

BAB V

HASIL DAN PEMBAHASAN

5.1. Hasil Penelitian

Pada bagian ini, disajikan data hasil penelitian yang diperoleh secara kronologis berdasarkan tahapan alur kerja penelitian yang dikelola menggunakan metodologi *Scrum*. Seluruh luaran disajikan dalam bentuk daftar (tabel), grafik, serta tangkapan layar (*screenshot*) antarmuka sistem SINEO.

5.1.1. Hasil studi awal

Sebelum melaksanakan penarikan dataset berskala besar di dalam ekosistem sistem informasi, alur kerja penelitian diawali dengan mengeksekusi tahapan studi awal. Langkah komprehensif ini bertujuan untuk memetakan akar permasalahan astronomi, meninjau kelayakan infrastruktur teknologi, serta mengisolasi variabel data orbital yang relevan. Berdasarkan alur penelitian, tahapan studi awal ini dibagi secara ketat ke dalam tiga sub-aktivitas ilmiah guna menjamin ketertelusuran dokumen dan akuntabilitas fungsional sejak fase pra-pemrosesan.

5.1.1.1. Hasil Tahapan Studi Literatur

Pada tahapan studi literatur, penulis melaksanakan pengumpulan, peninjauan, dan sintesis kritis terhadap tumpukan dokumen biner artikel jurnal ilmiah internasional Bereputasi, *website* pustaka resmi NASA, serta buku elektronik teks astronomi modern. Fokus utama dari peninjauan pustaka ini adalah untuk mengisolasi parameter koordinat *elips* orbit *Near-Earth Objects* (NEO) yang rumit dan fluktuatif agar dapat disajikan dalam visualisasi yang interaktif bagi masyarakat umum.

Melalui kajian literatur yang mendalam terhadap teori mekanika benda langit, penulis berhasil merangkum beberapa indikator krusial seperti eksentrisitas (e), inklinasi (i), kecepatan relatif (V), dan estimasi jarak meleset (*miss distance*). Selain itu, studi literatur ini digunakan untuk memperkuat landasan teoretis penggunaan arsitektur algoritma *Deep Learning*, khususnya model sekuensial *Dual-Stacked*

Recurrent Neural Network (RNN). Hasil kajian menegaskan bahwa memori keadaan tersembunyi (*hidden state*) pada unit sel rekuren memiliki kapabilitas matematis yang sangat tinggi untuk menangkap pola ketergantungan temporal jangka panjang pada data runtun waktu (*time-series*) tanpa kehilangan konteks sekuensialnya.

5.1.1.2. Hasil Tahapan Studi Dokumentasi

Selanjutnya, tahapan studi dokumentasi dieksekusi untuk membedah panduan teknis operasional dan spesifikasi pustaka eksternal yang akan diintegrasikan ke dalam sistem SINEO. Penulis melakukan bedah arsitektur terhadap dokumentasi *API Jet Propulsion Laboratory* (JPL) NASA guna memahami struktur pengiriman data, batas kueri publik (*rate limiting*), serta struktur *payload* mentah berformat JSON (*JavaScript Object Notation*) yang dilemparkan oleh server NASA.

Di sisi lain, studi dokumentasi juga diarahkan pada peninjauan instruksi pemrograman *web framework* Laravel 12, khususnya terkait optimalisasi manajemen rute, pemanfaatan *Guzzle HTTP Client*, serta penanganan *memory exhaustion* pada pustaka *DomPDF* sewaktu merender data dalam volume masif. Dokumentasi parameter latih model RNN yang mencakup nilai *Final Loss* berbasis *Mean Squared Error* (MSE) dan *Root Mean Squared Error* (RMSE) juga diarsipkan secara ketat. Hal ini dilakukan agar formula persentase akurasi kedekatan prediksi di ruang normalisasi (*normalized space*) dapat dikalibrasi secara objektif menggunakan faktor skala berbasis rujukan evaluasi analisis regresi.

5.1.1.3. Hasil Tahapan Studi Eksplorasi

Sebagai perwujudan praktis dari kedua tahapan sebelumnya, tahapan studi eksplorasi dijalankan dalam bentuk eksperimen hibrida (*hybrid connection*) skala kecil pada lingkungan server lokal (*localhost*) Laragon. Penulis menguji fungsionalitas pengiriman data ramalan kontinu hasil keluaran *Recursive Multi-step Forecasting* dari mesin inferensi *Python* menuju penyimpanan *cache* lokal Laravel 12. Eksplorasi taktis ini dilakukan untuk memastikan bahwa kedua lingkungan

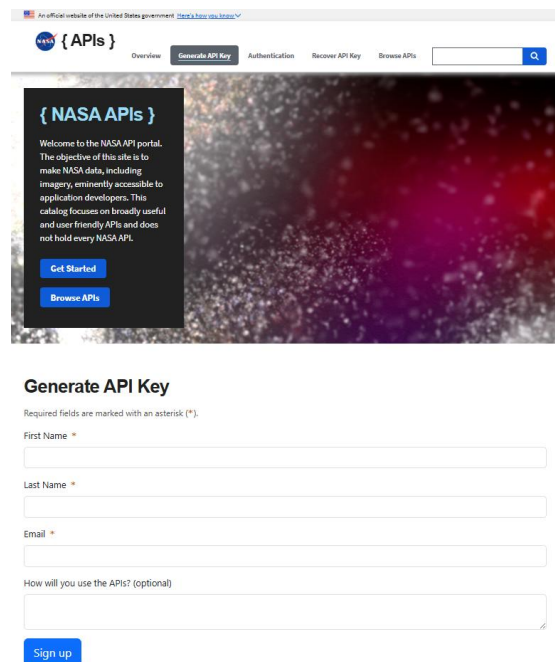
pemrograman yang berbeda dapat saling berkomunikasi secara harmonis tanpa memicu hambatan latensi tinggi.

Selain itu, dilakukan eksplorasi visual pada antarmuka halaman utama dengan merancang kurva volatilitas interaktif, di mana titik-titik koordinat secara dinamis dapat berubah warna untuk merepresentasikan visualisasi klasifikasi tingkat risiko pelintasan objek asteroid secara *riil*. Hasil temuan empiris dari tahapan studi eksplorasi ini membuktikan kelayakan teknis rancangan aplikasi (*proof of concept*), sekaligus bertindak sebagai landasan operasional (*blue-print*) bagi penulis untuk melangkah menuju fase penarikan data murni (*data acquisition*).

5.1.2. Hasil *data acquisition*

Tahapan *Data Acquisition* (akuisisi data) merupakan proses sistematis untuk mengidentifikasi, mengisolasi, dan menarik data sekunder dari sumber eksternal terpercaya (NASA CNEOS) guna mentransformasikannya ke dalam struktur pangkalan data lokal yang terstandarisasi. Dalam arsitektur SINEO, tahapan ini bertindak sebagai jembatan pengambilan data yang menjamin ketersediaan parameter elemen orbital asteroid secara berkelanjutan. Proses akuisisi data ini dieksekusi melalui sinkronisasi dan penarikan data (*data fetching and synchronization*) secara terprogram berbasis antarmuka API (*Application Programming Interface*).

Sebagai hasil dari tahap studi awal, proses akuisisi data mentah (*raw data*) karakteristik fisik dan lintasan asteroid diawali dengan mengonfigurasi hak akses koneksi menuju server *Center for Near-Earth Object Studies* (CNEOS) milik NASA. Penulis melakukan registrasi portal NASA APIs untuk mendapatkan token otentikasi unik berupa API Key. Token ini disematkan ke dalam berkas konfigurasi *.env* pada sistem Laravel 12 sebagai parameter *bearer token* yang valid guna menghindari batasan limitasi *request* (*rate limiting*).



An official website of the United States government [View a how-to guide](#)

{ APIs } Overview **Generate API Key** Authentication Recover API Key Browse APIs

{ NASA APIs }

Welcome to the NASA API portal. The objective of this site is to make NASA data, including imagery, eminently accessible to application developers. This catalog focuses on broadly useful and user friendly APIs and does not hold every NASA API.

[Get Started](#)

[Browse APIs](#)

Generate API Key

Required fields are marked with an asterisk (*).

First Name *

Last Name *

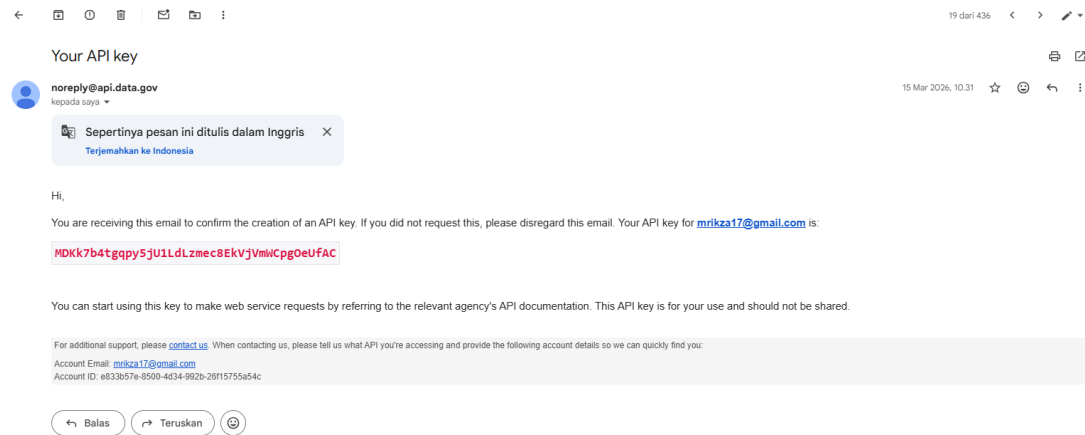
Email *

How will you use the APIs? (optional)

[Sign up](#)

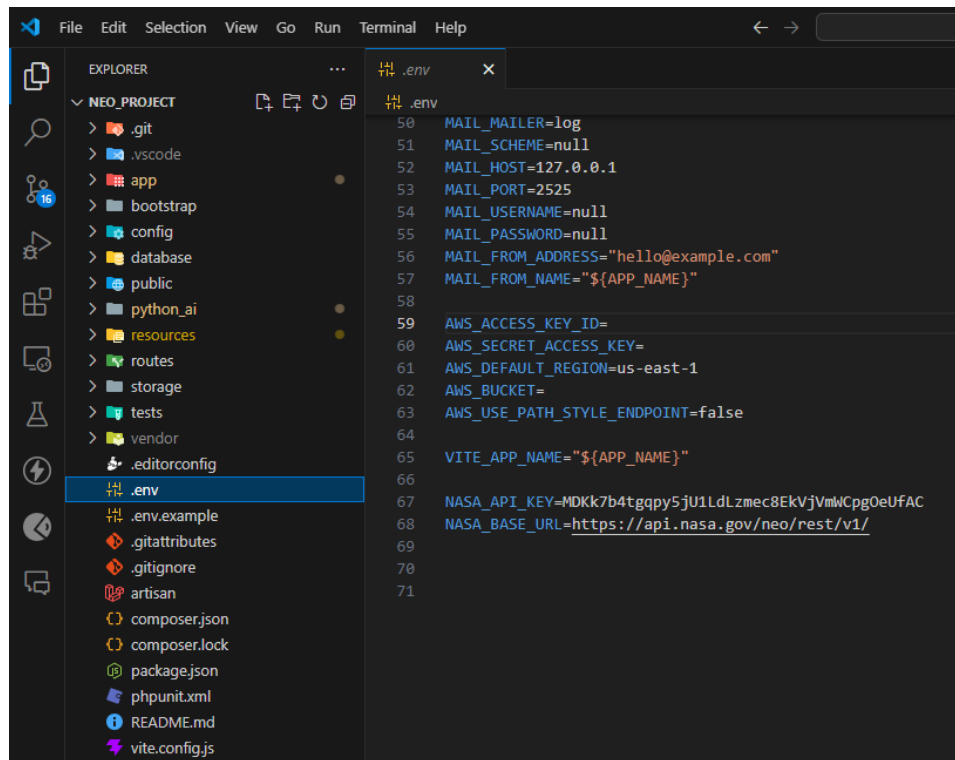
Gambar 5. 1 Proses *Generate API Key* pada NASA APIs *portal*
(Sumber: api.nasa.gov)

Tahap awal dalam implementasi sistem SINEO dimulai dengan melakukan otentikasi pada *platform* resmi penyedia data, yaitu NASA APIs *Portal*. Pada Gambar 5. 1, ditunjukkan proses pengisian formulir pendaftaran digital guna menggenerasikan API Key murni. Pengembang melakukan *input* data kredensial yang meliputi nama depan, nama belakang, serta alamat *email* aktif instansi/pribadi. Pembuatan API Key ini merupakan langkah krusial dalam arsitektur integrasi sistem, karena kunci otentikasi tersebut berfungsi sebagai token akses legal yang meloloskan HTTP *Request* yang dikirimkan oleh *framework* Laravel 12 menuju server *Center for Near-Earth Object Studies* (CNEOS) NASA tanpa terkena batasan kueri publik (*rate limiting*).



Gambar 5. 2 Penerimaan API Key Resmi dari NASA pada *Gmail*
(Sumber: Data Pribadi)

Setelah formulir pendaftaran pada portal NASA berhasil dikirimkan, server otentikasi NASA secara otomatis mengirimkan token akses kredensial ke alamat *Gmail* pengembang yang terdaftar. Seperti yang disajikan pada Gambar 5. 2, sebuah surel resmi dari NASA API *team* berhasil diterima pada kotak masuk *Gmail* pribadi. Surat elektronik tersebut memuat rangkaian *string* alfanumerik unik berupa API Key khusus beserta panduan penggunaan tautan kueri (*request URL*). Token inilah yang kemudian disalin dan dikonfigurasi ke dalam berkas lingkungan kerja sistem (*.env*) pada proyek Laravel 12, sehingga fungsi sinkronisasi otomatis dapat berjalan di latar belakang untuk menarik data parameter asteroid secara berkala.



