

BAB V

HASIL DAN PEMBAHASAN

5.1 Studi Awal

Tahap studi awal dilaksanakan sebagai langkah pertama penelitian, sebelum proses pengumpulan data di lapangan dimulai. Studi awal bertujuan membangun pemahaman teoritis yang memadai mengenai domain permasalahan, pendekatan teknologi yang akan digunakan, serta metode pengembangan yang akan diterapkan, sehingga proses identifikasi kebutuhan di lapangan dapat dilakukan secara lebih terarah dan tidak membias. Studi awal dalam penelitian ini mencakup dua kegiatan: studi literatur dan observasi awal lapangan.

5.1.1 Studi Literatur

Studi literatur dilaksanakan dengan mengkaji penelitian-penelitian terdahulu yang relevan dengan topik sistem informasi administrasi pendidikan, pemanfaatan *Chatbot* dan *Large Language Model (LLM)* dalam konteks administrasi dan pembelajaran, serta penerapan metode *Agile* pada pengembangan sistem informasi institusional. Kajian penelitian terdahulu dibahas secara mendalam pada Bab II, sedangkan landasan teori yang mendasari perancangan AISYA-RA diuraikan pada Bab III. Pada bagian ini disampaikan ringkasan temuan utama yang secara langsung membentuk kerangka konseptual perancangan sistem:

1. **Digitalisasi Administrasi PAUD/RA:** Fiarni et al. (2024) mengembangkan sistem informasi pemantauan PAUD berbasis *waterfall* yang berhasil mencapai nilai *User Acceptance Test (UAT)* sebesar 80%, namun masih menggunakan mekanisme input data manual. Nurjannah et al. (2024) mengembangkan sistem informasi administrasi guru SMK berbasis SDLC yang berhasil mendigitalisasi 24 poin administrasi, namun seluruhnya berbasis antarmuka GUI (*Graphical User Interface*) dengan hierarki menu berlapis. Kedua penelitian mengindikasikan bahwa digitalisasi administrasi pendidikan telah berjalan, namun pendekatan antarmuka yang digunakan

belum mempertimbangkan keterbatasan literasi digital pengguna di lembaga pendidikan anak usia dini.

2. **Perilaku Adopsi Teknologi Pendidik PAUD:** Merjovaara et al. (2024) melalui studi empiris menggunakan kerangka *Technological Pedagogical Content Knowledge (TPACK)* menemukan bahwa *perceived ease of use* (persepsi kemudahan penggunaan) merupakan prediktor terkuat yang memengaruhi niat pendidik dalam mengadopsi teknologi, lebih kuat dibandingkan faktor kegunaan fungsional. Temuan ini menjadi justifikasi ilmiah bahwa keberhasilan adopsi sistem digital di lingkungan PAUD sangat ditentukan oleh seberapa familier dan mudah antarmuka yang ditawarkan.
3. **Pemanfaatan AI dan Chatbot dalam Pendidikan:** Labadze et al. (2023) melalui *Systematic Literature Review* menemukan bahwa *Chatbot* berbasis AI memiliki potensi besar membantu pendidik dalam memberikan bantuan cepat, personal, serta menghemat waktu kerja secara signifikan. Kajian tersebut sekaligus memperingatkan risiko halusinasi informasi yang perlu dikelola melalui perancangan *system prompt* yang terstruktur dan pembatasan konteks model.
4. **Otomasi Dokumen Pembelajaran Berbasis LLM:** Zheng et al. (2025) melalui kerangka kerja LessonPlanLM membuktikan bahwa LLM mampu menghasilkan rencana pelaksanaan pembelajaran secara otomatis dengan integritas struktural dan akurasi tinggi apabila disertai konteks yang memadai. Temuan ini mengonfirmasi bahwa LLM dapat dioptimalkan untuk menghasilkan dokumen administratif berstruktur apabila *prompt* dirancang dengan menyertakan konteks format dan pedagogis yang relevan.
5. **Efektivitas Pengembangan Berbasis Agile:** Hidayah Nova et al. (2022) menyimpulkan bahwa metode *Agile* terbukti efektif karena melibatkan pengguna secara langsung pada setiap iterasi, menghasilkan produk yang lebih sesuai kebutuhan nyata, dan memungkinkan penyesuaian terhadap

perubahan kebutuhan yang muncul setelah sistem mulai digunakan. Kesimpulan ini memperkuat alasan pemilihan *Agile* sebagai metode pengembangan AISYA-RA, mengingat kebutuhan pengguna di RA Al-Islam Gunungcupu diprediksi akan berkembang seiring pengguna melihat dan mencoba prototipe sistem.

Sintesis dari kelima temuan membentuk kerangka konseptual perancangan AISYA-RA, yaitu:

1. Sistem harus memiliki antarmuka yang menurunkan hambatan adopsi, menjustifikasi penerapan *Conversational User Interface (CUI)* berbasis *Chatbot*.
2. LLM dapat menghasilkan dokumen berkualitas tinggi apabila instruksi *prompt* disertai konteks format baku lembaga yang tepat.
3. Risiko halusinasi dikelola melalui pendekatan *Retrieval-Augmented Generation (RAG)* berbasis dokumen lokal madrasah.
4. Metode *Agile* merupakan pendekatan paling adaptif karena kebutuhan pengguna dirumuskan dan disempurnakan secara bertahap melalui iterasi nyata.

5.1.2 Observasi Awal Lapangan

Sebelum pengumpulan data formal dilaksanakan, observasi awal lapangan dilakukan di Raudhatul Athfal (RA) Al-Islam Gunungcupu pada kunjungan pertama. Observasi awal bersifat eksploratif dan tidak terstruktur, bertujuan membangun gambaran umum kondisi operasional lembaga yang menjadi acuan dalam menyusun pedoman observasi dan panduan wawancara.

Dari observasi awal diperoleh gambaran kondisi lembaga:

1. RA Al-Islam Gunungcupu beroperasi dengan dua rombongan belajar (Kelompok A dan Kelompok B) dan dikelola oleh satu Kepala RA, tiga tenaga pendidik, dan satu tenaga kependidikan.

2. Tidak terdapat ruang khusus administrasi; kegiatan administrasi dilakukan di dalam kelas, di ruang guru, atau di ruang kepala madrasah. Satu unit laptop tersedia di ruang kepala sekolah dan sesekali digunakan untuk mengetik surat dinas.
3. Seluruh tenaga pendidik terlihat menggunakan *smartphone* secara aktif. Aplikasi WhatsApp digunakan secara intensif untuk komunikasi antarguru, koordinasi dengan Kepala RA, dan komunikasi dengan orang tua siswa.
4. Tidak ditemukan penggunaan aplikasi administrasi digital apapun selama observasi awal. RPPH (Rencana Pelaksanaan Pembelajaran Harian) tampak dalam bentuk berkas kertas yang disusun di meja guru, beberapa di antaranya tampak identik dengan hari-hari sebelumnya.

Temuan observasi awal ini memberikan dasar yang kuat untuk merancang pengumpulan data yang lebih terstruktur:

1. Seluruh pengguna potensial telah sangat terbiasa dengan antarmuka berbasis pesan teks melalui WhatsApp, menjustifikasi pendekatan CUI (*Conversational User Interface*).
2. Belum tersedia pangkalan data digital yang memuat identitas lembaga, data guru, dan data siswa secara terstruktur.
3. Beban administrasi manual guru yang tinggi terlihat jelas dari kondisi meja kerja yang dipenuhi berkas dan buku catatan fisik.

5.2 Pengumpulan Data

Pengumpulan data formal dilaksanakan di RA Al-Islam Gunungcupu menggunakan tiga teknik yang saling melengkapi, observasi terstruktur, wawancara semi-terstruktur, dan studi dokumentasi. Ketiga teknik dijalankan secara berurutan dalam periode yang berdekatan agar data yang diperoleh saling mengonfirmasi dan memperkuat.

5.2.1 Observasi

Observasi terstruktur dilaksanakan selama tiga hari kerja berturut-turut, dimulai sejak guru tiba di sekolah (± 07.00 WIB) hingga selesai jam pelajaran (± 11.00 WIB). Fokus pengamatan meliputi lima aspek, penyusunan RPPH, pencatatan presensi dan rekap kehadiran, pembuatan surat resmi, pengarsipan dokumen, serta pola penggunaan perangkat teknologi.

5.2.1.1 Temuan Observasi: Proses Penyusunan RPPH

Guru umumnya menyusun RPPH pada pagi hari sebelum siswa datang atau setelah jam pelajaran selesai. Alur kerja yang terekam adalah, guru membuka dokumen RPPH hari sebelumnya di laptop, kemudian menyalin dan mengubah bagian tema, sub-tema, dan kegiatan inti secara manual. Apabila laptop tidak tersedia, guru menulis RPPH secara manual di lembaran kertas. Rata-rata waktu yang dibutuhkan adalah 40 hingga 55 menit per dokumen.

Hambatan yang teramati antara lain:

1. Tidak terdapat templat digital terstruktur.
2. Komponen "Kegiatan Inti" yang harus dikarang ulang setiap hari merupakan bagian paling memakan waktu.
3. Pada dua dari tiga hari observasi, guru menyalin kegiatan inti dari hari sebelumnya tanpa perubahan substantif karena keterbatasan waktu.
4. Komponen penilaian hampir tidak pernah diisi.

5.2.1.2 Temuan Observasi: Pencatatan Presensi Siswa dan Kehadiran Staf

Pencatatan kehadiran siswa dilakukan menggunakan buku hadir berbentuk buku folio bergaris yang berbeda untuk setiap kelompok, dengan tanda centang (hadir), I (izin), S (sakit), dan A (alfa). Proses pencatatan harian per kelas memerlukan 5 menit.

Selain pencatatan harian, terdapat kewajiban membuat rekapitulasi presensi dalam tiga periode: rekap harian di buku jurnal kelas, rekap mingguan setiap hari Sabtu untuk Kepala RA, serta rekap bulanan yang paling menyita waktu. Pada rekap bulanan, guru menghitung ulang seluruh tanda absensi secara manual dari buku fisik lalu menjumlahkannya per siswa ke dalam tabel rekap. Proses ini memerlukan waktu 30 hingga 60 menit per kelas dan sangat rentan terhadap kesalahan hitung (*human error*).

Di samping itu, observasi menemukan bahwa tidak ada mekanisme pencatatan kehadiran guru dan Kepala RA yang terstruktur dan terdokumentasi, sehingga menjadi celah administratif bagi akuntabilitas madrasah. Ketika ditanyakan pada sesi wawancara, Kepala RA mengonfirmasi:

"Kita saling tahu siapa yang hadir dan siapa yang tidak, tapi memang tidak ada buku khususnya. Kalau ada yang izin, biasanya langsung kabari lewat WhatsApp."

— Kepala RA (R-01), Wawancara

Ketiadaan rekam jejak kehadiran guru dan Kepala RA menimbulkan persoalan akuntabilitas administratif, terutama apabila lembaga menghadapi keperluan supervisi atau akreditasi yang membutuhkan data kehadiran staf secara historis. Temuan ini dicatat sebagai kebutuhan administratif yang perlu dipertimbangkan dalam penyusunan dan pembaruan *backlog*, tanpa langsung menetapkan bentuk fitur final pada tahap awal.

5.2.1.3 Temuan Observasi: Pembuatan Surat Resmi

Proses pembuatan surat resmi diperagakan oleh Kepala RA, membuka berkas surat lama di laptop, memilih dokumen yang formatnya paling mendekati, menghapus isi lama, mengetik ulang kop surat dari awal, dan mengisi isi surat baru. Proses ini memerlukan waktu 25 hingga 40 menit per dokumen.

Hambatan yang teramati antara lain:

1. Kop surat diketik ulang setiap kali, Kepala RA mengakui "sering beda-beda tulisannya, kadang pakai singkatan, kadang tidak"
2. Nomor surat ditentukan dengan membuka folder arsip lama, melihat nomor terakhir, kemudian menambah satu, proses rentan kesalahan
3. Kepala RA menyebutkan pernah terjadi kasus dua surat dengan nomor yang sama dikirimkan kepada pihak yang berbeda.

5.2.1.4 Temuan Observasi: Pengarsipan dan Pencarian Dokumen

Arsip dokumen tersimpan dalam folder laptop tanpa konvensi penamaan yang teratur (misalnya *RPPH baru.docx*, *surat 2.docx*) dan sebagian berupa tumpukan fisik di map plastik tanpa indeks. Pencarian dokumen lama membutuhkan waktu 10 hingga 30 menit, dan beberapa dokumen bahkan tidak berhasil ditemukan kembali.

5.2.1.5 Temuan Observasi: Pola Penggunaan Teknologi

Seluruh tenaga pendidik aktif menggunakan *smartphone* Android untuk komunikasi WhatsApp harian. Tidak ada hambatan dalam menggunakan antarmuka berbasis teks, seluruh responden tampak nyaman mengetik pesan panjang dan membaca informasi dari layar *smartphone*.

Sebaliknya, penggunaan laptop dihindari karena tiga faktor utama:

1. Keterbatasan keterampilan teknis, sebagian besar guru tidak terbiasa mengoperasikan komputer di luar fungsi mengetik teks sederhana. Navigasi antarmuka berbasis menu, membuka dan menyimpan file di folder tertentu, serta mengoperasikan sistem operasi secara mandiri sudah melampaui zona kenyamanan sebagian responden.
2. Ketakutan merusak perangkat, laptop yang tersedia merupakan satu-satunya unit komputer lembaga dan dinilai sebagai barang mahal. Kekhawatiran "salah klik" dan mengakibatkan kerusakan membuat beberapa guru enggan menyentuh laptop. R-04 menyatakan:

"Saya takut nanti error, atau salah hapus file penting. Lebih baik tidak usah, nanti malah rugi."

— R-04, Tenaga Kependidikan, Wawancara

3. Ketidakhadiran menggunakan aplikasi pengolah kata, meskipun Microsoft Word paling sering diperlukan, sebagian guru merasa tidak mahir menggunakannya. Format teks, ukuran kertas, dan tabel di Word kerap memberikan hasil yang tidak sesuai. R-03 mengungkapkan:

"Word itu sering tiba-tiba formatnya berantakan sendiri. Sudah susah payah dirapikan, begitu dibuka lagi beda lagi. Bikin frustrasi."

— R-03, Guru Kelompok B, Wawancara

Temuan ketiga hambatan ini memiliki implikasi desain krusial, antarmuka berbasis menu atau formulir di komputer bukan pilihan tepat untuk konteks ini. Sistem yang dibangun harus dapat digunakan melalui *smartphone*, menggunakan antarmuka menyerupai komunikasi teks yang sudah familier, dan tidak menuntut pengguna membuka atau mengoperasikan aplikasi pengolah kata sama sekali. Fakta inilah yang secara langsung menjustifikasi pemilihan *Conversational User Interface (CUI)* berbasis *Chatbot* sebagai antarmuka utama AISYA-RA.

Ringkasan durasi rata-rata dan hambatan utama setiap tugas administrasi hasil observasi disajikan pada Tabel 5.1 berikut.

Tabel 5.1 Durasi dan Hambatan Utama Tugas Administrasi di

No	Jenis Tugas Administrasi	Durasi Rata-rata	Frekuensi	Hambatan Utama yang Teramati
1	Penyusunan RPPH Harian	40–55 menit	Setiap hari kerja	Diketik ulang manual atau disalin dari RPPH lama; format sering berantakan di Word.

No	Jenis Tugas Administrasi	Durasi Rata-rata	Frekuensi	Hambatan Utama yang Teramati
2	Pencatatan Presensi Siswa	5 menit	Setiap hari kerja	Ditulis manual di buku folio; risiko buku fisik rusak/hilang; belum ada rekap otomatis.
3	Rekapitulasi Presensi Siswa	30–60 menit (bulanan)	Mingguan & bulanan	Dihitung manual tanda per tanda dari buku hadir; memakan waktu dan rentan salah hitung.
4	Pencatatan Presensi GTK	— (tidak dilakukan)	Setiap hari kerja	Tidak ada buku hadir guru; ketiadaan rekam jejak kehadiran staf yang terstruktur.
5	Pembuatan Surat Resmi	25–40 menit	Insidental (2-4×/bulan)	Kop surat diketik ulang; nomor surat dicari manual; risiko nomor surat ganda.
6	Pencarian arsip dokumen (RPPH/surat lama)	10–30 menit	Insidental	Berkas tersebar tanpa indeks yang rapi di folder laptop dan map fisik.

(Sumber: Hasil observasi, 2026)

5.2.2 Wawancara

Wawancara semi-terstruktur dilaksanakan terhadap empat responden yang mencakup seluruh pengguna aktif di RA Al-Islam Gunungcupu. Profil responden disajikan pada Tabel 5.2 berikut.

Tabel 5.2 Profil Responden Wawancara

No	Kode	Jabatan	Masa Kerja	Perangkat Teknologi Harian
1	R-01	Kepala RA Al-Islam Gunungcupu	5 tahun	<i>Smartphone</i> (WhatsApp), laptop (sese kali)
2	R-02	Guru Kelompok A	10 tahun	<i>Smartphone</i> (WhatsApp)
3	R-03	Guru Kelompok B	20 tahun	<i>Smartphone</i> (WhatsApp), laptop (sese kali)
4	R-04	Guru Pendamping	7 tahun	<i>Smartphone</i> (WhatsApp)

(Sumber: Data penelitian, 2026)

Setiap wawancara berlangsung 30 hingga 45 menit, direkam dengan persetujuan responden, dan hasilnya ditranTugas Akhirkan untuk analisis. Wawancara diorganisasikan dalam lima tema utama.

5.2.2.1 Hambatan Administrasi

Seluruh responden mengidentifikasi penyusunan RPPH sebagai hambatan terbesar. Kepala RA mengungkapkan hambatan dari sudut pandang manajerial:

"Yang paling ribet itu kalau saya mau meriksa RPPH guru-guru. Harus minta diserahkan satu-satu atau difoto terus dikirim ke WA saya. Penumpukan di obrolan WA bikin bingung, kadang ada guru yang lupa kirim jadi tidak terperiksa."

— Kepala RA (R-01), Wawancara

Guru mengungkapkan hambatan dari sudut pandang operasional. R-02 menyatakan:

"Tiap hari harus nulis RPPH baru. Padahal temanya mah mirip-mirip, tapi tetep harus diketik ulang di laptop atau nulis di kertas. Kalau lagi repot ngurus anak-anak di kelas, ya paling saya nyalin dari RPPH kemarin terus diedit sedikit..."

— R-02, Guru Kelompok A, Wawancara

R-03 menambahkan permasalahan tentang surat:

"Kalau buat saya mah yang bikin pusing itu kalau disuruh buat surat. Ngetik kop suratnya di laptop sok suka berantakan, terus nyari nomor surat terakhir di tumpukan buku arsip juga lama. Pernah kita pakai nomor yang sama dua kali, baru ketahuan waktu ada pihak luar yang konfirmasi."

— R-03, Guru Kelompok B, Wawancara

5.2.2.2 Literasi Digital dan Kenyamanan Teknologi

Seluruh responden menyatakan memiliki keterbatasan kenyamanan menggunakan aplikasi komputer berbasis menu. R-04 menyatakan:

"Jujur mah saya jarang banget buka laptop. Takut salah pencet tombol, nanti malah rusak atau berkasnya hilang. Kalau di HP mah mendingan, soalnya udah biasa dipake tiap hari buat balasan WA"

— R-04, Tenaga Kependidikan, Wawancara

5.2.2.3 Preferensi Antarmuka

Ketika ditanya mengenai antarmuka yang diharapkan dari sistem baru tanpa diberi pilihan terlebih dahulu, seluruh responden secara spontan merujuk pada pengalaman menggunakan WhatsApp. R-02 menyatakan:

"Kalau bisa mah aplikasinya dibuat gampang saja di HP kayak WhatsApp. Soalnya kalau di WA kan kami sudah biasa tiap hari kirim pesan teks atau pesan suara. Daripada harus buka menu-menu di komputer yang bikin pusing dan takut salah isi form yang panjang, jauh lebih enak kalau tinggal kirim pesan biasa seperti lagi *chat* sehari-hari."

— R-02, Guru Kelompok A, Wawancara

Kepala RA menambahkan kebutuhan dari sisi manajemen:

"Saya mah pengennya pas ngebuka HP pagi-pagi, langsung keliatan rekapan serba praktis: siapa guru dan murid yang hadir hari ini, RPPH guru udah beres belum,

sama ada surat keluar apa saja. Jadi dalam satu layar langsung kelihatan semua tanpa perlu nyari-nyari."

— Kepala RA (R-01), Wawancara

5.2.2.4 Kebutuhan Fitur Spesifik

Terdapat perbedaan yang cukup mencolok antara kebutuhan Kepala RA dan guru. Kepala RA menekankan kebutuhan manajerial berupa rekap presensi terpusat, pengingat kelengkapan administrasi, dan akses cepat arsip surat. Guru menekankan kebutuhan operasional berupa RPPH otomatis, presensi yang sesederhana mungkin, dan surat tanpa mengetik ulang kop dan nomor.

Satu temuan penting yang tidak terantisipasi adalah permintaan agar sistem dapat "mengenal" identitas sekolah. R-01 menyatakan:

"Sistemnya harus otomatis hafal nama sekolah kita, alamat, dan nama Saya sebagai Kepala RA. Jangan sampai guru-guru harus ngetik ulang nama sekolah dari nol tiap kali bikin dokumen. Biar sistemnya aja yang udah otomatis nyimpen identitas sekolah kita."

— Kepala RA (R-01), Wawancara

Pernyataan ini menunjukkan bahwa sistem tidak cukup hanya menghasilkan dokumen, tetapi juga perlu memiliki sumber data internal yang mampu menyimpan informasi dasar lembaga dan pengguna. Namun pada tahap awal, temuan tersebut belum langsung ditetapkan sebagai modul tersendiri, melainkan dicatat sebagai masukan untuk dipertimbangkan dalam pembaruan *backlog* setelah pengguna mencoba increment sistem.

5.2.2.5 Ketersediaan Infrastruktur Teknologi

RA Al-Islam Gunungcupu memiliki jaringan Wi-Fi sekolah yang cukup stabil. Seluruh guru memiliki *smartphone* Android dengan sistem operasi minimal versi 7.0 dengan RAM minimal 2 GB, sesuai spesifikasi minimum yang ditetapkan pada Bab IV. Seluruh guru juga memiliki paket data seluler pribadi sebagai cadangan.

Tidak ada pengguna yang menggunakan iOS. Temuan ini mengonfirmasi bahwa sistem berbasis web melalui browser *smartphone* Android merupakan pilihan yang tepat dan terjangkau bagi seluruh pengguna.

Ringkasan temuan wawancara berdasarkan lima tema disajikan pada Tabel 5.3 berikut.

Tabel 5.3 Ringkasan Temuan Wawancara per Tema

Tema	Pertanyaan Inti	Temuan dari Kepala RA	Temuan dari Guru
Hambatan Administrasi	Tugas apa yang paling menyita waktu?	Pemantauan kelengkapan administrasi guru sulit; tidak ada sistem rekap terpusat	Penyusunan RPPH ±40–55 menit/hari; pembuatan surat mengharuskan mengetik ulang kop dan mencari nomor
Literasi Digital	Seberapa nyaman menggunakan aplikasi komputer?	Cukup nyaman untuk tugas dasar; menghindari aplikasi dengan navigasi menu berlapis	Lebih nyaman menggunakan <i>smartphone</i> ; enggan memakai laptop karena takut salah dan tidak mahir Word.
Preferensi Antarmuka	Antarmuka seperti apa yang diinginkan?	Tampilan ringkas yang menampilkan semua informasi penting sekaligus	Antarmuka mirip WhatsApp, ketik pesan langsung tanpa cari-cari menu
Kebutuhan Fitur Spesifik	Fitur apa yang paling diperlukan?	Rekap presensi terpusat; pengingat	RPPH otomatis; presensi cukup dengan ketik nama;

Tema	Pertanyaan Inti	Temuan dari Kepala RA	Temuan dari Guru
		kelengkapan; arsip surat terorganisir	surat langsung jadi tanpa ketik kop berulang
Ketersediaan Teknologi	Perangkat dan koneksi apa yang tersedia?	Wi-Fi sekolah tersedia; memiliki <i>smartphone</i> Android dan satu laptop di ruang kantor.	Seluruh guru memiliki <i>smartphone</i> ; semua menggunakan paket data sendiri sebagai cadangan

(Sumber: Hasil wawancara semi-terstruktur, 2026)

Tabel 5.3 menunjukkan pola konsisten: preferensi antarmuka berbasis percakapan teks muncul secara spontan dari seluruh responden tanpa sugesti, kebutuhan Kepala RA dan guru berbeda namun saling melengkapi; dan ketersediaan *smartphone* dengan WhatsApp menjadi aset yang dapat dimanfaatkan sebagai acuan desain antarmuka.

5.2.3 Studi Dokumentasi

Studi dokumentasi dilaksanakan terhadap dokumen administrasi yang digunakan RA Al-Islam Gunungcupu, mencakup tujuh jenis dokumen sebagaimana ditetapkan pada Tabel 4.3. Setiap dokumen dikaji dari aspek format, komponen, konvensi penulisan, dan potensi otomasi yang dapat dilakukan sistem.

5.2.3.1 Format RPPH Baku RA Al-Islam Gunungcupu

Studi dokumentasi terhadap 15 dokumen RPPH yang tersedia di arsip menghasilkan pemahaman mendetail mengenai format yang berlaku, Kepala RA pun meminta untuk merombak format penulisan dokumen RPPH supaya menjadi acuan sebagai format baku yang baru. Komponen RPPH baku lembaga disajikan pada Tabel 5.4 berikut.

Tabel 5.4 Komponen Format RPPH Baku RA Al-Islam Gunungcupu

No	Komponen RPPH	Deskripsi Isian	Temuan Studi Dokumentasi
1	Header Identitas & Karakteristik	Nama lembaga, NPSN, Semester/TA, Kelompok/Fase, Waktu, Tema/Topik, Fokus Nilai KBC (seperti Hubbullah & Hubbunnafs), Profil Lulusan, dan Karakteristik Peserta Didik.	Penulisan elemen dinamis (seperti tema, waktu, dan karakteristik peserta didik) diketik berulang setiap hari. Ditemukan beberapa inkonsistensi ejaan nama lembaga dan NPSN akibat ketiadaan templat yang mengunci data statis.
2	Desain & Capaian Pembelajaran	Capaian Pembelajaran (Nilai Agama & Budi Pekerti, Jati Diri, Dasar Literasi & STEAM), Model, Metode, Media Digital, dan Kemitraan Orang Tua.	Penulisan detail metode dan kemitraan orang tua memakan waktu jika diketik manual. Pada beberapa dokumen, bagian ini dibiarkan kosong atau disalin utuh (<i>copy-paste</i>) dari RPPH sebelumnya tanpa penyesuaian spesifik.
3	Alur Pembelajaran Harian: Pembukaan	Alokasi ±30 menit. Mencakup Penyambutan Penuh Cinta, Doa & Asmaul Husna, Apersepsi	Sebagian besar dokumen menggunakan urutan kegiatan pembukaan yang sangat baku dan identik, namun guru tetap mengetik urutan

No	Komponen RPPH	Deskripsi Isian	Temuan Studi Dokumentasi
		& Pemantik, serta <i>Ice Breaking</i> .	langkah tersebut dari awal akibat ketiadaan templat digital.
4	Alur Pembelajaran Harian: Inti	Alokasi ±90 menit. Dibagi menjadi 3 Sentra Aktivitas yang bergiliran, seperti Fisikmotorik, Konstruksi, dan Bermain Peran.	Merupakan bagian yang paling menyita waktu penulisan. Akibat keterbatasan waktu, sering ditemukan dokumen dengan kegiatan inti yang disalin persis dari hari sebelumnya dengan sangat sedikit variasi aktivitas.
5	Alur Pembelajaran Harian: Penutup	Alokasi ±30 menit. Mencakup Recalling & Show and Tell, Refleksi Perasaan, Pesan Cinta (KBC), dan Doa & Salam Penutup.	Kalimat yang digunakan cenderung repetitif dan monoton antarhari. Guru menghabiskan waktu mengetik ulang instruksi penutup yang sebenarnya bersifat relatif baku.
6	Lembar Kerja Anak (LKA) & Format Observasi	Rancangan tugas LKA konkret anak dan Rubrik Asesmen pengamatan guru (Spiritual, Sosio-Emosional, Kognitif, Kemandirian, Motorik)	Format tabel sering kali berantakan ketika dikerjakan di aplikasi pengolah kata. Ditemukan dokumen di mana bagian rubrik

No	Komponen RPPH	Deskripsi Isian	Temuan Studi Dokumentasi
		dengan skala BB, MB, BSH, BSB.	asesmen ini tidak diisi lengkap atau terlewat karena menyita waktu.
7	Tanda Tangan Pengesahan	Kolom tanda tangan Kepala RA dan Guru Kelas beserta keterangan tempat dan tanggal pengesahan.	Letak dan format kolom tanda tangan bervariasi antarguru. NIP sering tidak dicantumkan secara konsisten atau ditulis dengan singkatan yang berbeda-beda.

(Sumber: Hasil studi dokumentasi terhadap arsip RPPH, 2026)

Tabel 5.4 mengungkapkan bahwa komponen *header* identitas harus diisi ulang setiap hari dan memuat beberapa elemen data, baik yang bersifat statis (nama RA, nama guru, kelompok) maupun dinamis (hari/tanggal, tema, sub-tema). Temuan ini menjadi dasar kebutuhan agar sistem dapat mengurangi pengetikan berulang pada bagian identitas dan membantu pengguna menghasilkan RPPH sesuai format baku lembaga. Komponen "Kegiatan Inti" yang paling memakan waktu menjadi salah satu bagian yang ditargetkan untuk dibantu melalui generasi otomatis LLM berbasis konteks tema dan sub-tema.

5.2.3.2 Format dan Konvensi Surat Dinas Resmi

Studi dokumentasi terhadap 23 dokumen surat di arsip lembaga tahun 2025-2026 menghasilkan temuan mengenai format dan konvensi surat yang digunakan RA Al-Islam Gunungcupu. Komponen surat dinas yang dikaji meliputi kop surat, nomor surat, perihal, tujuan, isi, penutup, tanda tangan, dan jenis surat yang paling sering dibuat. Ringkasan komponen surat disajikan pada Tabel 5.5 berikut.

Tabel 5.5 Komponen Format Surat Dinas RA Al-Islam Gunungcupu

No	Komponen Surat	Format yang Berlaku	Temuan Studi Dokumentasi
1	Kop Surat	Logo, Nama RA, alamat lengkap, dipisahkan garis tebal.	Diketik ulang setiap membuat surat baru; inkonsistensi ejaan nama sekolah ditemukan pada beberapa dokumen
2	Nomor Surat	RA.043/[Jenis Surat]/[Nomor urut]/[Bulan Romawi]/[Tahun]; contoh: RA.04/SK/1/I/2025	Dicari dari arsip lama; ditemukan dua surat berbeda dengan nomor yang sama pada arsip tahun 2026
3	Perihal dan Tujuan	Satu baris perihal; nama/jabatan penerima	Format cukup konsisten
4	Pembuka dan Isi	Kalimat pembuka baku; paragraf isi sesuai keperluan	Diketik bebas tanpa template; kualitas bahasa tidak konsisten antarsurat
5	Penutup dan Tanda Tangan	Kalimat penutup; nama kepala RA, NIP/NIK, cap lembaga	Format baku; NIP kepala kadang tidak dicantumkan
6	Jenis Surat Paling Sering	Surat pemberitahuan, surat undangan, surat keterangan	Frekuensi: pemberitahuan ~2×/bulan, undangan ~1×/bulan, keterangan insidental

(Sumber: Hasil studi dokumentasi terhadap arsip surat, 2026)

Temuan kritis dari Tabel 5.5 adalah konfirmasi adanya insiden duplikasi nomor surat pada arsip, yaitu dua dokumen berbeda menggunakan nomor "RA.043/SK/12/III/2026". Temuan ini menunjukkan bahwa pembuatan surat tidak hanya membutuhkan bantuan penyusunan isi, tetapi juga konsistensi format, kop surat, dan penomoran. Pada tahap awal, temuan tersebut menjadi masukan desain untuk modul surat, sedangkan kebutuhan otomatisasi yang lebih mendalam dipertimbangkan melalui *review* pengguna pada iterasi berikutnya.

5.2.3.3 Format Presensi dan Data Peserta Didik

Format presensi yang berlaku mencatat empat status, Hadir, Izin, Sakit, dan Alfa. RA Al-Islam Gunungcupu memiliki dua kelompok belajar aktif dengan total 30 peserta didik, Kelompok A (usia 4–5 tahun) dengan 13 siswa, dan Kelompok B (usia 5–6 tahun) dengan 17 siswa. Dari data ini, tabel presensi pada basis data AISYA-RA dirancang untuk memuat empat status tersebut sesuai konvensi yang berlaku di lembaga.

5.2.3.4 Data Guru dan Tenaga Kependidikan

Data guru dan tenaga kependidikan diperoleh dari berkas administrasi kepegawaian. RA Al-Islam Gunungcupu memiliki empat tenaga aktif, satu Kepala RA merangkap guru, dua guru kelas, dan satu guru pembimbing. Data mencakup nama lengkap, NIP/NIK, jabatan, dan bidang studi. Data ini menjadi informasi penting yang dapat digunakan apabila sistem membutuhkan konteks identitas pengguna dalam proses administrasi.

5.2.3.5 Dokumen Pedoman dan Kalender Pendidikan

Selain dokumen administrasi harian, studi dokumentasi juga mengkaji pedoman administrasi RA, kalender pendidikan tahun ajaran berjalan, dan panduan kegiatan tema. Ketiga dokumen ini diidentifikasi sebagai kandidat utama basis pengetahuan (*knowledge base*) untuk fitur RAG pada Iterasi 3. Kalender pendidikan memuat informasi libur nasional, libur sekolah, dan jadwal kegiatan lembaga yang relevan untuk menjawab pertanyaan kontekstual guru melalui *Chatbot*.

5.2.3.6 Sintesis Temuan Studi Dokumentasi

Studi dokumentasi menghasilkan tiga kategori temuan. Pertama, temuan yang langsung membentuk format keluaran sistem, yaitu struktur tujuh komponen RPPH baku, format kop surat, dan empat kategori status presensi. Kedua, temuan yang memvalidasi kebutuhan otomasi, seperti pengetikan ulang kop surat, pencarian nomor surat secara manual, dan pengarsipan dokumen yang belum terstruktur. Ketiga, temuan yang belum langsung masuk sebagai fitur inti pada rencana awal, tetapi dicatat sebagai masukan penting untuk pembaruan *backlog* setelah sistem diuji oleh pengguna.

5.3 Tahap *Plan* (Perencanaan)

Berdasarkan temuan studi dan pengumpulan data, tahap *Plan* dilaksanakan untuk menyusun *product backlog* awal dan rencana iterasi yang menjadi panduan pengembangan AISYA-RA. Tahap *Plan* dilaksanakan satu kali sebelum seluruh iterasi dimulai dan menghasilkan artefak perencanaan yang terbuka untuk diperbarui berdasarkan umpan balik mitra pada setiap akhir iterasi.

5.3.1 Identifikasi Kebutuhan

Berdasarkan sintesis hasil observasi, wawancara, dan studi dokumentasi, kebutuhan sistem dikelompokkan menjadi kebutuhan fungsional dan kebutuhan nonfungsional.

Kebutuhan fungsional utama yang disepakati adalah:

1. KF-01: Sistem menyediakan autentikasi pengguna dengan hak akses berbasis peran (Admin, Kepala RA, Guru RA).
2. KF-02: Sistem menyediakan antarmuka percakapan CUI (*Chatbot*) berbasis bahasa alami sebagai pengendali utama fungsi administrasi.
3. KF-03: Sistem mengelola riwayat percakapan pengguna secara persisten.
4. KF-04: Sistem dapat menghasilkan draf dokumen RPPH otomatis berbasis 7 komponen kurikulum baku lembaga melalui percakapan chat.

5. KF-05: Sistem mencatat presensi siswa dan menghasilkan rekapitulasi kehadiran periodik otomatis melalui percakapan chat.
6. KF-06: Sistem menyusun surat dinas resmi dengan kop dan penomoran otomatis melalui percakapan chat.
7. KF-07: Sistem menyediakan lembar kerja visual (AISYA Canvas Modal) untuk melihat pratinjau (preview), menyunting teks secara langsung, menyimpan ke basis data, dan mencetak dokumen PDF.
8. KF-08: Sistem menyediakan manajemen dokumen untuk melihat arsip RPPH, rekap presensi siswa, dan surat keluar.
9. KF-09: Sistem menyediakan pengelolaan *Data Master* terpusat bagi Admin (Identitas Lembaga, Kelompok Belajar, Tahun Ajaran, Data Siswa, Data GTK).
10. KF-10: Sistem mengintegrasikan basis pengetahuan lokal (*RAG*) untuk menjawab pertanyaan seputar kurikulum dan SOP madrasah.
11. KF-11: Sistem menyediakan Dashboard ringkasan eksekutif bagi Kepala RA dan Admin.

Kebutuhan nonfungsional mencakup beberapa aspek, antara lain:

1. KNF-01 (*Usability*): Antarmuka responsif dan ramah pengguna *smartphone*, mengadopsi interaksi pesan instan untuk mengatasi keterbatasan literasi digital.
2. KNF-02 (*Performance*): Waktu inferensi percakapan AI rata-rata di bawah 5 detik dengan mekanisme *streaming* respons (SSE).
3. KNF-03 (*Security & Multi-Tenancy*): Seluruh akses data diisolasi menggunakan kunci lembaga (*ra_id*) dan diamankan dengan token JWT.

5.3.2 Penyusunan *Product Backlog*

Kebutuhan sistem disusun dan diprioritaskan ke dalam *Product Backlog* awal yang disajikan pada Tabel 5.6.

Tabel 5.6 Product Backlog AISYA-RA

Prioritas	Item Backlog	Uraian Kebutuhan	Rencana Pelaksanaan
Tinggi	Autentikasi dan Manajemen Pengguna	Sistem menyediakan fungsi <i>login, logout, remember me</i> , dan <i>reset password</i> dengan pembedaan peran (<i>Guru RA</i> dan <i>Kepala RA</i>).	Iterasi 1
Tinggi	<i>Chatbot</i> sebagai antarmuka utama	Sistem menyediakan ruang percakapan berbasis bahasa alami untuk interaksi awal dengan asisten kecerdasan buatan.	Iterasi 1
Tinggi	Manajemen Riwayat Percakapan	Sistem menyimpan sesi dan pesan obrolan secara persisten pada basis data.	Iterasi 1
Tinggi	Pembuatan RPPH Otomatis via <i>Chat</i>	Sistem menghasilkan draf RPPH berbasis kurikulum baku 7 komponen melalui instruksi bahasa alami.	Iterasi 2

Prioritas	Item Backlog	Uraian Kebutuhan	Rencana Pelaksanaan
Tinggi	Pencatatan & Rekap Presensi Siswa via Chat	Sistem mencatat absensi harian dan menghasilkan rekapitulasi periodik siswa via percakapan.	Iterasi 2
Tinggi	Pembuatan Surat Resmi via <i>Chat</i>	Sistem menghasilkan draf surat dinas lengkap dengan kop dan nomor surat atomik via percakapan.	Iterasi 2
Tinggi	<i>AISYA Canvas</i> Modal & Arsip Dokumen	Sistem menyediakan lembar kerja interaktif untuk <i>preview</i> , edit teks, simpan ke database, cetak PDF, dan akses arsip dokumen.	Iterasi 2
Tinggi	Pengelolaan <i>Data Master</i> Terpusat	Sistem menyediakan tampilan ringkasan informasi administrasi agar Kepala RA dapat memantau data penting secara cepat.	Iterasi 3
Sedang	<i>Dashboard</i> Ringkasan Eksekutif	Kepala RA memantau ringkasan informasi dan statistik administrasi penting madrasah.	Iterasi 3
Sedang	Basis Pengetahuan	Admin mengunggah dokumen panduan PDF dan sistem menjawab pertanyaan	Iterasi 3

Prioritas	Item Backlog	Uraian Kebutuhan	Rencana Pelaksanaan
	Dokumen Lokal (RAG)	regulasi/SOP berbasis pencarian semantik vektor.	
Sedang	Deployment Operasional Produksi	Sistem dideploy pada VPS <i>cloud</i> dengan konfigurasi kontainer Docker dan domain resmi madrasah.	Iterasi 3
Sedang	Penyempurnaan Berdasarkan <i>Review</i>	Sistem disesuaikan secara adaptif apabila ditemukan kebutuhan baru dari hasil uji coba pengguna pada iterasi sebelumnya	Iterasi 3
Sedang	Evaluasi Efisiensi & Penerimaan (System Usability Scale)	Mengukur efisiensi waktu kerja dan tingkat penerimaan pengguna akhir pasca-rilis operasional.	Evaluasi Akhir

(Sumber: Hasil perencanaan penelitian, 2026)

5.3.3 Use Case Diagram Rencana Awal

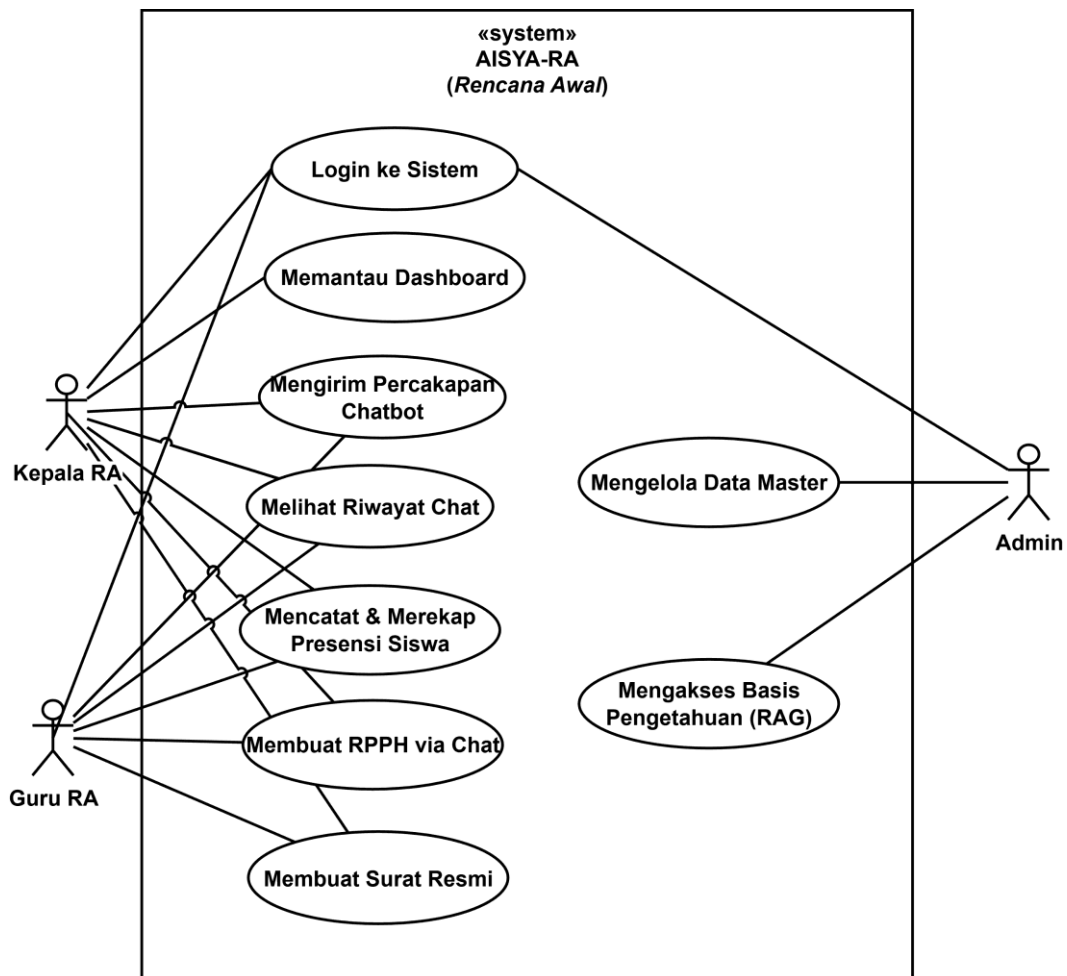
Berdasarkan *Product Backlog* pada Tabel 5.4, dirumuskan *Use Case Diagram* rencana awal pada tahap *Plan*. Diagram ini memetakan cakupan rencana fungsional awal pada level modul yang melibatkan tiga aktor (*Guru RA*, *Kepala RA*, dan *Admin*) dengan sembilan fungsi modul tingkat konseptual. Sesuai prinsip *Agile*, rancangan awal ini bersifat adaptif dan terbuka terhadap penyesuaian fungsional pada tahap *Design* di setiap iterasi maupun penyempurnaan berdasarkan hasil *Review* pengguna.

Diagram ini melibatkan dua aktor utama, Guru RA dan Kepala RA, yang berinteraksi dengan sembilan *Use Case* utama sebagaimana disajikan pada Tabel 5.7.

Tabel 5.7 Use Case Awal AISYA-RA Berdasarkan Product Backlog

<i>Use Case</i>	Deskripsi
<i>Login ke Sistem</i>	Autentikasi pengguna dengan pembedaan hak akses Guru RA, Kepala RA, dan Admin.
<i>Mengirim Percakapan Chatbot</i>	Menjalankan fungsi administrasi melalui instruksi bahasa alami (khusus Guru RA dan Kepala RA).
<i>Melihat Riwayat Chat</i>	Mengakses dan menelusuri riwayat percakapan yang tersimpan (khusus Guru RA dan Kepala RA).
<i>Membuat RPPH Otomatis</i>	Menyusun draf RPPH melalui instruksi bahasa alami sesuai format baku lembaga (khusus Guru RA dan Kepala RA).
<i>Mencatat & Merekap Presensi Siswa</i>	Mencatat kehadiran siswa dan menghasilkan rekapitulasi periodik secara otomatis (khusus Guru RA dan Kepala RA).
<i>Membuat Surat Resmi</i>	Menyusun, menyimpan, dan menelusuri surat resmi lembaga (khusus Guru RA dan Kepala RA).
<i>Mengelola Data Master</i>	Admin mengelola data dasar lembaga, akun GTK, dan data induk siswa.
<i>Memantau Dashboard</i>	Kepala RA memantau ringkasan informasi administrasi lembaga.
<i>Mengakses Basis Pengetahuan (RAG)</i>	Admin mengunggah dan mengelola dokumen lokal sekolah sebagai referensi pencarian konteks AI.

(Sumber: Hasil identifikasi kebutuhan, 2026)



Gambar 5.1 Use Case Diagram Rencana Awal

(Sumber: Hasil rencana Use Case Diagram, 2026)

Berdasarkan pemodelan *Use Case* awal di atas, ditetapkan pemisahan tugas (*Separation of Duties*) yang tegas. Aktor Kepala RA dan Guru RA difokuskan sebagai pengguna akhir untuk operasional harian melalui interaksi percakapan, sementara aktor Admin dialokasikan khusus untuk menangani inisialisasi basis data dan manajemen pengetahuan di balik layar. Cakupan tersebut kemudian direalisasikan secara bertahap, dimulai dari fondasi sistem pada Iterasi 1, dilanjutkan modul administrasi inti pada Iterasi 2, hingga penguatan sistem pada Iterasi 3.

5.3.4 Rencana Iterasi

Pengembangan sistem dibagi ke dalam tiga iterasi utama berbasis *Agile* dengan *timebox* terukur, yang disajikan pada Tabel 5.8 berikut.

Tabel 5.8 Rencana Iterasi Pengembangan AISYA-RA

Iterasi	Fokus	Target Luaran	Target Waktu
Iterasi 1	Fondasi sistem dan <i>Chatbot</i> dasar	Modul autentikasi pengguna (<i>Login, Logout, Reset Password, Remember Me</i>), antarmuka ruang <i>chat</i> , integrasi LLM Gemini, dan penyimpanan riwayat obrolan pada basis data lokal/pengembangan.	2 Minggu
Iterasi 2	Modul Administrasi Inti dan <i>Canvas Modal</i>	Kemampuan otomatisasi dokumen (RPPH baku 7 komponen, pencatatan & rekap presensi siswa, pembuatan surat keluar berpenomoran otomatis), lembar kerja <i>Canvas Modal</i> (<i>preview, edit, simpan DB, cetak PDF</i>), dan modul arsip dokumen.	2,5 Minggu
Iterasi 3	Penguatan Sistem: <i>Data Master, RAG, Presensi GTK, Dashboard &</i>	Pangkalan <i>Data Master</i> terpusat bagi Admin, pencatatan presensi GTK via <i>chat</i> , pencarian semantik dokumen kurikulum/SOP (RAG	2,5 Minggu

Iterasi	Fokus	Target Luaran	Target Waktu
	VPS Deployment	<i>pgvector</i> 768-dimensi), visualisasi <i>Dashboard</i> eksekutif, serta <i>deployment</i> penuh ke Cloud VPS DigitalOcean dengan domain resmi.	
Evaluasi Akhir	Pengujian Operasional & Pengukuran SUS	Pengujian efisiensi waktu tugas administrasi sebelum vs sesudah penerapan sistem, serta pengukuran skor SUS (<i>System Usability Scale</i>).	1 Minggu

(Sumber: Hasil perencanaan penelitian, 2026)

5.4 Iterasi 1: Fondasi Sistem dan *Chatbot* Dasar

Iterasi 1 dilaksanakan selama dua minggu sesuai *timebox* yang ditetapkan. Tujuan utama iterasi ini adalah membangun fondasi teknis sistem AISYA-RA yang memungkinkan pengguna mengakses sistem melalui antarmuka berbasis percakapan (*Conversational User Interface*), mencakup autentikasi berbasis peran, komunikasi *frontend-backend*, integrasi awal Gemini API, dan penyimpanan riwayat percakapan secara persisten.

Item backlog yang dikerjakan pada Iterasi 1 meliputi:

1. Autentikasi dan manajemen pengguna (*Login, Logout, Remember Me, Reset Password*).
2. *Chatbot* sebagai antarmuka kendali sistem dasar.
3. Manajemen riwayat percakapan (*Chat History*).

Definition of Done (DoD) Iterasi 1 dinyatakan selesai ketika kode program berjalan tanpa kesalahan pada lingkungan lokal, antarmuka pengguna sesuai dengan rancangan *wireframe*, dan seluruh skenario pengujian *Black Box Testing* untuk fungsi fondasi dinyatakan lulus (*Passed*).

5.4.1 *Design* Iterasi 1

Tahap *Design* Iterasi 1 menghasilkan enam artefak perancangan sistem menggunakan standar *Unified Modeling Language (UML)* dan pemodelan data sebagai pedoman teknis implementasi:

5.4.1.1 *Use Case Diagram*

Use Case Diagram Iterasi 1 mendefinisikan ruang lingkup fungsional sistem pada tahap fondasi. Terdapat dua aktor utama, yaitu Guru RA dan Kepala RA, yang memiliki hak akses setara pada iterasi ini. Kedua aktor berinteraksi dengan sembilan Use Case utama sebagaimana disajikan pada Tabel 5.9 berikut.

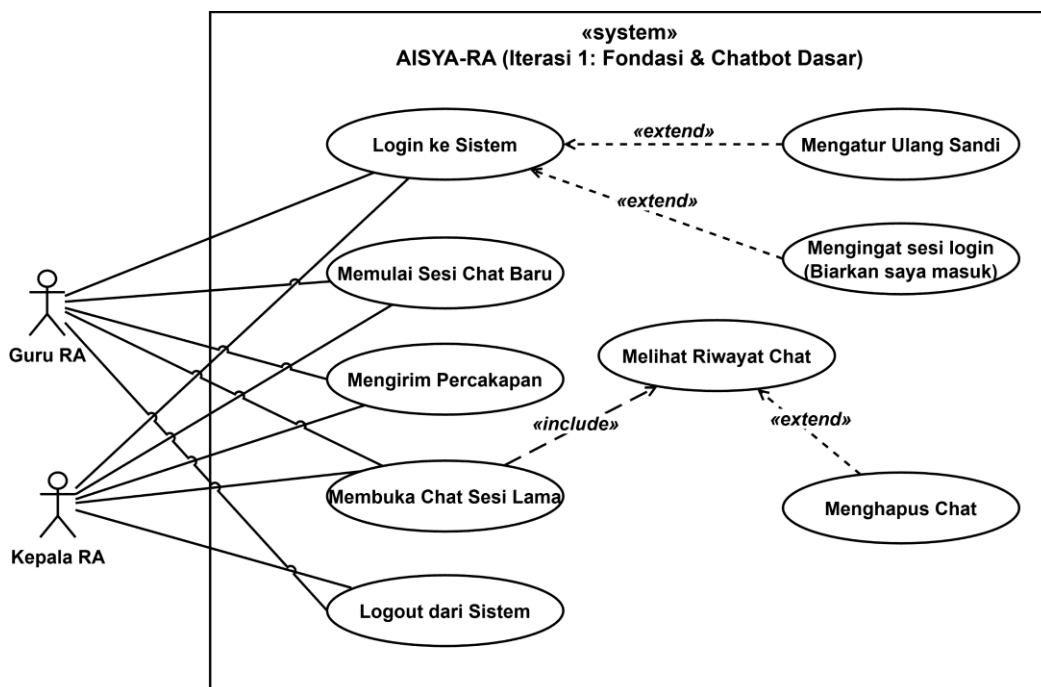
Tabel 5.9 Use Case Diagram Iterasi 1

<i>Use Case</i>	Deskripsi
<i>Login ke Sistem</i>	Autentikasi pengguna dengan email dan kata sandi
<i>Logout dari Sistem</i>	Mengakhiri sesi autentikasi pengguna dan menghapus token
<i>Memulai Sesi Chat Baru</i>	Membuat sesi ruang percakapan baru dengan <i>Chatbot</i>
<i>Mengirim Percakapan</i>	Mengirim pesan dan menerima respons AI di ruang <i>chat</i>
<i>Membuka Chat Sesi Lama</i>	Melanjutkan sesi percakapan yang tersimpan sebelumnya
<i>Melihat Riwayat Chat</i>	Melihat daftar riwayat percakapan (« <i>include</i> » dari <i>Membuka Chat Sesi Lama</i>)

<i>Use Case</i>	<i>Deskripsi</i>
Menghapus Chat	Menghapus sesi percakapan tertentu (« <i>extend</i> » ke Melihat Riwayat Chat)
Mengatur Ulang Sandi	Memperbarui kata sandi akun pengguna (« <i>extend</i> » ke <i>Login ke Sistem</i>)
Mengingat sesi <i>Login</i> (Biarkan saya masuk)	Menyimpan sesi autentikasi pengguna pada perangkat (« <i>extend</i> » ke <i>Login ke Sistem</i>)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

Dibandingkan dengan *Use Case Diagram* tahap *Plan* yang masih berada pada level modul umum, *Use Case Diagram* Iterasi 1 memecah fungsi "*Login ke Sistem*" menjadi beberapa sub-fungsi granular teknis (*Mengatur Ulang Sandi* dan *Mengingat Sesi Login*). Demikian pula fungsi "*Melihat Riwayat Chat*" di-include oleh "*Membuka Chat Sesi Lama*" dan di-extend oleh "*Menghapus Chat*".



