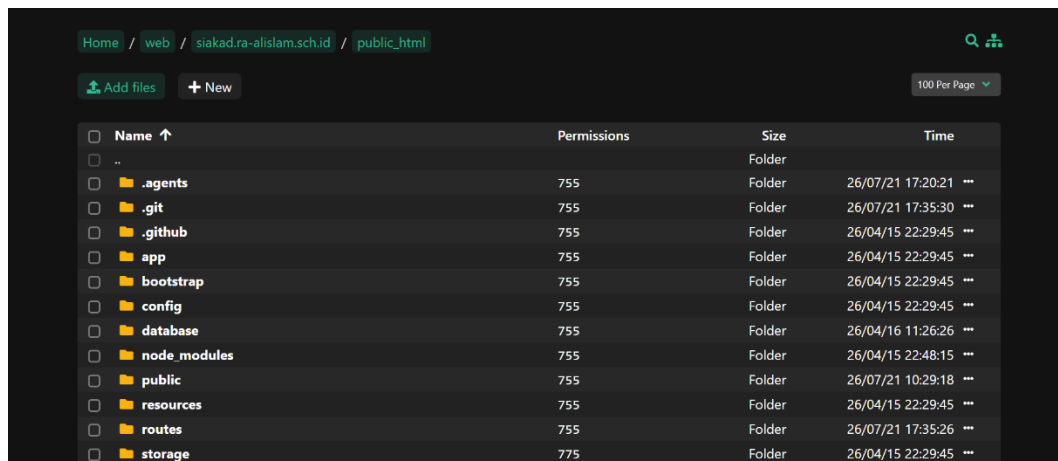
lingkungan nyata dapat dibuktikan sehingga seluruh fungsi yang telah dikembangkan dapat dimanfaatkan dalam mendukung pengelolaan akademik di RA Al-Islam Gunungcupu.

5.5.1 *Deployment dan hosting*

Agar sistem dapat digunakan secara daring (*online*), diperlukan proses *deployment* dan pengaturan layanan hosting. Sistem ini ditempatkan pada layanan *Virtual Private Server* (VPS) berbasis sistem operasi Linux yang dikelola menggunakan *Hestia Control Panel* (HestiaCP). Penggunaan VPS memberikan keleluasaan bagi pengembang dalam mengatur konfigurasi *server*, penggunaan sumber daya, serta versi perangkat lunak yang dibutuhkan oleh framework Laravel sehingga aplikasi dapat berjalan secara optimal.

Pemilihan VPS dilakukan dengan mempertimbangkan aspek performa, fleksibilitas konfigurasi, dan keamanan sistem. Berbeda dengan layanan *shared hosting* yang memiliki keterbatasan dalam pengaturan server, VPS memungkinkan pengembang mengelola lingkungan aplikasi secara lebih mandiri sesuai kebutuhan sistem. Pada subbab ini dijelaskan proses pemindahan aplikasi dari komputer lokal ke *server*, konfigurasi yang dilakukan agar sistem dapat berjalan pada lingkungan produksi, serta hasil akhir setelah sistem berhasil diimplementasikan dan dapat diakses secara daring.

5.5.1.1 Proses deployment ke hosting

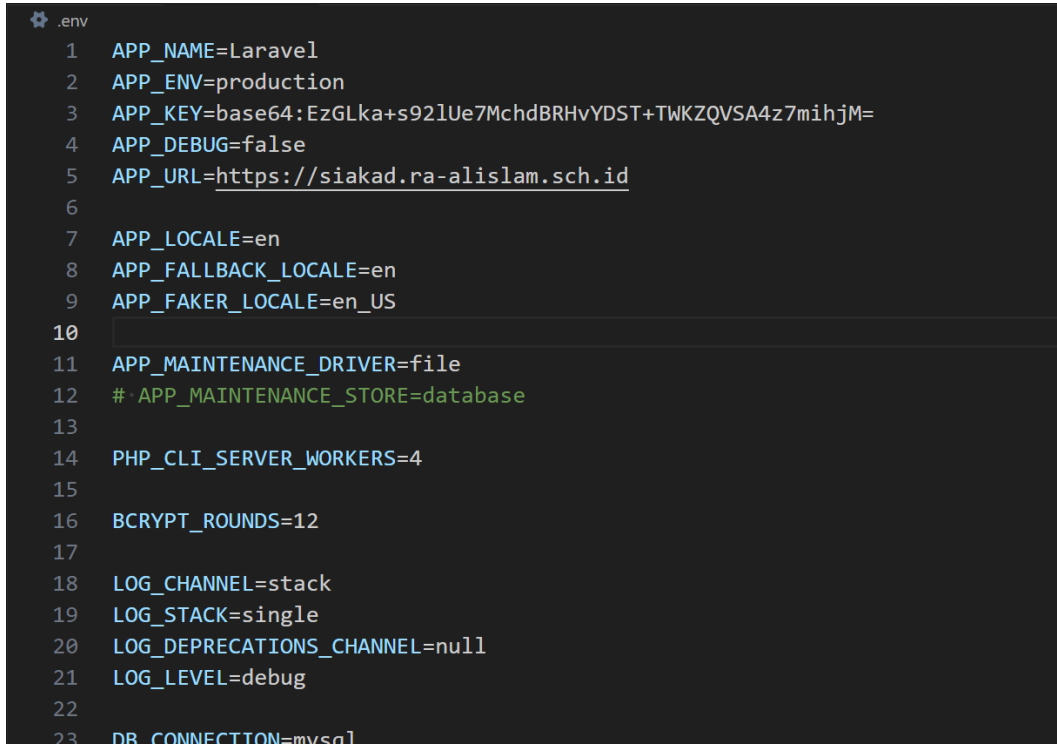


Gambar 5.145 Direktori aplikasi pada File Manager HestiaCP

Langkah pertama pada proses *deployment* dilakukan dengan memindahkan kode program dari komputer lokal menuju server melalui HestiaCP. Untuk mempercepat proses pengiriman data melalui jaringan internet, seluruh kode aplikasi dikompresi ke dalam berkas berformat *ZIP*. Namun demikian, folder vendor yang berisi dependensi PHP (*Hypertext Preprocessor*) serta folder *node_modules* yang berisi dependensi JavaScript tidak disertakan karena ukuran berkas yang cukup besar. Setelah proses unggah selesai, berkas *ZIP* diekstrak ke dalam direktori utama aplikasi, yaitu `/home/alislam/web/siakad.ra-alislam.sch.id/public_html`, kemudian seluruh dependensi diinstal kembali secara langsung pada *server*.