Gambar 5. 3 Input API Key pada skrip .env
(Sumber: Data Pribadi)

Hasil ekstraksi parameter elemen orbital dasar yang berhasil dihimpun untuk menjadi fitur masukan (*input features*) pemodelan sekuensial dapat dilihat pada Gambar 5. 4:

neo_reference_id	name	eccentricity	inclination	perihelion_distance	semi_major_axis	close_approach_date	miss_distance_km	relative_velocity_kph	is_potentially_hazardous	created_at	updated_at
1176 3754379	(2016 NR)	0.3355878967	6.5204403790	0.5869847992	0.830284969	2026-04-21 03:45:00	27748.45	45702470.99	0	2026-05-11 05:37:20	2026-05-11 05:37:20
1177 2088710	88718 (2001 SL9)	0.2700755178	21.9015782476	0.7747408273	1.0613866924	2026-04-26 23:20:00	29643746.34	53240.77	0	2026-05-11 05:37:21	2026-05-11 05:37:21
1178 3549564	(2019 TM167)	0.4961622255	5.3097386800	0.584456945	1.1600057084	2026-04-26 03:53:00	60817586.12	62224.86	0	2026-05-11 05:37:23	2026-05-11 05:37:23
1179 3732142	(2015 TD023)	0.631023247	48.9132877834	0.5963357956	1.6161883276	2026-04-26 18:11:00	63581585.15	145512.16	1	2026-05-11 05:37:25	2026-05-11 05:37:25
1180 3735656	(2015 XL1)	0.6997025260	10.6164916943	0.5600833203	1.8650950103	2026-04-26 01:34:00	56845320.61	62941.39	0	2026-05-11 05:37:26	2026-05-11 05:37:26
1181 3788454	(2017 TK6)	0.3801379084	16.4349942860	0.5023952507	0.8236374380	2026-04-26 08:25:00	69757866.37	71206.40	0	2026-05-11 05:37:29	2026-05-11 05:37:29
1182 3789618	(2017 WZ12)	0.1967541556	5.8578002589	0.6950087215	0.8652503171	2026-04-26 09:54:00	43505059.66	32032.43	0	2026-05-11 05:37:30	2026-05-11 05:37:30
1183 3802391	(2018 GD)	0.3183096700	6.3067247953	0.9378205641	1.3757369333	2026-04-26 08:17:00	66100064.84	69563.30	0	2026-05-11 05:37:31	2026-05-11 05:37:31
1184 3803910	(2018 HP)	0.4050735263	7.6405461826	0.4795649909	0.8059603402	2026-04-26 05:31:00	29383505.01	60733.57	0	2026-05-11 05:37:33	2026-05-11 05:37:33
1185 3843838	(2019 RE3)	0.3756271458	23.3274513723	0.6391391630	1.0236490251	2026-04-26 12:34:00	43197162.68	64276.35	0	2026-05-11 05:37:35	2026-05-11 05:37:35
1186 2376879	376879 (2001 WW1)	0.1216007314	21.7401617679	1.0626163059	1.2097190240	2026-04-24 08:32:00	49916526.77	41347.40	0	2026-05-11 05:37:37	2026-05-11 05:37:37
1187 3182189	(2004 KB)	0.289002976	20.677668934	0.8446658574	1.1878974838	2026-04-24 13:25:00	27046663.06	43679.48	1	2026-05-11 05:37:38	2026-05-11 05:37:38
1188 3409988	(2008 JN)	0.8033605693	32.0383611209	0.2118088818	1.0771282291	2026-04-24 23:16:00	65080939.72	151119.55	0	2026-05-11 05:37:39	2026-05-11 05:37:39
1189 3796764	(2018 AB12)	0.2020958870	12.4614758017	0.8633572144	1.0820949699	2026-04-24 04:26:00	57645144.62	22533.73	0	2026-05-11 05:37:40	2026-05-11 05:37:40
1190 3827322	(2018 RF2)	0.3399569073	9.6345673340	0.6759136273	1.0240446046	2026-04-24 02:46:00	54121832.87	75024.95	0	2026-05-11 05:37:42	2026-05-11 05:37:42
1191 3717090	(2015 HQ10)	0.6484004877	51.4938479659	0.3980591290	1.1372573481	2026-04-25 21:08:00	37552338.66	118525.49	0	2026-05-11 05:37:43	2026-05-11 05:37:43
1192 3797689	(2018 AO12)	0.3088609995	8.6213999552	0.7600316555	1.0990798835	2026-04-25 08:05:00	41425236.93	33704.05	0	2026-05-11 05:37:45	2026-05-11 05:37:45
1193 2318411	318411 (2009 AH14)	0.6113405529	12.7817788677	0.8964918139	2.3117715101	2026-05-12 08:55:00	14065559.34	47205.37	0	2026-05-12 03:16:23	2026-05-12 03:16:23
1194 3515662	(2010 GZ33)	0.425545383	5.1768416403	1.1049887036	1.9136822116	2026-05-12 01:09:00	78009536.85	60612.42	0	2026-05-12 03:16:25	2026-05-12 03:16:25
1195 3520682	(2010 JR34)	0.1446875079	8.7194950266	0.8205069957	0.9596280310	2026-05-12 06:48:00	21531520.78	31261.44	0	2026-05-12 03:16:27	2026-05-12 03:16:27
1196 3523261	(2010 KC)	0.6071021740	2.2099991506	0.3071876789	0.7816513075	2026-05-12 03:16:00	62244661.03	105023.73	0	2026-05-12 03:16:31	2026-05-12 03:16:31
1197 3830251	(2018 VB10)	0.1631923169	11.5847543935	0.8091174077	0.9905849620	2026-05-12 15:57:00	19716419.69	26339.83	0	2026-05-12 03:16:32	2026-05-12 03:16:32
1198 2375193	375193 (2007 TD1)	0.2798735366	49.7495667187	0.8252802953	1.2862744441	2026-05-13 19:49:00	13084899.86	96064.55	1	2026-05-13 03:31:26	2026-05-13 03:31:26
1199 3092303	(2001 MS3)	0.5337258629	4.4503631746	0.9973650015	2.1390999131	2026-05-13 02:25:00	7938230.02	27904.67	0	2026-05-13 03:31:27	2026-05-13 03:31:27
1200 3631951	(2013 EL28)	0.3961377782	5.8381329640	1.1383808317	1.8851660999	2026-05-13 22:38:00	58866498.37	14654.06	0	2026-05-13 03:31:28	2026-05-13 03:31:28

Gambar 5. 4 Sampel Hasil Akuisisi Elemen Orbital Asteroid dari API NASA CNEOS
(Sumber: Data Pribadi)

Sebagai wujud keberhasilan dari proses sinkronisasi dan penguraian (*parsing*) data JSON dari API NASA, data karakteristik fisik serta parameter esensial asteroid berhasil dimigrasikan ke dalam basis data relasional lokal. Gambar 5. 4 menampilkan struktur isi dari tabel *neo_objects* di dalam *database* MySQL yang diakses melalui antarmuka *phpMyAdmin*. Struktur tabel ini merepresentasikan parameter-parameter astronomi utama yang digunakan sebagai fitur masukan (*input features*) pada mesin kecerdasan buatan RNN. Proses akuisisi data mentah (*raw data*) melalui API Key NASA CNEOS dilaksanakan pada tanggal 15 Maret 2026 untuk menarik rentang data historis lintasan asteroid objek dekat Bumi ke dalam *database*. Di dalam tabel tersebut, terlihat rekaman data *riil* objek dekat bumi yang disajikan pada Tabel 5. 1 di bawah ini:

Tabel 5. 1 Deskripsi Parameter Kolom pada
Tabel *neo_objects* di *database* MySQL

No.	Nama Kolom (Parameter)	Tipe Data	Keterangan / Penjelasan	Pengambilan Data
1.	<i>Id</i> (PK)	<i>BigInt / Int</i>	<i>Primary Key</i> ; nomor identifikasi unik alfanumerik yang diberikan oleh NASA untuk melacak setiap objek asteroid secara spesifik.	15 Maret 2026
2.	<i>neo_reference_id</i>	<i>Varchar</i>	Nomor identifikasi unik alfanumerik yang diberikan secara resmi oleh NASA untuk melacak dan mengenali setiap objek asteroid secara spesifik.	15 Maret 2026
3.	<i>name</i>	<i>Varchar</i>	Nama resmi atau penamaan sementara (<i>designation name</i>) dari objek dekat Bumi (misalnya: 2026 HO2).	15 Maret 2026

No.	Nama Kolom (Parameter)	Tipe Data	Keterangan / Penjelasan	Pengambilan Data
4.	<i>eccentricity</i>	<i>Decimal</i>	Nilai eksentrisitas (e); rasio yang menunjukkan tingkat kelongongan lintasan orbit asteroid dalam rentang angka 0 hingga 1.	15 Maret 2026
5.	<i>inclination</i>	<i>Decimal</i>	Nilai inklinasi (i); sudut kemiringan bidang orbit objek langit terhadap bidang ekliptika Bumi dalam satuan derajat ($^{\circ}$).	15 Maret 2026
6.	<i>Perihelion _distance</i>	<i>Decimal</i>	Jarak perihelion (q); titik jarak terdekat antara objek asteroid dengan Matahari dalam lintasan orbitnya menggunakan satuan <i>Astronomical Unit</i> (AU)	15 Maret 2026
7.	<i>semi_ major _axis</i>	<i>Decimal</i>	Sumbu semi-mayor (a); setengah dari diameter terpanjang lintasan orbit elips asteroid dalam satuan <i>Astronomical Unit</i> (AU)	15 Maret 2026
8.	<i>relative_ velocity _kph</i>	<i>Decimal</i>	Nilai kecepatan (V); estimasi kecepatan relatif gerak asteroid saat melintasi orbitnya dalam satuan kilometer per jam (km/jam) atau km/s.	15 Maret 2026

No.	Nama Kolom (<i>Parameter</i>)	Tipe Data	Keterangan / Penjelasan	Pengambilan Data
9.	<i>miss_ distance _km</i>	<i>Decimal</i>	Jarak meleset (<i>miss distance</i>); parameter estimasi jarak pendekatan terdekat (<i>close approach</i>) asteroid terhadap inti Bumi dalam satuan kilometer (km). Kolom ini bertindak sebagai variabel target (<i>label</i>) dalam pemodelan AI.	15 Maret 2026
10.	<i>close_ approach _date</i>	<i>Datetime</i>	Tanggal dan waktu kronologis terjadinya pendekatan terdekat asteroid terhadap Bumi berdasarkan catatan radar observasi NASA.	15 Maret 2026
11.	<i>is_ potentially _hazardous</i>	<i>TinyInt</i>	Indikator potensi bahaya; penanda biner untuk mengklasifikasikan apakah objek tersebut memiliki potensi ancaman tabrakan berbahaya terhadap Bumi atau tidak.	15 Maret 2026

No.	Nama Kolom (<i>Parameter</i>)	Tipe Data	Keterangan / Penjelasan	Pengambilan Data
12.	<i>absolute_ magnitude</i>	<i>Decimal</i>	Mengukur tingkat kecerahan intrinsik asteroid (H) pada skala logaritmik tanpa pengaruh jarak, digunakan NASA sebagai basis utama perhitungan estimasi ukuran volume fisik objek.	15 Maret 2026
13.	<i>estimated_ diameter_max</i>	<i>Decimal</i>	Menampilkan batas nilai maksimum dimensi ukuran diameter asteroid dalam satuan meter, digunakan sebagai parameter penentu potensi bahaya dampak katastrofik regional (PHA).	15 Maret 2026

(Sumber: Data Pribadi)

Keberhasilan penyimpanan data ini memastikan ketersediaan *dataset* sekuensial yang siap diproses oleh algoritma *Sliding Window* pada modul *Python*.

Untuk melengkapi pemahaman pembaca mengenai aspek teknis penyimpanan data pada SINEO, berikut adalah penjelasan mengenai karakteristik dari tipe data yang diimplementasikan di Tabel 5. 2:

Tabel 5. 2 Penjelasan Karakteristik Tipe Data yang Digunakan pada *Database MySQL*

No.	Tipe data	Karakteristik Komputasi	Alasan Penggunaan
1.	<i>BigInt</i>	Bilangan bulat berukuran besar dengan kapasitas penyimpanan hingga 8 <i>byte</i> angka.	Digunakan pada kolom id untuk mengantisipasi lonjakan kapasitas baris data asteroid seiring berjalannya sinkronisasi otomatis.
2.	<i>Varchar</i>	Karakter <i>string</i> dengan panjang variabel dinamis yang dapat disesuaikan sesuai isi teks.	Digunakan pada kolom <i>neo_reference_id</i> dan <i>name</i> karena panjang karakter kode identifikasi dan nama asteroid dari NASA bervariasi.
3.	<i>Decimal</i>	Bilangan desimal pecahan (<i>floating-point</i>) dengan tingkat presisi ganda (<i>double precision</i>) hingga 15 digit angka di belakang koma.	Sangat krusial digunakan pada semua elemen orbital dasar (<i>e, i, q, a, V</i>) karena data koordinat astronomi NASA memiliki tingkat ketelitian desimal yang sangat panjang dan spesifik.
4.	<i>Datetime</i>	Format standar penulisan waktu berupa tahun, bulan, dan tanggal (<i>YYYY-MM-DD</i>).	Digunakan pada kolom <i>close_approach_date</i> untuk memetakan waktu kejadian objek secara kronologis temporal.

No.	Tipe data	Karakteristik Komputasi	Alasan Penggunaan
5.	<i>TinyInt</i>	Bilangan bulat berukuran sangat kecil dengan kapasitas rentang nilai yang terbatas (biasanya diisi angka 0 atau 1).	Digunakan pada kolom <i>is_potentially_hazardous</i> sebagai representasi tipe data <i>Boolean</i> (0 untuk Aman, dan 1 untuk Bahaya) agar alokasi memori penyimpanan <i>database</i> lokal tetap ringan.

(Sumber: Data Pribadi)

5.1.3. Hasil *product backlog*

5.1.3.1. Hasil *product backlog*

Berdasarkan inisiasi pengumpulan data sekunder yang telah divalidasi pada Tabel 5. 1, alur penelitian dilanjutkan ke tahap perencanaan siklus kerja melalui penyusunan *Product Backlog*. Tahapan ini merupakan representasi fungsional dari fase *Sprint Planning* dalam metodologi *Scrum*, di mana seluruh kebutuhan sistem dipecah secara terperinci menjadi modul menu dan fitur aplikasi *monitoring* SINEO.

Pembagian *Product Backlog* ini ditujukan untuk mempermudah pemetaan arsitektur perangkat lunak serta memastikan setiap komponen antarmuka (*user interface*) dikembangkan secara terarah. Proses pemetaan ini dilakukan pada tanggal 22-27 Maret 2026. Struktur pemetaan fitur dan menu sistem yang menjadi cetak biru operasional aplikasi dirangkum secara komprehensif pada Tabel 5.3 berikut:

Tabel 5. 3 Manifes *Product Backlog* Komponen Menu dan
Fitur Sistem *Monitoring SINEO*
(Sumber: Data Pribadi)

No.	Nama Menu Utama	Berkas Komponen UI (View)	Fungsionalitas Utama Fitur	Kategori Prioritas (Backlog)
1.	<i>Web Profile</i>	<i>welcome.blade.php</i>	Berfungsi sebagai <i>landing page</i> interaktif bertema luar angkasa yang memuat tombol navigasi <i>Call to Action</i> utama, pengontrol audio latar belakang, <i>Glassmorphism Pop-up authModal</i> reaktif berbasis <i>Alpine.js</i> , dan Fitur pembatasan keamanan dan rute penyelamat kustom (<i>Fallback Route</i>).	<i>Medium</i> (Modul Edukasi & Login Modal)
2.	<i>Dashboard Utama</i>	<i>Dashboard.blade.php</i>	Menyajikan <i>executive summary</i> berupa <i>button sync-nasa (Administrator)</i> , <i>widget card</i> visual proksimitas, <i>miss distance AI prediction</i> , analisis tren jarak lintasan, log pendekatan terbaru, proyeksi masa depan.	<i>High</i> (Sistem Inti)

No.	Nama Menu Utama	Berkas Komponen UI (View)	Fungsionalitas Utama Fitur	Kategori Prioritas (Backlog)
3.	Katalog NEO	<i>katalog.blade.php</i>	Menampilkan data tabular parameter orbital secaraurut kronologis (informasi asteroid, status risiko, jarak lintasan, analogi jarak, & tombol detail objek), dilengkapi fitur <i>filter</i> risiko dengan status waspada & bahaya, tombol <i>request</i> hasil cetak laporan data katalog, dan tombol eksekusi cetak dokumen katalog terformat ukuran media kertas A4 (<i>Administrator</i>).	<i>High</i> (Sistem Inti)
4.	Prediksi AI	<i>prediksi_bulanan.blade.php</i>	Menampilkan data tabular secara terurut (estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, & tingkat presisi model) berdasarkan <i>Recursive Multi-step Forecasting</i> model RNN untuk proyeksi jarak 7 hari ke depan, tombol <i>request</i> hasil cetak dokumen, dan tombol eksekusi cetak dokumen (<i>Administrator</i>).	<i>High</i> (Sistem Inti)

No.	Nama Menu Utama	Berkas Komponen UI (View)	Fungsionalitas Utama Fitur	Kategori Prioritas (Backlog)
5.	Analisis AI	<i>analisis.blade.php</i>	Menyajikan <i>monitoring log</i> pelatihan kecerdasan buatan, kurva pembelajaran RNN untuk penurunan <i>loss</i> (MSE) hingga <i>epoch</i> 100, <i>widget</i> total <i>dataset</i> , klasifikasi risiko (waspada/bahaya), tingkat prediksi model, & kecepatan rata-rata objek <i>dataset</i> .	<i>Medium</i> (Modul Evaluasi)
6.	Glosarium AI	<i>glosarium.blade.php</i>	Menyediakan kamus rujukan istilah ilmiah Astronomi, Kecerdasan Buatan, dan Arsitektur Sistem bagi pengguna lintas disiplin ilmu.	<i>Low</i> (Modul Pendukung)
7.	Kelola <i>Request</i> Cetak Dokumen	<i>req_management.blade.php</i>	Modul administratif bagi <i>Admin</i> untuk melihat, memvalidasi, menolak, atau menyetujui kueri daftar antrean permintaan ekspor <i>file</i> PDF laporan yang dikirimkan oleh aktor <i>Visitor</i> .	<i>Medium</i>
8.	Kelola <i>User</i>	<i>user_management.blade.php</i>	Pengelolaan data akun pengguna sistem secara terpusat (Aksi CRUD murni).	<i>Medium</i>