Gambar 5.2 Use Case Diagram Iterasi 1

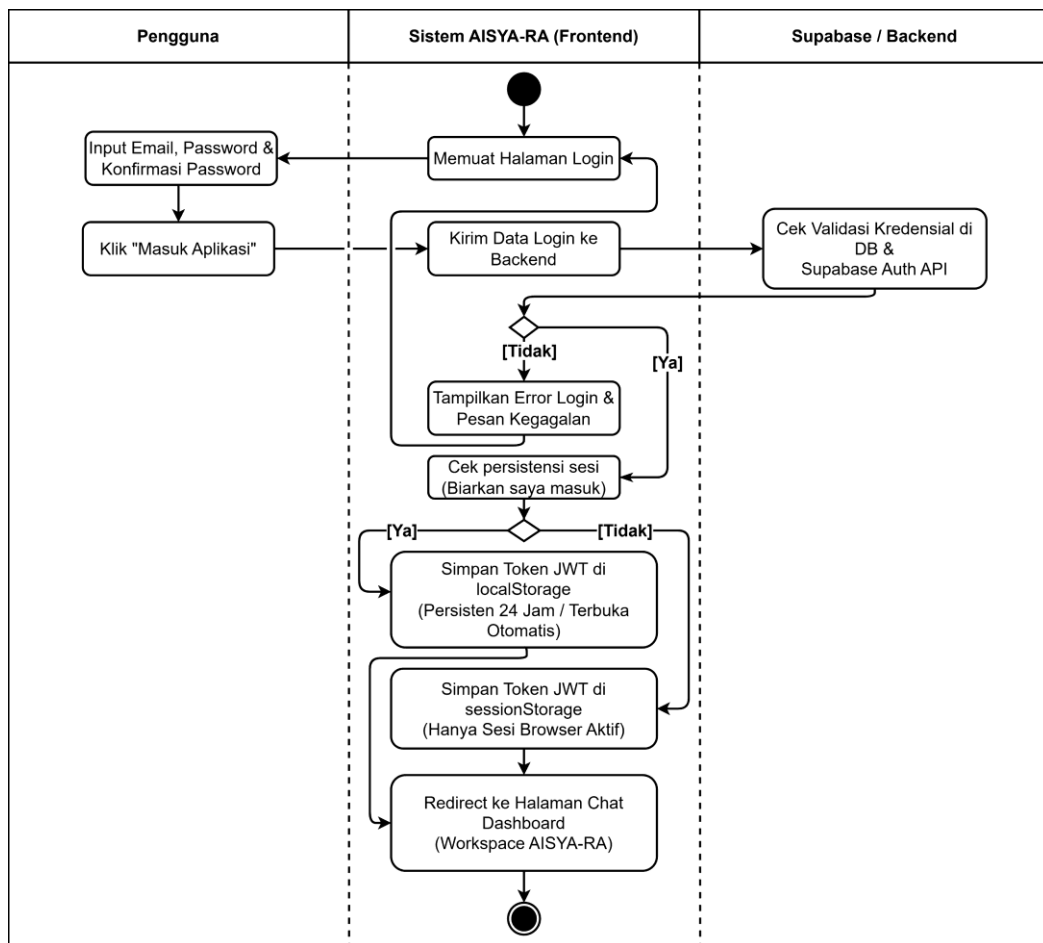
(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2 *Activity Diagram*

Pemodelan *Activity Diagram* pada tahap *Design Iterasi 1* bertujuan untuk memetakan alur kerja operasional, pertukaran data, serta cabang keputusan pada fungsionalitas fondasi sistem. Fungsionalitas tersebut meliputi modul autentikasi pengguna (keamanan akses), manajemen sesi peramban (*web browser*), serta layanan percakapan dasar berbasis *Large Language Model (LLM)*. Pemodelan ini disusun dalam enam diagram aktivitas utama yang masing-masing membagi peran aktivitas ke dalam jalur swimlanes aktor dan komponen sistem.

5.4.1.2.1 *Activity Diagram Autentikasi (Login)*

Activity Diagram Autentikasi (Login) menggambarkan alur masuk pengguna dengan membagi tanggung jawab ke dalam tiga *swimlanes*, Pengguna, Frontend (*Login.tsx*, *AuthContext.tsx*), dan Backend/Supabase (*authRouter*, Supabase Auth API). Setelah pengguna memasukkan email dan kata sandi serta menekan tombol masuk, sistem memvalidasi kredensial. Jika valid, backend menerbitkan token JWT. Jika pengguna mencentang opsi "Biarkan saya tetap masuk", token disimpan pada *localStorage* (persisten 24 jam), jika tidak dicentang, token disimpan pada *sessionStorage* (hanya selama tab peramban aktif). Pengguna kemudian dialihkan secara otomatis ke ruang *chat*.

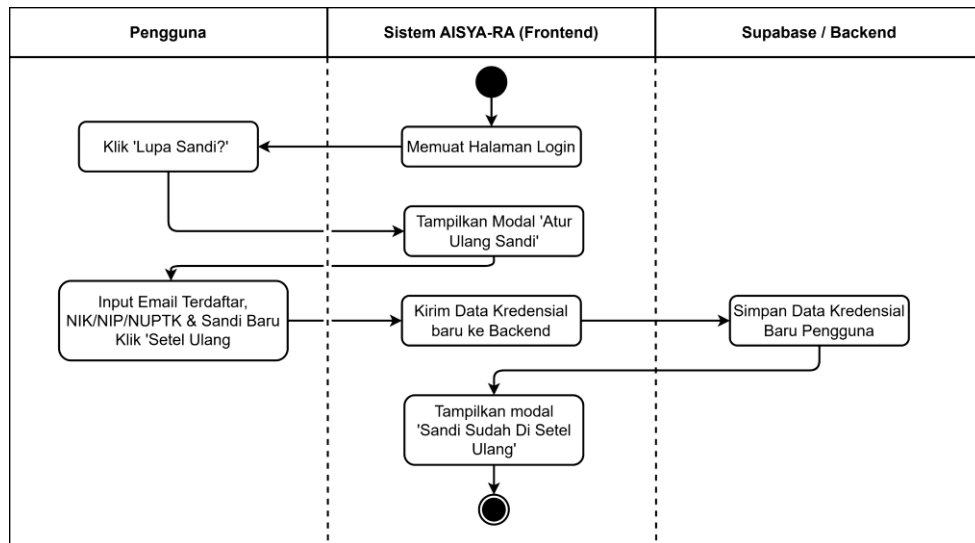


Gambar 5.3 Activity Diagram Iterasi 1 - Autentikasi (Login)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2.2 Activity Diagram Autentikasi (Lupa Sandi / Reset Sandi)

Activity Diagram Lupa Sandi Menggambarkan alur reset kata sandi langsung melalui modal verifikasi identitas NIK/NIP dan email tanpa tautan email keluar, memudahkan pengguna yang memiliki keterbatasan akses email madrasah.

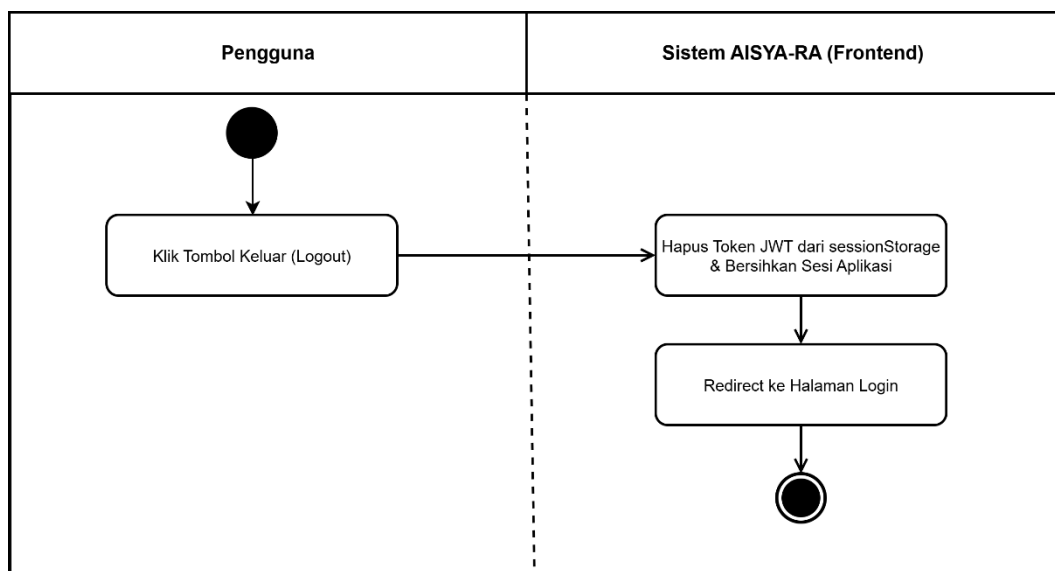


Gambar 5.4 Activity Diagram Iterasi 1 - Autentikasi (Lupa Sandi)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2.3 Activity Diagram Autentikasi (Logout)

Activity Diagram Logout menggambarkan alur penghentian sesi di mana token JWT dihapus dari memori peramban dan pengguna dikembalikan ke halaman Login.

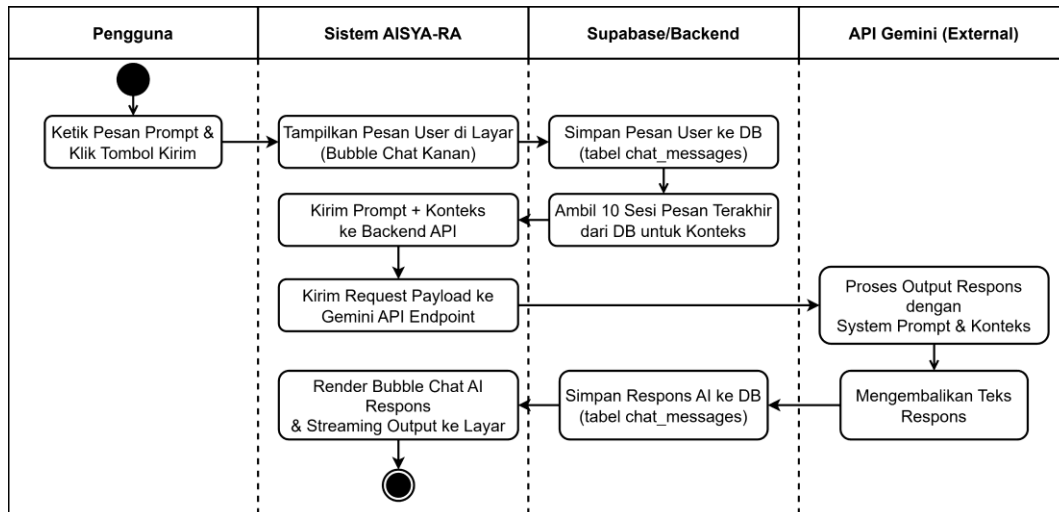


Gambar 5.5 Activity Diagram Iterasi 1 – Autentikasi (Logout)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2.4 Activity Diagram Interaksi Chatbot (Percakapan Dasar)

Activity Diagram Interaksi Chatbot Menggambarkan alur pertukaran pesan di mana teks masukan pengguna disimpan ke basis data (*role: user*), backend mengambil 10 riwayat pesan terakhir sebagai konteks percakapan, meneruskannya ke Gemini API, dan mengalirkan teks respons asisten kembali ke layar pengguna (*role: model*).

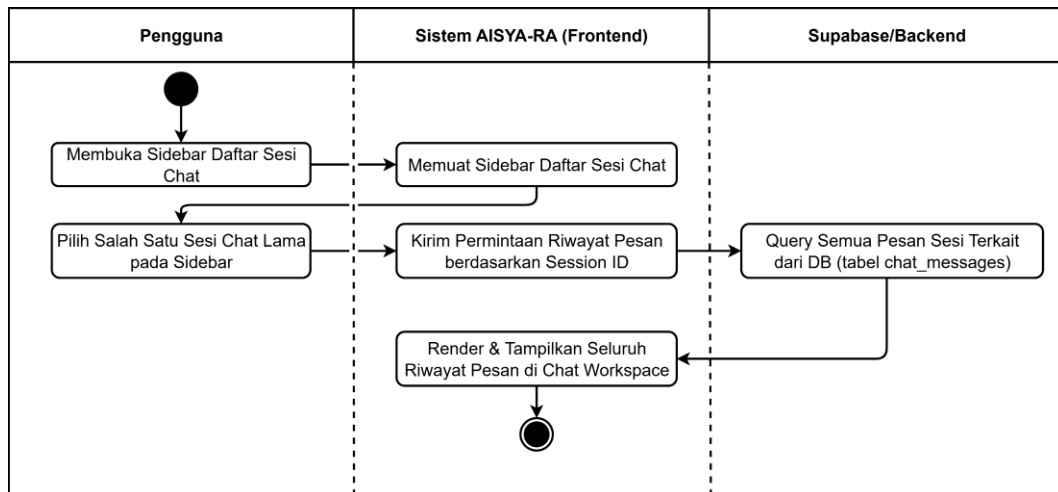


Gambar 5.6 Activity Diagram Iterasi 1 - Interaksi Chatbot (Percakapan Dasar)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2.5 Activity Diagram Riwayat Chatbot (Membuka Sesi Percakapan)

Activity Diagram Riwayat Chatbot menggambarkan pemuatan daftar sesi obrolan dari tabel *chat_sessions* ke panel bilah samping (*sidebar*), serta pemanggilan seluruh pesan terkait ketika pengguna memilih salah satu sesi.

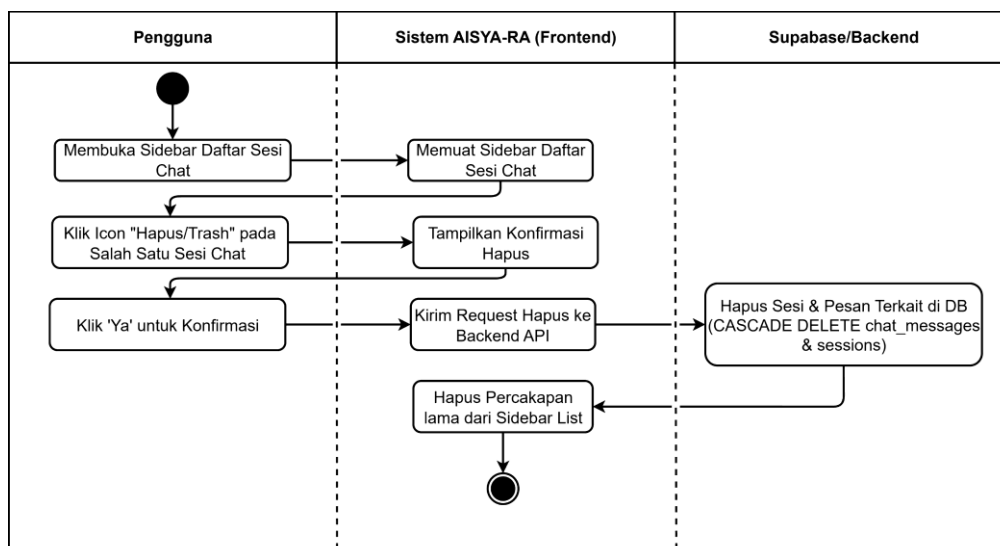


Gambar 5.7 Activity Diagram Iterasi 1 - Riwayat Chat (Membuka Sesi Percakapan)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.2.6 Activity Diagram Hapus Riwayat Chatbot

Activity Diagram Hapus Riwayat Chatbot menggambarkan konfirmasi dialog penghapusan sesi yang mengeksekusi perintah hapus beruntun (CASCADE delete) pada basis data.



Gambar 5.8 Activity Diagram Iterasi 1 - Hapus Riwayat Chatbot

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.3 Sequence Diagram

Sequence Diagram Iterasi 1 memodelkan urutan pesan antar objek pada saat *runtime*.

5.4.1.3.1 Sequence Diagram Autentikasi & Reset Sandi

Sequence Diagram ini melibatkan lima objek/komponen utama yang berinteraksi dalam proses otentikasi: **Pengguna** (*Kepala RA & Guru RA*), antarmuka **Login.tsx** (*Frontend View*), **AuthContext.tsx** (*Frontend State*), **authRouter / deps.py** (*FastAPI Backend Router*), dan **Supabase Auth & DB** (*Database*).

Alur eksekusi *login* terdiri dari tahapan linier berikut:

1. **Pengguna** mengisi email, kata sandi, mencentang opsi "Biarkan saya tetap masuk" (*rememberMe*), dan menekan tombol "Masuk" pada antarmuka **Login.tsx**.
2. **Login.tsx** memanggil fungsi *Login*(email, password, rememberMe) pada **AuthContext.tsx**.
3. **AuthContext.tsx** mengirimkan panggilan HTTP POST */api/auth/Login* yang membawa kredensial pengguna ke *backend deps.py / authRouter*.
4. *Backend* memvalidasi kredensial pengguna dengan mengeksekusi metode *signInWithPassword()* ke **Supabase Auth & DB**.
5. **Supabase Auth & DB** mengonfirmasi kredensial dan mengembalikan profil pengguna beserta peran (*role*) dan token JWT.
6. *Backend* mengembalikan respons HTTP 200 OK yang berisi token JWT dan data profil pengguna ke **AuthContext.tsx**.
7. **AuthContext.tsx** menyimpan token JWT secara adaptif ke *localStorage* (jika *rememberMe* = *true*) atau *sessionStorage* (jika *rememberMe* = *false*).

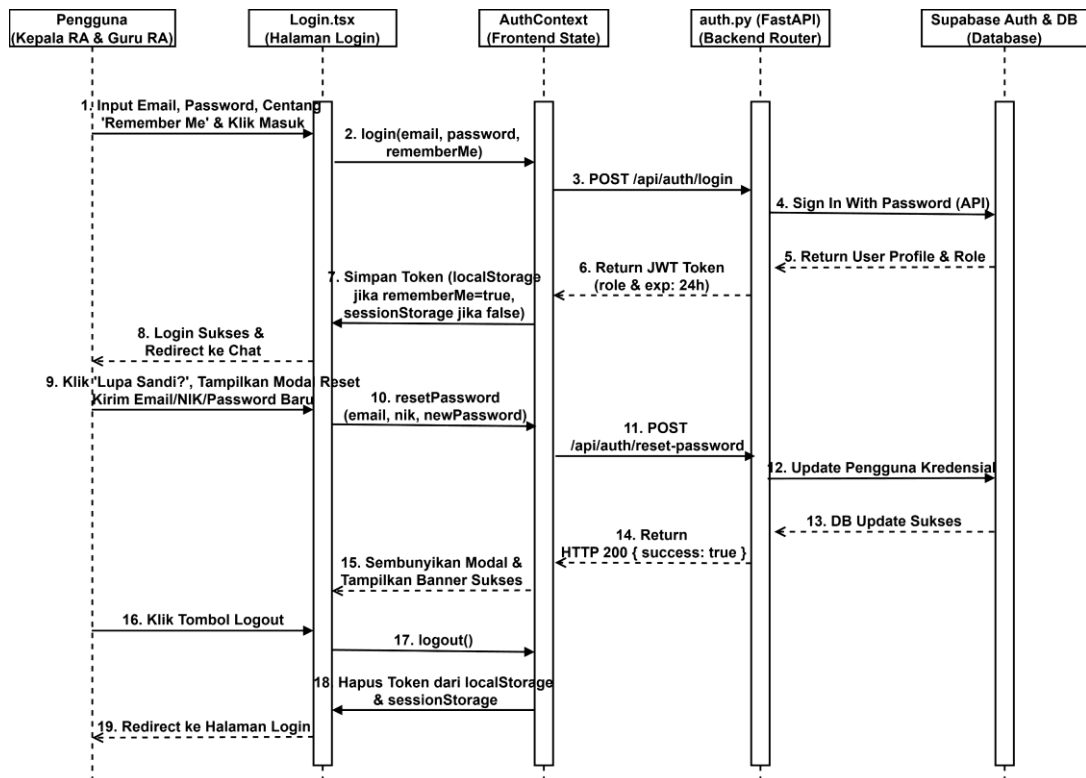
8. **Login.tsx** mengalihkan **Pengguna** secara otomatis ke antarmuka percakapan *Chatbot (/chat)*.

Alur eksekusi reset sandi terdiri dari tahapan linier berikut:

1. **Pengguna** mengklik tautan "Lupa Sandi?" dan mengisi formulir modal reset (email, NIK/NIP, dan kata sandi baru) pada **Login.tsx**.
2. **Login.tsx** memanggil fungsi `resetPassword(email, nik, newPassword)` pada **AuthContext.tsx**.
3. **AuthContext.tsx** mengirimkan panggilan HTTP POST `/api/auth/reset-password` ke *backend*.
4. *Backend* memverifikasi kesesuaian identitas pada tabel *pengguna* dan memperbarui kata sandi pengguna di **Supabase Auth & DB**.
5. **Supabase Auth & DB** mengembalikan konfirmasi keberhasilan pembaruan kata sandi ke *backend*.
6. *Backend* mengembalikan respons HTTP 200 OK `{success: true}` ke **AuthContext.tsx**.
7. **AuthContext.tsx** memerintahkan **Login.tsx** untuk menutup modal reset dan menampilkan notifikasi sukses.

Alur eksekusi *logout* terdiri dari tahapan linier berikut:

1. **Pengguna** menekan tombol "Logout" pada antarmuka aplikasi.
2. Komponen memanggil fungsi `Logout()` pada **AuthContext.tsx** yang menghapus token dari memori penyimpanan peramban (*localStorage/sessionStorage*) dan mengalihkan **Pengguna** kembali ke halaman **Login.tsx**.



Gambar 5.9 Sequence Diagram Iterasi 1 - Autentikasi & Reset Sandi

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

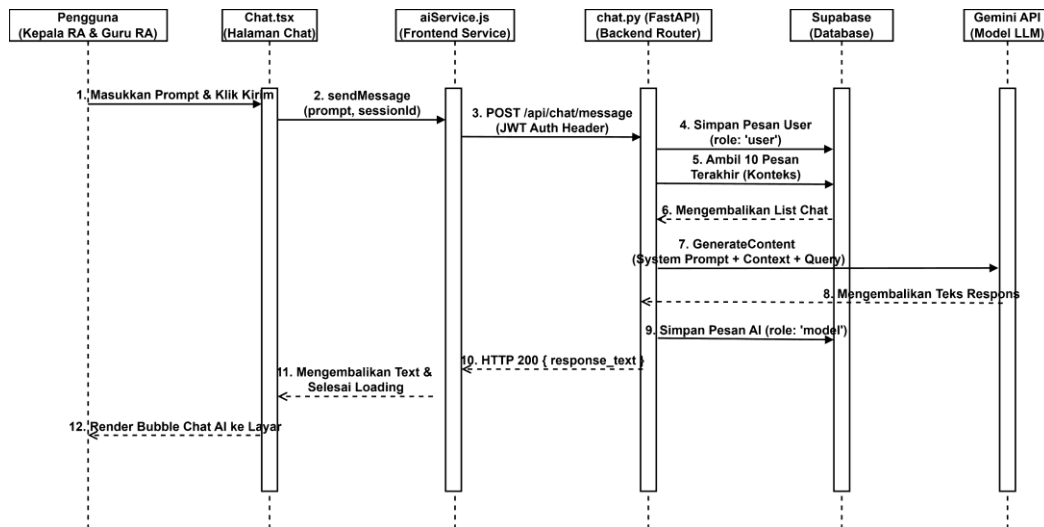
5.4.1.3.2 Interaksi Percakapan di Chatbot (Gemini AI)

Sequence Diagram ini memodelkan interaksi percakapan dinamis yang melibatkan **Pengguna**, antarmuka **Chat.tsx**, layanan klien **aiService.ts**, pengontrol backend **chat.py**, basis data **Supabase**, dan model kecerdasan buatan **Gemini API**.

Alur pertukaran pesan percakapan terdiri dari tahapan linier berikut:

1. **Pengguna** memasukkan teks instruksi (*prompt*) pada kotak pesan dan menekan tombol kirim di **Chat.tsx**.
2. **Chat.tsx** memanggil fungsi `sendMessage(prompt, sessionId)` pada modul **aiService.ts**.

3. **aiService.ts** mengirimkan permintaan HTTP POST `/api/chat/message` (membawa header otorisasi JWT, *prompt*, dan *sessionId*) ke backend **chat.py**.
4. Backend **chat.py** mengeksekusi kueri INSERT untuk menyimpan pesan pengguna (*role: user*) ke tabel *chat_messages* di **Supabase**.
5. Backend **chat.py** mengeksekusi kueri SELECT ke **Supabase** untuk mengambil 10 riwayat pesan percakapan terakhir sebagai konteks dialog.
6. **Supabase** mengembalikan riwayat pesan percakapan ke backend **chat.py**.
7. Backend **chat.py** menyusun *System Prompt* beserta konteks riwayat dan memanggil metode `generateContent` ke **Gemini API**.
8. **Gemini API** memproses instruksi dan mengembalikan teks respons hasil inferensi kecerdasan buatan ke **chat.py**.
9. Backend **chat.py** mengeksekusi kueri INSERT untuk menyimpan teks respons AI (*role: model*) ke tabel *chat_messages* di **Supabase**.
10. Backend **chat.py** mengembalikan respons HTTP 200 OK { `response_text` } ke **aiService.ts**.
11. **aiService.ts** meneruskan teks respons AI ke **Chat.tsx**.
12. **Chat.tsx** merender balasan asisten AI sebagai gelembung percakapan (*chat bubble*) berbasis komponen *ReactMarkdown* pada layar **Pengguna**.



Gambar 5.10 Sequence Diagram Iterasi 1 - Interaksi Percakapan di Chatbot (Gemini AI)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.3.3 Pengelolaan Riwayat Percakapan

Sequence Diagram ini memodelkan alur pemuatan daftar sesi obrolan, penelusuran pesan lama, dan penghapusan sesi percakapan secara permanen.

Alur pemuatan riwayat dan pembukaan sesi lama terdiri dari tahapan linier berikut:

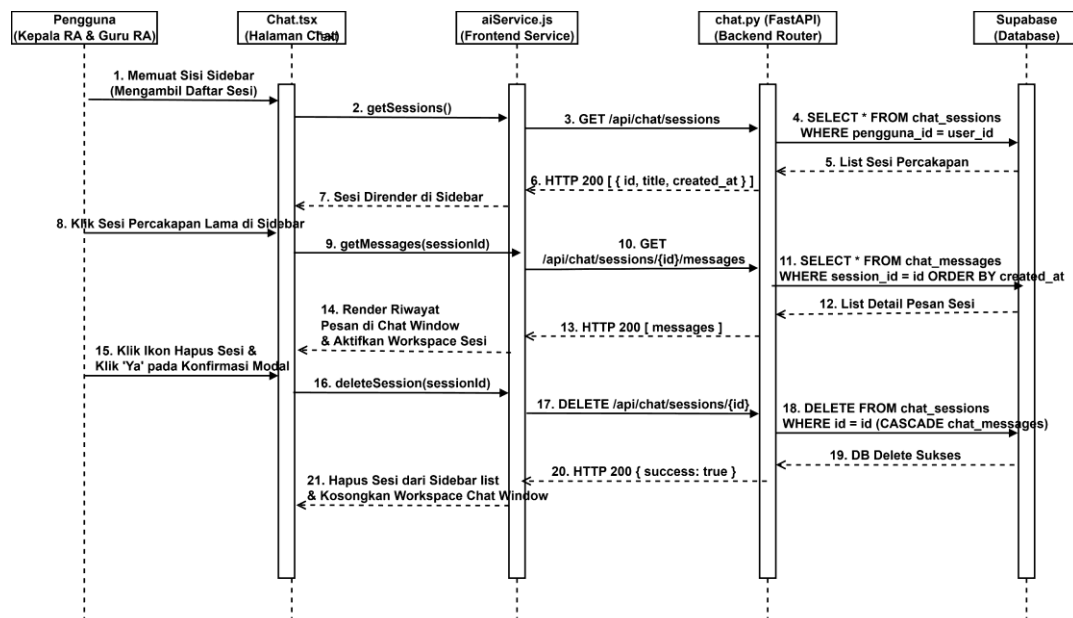
1. Saat halaman dimuat, **Chat.tsx** memanggil fungsi `getSessions()` pada **aiService.ts**.
2. **aiService.ts** mengirimkan permintaan HTTP GET `/api/chat/sessions` ke backend **chat.py**.
3. Backend **chat.py** mengeksekusi kueri `SELECT * FROM chat_sessions WHERE pengguna_id = user_id ORDER BY updated_at DESC` ke **Supabase**.
4. **Supabase** mengembalikan daftar rekam sesi percakapan milik pengguna.
5. Backend **chat.py** mengembalikan respons HTTP 200 OK berisi daftar sesi ke **aiService.ts**.

6. **aiService.ts** meneruskan daftar sesi ke **Chat.tsx** untuk dirender pada panel bilah samping (*sidebar*).
7. **Pengguna** memilih dan mengklik salah satu sesi percakapan lama pada bilah samping.
8. **Chat.tsx** memanggil fungsi `getMessages(sessionId)` pada **aiService.ts**.
9. **aiService.ts** mengirimkan panggilan HTTP GET `/api/chat/sessions/{id}/messages` ke backend **chat.py**.
10. Backend **chat.py** mengeksekusi kueri `SELECT * FROM chat_messages WHERE session_id = id ORDER BY created_at ASC` ke **Supabase**.
11. **Supabase** mengembalikan seluruh rekam pesan percakapan dalam sesi tersebut.
12. Backend **chat.py** mengembalikan respons HTTP 200 OK berisi daftar pesan ke **aiService.ts**.
13. **aiService.ts** meneruskan pesan ke **Chat.tsx**.
14. **Chat.tsx** merender seluruh pesan pada jendela obrolan utama dan mengaktifkan konteks percakapan sesi terpilih.

Alur penghapusan riwayat percakapan terdiri dari tahapan linier berikut:

1. **Pengguna** mengklik ikon hapus pada salah satu sesi di bilah samping dan mengonfirmasi dialog konfirmasi pada **Chat.tsx**.
2. **Chat.tsx** memanggil fungsi `deleteSession(sessionId)` pada **aiService.ts**.
3. **aiService.ts** mengirimkan permintaan HTTP DELETE `/api/chat/sessions/{id}` ke backend **chat.py**.
4. Backend **chat.py** mengeksekusi kueri `DELETE FROM chat_sessions WHERE id = id` ke **Supabase**.

5. **Supabase** menghapus rekam sesi dan secara otomatis menghapus seluruh pesan terkait melalui batasan *ON DELETE CASCADE*.
6. Backend **chat.py** mengembalikan respons HTTP 200 OK { success: true } ke **aiService.ts**.
7. **aiService.ts** meneruskan konfirmasi keberhasilan ke **Chat.tsx**.
8. **Chat.tsx** memperbarui antarmuka dengan menghapus sesi dari bilah samping dan mengosongkan jendela percakapan.



Gambar 5.11 Sequence Diagram Iterasi 1 - Pengelolaan Riwayat Percakapan
(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.4 Class Diagram

Class Diagram Iterasi 1 memodelkan struktur logika program dan representasi entitas data tahap fondasi sistem AISYA-RA. Arsitektur sistem diorganisasikan ke dalam tiga lapisan (*multi-tier architecture*), yaitu:

1. **Lapisan Antarmuka Pengguna (User Interface Layer)**: Terdiri dari komponen *React Login (Login.tsx)* yang menangani antarmuka formulir

otentikasi dan modal pemulihan kata sandi, serta komponen *Chat* (*Chat.tsx*) yang merender jendela percakapan, gelembung pesan (*chat bubbles*), daftar sesi percakapan pada bilah samping (*sidebar*), dan kontrol manajemen sesi.

2. Lapisan Logika dan Pengendali (*Logic & Controller Layer*):

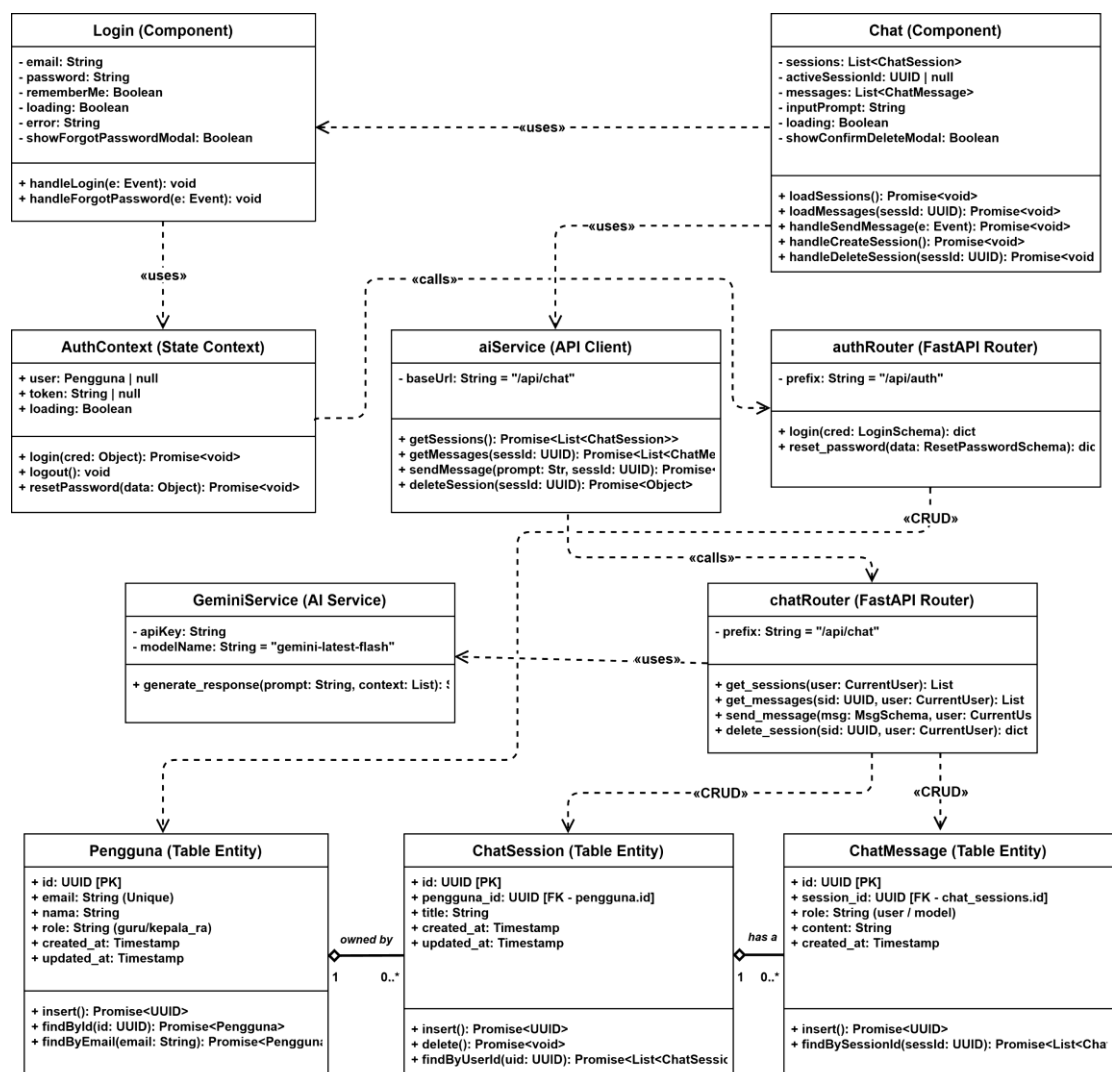
- a. *AuthContext* (*AuthContext.tsx*): Pengelola status (*state*) sesi autentikasi pengguna secara global di sisi peramban menggunakan React Context API.
- b. *aiService* (*aiService.ts*): Klien API HTTP sisi *frontend* untuk memanggil *endpoint* layanan percakapan backend.
- c. *authRouter* (*app/deps.py* dan *app/routers/pengguna.py*): Modul pengendali backend FastAPI yang bertugas memvalidasi otentikasi JWT (*JSON Web Token*) dan mengeksekusi fungsi Supabase Auth API.
- d. *chatRouter* (*app/routers/chat.py*): Modul pengendali backend FastAPI yang mengelola perutean pesan, memuat riwayat obrolan, dan mengorkestrasi inferensi kecerdasan buatan.
- e. *GeminiService*: Lapisan integrasi layanan eksternal ke Google Gemini API untuk membangkitkan teks respons percakapan secara cerdas.

3. Lapisan Entitas Data (*Data Entity Layer*): Terdiri dari tiga entitas tabel fondasi pada basis data PostgreSQL Supabase:

- a. *Pengguna*: Menyimpan data akun pengguna (*id, email, nama, role, created_at, updated_at*).
- b. *ChatSession*: Menyimpan metadata ruang percakapan (*id, pengguna_id, title, created_at, updated_at*).

- c. *ChatMessage*: Menyimpan rekam jejak pesan interaksi (*id*, *session_id*, *role*, *content*, *created_at*).

Hubungan antar entitas data memodelkan relasi *One-to-Many* (1 : 0..*) dari *Pengguna* ke *ChatSession* (*owned by*), serta dari *ChatSession* ke *ChatMessage* (*has a*) dengan batasan integritas referensial *ON DELETE CASCADE*.



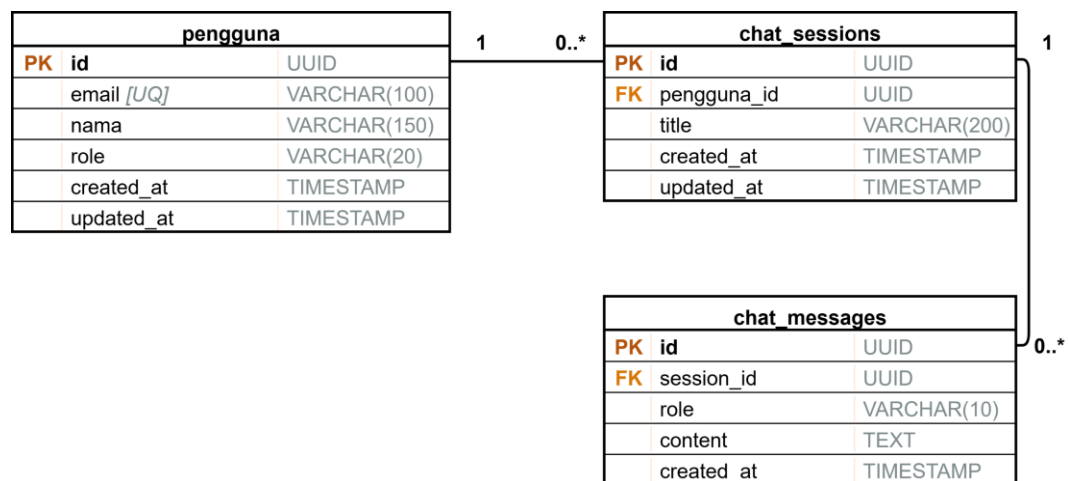
Gambar 5.12 Class Diagram Iterasi 1

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.5 Entity Relationship Diagram (ERD)

ERD Iterasi 1 memodelkan arsitektur basis data relasional fondasi yang mencakup tiga tabel utama pada PostgreSQL Supabase:

1. Tabel **pengguna**: Menyimpan data akun pengguna dengan kolom kunci primer *id* (tipe UUID) yang terhubung dengan Supabase Auth.
2. Tabel **chat_sessions**: Menyimpan riwayat sesi percakapan yang diikat oleh kunci asing *pengguna_id* ke tabel *pengguna(id)*.
3. Tabel **chat_messages**: Menampung pertukaran pesan pengguna dan asisten AI yang diikat oleh kunci asing *session_id* ke tabel *chat_sessions(id)* dengan aturan *ON DELETE CASCADE*.



Gambar 5.13 Entity Relationship Diagram Iterasi 1

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.1.6 Wireframe Antarmuka

Wireframe antarmuka Iterasi 1 dirancang dalam tiga Wireframe yang mencerminkan dua varian tampilan berdasarkan peran pengguna.

5.4.1.6.1 Halaman Login

Wireframe halaman Login Menampilkan logo lembaga, judul sistem, kolom email dan kata sandi, tautan *Lupa Sandi*, kotak centang *Biarkan saya tetap masuk*, dan tombol masuk utama.

AISYA-RA

https://aisya.ra-alislam.sch.id/login

logo

Masuk ke AISYA-RA

Alamat Email
guru@alislam.sch.id

Kata Sandi Lupa Sandi?

☐ Biarkan saya tetap masuk

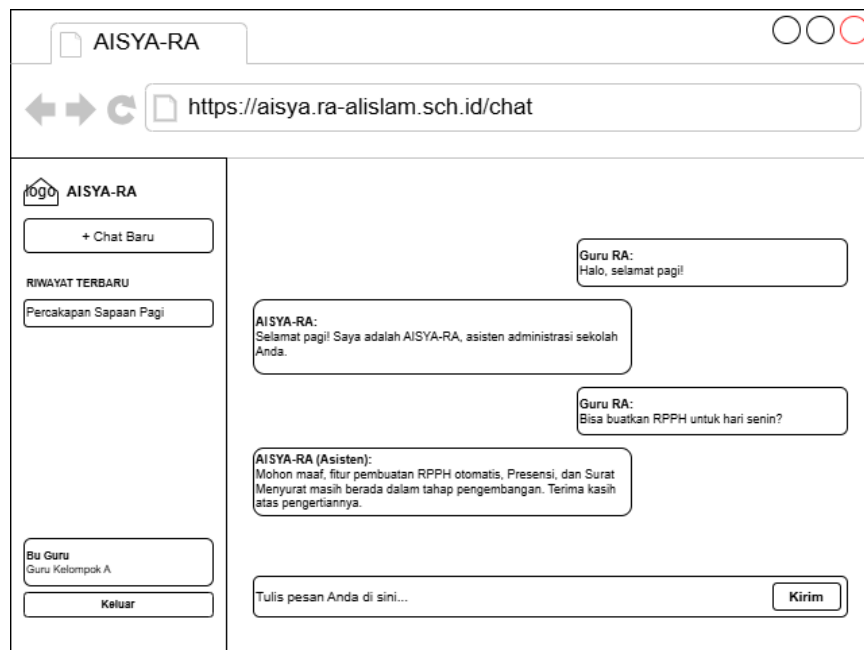
Masuk Aplikasi

Gambar 5.14 Wireframe Antarmuka 1 - Halaman Login

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

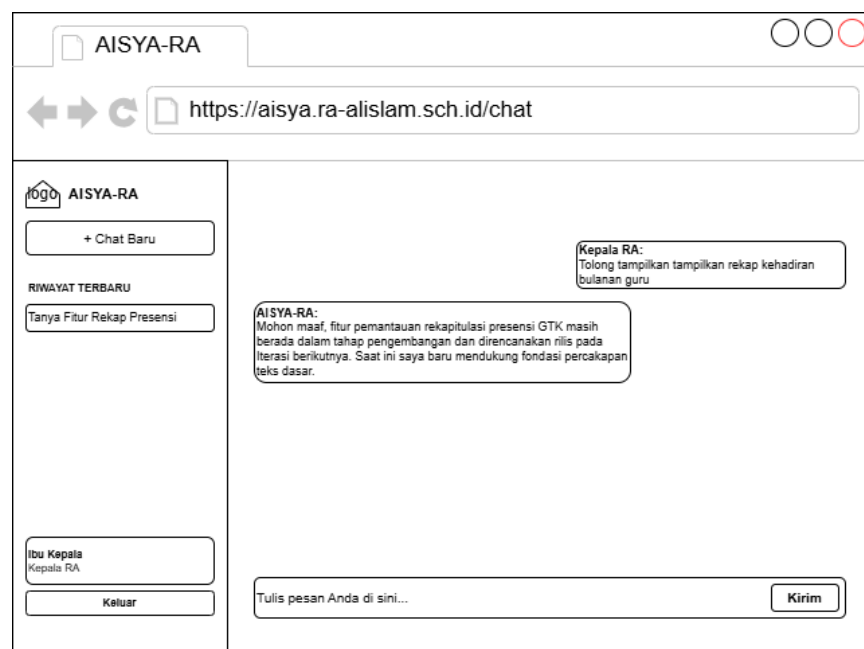
5.4.1.6.2 Halaman Chat (Guru RA & Kepala RA)

Wireframe halaman *Chat* (Guru RA) Menampilkan bilah samping daftar riwayat obrolan, tombol sesi baru, kartu identitas pengguna di sudut kiri bawah, jendela percakapan utama, serta kotak input teks pesan di bagian bawah.



Gambar 5.15 Wireframe Antarmuka 1 - Halaman Chat (Guru RA)

(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)



Gambar 5.16 Wireframe Antarmuka Iterasi 1 – Halaman Chat (Kepala RA)

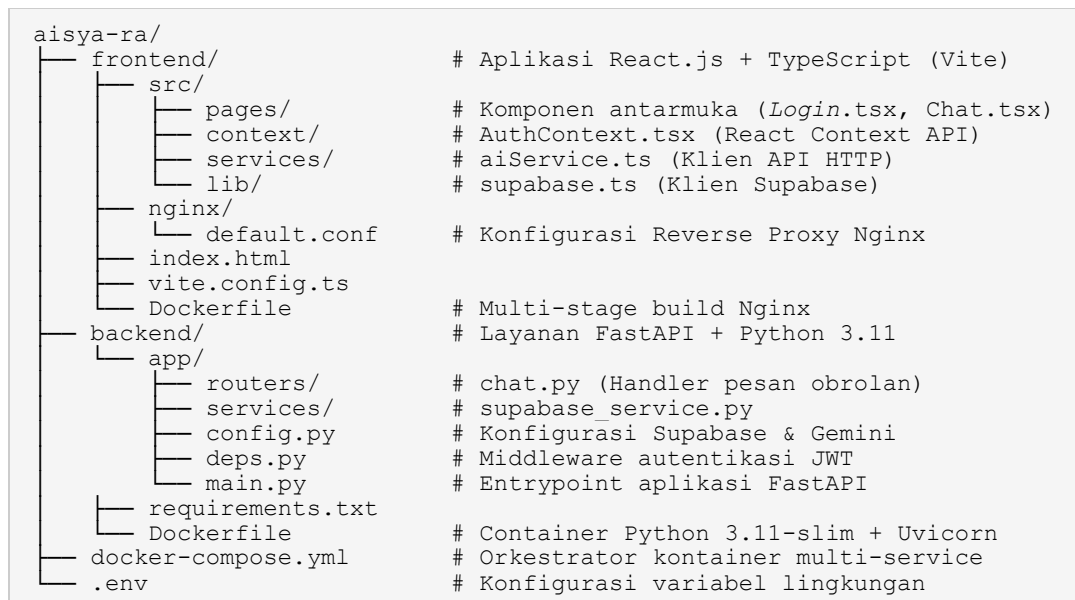
(Sumber: Hasil rancangan tahap design iterasi 1 AISYA-RA, 2026)

5.4.2 Develop Iterasi 1

Implementasi Iterasi 1 dilaksanakan selama kurang lebih dua minggu sesuai *timebox* yang ditetapkan. Selain mengimplementasikan item *backlog*, Iterasi 1 menetapkan struktur direktori proyek dan konfigurasi server yang berlaku sebagai fondasi bagi seluruh iterasi berikutnya.

5.4.2.1 Struktur Direktori dan Kontainerisasi

Arsitektur kode program pada Iterasi 1 difokuskan pada pemisahan yang tegas antara antarmuka pengguna (*frontend*) dan layanan pemroses logika (*backend*).



Kode Program 5.1 Struktur Direktori Proyek AISYA-RA

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.1 memperlihatkan struktur pengorganisasian berkas proyek pada fondasi awal sistem. Direktori *frontend/* memuat seluruh kode antarmuka berbasis React 18 dan TypeScript, dengan pemisahan lapisan komponen tampilan pada folder *pages/*, manajemen *state* global pada *context/*, serta lapisan komunikasi jaringan pada *services/*. Sementara itu, direktori *backend/* menggunakan kerangka kerja FastAPI dengan struktur modular yang menempatkan pengendali rute pada *routers/*, fungsi injeksi dependensi dan otorisasi pada *deps.py*, serta pengaturan

koneksi pihak ketiga pada config.py. Struktur ini memastikan prinsip pemisahan tanggung jawab (*Separation of Concerns*) terpelihara sejak iterasi awal.

5.4.2.2 Arsitektur Kontainerisasi Docker

AISYA-RA dikontainerisasi menggunakan Docker dengan pendekatan *multi-container*. Aplikasi terdiri dari dua kontainer: `aisya_backend` yang menjalankan FastAPI melalui server ASGI Uvicorn, dan `aisya_frontend` yang menjalankan Nginx untuk menyajikan berkas statis hasil kompilasi React sekaligus bertindak sebagai *reverse proxy* yang meneruskan permintaan berawalan `/api/*` ke *backend*. Kedua kontainer berkomunikasi melalui jaringan internal Docker `aisya-network`.

```
# backend/Dockerfile - Layanan Backend (FastAPI + Uvicorn)
FROM python:3.11-slim
WORKDIR /app
COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt
COPY . .
EXPOSE 8000
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--port", "8000"]
```

*Kode Program 5.2 Dockerfile Layanan Backend
(FastAPI + Uvicorn)*

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.2 menunjukkan konfigurasi berkas Dockerfile untuk layanan *backend*. Pemilihan citra dasar `python:3.11-slim` bertujuan meminimalkan ukuran citra Docker dan mengoptimalkan kecepatan *build*. Seluruh dependensi pustaka Python diinstalasi secara efisien menggunakan opsi `--no-cache-dir` dari berkas `requirements.txt`. Server Uvicorn dijalankan dengan mendengarkan pada alamat host `0.0.0.0` dan port internal `8000`, sehingga dapat menerima lalu lintas panggilan API secara aman dari kontainer lain di dalam jaringan Docker.

```
# frontend/Dockerfile - Layanan Frontend (React + Nginx, Multi-stage Build)
FROM node:20-alpine AS builder
WORKDIR /app
COPY package*.json ./
RUN npm ci
```

```

COPY . .
RUN npm run build

# --- Stage 2: Production Runner ---
FROM nginx:stable-alpine
COPY --from=builder /app/dist /usr/share/nginx/html
COPY nginx/default.conf /etc/nginx/conf.d/default.conf
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]

```

*Kode Program 5.3 Dockerfile Layanan Frontend
(React + Nginx, Multi-stage Build)*

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.3 mengimplementasikan teknik pembangunan bertahap (*multi-stage build*) pada layanan *frontend*. Tahap pertama (*stage builder*) memanfaatkan lingkungan `node:20-alpine` untuk mengompilasi kode sumber TypeScript dan React menjadi aset web statis (HTML, CSS, JavaScript) di dalam direktori `dist/`. Tahap kedua (*production runner*) menyalin berkas artefak tersebut ke dalam citra peladen `nginx:stable-alpine` yang sangat ringan, lalu menerapkan berkas konfigurasi `nginx/default.conf` untuk melayani permintaan klien pada port publik 80. Pendekatan ini menghasilkan citra produksi yang ramping dan aman karena tidak menyertakan dependensi pembangunan Node.js.

```

# docker-compose.yml - Orkestrator Layanan AISYA-RA
version: "3.9"
services:
  backend:
    build: ./backend
    container_name: aisyah_backend
    env_file: .env
    expose:
      - "8000"
    networks:
      - aisyah-network
    restart: unless-stopped

  frontend:

```

```

    build: ./frontend
    container_name: aisyah_frontend
    ports:
      - "80:80"
    depends_on:
      - backend
    networks:
      - aisyah-network
    restart: unless-stopped

networks:
  aisyah-network:
    driver: bridge

```

Kode Program 5.4 docker-compose.yml - Orkestrator Layanan AISYA-RA

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.4 mengonfigurasi orkestrasi kedua layanan aplikasi melalui Docker Compose. Layanan backend hanya mengekspos port 8000 ke dalam jaringan privat aisyah-network tanpa membuka port ke luar secara langsung, guna mencegah akses tidak sah. Sementara itu, layanan frontend memetakan port publik 80:80 dan memiliki ketergantungan (*depends_on*) terhadap layanan backend. Berkas konfigurasi variabel lingkungan `.env` diinjeksikan secara otomatis ke dalam *runtime* kontainer.

5.4.2.3 Konfigurasi Lingkungan Sistem

Pengelolaan parameter koneksi dan kunci otentikasi disimpan terisolasi di dalam berkas konfigurasi lokal.

```

# .env - Konfigurasi Variabel Lingkungan
GEMINI_API_KEY="AIzaSy..."
VITE_SUPABASE_URL="https://xxx.supabase.co"
VITE_SUPABASE_ANON_KEY="eyJhbG..."
SUPABASE_SERVICE_ROLE_KEY="eyJhbG..."

```

Kode Program 5.5 .env - Konfigurasi Variabel Lingkungan

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.5 memuat konfigurasi kredensial variabel lingkungan yang bersifat rahasia (*sensitive data*). Parameter GEMINI_API_KEY digunakan oleh backend untuk memanggil model generatif Google, VITE_SUPABASE_URL dan VITE_SUPABASE_ANON_KEY digunakan oleh lapisan klien antarmuka untuk otentikasi publik, sedangkan SUPABASE_SERVICE_ROLE_KEY digunakan secara eksklusif oleh backend FastAPI untuk mengeksekusi operasi basis data dengan hak istimewa tingkat tinggi. Berkas ini disimpan di server lokal dan dikecualikan dari *version control repository*.

5.4.2.4 Implementasi Autentikasi dan Manajemen *State*

Sistem autentikasi diimplementasikan menggunakan **Supabase Auth**. Di sisi *frontend*, `AuthContext.tsx` mengelola *state* pengguna dan memulihkan sesi persisten (`localStorage` jika *rememberMe* = *true* atau `sessionStorage` jika *false*).

```
// frontend/src/context/AuthContext.tsx (Inisialisasi Sesi)
export function AuthProvider({ children }: { children: ReactNode }) {
  const [user, setUser] = useState<User | null>(null);
  const [session, setSession] = useState<Session | null>(null);
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    const initializeAuth = async () => {
      // Pulihkan sesi aktif dari Supabase Auth
      const { data: { session: initSession } } = await
supabase.auth.getSession();
      setSession(initSession);
      setUser(initSession ? (await supabase.auth.getUser()).data.user :
null);
      setLoading(false);

      // Pantau perubahan status autentikasi secara reaktif
      supabase.auth.onAuthStateChange((_event, session) => {
        setSession(session);
        setUser(session?.user ?? null);
      });
    };
    initializeAuth();
  }, []);
```

```

const signOut = async () => {
  await supabase.auth.signOut();
  setUser(null);
  setSession(null);
};

return (
  <AuthContext.Provider value={{ user, session, loading, signOut }}>
    {children}
  </AuthContext.Provider>
);
}

```

*Kode Program 5.6 AuthContext.tsx - Inisialisasi Sesi
dan Manajemen State Autentikasi*

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.6 memperlihatkan logika inisialisasi sesi pada komponen penyedia konteks `AuthProvider`. Fungsi `initializeAuth()` memanggil `supabase.auth.getSession()` untuk memeriksa token sesi pengguna yang tersimpan di media penyimpanan peramban. Jika token valid ditemukan, *state* user dan session diperbarui sehingga antarmuka tidak meminta pengguna melakukan *login* ulang. Selain itu, fungsi pendengar `onAuthStateChange` didaftarkan untuk menangani pembaruan status sesi secara *real-time*, seperti ketika token kedaluwarsa atau saat fungsi `signOut()` dieksekusi.