Secara teknis, proses *deployment* juga menerapkan strategi optimasi untuk menjaga stabilitas *server* produksi. Proses kompilasi aset antarmuka yang dibangun menggunakan React dan Inertia.js tidak dilakukan pada *server*, melainkan tetap dijalankan di komputer lokal. Hanya hasil kompilasi (build) yang diunggah ke *server*. Keputusan ini diambil karena proses kompilasi JavaScript membutuhkan sumber daya memori yang cukup besar sehingga berpotensi menyebabkan penggunaan *Random Access Memory* (RAM) meningkat secara signifikan dan dapat memengaruhi stabilitas server apabila dilakukan langsung pada lingkungan produksi.

5.5.1.2 Konfigurasi sistem pada *server* hosting



```

1 APP_NAME=Laravel
2 APP_ENV=production
3 APP_KEY=base64:EzGLka+s92lUe7MchdBRHvYDST+TWKZQVSA4z7mihjM=
4 APP_DEBUG=false
5 APP_URL=https://siakad.ra-alislam.sch.id
6
7 APP_LOCALE=en
8 APP_FALLBACK_LOCALE=en
9 APP_FAKER_LOCALE=en_US
10
11 APP_MAINTENANCE_DRIVER=file
12 # APP_MAINTENANCE_STORE=database
13
14 PHP_CLI_SERVER_WORKERS=4
15
16 BCRYPT_ROUNDS=12
17
18 LOG_CHANNEL=stack
19 LOG_STACK=single
20 LOG_DEPRECATED_CHANNEL=null
21 LOG_LEVEL=debug
22
23 DB_CONNECTION=mysql

```

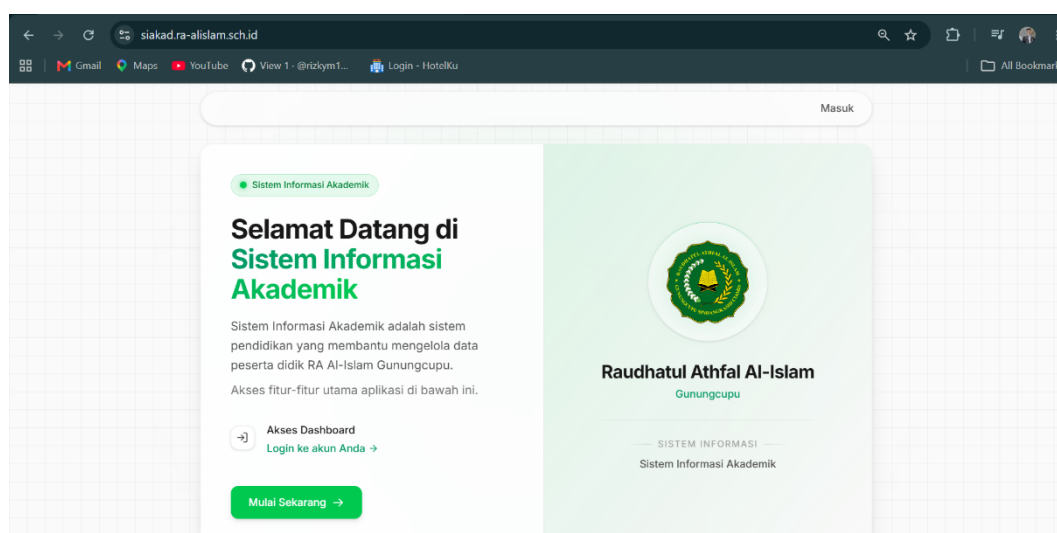
Gambar 5.146 Konfigurasi berkas *.env* pada *server production*

Setelah proses pemindahan berkas selesai dilakukan, aplikasi masih memerlukan proses konfigurasi agar dapat berjalan dengan baik pada lingkungan *server* produksi. Konfigurasi diawali dengan melakukan penyesuaian pada berkas *.env* (*environment*) yang berfungsi sebagai tempat penyimpanan konfigurasi aplikasi. Pada tahap ini, status aplikasi diubah menjadi mode produksi (*production*) dan fitur penelusuran kesalahan (*debug*) dinonaktifkan. Selanjutnya, aplikasi dikonfigurasi untuk terhubung dengan basis data pada *server*, kemudian struktur tabel dibentuk menggunakan perintah *php artisan migrate*.

Penonaktifan fitur *debug* merupakan langkah penting untuk meningkatkan keamanan aplikasi. Apabila fitur tersebut tetap diaktifkan, informasi mengenai struktur aplikasi maupun rincian kesalahan sistem dapat ditampilkan kepada pengguna ketika terjadi *error*, sehingga berpotensi dimanfaatkan oleh pihak yang tidak bertanggung jawab. Selain itu, dilakukan pengaturan hak akses (*permissions*) pada direktori penyimpanan agar aplikasi dapat menyimpan data yang diunggah

oleh pengguna. Sebagai tahap akhir konfigurasi, dijalankan perintah php artisan storage untuk membangun tautan antara direktori penyimpanan dan direktori publik sehingga berkas seperti foto profil, bukti absensi, maupun dokumen lain yang diunggah dapat ditampilkan dengan baik melalui aplikasi.