No.	Nama Menu Utama	Berkas Komponen UI (View)	Fungsionalitas Utama Fitur	Kategori Prioritas (Backlog)
9.	Histori <i>Login</i>	<i>login_history.blade.php</i>	Pemantauan rekaman jejak digital aktivitas masuk (<i>Histori Login</i>) pengguna serta Eksekusi cetak lembar rekaman jejak aktivitas masuk untuk dokumen audit berkala	<i>Medium</i>

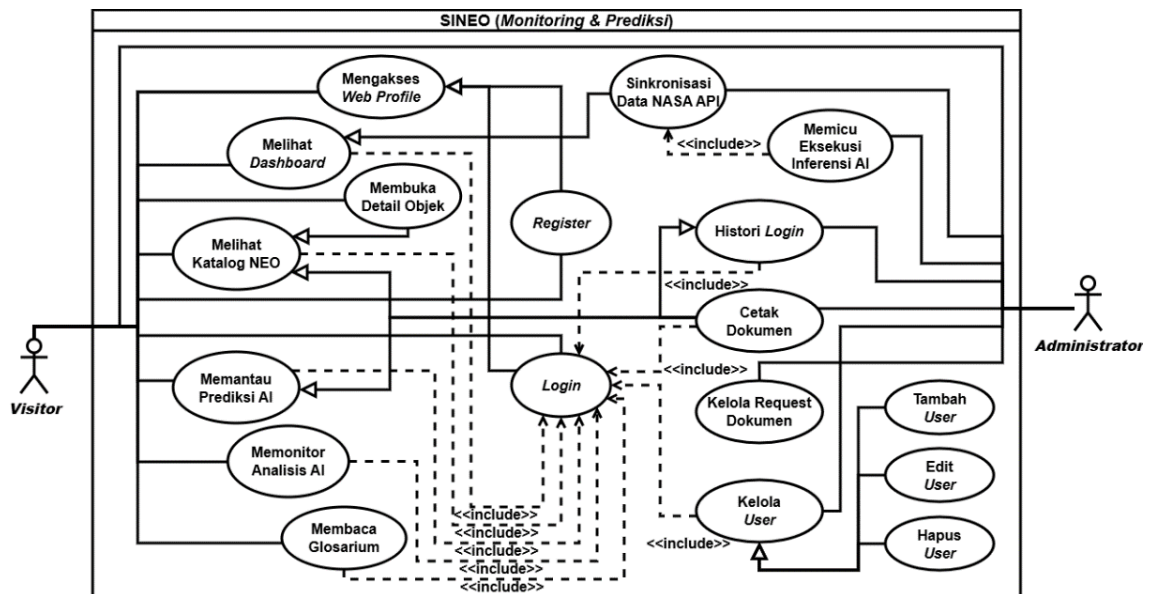
(Sumber: Data Pribadi)

Melalui pemetaan manifes *product backlog* pada Tabel 5. 3, seluruh komponen arsitektur sistem informasi *monitoring* SINEO telah terdefinisi secara utuh. Tahapan perencanaan komparatif ini menjadi landasan dasar bagi penulis untuk melangkah ke fase eksekusi pembuatan sistem, yang diawali dengan pembuktian performa komputasi algoritma kecerdasan buatan pada lingkungan *localhost*.

5.1.3.2. Hasil *use case diagram*

Selain melakukan pemetaan daftar fitur ke dalam manifes tabel, fase *product backlog* ini juga dilengkapi dengan pemodelan fungsionalitas sistem menggunakan *use case diagram*. Pemodelan ini bertujuan untuk mendefinisikan batas ruang lingkup sistem (*system boundary*) serta menggambarkan interaksi perilaku antara aktor luar dengan paket fungsionalitas yang disediakan oleh sistem informasi *monitoring* SINEO.

Aktor yang terlibat di dalam ekosistem sistem *monitoring* SINEO diidentifikasi menjadi dua poros, yaitu *visitor* (masyarakat umum/akademisi) sebagai pengguna akhir yang mengonsumsi visualisasi data, serta *administrator* yang memiliki hak akses otoritas dalam mengendalikan fungsionalitas *back-end* sistem. Representasi visual dari hubungan interaksi aktor terhadap fungsi sistem SINEO dipaparkan secara mendalam melalui *use case diagram* pada Gambar 5. 5 berikut:



Gambar 5. 5 Use Case Diagram Sistem Informasi Near-Earth Objects (SINEO)
(Sumber: Data Pribadi)

Untuk sistem *monitoring* SINEO ini memiliki 2 Aktor utama dengan hak akses yang jelas:

- Pengguna / *visitor* (masyarakat/akademisi): Hak akses membaca dan melihat informasi umum analitik astronomi SINEO.
- Administrator*: Hak akses penuh untuk mengelola siklus data dan memicu mesin AI, serta bertanggung jawab terhadap manajemen pengguna dan validasi dokumen laporan.

Berikut merupakan penjelasan untuk Gambar 5. 5:

Tabel 5. 4 Matriks Hubungan Aktor dan Fungsionalitas Use Case SINEO

No.	Nama Use Case	Deskripsi Singkat Operasional Fungsionalitas	Hak Akses Aktor
1.	Mengakses Web Profile	Menampilkan halaman utama profil pengenalan sistem informasi SINEO beserta modul pemutar musik latar belakang.	Visitor & Admin

No.	Nama Use Case	Deskripsi Singkat Operasional Fungsi	Hak Akses Aktor
2.	<i>Register</i>	Proses mendaftar akun menggunakan kredensial melalui <i>authModal pop-up</i> pada <i>landing page</i> .	<i>Visitor</i>
3.	<i>Login</i>	Proses otentikasi validasi akun menggunakan kredensial <i>email</i> dan <i>password</i> melalui <i>authModal pop-up</i> pada <i>landing page</i> .	<i>Visitor & Admin</i>
4.	Melihat <i>Dashboard</i>	Menampilkan visual proksimitas, <i>miss distance AI prediction</i> , analisis tren jarak lintasan, log pendekatan terbaru, proyeksi masa depan, serta tombol <i>sync data (Admin)</i> .	<i>Visitor & Admin</i>
5.	Melihat Katalog NEO	Menampilkan data tabular parameter orbital secaraurut kronologis (informasi asteroid, status risiko, jarak lintasan, analogi jarak, & tombol detail objek), dilengkapi fitur <i>filter</i> risiko dengan status waspada & bahaya, tombol <i>request</i> cetak dokumen data katalog, dan tombol ekspor <i>file</i> PDF (<i>administrator</i>).	<i>Visitor & Admin</i>

No.	Nama <i>Use Case</i>	Deskripsi Singkat Operasional Fungsi	Hak Akses Aktor
6.	Membuka Detail Objek	Menampilkan karakteristik fisik asteroid secara individual dan menerapkan penilaian risiko spasial otomatis (<i>Traffic Light System</i>), serta menampilkan parameter fisis objek secara lengkap (e , i , q , a , $\emptyset$, H), kecepatan V (km/s & km/h), menampilkan jarak objek ke bumi (km, AU, & LD), mengarahkan detail objek ke JPL NASA, analisis keamanan orbital dan visual 2D dengan keterangannya.	<i>Visitor & Admin</i>
7.	Memantau Prediksi AI	Menampilkan data tabular secara terurut (estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, & tingkat presisi model) berdasarkan <i>Recursive Multi-step Forecasting</i> model RNN untuk proyeksi jarak 7 hari ke depan, tombol <i>request</i> cetak dokumen (mengirimkan kueri permintaan dari antarmuka <i>visitor</i> ke <i>admin</i>), dan tombol ekspor <i>file</i> PDF (<i>administrator</i>).	<i>Visitor & Admin</i>

No.	Nama <i>Use Case</i>	Deskripsi Singkat Operasional Fungsi	Hak Akses Aktor
8.	Memonitor Analisis AI	Menyajikan <i>monitoring log</i> pelatihan kecerdasan buatan, kurva pembelajaran RNN untuk penurunan <i>loss</i> (MSE) hingga <i>epoch</i> 100, <i>widget</i> total <i>dataset</i> , klasifikasi risiko (waspada/bahaya), tingkat prediksi model, & kecepatan rata-rata objek <i>dataset</i> .	<i>Visitor & Admin</i>
9.	Membaca Glosarium	Menyediakan kamus rujukan istilah ilmiah astronomi, kecerdasan buatan, dan arsitektur sistem bagi pengguna lintas disiplin ilmu.	<i>Visitor & Admin</i>
10.	Cetak Dokumen	Mengeksekusi instruksi pencetakan menjadi dokumen <i>file</i> (PDF) siap unduh berukuran kertas A4, untuk data katalog, prediksi, dan histori <i>login</i> .	<i>Administrator</i>
11.	Kelola <i>Request</i> Dokumen	Modul administratif bagi <i>admin</i> untuk melihat, memvalidasi, menolak, atau menyetujui kueri daftarajuan ekspor <i>file</i> PDF laporan yang dikirimkan oleh aktor <i>visitor</i> , dan hasilnya dikirim via <i>email</i> .	<i>Administrator</i>
12.	Kelola <i>User</i>	Modul administratif penuh untuk melakukan manipulasi data (CRUD) terhadap akun pengguna di dalam sistem.	<i>Administrator</i>

No.	Nama <i>Use Case</i>	Deskripsi Singkat Operasional Fungsi	Hak Akses Aktor
13.	Tambah <i>User</i>	Menambahkan data rekaman akun pengguna baru (<i>admin</i> atau <i>visitor</i>) ke dalam <i>database</i> relasional MySQL.	<i>Administrator</i>
14.	Edit <i>User</i>	Melakukan pembaruan (<i>update</i>) data profil, <i>email</i> , sandi, atau hak akses peran akun yang telah tersimpan di dalam sistem.	<i>Administrator</i>
15.	Hapus <i>User</i>	Menghapus secara permanen (<i>delete</i>) rekaman akun pengguna tertentu dari basis data sistem SINEO.	<i>Administrator</i>
16.	Histori <i>Login</i>	Papan log digital rekaman aktivitas masuk akun pengguna sebagai langkah audit dan mitigasi celah keamanan web.	<i>Administrator</i>
17.	Sinkronisasi Data NASA API	Memicu HTTP <i>Request</i> ke server CNEOS NASA untuk menyedot data 6 minggu terakhir dan menyimpannya ke MySQL.	<i>Administrator</i>
18.	Memicu Eksekusi Inferensi AI	Menjalankan skrip <i>predict_neo.py</i> secara otomatis via <i>shell_exec</i> untuk memperbarui <i>file</i> hasil proyeksi JSON.	<i>Administrator</i>

(Sumber: Data Pribadi)

5.1.4. Hasil *sprint planning*

Tahapan penelitian dilanjutkan dengan melaksanakan fase *sprint planning* (perencanaan siklus kerja). Tahap ini bertujuan untuk menentukan komitmen serta memetakan beban kerja arsitektur sistem informasi SINEO yang telah disusun pada manifes *product backlog*. Di dalam ekosistem pengembangan rekayasa perangkat

lunak menggunakan metode *Scrum*, *sprint planning* bertindak sebagai penentu arah estimasi waktu pengerjaan agar proses penulisan kode program (*coding*) dapat berjalan secara terstruktur, fokus, dan tidak mengalami tumpang tindih fungsi.

Berdasarkan prioritas manifes kebutuhan, seluruh item didekonstruksi ke dalam rentang siklus kerja (*sprint*) berdurasi tetap yang dibagi menjadi dua fokus pengembangan utama, yaitu poros kecerdasan buatan (*Back-end Engine*) dan poros aplikasi *monitoring* (*Front-end Interface*). Distribusi perencanaan beban kerja *sprint* tersebut dirangkum secara mendetail pada Tabel 5.5 berikut:

Tabel 5. 5 Distribusi Pembagian Modul Kerja pada Fase
Sprint Planning Sistem SINEO

No.	Target Siklus Kerja (<i>Sprint Goal</i>)	Butir Tugas Teknis (<i>Task Backlog</i>)	Estimasi Waktu	Metrik Target Keberhasilan Evaluasi	Waktu Pelaksanaan
1.	<i>Sprint 1: Back-end Intelligence (Python AI)</i>	Pembuatan modul data lewat skrip <i>preprocess_data.py</i> untuk memicu kueri kalender dinamis 30 hari.	10 jam/hari	Menghasilkan struktur bentuk matriks dimensi sekuensial <i>input shape</i> (sampel, <i>window</i> , fitur) yang valid	08-09 April 2026
		Eksekusi komputasi pelatihan model via skrip <i>train_rnn.py</i> sepanjang 100 <i>epoch</i> penuh.	10 jam/hari	Menekan fungsi nilai kerugian kuadrat (<i>Mean Squared Error</i>) serendah mungkin hingga konvergen di bawah angka 0.001.	10-12 April 2026
		Integrasi mesin inferensi berantai via skrip <i>predict_neo.py</i> untuk <i>Recursive Multi-step Forecasting</i> .	10 jam/hari	Berhasil mengeksport struktur prediksi (<i>miss distance</i>) lintasan 7 hari ke depan menjadi dokumen luaran JSON murni.	13-14 April 2026

No.	Target Siklus Kerja (<i>Sprint Goal</i>)	Butir Tugas Teknis (<i>Task Backlog</i>)	Estimasi Waktu	Metrik Target Keberhasilan Evaluasi	Waktu Pelaksanaan
2.	<i>Sprint 2:</i> <i>Web Profile,</i> <i>Dashboard,</i> Autentikasi, Sinkronisasi Data NASA API & Memicu Eksekusi Inferensi AI.	Konfigurasi struktur inti pengendali web pada arsitektur <i>models</i> , kontroler, <i>public</i> , sistem pengamanan rute, serta (<i>.env</i>).	6 jam/hari	Gateway <i>Fallback Route</i> aktif & data kueri NASA tersimpan aman di MySQL.	15-16 April 2026
		Perancangan dan penyusunan komponen antarmuka reaktif <i>Blade</i> (<i>views folder</i>).	9 jam/hari	Modul <i>Dashboard & Web Profile</i> berhasil me-render grafik <i>Chart.js</i> & modal <i>Alpine.js</i> .	17-21 April 2026

No.	Target Siklus Kerja (<i>Sprint Goal</i>)	Butir Tugas Teknis (<i>Task Backlog</i>)	Estimasi Waktu	Metrik Target Keberhasilan Evaluasi	Waktu Pelaksanaan
3.	<i>Sprint 3:</i> Katalog NEO, Detail Objek, Prediksi AI, Analisis AI & Glosarium AI	Konfigurasi struktur inti pengendali web pada arsitektur models, kontroler, serta sistem pengamanan rute.	6 jam/hari	Mengimplementasikan <i>Traffic Light System</i> pada kontroler untuk menentukan warna tema berdasarkan <i>threshold</i> status asteroid. Menghitung persentase akurasi model dari nilai MSE menggunakan pembobotan RMSE & validasi rute akses data.	22-23 April 2026
		Perancangan dan penyusunan komponen antarmuka reaktif <i>Blade (views folder)</i> .	9 jam/hari	Visualisasi <i>Katalog & Prediksi ter-render</i> dinamis beserta fitur ekspor dokumen. Serta menu Analisis AI & Glosarium AI berhasil me-render grafik <i>Chart.js</i> & modal <i>Alpine.js</i> .	24-28 April 2026

No.	Target Siklus Kerja (<i>Sprint Goal</i>)	Butir Tugas Teknis (<i>Task Backlog</i>)	Estimasi Waktu	Metrik Target Keberhasilan Evaluasi	Waktu Pelaksanaan
4.	<i>Sprint 4:</i> Kelola <i>Req</i> Dokumen, Cetak Dokumen, Kelola <i>Users &</i> Histori <i>Login</i>	Konfigurasi struktur inti pengendali web pada arsitektur models, kontroler, <i>mail</i> , <i>providers public</i> , serta sistem pengamanan rute.	5 jam/hari	Menggunakan <i>flag is_responded</i> pada model <i>PrintRequest</i> untuk mengunci status tombol setelah <i>email</i> terkirim. Mengintegrasikan <i>Google SMTP</i> via <i>Mailable</i> untuk notifikasi <i>Gmail</i> dan menangkap IP <i>User Agent</i> pada fungsi histori <i>login</i> .	01-02 Mei 2026
		Perancangan dan penyusunan komponen antarmuka reaktif <i>Blade</i> (<i>views folder</i>).	6 jam/hari	Menyediakan tombol aksi (Setujui/Tolak) yang berubah menjadi tombol kirim <i>email</i> (<i>approve/reject</i>) dan berubah lagi menjadi <i>disabled</i> (abu-abu) untuk menandakan aksi sudah dilakukan. CRUD <i>User & Log Audit</i> Histori <i>Login</i> berjalan terintegrasi & tervalidasi.	03-07 Mei 2026

(Sumber: Data Pribadi)

Melalui penentuan parameter acuan taktis pada fase *Sprint Planning* yang disajikan pada Tabel 5. 5, seluruh target fungsionalitas kini telah memiliki cetak biru (*blueprint*) dan metrik evaluasi kelayakan yang terukur jelas. Penulis menggunakan hasil pembagian komitmen kerja ini sebagai panduan utama untuk mengawali fase eksekusi pembuatan sistem (*Sprint Execution*). Sesuai dengan pembagian urutan prioritas tersebut, tahapan implementasi fisik diawali dengan melakukan eksperimen komputasi algoritma pembelajaran mendalam (*Deep Learning*) pada poros kecerdasan buatan, yang dilaporkan secara rinci pada tahap berikutnya.