Di sisi *backend*, modul `backend/app/deps.py` menyediakan fungsi dependensi FastAPI untuk memverifikasi keabsahan token JWT yang dikirimkan pada *header* HTTP setiap permintaan.

```

# backend/app/deps.py - Middleware Validasi JWT
from fastapi import HTTPException, Depends
from fastapi.security import HTTPBearer,
HTTPAuthorizationCredentials
from app.config import config

security = HTTPBearer()

```

```

def get_current_user_token(credentials:
    HTTPAuthorizationCredentials = Depends(security)) -> dict:
    token = credentials.credentials
    try:
        sb = config.get_supabase()
        user_response = sb.auth.get_user(token)
        if not user_response or not user_response.user:
            raise HTTPException(status_code=401, detail="Invalid
token")
        return {"sub": user_response.user.id}
    except Exception as e:
        raise HTTPException(status_code=401,
detail=f"Authentication failed: {str(e)}")

def get_current_user(token_payload: dict =
    Depends(get_current_user_token)):
    user_id = token_payload.get("sub")
    sb = config.get_supabase()
    res = sb.table('pengguna').select('*').eq('id',
user_id).execute()
    if not res.data:
        raise HTTPException(status_code=403, detail="User not
found")
    return res.data[0]

```

Kode Program 5.7 deps.py - Validasi JWT dan Pengambilan Profil Pengguna

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.7 mengimplementasikan fungsi otorisasi `get_current_user_token` yang mengekstrak token dari skema HTTPBearer. Token tersebut diverifikasi langsung ke server Supabase Auth melalui pemanggilan `sb.auth.get_user(token)`. Apabila validasi berhasil, fungsi turunan `get_current_user` mengeksekusi kueri SELECT ke tabel `pengguna` berdasarkan `user_id` untuk mengambil seluruh informasi profil, hak akses peran (*role*), dan atribut madrasah (`ra_id`), lalu mengembalikannya sebagai objek terotentikasi yang siap diinjeksi ke *endpoint router*.

Pada tahap fondasi Iterasi 1 ini, fokus pengembangan diarahkan pada penyediaan mekanisme autentikasi dan ruang obrolan (*Minimum Viable Product*). Untuk

memfasilitasi pengujian fungsionalitas tersebut, data kredensial awal bagi aktor Guru RA dan Kepala RA disuntikkan secara langsung ke dalam tabel pengguna di basis data Supabase melalui metode *database seeding* oleh peneliti. Pengelolaan akun pengguna secara mandiri melalui antarmuka grafis direncanakan untuk direalisasikan pada iterasi berikutnya bersamaan dengan modul data master.

5.4.2.5 Hasil Implementasi Antarmuka Autentikasi

Hasil implementasi antarmuka autentikasi mencakup formulir masuk (*login*), opsi persistensi sesi (*Biarkan saya masuk*), dan modal pemulihan kata sandi (*Lupa Sandi*) berbasis verifikasi identitas internal:

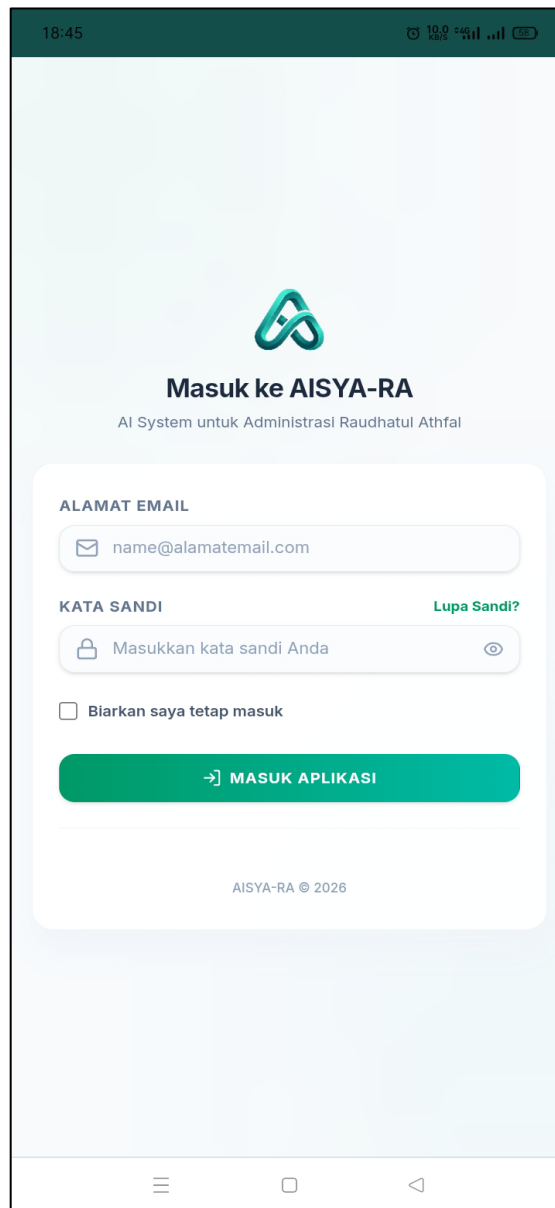
1. **Halaman *Login Desktop*:** Menyajikan antarmuka masuk yang bersih dan terpusat dengan validasi masukan langsung di sisi peramban (*client-side validation*).
2. **Modal *Lupa Sandi*:** Memfasilitasi pemulihan kata sandi secara mandiri dengan memverifikasi kesesuaian alamat email dan nomor identitas pendidik (NIP, NIK, atau NUPTK) yang terdaftar di basis data, tanpa memerlukan tautan email eksternal yang rumit bagi para pengajar di madrasah.
3. **Tampilan *Login Responsif (Smartphone)*:** Mengadaptasi tata letak antarmuka masuk secara proporsional untuk kenyamanan navigasi layar sentuh pada perangkat seluler guru.

Gambar 5.17 Tampilan Halaman Login AISYA-RA

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Gambar 5.18 Tampilan Modal Lupa Sandi

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)



Gambar 5.19 Tampilan Halaman Login AISYA-RA (Diakses dari Smartphone)

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

5.4.2.6 Implementasi *Chatbot* Dasar

Pengembangan modul percakapan dasar dikontrol oleh instruksi sistem (*System Instruction*) pada backend/app/routers/chat.py yang menetapkan persona, batasan topik, serta panduan respons penolakan ramah untuk fitur-fitur yang belum dirilis pada Iterasi 1.

```
# backend/app/routers/chat.py - SYSTEM_INSTRUCTION Iterasi 1
SYSTEM_INSTRUCTION = """
Anda adalah AISYA-RA, Asisten Administrasi Sekolah Pintar untuk
Raudhatul Athfal (RA) dan PAUD.

PERANAN ANDA:
Bantu guru dan kepala madrasah dalam menjawab pertanyaan seputar
administrasi dan kegiatan belajar mengajar di RA/PAUD melalui percakapan
ramah.

BATASAN TOPIK:
- Hanya layani topik yang berkaitan dengan PAUD/RA: administrasi dasar,
pedagogik, tumbuh kembang anak, parenting, dan kurikulum.
- Tolak dengan sopan pertanyaan di luar cakupan tersebut.

FITUR DALAM PENGEMBANGAN:
- Apabila pengguna meminta pembuatan draf RPPH, rekap presensi, atau
pembuatan surat dinas, sampaikan bahwa fitur otomasi tersebut sedang
disiapkan pada pembaruan iterasi berikutnya.
"""
```

Kode Program 5.8 chat.py - SYSTEM_INSTRUCTION Iterasi 1

(Persona Dasar + Batasan Topik)

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.8 mendefinisikan persona asisten cerdas AISYA-RA. *System Prompt* ini diinjeksikan pada setiap permintaan inferensi ke Google Gemini API untuk membatasi cakupan topik hanya pada ranah pendidikan anak usia dini dan tata kelola madrasah. Bagian *Fitur Dalam Pengembangan* secara eksplisit menginstruksikan model untuk merespons secara sopan dan informatif bahwa otomatisasi dokumen formal (seperti RPPH dan surat keluar) baru akan tersedia pada rilis iterasi berikutnya, sehingga mencegah model memberikan keluaran halusinasi atau di luar lingkup fungsional Iterasi 1.

Handler utama POST */chat* mengelola alur penyimpanan percakapan, pembentukan konteks riwayat obrolan, pemanggilan model Gemini via mekanisme *streaming Server-Sent Events* (SSE), serta persistensi balasan asisten ke basis data:

```
# backend/app/routers/chat.py - Alur Inti Handler POST /chat (Streaming SSE)
```

```

@router.post("/chat")
def handle_chat(req: ChatRequest, user = Depends(get_current_user)):
    # 1. Simpan pesan pengguna ke tabel chat_messages
    supabase.table('chat_messages').insert({
        "session_id": req.session_id,
        "role": "user",
        "content": req.message
    }).execute()

    # 2. Ambil 10 riwayat pesan terakhir sebagai konteks dialog
    history = supabase.table('chat_messages') \
        .select('*').eq('session_id', req.session_id) \
        .order('created_at').execute().data[-10:]

    contents = [{"role": "user" if m["role"] == "user" else "model",
                 "parts": [{"text": m["content"]}]} for m in history]

    # 3. Panggil Gemini API dengan streaming SSE
    payload = {
        "system_instruction": {"parts": [{"text": SYSTEM_INSTRUCTION}]},
        "contents": contents,
        "generationConfig": {"temperature": 0.2}
    }

    def event_generator():
        full_reply = []
        url =
f"https://generativelanguage.googleapis.com/v1beta/models/gemini-2.5-
flash:streamGenerateContent?key={config.GEMINI_API_KEY}&alt=sse"
        resp = requests.post(url, json=payload, stream=True)
        for line in resp.iter_lines():
            if line.startswith(b"data: "):
                chunk = json.loads(line.decode()[6:])
                for part in chunk.get("candidates",
[{}])[0].get("content", {}).get("parts", []):
                    if "text" in part:
                        full_reply.append(part["text"])
                        yield f"data:
{json.dumps({'type': 'content', 'content': part['text']})}\n\n"

    # Simpan respons asisten AI ke database
    supabase.table('chat_messages').insert({

```

```

        "session_id": req.session_id,
        "role": "model",
        "content": "".join(full_reply)
    }).execute()
    yield f"data:
{json.dumps({'type':'done','session_id':req.session_id})}\n\n"

    return StreamingResponse(event_generator(), media_type="text/event-
stream")

```

*Kode Program 5.9 chat.py - Alur Inti Handler POST /chat
(Simpan Pesan, Panggil Gemini, Stream SSE)*

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

Kode Program 5.9 memperlihatkan tiga tahapan utama pemrosesan pesan pada *handler* POST /chat:

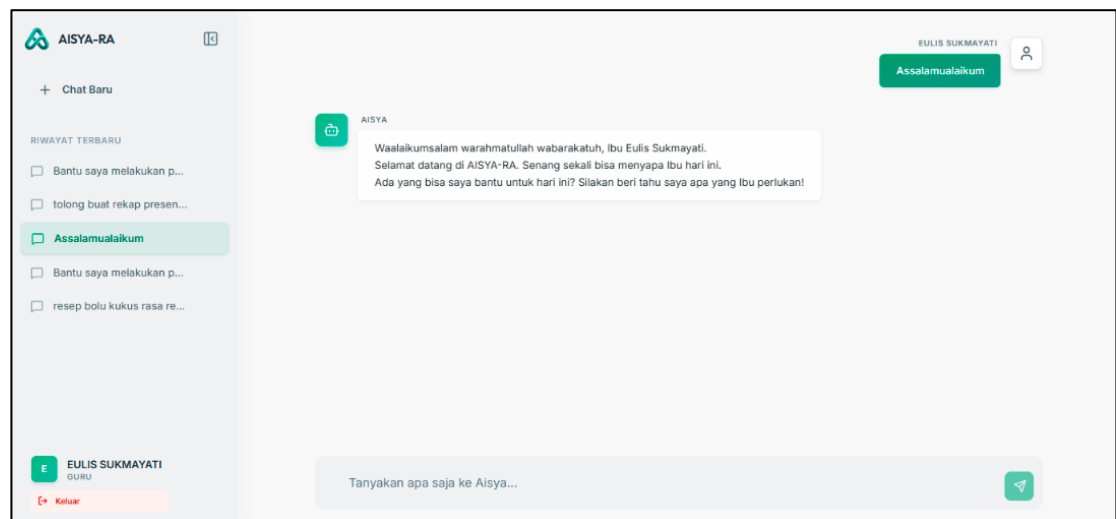
1. Pesan pengguna disimpan ke tabel `chat_messages` dengan peran `role: 'user'`, kemudian sistem mengambil maksimal 10 riwayat percakapan terakhir untuk dikirimkan sebagai memori konteks percakapan pendek (*short-term conversational context*).
2. Backend menyusun muatan *payload* yang memadukan instruksi sistem (`SYSTEM_INSTRUCTION`), riwayat percakapan terformat, dan konfigurasi suhu rendah (*temperature: 0.2*) untuk menjaga ketepatan jawaban.
3. Fungsi pembangkit `event_generator()` memanfaatkan fitur *Server-Sent Events* (SSE) untuk mengirimkan potongan teks respons (*text chunks*) secara seketika (*real-time streaming*) ke antarmuka pengguna seiring model menghasilkan kata-kata balasan, kemudian setelah seluruh aliran teks selesai, gabungan jawaban utuh disimpan secara permanen ke tabel `chat_messages` dengan peran `role: 'model'`.

5.4.2.7 Hasil Implementasi Antarmuka Ruang Chat

Hasil implementasi ruang obrolan *Chatbot* AISYA-RA menyajikan antarmuka percakapan yang ramah, panel bilah samping (*sidebar*) untuk penelusuran riwayat

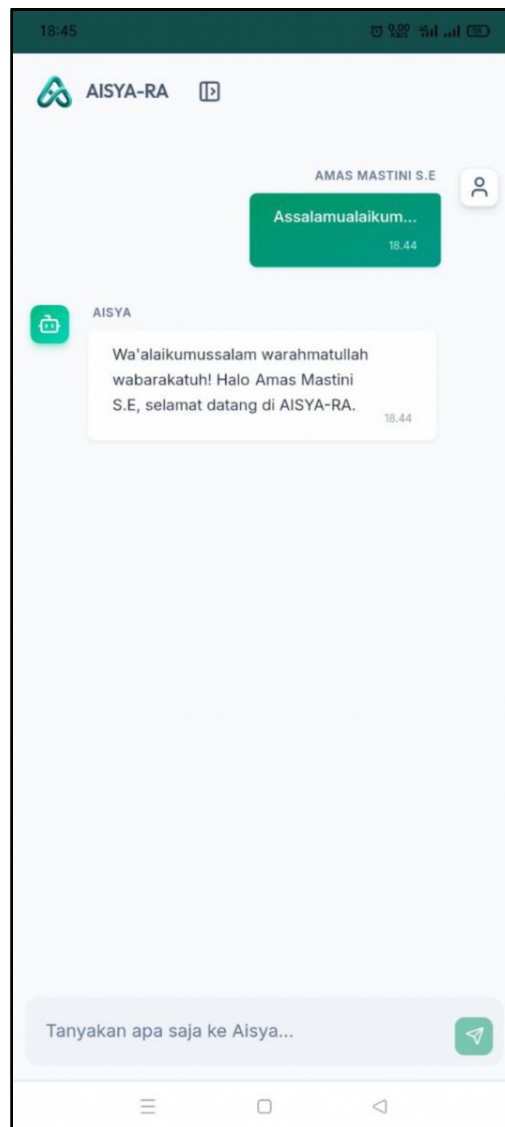
sesi, dan perenderan gelembung teks (*chat bubbles*) berbasis Markdown yang responsif di berbagai perangkat:

1. **Ruang Chat Desktop:** Menampilkan bilah navigasi riwayat percakapan di sebelah kiri, jendela dialog utama di tengah, dan informasi profil pengguna yang sedang masuk.
2. **Ruang Chat Smartphone:** Menggunakan laci navigasi geser (*collapsible drawer*) untuk riwayat obrolan, mengoptimalkan ruang layar sentuh saat pengguna mengetik pertanyaan.



Gambar 5.20 Tampilan Ruang Chat AISYA-RA - Antarmuka Guru RA

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)



Gambar 5.21 Tampilan Ruang Chat AISYA-RA (Diakses dari Smartphone)

(Sumber: Hasil pengembangan tahap develop iterasi 1 AISYA-RA, 2026)

5.4.3 Test Iterasi 1 - Black Box Testing

Pengujian pada Iterasi 1 difokuskan untuk memverifikasi keandalan fungsional fondasi sistem AISYA-RA yang mencakup mekanisme autentikasi pengguna (masuk, persistensi sesi JWT, pemulihan kata sandi via modal NIK/email, dan keluar), interaksi percakapan AI dasar berbasis streaming *Server-Sent Events* (SSE), serta manajemen riwayat obrolan (pemuatan, peralihan, dan penghapusan sesi).

Pengujian fungsionalitas dilaksanakan pada lingkungan lokal (*local staging environment*) di mana peladen dijalankan pada laptop pengembang (<http://localhost:80> dan <http://192.168.1.105>). Pengujian melibatkan perangkat laptop peneliti dan *smartphone* (Google Chrome Mobile) sebagai test client. Seluruh skenario diuji menggunakan dua akun awal hasil database seeding, yaitu satu akun Guru RA (R-02) dan satu akun Kepala RA (R-01) yang telah terdaftar pada Supabase Auth. Pengujian pada Iterasi 1 difokuskan untuk memverifikasi keandalan fungsional fondasi sistem AISYA-RA yang mencakup mekanisme autentikasi pengguna (masuk, persistensi sesi JWT, pemulihan kata sandi via modal NIK/email, dan keluar), interaksi percakapan AI dasar berbasis *streaming Server-Sent Events* (SSE), serta manajemen riwayat obrolan (pemuatan, peralihan, dan penghapusan sesi).

Pengujian fungsionalitas dilaksanakan pada lingkungan pementasan lokal (*local staging environment*) di mana peladen dijalankan pada laptop pengembang (<http://localhost:80> dan <http://192.168.1.105>). Pengujian melibatkan perangkat laptop peneliti dan *smartphone* (Google Chrome Mobile) sebagai *test client*. Seluruh skenario diuji menggunakan dua akun awal hasil *database seeding*, yaitu satu akun Guru RA (R-02) dan satu akun Kepala RA (R-01) yang telah terdaftar pada Supabase Auth.

Hasil pengujian disajikan pada Tabel 5.10 berikut.

Tabel 5.10 Hasil Black Box Testing Iterasi 1

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
TB-01	Login dengan kredensial valid	Pengguna berhasil masuk	Pengguna masuk ke ruang obrolan dengan profil yang sesuai.	Lulus

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
		dan dialihkan ke /chat.		
TB-02	<i>Login</i> dengan kata sandi salah	Sistem menampilkan pesan kesalahan <i>Login</i> .	Pesan kesalahan kredensial tidak valid muncul di layar.	Lulus
TB-03	Fitur "Biarkan saya tetap masuk"	Sesi <i>Login</i> bertahan saat tab peramban ditutup dan dibuka lagi.	Token JWT tersimpan di <i>localStorage</i> dan sesi otomatis aktif.	Lulus
TB-04	Pemulihan sandi via modal	Kata sandi berhasil diperbarui dengan verifikasi identitas NIK/email.	Kata sandi baru terupdate di database dan modal menampilkan pesan sukses.	Lulus
TB-05	<i>Logout</i> dari aplikasi	Sesi dibersihkan dan pengguna dialihkan ke / <i>Login</i> .	Token terhapus dan halaman otomatis kembali ke form <i>Login</i> .	Lulus

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
TB-06	Mengirim pesan obrolan dasar	<i>Chatbot</i>		
		membalas pesan sapaan pengguna secara interaktif.	Balasan AI muncul dalam gelembung percakapan secara <i>real-time</i> .	Lulus
TB-07	Memuat riwayat obrolan	Riwayat sesi percakapan lama tampil di bilah samping.	Daftar sesi termuat lengkap beserta judul dan tanggal.	Lulus
TB-08	Melanjutkan sesi lama	Mengklik sesi lama memuat pesan-pesan percakapan sebelumnya.	Seluruh gelembung <i>chat</i> dari sesi terpilih berhasil dirender ulang.	Lulus
TB-09	Menghapus sesi obrolan	Sesi terhapus dari bilah samping dan basis data.	Sesi dan seluruh pesan terkait terhapus secara <i>CASCADE</i> .	Lulus

(Sumber: Hasil pengujian black box testing iterasi 1 AISYA-RA, 2026)

Seluruh sembilan skenario pengujian dinyatakan lulus. Seluruh target kriteria Definition of Done Iterasi 2 sudah terpenuhi.

5.4.4 Deploy Iterasi 1

Tahap *Deploy* pada Iterasi 1 bertujuan untuk meluncurkan artefak perangkat lunak fondasi ke lingkungan pengujian operasional awal (*Local Staging Environment*)

guna memvalidasi portabilitas kontainer Docker dan memastikan sistem dapat diakses secara nirkabel oleh para pengguna di lingkungan madrasah.

Prosedur peluncuran Iterasi 1 dilaksanakan melalui tahapan teknis berikut:

1. **Penyusunan Lingkungan Peladen Lokal (*Local Host Server*):** Aplikasi dijalankan pada laptop peneliti yang difungsikan sebagai server lokal (*host server*). Perangkat server memiliki spesifikasi prosesor Intel Core i5 dengan memori RAM 16 GB dan menjalankan sistem operasi berbasis Linux Ubuntu 22.04 LTS yang telah terpasang Docker Engine v24.0 dan Docker Compose v2.20.
2. **Konfigurasi Jaringan Nirkabel Terisolasi (*Local Area Network*):** Server lokal dan gawai pengguna dihubungkan ke jaringan nirkabel Wi-Fi lokal RA Al-Islam Gunungcupu yang berada pada segmen subnet 192.168.1.0/24. Server lokal dialokasikan alamat IP privat statis 192.168.1.105.
3. **Penyuntikan Konfigurasi Lingkungan (*Environment Variables*):** Berkas .env disiapkan pada direktori kerja peladen dengan memuat kredensial GEMINI_API_KEY, URL instans Supabase, serta kunci otentikasi publik VITE_SUPABASE_ANON_KEY dan privat SUPABASE_SERVICE_ROLE_KEY.
4. **Pembangunan dan Orkestrasi Kontainer Docker:** Eksekusi perintah docker compose up --build -d dijalankan pada terminal peladen. Perintah ini membangun citra kontainer aisyah_backend (FastAPI) dan aisyah_frontend (React + Nginx Multi-stage) secara paralel, kemudian mengaktifkan kedua layanan di latar belakang dalam jaringan terisolasi aisyah-network.
5. **Verifikasi Kesiapan Layanan (*Health Check*):** Pemeriksaan status kontainer melalui perintah docker compose ps mengonfirmasi bahwa kontainer aisyah_backend aktif mendengarkan pada port internal 8000 dan

kontainer aisyah_frontend berhasil memetakan port HTTP publik 80:80. Pengujian perutean peramban lokal <http://localhost> memastikan aset statis antarmuka dan proksi API [/api/chat](http://localhost/api/chat) berfungsi normal tanpa kendala *Cross-Origin Resource Sharing* (CORS).

6. **Distribusi Akses Pengguna Akhir:** Para guru dan Kepala RA mengakses sistem secara nirkabel menggunakan peramban web pada ponsel pintar (*smartphone*) masing-masing dengan membuka tautan IP lokal <http://192.168.1.105>. Uji coba simultan pada empat perangkat seluler membuktikan bahwa formulir autentikasi masuk (*login*), sesi percakapan asisten, dan penyimpanan riwayat obrolan dapat dioperasikan secara lancar dengan waktu muat awal antarmuka rata-rata di bawah 1,2 detik.

5.4.5 Review Iterasi 1

Sesi *review* dilaksanakan segera setelah *deployment* terbatas, melibatkan Kepala RA serta dua perwakilan guru yang baru pertama kali membuka AISYA-RA. Kesan pertama yang tertangkap adalah tidak adanya jeda kebingungan saat pengguna berpindah ke ruang *Chatbot*, karena tampilannya dirasakan identik dengan aplikasi percakapan instan yang sering mereka gunakan.

Namun, tanpa diarahkan, salah seorang guru langsung mengetikkan "Buatkan RPPH untuk hari senin", berasumsi bahwa sistem sudah bisa mengerjakannya. Saat *Chatbot* menjawab fitur masih dalam pengembangan, terdapat sedikit kekecewaan. Hal ini menjadi kesempatan bagi peneliti untuk menjelaskan filosofi pengembangan iteratif.

R-01 menanggapi:

“Oh jadi ini baru pondasinya dulu ya, ibarat rumah baru pasang tiang. Nggak apa-apa, yang penting nanti jalan beneran. Yang saya suka, buka aplikasinya nggak pakai mikir, tinggal ketik kayak WhatsApp biasa.”

— Kepala RA (R-01), Sesi *Review* Iterasi 1

Selain menyoroti kemudahan navigasi dan ruang percakapan *chatbot*, Kepala RA juga mencermati komponen visual pada bilah samping (*sidebar*), khususnya kartu profil pengguna yang menampilkan nama dan peran pengguna saat *login*. Beliau mengusulkan agar bagian tersebut dapat dikembangkan lebih lanjut agar antarmuka terasa lebih interaktif:

“Di bagian pojok bawah itu kan sudah bagus muncul nama sama jabatan saya. Tapi kalau bisa, nanti ditambahkan fitur buat ganti foto profil sendiri langsung dari situ. Jadi biar kelihatan lebih personal dan menarik aplikasinya, tidak cuma tulisan saja.”

— Kepala RA (R-01), Sesi Review Review Iterasi 1

Karena Iterasi 1 baru menghadirkan modul autentikasi (*login/logout*), percakapan teks dasar, dan penyimpanan riwayat sesi obrolan, umpan balik dari para peserta *Review* pada iterasi ini mengonfirmasi arah pengembangan utama dan merumuskan fokus untuk iterasi selanjutnya:

1. **Validasi Antarmuka Percakapan:** Antarmuka *Chatbot* dinilai sudah sangat tepat karena ramah pengguna (*user-friendly*), intuitif, dan tidak membutuhkan sosialisasi teknis yang rumit.
2. **Penegasan Kebutuhan Administrasi Inti:** Guru menegaskan kembali harapan agar modul pembuatan RPPH pada Iterasi 2 benar-benar mengikuti format baku tujuh komponen madrasah yang selama ini digunakan secara manual.
3. **Pengelolaan Prioritas Backlog (Retrospektif Agile):** Terhadap usulan Kepala RA mengenai fitur kustomisasi foto profil pengguna, tim peneliti dan mitra menyepakati bahwa fitur tersebut dikategorikan sebagai kebutuhan sekunder (*nice-to-have*). Sesuai prinsip efisiensi *Agile*, item ini dimasukkan ke dalam daftar *backlog* pengembangan pascapenelitian, agar pelaksanaan Iterasi 2 dapat berfokus penuh pada penyelesaian modul otomatisasi administrasi sekolah inti yang menjadi kebutuhan kerja paling mendesak bagi para guru.

5.5 Iterasi 2: Modul Administrasi Inti

Iterasi 2 dilaksanakan selama dua setengah minggu sesuai *timebox* yang ditetapkan. Tujuan utama iterasi ini adalah mengimplementasikan modul otomatisasi administrasi sekolah yang dapat diakses langsung melalui perintah *Chatbot* bahasa alami maupun navigasi menu, dengan format keluaran yang konsisten dengan standar administrasi baku di RA Al-Islam Gunungcupu.

Item backlog yang dikerjakan pada Iterasi 2 meliputi:

1. Pembuatan draf RPPH otomatis via *chat* sesuai format baku 7 komponen kurikulum.
2. Pencatatan dan rekapitulasi presensi siswa harian dan bulanan via *chat*.
3. Pembuatan dan pengarsipan surat resmi/keluar via *chat* dengan penomoran berurutan otomatis.
4. Lembar kerja penyuntingan interaktif *AISYA Canvas Modal* (*preview, edit, simpan ke basis data, dan cetak PDF*).
5. Manajemen arsip dokumen sekolah.

Definition of Done (DoD) Iterasi 2 dinyatakan selesai ketika kode program berjalan tanpa kesalahan pada lingkungan lokal, seluruh fitur otomatisasi dokumen menghasilkan draf sesuai format baku lembaga, antarmuka *Canvas Modal* dapat melakukan penyuntingan dan pencetakan dokumen, serta seluruh skenario pengujian *Black Box Testing* dinyatakan lulus (*Passed*).

5.5.1 Design Iterasi 2

Tahap *Design* Iterasi 2 menghasilkan artefak perancangan sistem yang memperluas fondasi Iterasi 1.

5.5.1.1 Use Case Diagram

Use Case Diagram Iterasi 2 merupakan kelanjutan (*extension*) dari arsitektur fungsional Iterasi 1, mendefinisikan ruang lingkup fungsional baru untuk modul

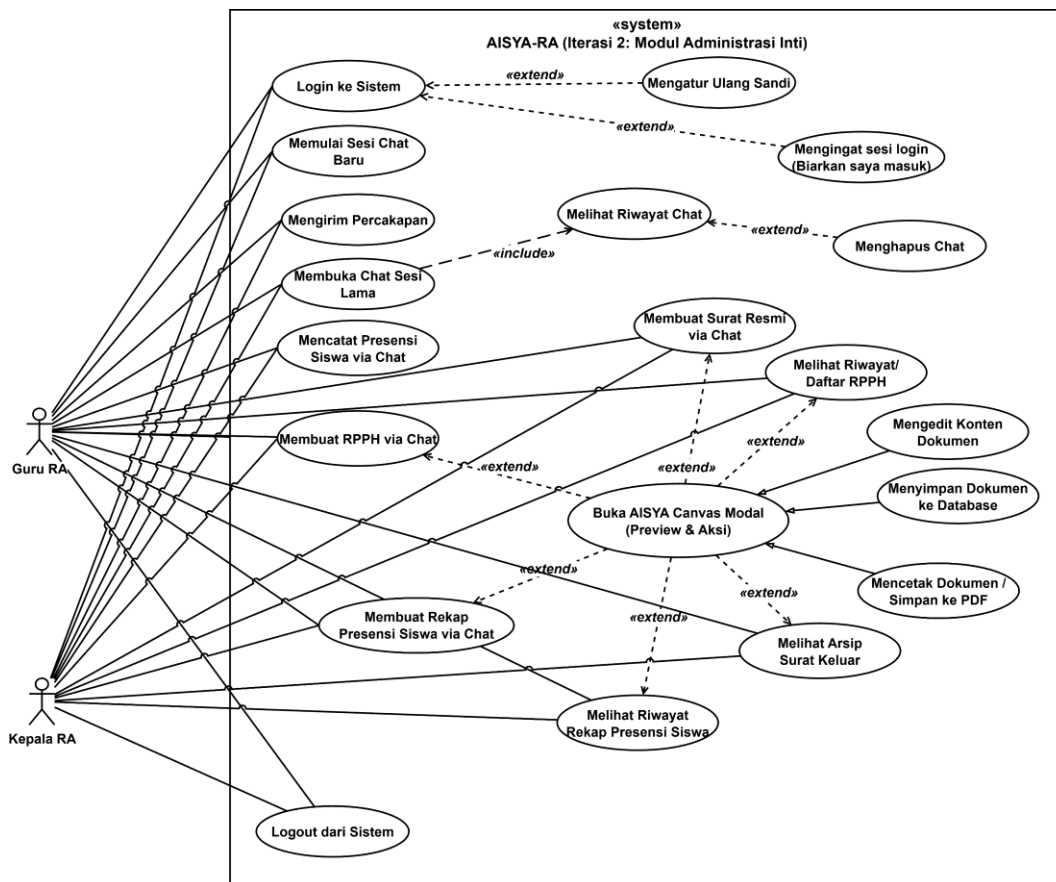
otomatisasi administrasi sekolah inti dan *AISYA Canvas Modal*. Dua aktor utama, yaitu **Guru RA** dan **Kepala RA**, berinteraksi dengan penambahan *Use Case* utama. Kedua aktor berinteraksi dengan tambahan *Use Case* utama sebagaimana disajikan pada Tabel 5.11 berikut.

Tabel 5.11 Use Case Diagram Iterasi 2

Use Case	Deskripsi Fungsional	Aktor Terkait
Membuat RPPH via Chat	Menginstruksikan <i>Chatbot</i> untuk menyusun draf RPPH berbasis 7 komponen kurikulum baku lembaga.	Guru RA, Kepala RA
Melihat Riwayat/Daftar RPPH	Mengakses dan menelusuri daftar arsip dokumen RPPH yang pernah dibuat.	Guru RA, Kepala RA
Mencatat Presensi Siswa via Chat	Menginput data kehadiran harian siswa secara alami melalui percakapan pesan obrolan.	Guru RA, Kepala RA
Membuat Rekap Presensi Siswa via Chat	Menginstruksikan <i>Chatbot</i> untuk mengagregasi data absensi siswa bulanan ke dalam format tabel rekap.	Guru RA, Kepala RA
Melihat Riwayat Rekap Presensi Siswa	Mengakses daftar rekapitulasi kehadiran siswa yang telah tersimpan di basis data.	Guru RA, Kepala RA
Membuat Surat Resmi via Chat	Menginstruksikan <i>Chatbot</i> untuk menyusun draf surat resmi keluar lengkap dengan format kop dinas.	Guru RA, Kepala RA
Melihat Arsip Surat Keluar	Mengakses dan menelusuri seluruh arsip surat dinas yang tersimpan di sistem.	Guru RA, Kepala RA

Use Case	Deskripsi Fungsional	Aktor Terkait
Buka AISYA Canvas Modal (Preview & Aksi)	Menampilkan lembar kerja visual modal untuk melihat pratinjau dan mengeksekusi aksi dokumen.	Guru RA, Kepala RA
Mengedit Konten Dokumen	Menyunting langsung teks dan struktur tabel draf dokumen pada antarmuka canvas (generalisasi <i>Canvas Modal</i>).	Guru RA, Kepala RA
Menyimpan Dokumen ke Database	Menyimpan dokumen hasil editan final ke tabel basis data Supabase (generalisasi <i>Canvas Modal</i>).	Guru RA, Kepala RA
Mencetak Dokumen / Simpan ke PDF	Mencetak fisik atau mengunduh dokumen dalam tata letak format PDF (generalisasi <i>Canvas Modal</i>).	Guru RA, Kepala RA

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)



Gambar 5.22 Use Case Diagram Iterasi 2

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

Melanjutkan realisasi cakupan pada tahap *Plan*, *Use Case Diagram* Iterasi 2 merumuskan *Use Case* “Membuat RPPH via Chat”, “Mencatat Presensi Siswa via Chat”, “Membuat Rekap Presensi Siswa via Chat”, dan “Membuat Surat Resmi via Chat” menjadi beberapa fungsi granular yang saling terkait. Iterasi 2 ini juga memperkenalkan *Use Case* baru “Buka AISYA Canvas Modal (Preview & Aksi)” beserta tiga sub-fungsi generalisasinya, yaitu “Mengedit Konten Dokumen”, “Menyimpan Dokumen ke Database”, dan “Mencetak Dokumen / Simpan ke PDF”. Rincian ini merupakan hasil rumusan teknis pada tahap Design Iterasi 2.

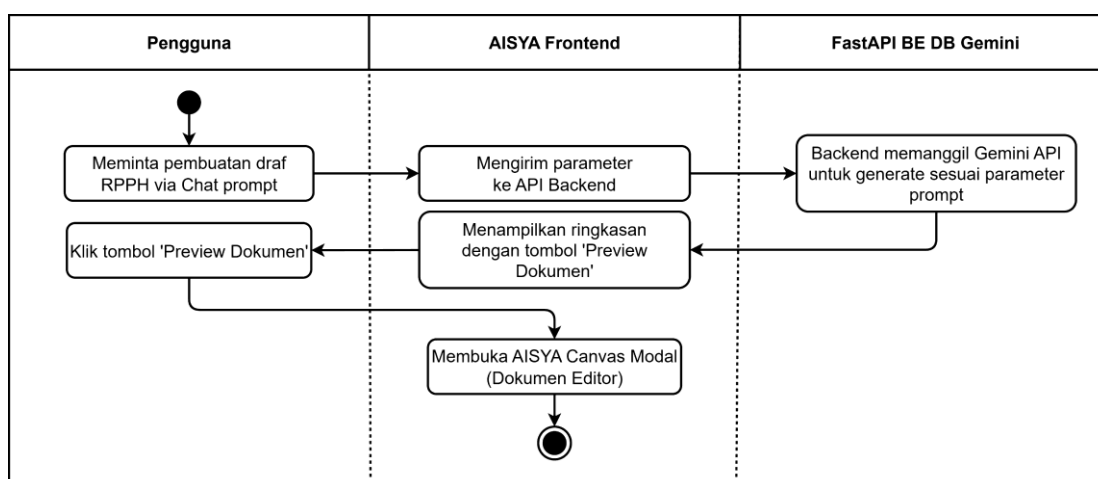
5.5.1.2 Activity Diagram

Pemodelan *Activity Diagram* pada tahap *Design* Iterasi 2 bertujuan untuk memetakan alur logika pada modul administrasi sekolah inti (RPPH, Presensi

Siswa, Rekapitulasi Presensi Siswa, dan Surat Keluar Resmi) serta integrasinya dengan komponen penyunting visual *AISYA Canvas Modal*. Pemodelan ini terbagi menjadi tujuh diagram aktivitas utama yang merincikan alur pertukaran data antar *swimlanes* aktor dan komponen sistem.

5.5.1.2.1 Activity Diagram Pembuatan RPPH (Rencana Pelaksanaan Pembelajaran Harian)

Activity Diagram Pembuatan RPPH menggambarkan alur saat guru memasukkan tema/sub-tema pembelajaran pada *Chat.tsx (Frontend)*, backend memicu *tool calling* untuk menyusun draf 7 komponen RPPH dengan GeminiAPI, kemudian merender kartu pratinjau, dan membuka lembar kerja *CanvasPanel* saat pengguna menekan tombol *Preview Dokumen*.

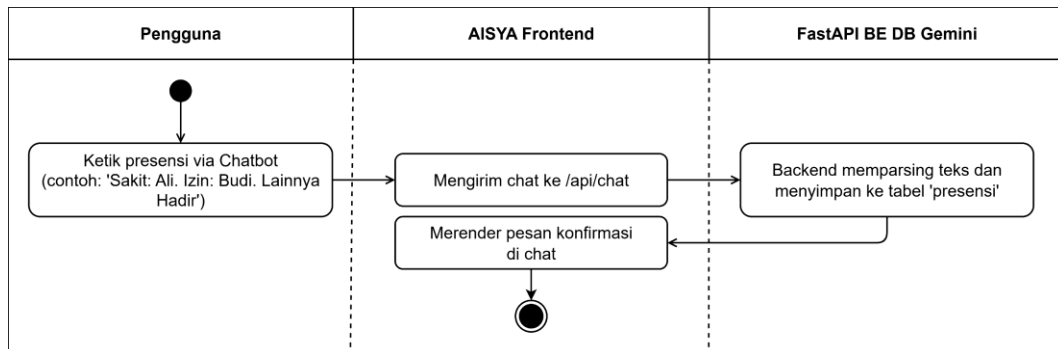


Gambar 5.23 Activity Diagram Iterasi 2 - Otomasi RPPH

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.2.2 Activity Diagram Pencatatan Presensi Siswa

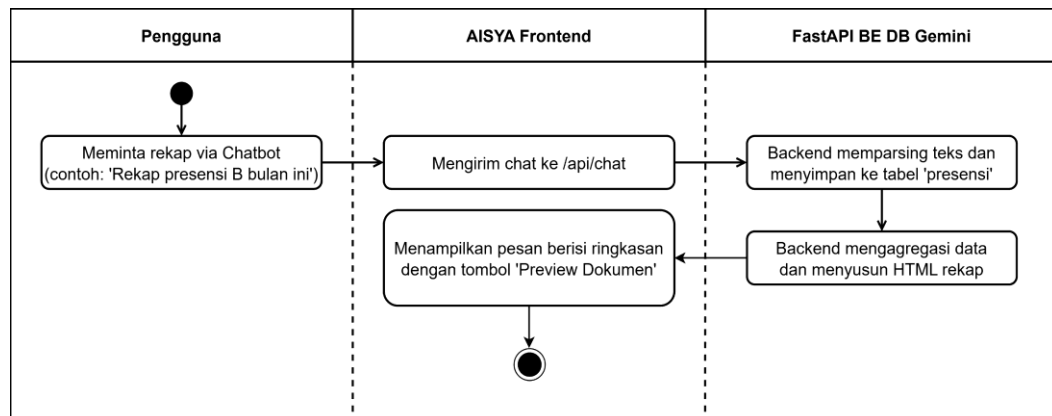
Activity Diagram Pencatatan Presensi Siswa menggambarkan alur saat pengguna mengetik daftar kehadiran harian siswa (misalnya "Semua hadir kecuali Budi sakit"), backend memanggil fungsi ekstraksi data untuk melakukan *UPSERT* data presensi siswa ke tabel *presensi* di Supabase, dan *Chatbot* menampilkan pesan konfirmasi kehadiran.



*Gambar 5.24 Activity Diagram Iterasi 2 - Pencatatan Presensi Siswa
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.2.3 Activity Diagram Rekap Presensi Siswa

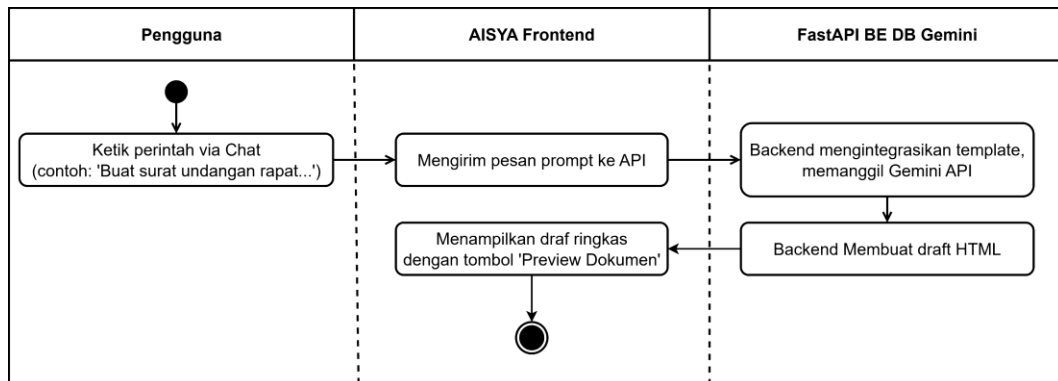
Activity Diagram Rekapitulasi Presensi Siswa menggambarkan kueri agregasi absensi siswa per rombongan belajar dalam periode bulan tertentu, membangkitkan draf tabel rekapitulasi, dan menampilkannya pada *editor canvas*.



*Gambar 5.25 Activity Diagram Iterasi 2 - Rekap Presensi Siswa
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.2.4 Activity Diagram Pembuatan Surat Keluar

Activity Diagram Pembuatan Surat Keluar menggambarkan alur perenderan templat surat dinas, injeksi susunan kop surat, dan pembentukan draf surat resmi pada editor canvas.

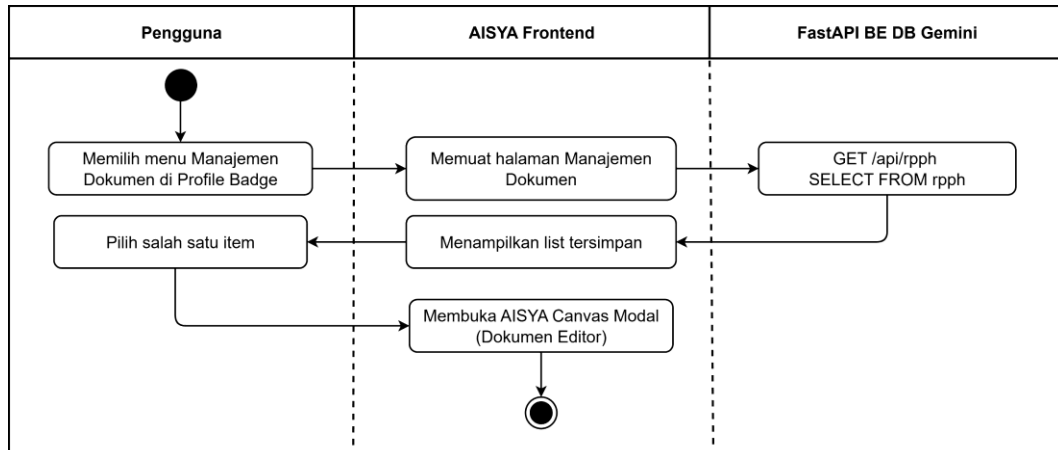


Gambar 5.26 Activity Diagram Iterasi 2 - Pembuatan Surat Keluar

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.2.5 Activity Diagram Manajemen Dokumen (Riwayat Dokumen)

Activity Diagram Manajemen Dokumen menggambarkan penelusuran arsip dokumen tersimpan pada antarmuka *Documents.tsx* dengan filter tab (RPPH, Rekap Presensi, Surat Resmi).

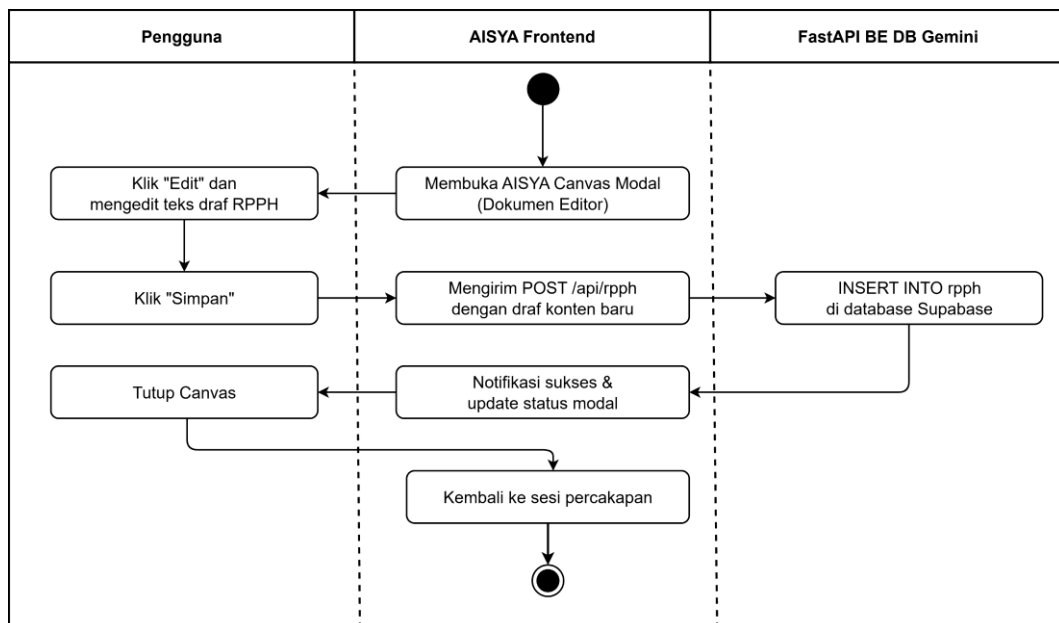


Gambar 5.27 Activity Diagram Iterasi 2 - Manajemen Dokumen

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.2.6 Activity Diagram AISYA Canvas Modal - Edit & Simpan Dokumen

Activity Diagram Edit & Simpan Dokumen menggambarkan penyuntingan teks langsung pada *CanvasPanel* dan penyimpanan data final ke basis data.

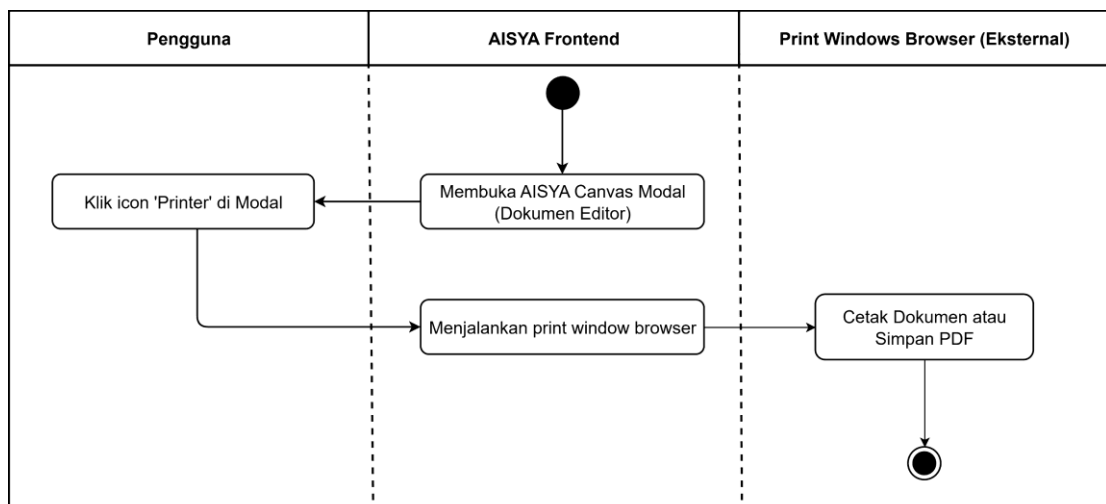


Gambar 5.28 Activity Diagram Iterasi 2 - Edit & Simpan Dokumen

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.2.7 Activity Diagram AISYA Canvas Modal - Cetak Dokumen / PDF

Activity Diagram Cetak Dokumen / PDF menggambarkan pembangkitan nomor surat resmi berurutan secara atomik dan pemanggilan fungsi dialog cetak peramban (*window.print*) untuk menghasilkan dokumen PDF siap cetak.



Gambar 5.29 Activity Diagram Iterasi 2 – Cetak Dokumen atau Simpan PDF

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.3 Sequence Diagram

Sequence Diagram pada tahap Design Iterasi 2 memodelkan pertukaran pesan antar modul pada proses otomatisasi penyusunan dokumen, lembar kerja interaktif AISYA Canvas Modal, dan manajemen arsip administrasi.

5.5.1.3.1 Pembuatan & Riwayat RPPH

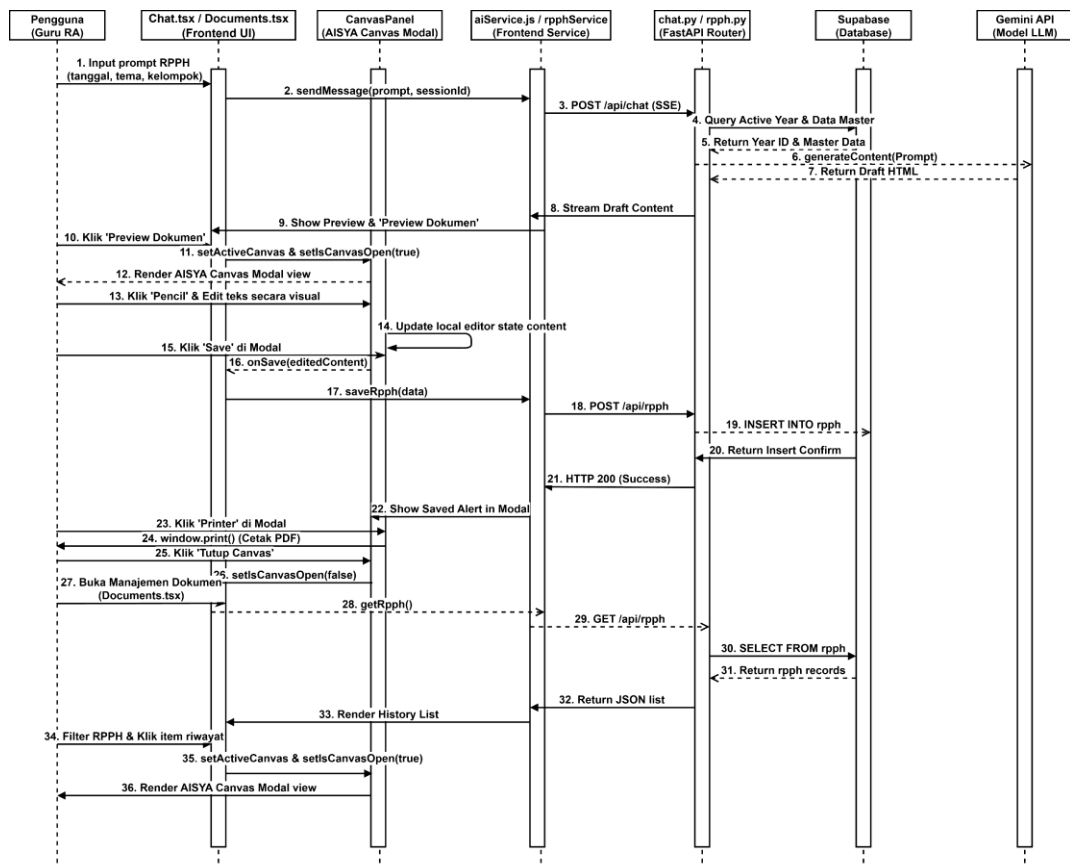
Sequence Diagram ini melibatkan komponen: Pengguna (*Kepala RA / Guru RA*), antarmuka Chat.tsx / Documents.tsx, editor visual CanvasPanel (*AISYA Canvas Modal*), layanan dataService.ts / aiService.ts, pengontrol backend chat.py / rpph.py, basis data Supabase, dan model kecerdasan buatan Gemini API.