5.5.1.3 Hasil deployment sistem



Gambar 5.147 Sistem Informasi Akademik RA Al-Islam yang telah berhasil diimplementasikan pada *server*

Setelah seluruh proses deployment dan konfigurasi *server* selesai dilakukan, Sistem Informasi Akademik RA Al-Islam berhasil diimplementasikan pada lingkungan produksi. Sistem ini telah dapat diakses secara daring melalui alamat <https://siakad.ra-alislam.sch.id> oleh pengguna yang memiliki hak akses, yaitu Admin, Guru, dan Orang Tua Siswa. Keberhasilan implementasi ini menunjukkan bahwa seluruh fitur utama yang telah dikembangkan, mulai dari pengelolaan data akademik, absensi, hingga integrasi data Rapor Digital Madrasah (RDM) pada fitur Penilaian Anak, dapat dijalankan dengan baik pada lingkungan operasional sekolah.

Dari aspek keamanan, sistem telah menerapkan protokol *Hypertext Transfer Protocol Secure* (HTTPS) yang didukung oleh sertifikat *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) dari *Let's Encrypt*. Penerapan mekanisme keamanan tersebut memungkinkan proses pertukaran data antara

pengguna dan *server* berlangsung dalam bentuk terenkripsi sehingga membantu melindungi informasi penting, seperti data pengguna, kata sandi, dan data akademik siswa dari potensi penyadapan selama proses komunikasi melalui jaringan internet.

Berdasarkan hasil deployment dan konfigurasi yang telah dilakukan, sistem berhasil dijalankan pada lingkungan server produksi dan siap digunakan sebagai media implementasi selama 30 hari di RA Al-Islam Gunungcupu. Tahapan implementasi tersebut akan dijelaskan pada subbab berikutnya.

5.5.2 Hasil implementasi sistem

Setelah Sistem Informasi Akademik RA Al-Islam Gunungcupu berhasil diunggah ke peladen produksi (*live hosting*), tahap selanjutnya adalah menerapkan sistem secara langsung pada kegiatan operasional sekolah. Implementasi ini dilaksanakan mulai tanggal 1 Juni 2026 sebagai bentuk penerapan nyata dari sistem yang telah dikembangkan menggunakan metode *Scrum*.

Penjelasan batasan operasional ini disajikan secara transparan untuk menggambarkan kondisi faktual penggunaan sistem oleh Admin, Guru, dan Orang Tua Siswa di lapangan sebelum dilanjutkannya evaluasi pengalaman pengguna menggunakan metode *System Usability Scale* (SUS).

5.5.2.1 Ringkasan hasil implementasi

Tabel 5.14 Ringkasan Hasil Implementasi Sistem

Pengguna	Aktivitas Implementasi	Hasil Implementasi
Admin	Mengelola data Admin, Guru, Orang Tua, Tahun Pelajaran, Kalender Pendidikan, Kelas, Data Siswa, mencetak Biodata Siswa PDF, dan melihat rekapitulasi absensi.	Seluruh aktivitas dapat dilakukan dengan baik sehingga pengelolaan data akademik menjadi lebih terstruktur dan efisien.
Guru	Melihat Kalender Pendidikan, Daftar Siswa, Profil Siswa, melakukan input absensi, serta melihat Penilaian Anak dari RDM.	Seluruh fitur berjalan sesuai kebutuhan sehingga mendukung proses pembelajaran dan pencatatan akademik secara lebih efektif.
Orang Tua Siswa	Melihat Kalender Pendidikan, Profil Siswa, Riwayat Kehadiran, dan Penilaian Anak dari RDM.	Informasi akademik dapat diakses dengan baik sehingga orang tua dapat memantau perkembangan anak secara mandiri.

Berdasarkan hasil implementasi selama masa operasional sistem, dapat diketahui bahwa seluruh pengguna mampu memanfaatkan fitur yang tersedia sesuai dengan hak akses masing-masing. Implementasi yang dilakukan menunjukkan bahwa sistem telah berhasil mendukung proses pengelolaan data akademik, penyampaian informasi kepada orang tua, serta pencatatan aktivitas akademik di RA Al-Islam Gunungcupu. Selain itu, seluruh fitur yang digunakan selama implementasi dapat berjalan dengan baik tanpa ditemukan kendala yang memengaruhi fungsi utama sistem.

Penggunaan sistem selama masa implementasi juga memberikan pengalaman langsung kepada Admin, Guru, dan Orang Tua Siswa dalam menjalankan aktivitas akademik menggunakan Sistem Informasi Akademik berbasis web. Pengalaman tersebut menjadi dasar yang penting untuk melakukan evaluasi terhadap tingkat kemudahan penggunaan sistem menggunakan metode *System Usability Scale* (SUS) yang akan dibahas pada subbab berikutnya.

5.6 Evaluasi Menggunakan *System Usability Scale* (SUS)

Setelah Sistem Informasi Akademik RA Al-Islam Gunungcupu berhasil diimplementasikan secara nyata pada operasional sekolah, tahapan selanjutnya adalah melakukan evaluasi usabilitas (*usability evaluation*). Evaluasi ini dilakukan melalui penyajian 10 pernyataan standar yang disusun secara bergantian antara kalimat bernada positif dan negatif dengan acuan skala Likert 5 poin, di mana nilai 68,0 ditetapkan sebagai standar batas minimal untuk menentukan apakah sistem yang dievaluasi sudah masuk dalam kategori acceptable (layak pakai) atau justru masih membutuhkan perbaikan lebih lanjut (Oktavidiyanto et al., 2026).