5.1.5. Hasil *sprint execution*

5.1.5.1. Hasil *sprint 1*

Sebagai langkah konkrit dari dekonstruksi tugas pada fase *Sprint Planning*, penelitian memasuki tahapan *Sprint Execution Sprint Pertama (Back-end Intelligence Sprint)*. Sesuai dengan pembagian komitmen kerja, fokus utama pada siklus eksekusi ini adalah membangun, melatih, dan mematangkan arsitektur jaringan saraf tiruan *Dual-Stacked Recurrent Neural Network* (RNN) di lingkungan pengembangan *localhost* menggunakan bahasa pemrograman *Python*. Seluruh rangkaian proses manipulasi data, iterasi pembaruan bobot lewat algoritma *Backpropagation Through Time* (BPTT), hingga pembentukan mesin inferensi otomatis dilaporkan secara terperinci pada tahap ini.

5.1.5.1.1. *Daily scrum*

Aktivitas koordinasi harian *Daily Scrum* sepanjang pelaksanaan *Sprint 1 (Back-end Intelligence Sprint)* dipimpin secara terstruktur oleh *Scrum Master* bersama analis untuk melacak progres harian, pembagian beban tugas, serta memitigasi kendala teknis komputasi sains data pada latar belakang server. Fokus utama peninjauan harian pada iterasi pertama ini diarahkan secara ketat untuk mematangkan dan menyinkronkan alur kerja dari tiga berkas skrip *Python* inti yang bertindak sebagai mesin kecerdasan buatan hibrida pada *platform* SINEO. Pada hari-hari awal siklus, diskusi harian difokuskan pada standarisasi parameter masukan dan manipulasi data runtun waktu (*time-series*) elemen orbital asteroid yang diisolasi lewat skrip

preprocess_data.py. Tim pengembang mengevaluasi stabilitas fungsi kueri bersyarat *DATE_SUB* pada *database* MySQL agar proses pemotongan jendela kalender dinamis sepanjang 30 hari ke belakang dapat berjalan instan tanpa memicu terjadinya *I/O bottleneck* sewaktu mentransformasikan varians nilai fitur murni ke dalam skala normalisasi (*normalized space*) menggunakan *MinMaxScaler*.

Memasuki pertengahan siklus *Sprint*, koordinasi harian beralih untuk memantau jalannya eksperimen pelatihan jaringan saraf rekuren bertumpuk yang dieksekusi oleh skrip *train_rnn.py*. Masalah kritis yang muncul dan dievaluasi dalam *Daily Scrum* pada fase ini adalah penentuan kombinasi *hyperparameters* yang optimal guna menekan laju fungsi kerugian kuadrat *Mean Squared Error* (MSE) agar dapat merosot secara konvergen di bawah ambang batas target 0.001. Pengembang melaporkan pergerakan fluktuasi nilai *loss* harian serta menguji efektivitas penyuntikan *Dropout Rate* sebesar 20% untuk menghentikan gejala bias orientasi (*overfitting*) selama model dilatih sepanjang 100 *epoch* penuh.

Pada hari-hari akhir menjelang penutupan *Sprint* 1, agenda *Daily Scrum* difokuskan pada pematangan modul penalaran berantai melalui skrip *predict_neo.py* yang bertugas menjalankan fungsi *Recursive Multi-step Forecasting*. Hambatan teknis yang diidentifikasi oleh tim adalah potensi terjadinya stagnasi titik tetap (*fixed point*) pada iterasi proyeksi jarak meleset (*miss distance*) sepekan ke depan. Melalui diskusi harian yang adaptif, tim pengembang menyepakati solusi teknis berupa implementasi *Jitter Correction* dengan menyuntikkan variasi gangguan acak (*stochastic noise*) minor pada skrip *Python* tersebut, sehingga kalkulasi regresi kontinu dapat diekspor secara valid ke dalam struktur berkas dokumen biner bersih berformat *prediction_results.json* tanpa tercampur oleh baris *native warning logs* internal *TensorFlow*. Rangkaian evaluasi harian ini memastikan integrasi ketiga skrip *Python* inti berada dalam kondisi matang dan siap dihubungkan dengan pengontrol *Laravel* 12 melalui fungsi perintah *shell* latar belakang.

Tabel 5. 6 Hasil *Daily Scrum Sprint 1*

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
08–09 April 2026	-	-	<i>Coding preprocess_data.py</i> (skema <i>Sliding Window</i>)
09 April 2026	<i>Pre-processing</i> & normalisasi fitur (<i>MinMaxScaler</i>)	Validasi <i>input shape</i> data sekuensial	Finalisasi <i>preprocess_data.py</i> untuk <i>input</i> RNN
10–11 April 2026	Integrasi data ke model RNN	Konvergensi <i>loss</i> yang tidak stabil	<i>Coding</i> & pelatihan <i>train_rnn.py</i> (100 <i>Epoch</i>)
12 April 2026	Pelatihan model & <i>tuning dropout</i> 0.2	<i>Overfitting</i> pada awal iterasi pelatihan	Optimasi bobot & <i>save</i> model <i>.keras</i>
13–14 April 2026	Penyimpanan model & <i>scaler</i>	<i>Path</i> absolut pada server lokal	<i>Coding predict_neo.py</i> (<i>Recursive Forecasting</i>)
14 April 2026	Eksekusi inferensi AI via <i>shell_exec</i>	<i>I/O bottleneck</i> pada akses file JSON	Implementasi <i>Cache</i> <i>Laravel</i> untuk akselerasi data

(Sumber: Data Pribadi)

5.1.5.1.2. Implementation

Pada bagian ini, dilaporkan data hasil pengujian komputasi dan eksperimen kecerdasan buatan (*Deep Learning*) yang bertindak sebagai mesin inferensi prediksi pada sistem SINEO. Proses pembuatan model rekuren ini dilakukan secara terpisah melalui *scripting* berbasis bahasa pemrograman *Python* dan *library TensorFlow/Keras* di dalam lingkungan pengembangan lokal.

Eksperimen dilakukan dengan menyusun arsitektur *Dual-Stacked Recurrent Neural Network* yang terdiri atas dua lapisan tersembunyi (*hidden layer*) bertumpuk. Konfigurasi arsitektur jaringan saraf, parameter pelatihan (*hyperparameters*), serta spesifikasi alokasi *neuron* yang diimplementasikan dalam eksperimen ini dirangkum secara mendetail pada Tabel 5. 7 di bawah ini:

Tabel 5. 7 Spesifikasi Arsitektur dan *Hyperparameters*
Model *Dual-Stacked RNN*

No.	Komponen / Parameter Model	Spesifikasi Komputasi	Fungsi / Peranan dalam Eksperimen
1	<i>Temporal Input Layer</i>	<i>Dynamic Sliding Window (30 Days Range)</i>	Membentuk matriks runtun waktu temporal fleksibel berdasarkan seluruh data yang masuk dalam rentang kalender 1 bulan terakhir.
2	<i>RNN Hidden Layer 1</i>	64 Neuron, Aktivasi <i>Tanh</i>	Mengekstraksi karakteristik sekuensial global dan meneruskan seluruh runtun waktu (<i>return sequences</i>).
3	<i>RNN Hidden Layer 2</i>	32 Neuron, Aktivasi <i>Tanh</i>	Menyaring informasi temporal lanjutan dari lapisan pertama untuk mendalami pola fluktuasi elemen orbital.
4	<i>Dense Hidden Layer</i>	32 Neuron, Aktivasi <i>ReLU</i>	Melakukan pemetaan fitur non-linear sebelum dialirkan menuju lapisan keluaran akhir.
5	<i>Output Layer</i>	1 Neuron, Aktivasi <i>Linear</i>	Menghasilkan nilai regresi kontinu berupa prediksi estimasi jarak terdekat (<i>miss distance_km</i>).
6	<i>Loss Function</i>	<i>Mean Squared Error (MSE)</i>	Mengukur tingkat kesalahan kuadrat rata-rata antara hasil prediksi dengan data aktual radar NASA.

No.	Komponen / Parameter Model	Spesifikasi Komputasi	Fungsi / Peranan dalam Eksperimen
7	<i>Optimizer</i>	<i>Adam Optimizer</i>	Mengoptimasi dan memperbarui bobot serta bias jaringan secara efisien selama proses pelatihan.
8	Total <i>Epochs</i>	100 Iterasi	Jumlah siklus pengerjaan pelatihan model untuk mempelajari keseluruhan <i>dataset</i> latihan.

(Sumber: Data Pribadi)

Secara mekanis, eksperimen pemodelan kecerdasan buatan *Dual-Stacked* RNN pada sistem SINEO dikembangkan secara modular ke dalam tiga berkas skrip *Python* utama yang saling berkesinambungan. Penjelasan fungsional dari masing-masing berkas kode sumber tersebut dipaparkan sebagai berikut:

a. Skrip pra-pemrosesan data kalender dinamis (*preprocess_data.py*)

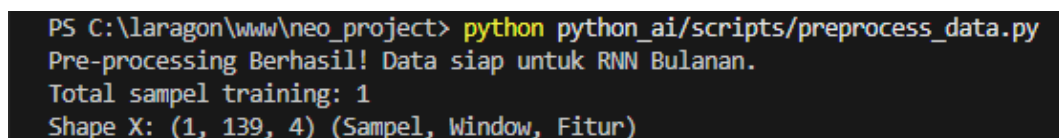
```

python > scripts > preprocess_data.py > get_clean_data
1 import pandas as pd
2 import numpy as np
3 from sqlalchemy import create_engine
4 from sklearn.preprocessing import MinMaxScaler
5
6 def get_clean_data():
7     # 1. Koneksi ke Database Relasional
8     engine = create_engine("mysql+mysqlconnector://root@localhost/neo_db")
9
10    # 2. QUERY DINAMIS: Mengambil data khusus 30 hari terakhir berdasarkan kalender
11    query = """
12        SELECT eccentricity, inclination, perihelion_distance, relative_velocity_kph, miss_distance_km, close_approach_date
13        FROM neo_objects
14        WHERE close_approach_date >= (
15            SELECT DATE_SUB(MAX(close_approach_date), INTERVAL 30 DAY) FROM neo_objects
16        )
17        ORDER BY close_approach_date ASC
18    """
19
20    try:
21        df = pd.read_sql(query, engine)
22
23        # Validasi batas minimal sampel dalam sebulan untuk kestabilan RNN
24        if df.empty or len(df) < 15:
25            print(f"Dataset 30 hari terakhir terlalu sedikit ({len(df)} data).")
26            return None, None, None
27
28        # 3. Normalisasi Fitur (Skala 0-1)
29        scaler_x = MinMaxScaler(feature_range=(0, 1))
30        scaler_y = MinMaxScaler(feature_range=(0, 1))
31
32        # Memisahkan fitur orbital dan label target
33        scaled_x = scaler_x.fit_transform(df[['eccentricity', 'inclination', 'perihelion_distance', 'relative_velocity_kph']])
34        scaled_y = scaler_y.fit_transform(df[['miss_distance_km']])
35
36        # 4. IMPLEMENTASI TEMPORAL SLIDING WINDOW
37        # Menggunakan ukuran window dinamis sesuai jumlah data yang masuk dalam rentang 30 hari
38        X, y = [], []
39        window_size = len(scaled_x) - 1 # Menggunakan seluruh data sebulan minus 1 untuk target
40
41        if window_size < 10:
42            return None, None, None
43
44        X.append(scaled_x[window_size])
45        y.append(scaled_y[window_size])
46
47        return np.array(X), np.array(y), (scaler_x, scaler_y, window_size)
48
49    except Exception as e:
50        print(f"Error pada Pra-pemrosesan Data: {str(e)}")
51        return None, None, None

```

Gambar 5. 6 Skrip file *preprocess_data.py*
(Sumber: Data Pribadi)

Berkas *preprocess_data.py* bertanggung jawab penuh atas tahapan manipulasi data awal (*data preparation*). Pada skrip ini, diterapkan pengambilan data menggunakan metode *Dynamic Sliding Window* berbasis rentang kalender temporal. Skrip mengirimkan kueri bersyarat menggunakan fungsi *DATE_SUB* ke *database MySQL* untuk mengisolasi seluruh populasi objek asteroid yang melintas dalam jangka waktu tepat 30 hari ke belakang dari titik waktu observasi terakhir. Atribut fisis tersebut kemudian dinormalisasi menggunakan *MinMaxScaler* ke dalam skala rentang 0 hingga 1. Seluruh deret data sebulan tersebut diolah menjadi satu kesatuan matriks sekuensial utuh sebagai fitur masukan (*X*) demi mempertahankan memori jangka pendek jaringan saraf tiruan secara komprehensif.



```
PS C:\laragon\www\neo_project> python python_ai/scripts/preprocess_data.py
Pre-processing Berhasil! Data siap untuk RNN Bulanan.
Total sampel training: 1
Shape X: (1, 139, 4) (Sampel, Window, Fitur)
```

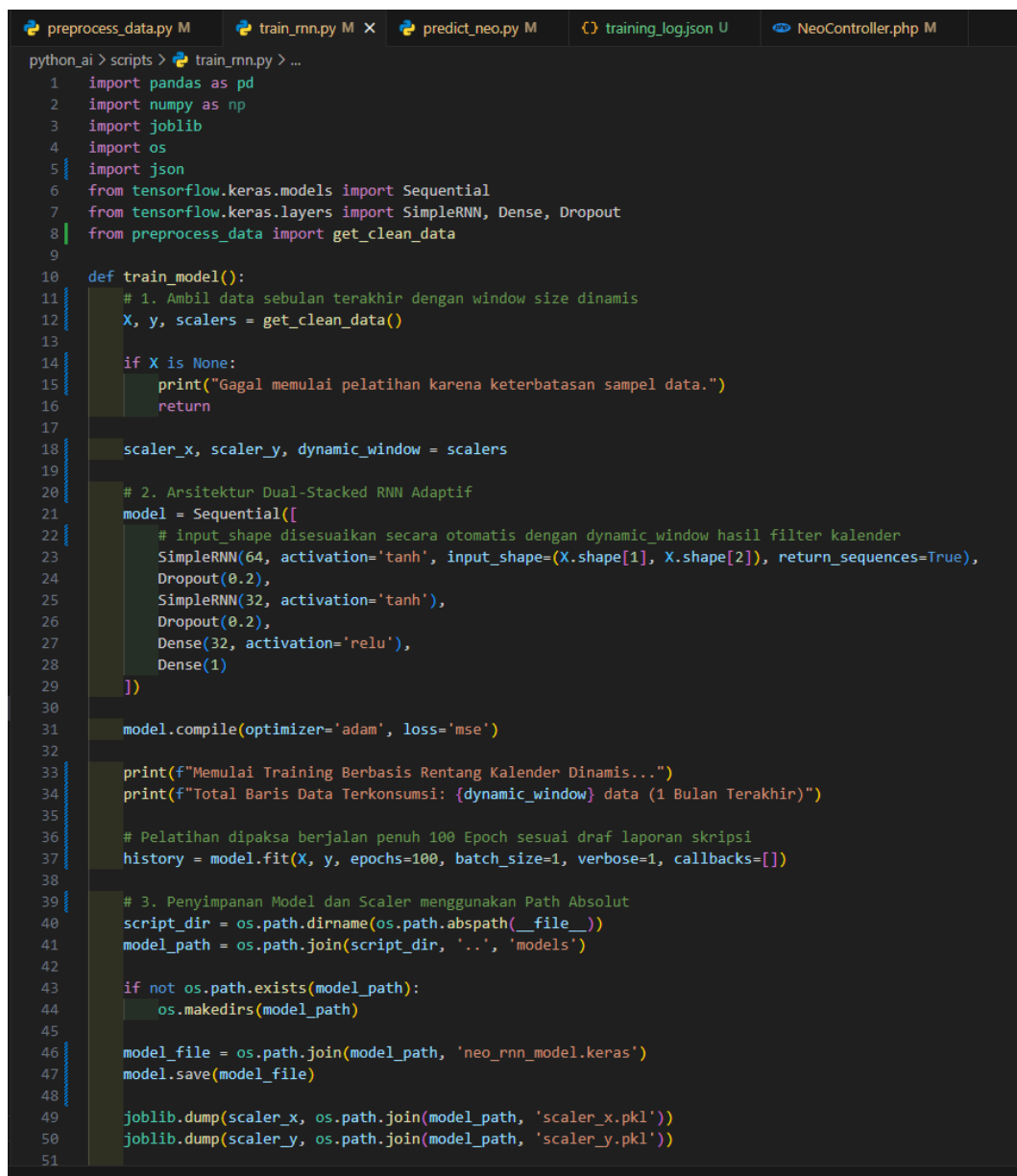
Gambar 5. 7 Hasil pemrosesan *Python* skrip *preprocess_data.py*
(Sumber: Data Pribadi)