Alur eksekusi pembuatan dan pengelolaan RPPH terdiri dari tahapan linier berikut:

1. **Pengguna** memasukkan instruksi penyusunan RPPH (memuat parameter tanggal, kelompok usia, tema, dan subtema) pada antarmuka percakapan **Chat.tsx**.
2. **Chat.tsx** memanggil fungsi `sendMessage(prompt, sessionId)` pada **aiService.ts**.
3. **aiService.ts** mengirimkan permintaan HTTP POST `/api/chat` (menggunakan mekanisme *Server-Sent Events / SSE*) ke backend **chat.py**.
4. Backend **chat.py** mengeksekusi kueri ke **Supabase** untuk mengambil konteks tahun ajaran aktif dan data profil madrasah.
5. **Supabase** mengembalikan data identitas dan periode akademik ke **chat.py**.
6. Backend **chat.py** menyusun *prompt* instruksi terstruktur 7 komponen baku dan memanggil metode `generateContent` ke **Gemini API**.
7. **Gemini API** memproses penyusunan rencana pembelajaran dan mengalirkan (*stream*) draf teks RPPH kembali ke **chat.py**.

8. Backend **chat.py** meneruskan aliran teks draf ke **aiService.ts** untuk ditampilkan pada **Chat.tsx** lengkap dengan kartu pratinjau dan tombol "Preview Dokumen".
9. **Pengguna** menekan tombol "Preview Dokumen" pada kartu percakapan.
10. **Chat.tsx** memanggil fungsi `setActiveCanvas(data)` dan mengeset status `setIsCanvasOpen(true)` untuk memuat draf dokumen ke dalam lembar kerja **CanvasPanel**.
11. **CanvasPanel** merender tampilan dokumen RPPH terstruktur dan mengaktifkan mode penyuntingan langsung (*live content editable*).
12. **Pengguna** menyesuaikan teks rincian kegiatan pembelajaran pada **CanvasPanel**.
13. **CanvasPanel** memperbarui *state* editor lokal secara reaktif.
14. **Pengguna** menekan tombol "Simpan" pada antarmuka canvas.
15. **CanvasPanel** memicu *callback* fungsi `onSave(editedContent)` ke **dataService.ts**.
16. **dataService.ts** mengirimkan panggilan HTTP POST `/api/rpph` ke backend **rpph.py**.
17. Backend **rpph.py** mengeksekusi kueri `INSERT INTO rpph` ke **Supabase**.
18. **Supabase** mengonfirmasi penyimpanan data dan mengembalikan nomor ID dokumen yang baru terbit.
19. Backend **rpph.py** mengembalikan respons HTTP 200 OK ke **dataService.ts**.
20. **CanvasPanel** menampilkan notifikasi bahwa dokumen RPPH telah tersimpan permanen di basis data.

21. **Pengguna** dapat menekan tombol "Cetak" pada modal canvas untuk memicu fungsi dialog cetak peramban (`window.print()`) dan mengekspor dokumen ke format PDF.
22. Pada alur penelusuran riwayat, **Pengguna** membuka halaman **Documents.tsx** yang memanggil `getRpph()` via **dataService.ts** (`GET /api/rpph`) ke backend dan **Supabase** untuk menampilkan daftar arsip RPPH, di mana pengguna dapat mengklik salah satu item untuk membukanya kembali pada **CanvasPanel**.



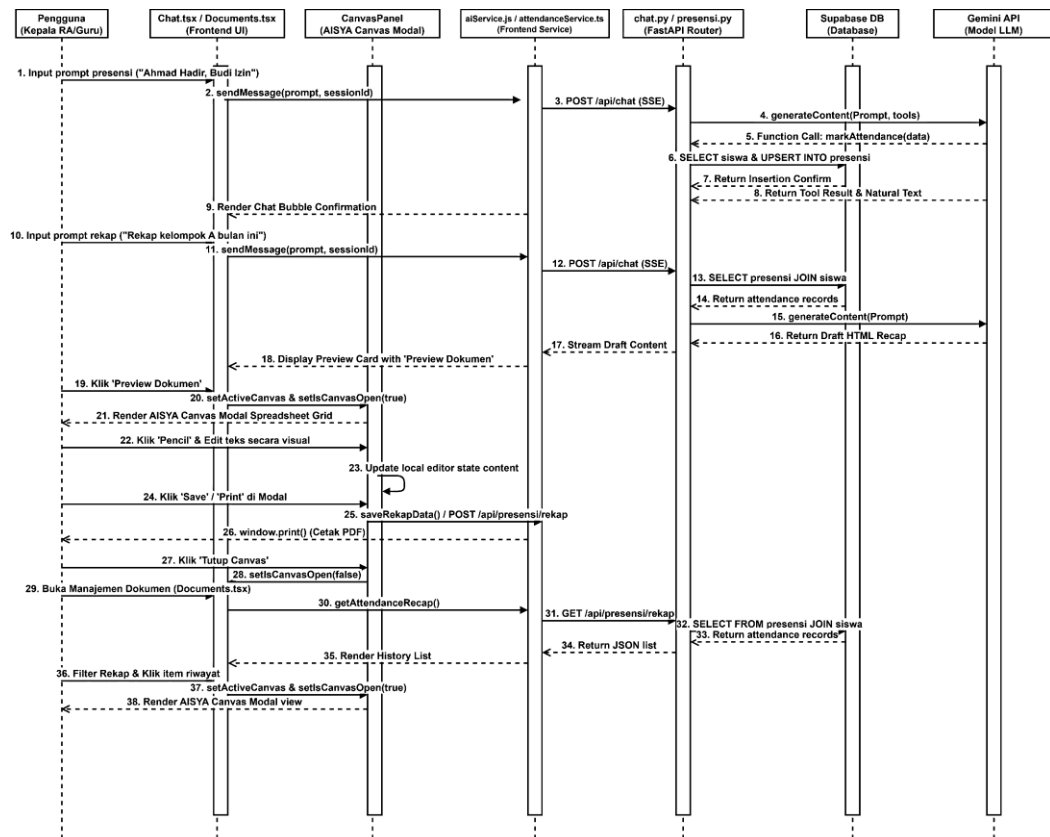
Gambar 5.30 Sequence Diagram Iterasi 2 - Otomasi & Riwayat RPPH
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.3.2 Pencatatan & Rekapitulasi Presensi

Sequence Diagram ini menggambarkan alur pencatatan kehadiran harian siswa dan pembuatan laporan rekapitulasi periodik. Alur eksekusi terdiri dari tahapan linier berikut:

1. **Pengguna** menginput instruksi pencatatan presensi harian siswa berbasis bahasa alami (misalnya: "*AISYA, catat presensi Kelompok A hari ini: Ahmad Hadir, Budi Sakit*") pada **Chat.tsx**.
2. **Chat.tsx** memanggil `sendMessage(prompt)` pada **aiService.ts**, yang meneruskan panggilan HTTP POST `/api/chat` ke backend **chat.py**.
3. Backend **chat.py** menyusun *prompt* dan memanggil metode `generateContent(Prompt, tools)` ke **Gemini API** untuk melakukan parsing intensi bahasa alami (*AI NLU Intent Parsing*).
4. **Gemini API** mengekstrak parameter presensi siswa dan mengembalikan *Function Call* `markAttendance(data)` ke backend **chat.py**.
5. Backend **chat.py** mengeksekusi kueri SELECT siswa dan kueri INSERT / UPSERT INTO presensi ke database **Supabase** untuk menyimpan rekam kehadiran harian siswa.
6. Basis data **Supabase** mengonfirmasi penyimpanan data, dan backend **chat.py** membangkitkan pesan balasan konfirmasi ramah yang dirender oleh **Chat.tsx**.
7. Pada alur pembuatan laporan rekapitulasi, **Pengguna** menginput perintah rekap (misalnya: "*AISYA, buatkan rekap presensi Kelompok A bulan ini*").
8. **Chat.tsx** memanggil fungsi `getAttendanceRecap()` via **dataService.ts** (GET `/api/presensi/rekap`) ke backend **presensi.py**.

9. Backend **presensi.py** mengeksekusi kueri SELECT presensi JOIN siswa ke **Supabase** untuk mengagregasi data absensi siswa dalam rentang tanggal yang diminta.
10. **Supabase** mengembalikan data agregasi kehadiran ke backend **presensi.py**.
11. Backend **presensi.py** bersama **Gemini API** membangkitkan draf tabel rekapitulasi presensi dan mengembalikannya ke **Chat.tsx** sebagai kartu pratinjau.
12. **Pengguna** menekan tombol "Preview Dokumen" pada kartu pratinjau.
13. **Chat.tsx** mengeset setActiveCanvas(recapData) dan setIsCanvasOpen(true) untuk menampilkan lembar rekapitulasi pada **CanvasPanel**.
14. **Pengguna** memverifikasi data dan menekan tombol cetak (window.print()) untuk menghasilkan dokumen PDF siap cetak.
15. Pada alur penelusuran riwayat, **Pengguna** dapat membuka tab Rekap Presensi pada antarmuka **Documents.tsx** untuk memuat seluruh rekapitulasi tersimpan ke dalam **CanvasPanel**.



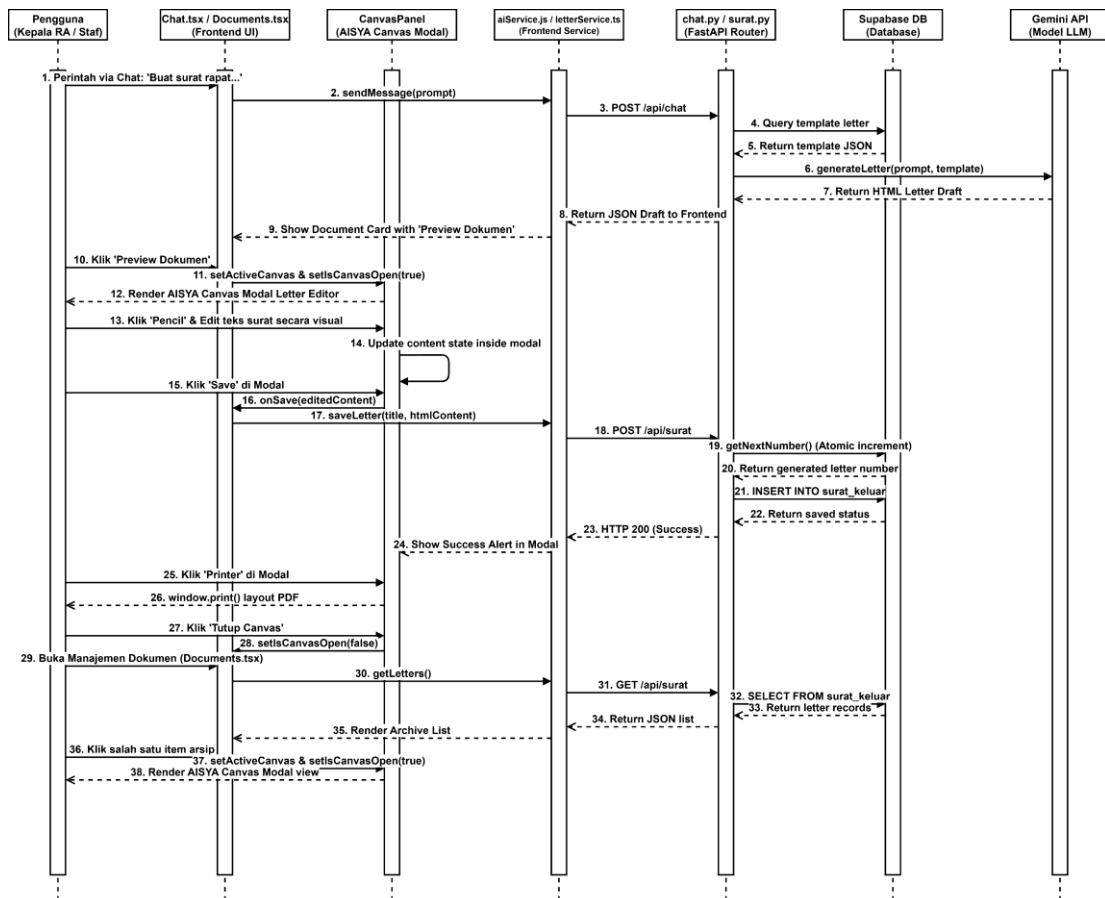
Gambar 5.31 Sequence Diagram Iterasi 2 - Pencatatan & Rekapitulasi Presensi
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.3.3 Pembuatan & Arsip Surat

Sequence Diagram ini menggambarkan alur penyusunan surat dinas resmi dengan penomoran surat otomatis. Alur eksekusi terdiri dari tahapan linier berikut:

1. **Pengguna** menginput perintah pembuatan surat resmi (misalnya permohonan izin atau undangan wali murid) pada antarmuka **Chat.tsx**.
2. Antarmuka **Chat.tsx** memanggil fungsi `sendMessage(prompt)` pada **aiService.ts** dan mengirimkan permintaan HTTP POST `/api/chat` ke backend **chat.py**.
3. Backend **surat.py** / **chat.py** mengeksekusi kueri ke **Supabase** untuk mengambil templat format surat baku dan identitas madrasah.
4. **Supabase** mengembalikan templat surat ke backend.

5. Backend menyusun *prompt* pemuatan surat dinas dan memanggil metode `generateContent` ke **Gemini API**.
6. **Gemini API** membangkitkan draf surat dinas resmi dan mengembalikannya ke backend.
7. Backend meneruskan draf surat ke **Chat.tsx** untuk ditampilkan sebagai kartu pratinjau dokumen.
8. **Pengguna** menekan tombol "Preview Dokumen" pada kartu pratinjau.
9. **Chat.tsx** membuka draf surat di dalam editor interaktif **CanvasPanel**.
10. **Pengguna** menyunting isi surat, perihal, atau titimangsa penandatanganan secara langsung pada **CanvasPanel** dan menekan tombol simpan.
11. **CanvasPanel** memicu fungsi *callback* `onSave()` ke **dataService.ts** yang mengirimkan permintaan HTTP POST `/api/surat` ke backend **surat.py**.
12. Backend **surat.py** mengeksekusi fungsi `getNextNumber()` di basis data **Supabase** untuk membangkitkan nomor surat resmi secara atomik (*atomic sequence counter*) berbasis kode klasifikasi lembaga dan tahun berjalan.
13. **Supabase** mengembalikan nomor surat resmi unik yang baru terbit.
14. Backend **surat.py** menyisipkan data surat final ke tabel *surat_keluar* di **Supabase** dengan status terarsip.
15. **Supabase** mengonfirmasi keberhasilan penyimpanan dan backend mengembalikan respons sukses ke **dataService.ts**.
16. **CanvasPanel** memperbarui tampilan nomor surat resmi dan mengaktifkan fungsi cetak dokumen PDF.



Gambar 5.32 Sequence Diagram Iterasi 2 - Pembuatan & Arsip Surat

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.4 Class Diagram

Class Diagram Iterasi 2 merupakan perluasan struktural dari tahap fondasi untuk mengakomodasi otomatisasi modul administrasi sekolah serta lembar kerja penyuntingan dokumen interaktif (*AISYA Canvas Modal*). Struktur kelas program diorganisasikan ke dalam tiga lapisan utama sebagai berikut:

1. Lapisan Antarmuka Pengguna (*User Interface Layer*):

- a. *RPPH (RPPH.tsx)*: Komponen pengelola dan pembuat draf Rencana Pelaksanaan Pembelajaran Harian (RPPH).
- b. *Attendance (Attendance.tsx)*: Komponen antarmuka pencatatan dan peninjauan rekapitulasi presensi harian siswa.

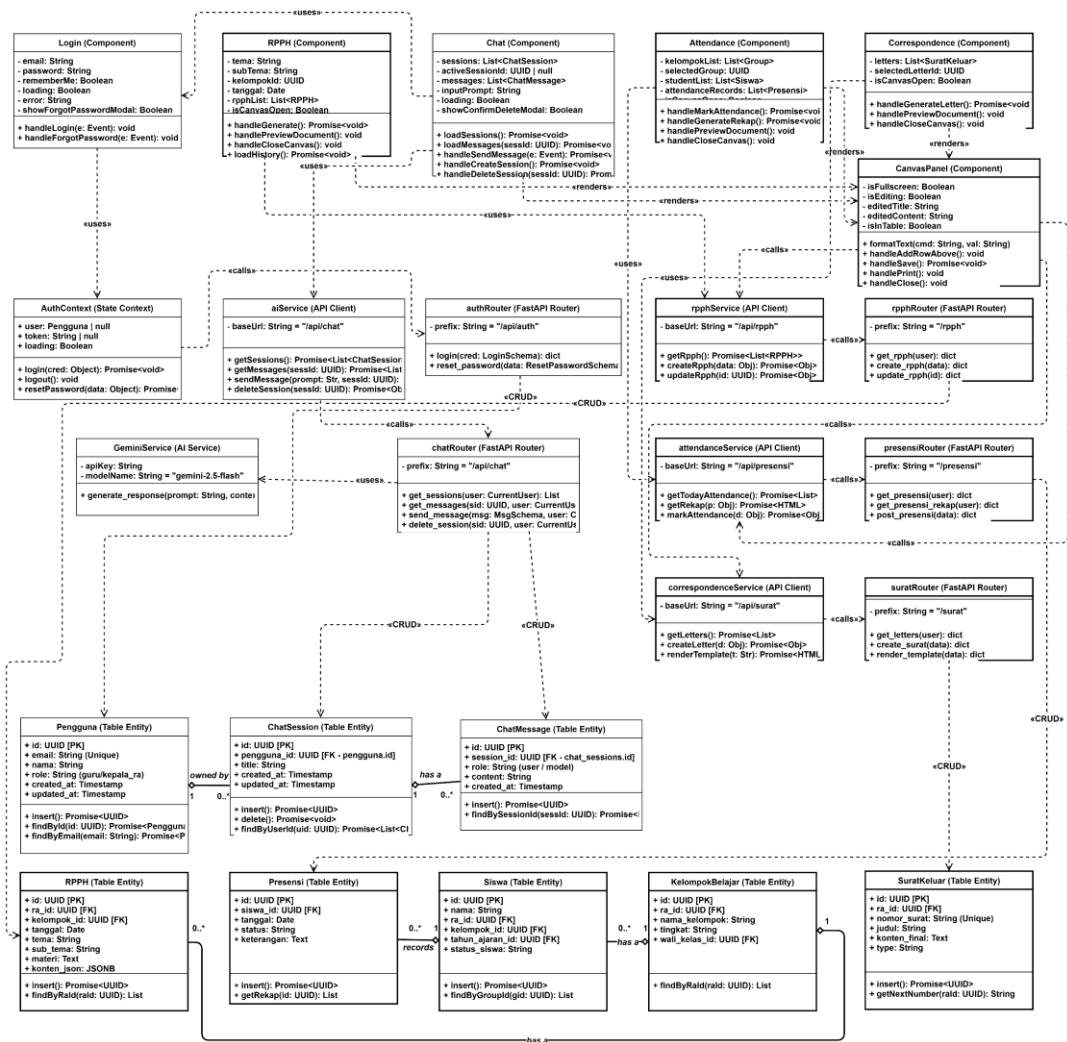
- c. *Correspondence (Correspondence.tsx)*: Komponen penyusunan dan pengarsipan surat dinas atau surat resmi lembaga.
- d. *Documents (Documents.tsx)*: Komponen sentral manajemen arsip seluruh dokumen administratif yang telah diterbitkan.
- e. *CanvasPanel (CanvasPanel.tsx)*: Komponen lembar kerja modal interaktif yang menyediakan fasilitas pratinjau dokumen (*preview*), penyuntingan teks langsung (*live editing*), penyimpanan data final ke basis data, dan fungsi pencetakan dokumen PDF melalui dialog cetak peramban (*browser print*).

2. Lapisan Logika dan Pengendali (*Logic & Controller Layer*):

- a. Modul Layanan Data Klien (*dataService.ts* dan *aiService.ts*): Mengabstraksikan pemanggilan API HTTP dan penanganan format dokumen (*rpphService*, *attendanceService*, dan *correspondenceService*).
- b. *rpphRouter (backend/app/routers/rpph.py)*: Melayani *endpoint* pembuatan, pembacaan, dan pembaruan dokumen RPPH berbasis format kurikulum baku.
- c. *presensiRouter (backend/app/routers/presensi.py)*: Melayani *endpoint* pencatatan kehadiran siswa dan agregasi rekapitulasi berkala kelompok belajar.
- d. *suratRouter (backend/app/routers/surat.py)*: Melayani *endpoint* perenderan templat surat dinas dan penomoran surat resmi secara berurutan (*atomic sequence*).
- e. *chatRouter (backend/app/routers/chat.py)*: Diperluas dengan kemampuan pemanggilan fungsi otomatis (*Function Calling*) untuk mengenali maksud pembuatan RPPH, pencatatan presensi, dan

penerbitan surat dinas langsung dari percakapan obrolan bahasa alami.

3. Lapisan Entitas Data (*Data Entity Layer*): Meliputi penambahan tujuh entitas basis data relasional yang saling terintegrasi, *Sekolah, TahunAjaran, KelompokBelajar, Siswa, Presensi, RP PH*, dan *SuratKeluar*.



Gambar 5.33 Class Diagram Iterasi 2

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

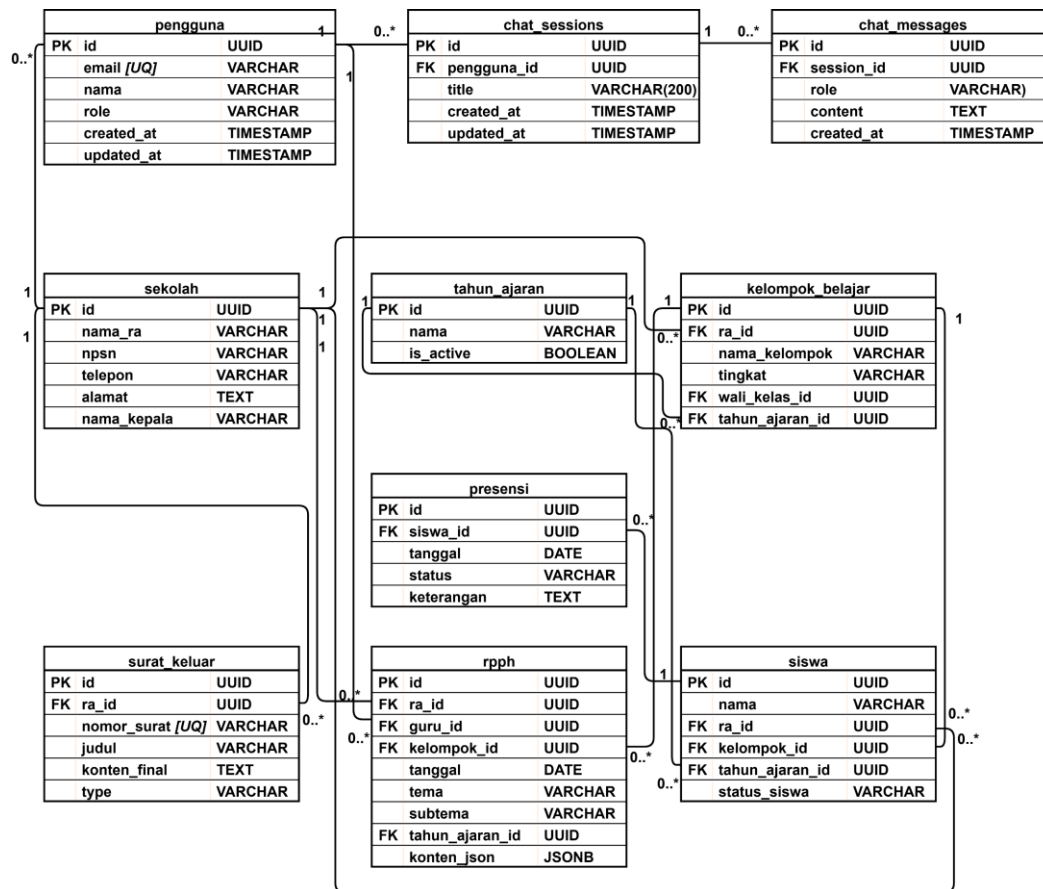
5.5.1.5 Entity Relationship Diagram (ERD)

Entity Relationship Diagram Iterasi 2 memodelkan penambahan entitas administrasi sekolah inti yang terhubung secara terstruktur terhadap data induk lembaga dan periode tahun pelajaran aktif.

Entitas **sekolah** bertindak sebagai entitas induk yang memiliki relasi *One-to-Many* (1 : 0..*) terhadap **kelompok_belajar**, **rpph**, dan **surat_keluar**.

Entitas **tahun_ajaran** mengikat secara *One-to-Many* (1 : 0..*) terhadap **kelompok_belajar**, **siswa**, **presensi**, dan **rpph** untuk memastikan data operasional terkategori dalam periode akademik yang aktif.

Struktur hierarki kehadiran dibangun melalui relasi *One-to-Many* (1 : 0..) dari **kelompok_belajar** ke **siswa**, yang selanjutnya terhubung *One-to-Many* (1 : 0..) ke entitas **presensi** dengan batasan unik pada kombinasi (*siswa_id*, *tanggal*) untuk mencegah duplikasi absensi harian.



Gambar 5.34 Entity Relationship Diagram Iterasi 2

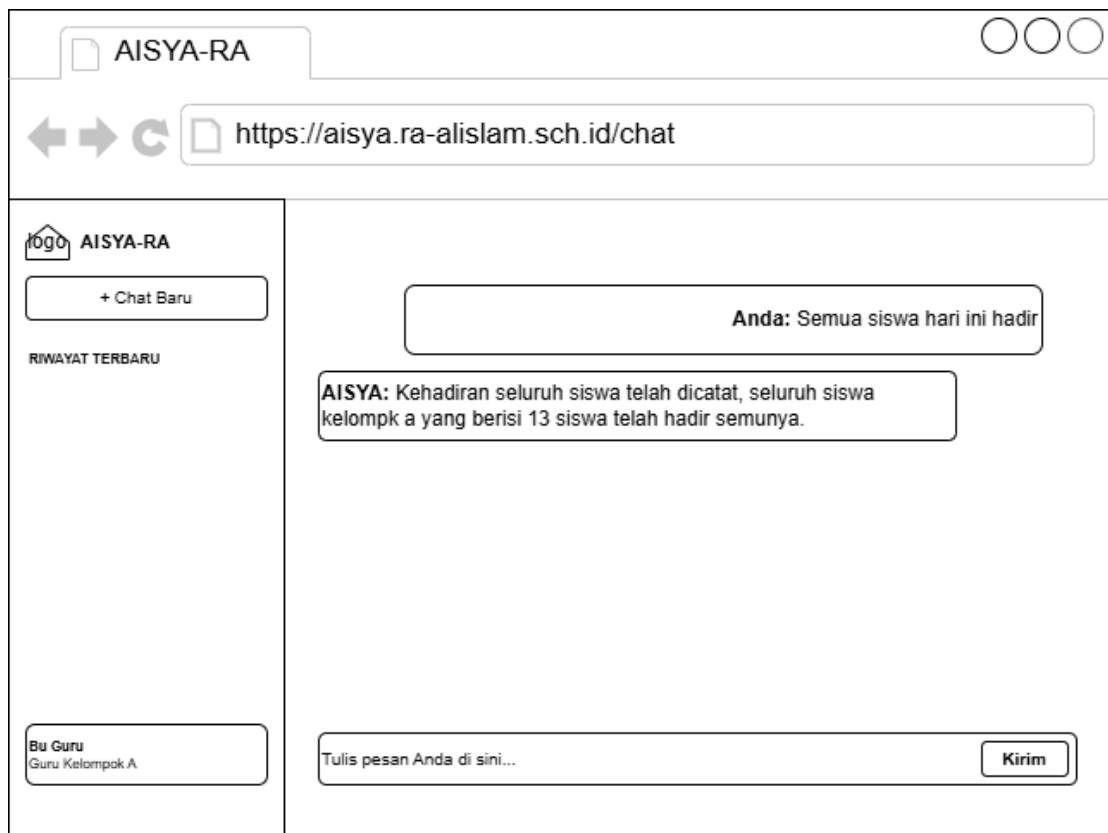
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.6 Wireframe Antarmuka

Wireframe antarmuka Iterasi 2 dirancang dalam tujuh Wireframe yang menggambarkan visualisasi interaksi *chatbot* dan komponen Manajemen Dokumen baru.

5.5.1.6.1 Tampilan Pencatatan Presensi

Wireframe Tampilan Pencatatan Presensi menampilkan browser mockup dengan URL “https://aisya.ra-alislam.sch.id/chat” dan judul AISYA-RA. Panel *chat* kanan menampilkan percakapan di mana pengguna menginput presensi harian ("Semua siswa hari ini hadir"), dan *chatbot* mengonfirmasi pencatatan kehadiran 13 siswa kelompok A. *Sidebar* kiri memuat logo, tombol "+ Chat Baru", list riwayat, dan profil *badeg* pengguna.

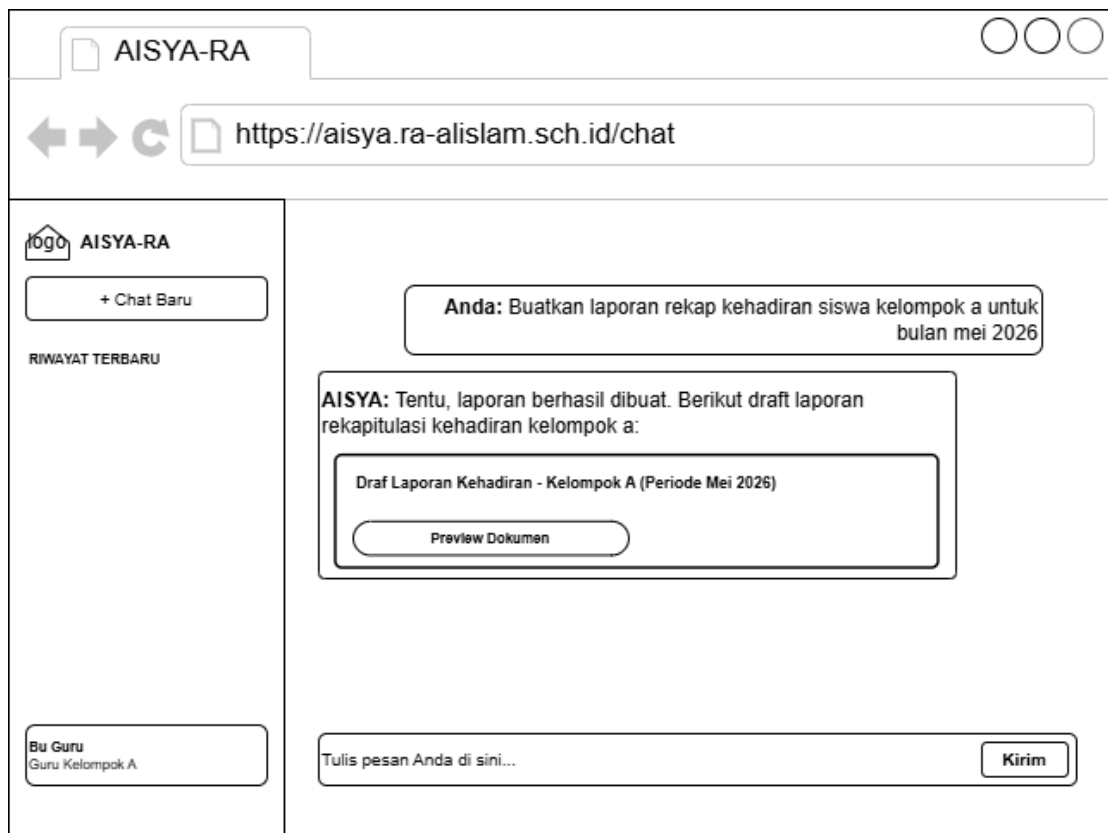


Gambar 5.35 Wireframe Antarmuka Iterasi 2 – Tampilan Pencatatan Presensi

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.1.6.2 Tampilan Rekap Presensi

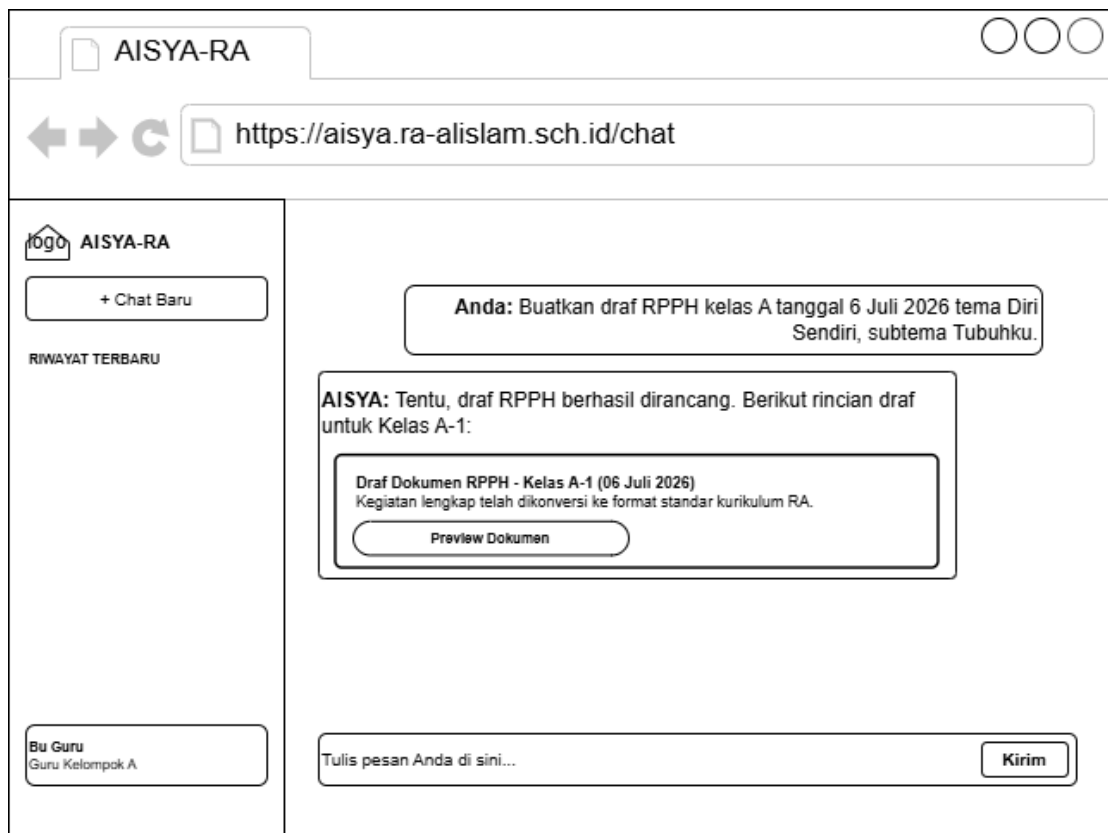
Wireframe Rekap Presensi menampilkan panel *chat* obrolan presensi. *Chatbot* menyajikan tabel ringkasan kehadiran siswa (hadir, sakit, izin, alpa) serta menampilkan tombol biru berlabel "Preview Dokumen" untuk memicu **AISYA Canvas Modal**.



Gambar 5.36 Wireframe Antarmuka Iterasi 2 - Tampilan Rekap Presensi

5.5.1.6.3 Tampilan Otomasi RPPH

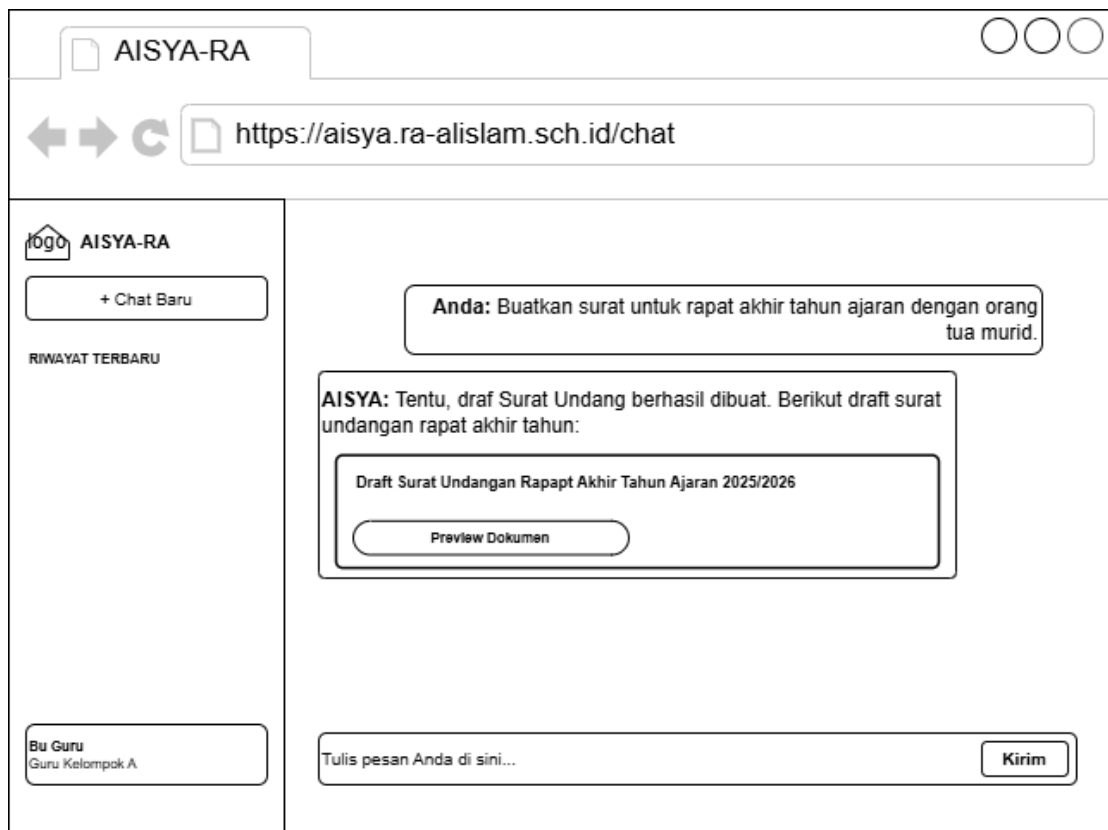
Wireframe Otomasi RPPH menampilkan *chat* percakapan penyusunan rencana harian. Pengguna memasukkan tema dan tanggal, lalu *chatbot* merespons dengan pesan sukses rancangan RPPH beserta kartu pratinjau dokumen yang memuat tombol "Preview Dokumen".



*Gambar 5.37 Wireframe Antarmuka Iterasi 2 – Tampilan Otomasi RPPH
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.6.4 Tampilan Pembuatan Surat Resmi

Wireframe Pembuatan Surat menampilkan *chat* pembuatan surat resmi. Pengguna menginstruksikan pembuatan surat undangan, lalu *chatbot* merespons dengan draf judul surat keluar dan tombol "Preview Dokumen".



*Gambar 5.38 Wireframe Antarmuka Iterasi 2 – Tampilan Pembuatan Surat Resmi
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.6.5 Tampilan Pop Up Menu

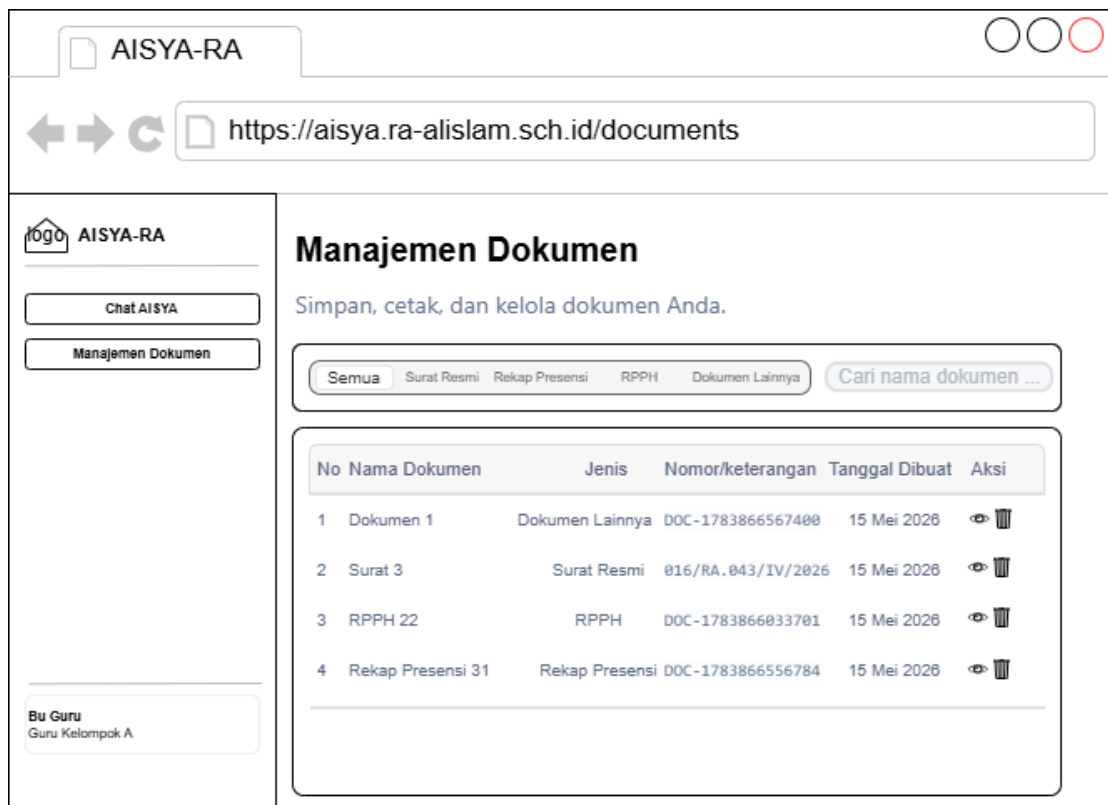
Wireframe Tampilan Pop Up Menu menampilkan antarmuka menu *pop-up* yang meluncur ke atas ketika kartu profil pengguna di bagian bawah *sidebar* diklik, menyajikan pilihan menu "Manajemen Dokumen" dan "Keluar".



*Gambar 5.39 Wireframe Antarmuka Iterasi 2 – Tampilan Pop Up Menu
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.6.6 Tampilan Manajemen Dokumen

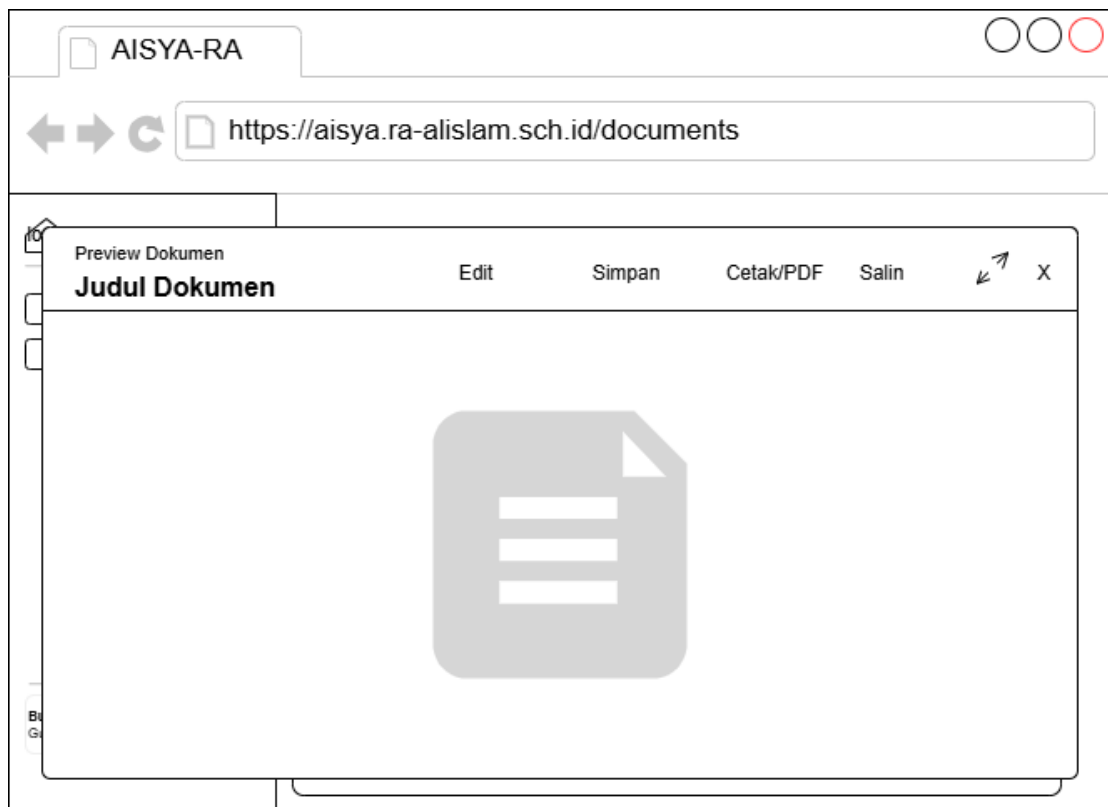
Wireframe Manajemen Dokumen (Riwayat Dokumen) menampilkan halaman pengarsipan dengan URL “<https://aisya.ra-alislam.sch.id/documents>”. Panel utama kanan menyajikan judul “Manajemen Dokumen” dengan Deskripsi “*Simpan, cetak, dan kelola dokumen Anda.*” Di bawahnya terdapat kolom pencarian dokumen dan tab filter kategori (“Semua”, “Surat Resmi”, “Rekap Presensi”, “RPPH” dan “Dokumen Lainnya”). Area tengah menampilkan tabel data arsip dokumen yang berisi No, Nama Dokumen, Jenis, Nomor/Keterangan, Tanggal Dibuat, dan tombol Aksi.



*Gambar 5.40 Wireframe Antarmuka Iterasi 2 – Tampilan Manajemn Dokumen
(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)*

5.5.1.6.7 Tampilan AISYA Canvas Modal (Preview Dokumen)

Wireframe tampilan AISYA Canvas Modal (Preview Dokumen) menampilkan tampilan *overlay* AISYA Canvas Modal di atas halaman Manajemen Dokumen. *Header* modal memuat judul dokumen dan *toolbar* tombol aksi (Edit, Simpan, Cetak/PDF, Salin, Close). Bagian tengah modal menampilkan lembar kertas dokumen A4.



Gambar 5.41 Wireframe Antarmuka Iterasi 2 – AISYA Canvas Modal (Preview Dokumen)

(Sumber: Hasil rancangan tahap design iterasi 2 AISYA-RA, 2026)

5.5.2 Develop Iterasi 2

Tahap *Develop* Iterasi 2 merealisasikan otomatisasi dokumen administrasi sekolah inti, modul presensi siswa, dan lembar kerja penyuntingan interaktif *AISYA Canvas Modal*.

Pada Iterasi 2 ini, sistem mulai mengelola entitas data relasional yang lebih luas, mencakup data identitas madrasah (sekolah), tahun pelajaran (tahun_ajaran), rombongan belajar (kelompok_belajar), serta daftar peserta didik (siswa). Mengingat fitur Manajemen *Data Master* berbasis antarmuka grafis (*GUI*) bagi peran Admin baru dijadwalkan untuk dibangun pada Iterasi 3, penyediaan data master awal untuk keperluan pengujian dan operasional harian pada Iterasi 2 dilakukan melalui metode *database seeding* oleh peneliti menggunakan skrip SQL langsung ke basis data PostgreSQL Supabase. Melalui penyuntikan data awal

(*seeding*) ini, seluruh skenario pencatatan presensi siswa per rombel, pembuatan RPPH kelas, serta penomoran surat keluar dapat dijalankan dan divalidasi secara langsung oleh para guru dan Kepala RA tanpa terhambat oleh ketiadaan data master.

5.5.2.1 Implementasi Otomasi RPPH via *Chat*

Pada Iterasi 2, proses penyusunan Rencana Pelaksanaan Pembelajaran Harian (RPPH) diotomatisasi melalui antarmuka percakapan (*Conversational User Interface*). Guru tidak lagi menyusun dokumen pembelajaran secara manual dari awal, melainkan cukup berinteraksi dengan asisten AI AISYA-RA melalui perintah berbasis teks alami (*chat prompt*). Instruksi tersebut memuat parameter esensial pembelajaran seperti tanggal pelaksanaan, kelompok rombel belajar yang diampu, tema, dan subtema kegiatan.

Sistem backend *chat.py* memproses instruksi tersebut dengan menginjeksi konteks tahun ajaran aktif serta profil kelompok belajar, lalu mengorkestrasi pemanggilan ke Google Gemini API. Mesin AI menghasilkan draf dokumen RPPH terstruktur yang secara otomatis mematuhi format baku 7 komponen kurikulum RA (Identitas Program/Tema, Capaian Pembelajaran, Materi Ajar, Kegiatan Pembukaan, Kegiatan Inti, Kegiatan Penutup, dan Asesmen/Penilaian). Keluaran percakapan ini kemudian memunculkan kartu pratinjau interaktif (*preview card*) dengan tombol aksi "Preview Dokumen" di antarmuka obrolan.

Saat tombol pratinjau ditekan, draf RPPH langsung dimuat ke dalam lembar kerja visual AISYA Canvas Modal (*CanvasPanel.tsx*). Pada antarmuka modal ini, guru dapat meninjau tata letak dokumen siap cetak, melakukan penyesuaian atau penyuntingan langsung (*live edit*), dan mencetak dokumen ke format PDF. Setelah isi dokumen disetujui, penekanan tombol simpan pada antarmuka canvas akan memicu pengiriman data ke endpoint `POST /api/rpph` yang ditangani oleh fungsi `create_rpph` pada berkas `backend/app/routers/rpph.py` untuk disimpan ke basis data Supabase sebagaimana disajikan pada Kode Program 5.10.


```

# backend/app/routers/rpph.py (Handler Pembuatan dan Penyimpanan RPPH)
@router.post("/rpph")
def create_rpph(
    data: RPPHCreate,
    user: Dict[str, Any] = Depends(get_current_user),
    ra_id: str = Depends(get_current_ra_id)
):
    kel_id = data.kelompok_id
    try:
        # Cari ID tahun ajaran yang sedang aktif
        years_res =
supabase.table('tahun_ajaran').select('id').eq('is_active',
True).limit(1).execute()
        active_year_id = years_res.data[0]['id'] if years_res.data else
None
    except Exception:
        active_year_id = None
    if user.get("role") == "guru":
        if not kel_id:
            assigned =
supabase.table('kelompok_belajar').select('id').eq('wali_kelas_id',
user['id']).limit(1).execute()
            if assigned.data:
                kel_id = assigned.data[0]['id']
        else:
            assigned =
supabase.table('kelompok_belajar').select('id').eq('wali_kelas_id',
user['id']).eq('id', kel_id).execute()
            if not assigned.data:
                raise HTTPException(status_code=403, detail="Tidak
memiliki hak akses membuat RPPH untuk kelompok ini.")

    insert_data = {
        "guru_id": user['id'],
        "kelompok_id": kel_id,
        "tanggal": data.tanggal,
        "tema": data.tema,
        "subtema": data.sub_tema,
        "tahun_ajaran_id": active_year_id,
        "konten_json": {
            "materi": data.materi,
            "kegiatan_pembukaan": data.kegiatan_pembukaan,
            "kegiatan_inti": data.kegiatan_inti,
            "kegiatan_penutup": data.kegiatan_penutup,

```

```

        "penilaian": data.penilaian
    }

}

res = supabase.table("rpph").insert(insert_data).execute()
if not res.data:
    raise HTTPException(status_code=400, detail="Gagal menyimpan
dokumen RPPH.")
return {"data": res.data[0], "message": "RPPH berhasil disimpan ke
basis data"}

```

Kode Program 5.10 rpph.py - Handler Pembuatan dan Penyimpanan RPPH ke Database

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

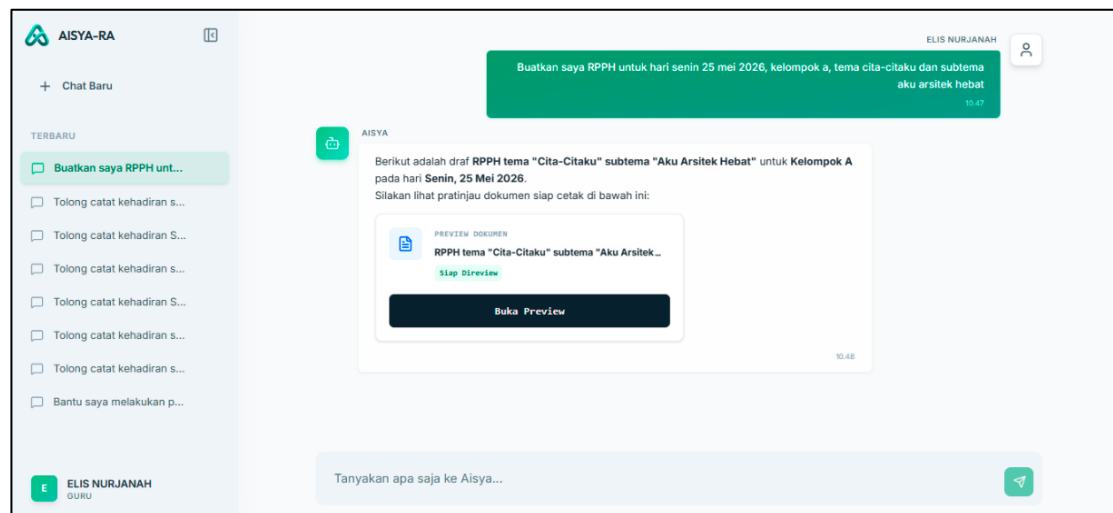
Kode Program 5.10 menunjukkan alur kerja fungsi `create_rpph` yang menangani permintaan HTTP POST `/api/rpph` untuk mengamankan dan mengarsipkan hasil generasi RPPH ke basis data Supabase. Dalam pengembangan Iterasi 2, implementasi ini mengintegrasikan mekanisme penjaminan integritas data sebagai berikut:

1. **Penentuan Periode Akademik Aktif:** Fungsi mengeksekusi kueri pada tabel `tahun_ajaran` (`is_active = True`) sehingga setiap dokumen RPPH yang diterbitkan secara otomatis terasosiasi dengan tahun ajaran berjalan tanpa membebani guru dengan input manual.
2. **Validasi Otoritas Pengampu Kelas (*Role-Based Access Control*):** Bagi pengguna dengan peran Guru RA, sistem memverifikasi relasi penugasan wali kelas (`wali_kelas_id`) pada tabel `kelompok_belajar`. Hal ini menjamin bahwa guru hanya dapat membuat dan menyimpan dokumen RPPH bagi kelompok belajar yang diampunya, menjaga isolasi hak akses data antarwali kelas.
3. **Penyimpanan Terstruktur Berbasis JSONB:** Seluruh butir rincian kegiatan pembelajaran (materi ajar, kegiatan pembukaan, kegiatan inti, kegiatan penutup, dan indikator penilaian) yang telah diformulasikan oleh AI dan disesuaikan oleh guru dienkapsulasi ke dalam kolom semiterstruktur `konten_json` bertipe JSONB di PostgreSQL Supabase. Skema ini

memberikan fleksibilitas tinggi terhadap variasi struktur kegiatan kurikulum RA tanpa memerlukan perombakan skema tabel relasional utama, sekaligus mempermudah pemanggilan kembali dokumen pada antarmuka Canvas Modal maupun modul arsip dokumen madrasah.