Selain itu, metode SUS memiliki tiga karakteristik utama yang mendasarinya, yaitu instrumen ini hanya terdiri dari sepuluh pertanyaan ringkas sehingga memudahkan responden untuk memahami dan menyelesaikannya dalam waktu singkat, bersifat *technology agnostic* yang berarti sangat fleksibel dan dapat diterapkan secara luas untuk mengevaluasi berbagai jenis antarmuka, serta hasil pengukurannya dikonversi ke dalam rentang skor 0 hingga 100 sebagai standar nilai yang sangat umum dan mudah dipahami secara relatif (Ginting et al., 2025). Pengukuran ini dijadikan acuan utama untuk menilai tingkat kemudahan dan penerimaan Sistem Informasi Akademik berbasis web di RA Al-Islam Gunungcupu.

5.6.1 Instrumen kuesioner dan penentuan jumlah responden

Instrumen kuesioner SUS tersusun atas 10 pernyataan standar yang dibuat secara bergantian antara kalimat bernada positif (item ganjil) dan kalimat bernada negatif (item genap). Pengukuran dilakukan menggunakan skala Likert 5 poin, mulai dari nilai 1 (Sangat Tidak Setuju / STS), 2 (Tidak Setuju / TS), 3 (Netral / N), 4 (Setuju

/ S), hingga 5 (Sangat Setuju / SS). Daftar pernyataan instrumen SUS disajikan pada Tabel 5.15.

Tabel 5.15 Daftar Pernyataan Kuesioner *System Usability Scale* (SUS)

Kode	Pernyataan Kuesioner	Nada
P1	Saya berpikir akan menggunakan sistem ini lagi secara berkala.	Positif
P2	Saya merasa sistem ini terlalu rumit untuk digunakan.	Negatif
P3	Saya merasa sistem ini mudah untuk digunakan.	Positif
P4	Saya membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.	Negatif
P5	Saya menemukan bahwa berbagai fungsi dalam sistem ini terintegrasi dengan baik.	Positif
P6	Saya merasa ada terlalu banyak hal yang tidak konsisten pada sistem ini.	Negatif
P7	Saya merasa orang lain akan mempelajari penggunaan sistem ini dengan sangat cepat.	Positif
P8	Saya merasa sistem ini sangat membingungkan saat digunakan.	Negatif
P9	Saya merasa sangat percaya diri saat menggunakan sistem ini.	Positif
P10	Saya perlu mempelajari banyak hal terlebih dahulu sebelum saya bisa menggunakan sistem ini.	Negatif

Pengumpulan data evaluasi dilaksanakan pada akhir periode sekolah aktif bulan Juni 2026. Kuesioner disebarkan kepada 6 orang responden, yang terdiri atas 1 orang Admin, 3 orang Guru (Guru Kelompok A, Guru Kelompok B, dan Guru

Pendamping), serta 2 orang perwakilan Orang Tua Siswa. Penentuan jumlah 6 responden ini didasarkan pada dua argumentasi akademis dan metodologis:

1. Representasi total populasi pengguna internal

Jumlah 1 orang Admin dan 3 orang Guru mencakup 100% dari keseluruhan entitas pengelola dan tenaga pendidik yang beroperasi di RA Al-Islam Gunungcupu. Dengan demikian, hasil evaluasi dari kelompok pengguna internal memiliki tingkat keabsahan data murni tanpa adanya sampel yang terlewat.

2. Kecukupan sampel usabilitas menurut literatur

Pengikutsertaan 2 orang perwakilan Orang Tua Siswa melengkapi total sampel menjadi 6 responden. Jumlah sampel ini dinilai sudah valid dan reliabel, karena menurut (Ginting et al., 2025), metode SUS tidak dipengaruhi oleh ukuran sampel dan dapat diterapkan pada sampel yang sangat kecil (minimal 2 responden) dengan hasil yang tetap dapat diandalkan.

5.6.2 Aturan perhitungan dan hasil skor SUS

Kalkulasi nilai akhir pada metode SUS dilakukan melalui konversi skor kontribusi untuk setiap item pertanyaan sebelum dijumlahkan secara keseluruhan. Sesuai dengan ketentuan pada Bab III Subbab 3.16, aturan perhitungan kontribusi skor untuk tiap-tiap responden ditetapkan sebagai berikut:

1. Untuk Pertanyaan Ganjil (P1, P3, P5, P7, P9)

$$\text{Skor Kontribusi} = \text{Skor Jawaban Responden} - 1$$

2. Untuk Pertanyaan Genap (P2, P4, P6, P8, P10)

$$\text{Skor Kontribusi} = 5 - \text{Skor Jawaban Responden}$$

3. Kalkulasi Skor Akhir Responden

$$\text{Skor SUS Responden} = \left(\sum \text{Skor Kontribusi Item 1 s.d. 10} \right) \times 2,5$$

Melalui aturan di atas, total akumulasi skor kontribusi dari kesepuluh item (rentang 0–40) dikalikan dengan konstanta 2,5, sehingga menghasilkan nilai akhir SUS dalam skala 0 hingga 100. Rincian akumulasi jawaban dan hasil kalkulasi skor SUS dari 6 responden disajikan pada Tabel 5.16.

Tabel 5.16 Hasil Perhitungan Skor SUS Seluruh Responden

Responden	Peran/Hak Akses	Total Skor Kontribusi P1 s.d P10)	Skor Akhir SUS (Skor × 2, 5)
R1	Admin	35	87,50
R2	Guru Kelompok A	34	85,00
R3	Guru Kelompok B	33	82,50
R4	Guru Pendamping	34	85,00
R5	Orang Tua Siswa (1)	32	80,00
R6	Orang Tua Siswa (2)	32	80,00
Rata-Rata Keseluruhan Pengguna		33,33	83,33

Berdasarkan hasil kalkulasi individual pada Tabel 5.16, diperoleh rata-rata skor SUS per kelompok pengguna yaitu: kelompok Admin sebesar 87,50, kelompok Guru sebesar 84,17, dan kelompok Orang Tua Siswa sebesar 80,00. Secara keseluruhan, Rata-Rata Skor SUS (Mean SUS Score) dari 6 responden pada Sistem Informasi Akademik RA Al-Islam Gunungcupu diperoleh sebesar 83,33.