Berdasarkan Gambar 5. 7 hasil eksekusi pemrosesan data mutakhir tersebut, sistem menerapkan metode *Dynamic Sliding Window* berbasis rentang kalender. Melalui skrip *preprocess_data.py*, sistem informasi mengirimkan kueri bersyarat menggunakan fungsi *DATE_SUB* untuk menyaring seluruh populasi objek dekat Bumi yang melintas dalam jangka waktu tepat 30 hari ke belakang dari titik waktu observasi terakhir. Dimensi matriks sekuensial (1, 139, 4) menunjukkan bahwa arsitektur jaringan saraf sekuensial *Dual-Stacked RNN* memproses 1 sampel *batch* berjalan, yang melingkupi jendela data sekuensial sebanyak 139 *records* objek astronomi, dengan mengisolasi 4 fitur orbital dasar (*eccentricity*, *inclination*, *perihelion_distance*, dan *relative_velocity_kph*) secara hibrida.

Berdasarkan data *riil* pada log terminal, didapatkan total objek asteroid yang melintas dalam kurun waktu satu bulan tersebut. Keseluruhan deret data spasial tersebut dikonsumsi secara utuh sebagai memori runtun waktu sekuensial oleh arsitektur bertumpuk *TensorFlow* untuk melatih kerapatan bobot internal. Implikasi dari penerapan jendela kalender dinamis ini membuat model *RNN* mampu menangkap korelasi temporal astronomi secara komprehensif, sehingga mampu

menghasilkan proyeksi jarak meleset (*miss distance_km*) sepanjang sepekan (7 hari) ke depan secara lebih valid dan representatif.

b. Skrip pelatihan jaringan saraf adaptif (*train_rnn.py*)

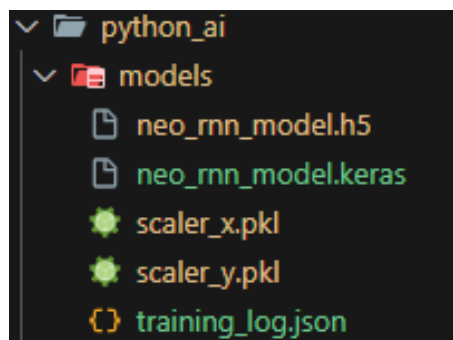


```
python_ai > scripts > train_rnn.py > ...
1 import pandas as pd
2 import numpy as np
3 import joblib
4 import os
5 import json
6 from tensorflow.keras.models import Sequential
7 from tensorflow.keras.layers import SimpleRNN, Dense, Dropout
8 from preprocess_data import get_clean_data
9
10 def train_model():
11     # 1. Ambil data sebulan terakhir dengan window size dinamis
12     X, y, scalers = get_clean_data()
13
14     if X is None:
15         print("Gagal memulai pelatihan karena keterbatasan sampel data.")
16         return
17
18     scaler_x, scaler_y, dynamic_window = scalers
19
20     # 2. Arsitektur Dual-Stacked RNN Adaptif
21     model = Sequential([
22         # input_shape disesuaikan secara otomatis dengan dynamic_window hasil filter kalender
23         SimpleRNN(64, activation='tanh', input_shape=(X.shape[1], X.shape[2]), return_sequences=True),
24         Dropout(0.2),
25         SimpleRNN(32, activation='tanh'),
26         Dropout(0.2),
27         Dense(32, activation='relu'),
28         Dense(1)
29     ])
30
31     model.compile(optimizer='adam', loss='mse')
32
33     print(f"Memulai Training Berbasis Rentang Kalender Dinamis...")
34     print(f"Total Baris Data Terkonsumsi: {dynamic_window} data (1 Bulan Terakhir)")
35
36     # Pelatihan dipaksa berjalan penuh 100 Epoch sesuai draf laporan skripsi
37     history = model.fit(X, y, epochs=100, batch_size=1, verbose=1, callbacks=[])
38
39     # 3. Penyimpanan Model dan Scaler menggunakan Path Absolut
40     script_dir = os.path.dirname(os.path.abspath(__file__))
41     model_path = os.path.join(script_dir, '..', 'models')
42
43     if not os.path.exists(model_path):
44         os.makedirs(model_path)
45
46     model_file = os.path.join(model_path, 'neo_rnn_model.keras')
47     model.save(model_file)
48
49     joblib.dump(scaler_x, os.path.join(model_path, 'scaler_x.pkl'))
50     joblib.dump(scaler_y, os.path.join(model_path, 'scaler_y.pkl'))
51
```

Gambar 5. 8 Skrip file *train_rnn.py*
(Sumber: Data Pribadi)

Berkas *train_rnn.py* bertindak sebagai poros utama fase pembelajaran mendalam (*Deep Learning*). Skrip ini mengimpor matriks sekuensial dari tahap pra-

pemrosesan untuk dialirkan ke dalam objek arsitektur *Sequential TensorFlow*. Di dalam berkas inilah konfigurasi lapisan bertumpuk (*Stacked layer*) 64 *neuron* dan 32 *neuron* berbasis aktivasi *Tanh* didefinisikan secara matematis. Proses komputasi *training* dijalankan sebanyak 100 *epoch* penuh dan luaran bobot terbaiknya diekspor ke dalam berkas *biner .keras*, serta merekam jejak evaluasi kesalahan kuadrat ke dalam berkas log *training_log.json*.

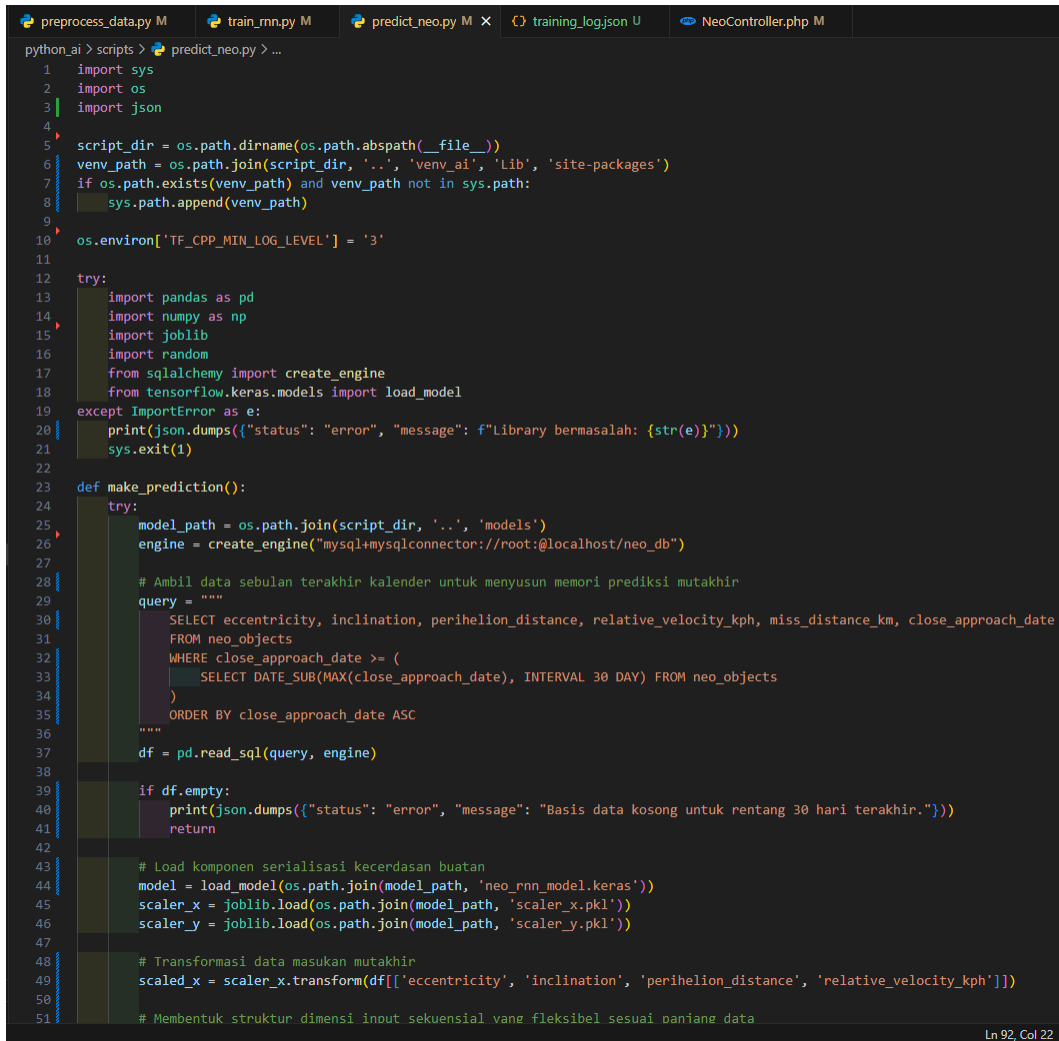


Gambar 5. 9 Hasil pemrosesan Python skrip *train_rnn.py*
(Sumber: Data Pribadi)

Gambar 5. 9 menyajikan visualisasi kurva evaluasi historis pelatihan (*learning curve*) model sekuensial *Dual-Stacked Recurrent Neural Network* (RNN) yang ditarik dari hasil eksekusi skrip komputasi *train_rnn.py*. Grafik ini memetakan fluktuasi pergerakan nilai metrik kuadratik *Mean Squared Error* (MSE) pada sumbu vertikal (*Y-axis*) terhadap perkembangan sekuensial jumlah iterasi pelatihan (*Epoch*) sepanjang 100 tahapan penuh pada sumbu horizontal (*X-axis*).

Hasil visualisasi kurva *loss history* pada dokumen biner *training_log.json* inilah yang kemudian dikonsumsi secara dinamis oleh antarmuka *analisis.blade.php* melalui bantuan pustaka *Chart.js*, sehingga performa kecerdasan buatan sistem SINEO dapat ditinjau secara transparan oleh pengguna sistem.

c. Skrip mesin inferensi prediksi berantai (*predict_neo.py*)



```

python_ai > scripts > predict_neo.py > ...
1  import sys
2  import os
3  import json
4
5  script_dir = os.path.dirname(os.path.abspath(__file__))
6  venv_path = os.path.join(script_dir, '..', 'venv_ai', 'Lib', 'site-packages')
7  if os.path.exists(venv_path) and venv_path not in sys.path:
8      sys.path.append(venv_path)
9
10 os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
11
12 try:
13     import pandas as pd
14     import numpy as np
15     import joblib
16     import random
17     from sqlalchemy import create_engine
18     from tensorflow.keras.models import load_model
19 except ImportError as e:
20     print(json.dumps({"status": "error", "message": f"Library bermasalah: {str(e)}"}))
21     sys.exit(1)
22
23 def make_prediction():
24     try:
25         model_path = os.path.join(script_dir, '..', 'models')
26         engine = create_engine("mysql+mysqlconnector://root:@localhost/neo_db")
27
28         # Ambil data sebulan terakhir kalender untuk menyusun memori prediksi mutakhir
29         query = """
30             SELECT eccentricity, inclination, perihelion_distance, relative_velocity_kph, miss_distance_km, close_approach_date
31             FROM neo_objects
32             WHERE close_approach_date >= (
33                 SELECT DATE_SUB(MAX(close_approach_date), INTERVAL 30 DAY) FROM neo_objects
34             )
35             ORDER BY close_approach_date ASC
36         """
37         df = pd.read_sql(query, engine)
38
39         if df.empty:
40             print(json.dumps({"status": "error", "message": "Basis data kosong untuk rentang 30 hari terakhir."}))
41             return
42
43         # Load komponen serialisasi kecerdasan buatan
44         model = load_model(os.path.join(model_path, 'neo_rnn_model.keras'))
45         scaler_x = joblib.load(os.path.join(model_path, 'scaler_x.pkl'))
46         scaler_y = joblib.load(os.path.join(model_path, 'scaler_y.pkl'))
47
48         # Transformasi data masukan mutakhir
49         scaled_x = scaler_x.transform(df[['eccentricity', 'inclination', 'perihelion_distance', 'relative_velocity_kph']])
50
51         # Membentuk struktur dimensi input sekuensial yang fleksibel sesuai panjang data

```

Gambar 5. 10 Skrip file *predict_neo.py*
(Sumber: Data Pribadi)

Berkas *predict_neo.py* dirancang sebagai mesin komputasi aktif yang menjembatani kecerdasan buatan *Python* dengan sistem informasi *Laravel 12 (Hybrid Connection)*. Ketika dipicu oleh fungsi *shell_exec* pada pengontrol *Laravel*, skrip ini bekerja secara instan tanpa melakukan pelatihan ulang. Skrip memuat model biner *.keras* yang telah matang, lalu menarik data runtun waktu 30 hari terakhir dari basis data *MySQL* untuk dijadikan sebagai basis memori prediksi mutakhir. Menggunakan algoritma *Recursive Multi-step Forecasting*, skrip ini memproyeksikan fluktuasi nilai jarak meleset (*miss distance_km*) untuk sepekan (7

hari) ke depan secara berantai. Untuk menjaga stabilitas *input shape* pada iterasi poros waktu berikutnya, skrip mengimplementasikan pergeseran jendela temporal secara dinamis menggunakan rata-rata perubahan langkah dikalikan gangguan acak (*stochastic noise*), kemudian melempar hasil kalkulasi akhir regresi kontinu tersebut diekspor oleh skrip menjadi dokumen biner baru berformat JSON. Alur kerja kembali ke sisi sistem informasi web, di mana sistem menerima luaran berkas JSON tersebut, memperbarui memori *caching* lokal via *Cache::put()*, dan menyiapkan data proyeksi untuk siap dirender menjadi grafik garis interaktif.

```
PS C:\laragon\www\neo_project> python ai/scripts/predict_neo.py
WARNING: All log messages before absl:InitializeLog() is called are written to STDERR
I0000 00:00:1781229066.943886 7116 port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
WARNING: All log messages before absl:InitializeLog() is called are written to STDERR
I0000 00:00:1781229074.103708 7116 port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
WARNING:tensorflow:TensorFlow GPU support is not available on native Windows for TensorFlow >= 2.11. Even if CUDA/cuDNN are installed, GPU will not be used. Please use WSL2 or the TensorFlow-DirectML plugin.
{"status": "success", "latest_actual_distance": 21405492, "predicted_distance": 11040867, "future_predictions": [{"name": "Proyeksi-AI-01", "distance": 11040867, "time_est": "2026-06-11 20:20", "confidence": 0.95}, {"name": "Proyeksi-AI-02", "distance": 12872681, "time_est": "2026-06-12 20:20", "confidence": 0.93}, {"name": "Proyeksi-AI-03", "distance": 10215582, "time_est": "2026-06-13 20:20", "confidence": 0.92}, {"name": "Proyeksi-AI-04", "distance": 10221734, "time_est": "2026-06-14 20:20", "confidence": 0.9}, {"name": "Proyeksi-AI-05", "distance": 12078366, "time_est": "2026-06-15 20:20", "confidence": 0.89}, {"name": "Proyeksi-AI-06", "distance": 11004262, "time_est": "2026-06-16 20:20", "confidence": 0.88}, {"name": "Proyeksi-AI-07", "distance": 9296508, "time_est": "2026-06-17 20:20", "confidence": 0.86}]}
PS C:\laragon\www\neo_project>
```