5.5.2.2 Hasil Implementasi Pembuatan RPPH via *Chat*

Hasil implementasi otomasi RPPH melalui antarmuka *chatbot* menampilkan percakapan *chatbot* di mana pengguna menginputkan parameter tema, tanggal, dan kelompok belajar, lalu sistem memberikan respons sukses beserta kartu draf dokumen RPPH yang dilengkapi tombol pratinjau.



Gambar 5.42 Tampilan Percakapan Pembuatan RPPH

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

5.5.2.3 Implementasi Pencatatan dan Rekap Presensi Siswa

Modul presensi melayani pencatatan harian via percakapan bahasa alami (*Tool Calling*) serta kalkulasi rekapitulasi kehadiran periodik per kelompok belajar.

```
# backend/app/routers/presensi.py (Handler Rekapitulasi Presensi Siswa)
@router.get("/presensi/rekap")
def get_attendance_recap(
    bulan: Optional[int] = None,
    tahun: Optional[int] = None,
    user: Dict[str, Any] = Depends(get_current_user),
    ra_id: str = Depends(get_current_ra_id)
)
```

```

):
    try:
        active_year = supabase.table('tahun_ajaran').select('id').eq('is_active',
True).limit(1).execute()
        active_year_id = active_year.data[0]['id'] if active_year.data else None

        query = supabase.table('presensi').select(
            'status, tanggal, siswa_id, siswa!inner(id, nama, nis, tingkat_rombel,
ra_id, kelompok_id, kelompok_belajar(wali_kelas_id))'
        ).eq('siswa.ra_id', ra_id)

        if active_year_id:
            query = query.eq('siswa.tahun_ajaran_id', active_year_id)
        if user.get("role") == "guru":
            query = query.eq('siswa.kelompok_belajar.wali_kelas_id', user['id'])

        response = query.execute()

        # Agregasi data kehadiran per siswa (Hadir, Sakit, Izin, Alpha)
        agg = {}
        for row in response.data:
            s_id = row['siswa_id']
            if s_id not in agg:
                s_data = row.get('siswa') or {}
                agg[s_id] = {
                    "id": s_id,
                    "nama": s_data.get('nama', 'Unknown'),
                    "hadir": 0, "sakit": 0, "izin": 0, "alpha": 0, "total": 0
                }
            status = row.get('status', 'alpha').lower()
            if status in agg[s_id]:
                agg[s_id][status] += 1
            agg[s_id]['total'] += 1

        return list(agg.values())
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

```

Kode Program 5.11 presensi.py - Agregasi Data Kehadiran untuk Rekapitulasi Presensi

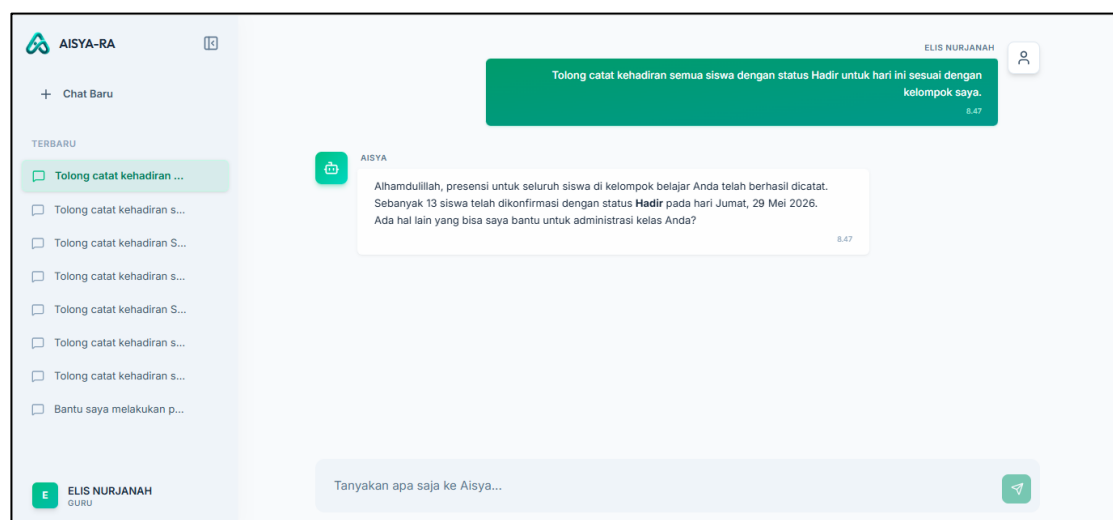
(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

Kode Program 5.11 memperlihatkan algoritma agregasi kehadiran siswa pada fungsi `get_attendance_recap`. Kueri mengeksekusi operasi relasional *INNER JOIN* antara tabel `presensi`, `siswa`, dan `kelompok_belajar` dengan memfilter

parameter kepemilikan lembaga `ra_id` serta pembatasan hak akses pengampu kelas bagi Guru RA. Data mentah kehadiran harian kemudian diagregasi secara dinamis ke dalam objek kamus (*dictionary agg*), yang menghitung akumulasi status kehadiran (Hadir, Sakit, Izin, dan Alpha) per individu siswa dalam satu rentang periode tertentu, menghasilkan luaran siap saji untuk tabel rekapitulasi pada lembar kerja canvas.

5.5.2.4 Hasil Implementasi Percakapan Pencatatan Presensi Siswa

Hasil implementasi pencatatan presensi siswa melalui antarmuka *chatbot* menampilkan percakapan di mana guru menginput kehadiran harian siswa secara cepat melalui teks percakapan biasa (*natural language*), dan *chatbot* secara otomatis memberikan respons konfirmasi jumlah siswa yang dicatat kehadirannya ke dalam *database*.



Gambar 5.43 Tampilan Percakapan Pencatatan Presensi Siswa
(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

5.5.2.5 Implementasi Manajemen Surat Resmi

Modul surat keluar mengotomatisasi penomoran surat resmi secara atomik berbasis format kode klasifikasi madrasah dan tahun berjalan.

```

# backend/app/routers/surat.py (bagian POST /surat)
@router.post("/surat")
def create_surat(data: SuratCreate, ra_id: str = Depends(get_current_ra_id)):
    try:
        nomor = data.nomor_surat
        konten_final = data.konten_final

        if not nomor or nomor.startswith("DOC-") or nomor.startswith("UNNUMBERED-") or data.auto_number_pattern:
            if is_surat_resmi(data.judul, data.konten_final):
                nomor = generate_nomor_surat_helper(ra_id, data.judul, data.konten_final)
                if data.nomor_surat:
                    konten_final = konten_final.replace(data.nomor_surat, nomor)
                    konten_final = konten_final.replace("{{nomor_surat}}", nomor)
                    konten_final = konten_final.replace("{{ nomor_surat }}", nomor)
                else:
                    nomor = None
                    if data.nomor_surat:
                        konten_final = konten_final.replace(data.nomor_surat, "-")
                        konten_final = konten_final.replace("{{nomor_surat}}", "-")

            insert_data = {
                'ra_id': ra_id,
                'judul': data.judul,
                'konten_final': konten_final,
                'nomor_surat': nomor
            }

            response = supabase.table('surat_keluar').insert([insert_data]).execute()
            return response.data[0] if response.data else None
        except Exception as e:
            raise HTTPException(status_code=500, detail=str(e))

```

Kode Program 5.12 surat.py - Otomatisasi Penomoran Surat Resmi dan Penyimpanan ke Basis Data

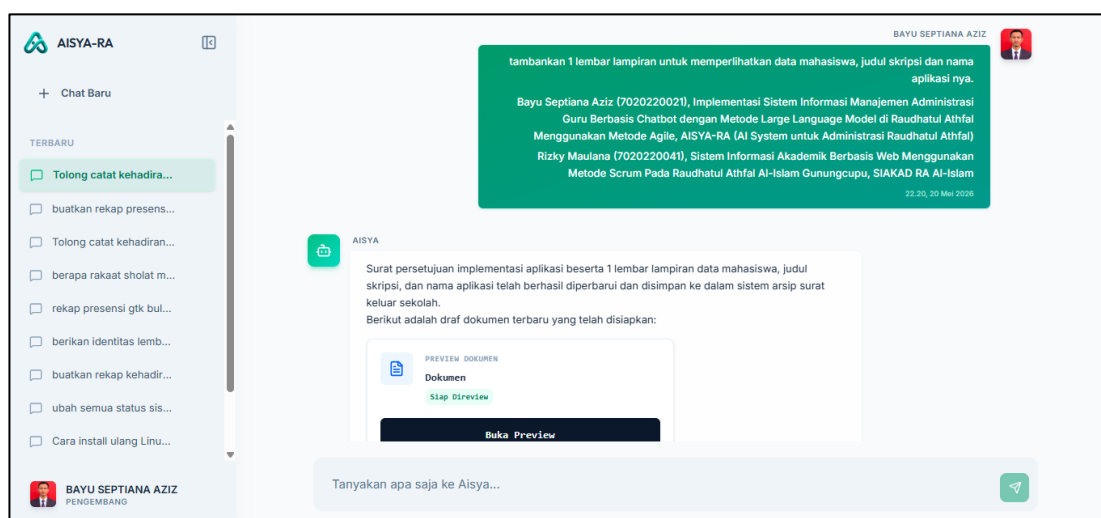
(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

Kode Program 5.12 mengimplementasikan fungsi `create_surat` yang mengotomatisasi penomoran surat dinas lembaga. Jika draf surat yang dikirimkan

belum memiliki nomor resmi definitif (misalnya masih berupa penanda sementara DOC- atau templat tag `{{nomor_surat}}`), sistem memanggil fungsi pembantu `generate_nomor_surat_helper()`. Fungsi ini secara atomik menaikkan nilai penghitung urutan (atomic counter) dan memformat string nomor baku sesuai pola administrasi madrasah (misalnya RA.043/SPM/15/V/2026), lalu menggantikan tag penanda pada badan surat sebelum seluruh rekam surat disimpan ke tabel `surat_keluar`.

5.5.2.6 Hasil Implementasi Pembuatan Surat Keluar via *Chatbot*

Hasil implementasi pembuatan surat resmi keluar melalui antarmuka chatbot menampilkan interaksi chatbot saat menerima instruksi penyusunan draf surat undangan resmi, serta menyajikan kartu draf surat yang terbit nomor suratnya secara berurutan beserta tombol pratinjau.



Gambar 5.44 Tampilan Percakapan Pembuatan Surat Keluar

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

5.5.2.7 Implementasi *Canvas Modal* (Dokumen Editor) dan Manajemn Dokumen
Komponen `CanvasPanel.tsx` dirancang untuk menyajikan pratinjau lembar kerja, fasilitas penyuntingan teks langsung (*contentEditable*), tombol aksi penyimpanan reaktif, dan pemanggilan dialog cetak peramban.

```
// frontend/src/components/CanvasPanel.tsx (Komponen Rich Editor &
Pratinjau Dokumen)

export function CanvasPanel({ isOpen, onClose, activeCanvas, onSave }:
CanvasPanelProps) {
  const [isEditing, setIsEditing] = useState(false);
  const [editedContent, setEditedContent] = useState(activeCanvas?.content
|| '');
  const editorRef = useRef<HTMLDivElement>(null);

  const handleToggleEdit = () => {
    if (isEditing && editorRef.current) {
      setEditedContent(editorRef.current.innerHTML);
    }
    setIsEditing(!isEditing);
  };

  const handleSaveClick = async () => {
    const finalContent = isEditing && editorRef.current ?
editorRef.current.innerHTML : editedContent;
    await onSave(finalContent);
  };

  const handlePrint = () => {
    window.print();
  };

  if (!isOpen) return null;

  return (
    <div className="fixed inset-0 z-50 flex items-center justify-center bg-
slate-900/60 backdrop-blur-sm p-4 print:p-0">
      <div className="bg-white w-full max-w-5xl h-[90vh] rounded-2xl
shadow-2xl flex flex-col overflow-hidden print:h-auto print:shadow-none">
        <div className="px-6 py-4 border-b border-slate-100 flex items-
center justify-between bg-slate-50/50 print:hidden">
          <h3 className="font-bold text-slate-800 text-
base">{activeCanvas?.title}</h3>
          <div className="flex items-center gap-2">
            <button onClick={handleToggleEdit} className="p-2 border
border-slate-200 rounded-lg text-xs font-semibold text-slate-600 hover:bg-
slate-50 flex items-center gap-1.5">
              {isEditing ? <Eye size={14} /> : <Pencil size={14} />}
              <span>{isEditing ? "Pratinjau" : "Edit"}</span>
            </button>
          </div>
        </div>
      </div>
    </div>
  );
}
```



```

        <button onClick={handleSaveClick} className="p-2 bg-emerald-600
text-white rounded-lg text-xs font-semibold hover:bg-emerald-700 flex
items-center gap-1.5">
            <Save size={14} />
            <span>Simpan</span>
        </button>

        <button onClick={handlePrint} className="p-2 bg-indigo-600
text-white rounded-lg text-xs font-semibold hover:bg-indigo-700 flex items-
center gap-1.5">
            <Printer size={14} />
            <span>Cetak PDF</span>
        </button>

        <button onClick={onClose} className="p-2 text-slate-400
hover:text-slate-600">
            <X size={18} />
        </button>
    </div>
</div>

    { /* Lembar Kerja Dokumen */ }

    <div className="flex-1 overflow-y-auto p-8 bg-slate-50/50 flex
flex-col items-center print:bg-white print:p-0">
        {isEditing ? (
            <div className="bg-white rounded-2xl border border-slate-200 p-
8 max-w-4xl w-full min-h-[400px]">
                <div ref={editorRef} contentEditable="true"
dangerouslySetInnerHTML={{ __html: editedContent }} className="prose max-w-
none text-slate-800 focus:outline-none min-h-[350px] p-4 border border-
dashed border-slate-200 rounded-xl" />
            </div>
        ) : (
            <div className="bg-white rounded-2xl border border-slate-100 p-
8 max-w-4xl w-full shadow-sm print:border-none print:shadow-none">
                <div className="prose max-w-none text-slate-800"
dangerouslySetInnerHTML={{ __html: editedContent }} />
            </div>
        )}
    </div>
</div>
</div>
);
}

```

Kode Program 5.13 Kode Program 5.13 CanvasPanel.tsx - Komponen Visual Rich Document Editor dan Pratinjau Dokumen

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

Kode Program 5.13 memperlihatkan arsitektur komponen CanvasPanel yang menjadi jembatan visual antara hasil inferensi asisten AI dan kebutuhan cetak dokumen formal:

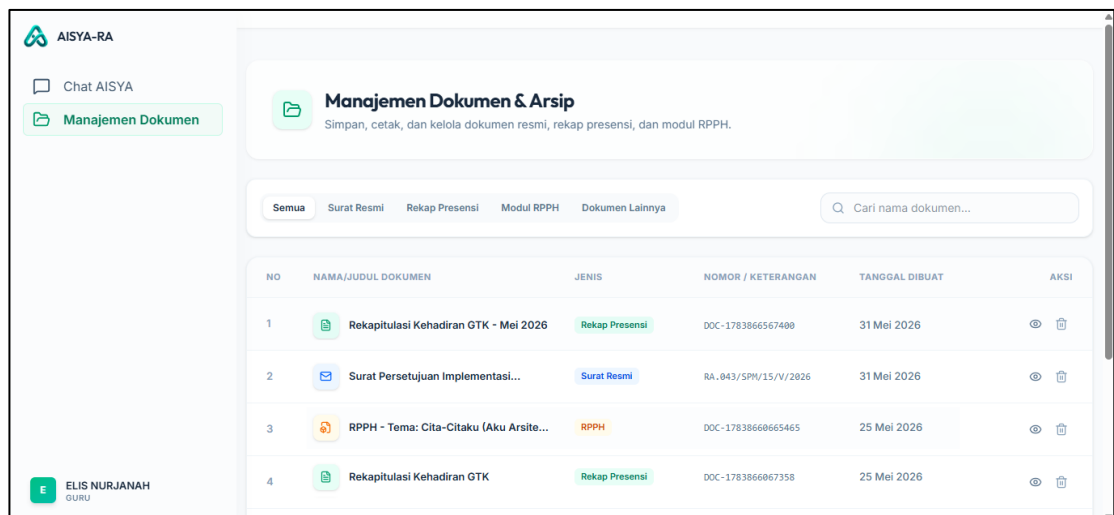
1. Komponen mengelola *state* peralihan antara mode tampilan pratinjau (*Preview*) dan mode penyuntingan langsung (*Edit*) melalui *state* *isEditing*. Pada mode edit, atribut *contentEditable="true"* diaktifkan pada elemen referensi *editorRef* sehingga guru dapat langsung mengetik atau memformat teks kegiatan secara bebas di peramban.
2. Tombol simpan memicu fungsi *handleSaveClick* yang mengambil isi HTML termutakhir dan mengeksekusi *callback* *onSave* untuk mengirimkannya ke *backend router*.
3. Tombol "Cetak PDF" memicu fungsi *handlePrint()* yang mengeksekusi *window.print()*. Dengan aturan gaya CSS media cetak (*print:hidden* pada bilah navigasi dan *print:p-0 print:border-none* pada wadah dokumen), peramban secara otomatis menghasilkan tata letak lembar dokumen A4 yang bersih dan rapi tanpa menyertakan elemen antarmuka situs.

5.5.2.8 Hasil Implementasi Halaman Manajemen Dokumen dan Canvas Modal

Hasil implementasi antarmuka pengarsipan dan penyuntingan dokumen mencakup:

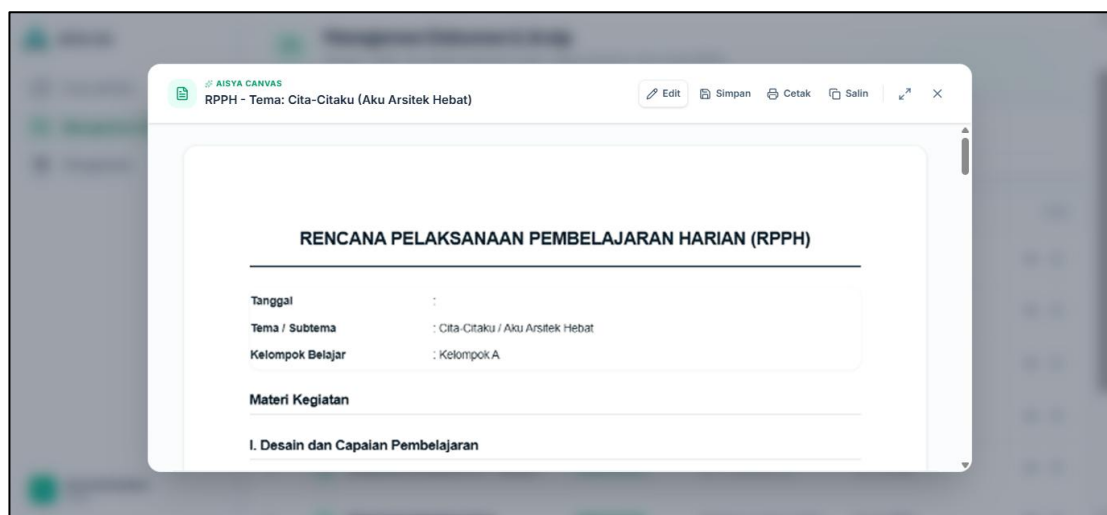
1. **Halaman Manajemen Dokumen (Documents.tsx):** Menyajikan tabel arsip seluruh dokumen sekolah (RPPH, Rekap Presensi, Surat Keluar) dengan filter kategori, kolom pencarian, penanda nomor dokumen, tanggal penerbitan, dan tombol aksi pratinjau.
2. **Editor Interaktif AISYA Canvas Modal (CanvasPanel.tsx):** Lembar kerja visual yang muncul meluncur di atas layar obrolan (*modal overlay*), memungkinkan guru meninjau tata letak dokumen formal, melakukan

koreksi langsung (*live edit*), menyimpan perubahan ke database, dan mencetak dokumen ke format PDF.



Gambar 5.45 Tampilan Halaman Manajemen Dokumen

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)



Gambar 5.46 Tampilan AISYA Canvas Madal (Dokumen Editor)

(Sumber: Hasil pengembangan tahap develop iterasi 2 AISYA-RA, 2026)

5.5.3 Test Iterasi 2 – Black Box Testing

Pengujian pada Iterasi 2 dilaksanakan untuk memvalidasi fungsionalitas modul administrasi inti yang baru dikembangkan, mencakup otomatisasi penyusunan RPPH

7 komponen, pencatatan dan rekapitulasi presensi siswa, pembuatan surat keluar berpenomoran atomik, interaktivitas lembar kerja AISYA Canvas Modal, serta pencetakan dokumen PDF. Selain menguji fitur-fitur baru, pengujian ini juga bertujuan untuk memastikan tidak terjadinya penurunan kinerja atau kerusakan fungsi (*regression testing*) pada modul autentikasi dan manajemen obrolan yang telah dibangun pada Iterasi 1.

Pengujian fungsionalitas dilakukan secara langsung di lingkungan operasional madrasah melalui jaringan Wi-Fi lokal (*Local Area Network*) 192.168.1.105 menggunakan perangkat laptop pengembang, laptop guru, dan *smartphone* pengajar (Google Chrome Mobile) sebagai *test client*. Seluruh skenario diuji menggunakan akun aktual pendidik yang terdaftar di basis data, yaitu akun Guru Kelompok A (R-02), Guru Kelompok B (R-03), Guru Pendamping (R-04), dan Kepala RA (R-01).

Hasil pengujian disajikan pada Tabel 5.12 berikut.

Tabel 5.12 Hasil Black Box Testing Iterasi 2

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
TB-10	Membuat draf RPPH via chat	<i>Chatbot</i>	Draf RPPH berhasil terbentuk lengkap	Lulus
		draf RPPH lengkap dengan format 7 asesmen. komponen baku.	dengan tema, KBC, langkah kegiatan, dan	
TB-11	Membuka draf RPPH di Canvas	Menekan tombol Preview Dokumen membuka draf di	Dokumen RPPH termuat rapi di dalam modal editor canvas.	Lulus

		dalam lembar kerja canvas.	
TB-12	Menyunting draf RPPH di Canvas	Pengguna dapat mengubah teks kegiatan langsung di antarmuka canvas.	Teks kegiatan berhasil diedit dan tersimpan di memori editor. Lulus
TB-13	Menyimpan RPPH ke basis data	Menekan tombol simpan menyimpan dokumen ke tabel rpph di Supabase.	Dokumen tersimpan di database dan status berubah menjadi terarsip. Lulus
TB-14	Mencatat presensi siswa via chat	<i>Chatbot</i> mengeksrak data absensi dan menyimpannya ke tabel presensi.	Data absensi siswa berhasil tersimpan dan pesan konfirmasi muncul di chat. Lulus
TB-15	Membuat rekap presensi via chat	<i>Chatbot</i> mengagregasi data absensi bulanan ke dalam tabel rekapitulasi.	Tabel rekap presensi periodik berhasil dibangkitkan dan dapat dibuka di canvas. Lulus
TB-16	Membuat surat resmi via chat	<i>Chatbot</i> menyusun surat keluar lengkap dengan kop dinas	Draf surat dinas terbentuk dengan nomor surat unik berurutan. Lulus

		dan nomor urut atomik.		
TB-17	Mencetak dokumen ke format PDF	Menekan tombol printer membuka dialog cetak peramban dengan format dokumen rapi.	Dokumen tercetak dalam tata letak kertas standar tanpa elemen navigasi website.	Lulus
TB-18	Akses riwayat di Manajemen Dokumen	Pengguna dapat menelusuri seluruh arsip RPPH, presensi, dan surat keluar.	Seluruh arsip dokumen tampil dalam tabel dan dapat dibuka kembali di canvas.	Lul

(Sumber: Hasil pengujian Black Box Testing iterasi 2 AISYA-RA, 2026)

Seluruh sembilan skenario pengujian dinyatakan lulus. Seluruh target kriteria *Definition of Done* Iterasi 2 sudah terpenuhi.

5.5.4 Deploy Iterasi 2

Tahap *Deploy* pada Iterasi 2 difokuskan pada peluncuran pembaruan perangkat lunak (*software increment update*) yang mengintegrasikan modul administrasi inti (otomasi RPPH, presensi siswa, penomoran surat keluar, dan antarmuka *AISYA Canvas Modal*) ke lingkungan pengujian operasional madrasah.

Prosedur peluncuran Iterasi 2 mencakup serangkaian langkah teknis sebagai berikut:

1. **Pembaruan Basis Kode (*Codebase Integration*):** Seluruh modul baru pada direktori backend/app/routers/ (rpqh.py, presensi.py, surat.py) dan komponen antarmuka frontend/src/ (CanvasPanel.tsx, Documents.tsx) disatukan ke dalam lingkungan kerja peladen lokal.

2. **Pembangunan Ulang Kontainer (*Container Re-building*):** Eksekusi perintah docker compose down dilakukan untuk menghentikan instans kontainer Iterasi 1 secara terkontrol, diikuti perintah docker compose up --build -d untuk mengompilasi ulang aset statis TypeScript/Vite pada *stage builder* dan memuat dependensi terbaru backend.
3. **Simulasi Akses Operasional Simultan (*Concurrent User Testing*):** Sistem diuji coba secara serempak oleh Kepala RA dan tiga orang guru kelas di ruang kantor RA Al-Islam Gunungcupu melalui alamat IP <http://192.168.1.105>. Setiap guru mengeksekusi instruksi pencatatan presensi kelompok belajar masing-masing dan meminta asisten AI menyusun draf RPPH secara bersamaan.
4. **Pemantauan Kinerja dan Utilisasi Sumber Daya:** Melalui perintah pemantau kontainer docker stats, tercatat bahwa penggunaan memori RAM pada kontainer aisyah_backend stabil pada kisaran 85–110 MB saat memproses permintaan streaming SSE bersamaan, sedangkan beban komputasi CPU tidak melebihi 15%, membuktikan efisiensi tinggi arsitektur asinkron FastAPI dan Nginx dalam melayani aktivitas administrasi harian madrasah.

5.5.5 Review Iterasi 2

Sesi *review* iterasi ini berlangsung sekitar satu setengah jam dan menjadi momen pertama seluruh pengguna benar-benar meminta AISYA-RA mengerjakan tugas administrasi sungguhan, bukan sekadar sapaan seperti pada Iterasi 1. Pengujian langsung oleh pengguna memunculkan evaluasi kritis dan reaksi yang sangat natural.

R-02 mencoba modul RPPH lebih dulu dengan menyebutkan tema dan kelompok secara singkat. Beliau sempat terkejut ketika draf lengkap tiga kegiatan langsung muncul dalam hitungan detik, namun segera menyadari adanya pekerjaan manual yang tersisa:

“Ini beneran otomatis? Saya kira paling cuma judul sama tanggalnya doang yang keisi, ternyata kegiatan intinya juga langsung ada. Tapi kelompoknya sama namanya saya yang isi manual lagi kan nanti?”

— R-02, Guru Kelompok A, Sesi *Review* Iterasi 2

Pertanyaan kritis dari R-02 tersebut memberikan masukan berharga bahwa draf RPPH yang dibangkitkan LLM saat ini belum sepenuhnya kontekstual karena belum menyertakan *header* identitas guru dan institusi secara otomatis. Umpan balik ini secara langsung memvalidasi urgensi rencana pembangunan Modul *Data Master* terpusat yang dijadwalkan pada Iterasi 3.

Pada modul kehadiran siswa, Guru Pendamping (R-04) mencoba mencatatkan presensi kelas melalui antarmuka percakapan *Chatbot*. Pada awalnya, beliau sempat merasa ragu terkait format penyusunan kalimat perintah:

“Saya ketiknya bener nggak ya? Soalnya kalau di buku kan tinggal saya ceklis aja, ini saya harus nulis kalimat”

— R-04, Guru Pendamping, Sesi *Review* Iterasi 2

Setelah penulis mendemonstrasikan bahwa guru cukup menuliskan nama-nama siswa yang berhalangan hadir saja (sakit/izin) dan membiarkan siswa lainnya secara otomatis tercatat sebagai hadir, R-04 langsung dapat melakukannya dengan lancar pada percobaan berikutnya. Keberadaan fitur rekapitulasi presensi otomatis yang terhitung secara instan dinilai para guru sangat revolusioner, karena menghilangkan beban rutin menghitung manual tanda centang di buku folio yang biasanya menyita waktu hingga 45 menit setiap akhir bulan.

Selanjutnya, Kepala RA (R-01) mengevaluasi modul persuratan dinas dengan meminta pembuatan surat undangan rapat wali murid. Beliau mengapresiasi kerapian struktur surat dan keandalan penomoran surat atomik, namun memberikan catatan manajerial penting terkait kelengkapan formal dokumen:

“Nomor suratnya sudah otomatis berurutan dengan rapi dan isinya juga sudah sesuai format resmi. Tapi ini kop suratnya kok belum ada logo sekolahnya ya? Kalau bisa ditambahkan logo dan nama sekolah biar kelihatan resmi sekali saat dicetak.”

— Kepala RA (R-01), Sesi *Review* Iterasi 2

Melihat kepraktisan dan kecepatan pencatatan presensi siswa melalui *Chatbot*, Kepala RA kembali menyoroti celah administratif yang telah diidentifikasi pada observasi awal penelitian, yaitu ketiadaan rekam jejak kehadiran guru dan staf tenaga kependidikan (GTK). Beliau secara eksplisit meminta agar fitur pencatatan presensi diperluas ke ranah kepegawaian:

“Melihat presensi anak-anak bisa dicatat secepat ini hanya lewat *chat*, saya rasa masalah tidak adanya absen guru yang sempat kita bahas di awal dulu bisa diselesaikan sekaligus di aplikasi ini. Tolong fitur presensinya diperluas, agar saya dan guru-guru juga bisa absen harian langsung lewat AISYA-RA.”

— Kepala RA (R-01), Sesi *Review* Iterasi 2

Permintaan eksplisit ini mengubah status kebutuhan presensi GTK dari sekadar temuan awal menjadi *item backlog* baru berprioritas tinggi. Selain itu, ketika guru mencoba mengajukan pertanyaan mengenai pedoman kurikulum dan jadwal lokal madrasah, *Chatbot* masih memberikan jawaban yang bersifat umum karena ketiadaan dokumen acuan lokal. Hal ini memvalidasi secara empiris kebutuhan integrasi fitur RAG (*Retrieval-Augmented Generation*). Di sisi lain, Kepala RA juga menegaskan kembali harapannya untuk memiliki antarmuka *Dashboard* pemantauan terpusat.

Berdasarkan hasil evaluasi *Review & Retrospective* Iterasi 2, tim pengembang dan mitra menyepakati prioritas penyempurnaan sistem pada Iterasi 3:

1. **Modul Manajemen *Data Master* (Lembaga, Siswa, GTK):** Diperlukan agar sistem dapat menyuntikkan profil madrasah, alamat, nama Kepala RA,

dan guru kelas ke dalam *header* RPPH dan kop surat secara otomatis (Prioritas Tinggi).

2. **Pencatatan Presensi GTK via *Chatbot (Item Backlog Baru)***: Perluasan modul presensi untuk mengakomodasi pencatatan dan rekapitulasi kehadiran harian guru dan staf berdasarkan permintaan langsung Kepala RA (Prioritas Tinggi).
3. **Integrasi RAG (*Knowledge Base*)**: Mengunggah berkas PDF pedoman kurikulum lokal agar *Chatbot* dapat menjawab pertanyaan spesifik sekolah secara presisi (Prioritas Tinggi).
4. **Antarmuka *Dashboard* Eksekutif**: Menyajikan ringkasan visual data kehadiran dan administrasi sekolah secara terpusat khusus bagi Kepala RA (Prioritas Tinggi).

5.6 Iterasi 3: Penguatan Sistem, *Data Master*, RAG, Presensi GTK, *Dashboard* dan *Cloud Deployment*

Iterasi 3 dilaksanakan selama dua setengah minggu sesuai *timebox* yang ditetapkan. Tujuan utama iterasi ini adalah menyempurnakan arsitektur sistem secara menyeluruh guna menjawab kebutuhan yang diidentifikasi pada sesi *Review* Iterasi 2, mencakup pembangunan modul Manajemen *Data Master* terpusat, infrastruktur basis pengetahuan lokal (*Retrieval-Augmented Generation* atau RAG), pencatatan presensi GTK harian, visualisasi ringkasan eksekutif pada *Dashboard*, serta *deployment* operasional penuh ke server produksi berbasis *Cloud VPS* (*Virtual Private Server*).

Item backlog yang dikerjakan pada Iterasi 3 meliputi:

1. Manajemen *Data Master* terpusat oleh aktor Admin (Identitas Lembaga, Kelompok Belajar, Tahun Ajaran, Data Siswa dengan fitur impor massal *spreadsheet*, dan Data GTK/Staff).

2. Pencatatan dan rekapitulasi presensi GTK via *chat* bagi Kepala RA (sebagai *item backlog* adaptif hasil masukan sesi *Review Iterasi 2*).
3. Basis Pengetahuan *Retrieval-Augmented Generation (RAG)* Dokumen Lokal (Admin mengunggah dokumen kurikulum/SOP madrasah, serta Guru dan Kepala RA dapat bertanya regulasi/panduan via percakapan *chat*).
4. *Dashboard* pemantauan statistik eksekutif bagi Kepala RA.
5. *Deployment* operasional penuh ke server produksi VPS DigitalOcean dengan konfigurasi kontainer Docker dan domain resmi <https://aisya.ra-alislam.sch.id>.

Definition of Done (DoD) Iterasi 3 dinyatakan selesai ketika seluruh data master terintegrasi secara otomatis ke dalam instruksi kontekstual LLM, modul RAG berhasil menjawab kueri berdasarkan dokumen lokal madrasah, presensi GTK dan statistik *dashboard* beroperasi akurat, seluruh pengujian *Black Box Testing* dinyatakan lulus (*Passed*), serta sistem dapat diakses secara publik dan stabil melalui internet.

5.6.1 *Design Iterasi 3*

Tahap *Design Iterasi 3* menghasilkan artefak perancangan sistem komprehensif sebagai acuan implementasi akhir.

5.6.1.1 *Use Case Diagram*

Use Case Diagram Iterasi 3 mendefinisikan penambahan fungsionalitas sistem (increment) yang melibatkan tiga aktor utama dengan pembagian peran sebagai berikut:

1. **Guru RA:** Berinteraksi dengan fungsi operasional percakapan harian, meliputi autentikasi (*Login/Logout*), percakapan cerdas berkonteks RAG (“Mengirim Percakapan (Penguatan Konteks RAG)”), pembuatan & riwayat RPPH, pencatatan & rekap presensi siswa, pembuatan & arsip surat keluar, serta fitur presensi GTK baru (“Mencatat Presensi GTK via Chat”,

“Membuat Rekap Presensi GTK via Chat”, “Melihat Riwayat Rekap Presensi GTK”).

2. **Kepala RA:** Mengakses seluruh fitur operasional harian yang dimiliki Guru RA, ditambah wewenang memantau ringkasan statistik sekolah pada “Memantau Dashboard” (yang direalisasikan pada Iterasi 3 sesuai perencanaan tahap Plan).
3. **Admin:** Memiliki wewenang khusus untuk mengelola data sistem dan konfigurasi sekolah melalui antarmuka GUI khusus, meliputi manajemen akun sistem (*Login/Logout*), pengelolaan lima modul *Data Master* (“Kelola Identitas Lembaga”, “Kelola Kelompok Belajar”, “Kelola Tahun Ajaran”, “Kelola Data Siswa”, “Kelola Data GTK/Staff”), dan “Mengelola Dokumen RAG (*Knowledge Base*)”.

Rincian fungsionalitas *Use Case Diagram* Iterasi 3 disajikan pada Tabel 5.13 berikut.

Tabel 5.13 Use Case Diagram Iterasi 3

<i>Use Case</i>	Deskripsi	Aktor Terkait
<i>Login ke Sistem</i>	Autentikasi akun pengguna (Guru, Kepala RA, dan Admin) ke dalam sistem	Guru RA, Kepala RA, Admin
<i>Logout dari Sistem</i>	Mengakhiri sesi autentikasi pengguna dan menghapus token akun	Guru RA, Kepala RA, Admin
Memulai Sesi Chat Baru	Membuat ruang percakapan baru berbasis AI <i>Chatbot</i>	Guru RA, Kepala RA

Mengirim Percakapan (Penguatan Konteks RAG)	Memproses percakapan bahasa alami di <i>Chatbot</i> dengan pencarian semantik RAG (<i>Cosine Similarity</i>) dan injeksi <i>Data Master</i>	Guru RA, Kepala RA
Membuka <i>Chat Sesi Lama</i>	Melanjutkan percakapan dari daftar riwayat percakapan sebelumnya	Guru RA, Kepala RA
Melihat Riwayat <i>Chat</i>	Menampilkan daftar riwayat percakapan (« <i>include</i> » dari <i>Membuka Chat Sesi Lama</i>)	Guru RA, Kepala RA
Menghapus <i>Chat</i>	Menghapus sesi percakapan tertentu (« <i>extend</i> » ke <i>Melihat Riwayat Chat</i>)	Guru RA, Kepala RA
Mengatur Ulang Sandi	Memperbarui kata sandi akun (« <i>extend</i> » ke <i>Login ke Sistem</i>)	Guru RA, Kepala RA
Mengingat sesi <i>Login</i>	Menyimpan sesi autentikasi pengguna (« <i>extend</i> » ke <i>Login ke Sistem</i>)	Guru RA, Kepala RA
Membuat RPPH via <i>Chat</i>	Menginstruksikan <i>Chatbot</i> untuk menyusun draf RPPH berbasis kurikulum	Guru RA, Kepala RA
Melihat Riwayat/ Daftar RPPH	Mengakses daftar dokumen RPPH yang tersimpan di <i>database</i>	Guru RA, Kepala RA
Mencatat Presensi Siswa via <i>Chat</i>	Mencatat kehadiran harian siswa secara alami melalui <i>Chatbot</i>	Guru RA, Kepala RA
Membuat Rekap Presensi Siswa via <i>Chat</i>	Mengagregasi rekapitulasi presensi kehadiran siswa via <i>Chatbot</i>	Guru RA, Kepala RA

Melihat Rekap Siswa	Riwayat Presensi	Mengakses daftar rekapitulasi kehadiran siswa yang tersimpan	Guru RA, Kepala RA
Mencatat GTK via Chat	Presensi	Melakukan pencatatan presensi harian guru & staf melalui <i>Chatbot</i>	Guru RA, Kepala RA
Membuat Presensi GTK via Chat	Rekap Presensi	Meminta laporan rekapitulasi presensi guru & staf via <i>Chatbot</i>	Guru RA, Kepala RA
Melihat Rekap GTK	Riwayat Presensi	Mengakses daftar rekapitulasi kehadiran GTK yang tersimpan	Guru RA, Kepala RA
Membuat Resmi via Chat	Surat	Menginstruksikan <i>Chatbot</i> menyusun draf surat resmi sekolah	Guru RA, Kepala RA
Melihat Arsip Surat Keluar	Surat	Mengakses daftar dokumen surat resmi keluar yang tersimpan	Guru RA, Kepala RA
Buka Canvas (Preview & Aksi)	AISYA Modal	Menampilkan antarmuka visual canvas untuk preview & aksi dokumen	Guru RA, Kepala RA
Mengedit Dokumen	Konten	Mengubah teks draf dokumen pada canvas (generalisasi dari <i>Canvas Modal</i>)	Guru RA, Kepala RA
Menyimpan Dokumen Database	ke	Menyimpan draf dokumen hasil suntingan ke database (<i>generalisasi dari Canvas Modal</i>)	Guru RA, Kepala RA

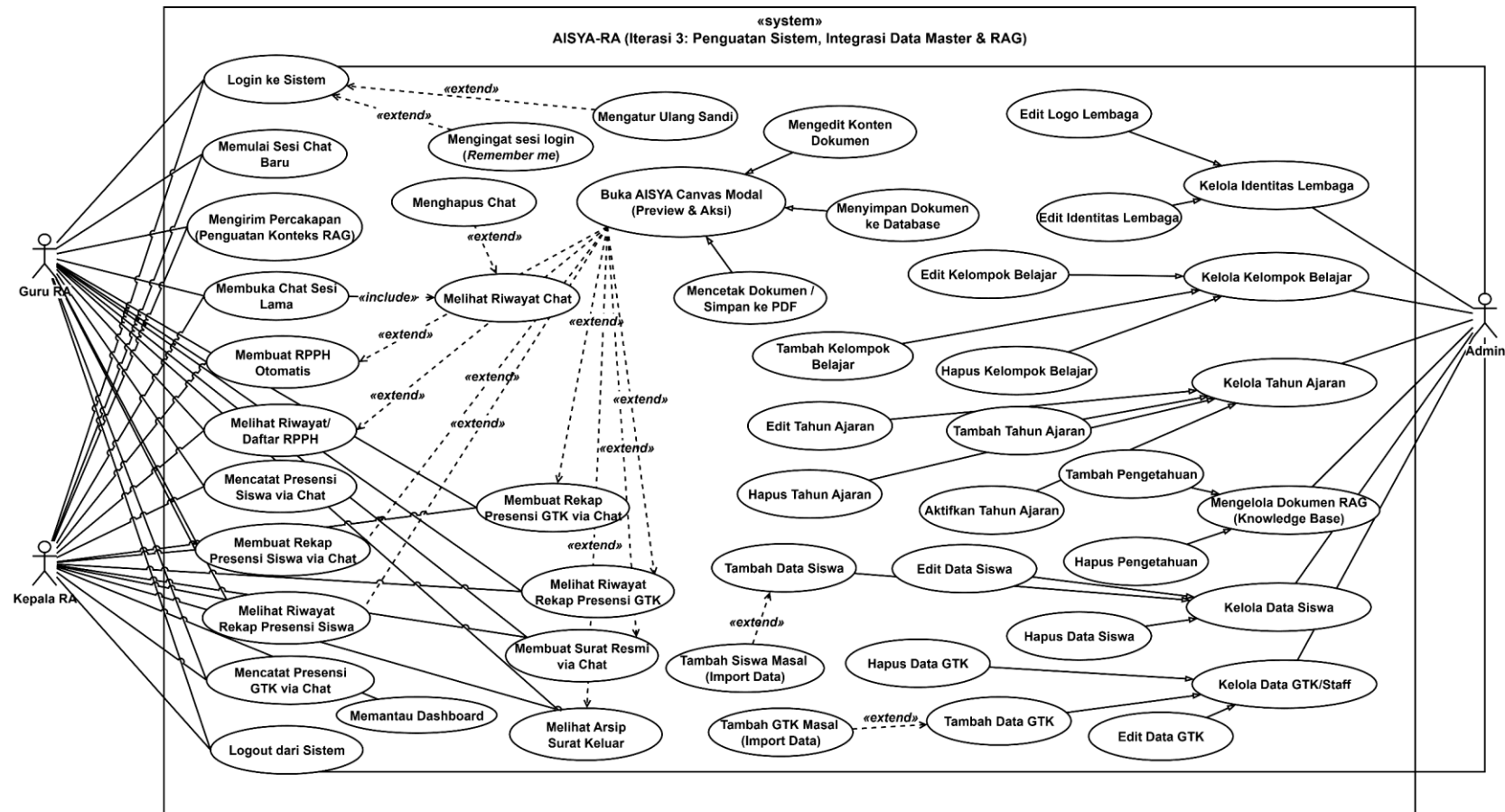
Mencetak Dokumen / Simpan ke PDF	Mencetak atau mengunduh draf dokumen sebagai PDF (<i>generalisasi dari Canvas Modal</i>)	Guru RA, Kepala RA
Memantau Dashboard	Memantau ringkasan eksekutif statistik total siswa, GTK, presensi, tren grafik, aktivitas pengguna dan dokumen yang dibuat.	Kepala RA
Kelola Identitas Lembaga	Modul induk pengelolaan data profil resmi sekolah	Admin
Kelola Kelompok Belajar	Modul induk pengelolaan rombel kelas	Admin
Kelola Tahun Ajaran	Modul induk pengelolaan periode akademik	Admin
Kelola Data Siswa	Modul induk pengelolaan data siswa	Admin
Kelola Data GTK/Staff	Modul induk pengelolaan data guru & staf	Admin
Mengelola Dokumen RAG (<i>Knowledge Base</i>)	Modul induk pengelolaan dokumen pedoman sekolah untuk RAG	Admin

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

Secara arsitektural, lima modul *Data Master* (“Kelola Identitas Lembaga”, “Kelola Kelompok Belajar”, “Kelola Tahun Ajaran”, “Kelola Data Siswa”, dan “Kelola Data GTK/Staff”) bertindak sebagai parent *Use Case* yang digeneralisasi oleh sub-fungsi CRUD (*Create, Read, Update, Delete*) spesifik di dalamnya. Sub-fungsi impor data masal seperti “Tambah Siswa Masal (*Import Data*)” dan “Tambah GTK Masal (*Import Data*)” memperluas («extend») masing-masing fungsi

penambahan data tunggal. Data master yang dikelola oleh aktor Admin ini selanjutnya disuntikkan secara otomatis ke dalam prompt sistem AI untuk memberikan konteks identitas lembaga resmi saat membuat dokumen RPPH, Surat, maupun Rekap Presensi.

Sementara itu, modul “Mengelola Dokumen RAG (*Knowledge Base*)” memberikan wewenang khusus kepada Admin untuk mengunggah dan mengindeks dokumen pedoman sekolah (format PDF) ke basis data vektor Supabase (pgvector) melalui Gemini *Embedding* API. Basis pengetahuan ini dimanfaatkan secara otomatis oleh *Use Case* “Mengirim Percakapan (Penguatan Konteks RAG)” melalui pencarian kemiripan semantik (*Cosine Similarity Search*) ketika kueri pengguna menanyakan aturan kurikulum lokal. Selain itu, seluruh fitur pembuatan dan pemuatan dokumen (RPPH, Presensi Siswa, Presensi GTK, dan Surat Keluar) terhubung melalui mekanisme «*extend*» ke “Buka AISYA Canvas Modal (Preview & Aksi)” untuk proses penyuntingan dan pencetakan PDF secara fleksibel



Gambar 5.47 Use Case Iterasi 3

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

Berbeda dengan cakupan pada tahap *Plan* maupun *Use Case Diagram* Iterasi 2, *Use Case Diagram* Iterasi 3 memuat fungsionalitas komprehensif mencakup pencatatan & rekap presensi GTK via *Chat*, ingesti dokumen basis pengetahuan RAG, pemantauan statistik eksekutif pada *Dashboard*, serta pengelolaan *Data Master* sekolah secara terpusat. *Use Case* “Memantau *Dashboard*” yang pada tahap *Plan* direncanakan untuk Kepala RA, pada Iterasi 3 ini direalisasikan secara penuh menjadi tampilan visual ringkasan eksekutif (statistik total siswa, total GTK, persentase presensi harian, serta grafik tren bulanan, aktivitas pengguna dan laporan dokumen yang dibuat). Selain itu, *Use Case* “Mengelola *Data Master*” yang pada tahap awal (*Plan*) masih berada pada level modul konseptual, pada Iterasi 3 ini dirumuskan secara rinci dan dipecah menjadi lima *parent Use Case Data Master* teknis (“Kelola Identitas Lembaga”, “Kelola Kelompok Belajar”, “Kelola Tahun Ajaran”, “Kelola Data Siswa”, “Kelola Data GTK/Staff”), lengkap dengan sub-fungsi generalisasi operasi CRUD serta fungsi impor masal («*extend*») yang seluruhnya dikelola oleh aktor Admin. Begitu pula modul RAG dirumuskan menjadi *parent Use Case* “Mengelola Dokumen RAG (*Knowledge Base*)” dengan sub-fungsi “Tambah Pengetahuan” dan “Hapus Pengetahuan”.

5.6.1.2 *Activity Diagram*

Pemodelan *Activity Diagram* pada tahap *Design* Iterasi 3 bertujuan untuk memetakan alur kerja penyempurnaan sistem operasional. Modul ini berfokus pada percakapan cerdas berkonteks *Retrieval-Augmented Generation (RAG)*, ingesti basis pengetahuan dokumen pedoman sekolah (PDF), perluasan presensi harian Guru dan Tenaga Kependidikan (GTK), serta *Dashboard* pemantauan eksekutif untuk Kepala RA. Pemodelan ini terbagi dalam lima diagram aktivitas utama yang merincikan alur pertukaran data antar *swimlanes* aktor dan komponen sistem.

5.6.1.2.1 *Activity Diagram* Ingesti (*Data Ingestion Flow*) & Pengindeksan Dokumen RAG (Manajemen Basis Pengetahuan)

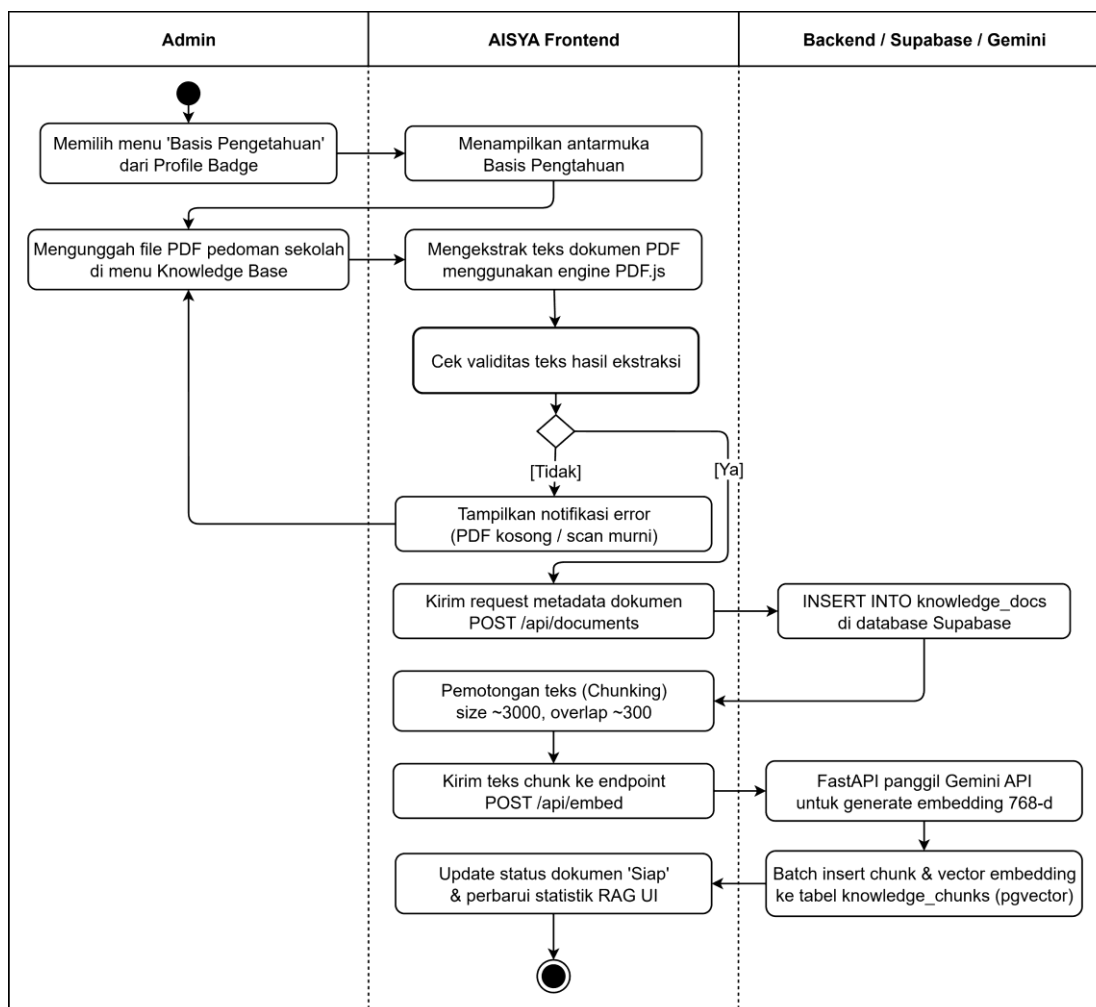
Activity Diagram Ingesti & *Indexing* Dokumen RAG memodelkan alur pemrosesan berkas PDF pedoman sekolah yang diunggah oleh Admin. Diagram ini melibatkan *swimlanes* Admin, AISYA Frontend (**Knowledge.tsx**), dan Backend / Supabase

/ **Gemini** (**knowledge.py**, basis data **Supabase** “**knowledge_docs**” & “**knowledge_chunks**”, serta **Gemini Embedding API**).

Alur kerja dimulai ketika **Pengguna (Admin)** membuka halaman **Basis Pengetahuan (Knowledge.tsx)** dan mengunggah berkas PDF pedoman sekolah. **Frontend** mengekstrak isi teks dokumen PDF secara langsung di peramban menggunakan engine pdf.js. Selanjutnya terdapat titik keputusan validitas teks hasil ekstraksi:

1. Apabila teks hasil ekstraksi tidak valid [**Tidak**] (berkas PDF kosong atau berupa scan gambar murni tanpa lapisan OCR teks), **Frontend** merender notifikasi kesalahan (*error toast*) dan meminta pengguna mengunggah kembali berkas PDF yang valid.
2. Apabila teks hasil ekstraksi valid [**Ya**], **Frontend** mengirimkan permintaan metadata dokumen (“POST /api/documents”) ke **Backend**, dan **Backend** menyimpan *record* pada tabel “**knowledge_docs**”.

Frontend kemudian melakukan pemotongan teks (*text chunking*) dengan ukuran ~3.000 karakter dan *overlap* ~300 karakter *per chunk*, lalu mengirimkan potongan teks tersebut ke *endpoint* “/api/embed”. **Backend** memanggil **Gemini Embedding API** untuk mengubah fragmen teks menjadi *vector embedding* 768-dimensi (“vector(768)”), lalu mengeksekusi batch insert data chunk beserta vektor ke tabel “**knowledge_chunks**” di **Supabase (pgvector)**. **Frontend** memperbarui status dokumen menjadi 'Siap' dan mereset statistik antarmuka RAG.