5.6.3 Analisis dan pembahasan kelayakan usability

Untuk menentukan tingkat kelayakan dan kemudahan penggunaan aplikasi secara akademis, rata-rata skor SUS sebesar 83,33 dievaluasi berdasarkan kriteria batas minimal nilai 68,0 serta kriteria *Acceptability Ranges*, *Grade Scale*, dan *Adjective Rating*.

1. Berdasarkan *acceptability ranges*

Nilai rata-rata 83,33 berada jauh di atas standar batas minimal kelayakan usabilitas yaitu 68,0 (Oktavidianto et al., 2026). Dengan demikian, Sistem Informasi Akademik RA Al-Islam Gunungcupu masuk ke dalam kategori *Acceptable* (Sangat Layak Pakai).

2. Berdasarkan *grade scale*

Skor 83,33 mengindikasikan bahwa kualitas antarmuka dan fungsionalitas sistem berada pada tingkatan Grade A, yang mencerminkan tingkat kepuasan tinggi dari pengguna akhir.

3. Berdasarkan *adjective rating*

Pencapaian nilai ini memosisikan usabilitas sistem pada kategori *Excellent* (Sangat Bagus).

Secara analitis, tingginya skor usabilitas yang diraih (83,33) dipengaruhi oleh beberapa faktor teknis dan metodologis selama proses pengembangan sistem berbasis *Scrum*.

1. Faktor pertama adalah keterlibatan pengguna secara iteratif selama penerapan metode *Scrum* dari *Sprint* 1 hingga *Sprint* 3, yang memungkinkan seluruh umpan balik stakeholder diserap dan ditindaklanjuti secara cepat. Sebagai contoh, penyesuaian antarmuka berupa penyempurnaan format cetak PDF Buku Induk pada *Sprint* 2 terbukti secara nyata mampu meningkatkan rasa familiaritas pengguna saat mengoperasikan sistem.
2. Faktor kedua berkaitan dengan kejelasan tata letak dan aksesibilitas antarmuka. Penggunaan pustaka React JS dan Shaden UI berhasil menghasilkan antarmuka yang responsif, bersih, dan konsisten di berbagai perangkat. Kualitas rancangan ini secara efektif meminimalkan tingkat kebingungan pengguna, yang dibuktikan dengan rendahnya bobot jawaban responden pada pertanyaan-pertanyaan bernada negatif seperti P2, P6, dan P8.

3. Faktor ketiga adalah peningkatan efisiensi operasional melalui integrasi data Rapor Digital Madrasah (RDM) dan otomatisasi rekapitulasi absensi bulanan yang mampu mengurangi beban pencatatan berulang secara signifikan. Kemudahan ini dirasakan langsung oleh pihak Guru dan Admin sehingga menaikkan rasa percaya diri mereka dalam mengoperasikan aplikasi, sebagaimana tercermin pada tingginya skor jawaban untuk item P3, P5, dan P9.

Dengan demikian, hasil evaluasi menggunakan metode *System Usability Scale* (SUS) membuktikan secara ilmiah bahwa pengerjaan Sistem Informasi Akademik berbasis web pada RA Al-Islam Gunungcupu telah berhasil mencapai tujuannya. Sistem ini tidak hanya berfungsi secara teknis tanpa kesalahan logika (*bug-free*), tetapi juga memberikan tingkat kemudahan dan kenyamanan penggunaan yang sangat baik bagi Admin, Guru, maupun Orang Tua Siswa.

BAB VI

PENUTUP

6.1 Simpulan

Berdasarkan hasil penelitian mengenai Sistem Informasi Akademik Berbasis Web Menggunakan Metode Scrum pada RA Al-Islam Gunungcupu, maka dapat disimpulkan sebagai berikut:

1. Sistem Informasi Akademik berbasis web berhasil dirancang dan dibangun pada RA Al-Islam Gunungcupu untuk membantu mengatasi pengelolaan data akademik yang sebelumnya dilakukan secara manual. Sistem ini telah memenuhi kebutuhan pengguna dalam mendukung proses pengelolaan data akademik secara lebih efektif.
2. Metode *Scrum* berhasil diterapkan dalam proses pengembangan sistem melalui tahapan *Product Backlog*, *Sprint Planning*, tiga *Sprint* yang dilakukan secara bertahap. Setiap *Sprint* menghasilkan fitur yang kemudian diuji menggunakan *Black Box Testing* serta dievaluasi melalui *Sprint Review* dan *Sprint Retrospective* sehingga pengembangan dapat disesuaikan dengan kebutuhan pengguna.
3. Sistem berhasil diimplementasikan dan digunakan oleh Admin, Guru, dan Orang Tua Siswa sesuai dengan hak akses masing-masing. Hasil evaluasi menggunakan *System Usability Scale* (SUS) memperoleh skor rata-rata 83,33 yang termasuk kategori *Acceptable*, Grade A, dan *Excellent*, sehingga sistem dinilai dapat diterima dan mudah digunakan oleh pengguna.

6.2 Saran

Berdasarkan hasil penelitian dan implementasi Sistem Informasi Akademik berbasis web pada RA Al-Islam Gunungcupu, terdapat beberapa saran yang dapat dijadikan bahan pengembangan selanjutnya, yaitu:

1. Sistem dapat dikembangkan lebih lanjut dengan menambahkan fitur administrasi keuangan sekolah yang belum termasuk dalam penelitian ini,

sehingga pengelolaan data akademik dan kebutuhan pengelolaan sekolah dapat terintegrasi secara lebih lengkap.

2. Integrasi data Penilaian Anak dengan Rapor Digital Madrasah (RDM) dapat dikembangkan lebih lanjut, terutama dalam proses sinkronisasi dan pengambilan data, agar penyajian informasi akademik dapat dilakukan secara lebih optimal tanpa mengubah sumber data utama yang digunakan oleh sekolah.
3. Evaluasi penggunaan sistem dapat dilakukan secara berkala dengan periode implementasi yang lebih panjang, sehingga pihak sekolah dapat mengetahui kebutuhan perbaikan dan pengembangan sistem berdasarkan pengalaman penggunaan Admin, Guru, dan Orang Tua Siswa dalam kegiatan operasional sehari-hari.