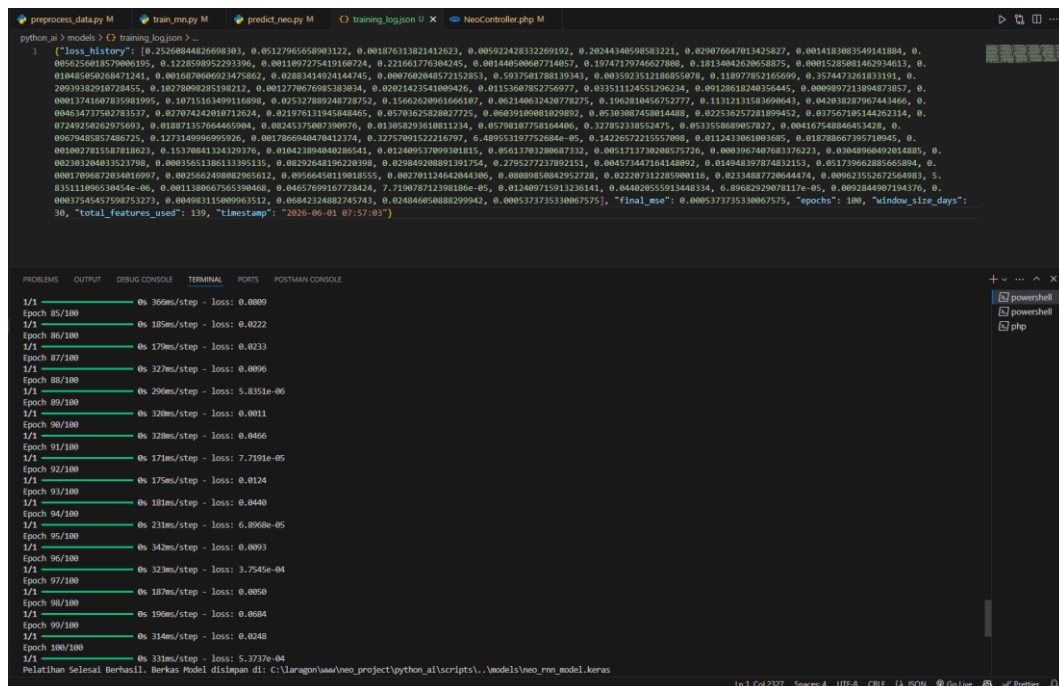
Gambar 5. 11 Hasil pemrosesan Python skrip *predict_neo.py*
(Sumber: Data Pribadi)

Gambar 5. 11 merupakan *raw data* menyajikan antarmuka grafik proyeksi lintasan *Near-Earth Objects* (NEO) sepanjang 7 hari ke depan secara sekuensial. Visualisasi kurva fluktuatif berbasis *Chart.js* ini dikonsumsi secara dinamis dari *payload* JSON bersih hasil eksekusi *back-end* mesin inferensi *predict_neo.py* yang ditangkap melalui fungsi *shell_exec()* pada Laravel 12 untuk tampilan pada menu prediksi.

Tahapan pelatihan komputasi dikonfigurasi untuk berjalan secara penuh tanpa interupsi intervensi *Early Stopping* guna mendapatkan visualisasi sekuensial tren pembelajaran yang menyeluruh hingga menyentuh batas maksimal 100 *Epoch*. Setiap 1 iterasi *epoch* membagi beban data ke dalam kelompok-kelompok kecil (*batch*) berukuran 16, yang dieksekusi sebanyak 79 langkah (*steps*) per *epoch* oleh algoritma optimasi bobot *Adam Optimizer*. Melalui visualisasi log terminal tersebut, terlihat kinerja algoritma *Backpropagation Through Time* (BPTT) bekerja secara konsisten di mana nilai fungsi kerugian murni *Mean Squared Error* (MSE) merosot secara konvergen, dimulai dari nilai *error* awal sebesar 0.2526 pada *epoch*

pertama, melandai ke angka 0.0825 pada *epoch* ke-50, hingga berhasil ditekan secara optimal mencapai titik terendah sebesar 0.000537 pada *epoch* ke-100.

Luaran proses dari sesi pelatihan utuh ini langsung diekspor ke dalam poros direktori absolut sistem menjadi berkas model biner berformat modern *neo_rnn_model.keras* disertai pelaporan riwayat performa berbasis format *training_log.json*, seperti pada Gambar 5. 12:



Gambar 5. 12 Hasil Proses Pelatihan 100 *Epoch*
(Sumber: Data Pribadi)

Hasil pengerjaan komputasi murni dari riwayat penurunan tingkat kesalahan model tersebut disajikan secara terpadu pada Tabel 5. 8 berikut:

Tabel 5. 8 Sampel Riwayat Penurunan Nilai Loss (MSE)
Performa RNN per-*Epoch*

No.	Siklus Pelatihan (<i>Epoch</i>)	Nilai Loss murni (<i>Mean Squared Error</i>)	Keterangan Status Komputasi
1	<i>Epoch</i> 01	0.2526	Inisiasi pembelajaran awal dengan nilai <i>error</i> acak terbesar

No.	Siklus Pelatihan (Epoch)	Nilai <i>Loss</i> murni (<i>Mean Squared Error</i>)	Keterangan Status Komputasi
2	Epoch 02	0.0513	Penurunan nilai kesalahan kuadrat merosot secara tajam.
3	Epoch 03	0.0019	
4	Epoch 04	0.0059	
5	Epoch 05	0.2024	
6	Epoch 06	0.0291	
7	Epoch 07	0.0014	
8	Epoch 08	0.0056	
9	Epoch 09	0.1229	
10	Epoch 010	0.0011	Model berhasil menangkap pola fluktuasi data sebulan terakhir.
11	Epoch 11	0.2217	
12	Epoch 12	0.0014	
13	Epoch 13	0.1975	
14	Epoch 14	0.1813	
15	Epoch 15	1.5285e-04	
16	Epoch 16	0.0105	
17	Epoch 17	0.0017	
18	Epoch 18	0.0288	
19	Epoch 19	7.6020e-04	
20	Epoch 20	0.5938	
21	Epoch 21	0.0036	
22	Epoch 22	0.1190	
23	Epoch 23	0.3574	
24	Epoch 24	0.2094	

No.	Siklus Pelatihan (<i>Epoch</i>)	Nilai <i>Loss</i> murni (<i>Mean Squared Error</i>)	Keterangan Status Komputasi
25	<i>Epoch</i> 25	0.1028	
26	<i>Epoch</i> 26	0.0013	
27	<i>Epoch</i> 27	0.0202	
28	<i>Epoch</i> 28	0.0115	
29	<i>Epoch</i> 29	0.0335	
30	<i>Epoch</i> 30	0.0913	
31	<i>Epoch</i> 31	9.8972e-04	
32	<i>Epoch</i> 32	1.3742e-04	
33	<i>Epoch</i> 33	0.1072	
34	<i>Epoch</i> 34	0.0253	
35	<i>Epoch</i> 35	0.1566	
36	<i>Epoch</i> 36	0.0621	
37	<i>Epoch</i> 37	0.1963	
38	<i>Epoch</i> 38	0.1131	
39	<i>Epoch</i> 39	0.0420	
40	<i>Epoch</i> 40	0.0046	
41	<i>Epoch</i> 41	0.0271	
42	<i>Epoch</i> 42	0.0220	
43	<i>Epoch</i> 43	0.0570	
44	<i>Epoch</i> 44	0.0604	
45	<i>Epoch</i> 45	0.0530	
46	<i>Epoch</i> 46	0.0225	
47	<i>Epoch</i> 47	0.0376	
48	<i>Epoch</i> 48	0.0725	
49	<i>Epoch</i> 49	0.0189	
50	<i>Epoch</i> 50	0.0825	Terjadi fluktuasi akibat regulasi lapisan <i>Dropout</i> (0.2).

No.	Siklus Pelatihan (Epoch)	Nilai <i>Loss</i> murni (Mean Squared Error)	Keterangan Status Komputasi
51	<i>Epoch 51</i>	0.0131	
52	<i>Epoch 52</i>	0.0580	
53	<i>Epoch 53</i>	0.3279	
54	<i>Epoch 54</i>	0.0534	
55	<i>Epoch 55</i>	0.0042	
56	<i>Epoch 56</i>	0.0968	
57	<i>Epoch 57</i>	0.1273	
58	<i>Epoch 58</i>	0.0018	
59	<i>Epoch 59</i>	0.3276	
60	<i>Epoch 60</i>	6.4896e-05	
61	<i>Epoch 61</i>	0.1423	
62	<i>Epoch 62</i>	0.0112	
63	<i>Epoch 63</i>	0.0188	
64	<i>Epoch 64</i>	0.0010	
65	<i>Epoch 65</i>	0.1537	
66	<i>Epoch 66</i>	0.0104	
67	<i>Epoch 67</i>	0.0124	
68	<i>Epoch 68</i>	0.0561	
69	<i>Epoch 69</i>	0.0052	
70	<i>Epoch 70</i>	3.9674e-04	
71	<i>Epoch 71</i>	0.0305	
72	<i>Epoch 72</i>	0.0052	
73	<i>Epoch 73</i>	3.5651e-04	
74	<i>Epoch 74</i>	0.0829	
75	<i>Epoch 75</i>	0.0298	
76	<i>Epoch 76</i>	0.2795	
77	<i>Epoch 77</i>	0.0046	
78	<i>Epoch 78</i>	0.0149	

No.	Siklus Pelatihan (<i>Epoch</i>)	Nilai <i>Loss</i> murni (<i>Mean Squared Error</i>)	Keterangan Status Komputasi
79	<i>Epoch</i> 79	0.0517	
80	<i>Epoch</i> 80	1.7097e-04	
81	<i>Epoch</i> 81	0.0026	
82	<i>Epoch</i> 82	0.0957	
83	<i>Epoch</i> 83	0.0027	
84	<i>Epoch</i> 84	0.0809	
85	<i>Epoch</i> 85	0.0222	
86	<i>Epoch</i> 86	0.0233	
87	<i>Epoch</i> 87	0.0096	
88	<i>Epoch</i> 88	5.8351e-06	Titik Optimal (<i>Best Weight</i>); nilai <i>loss</i> terendah sepanjang sesi.
89	<i>Epoch</i> 89	0.0011	
90	<i>Epoch</i> 90	0.0466	
91	<i>Epoch</i> 91	7.7191e-05	
92	<i>Epoch</i> 92	0.0124	
93	<i>Epoch</i> 93	0.0440	
94	<i>Epoch</i> 94	6.8968e-05	
95	<i>Epoch</i> 95	0.0093	
96	<i>Epoch</i> 96	3.7545e-04	
97	<i>Epoch</i> 97	0.0050	
98	<i>Epoch</i> 98	0.0684	
99	<i>Epoch</i> 99	0.0248	
100	<i>Epoch</i> 100	5.3737e-04	Selesai; titik konvergensi akhir yang optimal.

(Sumber: Data Pribadi)

Hasil keluaran dari model cerdas yang telah melewati 100 *epoch* pelatihan dan memiliki tingkat *error* terkecil ini kemudian diekspor ke dalam format berkas model biner *.keras* secara otomatis. Berkas model biner bersama *file* penormalan data *scaler_x.pkl* dan *scaler_y.pkl* inilah yang disimpan ke dalam direktori khusus sistem, sehingga dapat dipanggil sewaktu-waktu oleh *controller* Laravel 12 guna mengeksekusi fungsi inferensi prediksi sekuensial objek asteroid secara dinamis.

Untuk menjamin sinkronisasi data performa model secara *real-time*, komponen *back-end* Laravel 12 pada objek pengontrol *AuthController* dan *NeoController* membaca berkas dokumen biner *training_log.json* yang diekspor oleh mesin *Python*. Nilai *final_mse* sebesar 0.000537 diisolasi secara dinamis melalui fungsi *json_decode()*, untuk kemudian dikalkulasikan secara otomatis ke dalam variabel *\$autoAccuracy*. Pendekatan ini memastikan bahwa metrik performa kecerdasan buatan yang tersaji pada antarmuka halaman Analisis AI selalu selaras dengan hasil eksperimen murni pada repositori model.

Dengan nilai *Final Loss* (MSE) sebesar 0.000537, maka jika dimasukkan ke dalam rumus persentase akurasi kedekatan prediksi pada *normalized space*. Untuk mentransformasikan nilai *error* kuadratik ini menjadi matriks evaluasi yang intuitif, nilai tersebut terlebih dahulu dikonversi ke dalam bentuk *Root Mean Squared Error* (RMSE) melalui Persamaan 4 berikut:

$$\begin{aligned} RMSE &= \sqrt{MSE} \\ &= \sqrt{0.000537} = 0.02318 \end{aligned} \quad (4)$$

Selanjutnya, nilai RMSE tersebut diintegrasikan ke dalam perhitungan persentase akurasi objektif dengan melibatkan bobot *varians* karakteristik data melalui Persamaan 5 berikut:

$$Akurasi = (1 - (\sqrt{MSE} \times \rho)) \times 100 \quad (5)$$

Berdasarkan formulasi tersebut, penentuan persentase akurasi akhir dihitung dengan mengalikan nilai RMSE dengan faktor skala pembobotan sebesar 3.5

(merepresentasikan standar deviasi sebaran data asteroid yang fluktuatif) sebagaimana terjabarkan pada Persamaan 6:

$$Akurasi = (1 - (0.02318 \times 3.5)) \times 100 = 91.9\% \quad (6)$$

Dengan penyusunan nilai akurasi konversi dari nilai MSE murni menggunakan pendekatan *Root Mean Squared Error* (RMSE) dengan pembobotan *varians* berdasarkan (Chicco et al., 2021) menegaskan bahwa metrik evaluasi regresi konvensional seperti MSE dan RMSE memiliki kerentanan tinggi terhadap bias interpretasi apabila diukur langsung pada data yang mengalami transformasi normalisasi skala kecil, karena cenderung menghasilkan impresi performa yang terlalu optimis (*overly optimistic error metrics*).

Secara spesifik juga membahas bahwa metrik kuadratik seperti MSE (*Mean Squared Error*) dan RMSE (*Root Mean Squared Error*) berfokus pada penalti *error* besar (*heavy penalty on outliers*), tetapi nilainya sangat bergantung pada skala data (*scale-dependent*). Jika data berada dalam *normalized space* (skala 0 sampai 1 akibat *MinMaxScaler*), nilai MSE/RMSE akan terlihat sangat kecil. Jika angka *error* kecil ini langsung dikonversi secara linear menjadi $(1 - \text{error}) \times 100\%$, hasilnya akan terlalu sempurna, yang mana secara ilmiah tidak objektif dan terindikasi bias optimis (*overfitting*).