Gambar 5.48 Activity Diagram Iterasi 3 - Ingesti dan Pengindeksan Dokumen RAG

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.2.2 Activity Diagram Chat Berbasis RAG (Pencarian Semantik Basis Pengetahuan)

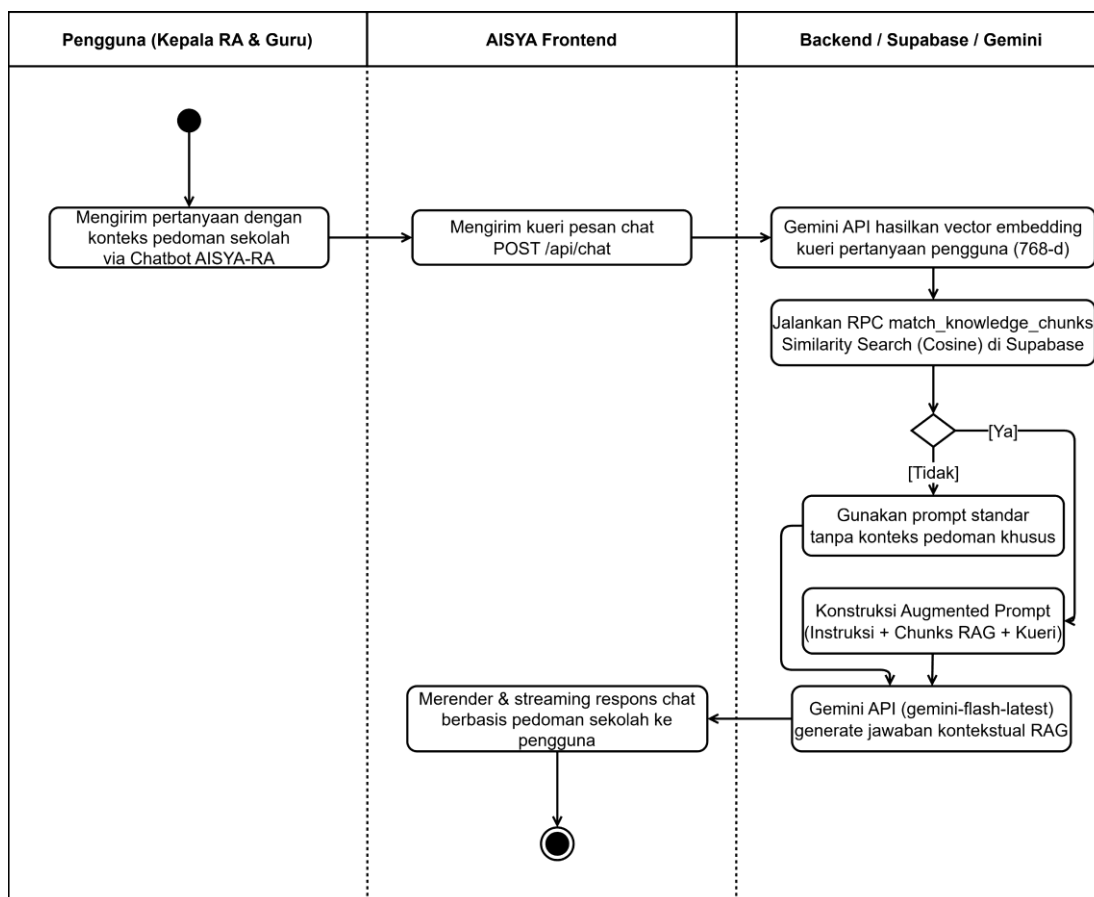
Activity Diagram Chat Berbasis RAG memodelkan alur pencarian semantik (*Cosine Similarity Search*) dan pencetusan jawaban berkonteks pedoman lokal sekolah. Diagram ini melibatkan tiga *swimlanes*, yaitu **Pengguna (Kepala RA & Guru)**, **AISYA Frontend (Chat.tsx)**, dan **Backend / Supabase / Gemini (chat.py, basis data Supabase pgvector, dan API Gemini)**.

Proses diawali ketika **Pengguna** memasukkan pertanyaan terkait kebijakan atau pedoman kurikulum sekolah via *chatbot*. **Frontend** mengirimkan kueri obrolan via HTTP “POST” (“POST /api/chat”) ke **Backend**. **Backend** memanggil **Gemini Embedding API** untuk membangkitkan *vector embedding* 768-dimensi dari teks kueri **Pengguna**, kemudian mengeksekusi fungsi RPC “match_knowledge_chunks” pada basis data **Supabase pgvector**.

Pada tahap ini terdapat titik keputusan (*decision node*) keberadaan *context chunk*:

1. Apabila *context chunk* pedoman tidak ditemukan [**Tidak Ditemukan**], **Backend** menggunakan *System Prompt* standar tanpa konteks pedoman khusus.
2. Apabila *context chunk* pedoman ditemukan [**Ya Ditemukan**], **Backend** menyusun *Augmented Prompt* yang menggabungkan *Instruksi Sistem*, data *context chunks* RAG yang diekstrak, *Data Master* sekolah, dan kueri **Pengguna**.

Backend mengirimkan *Augmented Prompt* ke **Gemini API** (gemini-flash-latest), dan **Frontend** merender serta mengalirkan (*streaming output*) teks respons obrolan berbasis rujukan resmi pedoman sekolah kepada **Pengguna**.



Gambar 5.49 Activity Diagram Iterasi 3 – Chat Berbasis RAG (Pencarian Semantik Basis Pengetahuan)

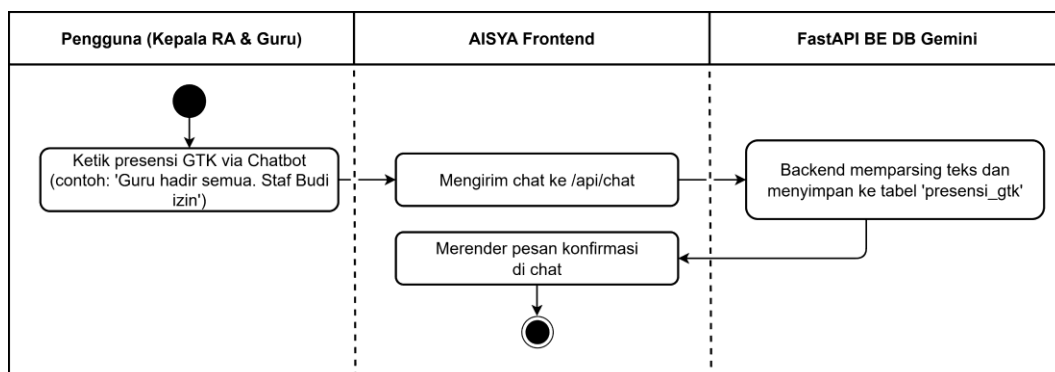
(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.2.3 Activity Diagram Pencatatan Presensi GTK (Guru & Tenaga Kependidikan)

Activity Diagram Pencatatan Presensi GTK memodelkan alur kerja perekaman kehadiran harian guru dan staf sekolah via obrolan *chatbot*. Pemodelan ini melibatkan *swimlanes* **Pengguna**, **AISYA Frontend**, dan **FastAPI BE (presensi.py) / DB / Gemini**.

Alur kerja diinisiasi ketika **Pengguna** menginputkan laporan kehadiran GTK via chatbot (contoh: "Guru hadir semua. Staf Budi izin"). **Frontend** mengirimkan pesan obrolan ke "/api/chat". **Backend** memarsing masukan bahasa alami, mencocokkan ID pengguna pada tabel "pengguna", dan menyisipkan/meng-upsert

record kehadiran ke tabel “presensi_gtk”. **Frontend** kemudian merender pesan konfirmasi pencatatan presensi GTK pada ruang obrolan.



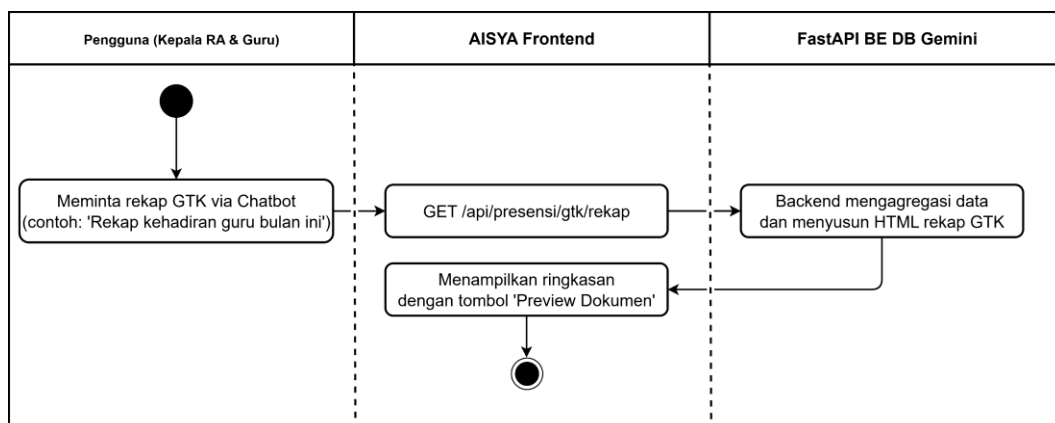
Gambar 5.50 Activity Diagram Iterasi 3 - Pencatatan Presensi GTK

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.2.4 Activity Diagram Rekap Presensi GTK

Activity Diagram Rekap Presensi GTK memodelkan alur kerja pembentukan laporan rekapitulasi kehadiran staf periodik. Pemodelan ini melibatkan *swimlanes* **Pengguna**, **AISYA Frontend**, dan **FastAPI / BE / DB / Gemini**.

Proses diawali saat **Pengguna** menginputkan perintah permintaan rekapitulasi kehadiran GTK via *chatbot*. **Frontend** mengirimkan permintaan HTTP “GET” ke “/api/presensi/gtk/rekap”. **Backend** menarik *record* kehadiran dari tabel “presensi_gtk”, mengagregasi statistik kehadiran staf, dan merangkai laporan berformat HTML. **Frontend** menampilkan ringkasan laporan di ruang obrolan beserta tombol "Preview Dokumen" untuk pembukaan pada AISYA *Canvas Modal*.



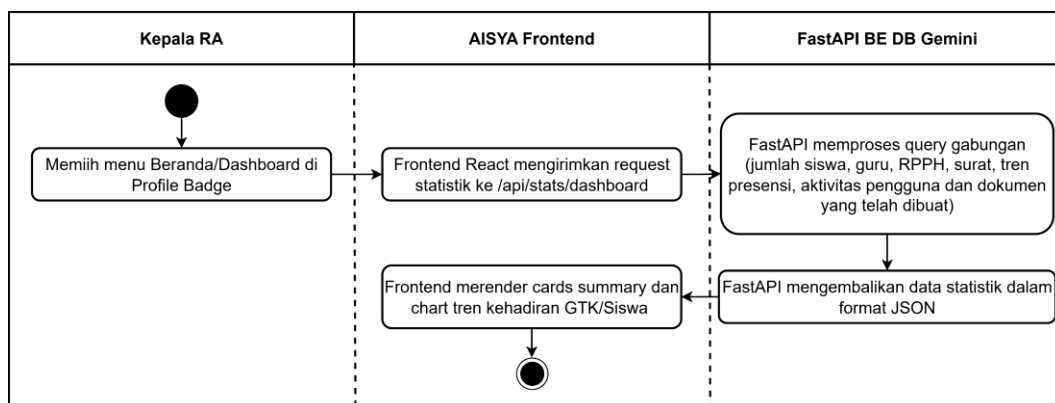
Gambar 5.51 Activity Diagram Iterasi 3 - Rekapitulasi Presensi GTK

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.2.5 Activity Diagram Dashboard Kepala RA

Activity Diagram Dashboard Kepala RA memodelkan alur kerja pemantauan statistik eksekutif operasional sekolah oleh **Kepala RA**. Pemodelan ini melibatkan *swimlanes* **Kepala RA**, **AISYA Frontend (Dashboard.tsx)**, dan **FastAPI BE (stats.py) / DB / Gemini**.

Alur kerja diinisiasi ketika **Kepala RA** memilih menu Beranda/*Dashboard* pada navigasi utama. **Frontend** mengirimkan permintaan data statistik eksekutif ke endpoint **Backend** (“GET /api/stats/dashboard”). **Backend** mengeksekusi kueri agregasi SQL gabungan (*join query*) pada basis data **Supabase** untuk menghitung jumlah total siswa aktif, total GTK, total RPPH tersimpan, total surat keluar, serta persentase rata-rata kehadiran. **Backend** mengembalikan data agregasi dalam format JSON, dan **Frontend** merender *card summary* statistik eksekutif serta grafik tren visual (Chart.js / Recharts) pada layar antarmuka *Dashboard*.



Gambar 5.52 Activity Diagram Iterasi 3 - Mengakses Dashboard

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

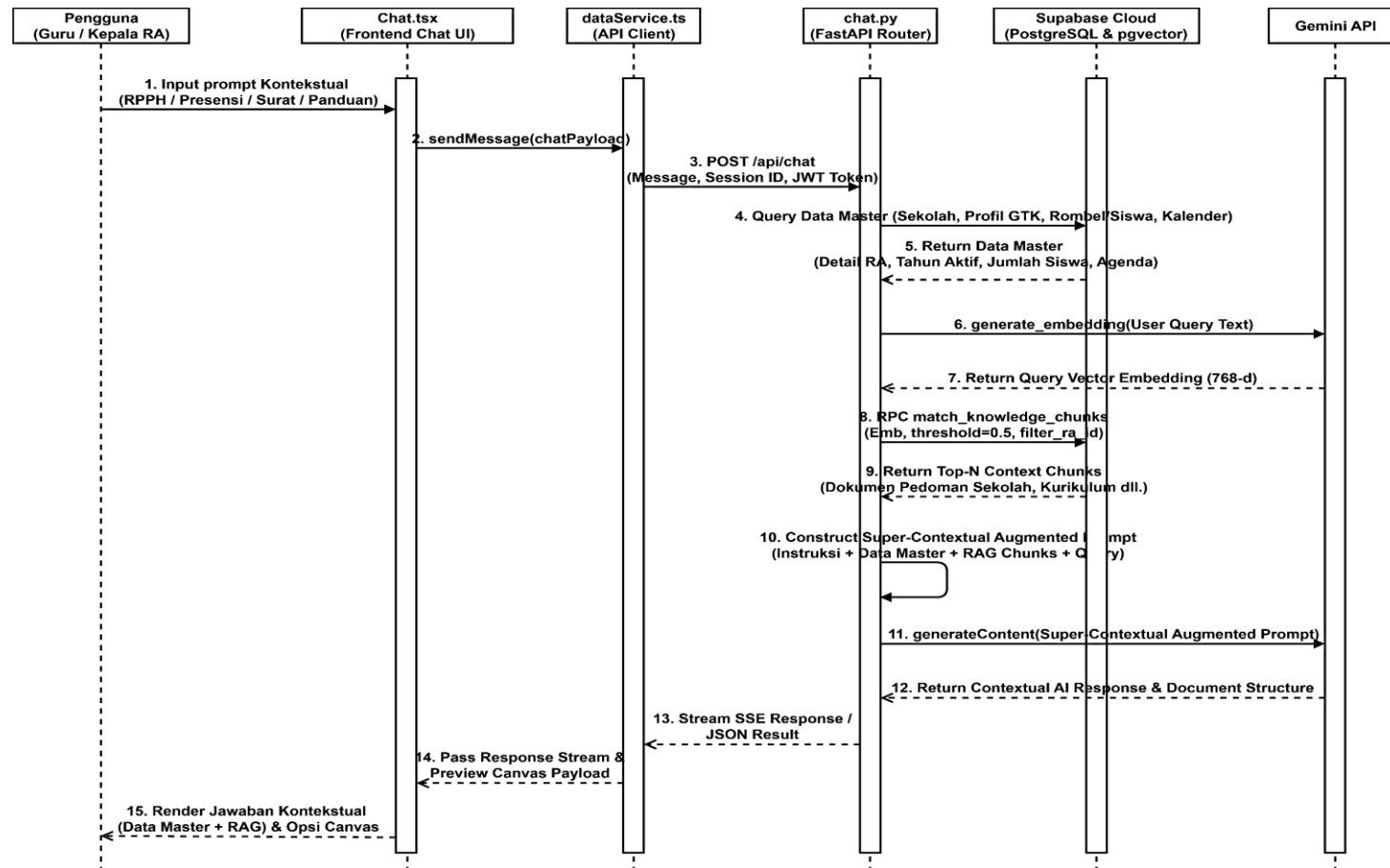
5.6.1.3 Sequence Diagram

Sequence Diagram Iterasi 3 menggambarkan alur pemrosesan pertukaran pesan *end-to-end* yang melibatkan injeksi *Data Master* kontekstual, pencarian semantik vektor dokumen basis pengetahuan (*Retrieval-Augmented Generation* atau RAG), pencatatan presensi GTK, dan pemuatan agregasi statistik manajerial pada *Dashboard*.

Alur eksekusi percakapan kontekstual berbasis RAG terdiri dari tahapan terstruktur berikut:

1. **Pengguna** mengirimkan perintah atau pertanyaan seputar kurikulum, SOP madrasah, atau agenda kegiatan melalui antarmuka percakapan *Chat.tsx*.
2. Modul *dataService.ts* mengirimkan permintaan HTTP *POST /api/chat* yang membawa teks pesan pengguna dan token otentikasi JWT ke backend *chat.py*.
3. Backend *chat.py* mengeksekusi kueri ke Supabase untuk mengambil *Data Master* lembaga (nama RA, NPSN, semester dan tahun ajaran aktif, data rombel, serta nama kepala madrasah) berdasarkan atribut *ra_id*.

4. Backend *chat.py* memanggil model *embedding* Google (*text-embedding-004*) untuk mengonversi kueri pengguna menjadi representasi vektor 768-dimensi.
5. Backend *chat.py* memicu fungsi *Remote Procedure Call* (RPC) Supabase *match_knowledge_chunks* untuk mencari potongan teks panduan sekolah yang memiliki tingkat kemiripan kosinus (*cosine similarity*) tertinggi di atas ambang batas yang ditentukan.
6. Backend *chat.py* menyusun instruksi sistem berkekuatan konteks tinggi (*Super-Contextual System Instruction*) yang memadukan aturan peran, data master madrasah, dan potongan teks RAG, lalu mengirimkannya ke model Gemini Flash (*gemini-latest-flash*).
7. Model Gemini API mengembalikan respons terstruktur dan backend *chat.py* mengalirkan respons tersebut kembali ke *Chat.tsx* untuk ditampilkan secara reaktif dan instan pada layar pengguna.



Gambar 5.53 Sequence Diagram Iterasi 3

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.4 Class Diagram

Class Diagram Iterasi 3 menyempurnakan arsitektur sistem AISYA-RA secara menyeluruh dengan mengintegrasikan modul Manajemen *Data Master* terpusat, infrastruktur basis pengetahuan vektor RAG, modul analitik eksekutif *Dashboard*, serta diferensiasi hak akses multi-peran (*Admin*, *Kepala RA*, dan *Guru RA*).

Struktur kelas pada Iterasi 3 mencakup penambahan komponen dan layanan berikut:

1. Lapisan Antarmuka Pengguna (*User Interface Layer*):

- a. *Dashboard (Dashboard.tsx)* dan *AiAnalytics (AiAnalytics.tsx)*: Menyajikan ringkasan statistik kehadiran harian, grafik tren penerbitan dokumen, log aktivitas sistem, serta pemantauan penggunaan token AI.
- b. *Staff (Staff.tsx)*: Komponen antarmuka GUI bagi Admin untuk mengelola data kepegawaian GTK, reset sandi staf, serta fitur impor data massal dari berkas spreadsheet.
- c. *Students (Students.tsx)*: Komponen antarmuka GUI bagi Admin untuk mengelola data siswa, pembagian rombongan belajar, kenaikan kelas, dan impor massal data peserta didik.
- d. *Settings (Settings.tsx)*: Komponen antarmuka GUI bagi Admin untuk memperbarui profil lembaga (NPSN, alamat, visi-misi, susunan kop surat) dan manajemen kunci API Gemini.
- e. *KnowledgeBase (KnowledgeBase.tsx)*: Komponen pengunggah berkas PDF panduan madrasah, pemantau status indeks vektor, dan repositori pengetahuan RAG.

- f. *AcademicCalendar* (*AcademicCalendar.tsx*): Komponen kalender akademik terintegrasi untuk pengaturan hari efektif pembelajaran, libur nasional, dan agenda kegiatan sekolah.

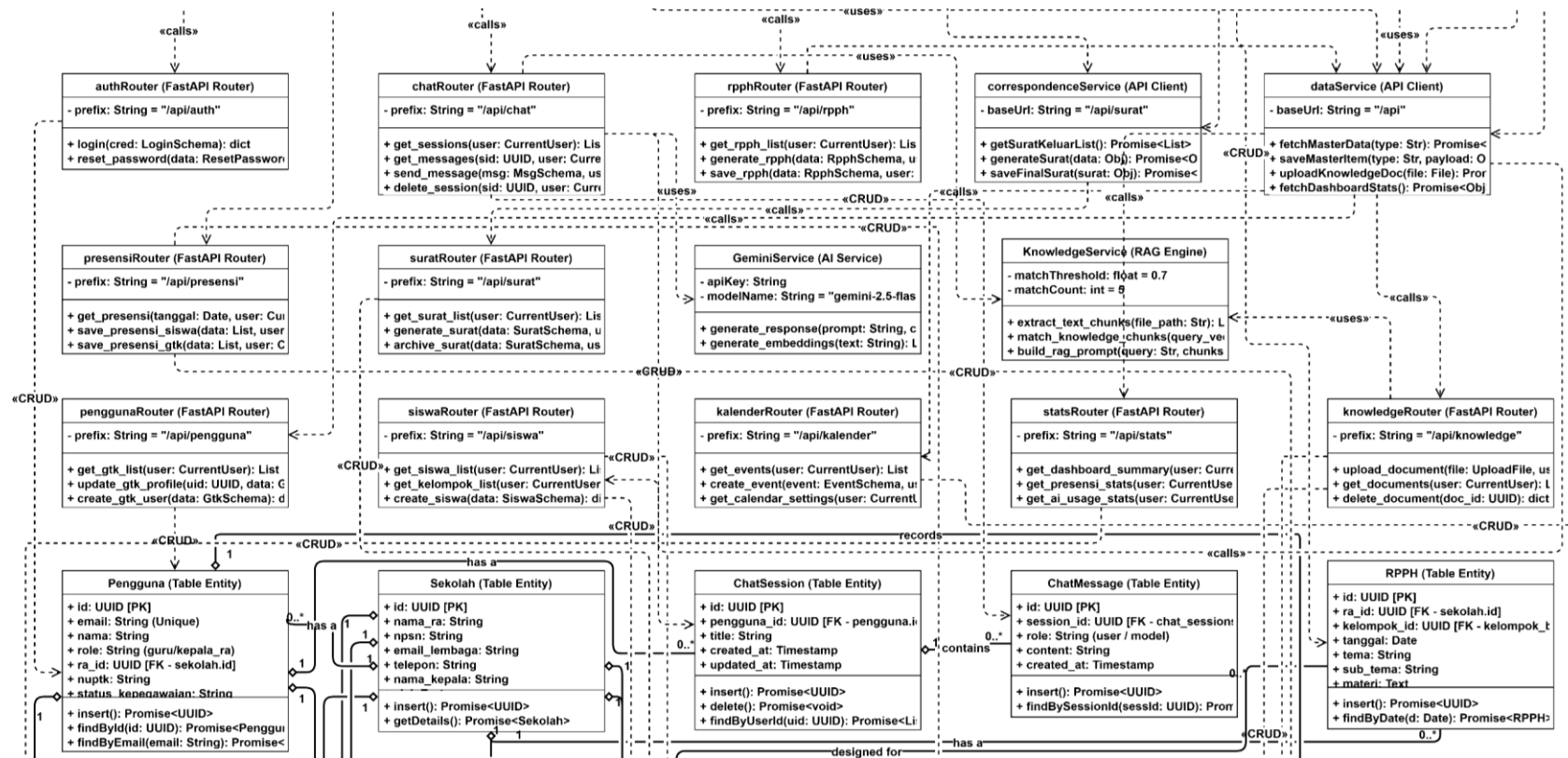
2. Lapisan Logika dan Pengendali (*Logic & Controller Layer*):

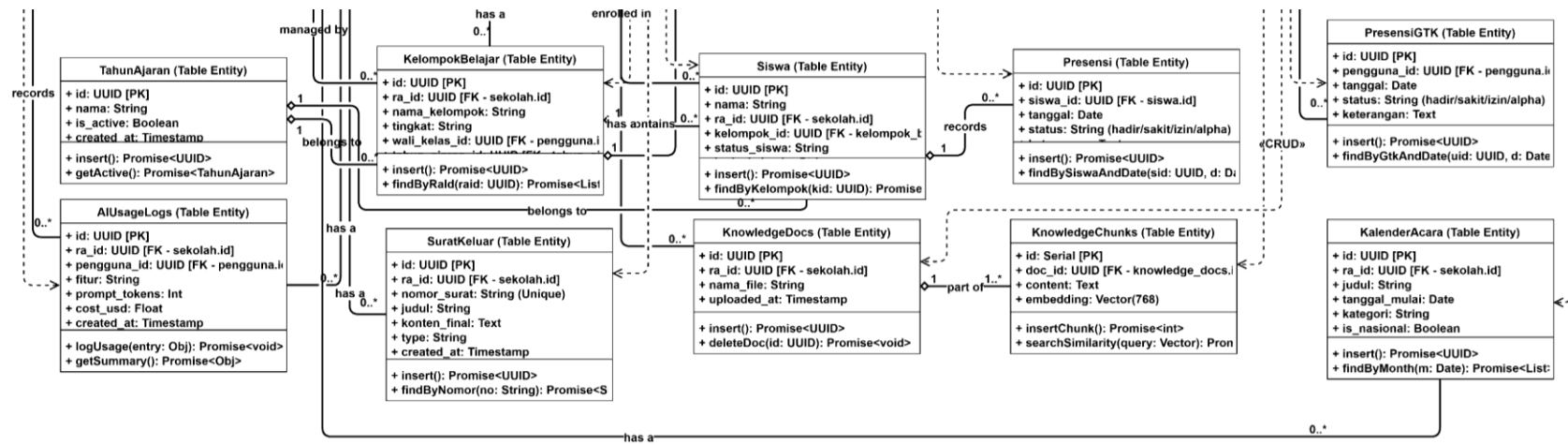
- a. Penambahan modul router FastAPI: *penggunaRouter* (*pengguna.py*), *siswaRouter* (*siswa.py*), *kalenderRouter* (*kalender.py*), *statsRouter* (*stats.py*), dan *knowledgeRouter* (*knowledge.py*).
- b. *KnowledgeService* (*RAG Engine*): Layanan pengurai teks dokumen PDF, pembangkit vektor embedding 768-dimensi, dan pelaksana fungsi pencarian semantik *match_knowledge_chunks*.

3. Lapisan Entitas Data (*Data Entity Layer*):

- a. Penambahan entitas tabel data: *PresensiGTK*, *KnowledgeDocs*, *KnowledgeChunks* (menyimpan vektor embedding 768-dimensi terindeks pgvector), *KalenderAcara*, *KalenderSettings*, *AIUsageLogs*, dan *RolePermissions*.







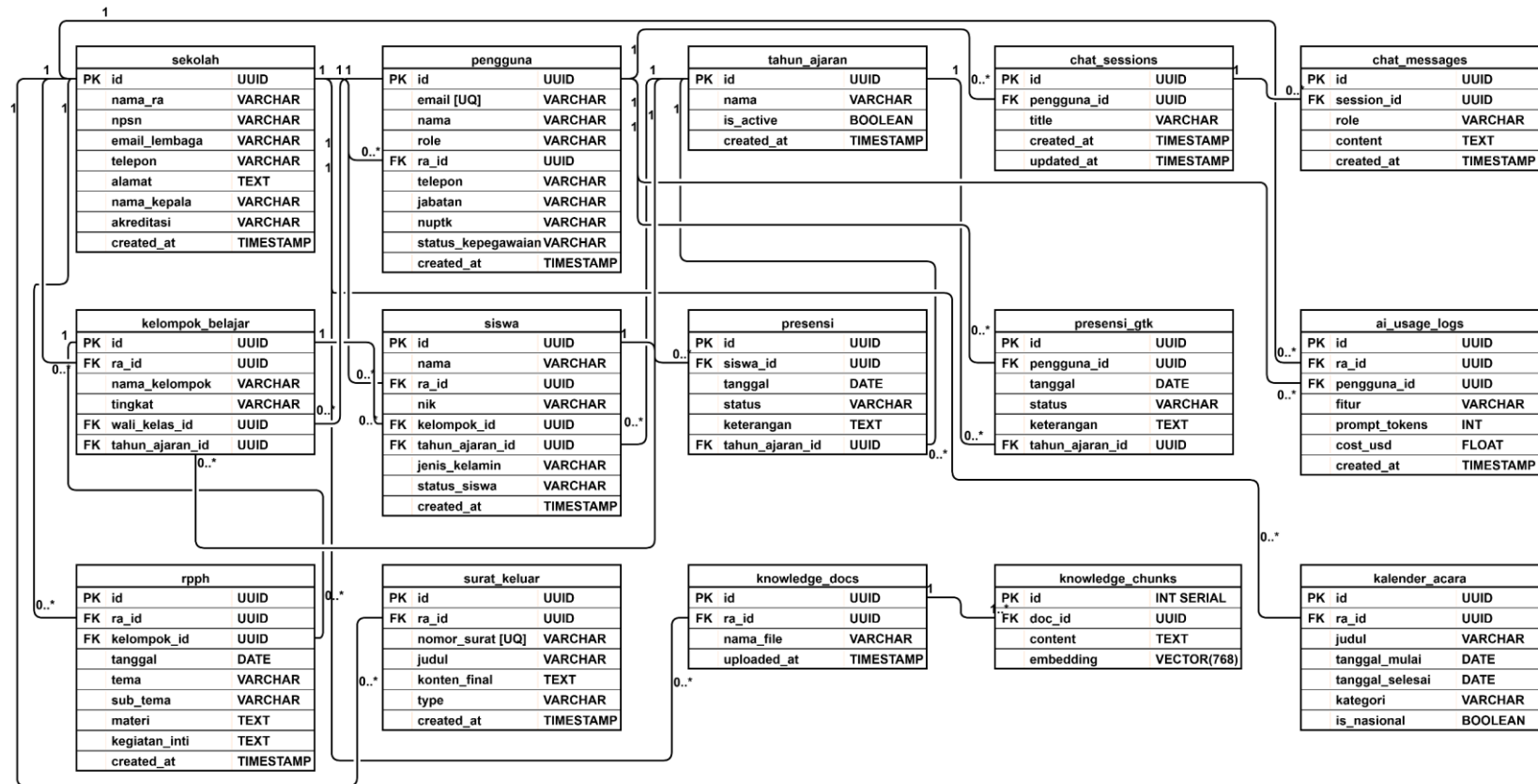
Gambar 5.54 Class Diagram Iterasi 3

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.5 Entity Relationship Diagram (ERD)

ERD Iterasi 3 memodelkan arsitektur basis data relasional dan vektor terpadu pada Supabase PostgreSQL:

1. **Isolasi Lembaga:** Seluruh tabel operasional dan data master (*kelompok_belajar*, *siswa*, *presensi*, *presensi_gtk*, *surat_keluar*, *knowledge_docs*, dan *rpph*) diikat oleh kolom kunci asing (*foreign key*) *ra_id* yang mereferensikan tabel *sekolah(id)*.
2. **Pembedaan Hak Akses Peran:** Tabel *pengguna* memfasilitasi peran *Admin*, *Kepala RA*, dan *Guru RA* yang mengatur hak akses menu dan fungsionalitas sistem secara konsisten.
3. **Infrastruktur Vektor RAG:** Tabel *knowledge_docs* terhubung secara *One-to-Many* (1 : 0..*) ke tabel *knowledge_chunks*. Kolom *embedding* bertipe data *vector(768)* dioptimalkan menggunakan indeks *pgvector* dan dieksekusi melalui fungsi SQL *match_knowledge_chunks* dengan metrik jarak kosinus (*Cosine Distance*).



Gambar 5.55 Entity Relationship Diagram Iterasi 3

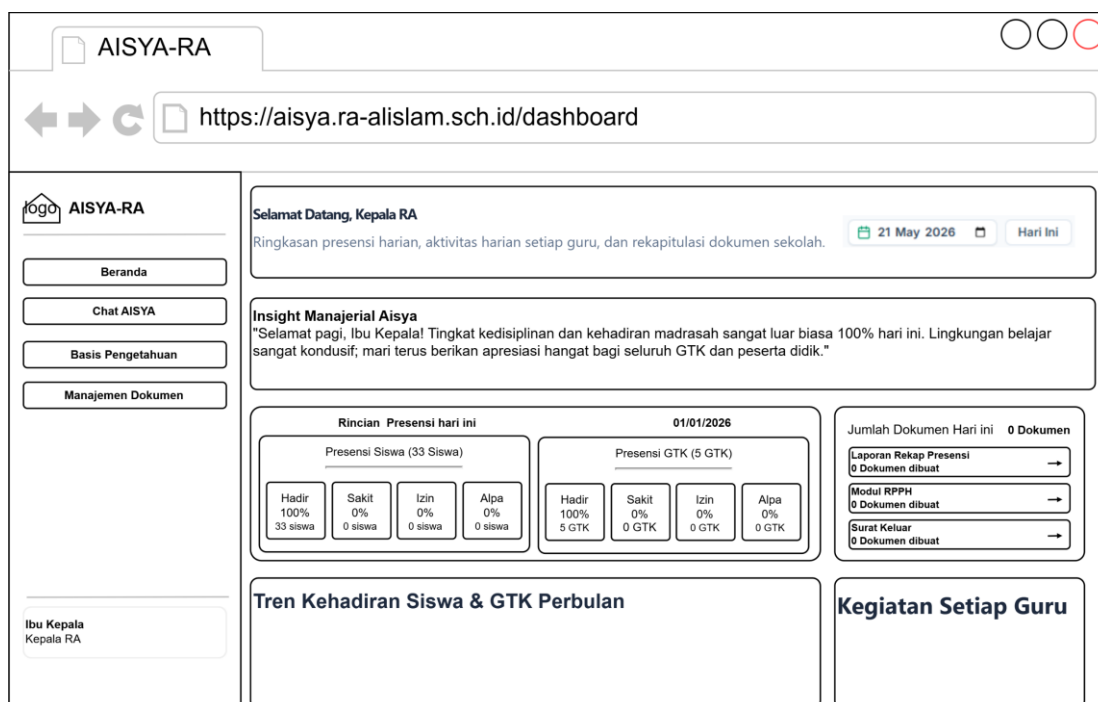
(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.6 Wireframe Antarmuka

Wireframe antarmuka Iterasi 3 dirancang dalam tujuh *Wireframe* antarmuka utama memvisualisasikan tata letak komponen visual, bilah navigasi utama, widget informasi, serta area interaksi pengguna.

5.6.1.6.1 Wireframe Dashboard

Wireframe Dashboard menyajikan antarmuka visual utama bagi Kepala RA saat pertama kali masuk ke sistem. Halaman ini memuat *widget Date Picker* menampilkan ringkasan pada hari yang aktif atau diatur, *widget Insight* Manajerial AISYA yang memberikan sapaan hangat dan ringkasan kondisi operasional sekolah secara otomatis, diikuti kartu statistik kehadiran harian, total dokumen yang dibuat hari ini, serta panel Aktivitas Terbaru yang menampilkan rekam pencatatan persuratan panel Kegiatan Setiap Guru yang menampilkan ceklis kegiatan administrasi yang dilakukan oleh setiap guru di AISYA-RA.



Gambar 5.56 Wireframe Antarmuka Iterasi 3 - Tampilan Dashboard

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.6.2 Wireframe Manajemen Data Master GTK (Guru dan Tenaga Kependidikan)

Wireframe Manajemen Data Master GTK dirancang untuk memfasilitasi pengelolaan data akun guru dan tenaga kependidikan. Antarmuka ini menampilkan bilah navigasi menu di sebelah kiri, serta area konten utama berisi tabel data GTK (memuat kolom NIP/ID Pegawai, Nama Lengkap, Jabatan, Nomor Kontak WhatsApp) dilengkapi tombol aksi cepat “Profil”, “Ubah Data”, “Reset Sandi”, dan “Hapus”.

The wireframe shows a web application interface for AISYA-RA. The top navigation bar includes the AISYA-RA logo and a search bar with the URL `https://aisyara-alislam.sch.id/staff`. The sidebar on the left contains navigation links: "Manajemen Data Siswa", "Manajemen Data GTK", "Manajemen Dokumen", "Basis Pengetahuan", and "Pengaturan". The main content area is titled "Manajemen GTK" and displays "Data Guru dan Tenaga Kependidikan Raudhatul Athfal." It includes buttons for "Tambah GTK" and "Import Excel". Below this is a search bar with the placeholder "Cari nama, NIP, jabatan ..." and filter tabs for "Semua", "Kepala RA", "Guru", and "Staf". The main table lists staff members with columns for "NAMA & ID PEGAWAI", "JABATAN", "KONTAK", and "Aksi".

NAMA & ID PEGAWAI	JABATAN	KONTAK	Aksi
Dra. Hj. Siti Aminah, M.Pd 198504122008012003	Kepala RA	6287777777777	[Profil] [Ubah Data] [Reset Sandi] [Hapus]
Bu Ani Rahmawati, S.Pd 199008232015032001	Guru	6287777777777	[Profil] [Ubah Data] [Reset Sandi] [Hapus]
Bu Budiarti, S.Pd.I 199211052018022004	Guru	6287777777777	[Profil] [Ubah Data] [Reset Sandi] [Hapus]
Pak Ahmad Fauzi, S.T 199503142020011002	Guru	6287777777777	[Profil] [Ubah Data] [Reset Sandi] [Hapus]

At the bottom left of the sidebar, the user is identified as "Bayu Septiana Aziz Admin".

Gambar 5.57 Wireframe Antarmuka Iterasi 3 - Tampilan Manajemen Data GTK

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.6.3 Wireframe Form Modal Tambah Data Master

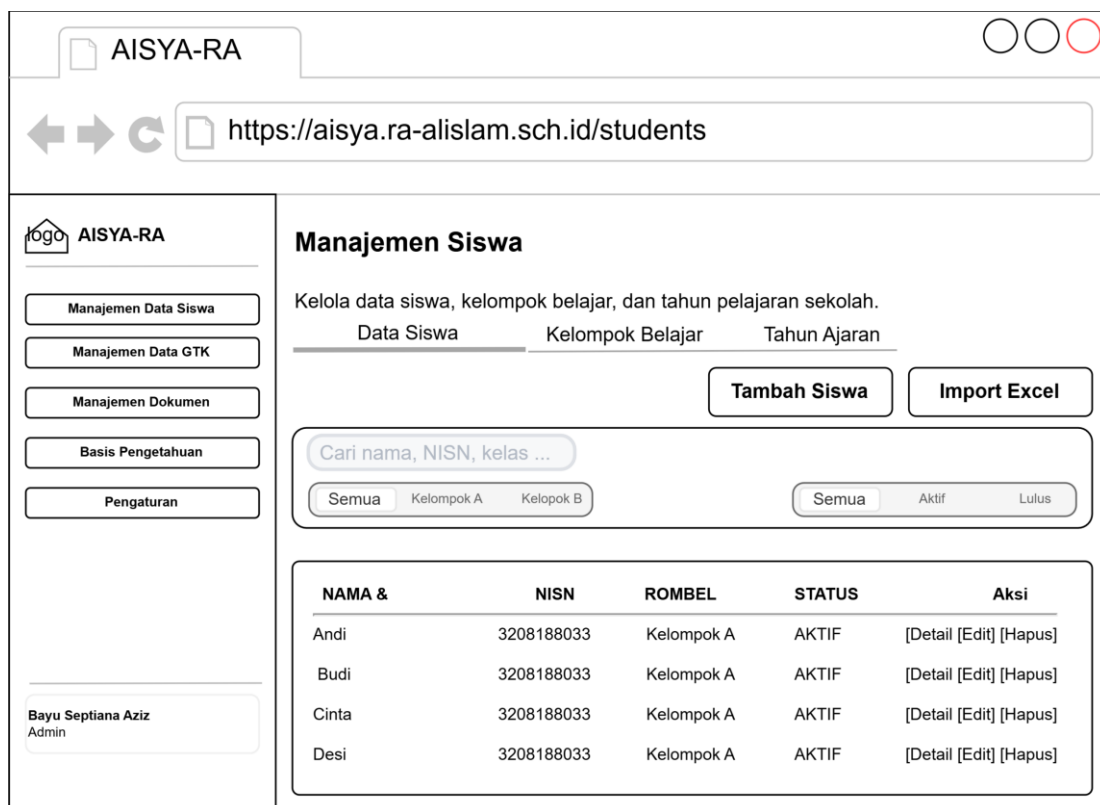
Wireframe Form Modal Tambah Data Master memvisualisasikan jendela dialog overlay interaktif yang muncul di atas halaman tabel utama saat pengguna menekan tombol "Tambah Data Baru". Form modal ini memuat bidang pengisian data secara langsung (seperti Nama, NIP/NISN, Jabatan, dan Kelompok) beserta tombol

“Simpan GTK” dan “Batal” tanpa perlu melakukan pengalihan halaman (*page reload*).

*Gambar 5.58 Wireframe Antarmuka Iterasi 3 - Form Modal Tambah Data Master
(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)*

5.6.1.6.4 Wireframe Manajemen Data Master Siswa

Wireframe Manajemen *Data Master* Siswa menyajikan antarmuka pengolahan data induk murid RA Al-Islam Gunungcupu. Halaman ini dilengkapi fitur filter kelompok belajar (Kelompok A dan Kelompok B), bilah pencarian nama siswa, serta tabel data siswa yang menampilkan NISN, nama lengkap, kelompok kelas, jenis kelamin, dan status keaktifan murid.



Gambar 5.59 Wireframe Antarmuka Iterasi 3 - Tampilan Manajemen Data Master Siswa

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.1.6.5 Wireframe Manajemen Data Master Lembaga

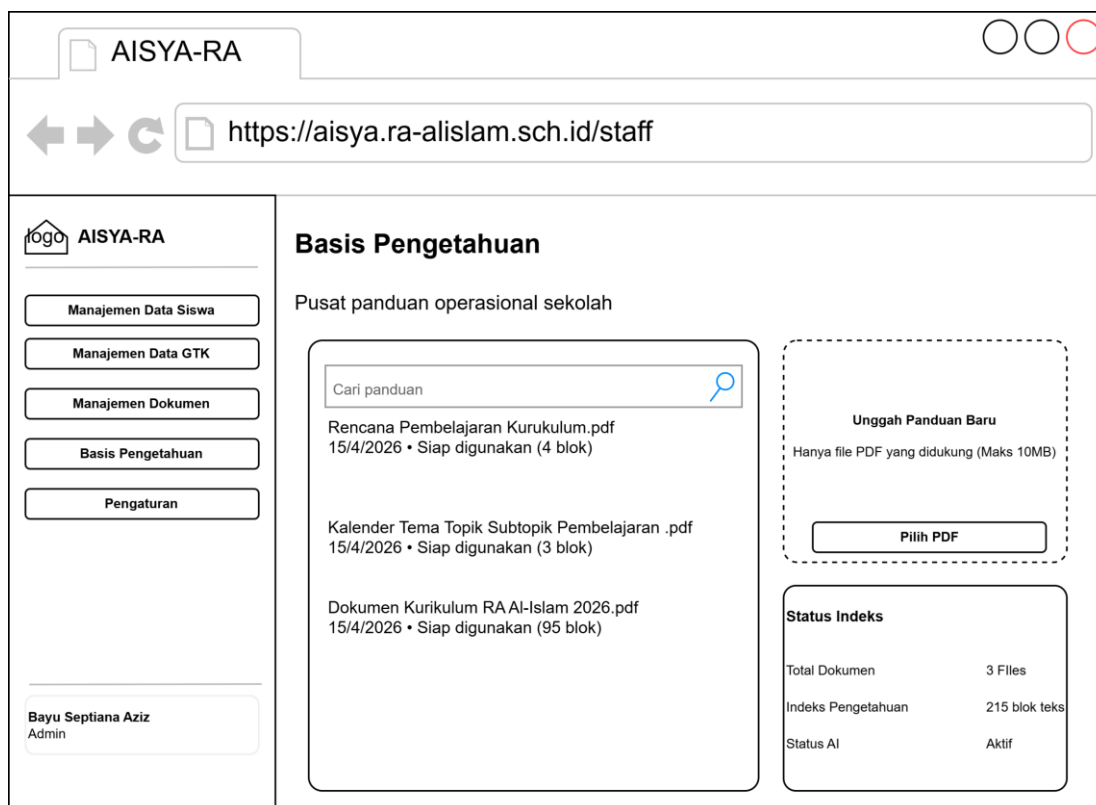
Wireframe Manajemen Data Master Lembaga dirancang untuk mengelola profil identitas resmi sekolah. Halaman ini menampilkan formulir data profil institusi yang memuat fitur penambahan Logo, bidang input NPSN, Alamat Lengkap, Kelurahan/Desa, Kecamatan, Kabupaten/Kota, Provinsi, Nomor Telepon/Email Resmi, serta Nama Kepala RA Aktif untuk disuntikkan secara dinamis ke dalam instruksi sistem LLM.

The wireframe shows a web browser window titled 'AISYA-RA' with the URL 'https://aisya.ra-alislam.sch.id/settings'. The page layout includes a left sidebar with navigation links: 'Manajemen Data Siswa', 'Manajemen Data GTK', 'Manajemen Dokumen', 'Basis Pengetahuan', and 'Pengaturan'. The main content area is titled 'Pengaturan Profil Lembaga' and contains a 'Ubah' button. Below the title is the instruction 'Kelola informasi profil lembaga Anda'. The form is divided into three main sections: 'Identitas Lembaga' (with fields for Logo, Nama Sekolah, NSPN, Kepala RA, Akreditasi, Status Sekolah, Kurikulum, No Izin Operasional, and Tanggal Izin Operasional), 'Alamat' (with fields for Jalan/nomor, Kelurahan/desa, Kecamatan, Kabupaten, Provinsi, and Kode Pos), and 'Kontak' (with fields for Telpon/WA, Email, and Alamat Web). There is also a 'Visi Misi' section with a 'Visi' field and three 'Misi' fields.

Gambar 5.60 Wireframe Antarmuka Iterasi 3 - Tampilan Manajemen Data Master Lembaga

5.6.1.6.6 Wireframe Basis Pengetahuan

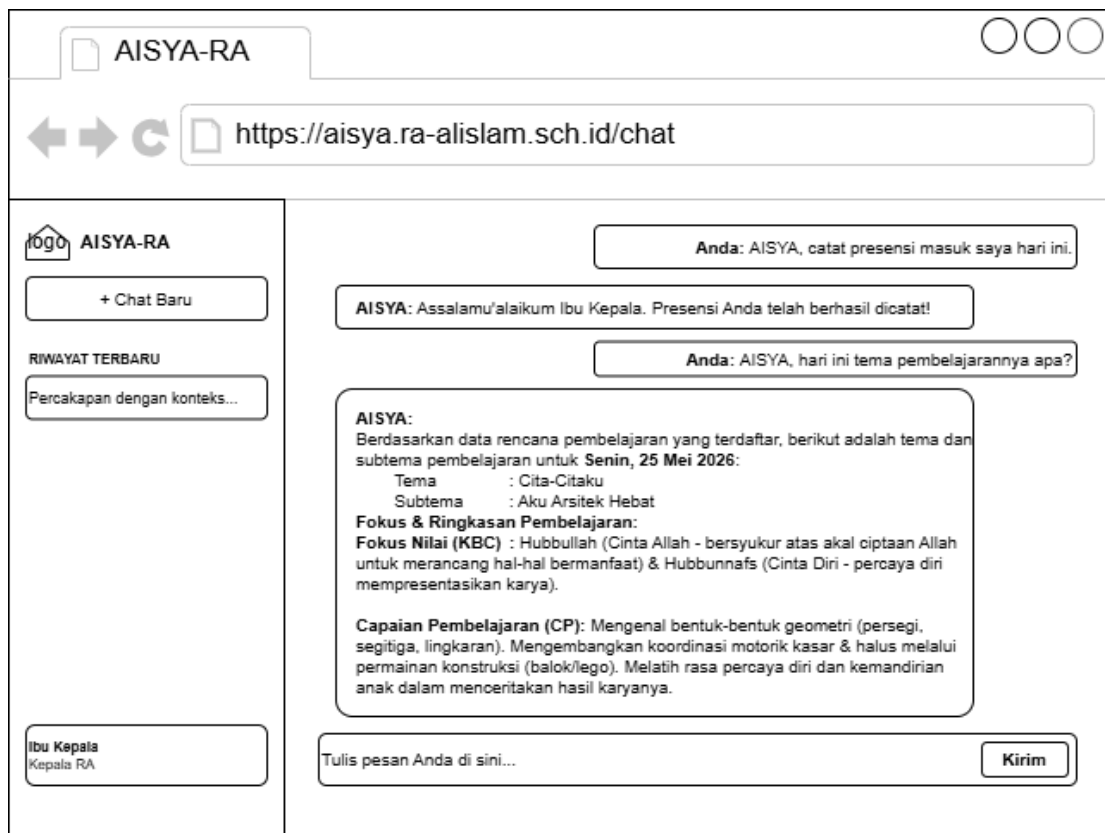
Wireframe Basis Pengetahuan menggambarkan antarmuka manajemen basis pengetahuan dokumen pedoman sekolah. Halaman ini menyajikan area pengunggahan berkas PDF pedoman sekolah, indikator status pengindeksan, serta tabel daftar dokumen terunggah yang dilengkapi tombol “Hapus Dokumen”.



*Gambar 5.61 Wireframe Antarmuka Iterasi 3 - Tampilan Basis Pengetahuan
(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)*

5.6.1.6.7 Wireframe Interaksi Chat (RAG & Presensi GTK)

Wireframe Interaksi Chat (RAG & Presensi GTK) memvisualisasikan ruang percakapan *Chatbot* interaktif tempat Guru dan Kepala RA berinteraksi menggunakan bahasa alami. Halaman ini menampilkan gelembung obrolan percakapan saat pengguna mengeksekusi presensi harian ("AISYA, catat presensi masuk saya hari ini") yang direspons konfirmasi instan oleh AISYA, serta percakapan konsultasi kurikulum ("AISYA, hari ini tema pembelajarannya apa?") yang direspons secara cerdas oleh AI berbasis rujukan dokumen RAG dan *Data Master*



Gambar 5.62 Wireframe Antarmuka Iterasi 3 - Tampilan Interaksi Chat (RAG & Presensi GTK)

(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

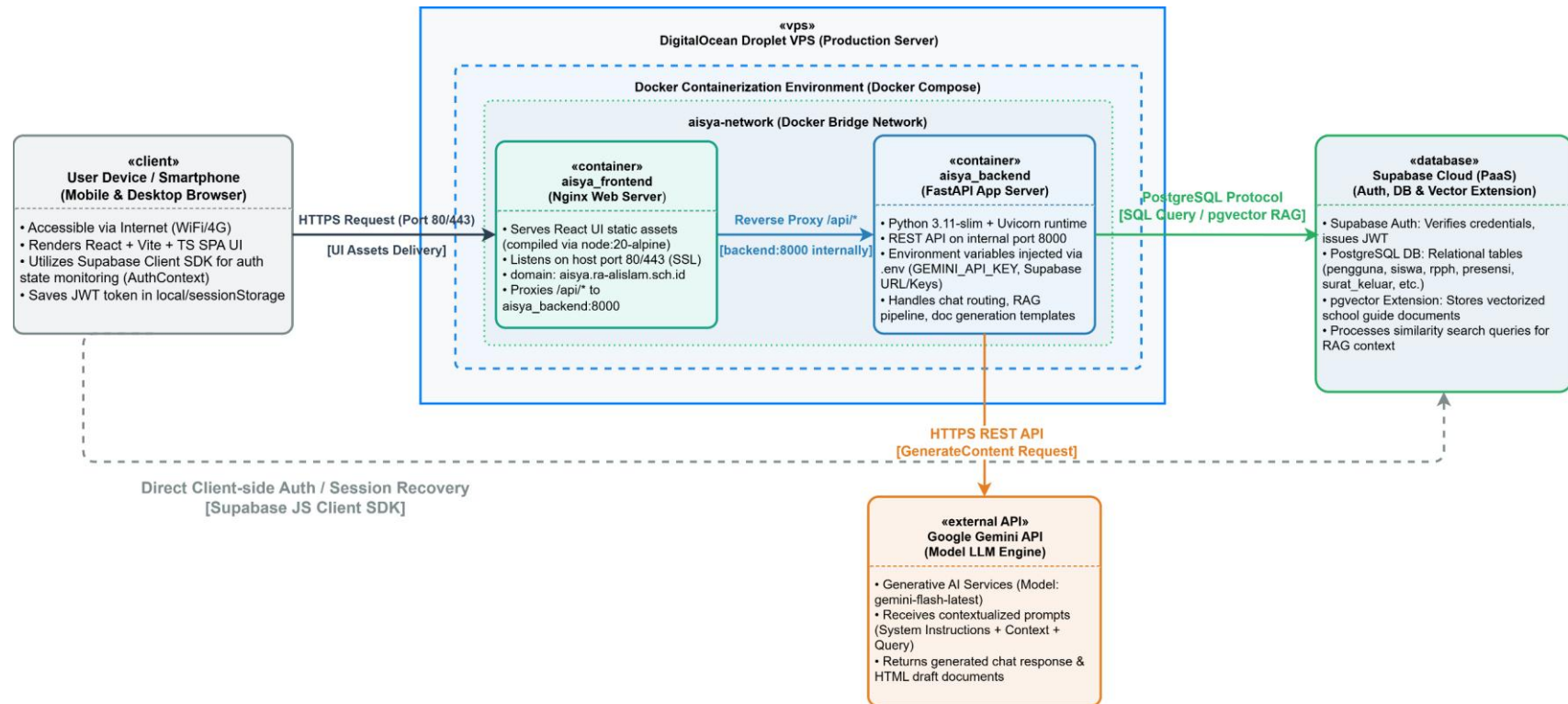
5.6.1.7 Arsitektur Sistem *Deployment* (*Production*)

Arsitektur *Deployment* Iterasi 3 menggambarkan topologi fisik dan infrastruktur sistem saat berjalan di lingkungan produksi (*cloud VPS*). Sistem ditempatkan pada server DigitalOcean Droplet VPS berbasis containerisasi *Docker Compose* dalam jaringan terisolasi “aisya-network”.

Komponen arsitektur mencakup:

1. *Client Device: Smartphone/Desktop* pengguna mengakses “https://aisya.ra-alislam.sch.id” via *browser* melalui enkripsi HTTPS.
2. “aisya_frontend” *Container*: Kontainer Nginx (*port* 80/443 SSL) yang melayani aset React SPA dan mem-proxy permintaan “/api/*” ke *backend*.

3. “aisya_backend” *Container*: Kontainer FastAPI (Python 3.11 + Uvicorn) di *port* internal 8000 yang mengelola bisnis logika, otorisasi JWT, presensi, RAG, dan API *routing*.
4. Supabase Cloud PaaS: Menyediakan *Supabase Auth*, *PostgreSQL DB*, dan ekstensi *pgvector* untuk RAG.
5. Google Gemini API: *External LLM Engine* untuk eksekusi model gemini-flash-latest.