DAFTAR PUSTAKA

- Ahmad, N., Krisnanik, E., Rupilele, F. G. J., Muliawati, A., Syamsiyah, N., Kraugusteeliana, K., Cahyono, B. D., Sriyeni, Y., Kristanto, T., & Irwanto, I. (2022). *ANALISA & PERANCANGAN SISTEM INFORMASI BERORIENTASI OBJEK*. Penerbit Widina.
<https://books.google.co.id/books?id=wSSFEAAAQBAJ>
- Aldi, M., & Aji, M. I. (2025). Perancangan Sistem Digital Surat Masuk Dan Surat Keluar Di Uptd Kesmavet Berbasis Web. *Jurnal Multidisiplin Ilmu Akademik*, 2(2), 65–77.
- Amirudin, J., Herlina, E., & Nuraeni, H. S. (2024). Penerapan Metode Pembiasaan Sholat Pada Anak Usia Dini:(Studi di Raudhatul Athfal Al-Ittihad Kabupaten Garut Provinsi Jawa Barat). *IHSAN: Jurnal Pendidikan Islam*, 2(2), 82–90.
- Andarsyah, R., Pratama, C. Y., & Kishendrian, H. D. (2022). Implementasi Code Coverage Pada Chatbot Telegram Sebagai Media Alternatif Sistem Informasi. *Jurnal Teknik Informatika*, 14(2), 112–117.
- Ardiansyah, F. (2023). Pengembangan Sistem Informasi Keanggotaan Online Berbasis Web menggunakan Framework Laravel dengan Metode Prototype pada Asosiasi Inkindo. *Journal of Research and Publication Innovation*, 1(2), 266–271.
- Ariansyah, P. M., & Wijaya, K. (2021). Rancang Bangun Sistem Informasi Akademik Berbasis Web: Studi Kasus: SD Negeri 18 Tanah Abang. *Jurnal Pengembangan Sistem Informasi Dan Informatika*, 2(3), 138–156.
- Arifin, N. Y., Kom, S., Kom, M., Tyas, S. S., Kom, S., Sulistiani, H., Kom, M., Hardiansyah, A., Suri, G. P., & Kom, M. (2022). *Analisa Perancangan Sistem Informasi*. Cendikia Mulia Mandiri.
- Asih Della Saftini. (2024). Perancangan Sistem Informasi Penjualan Sembako pada Toko Misman. *Switch : Jurnal Sains Dan Teknologi Informasi*, 3(1), 142–147.
<https://doi.org/10.62951/switch.v3i1.339>
- Banjarnahor, W. S. A., & Sinuraya, J. (2025). *BUKU AJAR: PRAKTIK PENGUJIAN PERANGKAT LUNAK*. Faaslib Serambi Media.
<https://books.google.co.id/books?id=UrCcEQAAQBAJ>
- Bardadi, H. S. A. D. F. A. A. (2023). *Pengembangan Aplikasi Berbasis Web*. Media Nusa Creative (MNC Publishing).
<https://books.google.co.id/books?id=7rruEAAAQBAJ>
- Destriana, R., Husain, S. M., Handayani, N., & Siswanto, A. T. P. (2022). *Diagram UML Dalam Membuat Aplikasi Android Firebase “Studi Kasus Aplikasi Bank Sampah.”* Deepublish. <https://books.google.co.id/books?id=6bMEQAAQBAJ>

- Erawati, W., Heristian, S., & Purnama, R. A. (2023). Rancang bangun sistem informasi akademik berbasis website dengan metode SDLC. *Computer Science (CO-SCIENCE)*, 3(2), 68–77.
- Giandari Maulani, S. K. M. K., Lilik Hari Santoso, S. S. M. T., Dr. Ali Ibrahim, M. T., Nurul Chafid, S. K. M. K., Arief Hidayat, S. K. M. K., Mustakim, S. K. M. K., Farid Fitriyadi, S. K. M. K., Anyan, S. K. M. K., Oleh Soleh, S. K. M. M. S. I., & Ade Suparman, S. S. I. M. K. (2024). *REKAYASA PERANGKAT LUNAK*. Cendikia Mulia Mandiri. https://books.google.co.id/books?id=__zEAAQBAJ
- Ginting, G. L., Hondro, R. K., & Fau, A. (2025). Implementasi System Usability Scale Guna Analisis Usability Aplikasi Moka. *Informatics*, 1(01), 6–14.
- Halim, Muh. F. S. S. I. M. P., & Rahmayanti, N. S. M. H. (2022). *Pelayanan Perubahan Data Pada Sistem Informasi Akademik Di Perguruan Tinggi*. Bening Media Publishing. <https://books.google.co.id/books?id=EemnEAAQBAJ>
- Indra Gunawan, M. K. S. M. K. H. S. T. M. K. D. D. H. S. T. M. K. (2022). *MONOGRAF ALGORITMA TABU SEARCH DALAM KASUS TRAVELING SALESMAN PROBLEM*. Penerbit Adab. <https://books.google.co.id/books?id=5ip9EAAQBAJ>
- Iswari, L. (2025). *Penerapan React JS Pada Pengembangan FrontEnd*.
- Jurnaidi, A., Ahmad, W., Wahidin, J., Miftachurohmah, N., Alfa, D., Daniel, P., Puryono, A., Hanes, H., Agung, R., Rajif, Y., Yunmar, A., Br, S., Salsalina, S., Sembiring, B., Jayawardana, H., Surya, R., Ridho, K., Kusuma, S., Nuril, H., ... Rahayu, K. (2025). *DATABASE DATABASE Penulis : Penulis*.
- Lukman, L., Budiman, T., Kurniawan, E., & Hasibuan, D. R. (2023). Rancang bangun sistem informasi manajemen proyek pada PT ABC. *Jurnal Manajemen Informatika Jayakarta*, 3(2), 128–141.
- Magdalena, L., Efitra, E., & Safitri, N. (2023). *SCRUM AGILE : OPTIMALISASI KUALITAS PRODUK MANAJEMEN*. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=5eXMEAAQBAJ>
- Mardiati, D., & Saputra, Y. (2025). Implementasi sistem informasi manajemen klinik menggunakan metode black box testing. *Jurnal Informatika Dan Teknik Elektro Terapan*, 13(1).
- Maulana, H., Dwicahya Supriatman, R., Rohayati, T., & Yudi Permana, N. (2026). *Development and Evaluation of an Adaptive Scrum Model for Wedding Organizer Systems Using UML and ISO 25010*. 8(2), 89–105. <https://doi.org/10.20895/INISTA.V8I2.2124>
- Marlina Ariansyah, P., & Wijaya Sekolah Tinggi Manajemen Informatika dan Komputer Prabumulih, K. (2021). Rancang Bangun Sistem Informasi