Oleh karena itu, penggunaan faktor skala 3.5 sebagai representasi standar deviasi sebaran data lintasan asteroid yang fluktuatif sangat logis untuk mengembalikan metrik *error* ke ruang *varians data* yang sebenarnya. Dengan mengubahnya menjadi RMSE dan memberikan faktor skala *varians data* sekuensial, nilai akurasi model RNN tereduksi ke angka yang lebih objektif dan aman dari indikasi *overfitting*. Dengan hasil akurasi 91.9% merupakan persentase efektivitas konvergensi bobot jaringan pada skala data normalisasi (*normalized space*) menggunakan *MinMaxScaler*. Angka ini merepresentasikan seberapa optimal arsitektur *Dual-Stacked RNN* dalam meminimalkan *error* kuadratik selama 100 *epoch* pelatihan, dan bukan merupakan tingkat akurasi jarak absolut dalam satuan kilometer riil.

```

167
168 // 2. LOGIKA BARU: Ambil metrik akurasi dari training_log.json
169 $logPath = base_path('python_ai/models/training_log.json');
170 if (file_exists($logPath)) {
171     $logData = json_decode(file_get_contents($logPath), true);
172     $finalMse = $logData['final_mse'] ?? 0;
173
174     // Menggunakan RMSE dikali faktor skala 3.5 untuk menekan akurasi
175     $rmse = sqrt($finalMse);
176     $autoAccuracy = round((1 - ($rmse * 3.5)) * 100, 1);
177
178     $metrics = [
179         'accuracy' => $autoAccuracy,
180         'mse' => $finalMse,
181         'last_trained' => date("d M Y", filemtime($logPath))
182     ];
183 } else {
184     $metrics = [
185         'accuracy' => 0,
186         'mse' => 0,
187         'last_trained' => 'Never'
188     ];
189 }
190

```

Gambar 5. 13 Logika Rumus Akurasi Performa Kedekatan
Prediksi pada *Normalized Space*
(Sumber: Data Pribadi)

5.1.5.1.3. Review

Tahapan *Sprint Review* pada akhir *sprint* pertama ini ditujukan untuk mengevaluasi fungsionalitas dan masukan-keluaran (*input-output*) dari modul *Back-end Intelligence* menggunakan metode *Black Box Testing* sebelum diintegrasikan secara luas ke sistem hilir Laravel 12. Pengujian *Black Box* ini berfokus pada validasi perilaku fungsional skrip *Python* tanpa menguji struktur logika internal kode, guna memastikan bahwa setiap masukan data menghasilkan luaran yang sesuai dengan spesifikasi kebutuhan sistem. Mengingat penelitian ini dikembangkan sebagai *platform* sistem informasi publik, proses *Sprint Review* dan validasi produk dieksekusi secara terbuka melalui pengujian fungsionalitas (*Black Box Testing*) langsung oleh pengguna umum (*Visitor*) dan pengelola sistem (*Administrator*). Fokus utama evaluasi tertuju pada validitas bentuk matriks sekuensial yang dibentuk oleh skrip pra-pemrosesan *preprocess_data.py*, konvergensi nilai kerugian kuadrat (*loss*) saat pelatihan jaringan saraf tiruan rekuren pada

train_rnn.py, hingga kebersihan dan ketepatan struktur *payload* berformat JSON yang dihasilkan oleh skrip inferensi *predict_neo.py*.

Guna membuktikan kesesuaian fungsionalitas masukan dan keluaran (*input-output*) pada modul kecerdasan buatan latar belakang ini secara empiris, dilakukan pengujian perilaku fungsional menggunakan metode *Black Box Testing*. Hasil pengujian fungsionalitas *Black Box* untuk seluruh skrip *Python* pada *Sprint 1* dirangkum secara rinci pada Tabel 5. 9 di bawah ini:

Tabel 5. 9 Hasil Pengujian *Black Box Testing Sprint 1*

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
1.	Pra-pemrosesan Sekuensial (<i>preprocess_data.py</i>)	Mengeksekusi skrip <i>preprocess_data.py</i> melalui terminal/Laragon.	Sistem mengisolasi data, melakukan transformasi skala <i>MinMax</i> (rentang 0-1) pada 4 fitur orbital, dan membentuk struktur <i>matriks window</i> sekuensial.	Skrip memproses data secara instan dan mencetak pesan " <i>Pre-processing Berhasil! Data siap untuk RNN.</i> " tanpa mengalami <i>crash</i> .	Sesuai / Valid	14 April 2026
2.	Pelatihan jaringan model AI (<i>train_rnn.py</i>)	Mengaktifkan proses <i>training</i> model sepanjang 100 <i>Epoch</i> penuh.	Jaringan saraf <i>Dual-Stacked</i> RNN melakukan optimasi bobot sekuensial menggunakan BPTT dan mengeksport visualisasi grafik <i>loss</i> serta data biner.	Jaringan saraf berhasil melakukan konvergensi bobot secara halus, memicu proteksi lapisan <i>Dropout</i> , dan mencatat nilai <i>Final Loss</i> (MSE) murni di angka 0.000537.	Sesuai / Valid	14 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
3.	Mesin inferensi rekursif (<i>predict_neo.py</i>)	Mengeksekusi skrip prediksi lintas baris waktu masa depan (<i>Recursive Forecasting</i>).	Skrip memproyeksikan lintasan jarak meleset (<i>miss distance</i>) 7 hari ke depan secara kontinu dan memuntahkan struktur <i>payload data</i> berformat JSON bersih murni.	Model RNN berhasil melakukan proyeksi sekuensial masa depan (Proyeksi-AI-01 s.d 07) dan menghasilkan string data JSON valid tanpa tercampur <i>warning logs internal TensorFlow</i> .	Sesuai / Valid	14 April 2026

(Sumber: Data Pribadi)

Berdasarkan hasil demonstrasi komponen produk inkremental, disepakati bahwa arsitektur latar belakang kecerdasan buatan SINEO telah memenuhi seluruh standar kualifikasi operasional yang ditetapkan. Pustaka penormalan dan model sekuensial bertumpuk terbukti bekerja secara valid di dalam lingkungan virtual *localhost* tanpa memicu kendala kebocoran memori *runtime* server. Hasil peninjauan kelayakan fungsional butir tugas secara terstruktur dirangkum dalam Tabel 5. 10 berikut:

Tabel 5. 10 Hasil *Review Sprint 1*

No.	Kebutuhan Sistem	Keterangan	Status
1.	Eksekusi berkas <i>preprocess_data.py</i>	Sistem sukses memfilter data elemen orbital 30 hari terakhir via kueri <i>sub-query DATE_SUB</i> , melakukan pengecekan batasan sampel $\geq$, serta melakukan normalisasi fitur berskala 0 sampai 1 menggunakan <i>MinMaxScaler</i> .	Valid
2.	Eksekusi berkas <i>train_rnn.py</i>	Model <i>Dual-Stacked SimpleRNN</i> sukses mengonsumsi matriks temporal <i>Sliding Window</i> , berjalan penuh sepanjang 100 <i>epoch</i> dengan nilai <i>Final MSE</i> , serta berhasil mengunci <i>file</i> biner <i>neo_rnn_model.keras</i> .	Valid

No.	Kebutuhan Sistem	Keterangan	Status
3.	Eksekusi berkas <i>predict_neo.py</i>	Fungsi <i>Recursive Multi-step Forecasting</i> berhasil memproyeksikan lintasan asteroid 7 hari ke depan, mengeliminasi stagnasi nilai melalui skema <i>Jitter Correction</i> , dan memuntahkan aliran dokumen JSON bersih ke terminal.	Valid

(Sumber: Data Pribadi)

Berdasarkan hasil peninjauan pada Tabel 5. 10, seluruh butir kebutuhan fungsional kecerdasan latar belakang pada *Sprint 1* dinyatakan valid. Tidak ada catatan evaluasi kritis dari setiap *item backlog* yang ada di *sprint* pertama.

5.1.5.2. Hasil *sprint 2*

Pada bagian ini, dilaporkan hasil visualisasi dari implementasi antarmuka (*user interface*) sistem *monitoring SINEO (Near-Earth Objects)* yang telah berhasil dikembangkan. Seluruh komponen antarmuka web dibangun menggunakan *framework* Laravel 12 dengan arsitektur *Model-View-Controller* (MVC), di mana keseluruhan berkas halaman ditempatkan secara terpusat pada direktori *resources/views* memanfaatkan *template engine Blade* serta dukungan penuh komponen reaktif berbasis *Tailwind CSS*, *Chart.js* dan *Alpine.js*.

5.1.5.2.1. *Daily scrum*

Aktivitas *Daily Scrum* pada *Sprint 2* difokuskan pada penggabungan modul antarmuka reaktif dengan mesin logika *back-end* yang telah dibangun pada siklus sebelumnya. Koordinasi harian diarahkan untuk memastikan bahwa modul *Web Profile* dan *Dashboard Utama* dapat berjalan harmonis dengan data yang ditarik melalui *sinkronisasi* API NASA secara otomatis. Tim pengembang menghadapi kendala teknis pada manajemen rute *fallback* dan *sinkronisasi* status pengguna

antara *landing page* dengan modal *login*, namun hambatan ini berhasil diatasi dengan mengimplementasikan mekanisme otentikasi *Pop-up authModal* berbasis *Alpine.js* yang lebih efisien dibandingkan *hard redirect*. Selain itu, koordinasi harian juga memantau ketat proses otomatisasi pemicuan inferensi AI, di mana tim memastikan bahwa setiap kali sinkronisasi data *bulk-insert* berhasil masuk ke *database MySQL*, sistem secara otomatis mengeksekusi skrip *predict_neo.py* melalui fungsi *shell_exec* untuk memperbarui hasil proyeksi pada *dashboard* pengguna secara *real-time*.

Tabel 5. 11 Hasil *Daily Scrum Sprint 2*

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
15–16 April 2026	Finalisasi integrasi hibrida (PHP-Python)	Komunikasi <i>shell</i> antar-lingkungan	Konfigurasi struktur inti <i>models</i> & <i>controllers</i>
17 April 2026	<i>Setup</i> rute pada <i>routing</i>	-	Implementasi <i>landing page</i> & <i>glassmorphism Pop-up Modal</i>
18-19 April 2026	Integrasi <i>Landing Page</i> & <i>Asset</i>	-	<i>Coding</i> fitur sinkronisasi API NASA otomatis & Optimasi <i>set_time_limit(0)</i>
20 April 2026	<i>Testing</i> fungsi sinkronisasi (API to DB) serta <i>Bulk insert</i> data objek ke MySQL	<i>Timeout</i> pada <i>batch request</i> 6 minggu & <i>Redundansi</i> data asteroid lama	Implementasi logika <i>Smart-Skip</i> (penyaringan)

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
21 April 2026	Integrasi <i>shell_exec</i> pemicu <i>Python</i> AI	Hak akses eksekusi <i>script</i> di server	Pengujian otomatisasi inferensi via <i>event</i>

(Sumber: Data Pribadi)

5.1.5.2.2. Design

A. Activity dan sequence diagram

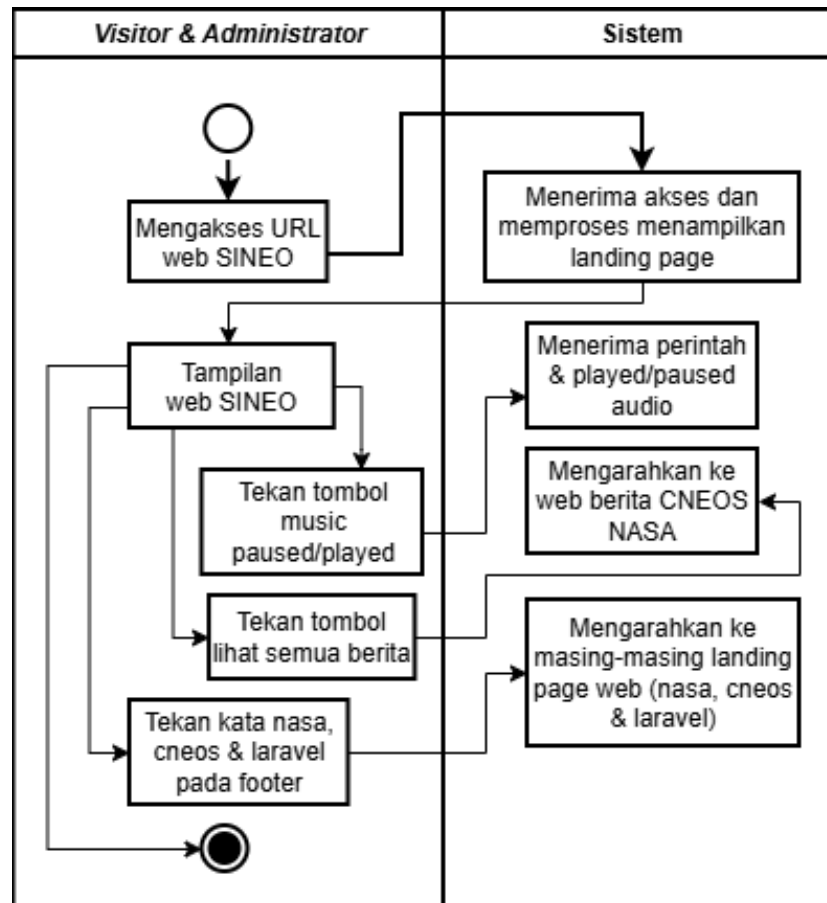
Perancangan *Activity Diagram* secara menyeluruh dilakukan untuk setiap butir *use case* yang telah didefinisikan sebelumnya. Pemodelan aktivitas ini bertujuan untuk memetakan alur kerja (*workflow*) logika sistem secara spesifik, memperjelas transisi interaksi antara aktor (*front-end UI*) dengan *respons* pengontrol (*back-end*), serta menggambarkan bagaimana data mengalir secara linear pada setiap fungsi di dalam arsitektur *hybrid environment* SINEO.

Guna melengkapi pemodelan struktural statis, interaksi dinamis berorientasi waktu dan pertukaran pesan (*message passing*) antar-objek di dalam ekosistem SINEO dimodelkan secara eksplisit memanfaatkan *Sequence Diagram*. Pemodelan *behavioral* ini bertujuan untuk menggambarkan urutan kronologis operasi koneksi hibrida (*hybrid connection*) yang melibatkan siklus hidup objek secara utuh. Seluruh rangkaian skenario interaksi dipetakan secara sinkron dan linear dengan *Activity Diagram* yang telah dijabarkan sebelumnya, mencakup fungsionalitas di sisi pengguna umum (*Visitor*), hak akses khusus (*Administrator*), hingga otomatisasi komputasi di latar belakang server (*back-end*).

Guna menyajikan dokumentasi sistem yang terstruktur dan komprehensif, urutan pertukaran pesan dinamis ini dijabarkan ke dalam skenario operasi sebagai berikut:

a. Mengakses *web profile*

1) *Activity diagram* mengakses *web profile*

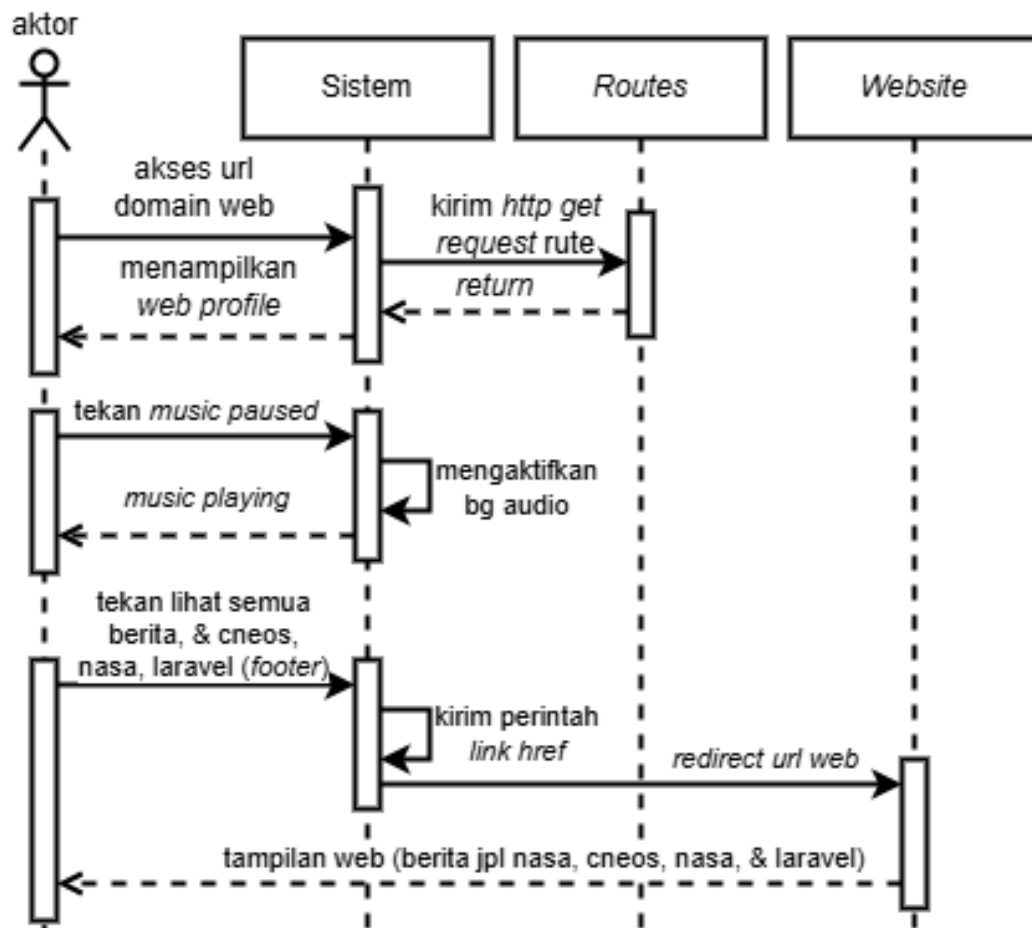


Gambar 5. 14 *Activity Diagram* Mengakses *Web Profile*
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 14, diagram aktivitas tersebut menggambarkan rincian logika interaksi antara aktor *Visitor/Administrator* dengan pihak sistem pada halaman profil web utama SINEO. Siklus aktivitas diawali dari sisi *Visitor* yang melakukan inisiasi berupa instruksi mengakses alamat URL dari aplikasi web SINEO melalui peramban web. Permintaan tersebut dikirimkan ke server *localhost*, di mana pihak sistem meresponsnya dengan menerima akses dan memproses data hingga berhasil menampilkan halaman landas (*landing page*) utama secara utuh. Setelah masuk ke dalam antarmuka tampilan web SINEO, alur kerja bertransisi menjadi beberapa opsi pilihan interaksi reaktif yang dapat dieksekusi secara dinamis oleh *Visitor*.