Gambar 5.63 Arsitektur Sistem Deployment AISYA-RA
(Sumber: Hasil rancangan tahap design iterasi 3 AISYA-RA, 2026)

5.6.2 Develop Iterasi 3

Tahap *Develop* Iterasi 3 merealisasikan seluruh rancangan sistem tingkat lanjut, mencakup pembangunan GUI Manajemen *Data Master*, injeksi konteks data master ke LLM (*Super-Contextual Prompting*), perluasan presensi GTK, infrastruktur semantik RAG, dasbor analitik manajerial, registrasi perutean sistem, dan skrip *deployment* produksi di *cloud*.

Mengingat Iterasi 3 bertujuan mempersiapkan sistem untuk *deployment* operasional secara nyata, ketergantungan pada metode *database seeding* manual oleh peneliti seperti pada iterasi sebelumnya tidak lagi relevan. Sistem harus dapat diinisialisasi, dimutakhirkan, dan dikelola secara mandiri oleh pihak madrasah. Oleh sebab itu, realisasi *Product Backlog* untuk 'Manajemen *Data Master*' pada iterasi ini diwujudkan secara penuh melalui pembangunan antarmuka grafis (*Graphical User Interface / GUI*) *back-office* yang dikhususkan bagi peran Admin (Staff.tsx, Students.tsx, KnowledgeBase.tsx). Pendekatan ini mengeksekusi rencana arsitektur awal, di mana Admin bekerja menyiapkan pangkalan data master di balik layar, sehingga *Chatbot* dapat secara otomatis menyuntikkan data resmi lembaga saat memproses permintaan Guru dan Kepala RA

5.6.2.1 Implementasi *Data Master* dan Otomasi Kontekstual LLM

Pembangunan modul *Data Master* menyelesaikan kendala yang diidentifikasi pada sesi *Review* Iterasi 2. Dokumen RPPH dan surat dinas kini secara otomatis memuat identitas madrasah, alamat, nama kepala RA, dan periode akademik aktif tanpa perlu diketik ulang.

```
# backend/app/routers/chat.py (Konstruksi Super-Contextual System Instruction
dengan Data Master)

from app.routers.surat import get_sekolah_and_kepala_info, format_alamat_lengkap
info = get_sekolah_and_kepala_info(ra_id)
nama_yayasan = info['nama_yayasan']
sekolah_nama = info['nama_ra']
sekolah_npsn = info['npsn']
sekolah_alamat_full = format_alamat_lengkap(info)
kode_sekolah = info['kode_sekolah']
nama_kepala = info['nama_kepala']
```

```

nip_kepala = info['nip_kepala']
current_time_str = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
current_day_str = datetime.now().strftime("%A")
indo_days = {
    "Monday": "Senin", "Tuesday": "Selasa", "Wednesday": "Rabu",
    "Thursday": "Kamis", "Friday": "Jumat", "Saturday": "Sabtu", "Sunday":
    "Minggu"
}
current_day_indo = indo_days.get(current_day_str, current_day_str)
dynamic_context = f"""
Konteks Saat Ini (PENTING):
- Waktu server saat ini: {current_day_indo}, {current_time_str}
- Anda sedang berbicara dengan: {user.get('nama', 'Unknown')}
- Role pengguna: {user.get('role', 'Unknown')}
- Nama Yayasan: {nama_yayasan}
- Sekolah pengguna saat ini: {sekolah_nama} (NPSN: {sekolah_npsn})
- Alamat Sekolah: {sekolah_alamat_full}
- Kode Sekolah: {kode_sekolah}
- Nama Kepala Sekolah: {nama_kepala}
- NIP Kepala Sekolah: {nip_kepala}
ATURAN SURAT & DOKUMEN RESMI:
- Semua surat resmi WAJIB menyertakan Kop Surat (Nama Yayasan: {nama_yayasan},
Nama Sekolah: {sekolah_nama}, Alamat: {sekolah_alamat_full}).
- Penomoran Surat WAJIB menggunakan format:
{kode_sekolah}/{KODE_SURAT}/{URUTAN_BULAN}/{BULAN_ROMAWI}/{TAHUN}.
- Tanda Tangan Kepala Sekolah WAJIB memakai nama {nama_kepala} dan NIP:
{nip_kepala}.
"""
if req.system_instruction:
    full_system_instruction = req.system_instruction + "\n\n" + dynamic_context
else:
    full_system_instruction = SYSTEM_INSTRUCTION + "\n\n" + dynamic_context

payload = {
    "system_instruction": {"parts": [{"text": full_system_instruction}]},
    "contents": contents,
    "tools": [{"function_declarations": TOOLS_SCHEMA}],
    "generationConfig": {"temperature": 0.2}
}

```

Kode Program 5.14 chat.py - Konstruksi Super-Contextual Augmented Prompt dengan Data Master

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

Kode Program 5.14 mengimplementasikan teknik *Super-Contextual Prompting* yang secara dinamis mengekstrak profil fisik madrasah dari tabel sekolah berdasarkan nilai `ra_id` pengguna yang sedang aktif melalui fungsi `get_sekolah_and_kepala_info`. Variabel nama yayasan, nama madrasah, nomor NPSN, struktur alamat lengkap berjenjang, nama kepala madrasah, NIP, hingga penanggalan waktu server lokal dienkapsulasi ke dalam variabel `string dynamic_context`. Nilai ini digabungkan secara mulus dengan `SYSTEM_INSTRUCTION` utama sebelum dikirimkan ke model Gemini API. Dengan strategi ini, mesin LLM selalu memahami konteks institusional secara utuh, sehingga setiap draf dokumen formal otomatis dilengkapi dengan kop resmi dan identitas lembaga yang presisi.

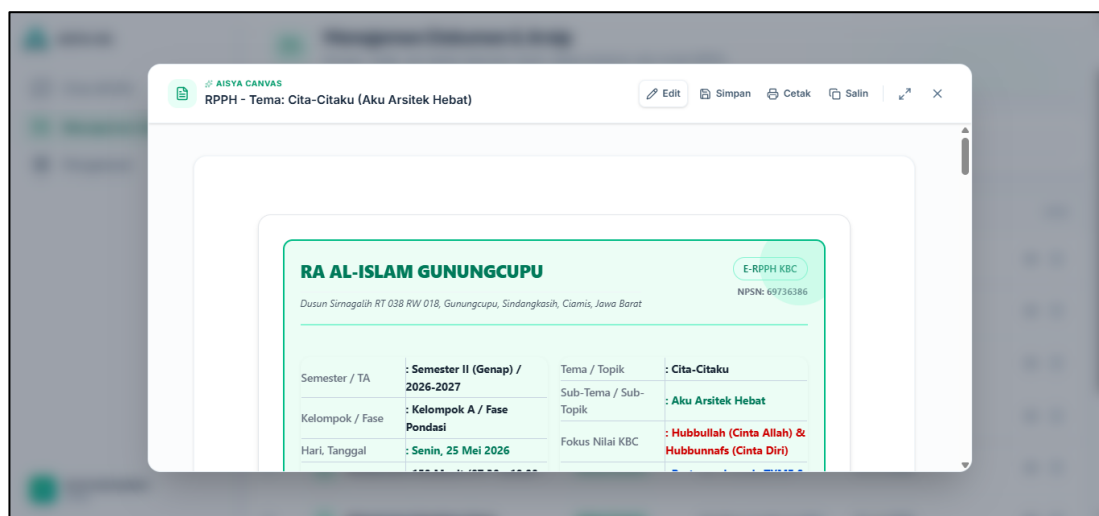
Perubahan pada modul `chat.py` di Iterasi 3 ini menunjukkan evolusi arsitektural yang sangat signifikan dibandingkan iterasi-iterasi sebelumnya:

1. **Perbandingan dengan Iterasi 1:** Pada Iterasi 1 (Kode Program 5.9), berkas `chat.py` hanya menggunakan instruksi sistem statis (*hardcoded*) yang bertindak sebagai agen percakapan umum tanpa pengenalan data sekolah, bahkan secara eksplisit menolak perintah pembuatan dokumen administratif karena fungsionalitasnya belum dibangun.
2. **Perbandingan dengan Iterasi 2:** Pada Iterasi 2, `chat.py` mulai dilengkapi dengan skema pemanggilan fungsi (*Tool Calling / Function Calling*) untuk membentuk struktur JSON RPPH dan surat, namun draf yang dihasilkan masih bersifat generik (*placeholder*) sehingga guru masih harus mengisi ulang nama sekolah, alamat, dan nama kepala madrasah secara manual di editor canvas.
3. **Penyempurnaan di Iterasi 3:** Pada Iterasi 3, `chat.py` berevolusi menjadi mesin orkestrasi kontekstual cerdas (*Super-Contextual Engine*) yang mengotomatisasi injeksi *Data Master* lembaga secara *real-time* sekaligus mengintegrasikan pencarian semantik vektor dokumen kurikulum RAG (`knowledge_search`), mengeliminasi pekerjaan penyuntingan manual oleh guru secara total.

5.6.2.2 Hasil Implementasi Dokumen Kontekstual *Data Master*

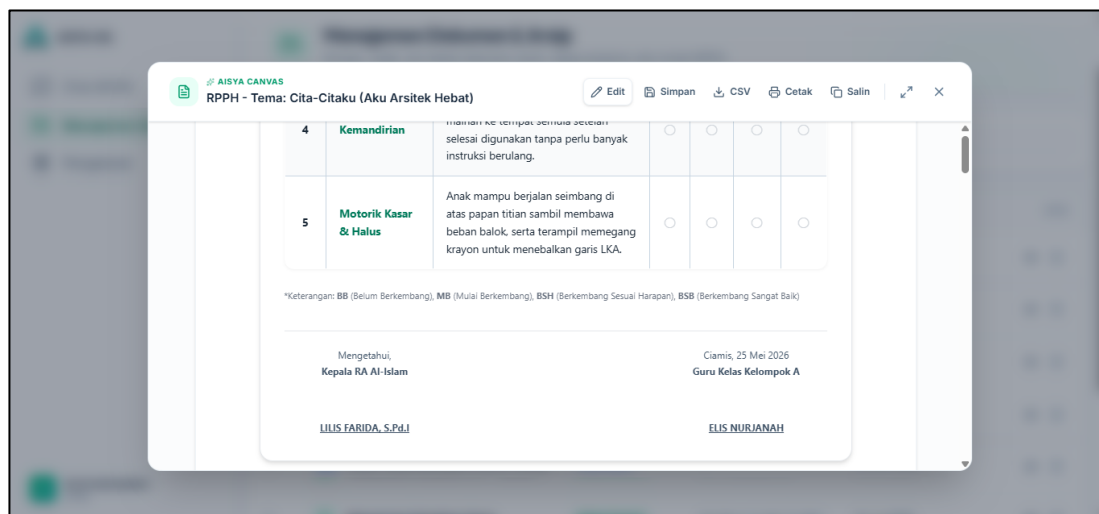
Hasil implementasi injeksi *Data Master* secara langsung menghasilkan dokumen rencana pembelajaran (RPPH) dan surat keluar yang lengkap dan formal:

1. **Header Dokumen RPPH:** Otomatis memuat nama resmi madrasah (RA Al-Islam Gunungcupu), nomor NPSN, susunan alamat lengkap lembaga, semester aktif, dan kelompok usia siswa tanpa perlu diketik ulang oleh guru.
2. **Identitas Pengesahan:** Bagian akhir dokumen secara otomatis memuat titimangsa daerah, nama dan gelar Kepala RA, serta nama guru kelas pengampu.



Gambar 5.64 Tampilan Kontekstual Header RPPH

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



Gambar 5.65 Tampilan Identitas Kepala RA dan Wali Kelas

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

5.6.2.3 Implementasi Perluasan Modul Presensi untuk GTK

Pencatatan kehadiran staf GTK dilakukan melalui percakapan bahasa alami dengan validasi hak akses dan pencegahan duplikasi data harian (*upsert*):

```
# backend/app/routers/presensi.py (Handler Presensi GTK via Chatbot)
@router.post("/presensi-gtk")
def create_presensi_gtk(
    data: dict,
    user: Dict[str, Any] = Depends(get_current_user),
    ra_id: str = Depends(get_current_ra_id)
):
    if 'pengguna_id' not in data or 'status' not in data:
        raise HTTPException(status_code=400, detail="Data tidak lengkap")

    if user.get("role") not in ["kepala_ra", "admin"] and user.get("id") != data["pengguna_id"]:
        raise HTTPException(status_code=403, detail="Anda hanya dapat menandai kehadiran Anda sendiri atau memerlukan hak akses admin.")

    try:
        today = data.get('tanggal') or datetime.date.today().isoformat()
        years_res =
        supabase.table('tahun_ajaran').select('id').eq('is_active', True).limit(1).execute()
```



```

        if not years_res.data:
            raise HTTPException(status_code=500, detail="Tahun ajaran
aktif tidak ditemukan.")

        active_year_id = years_res.data[0]['id']

        insert_data = {
            'pengguna_id': data['pengguna_id'],
            'status': data['status'],
            'tanggal': today,
            'tahun_ajaran_id': active_year_id
        }

        response = supabase.table('presensi_gtk').upsert(insert_data,
on_conflict='pengguna_id,tanggal').execute()
        return response.data[0] if response.data else None
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

```

Kode Program 5.15 presensi.py - Endpoint Pencatatan Presensi GTK via Chatbot dengan Mekanisme Upsert

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

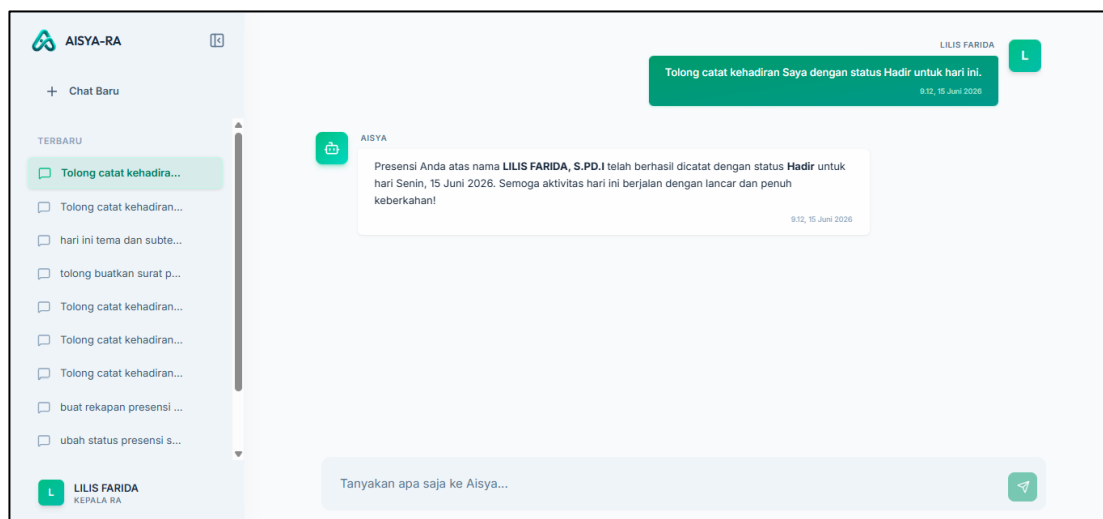
Kode Program 5.15 memperlihatkan logika pencatatan kehadiran kepegawaian pada fungsi `create_presensi_gtk`. Fungsi ini menerapkan pemeriksaan otoritas peran (*Role-Based Access Control*), di mana peran Kepala RA dan Admin memiliki wewenang untuk mencatatkan kehadiran seluruh staf, sedangkan peran Guru RA hanya diizinkan mencatat kehadiran untuk akun pribadinya sendiri (`user.get("id") == data["pengguna_id"]`). Untuk menjamin integritas data dan mencegah anomali data ganda, perintah penyimpanan mengeksekusi operasi `upsert` pada tabel `presensi_gtk` dengan klausa `on_conflict="pengguna_id,tanggal"`, sehingga jika terjadi perubahan status presensi pada hari yang sama, data lama akan diperbarui tanpa membuat baris baru.

Perubahan pada modul `presensi.py` ini menghadirkan perbedaan mendasar dibandingkan implementasi pada Iterasi 2:

1. **Cakupan Entitas pada Iterasi 2:** Pada Iterasi 2 (Kode Program 5.11), berkas `presensi.py` hanya dirancang untuk melayani pencatatan dan rekapitulasi kehadiran peserta didik (siswa) per rombongan belajar yang dikelola oleh Guru RA sebagai wali kelas.
2. **Perluasan Fungsional di Iterasi 3:** Pada Iterasi 3, modul diperluas secara substansial dengan menambahkan *handler* presensi kepegawaian (POST /presensi-gtk dan GET /presensi-gtk/rekap) sebagai tindak lanjut langsung atas masukan Kepala RA pada sesi *Review* Iterasi 2. Modul ini secara otomatis mengaitkan setiap rekam kehadiran dengan `tahun_ajaran_id` yang sedang aktif serta menyediakan rekapitulasi berkala kehadiran staf madrasah.

5.6.2.4 Hasil Implementasi Presensi GTK via *Chatbot*

Hasil implementasi presensi GTK memungkinkan Kepala RA mencatat kehadiran pengajar atau dirinya sendiri melalui percakapan obrolan (misalnya: *"Tolong catat kehadiran Saya dengan status Hadir untuk hari ini"*). Sistem memproses instruksi, mengonfirmasi nama tenaga pendidik yang tercatat, dan menyimpan status kehadiran secara *real-time*.



Gambar 5.66 Tampilan Presensi GTK Kepala RA

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

5.6.2.5 Implementasi RAG (Retrieval-Augmented Generation)

Pengembangan fitur RAG diimplementasikan melalui ingesti berkas PDF di backend serta eksekusi pencarian vektor semantik menggunakan *stored procedure* PostgreSQL Supabase (`match_knowledge_chunks`).

```
# backend/app/routers/knowledge.py (Pipeline Ingesti dan Pencarian RAG)
@router.post("/documents/{doc_id}/process")
def process_document(
    doc_id: str,
    payload: ProcessRequest,
    ra_id: str = Depends(get_current_ra_id),
    admin: dict = Depends(require_admin_for_docs)
):
    if not payload.chunks:
        raise HTTPException(status_code=400, detail="No chunks provided
for processing.")

    try:
        data_to_insert = [
            {
                "doc_id": doc_id,
                "content": chunk.content,
                "embedding": chunk.embedding[:768] if
len(chunk.embedding) > 768 else chunk.embedding + [0.0]*(768 -
len(chunk.embedding))
            }
            for chunk in payload.chunks
        ]
        response =
supabase.table('knowledge_chunks').insert(data_to_insert).execute()
        return {"success": True, "chunksProcessed": len(payload.chunks)}
    except Exception as e:
        raise HTTPException(status_code=500, detail=str(e))

@router.post("/knowledge-search")
def knowledge_search(payload: SearchRequest, ra_id: str =
Depends(get_current_ra_id)):
    if not payload.queryEmbedding:
        raise HTTPException(status_code=400, detail="No query embedding
provided.")

    try:
```

```

emb = payload.queryEmbedding
emb_768 = emb[:768] if len(emb) > 768 else emb + [0.0]*(768 -
len(emb))

response = supabase.rpc('match_knowledge_chunks', {
    'query_embedding': emb_768,
    'match_threshold': 0.5,
    'match_count': 5,
    'filter_ra_id': ra_id
}).execute()

return {"chunks": response.data or []}
except Exception as e:
    raise HTTPException(status_code=500, detail=str(e))

```

Kode Program 5.16 knowledge.py & chat.py - Pipeline Ingesti dan Pencarian RAG via Supabase RPC

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

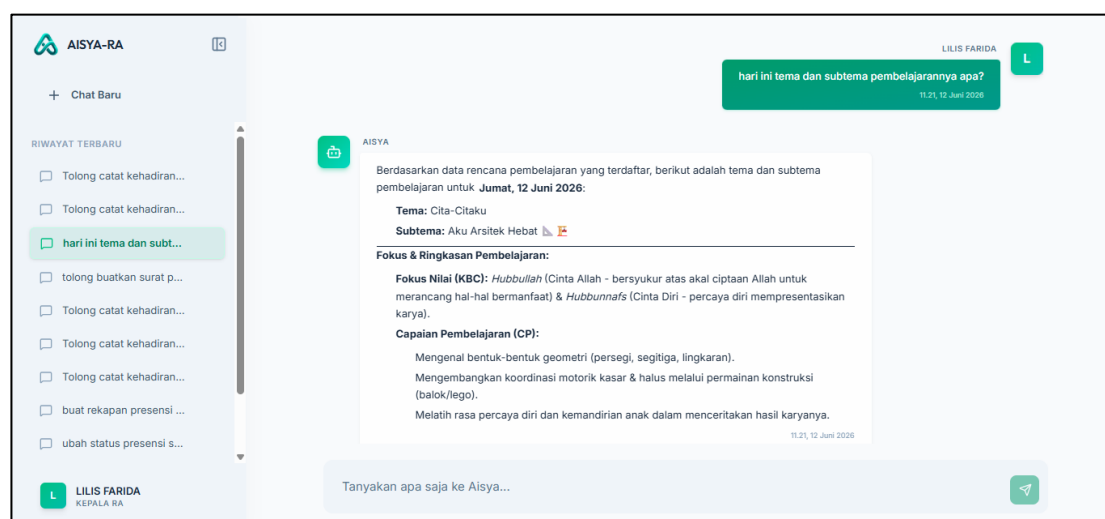
Kode Program 5.16 memperlihatkan dua komponen inti arsitektur RAG:

1. *Fungsi Ingesti Dokumen* (process_document): Menerima potongan teks (*chunks*) dari berkas PDF panduan kurikulum/SOP yang telah diekstrak dan dihitung vektor *embedding*-nya (berdimensi 768 melalui model text-embedding-004), kemudian menyimpannya secara massal (*batch insert*) ke tabel knowledge_chunks.
2. *Fungsi Pencarian Semantik* (knowledge_search_helper): Mengonversi kueri obrolan pengguna menjadi vektor *embedding*, lalu memanggil fungsi RPC PostgreSQL match_knowledge_chunks pada Supabase. Fungsi basis data ini mengeksekusi kalkulasi jarak kemiripan kosinus (*Cosine Similarity*) menggunakan ekstensi *pgvector*, dengan ambang batas kecocokan minimum 0.5 dan batas maksimal 5 potongan teks teratas yang secara ketat difilter berdasarkan atribut filter_ra_id.

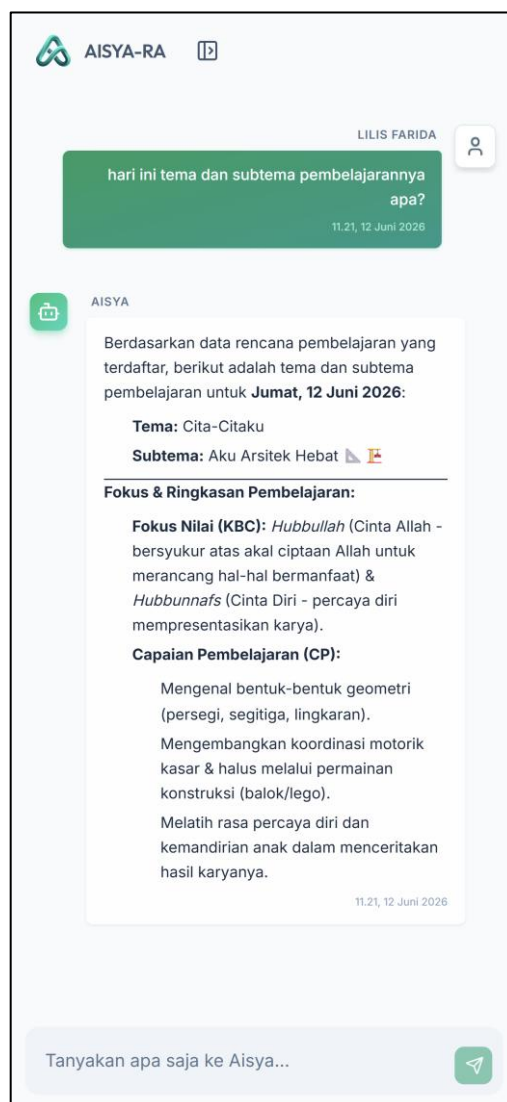
5.6.2.6 Hasil Implementasi Percakapan Berbasis RAG

Hasil implementasi percakapan RAG memungkinkan asisten AI menjawab pertanyaan seputar pedoman kurikulum madrasah, silabus tema ajar, atau SOP operasional dengan merujuk langsung pada dokumen lokal yang diunggah Admin. Asisten AI menyajikan jawaban yang akurat, lengkap dengan rujukan nilai karakter (KBC: Hubbullah dan Hubbunnafs) serta Capaian Pembelajaran (CP) yang sesuai:

1. **Tampilan RAG Desktop:** Menampilkan jawaban terstruktur dengan format Markdown yang rapi.
2. **Tampilan RAG *Smartphone*:** Menyajikan kemudahan konsultasi kurikulum langsung dari genggamannya guru di dalam kelas.



*Gambar 5.67 Tampilan Percakapan Chatbot dengan Respons Berbasis RAG
 (Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)*



*Gambar 5.68 Tampilan Percakapan Chatbot dengan Respons Berbasis RAG
(Diakses dari Smartphone)*

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

5.6.2.7 Implementasi Agregasi Statistik Dashboard

Modul *stats.py* menyediakan *endpoint* data manajerial teragregasi tingkat lanjut untuk menyajikan ringkasan eksekutif yang komprehensif bagi Kepala RA, mendukung pemilihan tanggal aktif (*target date filter*), rincian status presensi, jumlah dokumen harian, grafik tren bulanan, dan rekam aktivitas guru.

```

# backend/app/routers/stats.py (Endpoint Agregasi Statistik Dashboard Eksekutif)

@router.get("/stats", response_model=StatsResponse)
def get_stats(
    ai_period: str = "all",
    target_date: Optional[str] = None,
    user: Dict[str, Any] = Depends(get_current_user),
    ra_id: str = Depends(get_current_ra_id)
):
    today = target_date if target_date else
datetime.date.today().isoformat()

    if not ra_id:
        return {
            "studentCount": 0, "staffCount": 0, "unreadLetters": 0,
            "attendanceRate": 0, "recentActivities": [],
            "upcomingAgendas": []
        }

    try:
        today_start = f"{today}T00:00:00"
        today_end = f"{today}T23:59:59"

        # 1. Agregasi total master siswa dan GTK
        total_siswa = supabase.table('siswa').select('id',
count='exact').eq('ra_id', ra_id).execute().count or 0
        total_gtk = supabase.table('pengguna').select('id',
count='exact').eq('ra_id', ra_id).execute().count or 0

        # 2. Transaksi presensi siswa dan GTK pada tanggal aktif
        p_siswa =
supabase.table('presensi').select('status').eq('tanggal',
today).execute().data or []
        p_gtk =
supabase.table('presensi_gtk').select('status').eq('tanggal',
today).execute().data or []

        # 3. Hitung rincian dokumen yang dibuat pada tanggal aktif
        rpph_today = supabase.table('rpph').select('id').eq('ra_id',
ra_id).gte('created_at', today_start).lte('created_at',
today_end).execute().data or []
        surat_today =
supabase.table('surat_keluar').select('id').eq('ra_id',
ra_id).gte('created_at', today_start).lte('created_at',
today_end).execute().data or []
        rekap_today =
supabase.table('ai_usage_logs').select('id').eq('ra_id',

```

```

ra_id).gte('created_at', today_start).lte('created_at',
today_end).execute().data or []

docs_breakdown = {
    "total": len(rpph_today) + len(surat_today) +
len(rekap_today),
    "rekap_count": len(rekap_today),
    "surat_count": len(surat_today),
    "rpph_count": len(rpph_today)
}

return {
    "studentCount": total_siswa,
    "staffCount": total_gtk,
    "presensi_siswa_today": len(p_siswa),
    "presensi_gtk_today": len(p_gtk),
    "documentsTodayBreakdown": docs_breakdown,
    "status": "success"
}

except Exception as e:
    raise HTTPException(status_code=500, detail=str(e))

```

Kode Program 5.17 stats.py - Endpoint Agregasi Statistik Dashboard

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

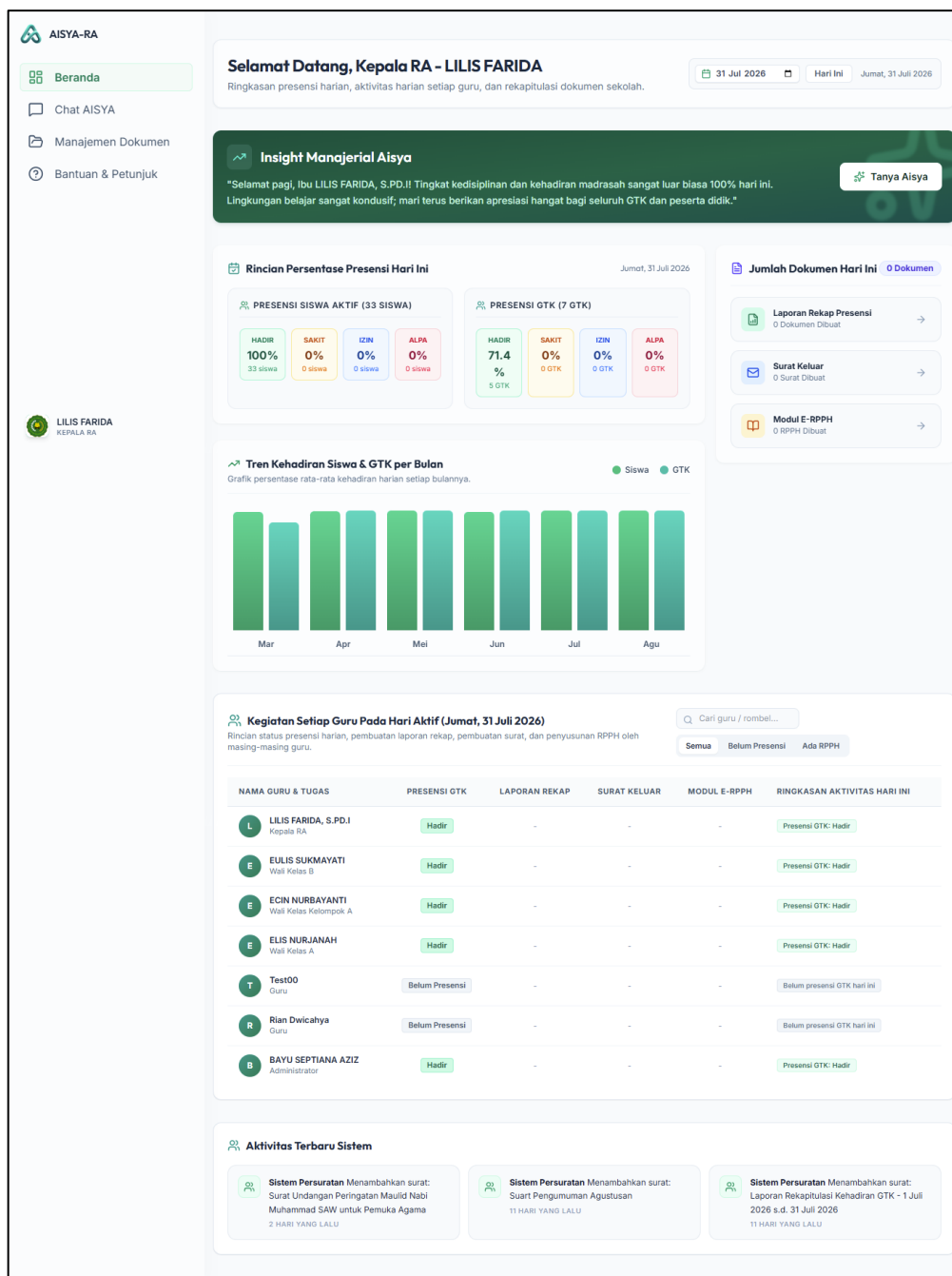
Kode Program 5.17 mengimplementasikan fungsi pengumpulan statistik operasional `get_stats` pada berkas `backend/app/routers/stats.py`. Fungsi ini menerima parameter opsional `target_date` yang memfasilitasi pemfilteran data analitik berdasarkan tanggal tertentu yang dipilih Kepala RA pada antarmuka. Logika backend mengeksekusi kalkulasi agregat jumlah total siswa aktif dan staf GTK terdaftar (`count='exact'`), persentase kehadiran harian siswa dan GTK (`p_siswa` dan `p_gtk`), serta rincian jenis dokumen administrasi yang diterbitkan pada rentang waktu tanggal yang aktif (`documentsTodayBreakdown`: RPPH, surat keluar, dan rekapitulasi kehadiran). Data ini dikirimkan ke klien antarmuka `Dashboard.tsx` untuk dirender secara terpadu.

5.6.2.8 Hasil Implementasi Antarmuka Dashboard Eksekutif

Hasil implementasi *Dashboard* khusus bagi Kepala RA menyajikan pusat pemantauan manajerial yang komprehensif dan *real-time*:

1. **Header dan Fitur Filter Tanggal Aktif (*Date Selector*):** Menyediakan komponen pemilih tanggal (*date picker*) interaktif dan tombol pintas "*Hari Ini*", memungkinkan Kepala RA meninjau kondisi kehadiran serta dokumen yang diterbitkan pada tanggal mana pun secara historis maupun waktu nyata, lengkap dengan indikator status sinkronisasi data (*live date syncing*).
2. **Banner Insight Manajerial AISYA (*Executive AI Insight*):**
Menampilkan ringkasan naratif cerdas hasil analisis otomatis AI mengenai stabilitas operasional, persentase kehadiran, dan saran manajerial hari ini, disertai tombol aksi cepat "*Tanya AISYA*" yang langsung membuka ruang konsultasi *chat*.
3. **Rincian Persentase Presensi Hari Ini (*Detailed Attendance Cards*):**
Menyajikan dua kartu status terpisah untuk presensi siswa aktif dan presensi staf GTK, lengkap dengan kalkulasi persentase dan jumlah jiwa pada status *Hadir*, *Sakit*, *Izin*, dan *Alpa*.
4. **Grafik Tren Kehadiran Siswa & GTK per Bulan (*Monthly Trend Chart*):** Grafik batang visual yang membandingkan persentase rata-rata kehadiran bulanan antara peserta didik dan tenaga kependidikan selama enam bulan terakhir.
5. **Jumlah Dokumen Administrasi yang Dibuat (*Documents Breakdown Sidebar*):** Panel statistik yang merangkum total dokumen terbit pada tanggal aktif terpilih, dengan pembagian kategori: *Laporan Rekap Presensi*, *Surat Keluar*, dan *Modul E-RPPH*, lengkap dengan tombol navigasi instan menuju daftar arsip dokumen terkait.
6. **Tabel Pemantauan Aktivitas Kegiatan Setiap Guru (*Teacher Daily Activity Monitoring Table*):** Tabel analitik mendalam yang memetakan aktivitas harian masing-masing guru kelas pada hari aktif, mencakup kolom nama & jabatan guru, rombongan belajar yang diampu, status presensi GTK

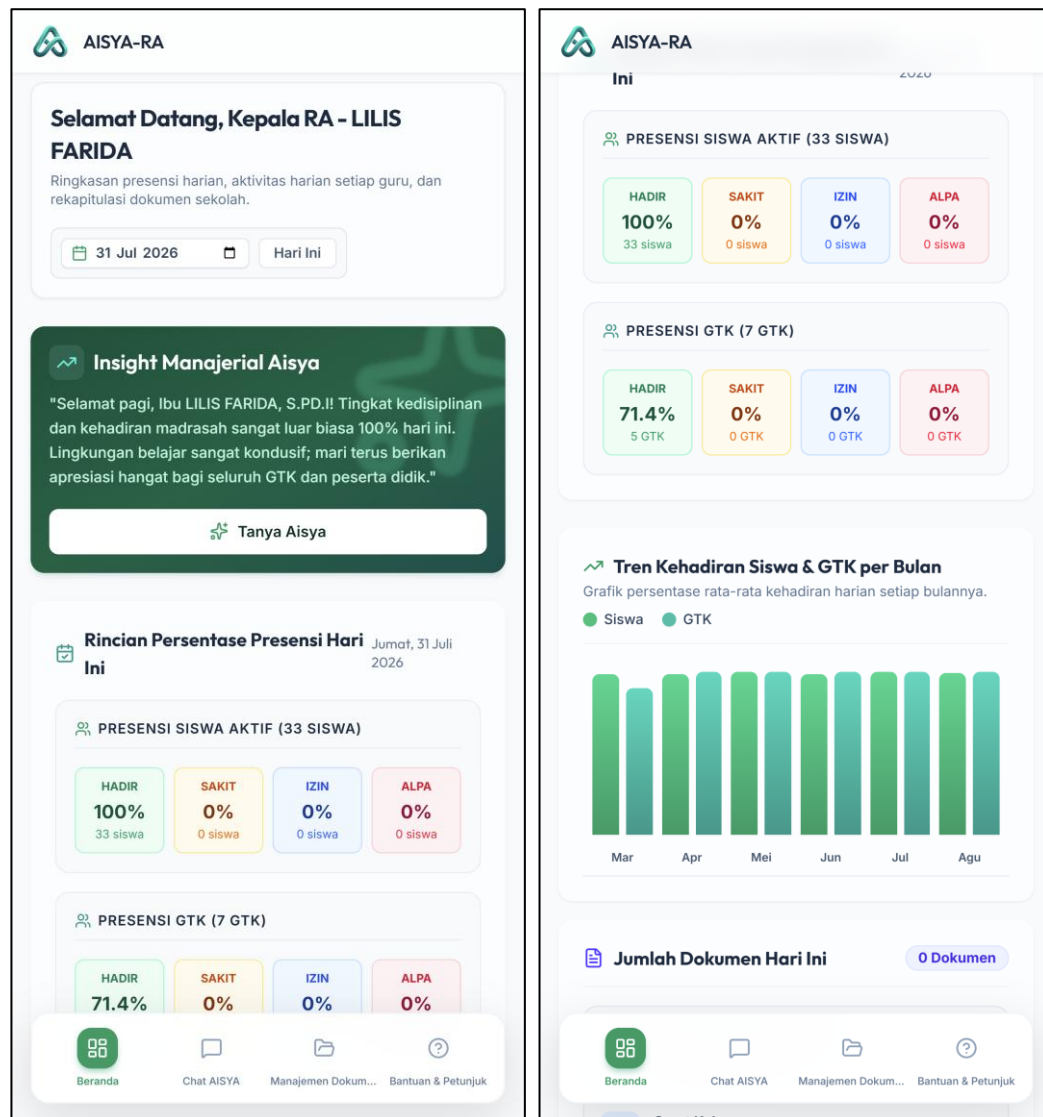
harian, status pembuatan laporan rekap, status pembuatan surat keluar, dan status penyusunan RPPH, dilengkapi fitur pencarian (*search*) nama guru dan penyaringan status (*filter*).



Gambar 5.69 Tampilan Dashboard Kepala RA (Desktop Web Browser)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

Halaman antarmuka *dashboard* yang sama saat diakses melalui peramban *smartphone*, secara responsif menyesuaikan tata letak dan mengubah tampilan menu navigasi supaya layar tidak terlalu sempit, menu navigasi ini ditampilkan menjadi sebuah *floating* menu navigasi yang hanya memuat beberapa item menu, item menu yang lainnya bisa ditampilkan dengan tombol menu “Lainnya”.



Gambar 5.70 Tampilan Dashboard Kepala RA (Diakses dari Smartphone)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

5.6.2.9 Implementasi Registrasi *Router* Utama

Berkas `main.py` mengintegrasikan seluruh *router* modular sistem AISYA-RA dan mengonfigurasi *middleware* CORS.

Kode Program 5.18 menyajikan konfigurasi registrasi *router* utama pada “`main.py`” dan Kode Program 5.19 menyajikan skrip eksekusi server VPS.

```
# backend/app/main.py - Registrasi Router Utama FastAPI
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware
from app.routers import chat, rpph, presensi, surat, stats, knowledge,
sekolah, siswa, pengguna, kalender, tahun_ajaran

app = FastAPI(title="AISYA-RA Backend Bridge")

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

app.include_router(chat.router, prefix="/api", tags=["chat"])
app.include_router(rpph.router, prefix="/api", tags=["rpph"])
app.include_router(presensi.router, prefix="/api", tags=["presensi"])
app.include_router(surat.router, prefix="/api", tags=["surat"])
app.include_router(stats.router, prefix="/api", tags=["stats"])
app.include_router(knowledge.router, prefix="/api", tags=["knowledge"])
app.include_router(sekolah.router, prefix="/api", tags=["sekolah"])
app.include_router(siswa.router, prefix="/api", tags=["siswa"])
app.include_router(pengguna.router, prefix="/api", tags=["pengguna"])
app.include_router(tahun_ajaran.router, prefix="/api",
tags=["tahun_ajaran"])
```

Kode Program 5.18 main.py - Registrasi Router FastAPI

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

Kode Program 5.18 memperlihatkan berkas gerbang utama (*entrypoint*) backend FastAPI `main.py`. Konfigurasi `CORSMiddleware` diterapkan untuk memfasilitasi komunikasi lintas domain antara klien peramban dan peladen API. Seluruh sebelas modul *router* spesifik diintegrasikan secara terstruktur di bawah awalan rute seragam `/api` lengkap dengan pengelompokan dokumentasi OpenAPI (*Swagger UI*), menjamin arsitektur backend yang bersih, tertib, dan mudah dikembangkan lebih lanjut.

Struktur pendaftaran *router* pada `main.py` ini mencerminkan perkembangan modular sistem di setiap iterasi:

1. **Iterasi 1:** Berkas `main.py` hanya mendaftarkan dua modul rute dasar, yaitu rute otentikasi (`/api/auth`) dan rute obrolan teks awal (`/api/chat`).
2. **Iterasi 2:** Mengalami penambahan tiga modul rute administrasi inti, yaitu perute rencana pembelajaran (`/api/rpph`), perute kehadiran siswa (`/api/presensi`), dan perute tata persuratan dinas (`/api/surat`).
3. **Iterasi 3:** Menuntaskan integrasi seluruh layanan sistem dengan menambahkan enam modul rute baru, mencakup dasbor eksekutif (`/api/stats`), basis pengetahuan RAG (`/api/knowledge`), data lembaga (`/api/sekolah`), master siswa (`/api/siswa`), master GTK (`/api/pengguna`), kalender akademik (`/api/kalender`), serta pengelolaan tahun pelajaran (`/api/tahun_ajaran`).

5.6.2.10 Implementasi Skrip Deployment Cloud *Production*

tomatisasi pembaruan dan orkestrasi kontainer di server VPS DigitalOcean dijalankan melalui skrip bash *deployment*.

```
#!/bin/bash
# deploy.sh - Skrip Deployment Otomatis Server VPS Cloud
set -e
echo "Memulai proses deployment AISYA-RA ke Server Produksi..."

git pull origin main
```

```
docker compose down
docker compose up --build -d

echo "Deployment berhasil! Sistem AISYA-RA aktif berjalan di
https://aisya.ra-alislam.sch.id"
```

Kode Program 5.19 bash - Skrip Deployment Production

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

Kode Program 5.19 menyajikan skrip otomatisasi peluncuran sistem `deploy.sh` pada lingkungan produksi VPS DigitalOcean. Perintah “set -e” memastikan eksekusi skrip segera berhenti jika terjadi galat di salah satu tahapan. Alur skrip mengeksekusi pembaruan basis kode dari cabang utama repositori Git (“git pull origin main”), menghentikan kontainer lama (“docker compose down”), dan membangun ulang citra kontainer termutakhir dalam mode latar belakang (“docker compose up --build -d”), sehingga proses pembaruan rilis sistem dapat terlaksana secara andal dan minim *downtime*.

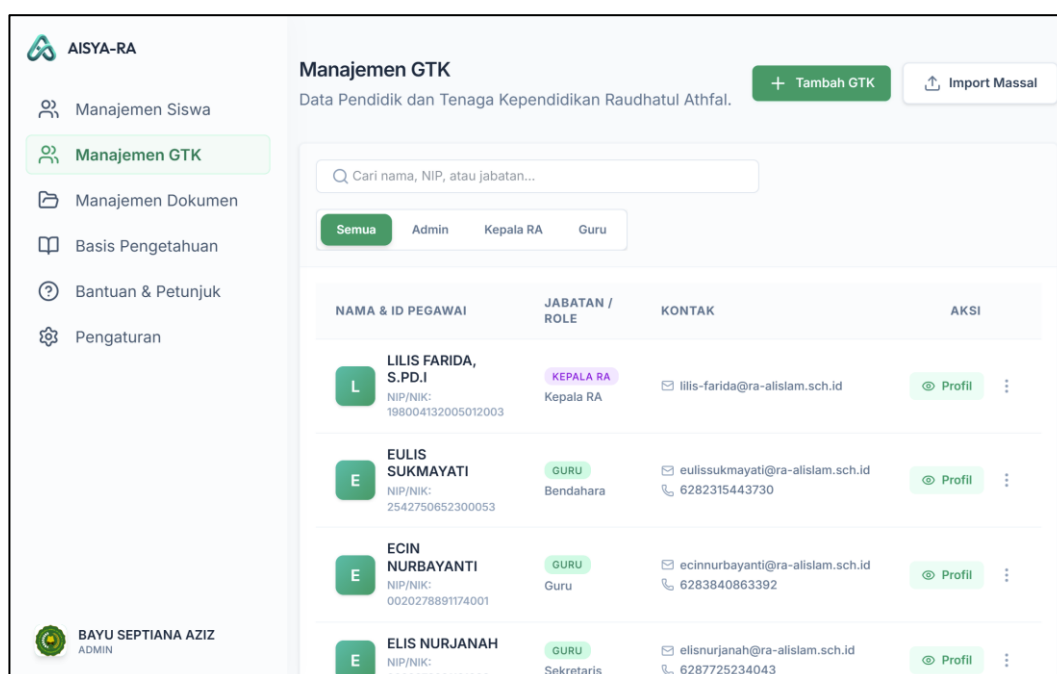
5.6.2.11 Implementasi Antarmuka Admin (GUI).

Implementasi antarmuka untuk aktor Admin dirancang menggunakan pendekatan tabel dan formulir visual (GUI) standar. Format ini dinilai paling efektif untuk mengakomodasi kebutuhan input data massal seperti pendaftaran siswa dan pengunggahan berkas PDF yang menuntut presisi struktural. Data yang di-input oleh Admin melalui antarmuka ini akan disimpan di basis data relasional, yang kemudian menjadi sumber konteks otomatis (Super-Contextual Prompt) bagi mesin LLM saat merespons instruksi dari pengguna lain.

Implementasi antarmuka untuk peran Admin dirancang menggunakan pendekatan GUI berbasis tabel dan formulir terstruktur. Antarmuka ini mengelola basis data primer yang menjadi sumber konteks otomatis bagi mesin AI:

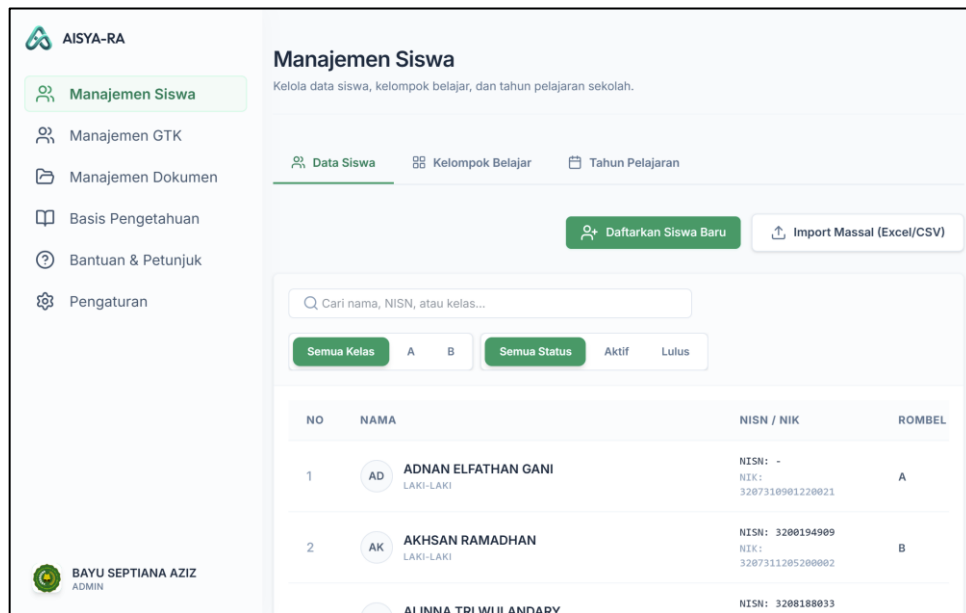
- **Pengelolaan Data GTK (Staff.tsx):** Fasilitas manajemen akun pengajar dan staf kependidikan, penugasan wali kelas, dan ubah profil kepegawaian.

- **Pengelolaan Data Peserta Didik (Students.tsx):** Fasilitas manajemen data siswa, filter kelompok belajar, kenaikan kelas, dan impor data massal dari spreadsheet.
- **Pengelolaan Basis Pengetahuan RAG (KnowledgeBase.tsx):** Fasilitas pengunggahan berkas PDF panduan kurikulum/SOP madrasah, pemantau status indeks vektor 768-dimensi, dan manajemen repositori dokumen.
- **Pengaturan Profil Lembaga (Settings.tsx):** Fasilitas untuk mengubah profil lembaga.