- Akademik Berbasis Web: Studi Kasus: SD Negeri 18 Tanah Abang. In *Jurnal Pengembangan Sistem Informasi dan Informatika* (Vol. 2, Number 3).
- Maulana, M. N., & Prianto, C. (2023). *Pembuatan website layanan keuangan dengan metode scrum*. Penerbit Buku Pedia. <https://books.google.co.id/books?id=7cy4EAAQBAJ>
- Nadhira, F., Wahyuddin, M. I., & Sari, R. T. K. (2022). Penerapan metode agile scrum pada rancangan SisIAM4. *Jurnal Media Informatika Budidarma*, 6(1), 560.
- Namruddin, R., Basalamah, A., Ali, M. Z. A., Syarifuddin, A., Alam, S., Wardhani, N., & Abdurrahman, T. S. D. (2023). *BELAJAR DATABASE DENGAN MUDAH MENGGUNAKAN MYSQL*. TOHAR MEDIA. <https://books.google.co.id/books?id=mHnEEAAQBAJ>
- Nuh, M. (2022). Penyuluhan Mengelola Website Sebagai Media Publikasi, Komunikasi Dan Informasi Pada Pesantren Hidayatullah Jonggol. *Jurnal Pedes-Pengabdian Bidang Desain*, 2(1), 110–117.
- Nur Fitrianiingsih Hasan, M. K., Vera Wati, S. K. M. K., Shella Gilby Sapulette, S. E. M. S., Dr. Sri Supadmini SE., M. M., Wartono, S. K. M. K., Franco Benony Limba SE., M. A., Isfaatun, E., Purwanto S. E., M. C., Wico J Tarigan, S. E. M. S., & Ade Suparman, S. S. I. M. K. (2023). *Dasar Analisa Perancangan Sistem Informasi*. Cendikia Mulia Mandiri. <https://books.google.co.id/books?id=Upy-EAAQBAJ>
- Nurhadi, N., Jatmiko, A. R., Legito, L., Saputra, E. A., syafa'at, F., Surianto, D. F., Komalasari, R., Mukhlis, I. R., & Sulistyowati, S. (2023). *BUKU AJAR LOGIKA & ALGORITMA*. PT. Sonpedia Publishing Indonesia. <https://books.google.co.id/books?id=n2XOEAAQBAJ>
- Nurmasani, A., Setiawan, A., & Hartanto, A. D. (2024). Pengembangan Fitur Rekognisi Kegiatan Dengan Metode Scrum. *Information System Journal*, 7(02), 95–103.
- Oktavidianto, T. R. R., Chan, M. D. C., Servanda, Y., & Idris, N. (2026). Analisis Usabilitas Pada Website Perpustakaan Universitas Mulia Menggunakan System Usability Scale. *JOURNAL SAINS STUDENT RESEARCH*, 4(1), 219–227.
- Prayoga, J., Sinurat, S., Rachman, A., Carolina, I., Jumaryadi, Y., Irmayana, A., Supriyatna, A., & Meilani, D. (2023). *Sistem Basis Data GRAHA MITRA EDUKASI*.
- Putri, M. P., Nadeak, E., Malahayati, M., Rahmi, N., Rini, A., Sari, D. N., Kurniati, K., Kusmiati, H., & Pratama, R. A. A. (2023). *sistem manajemen basis data menggunakan MYSQL*.
- Ramdany, S. W., Kaidar, S. A., Aguchino, B., Amelia, C., Putri, A., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi

- Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1), 30–41.
- Ruswa Dwipa, U. U. O. A. G. (2021). *RANCANG BANGUN APLIKASI PENYEWAAN ALAT BERAT PADA PT. 1001 NIAN BERBASIS WEB*. <https://jurnal.uss.ac.id/index.php/klik/article/download/300/139>
- Santoso, L., & Amanullah, J. (2022). *PENGEMBANGAN SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT (RAD)*. 15(2), 250–259. <http://journal.stekom.ac.id/index.php/elkom> page250
- Sastradipraja, C. K., & Munawar, Z. (2022). *Konsep Dasar Teknologi Web*. Kaizen Media Publishing. <https://books.google.co.id/books?id=E16YEAAAQBAJ>
- Sinlae, F., Steno Birama, P., Ardian Nugraha Siregar, D., Safriadi, W., & Tawakal, H. (2024). Design dan implementasi sistem informasi pembelian properti berbasis web menggunakan framework Laravel. *Jurnal Siber Multi Disiplin (JSMD)*, 2(2), 152–157.
- Siregar, F. A. (2025). *Analisis dan Desain Sistem*. umsu press. <https://books.google.co.id/books?id=wi5mEQAAQBAJ>
- Subagyo, U., & Jannah, M. (2025). Pengembangan Sistem Informasi Akademik Dosen dan Mahasiswa dengan Metode Scrum. *Jurnal Informatika Komputer, Bisnis Dan Manajemen*, 23(1), 108–117.
- Sulistiyorini, T., Sova, E., & Ramadhan, R. (2022). Pemantauan kasus penyebaran Covid-19 berbasis website menggunakan framework React JS dan API. *Jurnal Ilmiah Multidisiplin*, 1(04), 1–13.
- Sururiyah, M. P. (2023). *Manajemen Pendidikan Karakter Berbasis Kearifan Lokal: Untuk Raudhatul Athfal*. PT Arr rad Pratama, IAINU Kebumen Press.
- Suwanda, R., Siregar, A. M., Kurniawan, H., P, I. S., Andisana, S., Wulansari, T. T., KW, P. P. O. J., Maknunah, J., Wahyuningsih, D., & Dewa, W. A. (2025). *Analisis dan Perancangan Sistem*. Mifandi Mandiri Digital. <https://books.google.co.id/books?id=AsDMEQAAQBAJ>
- Syahputra, G., Calam, A., Nugroho, C., Faisal, F., & Syafrizal, S. (2021). Pembuatan Website Stkip Amal Bakti. *JURNAL PRODIKMAS Hasil Pengabdian Kepada Masyarakat*, 6(1), 43–48. <http://download.garuda.kemdikbud.go.id/article.php?article=2248307&val=11583&title=PEMBUATAN%20WEBSITE%20STKIP%20AMAL%20BAKTI>
- Ummy Gusti Salamah, S. S. (2021). *Tutorial Visual Studio Code*. Media Sains Indonesia.

- Wahidin, U., Sarbini, M., Maulida, A., & Wangsadanureja, M. (2021). Implementasi Pembelajaran Agama Islam Berbasis Multimedia Di Pondok Pesantren. *Edukasi Islami: Jurnal Pendidikan Islam*, 10(01), 21–32.
- Wayahdi, M. R., & Ruziq, F. (2023). Pemodelan sistem penerimaan anggota baru dengan unified modeling language (UML)(Studi kasus: Programmer Association of Battuta). *Jurnal Minfo Polgan*, 12(1), 1514–1521.
- Wibowo, M. Z. (2023). Implementasi pendidikan karakter tanggung jawab mampu meningkatkan hasil belajar siswa. *Jurnal Bintang Pendidikan Indonesia*, 1(1), 76–83.
- Zulfa, A. A., & Arifudin, O. (2025). PERAN SISTEM INFORMASI AKADEMIK BERBASIS WEB DALAM UPAYA MENINGKATKAN EFEKTIVITAS DAN EFISIENSI PENGELOLAAN AKADEMIK DI PERGURUAN TINGGI. In *Jurnal Tahsinia* (Vol. 6, Number 1).

LAMPIRAN

Lampiran 1. Surat Bukti Implementasi Sistem



**YAYASAN AL-ISLAM GUNUNGCUPU
RAUDHATULATHFAL (RA) AL-ISLAM**

*Dusun Sirnagalih RT 038 RW 018, Kel. Gunungcupu, Kec. Sindangkasih,
Kab. Ciamis, Jawa Barat*

Nomor : RA.043/SL/15/V/2026
Lampiran : 1 (satu) Lembar
Perihal : **Persetujuan Implementasi Aplikasi
Tugas Akhir Sistem Informasi
UNIGAL CIAMIS**

Ciamis, 20 Mei 2026

Kepada Yth.
Ketua Program Studi Sistem Informasi
Universitas aluh Ciamis
di
Tempat

Assalamualaikum Wr. Wb.

Dengan hormat, sehubungan dengan permohonan izin implementasi aplikasi Tugas Akhir jalur prototipe hilirisasi teknologi tepat guna oleh mahasiswa Program Studi Sistem Informasi Universitas Galuh Ciamis, maka dengan ini kami atas nama Kepala RA Al-Islam Gunungcupu memberikan persetujuan kepada mahasiswa yang bersangkutan untuk melaksanakan implementasi sistem di lingkungan sekolah kami.

Kegiatan implementasi ini direncanakan akan berlangsung selama kurang lebih 1 (satu) bulan, terhitung mulai tanggal **25 Mei 2026** sampai dengan selesai. Adapun data detail mengenai mahasiswa, judul penelitian, serta nama aplikasi yang akan diimplementasikan tercantum dalam lampiran surat ini.

Demikian surat persetujuan ini kami sampaikan agar dapat dipergunakan sebagaimana mestinya. Atas perhatian dan kerja sama yang baik, kami ucapkan terima kasih.

Wassalamualaikum Wr. Wb.



LILIS FARIDA, S.Pd.I

NIP. 198004132005012003

LAMPIRAN

Nomor: RA.043/SL/15/V/2026

Tanggal: 20 Mei 2026

Berikut adalah data mahasiswa Universitas Galuh Ciamis beserta judul tugas akhir dan aplikasi yang diimplementasikan di RA Al-Islam Gunungcupu:

No	Nama & NIM	Judul Tugas Akhir (Prototipe Hilirisasi)	Nama Aplikasi
1	Bayu Septiana Aziz NIM. 7020220021	Implementasi Sistem Informasi Manajemen Administrasi Guru Berbasis <i>Chatbot</i> dengan Metode Large Language Model di Raudhatul Athfal Menggunakan Metode <i>Agile</i>	AISYA-RA (AI System untuk Administrasi Raudhatul Athfal)
2	Rizky Maulana NIM. 7020220041	Sistem Informasi Akademik Berbasis Web Menggunakan Metod Scrum Pada Raudhatul Athfal Al-Islam Gunungcupu	SIKAD RA AL-ISLAM Sistem Informasi Akademik)



Kepala RA Al-Islam,



LILIS FARIDA, S.Pd.I

NIP. 198004132005012003