Gambar 5.71 Admin GUI - Manajemen Guru dan Tenaga Kependidikan

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



Gambar 5.72 GUI Admin - Manajemen Siswa

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

Siswa Baru
Pendaftaran Pendidik

BIODATA DIRI
Identitas & Status

ORANG TUA
Ayah, Ibu & Wali

KONTAK & ALAMAT
Domisili & Telepon

NAMA LENGKAP: Sesuai Akte Kelahiran

NIK (SESUAI KK): Nomor Induk Kependudukan

NISN (10 DIGIT): Opsional - 10 digit

JENIS KELAMIN: LAKI-LAKI, PEREMPUAN

KELOMPOK BELAJAR (KELAS): A

STATUS KEAKTIFAN: Aktif Belajar

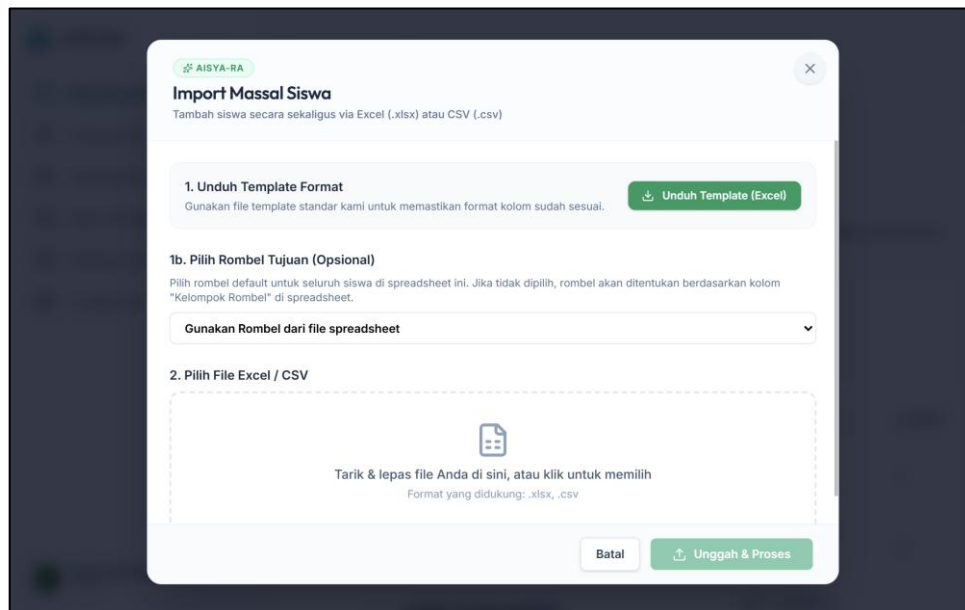
TEMPAT LAHIR: Contoh: Jakarta

TANGGAL LAHIR: dd --- yyyy

Batal Simpan Data

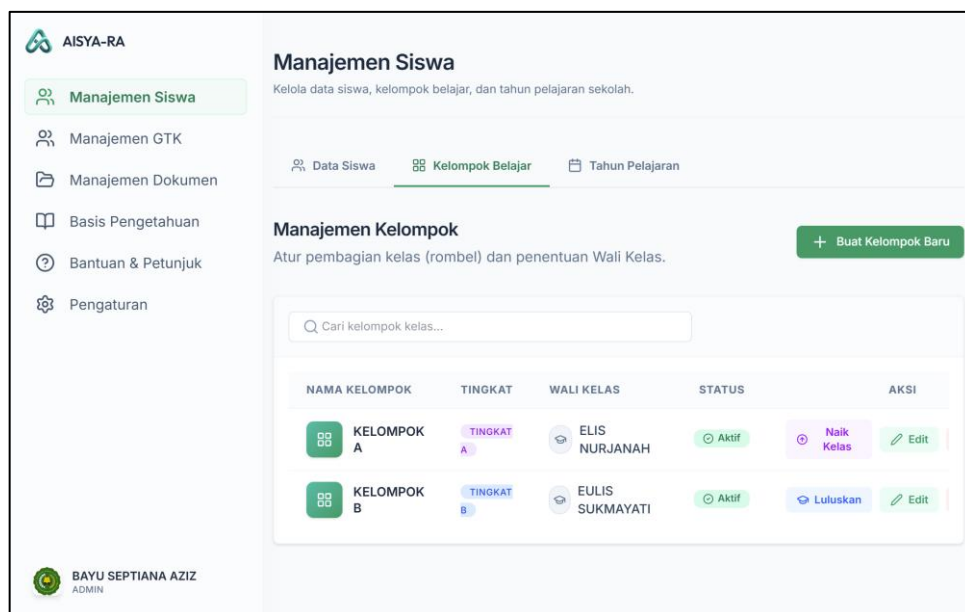
Gambar 5.73 Admin GUI - Manajemen Siswa (Tambah Siswa)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



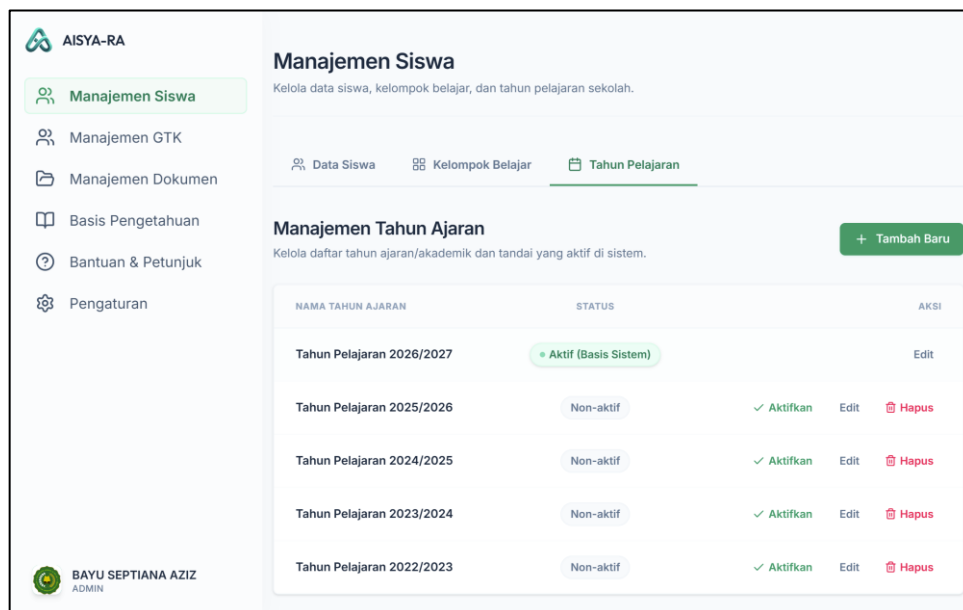
Gambar 5.74 Admin GUI - Manajemen Siswa (Import Massal Siswa)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



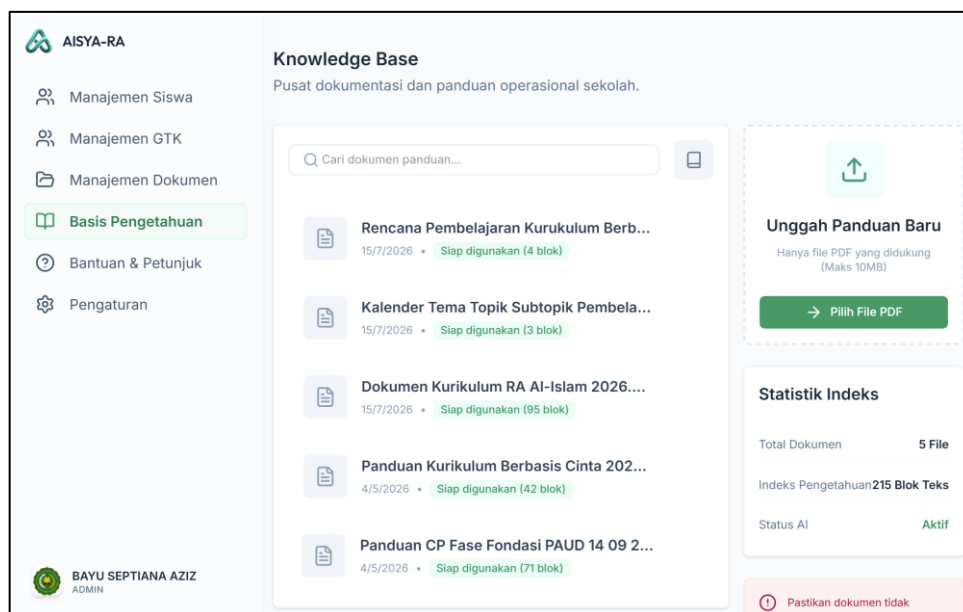
Gambar 5.75 Admin GUI - Manajemen Siswa (Kelompok Belajar)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



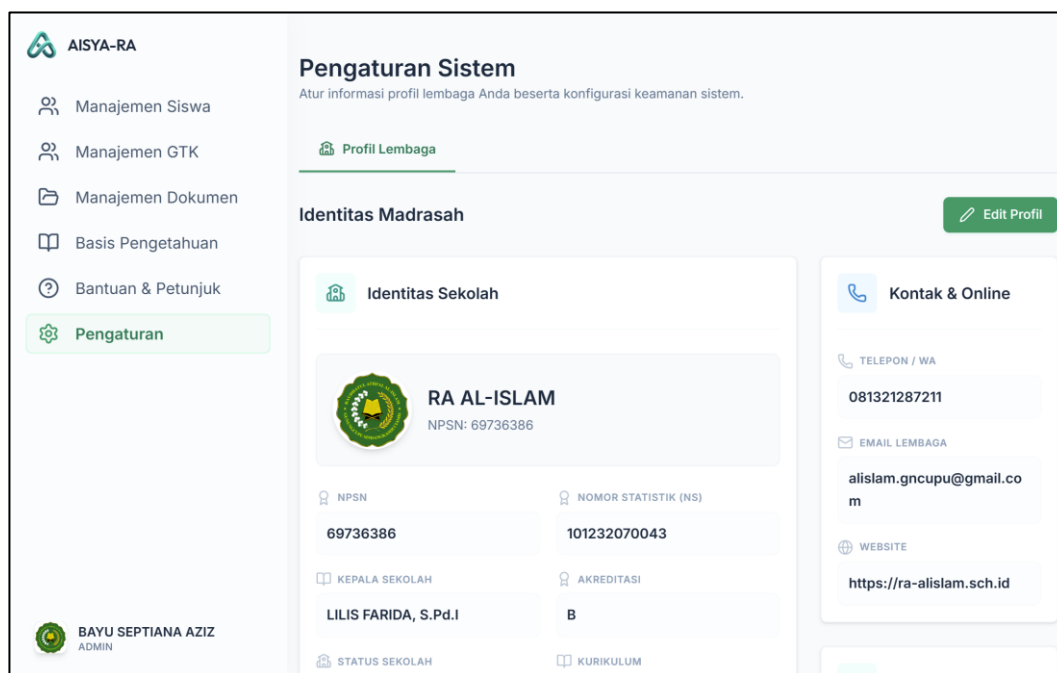
Gambar 5.76 Admin GUI - Manajemen Siswa (Tahun Pelajaran)

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



Gambar 5.77 Admin GUI - Basis Pengetahuan

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)



Gambar 5.78 Admin GUI - Pengeturan Profil Lembaga

(Sumber: Hasil pengembangan tahap develop iterasi 3 AISYA-RA, 2026)

5.6.3 Test Iterasi 3 – Black Box Testing

Berbeda dengan Iterasi 1 dan 2 yang diuji terbatas di jaringan Wi-Fi lokal, pengujian pada Iterasi 3 dilaksanakan secara komprehensif langsung pada lingkungan produksi (*production environment*) di server *cloud* VPS “<https://aisya.ra-alislam.sch.id>”. Pengujian bertujuan untuk memvalidasi keandalan fungsional modul *Data Master*, Presensi GTK, Basis Pengetahuan, *Dashboard*, serta memastikan tidak terjadinya degradasi fungsional (*regression*) pada fitur-fitur yang dibangun pada iterasi sebelumnya.

Pengujian dilaksanakan menggunakan perangkat laptop peneliti dan *smartphone* (*Google Chrome Mobile*) sebagai *test client*. Seluruh skenario diuji menggunakan dua akun uji aktual, satu akun Guru (R-02) dan satu akun Kepala RA (R-01), keduanya telah terverifikasi di *database* Supabase Cloud.

Tabel 5.14 Hasil Black Box Testing Iterasi 3

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
TB-19	Injeksi otomatis data master ke RPPH	Header draf RPPH otomatis memuat nama RA, NPSN, semester, dan nama guru.	Dokumen RPPH otomatis memuat identitas lengkap madrasah tanpa diketik manual.	Lulus
TB-20	Injeksi kop surat resmi otomatis	Draf surat dinas otomatis memuat kop resmi, logo lembaga, dan titimangsa Kepala RA.	Surat keluar memuat susunan kop resmi madrasah secara sempurna.	Lulus
TB-21	Unggah & pengindeksan dokumen PDF RAG	Berkas PDF panduan kurikulum berhasil diekstrak dan tersimpan sebagai vektor 768-dimensi.	Potongan teks dan vektor embedding berhasil terindeks di tabel knowledge_chunks.	Lulus
TB-22	Tanya jawab regulasi via chat (RAG)	Chatbot menjawab pertanyaan kurikulum/SOP berbasis teks dokumen lokal yang diunggah.	Jawaban asisten AI akurat dan mencantumkan rujukan pedomannya madrasah.	Lulus

ID Uji	Skenario Pengujian	Hasil yang Diharapkan		Hasil Pengujian	Status
TB-23	Pencatatan presensi GTK via chat	Kepala RA mencatat kehadiran guru via <i>chat</i> dan tersimpan ke tabel presensi_gtk.		Kehadiran staf berhasil tercatat dan konfirmasi pesan muncul di chat.	Lulus
TB-24	Membuat rekap presensi GTK via chat	<i>Chatbot</i> mengagregasi data absensi bulanan staf ke dalam tabel rekapitulasi.		Tabel rekap presensi GTK berhasil dibangkitkan dan dapat dibuka di canvas.	Lulus
TB-25	Tampilan statistik Dashboard	Halaman dashboard menampilkan kartu kehadiran, total dokumen, dan grafik tren secara akurat.		Seluruh kartu metrik dan grafik statistik termuat sesuai data transaksi riil.	Lulus
TB-26	Manajemen <i>Data Master</i> Siswa & GTK	Admin dapat menambah, mengedit, menghapus, dan mengimpor data siswa/GTK dari Excel.		Operasi CRUD dan impor spreadsheet berhasil memperbarui pangkalan data master.	Lulus
TB-27	Konfigurasi Profil	Admin dapat memperbarui		Data profil madrasah berhasil diperbarui dan	Lulus

ID Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
	Lembaga di Settings	alamat, visi-misi, kontak, dan kunci API Gemini.	tersimpan permanen di Supabase.	

(Sumber: Hasil pengujian Black Box Testing iterasi 3 AISYA-RA, 2026)

Seluruh sepuluh skenario pengujian dinyatakan lulus. Integrasi *Data Master* terbukti mampu mengotomatiskan pencetakan *header* lembaga pada RPPH dan Surat Keluar secara konsisten. Mekanisme *upsert* pada tabel “presensi_gtk” berhasil mencegah timbulnya data kehadiran ganda pada hari yang sama. *Pipeline* RAG terbukti mampu menjawab pertanyaan kebijakan lokal sekolah dalam waktu rata-rata 4 hingga 6 detik berbasis kemiripan semantik pgvector. Antarmuka *dashboard* menyajikan agregasi statistik yang akurat sesuai dengan kondisi basis data terkini. Aplikasi berhasil diakses secara publik melalui jaringan seluler dengan enkripsi HTTPS yang stabil. Pengujian regresi mengonfirmasi bahwa penambahan modul baru tidak mengganggu fungsionalitas yang telah dibangun pada Iterasi 1 dan 2. Dengan demikian, seluruh kriteria *Definition of Done* (DoD) Iterasi 3 dinyatakan terpenuhi

5.6.4 Deploy Iterasi 3 – Deployment Operasional

Tahap *Deploy* pada Iterasi 3 merealisasikan rilis penuh sistem AISYA-RA dari lingkungan pengujian lokal terbatas ke lingkungan produksi komputasi awan (*Production Cloud Environment*). Peluncuran ini bertujuan agar seluruh pemangku kepentingan (Kepala RA, Guru RA, dan Admin) dapat mengakses sistem secara berkelanjutan 24 jam sehari dari mana saja melalui jaringan internet publik dengan standar keamanan terenkripsi.

Rangkaian prosedur *deployment* produksi mencakup tahapan teknis berikut:

1. **Penyediaan Server Virtual (*Cloud VPS Provisioning*):** Infrastruktur server dibangun menggunakan layanan *Virtual Private Server (VPS)* DigitalOcean Droplet yang berlokasi di data center Region Singapura

(SGP1) guna meminimalkan latensi jaringan (*round-trip time*) ke Indonesia. Peladen dialokasikan spesifikasi komputasi 1 vCPU, 2 GB RAM, 50 GB penyimpanan *Solid State Drive* (NVMe SSD), dan menjalankan sistem operasi Ubuntu 24.04 LTS x64 dengan perlindungan *firewall* jaringan tingkat peladen.

2. **Konfigurasi Domain dan DNS (Domain Name System):** Sistem dihubungkan dengan sub-domain resmi lembaga madrasah: <https://aisya.ra-alislam.sch.id>. Rekam *DNS A Record* dikonfigurasi pada panel registrar DNS untuk memetakan alamat sub-domain secara langsung ke alamat IP publik peladen VPS.
3. **Penerbitan dan Otomatisasi Sertifikat Keamanan SSL/TLS:** Untuk mengamankan saluran transmisi data dan enkripsi panggilan API dari intersepsi pihak ketiga (*Man-In-The-Middle Attack*), sertifikat SSL/TLS diterbitkan menggunakan otoritas sertifikat *Let's Encrypt* melalui Certbot dengan enkripsi RSA 2048-bit. Konfigurasi Nginx diatur untuk secara otomatis mengalihkan seluruh lalu lintas HTTP biasa (port 80) menuju saluran terenkripsi HTTPS (port 443), disertai mekanisme pembaruan sertifikat otomatis (*auto-renewal cron job*).
4. **Otomatisasi Skrip Peluncuran Produksi (deploy.sh):** Penerapan pembaruan sistem diotomatisasi melalui eksekusi skrip `deploy.sh`. Skrip ini melakukan sinkronisasi kode dari repositori cabang utama (`git pull origin main`), memvalidasi integritas variabel lingkungan produksi pada berkas `.env`, menghentikan kontainer lama secara mulus, dan mengeksekusi pembangunan kontainer baru (`docker compose up --build -d`). Konfigurasi kontainer diperkuat dengan klausul `restart: unless-stopped` untuk menjamin bahwa seluruh kontainer otomatis hidup kembali apabila peladen mengalami *reboot* atau lonjakan beban sesaat.
5. **Verifikasi Kesiapan Operasional Global (Production Verification):** Setelah proses peluncuran selesai, dilakukan pengujian akses dari berbagai

jaringan eksternal (koneksi seluler 4G/5G Telkomsel, Indosat, XL, serta jaringan internet kabel rumah). Pengujian mengonfirmasi bahwa seluruh modul sistem, termasuk *Super-Contextual System Instruction*, pencarian semantik vektor RAG pada basis data Supabase *pgvector*, pencatatan presensi GTK, dan agregasi *Dashboard* eksekutif—berjalan secara stabil dengan rata-rata waktu respons di bawah 1,5 detik. Seluruh pengguna akhir kini dapat memanfaatkan sistem AISYA-RA secara penuh dalam kegiatan operasional madrasah sehari-hari.

5.6.5 Review Iterasi 3

Sesi *review* ini dilaksanakan sebagai tahap evaluasi pencapaian hasil pengembangan terhadap penambahan modul *Data Master*, Presensi GTK via *Chatbot*, Basis Pengetahuan (*RAG Knowledge Base*), dan *Dashboard*. Sesi *review* ini dilaksanakan selama dua jam di ruang rapat RA Al-Islam Gunungcupu, yang dihadiri oleh Kepala RA (R-01), Guru Kelompok A (R-02), Guru Kelompok B (R-03), dan Guru Pendamping (R-04).

Berbeda dengan dua iterasi sebelumnya yang diuji terbatas pada prototipe jaringan Wi-Fi lokal, sesi *review* ini memberikan kesempatan bagi para pengampu untuk menguji coba secara langsung platform AISYA-RA yang telah aktif di server produksi (cloud VPS) melalui domain resmi “<https://aisya.ra-alislam.sch.id>”.

Sesi diawali dengan pengujian otomatisasi pembuatan dokumen beridentitas *Data Master*. Guru Kelompok A (R-02) menguji pembuatan RPPH harian kelompok kelasnya via *Chatbot*. R-02 terkesan ketika mendapati bahwa draf RPPH yang dihasilkan LLM tidak lagi berupa draf generik seperti pada Iterasi 2, melainkan telah memuat *header* resmi RA Al-Islam Gunungcupu, nama guru pengampu, kelompok kelas, serta tanggal pelaksanaan secara presisi tanpa perlu pengisian manual. R-02 menyampaikan kesannya dengan penuh kepuasan:

"Nah, kalau penampilannya seperti ini mah bagus banget! Waktu review Iterasi 2 kemarin kan saya masih harus nulis ulang nama sekolah dan nama saya di bagian atas. Sekarang tinggal kirim pesan 'AISYA, buat RPPH tema Cita-Citaku',

langsung keluar lengkap ada kop resmi RA Al-Islam dan nama saya. Benar-benar bikin kerjaan kami jadi jauh lebih cepat."

— R-02, Guru Kelompok A, Sesi *Review* Iterasi 3

Selanjutnya, Kepala RA (R-01) mengevaluasi pengoperasian *Dashboard* dan modul Basis Pengetahuan. R-01 mencoba mengajukan pertanyaan konsultatif mengenai tema dan subtema pembelaran pada hari itu. Ketika *Chatbot* merespons secara akurat dengan mengutip pasal-pasal dari berkas Pedoman Kurikulum yang telah diunggah sebelumnya, Beliau mengapresiasi tinggi kecerdasan emosional dan relevansi konteks lokal sistem. R-01 memberikan penegasan evaluasinya:

"Hasil di Iterasi 3 ini luar biasa sekali. Sebelum ada AISYA-RA, nyari dokumen pedoman sekolah dan bikin rekap presensi mah selalu menyita waktu. Sekarang lewat tampilan dashboard saya bisa mantau kehadiran guru dan siswa secara langsung, terus kalau ada pertanyaan kurikulum, AI-nya langsung jawab sesuai dokumen sekolah kita. Sangat membantu operasional sekolah kami."

— Kepala RA (R-01), Sesi *Review* Iterasi 3

Guru Pendamping (R-04) turut memberikan ulasan mengenai kemudahan penggunaan antarmuka percakapan (*conversational interface*) yang dinilai sangat ramah bagi guru:

"Awalnya mah kami para guru sempat takut kalau sistem baru ini bakal rumit dan nambah beban. Tapi karena bentuknya *chat* kayak WhatsApp biasa, kami tinggal ngetik pesan buat nyatat presensi sama bikin surat. Kami merasa terbantu sekali."

— R-04, Guru Pendamping, Sesi *Review* Iterasi 3

Sesi *review* ini ditutup dengan kesepakatan bersama bahwa seluruh item *Product Backlog* yang dirancang hingga Iterasi 3 telah terealisasi sepenuhnya dan memenuhi seluruh kriteria *Definition of Done* (DoD). Penutupan sesi ini secara resmi menandai selesainya siklus pengulangan pengembangan pada tahapan *Agile* akhir (*final Agile iteration stage*), di mana platform AISYA-RA dinyatakan stabil

dan beroperasi penuh di lingkungan produksi untuk selanjutnya memasuki tahap pengujian dan evaluasi sistem (*System Evaluation*).

5.7 Evaluasi Sistem

Evaluasi sistem AISYA-RA dilakukan setelah semua iterasi selesai dan sistem digunakan secara operasional oleh mitra selama minimal 30 hari. Evaluasi mencakup dua aspek, efisiensi waktu administrasi, dan penerimaan pengguna menggunakan *System Usability Scale (SUS)*.

5.7.1 Evaluasi Efisiensi Waktu Administrasi

Evaluasi efisiensi waktu dilakukan dengan membandingkan rata-rata durasi pengerjaan setiap jenis tugas administrasi sebelum dan sesudah menerapkan sistem AISYA-RA. Pengukuran kondisi sebelum penerapan sistem diperoleh dari hasil observasi terstruktur dan wawancara lapangan (Tabel 5.1). Pengukuran sesudah penerapan sistem dilakukan melalui observasi waktu respons riil selama periode penggunaan operasional.

Tingkat efisiensi dihitung menggunakan rumus persentase penghematan waktu:

$$\text{Penghematan} = \left(\frac{\text{Waktu Sebelum} - \text{Waktu Sesudah}}{\text{Waktu Sebelum}} \right) \times 100\%$$

Berdasarkan rumus tersebut, berikut adalah rincian langkah perhitungan efisiensi waktu untuk masing-masing tugas administrasi:

1. Penyusunan RPPH

- a. Waktu Sebelum = 45 menit, Waktu Sesudah = 1 menit
- b. Penghematan = $\left((45 - 1)/45 \right) \times 100\% = (44/45) \times 100\% \approx 97,8\%$

2. Pencatatan presensi siswa per kelas

- a. Waktu Sebelum = 5 menit, Waktu Sesudah = 1 menit
- b. Penghematan = $\left((5 - 1)/5 \right) \times 100\% = (4/5) \times 100\% = 80,0\%$

3. Rekapitulasi presensi siswa (mingguan/bulanan)

- a. Waktu Sebelum = 45 menit, Waktu Sesudah = 1 menit
- b. Penghematan = $((45 - 1)/45) \times 100\% = (44/45) \times 100\% \approx 97,8\%$

4. Pencatatan presensi GTK (guru dan kepala)

- a. Tugas ini merupakan pencatatan baru yang belum pernah dilakukan sebelum adanya AISYA-RA. Oleh karena itu, persentase penghematan tidak dapat dihitung dan tidak dimasukkan ke dalam perhitungan rata-rata.

5. Pembuatan surat resmi (kop dan nomor)

- a. Waktu Sebelum = 35 menit, Waktu Sesudah = 1 menit
- b. Penghematan = $((35 - 1)/35) \times 100\% = (34/35) \times 100\% \approx 97,1\%$

6. Pencarian arsip dokumen

- a. Waktu Sebelum = 15 menit, Waktu Sesudah = 1 menit
- b. Penghematan = $((15 - 1)/15) \times 100\% = (14/15) \times 100\% \approx 93,3\%$

Untuk mendapatkan rata-rata penghematan keseluruhan, akumulasi persentase dari kelima tugas terukur ($97,8\% + 80,0\% + 97,8\% + 97,1\% + 93,3\% = 466,0\%$) dibagi dengan jumlah tugas (5), sehingga menghasilkan rata-rata penghematan sebesar **93,2%**. Hasil perhitungan secara lengkap disajikan pada Tabel 5.15.

Tabel 5.15 Pengukuran Evaluasi Efisiensi Waktu Administrasi AISYA-RA

No	Jenis Tugas Administrasi	Sebelum	Sesudah	Penghematan (%)
		AISYA-RA (menit)	AISYA-RA (menit)	
1	Penyusunan RPPH	45	1	97,8%

No	Jenis Tugas Administrasi	Sebelum AISYA-RA (menit)	Sesudah AISYA-RA (menit)	Penghematan (%)
2	Pencatatan presensi siswa per kelas	5	1	80,0%
3	Rekapitulasi presensi siswa (mingguan/bulanan)	45	1	97,8%
4	Pencatatan presensi GTK (guru dan kepala)	— (Tidak ada)	1	— (Tugas baru)
5	Pembuatan surat resmi (kop dan nomor)	35	1	97,1%
6	Pencarian arsip dokumen	15	1	93,3%
Rata-rata penghematan (tugas terukur)				93,2%

(Sumber: Hasil pengukuran evaluasi efisiensi waktu administrasi , 2026)

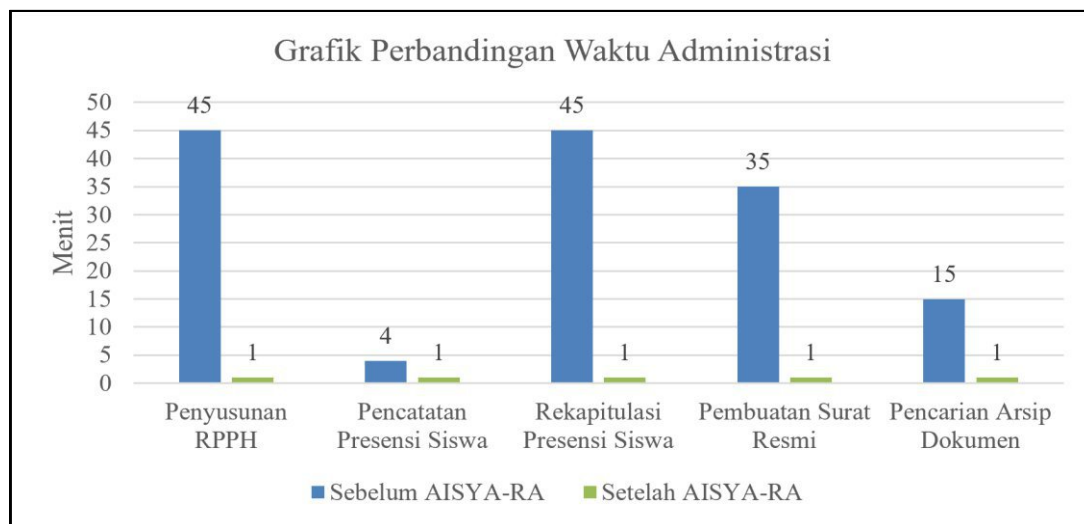
Berdasarkan Tabel 5.15, AISYA-RA terbukti mampu memberikan dampak penghematan durasi kerja yang sangat signifikan pada berbagai tugas administratif operasional di RA Al-Islam Gunungcupu. Penghematan pada pembuatan Rencana Pelaksanaan Pembelajaran Harian (RPPH) berasal dari dua lapisan utama, yaitu otomatisasi struktur format dokumen baku dan kemampuan sistem dalam mengenerate komponen kegiatan inti pembelajaran secara langsung.

Perbedaan signifikansi penghematan ini sangat jelas jika membandingkan proses manual dengan integrasi Large Language Model (LLM). Pada metode manual, proses penyusunan RPPH memakan waktu rata-rata 40 hingga 55 menit karena tingginya beban kognitif guru yang harus memikirkan dan mengetik ulang format dari awal menggunakan perangkat lunak pengolah kata yang sering kali formatnya berubah. Sebaliknya, dengan pemanfaatan LLM, guru hanya perlu memberikan instruksi singkat berbasis bahasa alami (contoh: "Buatkan RPPH tema Binatang Peliharaan untuk Kelompok A"). Model LLM mengambil alih beban komputasi tersebut dan menghasilkan draf dokumen yang secara otomatis menyesuaikan dengan 7 komponen baku kurikulum lembaga hanya dalam hitungan detik. Waktu guru yang tersisa kini dialihkan untuk membaca pratinjau (*preview*) dan sekadar melakukan penyesuaian minor.

Pada baris nomor 4 (Pencatatan presensi GTK) memiliki catatan khusus, di mana sebelumnya proses ini tidak pernah dilakukan secara formal dan tidak memiliki durasi pembandingan secara manual. Oleh karena itu, fitur ini tidak sekadar menghasilkan persentase penghematan, melainkan memberikan nilai tambah administratif (*value proposition*) baru yang meningkatkan akuntabilitas kehadiran staf di madrasah.

Perbandingan efisiensi yang sama tingginya juga berlaku pada otomatisasi pembuatan surat resmi. Pada proses manual yang bisa memakan waktu hingga 40 menit, Kepala RA atau guru harus mencari berkas arsip fisik untuk memastikan nomor surat terakhir agar tidak terjadi duplikasi, lalu mengetik ulang bagian kop dan mengarang narasi isi surat. Melalui kemampuan penalaran teks pada LLM, sistem dapat mengenali intensi pembuatan surat, menyusun narasi isi surat secara formal, dan sistem backend secara otomatis menyisipkan nomor urut surat terbaru beserta kop surat resmi madrasah tanpa kesalahan pengetikan. Pengambilalihan proses perumusan kalimat dan penelusuran arsip data oleh LLM inilah yang memangkas waktu operasional secara drastis, memungkinkan tenaga pendidik untuk kembali

berfokus pada interaksi pedagogis bersama peserta didik. Grafik perbandingan efisiensi waktu disajikan pada Gambar 5.79.



Gambar 5.79 Grafik Perbedaan Waktu Administrasi
(Sumber: Hasil pengukuran evaluasi efisiensi waktu administrasi , 2026)

5.7.2 Hasil System Usability Scale(SUS)

Pengukuran tingkat penerimaan pengguna terhadap sistem dilakukan menggunakan instrumen *System Usability Scale (SUS)* yang terdiri dari sepuluh butir pertanyaan, sebagaimana dirancang pada Tabel 4.4. Kuesioner ini didistribusikan kepada Kepala RA dan seluruh tenaga pendidik setelah sistem diimplementasikan secara operasional selama minimal 30 hari pasca-Iterasi 3. Pengukuran respons pengguna menggunakan skala Likert dengan rentang nilai 1 hingga 5.

Penilaian akhir SUS diperoleh melalui tahapan perhitungan matematis yang baku. Skor kontribusi untuk setiap butir pertanyaan dihitung dengan ketentuan sebagai berikut:

1. Untuk pernyataan bernada positif (item ganjil 1, 3, 5, 7, dan 9), skor kontribusi adalah nilai skala jawaban dikurangi 1, atau dirumuskan sebagai $X - 1$.

2. Untuk pernyataan bernada negatif (item genap 2, 4, 6, 8, dan 10), skor kontribusi adalah 5 dikurangi nilai skala jawaban, atau dirumuskan sebagai $5 - X$.

Nilai keseluruhan (*Total Score*) untuk masing-masing responden didapatkan dengan mengakumulasi seluruh skor kontribusi, yang kemudian dikalikan dengan konstanta 2,5 untuk mentransformasi skor ke dalam rentang 0 hingga 100. Secara matematis, rumusan tersebut dituliskan sebagai berikut:

$$\text{Skor Akhir SUS} = \left(\sum (X_{\text{ganjil}} - 1) + \sum (5 - X_{\text{genap}}) \right) \times 2,5$$

Berdasarkan formula baku pengujian SUS, berikut adalah penjabaran langkah-langkah perhitungan skor akhir untuk keempat responden (R1 hingga R4) berdasarkan data kuesioner yang telah dikumpulkan:

1. Perhitungan Responden 1 (R1 - Kepala RA)

- a. Skor Butir Ganjil (1, 3, 5, 7, 9): 5, 5, 5, 5, 5

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

- b. Skor Butir Genap (2, 4, 6, 8, 10): 1, 1, 1, 1, 1

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

- c. Total Nilai Keseluruhan: $20 + 20 = 40$

- d. Skor Akhir SUS R1: $40 \times 2,5 = 100$

2. Perhitungan Responden 2 (R2 - Guru Kelompok A)

- a. Skor Butir Ganjil (1, 3, 5, 7, 9): 5, 5, 4, 5, 5

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (4 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 3 + 4 + 4 = 19$$

- b. Skor Butir Genap (2, 4, 6, 8, 10): 1, 1, 1, 1, 1

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

$$\text{c. Total Nilai Keseluruhan: } 19 + 20 = 39$$

$$\text{d. Skor Akhir SUS R2: } 39 \times 2,5 = 97,5$$

3. Perhitungan Responden 3 (R3 - Guru Kelompok B)

$$\text{a. Skor Butir Ganjil (1, 3, 5, 7, 9): } 5, 5, 5, 5, 5$$

$$\text{b. Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

$$\text{c. Skor Butir Genap (2, 4, 6, 8, 10): } 1, 1, 1, 1, 1$$

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

$$\text{d. Total Nilai Keseluruhan: } 20 + 20 = 40$$

$$\text{e. Skor Akhir SUS R3: } 40 \times 2,5 = 100$$

4. Perhitungan Responden 4 (R4 - Guru Pendamping / Tenaga Kependidikan)

$$\text{a. Skor Butir Ganjil (1, 3, 5, 7, 9): } 4, 5, 4, 4, 4$$

$$\text{Nilai Kontribusi: } (4 - 1) + (5 - 1) + (4 - 1) + (4 - 1) + (4 - 1) = 3 + 4 + 3 + 3 + 3 = 16$$

$$\text{b. Skor Butir Genap (2, 4, 6, 8, 10): } 1, 1, 1, 1, 1$$

$$\text{Nilai Kontribusi: } (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) + (5 - 1) = 4 + 4 + 4 + 4 + 4 = 20$$

$$\text{c. Total Nilai Keseluruhan: } 16 + 20 = 36$$

$$\text{d. Skor Akhir SUS R4: } 36 \times 2,5 = 90,0$$

Setelah mendapatkan skor akhir SUS untuk masing-masing responden, tingkat penerimaan sistem secara keseluruhan dievaluasi dengan mencari nilai rata-rata dari seluruh populasi pengujian. Perhitungannya adalah sebagai berikut:

$$\text{Rata-rata Skor SUS} = \frac{100 + 97,5 + 100 + 90,0}{4}$$

$$\text{Rata-rata Skor SUS} = \frac{387,5}{4}$$

$$\text{Rata-rata Skor SUS} = 96,875 \approx \mathbf{96,88}$$

Berdasarkan hasil perhitungan komprehensif tersebut, diperoleh skor rata-rata akhir SUS sebesar **96,88** yang memvalidasi bahwa sistem AISYA-RA berada pada kategori kelayakan "Sangat Baik" (*Excellent*).

Hasil tabulasi data kuesioner dan perhitungan secara komprehensif disajikan pada Tabel 5.14.

Tabel 5.16 Hasil Kuesioner System Usability Scale (SUS) AISYA-RA

No	Pertanyaan SUS	R-01	R-02	R-03	R-04	Rata-Rata
1	Saya pikir saya ingin sering menggunakan sistem AISYA-RA.	5	5	5	4	4,75
2	Saya merasa sistem AISYA-RA terlalu rumit untuk digunakan.	1	1	1	1	1,00
3	Saya pikir sistem AISYA-RA mudah digunakan.	5	5	5	5	5,00
4	Saya membutuhkan bantuan teknis untuk dapat menggunakan AISYA-RA.	1	1	1	1	1,00
5	Saya menemukan berbagai fungsi dalam	5	4	5	4	4,50

No	Pertanyaan SUS	R-01	R-02	R-03	R-04	Rata-Rata
	AISYA-RA terintegrasi dengan baik.					
6	Saya pikir terlalu banyak ketidakconsistenan dalam sistem AISYA-RA.	1	1	1	1	1,00
7	Saya merasa orang lain akan belajar menggunakan AISYA-RA dengan sangat cepat.	5	5	5	4	4,75
8	Saya merasa AISYA-RA sangat rumit dan memerlukan upaya ekstra.	1	1	1	1	1,00
9	Saya merasa sangat percaya diri ketika menggunakan AISYA-RA.	5	5	5	4	4,75
10	Saya perlu belajar banyak hal terlebih dahulu sebelum bisa menggunakan AISYA-RA.	1	1	1	1	1,00

No	Pertanyaan SUS	R-01	R-02	R-03	R-04	Rata-Rata
Total Skor Nilai						
	Ganjil	20	19	20	16	
	(Skor - 1)					
Total Skor Nilai						
	Genap	20	20	20	20	
	(5 - Skor)					
Total Nilai						
	Keseluruhan (Ganjil + Genap)	40	39	40	36	
SKOR AKHIR SUS						
	(Total Nilai $\times$ 2,5)	100	97,5	100	90,0	96,88

(Sumber: Hasil kuesioner SUS (System Usability Scale) AISYA-RA, 2026)

Berdasarkan Tabel 5.14, skor rata-rata *System Usability Scale (SUS)* untuk sistem AISYA-RA mencapai 96,88 dari skala maksimum 100. Merujuk pada kriteria interpretasi standar SUS, skor yang melampaui ambang batas 80 diklasifikasikan ke dalam predikat penerimaan tertinggi, yakni "Sangat Baik" (*Excellent*). Perolehan skor yang hampir menyentuh nilai sempurna ini mengindikasikan tingkat penerimaan (*acceptability*) yang luar biasa positif dari para pendidik usia dini dan pimpinan lembaga terhadap fungsionalitas dan keandalan sistem yang dikembangkan.

Terdapat korelasi empiris yang menarik antara pendekatan antarmuka sistem dengan pemecahan hambatan adaptasi teknologi pengguna. Responden R-01 (Kepala RA) dan R-03 (Guru Kelompok B) memberikan skor sempurna, yaitu 100. Nilai maksimal dari Kepala RA ini memvalidasi keberhasilan modul operasional dan *dashboard* manajerial, sedangkan nilai dari guru menegaskan efisiensi modul otomatisasi RPPH harian. Di sisi lain, responden R-04 (Guru Pendamping)

memberikan skor terendah di dalam kelompok uji, yaitu 90. Meskipun berada di urutan terbawah, nilai 90 tersebut masih terkategori "Sangat Baik" (Excellent). Hal ini merupakan pencapaian krusial mengingat pada tahap observasi awal, R-04 adalah pengguna yang paling eksplisit menunjukkan resistensi terhadap komputer dan ketakutan akan kegagalan sistem. Perolehan skor 90 dari R4 membuktikan secara mutlak bahwa penerapan *Conversational User Interface (CUI)* sukses memfasilitasi dan meruntuhkan hambatan literasi digital pada tingkat akar rumput.

Keberhasilan meruntuhkan hambatan tersebut mempertegas perbandingan psikologis dan operasional yang nyata antara metode administrasi manual dengan penggunaan sistem berbasis *Large Language Model (LLM)*. Pada metode manual atau saat menggunakan *Graphical User Interface (GUI)* konvensional yang mengandalkan hierarki menu berlapis, pengguna sering kali mengalami ketakutan akan kesalahan operasional ("takut salah klik") serta merasa terbebani secara kognitif karena harus mengisi formulir digital yang panjang dan kaku. Kehadiran LLM sebagai pengendali utama membalikkan keadaan tersebut. Berbekal kemampuan *Natural Language Processing (NLP)*, LLM mampu memahami perintah informal dalam bahasa sehari-hari, mendeteksi maksud (*intent*) pengguna secara presisi, dan memandu pemenuhan administrasi layaknya rutinitas berbalas pesan. Kemampuan LLM inilah yang secara langsung menghilangkan hambatan psikologis proses manual dan meningkatkan persepsi kemudahan penggunaan (*perceived ease of use*).

Evaluasi pada tingkat butir pertanyaan menunjukkan keunggulan absolut sistem pada parameter kemudahan penggunaan (*usability*). Pada kelompok pertanyaan positif (nomor 1, 3, 5, 7, dan 9), rata-rata jawaban sangat konsisten di atas 4,50, dengan pertanyaan nomor 3 ("Saya pikir sistem AISYA-RA mudah digunakan") meraih nilai sempurna 5,00 dari seluruh responden. Sebaliknya, kelompok pertanyaan negatif (nomor 2, 4, 6, 8, dan 10) secara seragam mendapatkan nilai mutlak 1,00 dari seluruh responden. Jawaban seragam pada pertanyaan nomor 4 dan 10 ini membuktikan secara meyakinkan bahwa pengoperasian sistem AISYA-

RA tidak lagi mensyaratkan dukungan teknis khusus maupun pelatihan ekstensif sebelum penggunaan.

Pencapaian skor SUS yang sangat tinggi (96,88) ini secara komprehensif dipengaruhi oleh empat faktor keputusan desain utama:

1. Implementasi antarmuka *Chatbot* yang mengadopsi struktur bahasa alami mirip aplikasi pesan instan telah sukses mereduksi kurva pembelajaran (*learning curve*). Pengguna merasa familier sejak interaksi pertama.
2. Eliminasi kewajiban menggunakan aplikasi pengolah kata konvensional berhasil membebaskan pengguna dari keluhan format dokumen yang rentan berubah (*layout error*) ketika dipindahkan antarperangkat.
3. Integrasi sistem kecerdasan buatan dengan *Data Master* pada Iterasi 3 membebaskan pengguna dari keharusan mengetik format baku (seperti *header* RPPH dan kop surat lembaga) yang berulang, menjadikan alur kerja administrasi sangat efisien dan otomatis

5.8 *Backlog* Lanjutan dari Mitra

Sesuai prinsip *Agile* yang diterapkan, *product backlog* tetap terbuka terhadap kebutuhan baru setelah sistem utama selesai dievaluasi. Pada sesi *review* akhir, mitra menyampaikan beberapa usulan fitur tambahan yang tidak dimasukkan sebagai ruang lingkup pengujian utama pada penelitian ini, tetapi dicatat sebagai arah pengembangan (*roadmap*) berikutnya. Ringkasan *backlog* lanjutan disajikan pada Tabel 5.17 berikut.

Tabel 5.17 *Backlog* Lanjutan dari Mitra

No	Fitur yang Diminta	Deskripsi Singkat	Status dalam Penelitian
1	Tanda Tangan Digital (<i>E-Signature</i>)	Pembubuhan tanda tangan digital secara otomatis pada dokumen RPPH dan Surat	Dicatat sebagai <i>backlog</i> lanjutan

No	Fitur yang Diminta	Deskripsi Singkat	Status dalam Penelitian
		Keluar sebelum diekspor ke PDF.	
2	Notifikasi Peningat Proaktif	<i>Chatbot</i> mengirimkan pesan pengingat kelengkapan administrasi (misal: pengingat untuk membuat RPPH) secara proaktif kepada guru.	Dicatat sebagai <i>backlog</i> lanjutan
3	Pengaturan Profil Visual (Foto Profil)	Pengguna dapat mengunggah foto pribadi ke dalam sistem agar antarmuka profil terasa lebih personal.	Dicatat sebagai <i>backlog</i> lanjutan

(Sumber: Catatan umpan balik mitra, 2026)

Ketiga usulan pada Tabel 5.17 tidak dimasukkan ke dalam perhitungan efisiensi waktu maupun skor SUS karena muncul setelah sistem utama dinyatakan memenuhi *Definitions of Done*. Pencatatan *backlog* lanjutan tetap penting karena menunjukkan bahwa sistem telah digunakan secara aktif dan mitra mulai mengidentifikasi peluang pengembangan berikutnya berdasarkan pengalaman penggunaan nyata.

Dengan demikian, *backlog* lanjutan pada bagian ini diposisikan sebagai catatan pengembangan ke depan, bukan sebagai bagian dari cakupan hasil utama penelitian.

5.9 Pembahasan

Tahap implementasi dan evaluasi sistem AISYA-RA telah berhasil menjawab seluruh rumusan masalah penelitian. Pengembangan sistem informasi manajemen administrasi guru berbasis *Chatbot* dengan integrasi *Large Language Model (LLM)* terbukti bukan sekadar inovasi teknologi secara konseptual, melainkan sebuah solusi praktis yang memberikan dampak terukur bagi penyelesaian beban kerja

administratif di Raudhatul Athfal (RA) Al-Islam Gunungcupu. Pembahasan hasil penelitian ini diuraikan melalui empat dimensi utama, pendekatan antarmuka pengguna, efisiensi operasional, efektivitas metode pengembangan *Agile*, dan keandalan pemrosesan konteks lokal melalui kecerdasan buatan.

5.9.1 Implementasi Antarmuka Percakapan (CUI) dan Pemecahan Hambatan Literasi Digital

Salah satu tantangan mendasar yang teridentifikasi pada studi awal adalah tingginya resistensi tenaga pendidik terhadap penggunaan aplikasi komputer berbasis menu (*GUI*) akibat keterbatasan literasi digital dan ketakutan akan kesalahan sistem (*error*). Mengacu pada kebiasaan pengguna yang sangat familier dengan aplikasi perpesanan instan seperti WhatsApp, AISYA-RA dikembangkan dengan paradigma *Conversational User Interface (CUI)* sebagai pengendali sistem utama.

Keberhasilan pendekatan ini dibuktikan secara kuantitatif melalui perolehan skor akhir *System Usability Scale (SUS)* sebesar 96,88, yang menempatkan AISYA-RA pada kategori penerimaan "Sangat Baik" (*Excellent*). Fakta empiris yang paling menonjol adalah pemberian skor 90 oleh Responden 4 (Guru Pendamping). Pada tahap observasi awal, Responden 4 merupakan pihak yang paling eksplisit menunjukkan keengganan menggunakan laptop untuk tugas administrasi. Namun, karena instruksi pada AISYA-RA cukup diketikkan menggunakan bahasa alami (seperti "AISYA, catat presensi masuk saya hari ini"), hambatan psikologis dan teknis tersebut berhasil diruntuhkan.

Temuan ini secara langsung mengonfirmasi teori *Technological Pedagogical Content Knowledge (TPACK)* dari Merjovaara et al. (2024), yang menyatakan bahwa persepsi kemudahan penggunaan (*perceived ease of use*) adalah prediktor paling krusial dalam adopsi teknologi oleh pendidik usia dini. Berbeda dengan penelitian Nurjannah et al. (2024) yang mendigitalisasi 24 poin administrasi guru namun masih terjebak pada hierarki GUI yang kompleks, AISYA-RA berhasil membuktikan bahwa penyederhanaan interaksi melalui *Chatbot* merupakan strategi desain yang jauh lebih efektif untuk lingkungan RA/PAUD.

5.9.2 Otomatisasi Administrasi dan Peningkatan Efisiensi Waktu

Pengukuran kinerja sistem pada tahap operasional menunjukkan bahwa kehadiran AISYA-RA berdampak masif terhadap pemangkasan durasi kerja guru. Berdasarkan hasil perbandingan waktu (Tabel 5.15), terjadi rata-rata penghematan efisiensi waktu administrasi sebesar 93,2%. Penghematan tertinggi terjadi pada penyusunan Rencana Pelaksanaan Pembelajaran Harian (RPPH) dan pembuatan rekapitulasi presensi bulanan, yang masing-masing mengalami penurunan waktu dari 45 menit menjadi hanya 1 menit (efisiensi 97,8%).

Kemampuan LLM Gemini dalam mengurai perintah bahasa alami menjadi data terstruktur (*intent parsing*) membuat pencatatan presensi siswa harian dapat diselesaikan dalam hitungan detik. Sementara itu, generasi dokumen RPPH dan pembuatan surat keluar tidak hanya dilakukan secara cepat, tetapi juga presisi. Berbeda dengan pendekatan manual yang sangat rentan terhadap inkonsistensi (*human error*) dan duplikasi nomor surat, sistem secara otomatis mengeksekusi penomoran atomik dari pangkalan data dan menerapkan standar format baku tujuh komponen dokumen RA Al-Islam.

Pencapaian otomatisasi dokumen ini sejalan dengan temuan Zheng et al. (2025) mengenai keandalan LLM dalam merangkai draf pembelajaran (RPP) yang koheren. Namun, penelitian ini melangkah lebih jauh dari studi Zheng dengan mengintegrasikan kemampuan generatif tersebut langsung ke dalam alur kerja administrasi (sistem presensi, persuratan, dan rekapitulasi) di dalam satu platform utuh, mengubah status LLM dari sekadar "alat pembuat teks" menjadi "mesin operasional administratif".

Secara keseluruhan, perbandingan *head-to-head* antara pengerjaan manual dan AISYA-RA membuktikan adanya pergeseran paradigma operasional. Jika pada proses manual pendidik adalah pekerja teknis yang mengolah data menjadi dokumen dari nol, maka dengan kehadiran LLM, peran pendidik berevolusi menjadi kurator atau peninjau (*reviewer*). Pemindahan beban komputasi penulisan dari kognisi manusia ke arsitektur jaringan LLM inilah yang pada akhirnya merevolusi efisiensi waktu administrasi guru di RA Al-Islam Gunungcupu.

5.9.3 Penerapan Metode *Agile* dalam Mengakomodasi Dinamika Kebutuhan

Penelitian ini membuktikan bahwa metode *Agile* adalah kerangka kerja yang paling relevan untuk diimplementasikan pada lingkungan yang belum sepenuhnya memiliki sistem digital. Pengembangan sistem melalui siklus bertahap (Iterasi 1 hingga Iterasi 3) memastikan bahwa fitur yang dibangun benar-benar diuji dan divalidasi oleh pengguna secara berkala.

Momen krusial yang menunjukkan keberhasilan metode *Agile* terjadi pada sesi *Review* Iterasi 2. Ketika prototipe modul inti diujicobakan, pengguna menyadari bahwa kecepatan LLM dalam merancang RPPH belum cukup jika pengguna masih harus mengetik ulang *header* identitas sekolah dan nama guru di dalam Canvas Modal. Umpan balik ini secara langsung memvalidasi pentingnya eksekusi *Product Backlog* Iterasi 3 (Manajemen *Data Master*). Selain itu, pengujian lapangan di Iterasi 2 juga memancing munculnya kebutuhan baru yang sebelumnya tidak terencana dalam skenario otomatisasi, yaitu pencatatan presensi Guru dan Tenaga Kependidikan (GTK). Karena menggunakan pendekatan *Agile*, peneliti memiliki fleksibilitas penuh untuk mengakomodasi penambahan modul Presensi GTK ke dalam Iterasi 3 tanpa merusak struktur sistem.

Adaptabilitas ini mendukung kesimpulan Hidayah Nova et al. (2022) yang menyatakan bahwa pelibatan pengguna pada setiap iterasi *Agile* menghasilkan produk yang lebih selaras dengan kebutuhan riil pengguna di lapangan, jauh lebih unggul dibandingkan dengan pendekatan sekuensial yang kaku (*Waterfall*) seperti yang diimplementasikan pada penelitian sistem PAUD oleh Fiarni et al. (2024).

5.9.4 Pemanfaatan *Retrieval-Augmented Generation (RAG)* untuk Konteks Lokal

Sistem kecerdasan buatan (*generative AI*) umumnya memiliki kelemahan bawaan berupa risiko halusinasi informasi atau pemberian jawaban yang terlalu umum (*generic*). Hal ini sempat dikhawatirkan dalam telaah literatur *Chatbot* pendidikan oleh Labadze et al. (2023). Pada penelitian ini, risiko tersebut berhasil dimitigasi dan diselesaikan pada Iterasi 3 melalui penerapan arsitektur *Retrieval-Augmented Generation (RAG)*.

Dengan memadukan fungsi ekstraksi teks dari PDF pedoman kurikulum internal sekolah, pembentukan vektor 768-dimensi via Gemini API, serta komputasi kemiripan semantik menggunakan pgvector di dalam basis data PostgreSQL Supabase, *Chatbot AISYA-RA* tidak lagi menjawab berdasarkan pengetahuan umum dari internet. Saat guru menanyakan kebijakan kurikulum atau fokus pembelajaran, sistem melakukan kueri ke dalam potongan (*chunks*) dokumen pedoman kurikulum RA Al-Islam Gunungcupu yang telah diunggah sebelumnya. Konstruksi *Super-Contextual System Instruction* yang menggabungkan *Data Master* (identitas sekolah) dan referensi dokumen lokal memastikan *Chatbot* mampu bertindak layaknya asisten manajerial yang benar-benar memahami ruang lingkup dan aturan main di lembaga tersebut.

5.9.5 Analisis Biaya-Manfaat (*Cost-Benefit Analysis*) dan Keberlanjutan Sistem
 Dalam implementasi sistem informasi yang mengintegrasikan layanan kecerdasan buatan, pertimbangan mengenai biaya operasional infrastruktur menjadi aspek krusial bagi keberlanjutan aplikasi di masa depan. Berdasarkan arsitektur pada penelitian ini, sistem secara spesifik di-deploy pada layanan VPS DigitalOcean Droplet dengan spesifikasi menengah (2 vCPUs, 2 GB RAM). Spesifikasi ini memang ditujukan untuk menopang proses kompilasi container Docker dan pengujian intensif selama masa pengembangan aplikasi. Meskipun infrastruktur pengembangan tersebut memiliki beban biaya bulanan yang relatif tinggi, implementasi AISYA-RA untuk operasional jangka panjang di RA Al-Islam Gunungcupu dapat ditekan dan dinilai sangat layak secara finansial (*cost-effective*) melalui beberapa pendekatan strategis berikut:

1. **Transisi ke VPS Ekonomis atau Pemanfaatan Laptop Eksisting.** Skala pengguna aktif harian di lembaga ini sangat terukur, yakni hanya empat orang (satu Kepala RA dan tiga tenaga pendidik), sehingga beban lalu lintas data (*traffic*) yang dihasilkan sangatlah ringan. Untuk operasional jangka panjang pasca-penelitian, pihak sekolah dapat memigrasikan sistem dari penyedia *cloud* asing ke penyedia VPS lokal (Indonesia) dengan skema paket *entry-level* berbayar tahunan yang biayanya jauh lebih terjangkau.

Sebagai alternatif yang bebas biaya langganan *server* (*zero hosting cost*), aplikasi AISYA-RA dapat dijalankan secara lokal dengan mengoptimalkan fasilitas yang sudah ada. Berdasarkan temuan observasi, lembaga memiliki satu unit laptop di ruang Kepala RA dan fasilitas jaringan Wi-Fi sekolah yang cukup stabil. Karena aplikasi telah dikemas secara portabel menggunakan teknologi Docker, sistem dapat dijalankan pada laptop eksisting tersebut sebagai *host* lokal selama jam operasional sekolah. Para guru tetap dapat mengakses sistem secara nirkabel melalui *browser smartphone* masing-masing yang terhubung ke jaringan Wi-Fi sekolah. Dipadukan dengan model bahasa *gemini-latest-flash* yang memiliki efisiensi biaya token sangat tinggi, operasional sistem ini dapat berjalan optimal tanpa membebani kas sekolah untuk pembelian perangkat keras (*hardware*) baru.

2. ***Return on Investment (ROI) dan Pengurangan Biaya Outsourcing Administrasi.*** Investasi infrastruktur sekecil apa pun itu harus dikomparasikan dengan efisiensi tenaga kerja. Sebelum adanya sistem, guru harus menghabiskan waktu 40–55 menit per hari untuk draf RPPH, 30–60 menit setiap bulan untuk rekapitulasi presensi manual, dan 25–40 menit untuk menyusun surat resmi. Tingginya beban repetitif ini kerap memaksa institusi pendidikan untuk melakukan *outsourcing* (menggunakan jasa pihak ketiga, merekrut tenaga tata usaha tambahan, atau tenaga *freelance*) khusus untuk membereskan tumpukan administrasi.

Hadirnya AISYA-RA bertindak sebagai "Asisten Virtual" yang mengotomatisasi seluruh pengerjaan dokumen tersebut melalui percakapan alami. Hal ini secara langsung mengeliminasi kebutuhan (dan pengeluaran) untuk *outsourcing* pengerjaan administrasi, sekaligus mengonversi biaya peluang (*opportunity cost*) tersebut menjadi waktu produktif bagi guru untuk fokus pada interaksi pedagogis di kelas.

3. **Penghematan Belanja Barang Fisik (Alat Tulis Kantor)** Modul *AISYA Canvas Modal* memungkinkan guru melakukan pratinjau (*preview*) dan menyunting dokumen secara interaktif sebelum diubah menjadi format PDF siap cetak. Fitur ini secara langsung menekan pengeluaran belanja Alat Tulis Kantor (ATK) madrasah terutama konsumsi kertas dan tinta *printer* yang selama ini sering terbuang akibat kesalahan format cetak (*trial and error*) saat menggunakan aplikasi pengolah kata konvensional.
4. **Integrasi dengan Pendanaan BOP RA.** Apabila sekolah memutuskan untuk menggunakan layanan VPS berbiaya tahunan rendah dan token API berbayar, biaya tersebut dapat dilegitimasi dalam Rencana Kegiatan dan Anggaran Raudhatul Athfal (RKARA). Lembaga dapat mengalokasikan anggaran ini melalui dana Bantuan Operasional Penyelenggaraan (BOP) RA pada komponen pembiayaan langganan daya atau jasa digital atau komponen administrasi kegiatan sekolah, sehingga ketersediaan anggaran tetap terjamin.

Secara konklusi, perpaduan antara antarmuka berbasis pesan instan, otomatisasi LLM, arsitektur basis data relasional-vektor, dan kerangka pengembangan yang adaptif (*Agile*), telah menjadikan AISYA-RA sebagai model sistem informasi manajemen modern yang sangat inklusif bagi pendidik dengan tingkat literasi digital yang beragam.