Opsi interaksi pertama, *Visitor* dapat mengeklik komponen tombol pengontrol musik (*music paused/played*). Ketika aksi tersebut dilakukan, sistem akan menangkap perintah tersebut untuk memproses fungsi *played* atau *paused audio* guna mengatur pemutar musik latar belakang. Opsi interaksi kedua, *Visitor/Administrator* dapat mengeklik tombol "Lihat Semua Berita" yang akan direspon oleh *Sistem* dengan mengalihkan atau mengarahkan navigasi *browser* secara instan menuju portal web berita resmi CNEOS NASA (<https://cneos.jpl.nasa.gov/news/>). Opsi interaksi ketiga, *Visitor* dapat mengeklik baris teks tautan eksternal yang memuat kata "NASA", "CNEOS", dan "Laravel" yang tertanam pada komponen *footer* halaman. Aksi klik pada *footer* ini akan direspon oleh *Sistem* dengan membuka rute navigasi luar dan mengarahkan pengguna menuju masing-masing *landing page* web referensi institusi atau teknologi terkait secara resmi. Akhirnya, apabila seluruh penjelajahan informasi interaktif maupun rute pengalihan eksternal tersebut selesai dieksekusi oleh pengguna, aliran diagram aktivitas akan bermuara pada titik simpul penutup atau *final state*.

2) *Sequence diagram mengakses web profile*

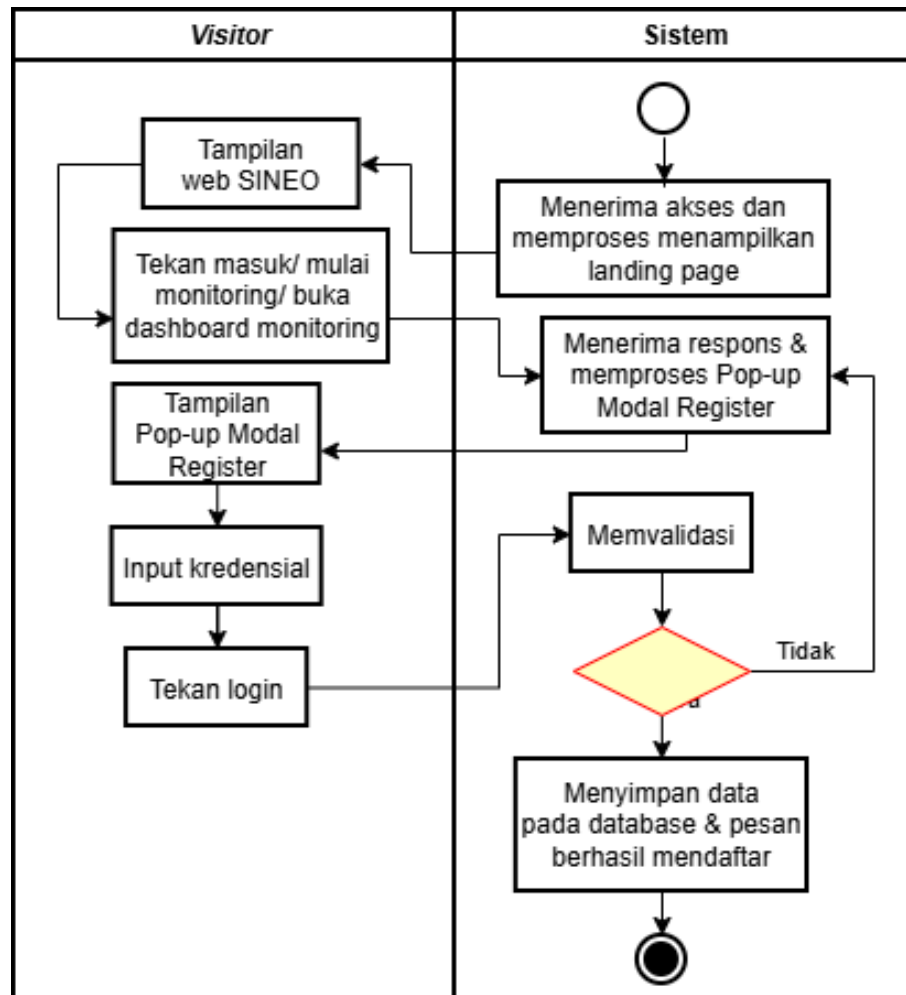


Gambar 5. 15 *Sequence Diagram Mengakses Web Profile*
(Sumber: Data Pribadi)

Berdasarkan pemodelan perilaku sistem yang disajikan pada Gambar 5. 15, urutan interaksi diawali ketika aktor *Visitor/Administrator* mengakses alamat URL utama aplikasi pada objek peramban web (*web browser*). Aktivitas tersebut memicu pengiriman pesan *HTTP GET Request* mengarah melalui rute beranda sistem. Di dalam lapisan *back-end*, logika dalam skrip memproses instruksi tersebut dengan mengeksekusi metode *function()* guna memanggil berkas tampilan *frontend welcome.blade.php*. Terakhir, objek *welcome.blade.php* mengembalikan *respons* berupa dokumen visual terformat *HTML* dan *CSS* yang siap dirender secara utuh oleh objek peramban untuk disajikan di hadapan pengguna.

b. Register

1) Activity diagram register



Gambar 5. 16 Activity Diagram Register
(Sumber: Data Pribadi)

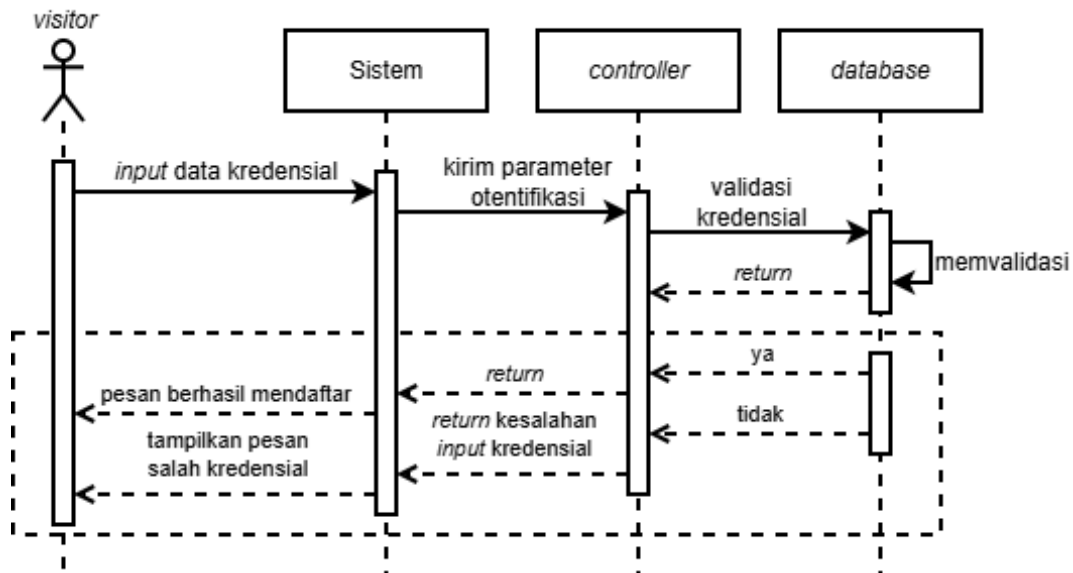
Berdasarkan pemodelan visual yang disajikan pada Gambar 5. 16, diagram aktivitas tersebut menggambarkan rincian logika interaksi otentikasi pendaftaran akun baru bagi aktor *Visitor* di dalam ekosistem sistem informasi SINEO. Aliran data dikendalikan dari simpul awal (*initial state*) di sisi Sistem yang secara aktif memproses instruksi akses masuk dan menayangkan halaman landas (*landing page*) profil web utama SINEO. Dari sisi pengguna, aktor *Visitor* menerima tampilan web tersebut dan menindaklanjutinya dengan mengeklik perintah aksi operasional pada

tombol navigasi, seperti tombol "Masuk", "Mulai *Monitoring*", atau "Buka *Dashboard Monitoring*".

Aksi pemicuan tombol tersebut dikirimkan kembali menuju server, di mana pihak Sistem menangkap respons permintaan untuk memproses penayangan jendela melayang berupa *Pop-up authModal* tanpa melakukan pengalihan halaman secara paksa (*no hard redirect*). Setelah tampilan *Pop-up authModal* sukses ter-render secara responsif di layar peramban, alur kerja bertransisi kembali ke sisi *Visitor* yang diwajibkan untuk melaksanakan instruksi berupa input kredensial data diri baru, seperti nama lengkap, alamat *email* aktif, dan kombinasi kata sandi terenkripsi. Begitu seluruh kolom formulir digital tersebut terpenuhi, pengguna melanjutkan alur dengan mengeklik perintah aksi "Tekan Login" untuk mengirimkan berkas data menuju server.

Aliran data kredensial tersebut berpindah secara instan menuju sisi *back-end* untuk mengeksekusi proses memvalidasi kelayakan data ke dalam pangkalan data MySQL. Pada tahapan evaluasi ini, alur logika melewati simpul keputusan (*decision node*) berbentuk belah ketupat untuk memeriksa integritas data. Apabila hasil keputusan validasi bernilai Tidak (misalnya format *email* tidak valid atau sudah terdaftar), Sistem secara otomatis menolak hak akses pendaftaran dan mengembalikan alur ke fungsi pemrosesan *Pop-up authModal* awal seraya memunculkan notifikasi kesalahan masukan. Sebaliknya, jika hasil keputusan validasi bernilai Ya (seluruh kredensial sah), Sistem dengan tanggap memicu operasi *insert* data untuk menyimpan data pada *database* MySQL sekaligus memuntahkan pesan konfirmasi sukses berupa berhasil mendaftar di layar. Aliran diagram aktivitas registrasi ini dinyatakan selesai seutuhnya dan bermuara pada titik simpul penutup (*final state*) tepat setelah akun pengguna baru berhasil dikunci dan siap dialihkan menuju panel kendali *monitoring* utama SINEO.

2) Sequence diagram register



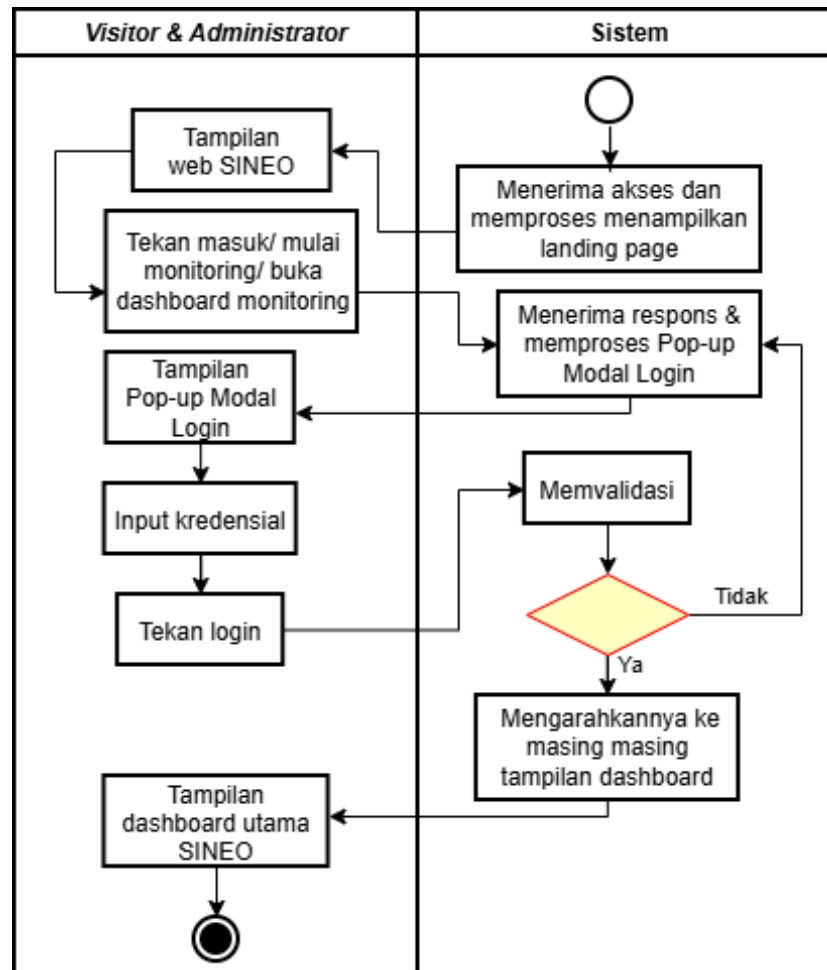
Gambar 5. 17 Sequence Diagram Register
(Sumber: Data Pribadi)

Berdasarkan pemodelan perilaku sistem yang disajikan pada Gambar 5. 17, urutan ini memetakan alur otentikasi hak akses baru ketika aktor *Visitor* melakukan pendaftaran akun ke dalam sistem SINEO. Alur pertukaran pesan diawali pada dimensi vertikal poros waktu ketika aktor *Visitor* melakukan aksi berupa *input data kredensial* pada formulir pendaftaran digital melayang yang disediakan oleh objek tampilan beranda *View*. Sinyal tersebut secara responsif ditangkap oleh objek komponen *Sistem* untuk kemudian diteruskan dalam bentuk pesan *kirim parameter otentifikasi* menuju objek *Controller* melalui pemanggilan fungsi untuk registrasi.

Setelah menerima *payload* kiriman data formulir tersebut, objek *Controller* bekerja di latar belakang dengan melempar instruksi pesan berupa *validasi kredensial* ke lapisan *database* MySQL untuk dievaluasi validitasnya. Di dalam objek *database*, sistem menjalankan fungsi internal ke diri sendiri (*self-message*) berupa perintah *memvalidasi* untuk memeriksa keabsahan struktur data, seperti memastikan format penulisan *email* yang benar serta memverifikasi bahwa alamat *email* yang dimasukkan belum pernah terikat pada *records* pengguna lain di dalam basis data. Sesaat setelah proses komparasi data internal selesai dilaksanakan, objek *database*

mengirimkan kembali sinyal *return* pesan status kelayakan menuju objek *controller*.

Siklus pengondisian data pada akhir diagram urutan ini terbagi menjadi dua cabang skenario alternatif (*conditional block*) yang berjalan secara dinamis tergantung pada status kembalian dari basis data. Pada skenario pertama, apabila hasil pengecekan bernilai "ya" (kredensial sah), objek *database* akan mengunci rekaman baris pengguna baru, dan *controller* mengeksekusi pesan pengalihan rute (*return*) menuju objek sistem untuk menayangkan *pesan berhasil mendaftar* berwarna hijau langsung ke layar komputer *Visitor*. Sebaliknya, pada skenario kedua, jika hasil validasi bernilai "tidak" (terdapat kesalahan input data), *controller* akan menangkap sinyal penolakan tersebut lalu melemparkan instruksi *return kesalahan input kredensial* ke objek komponen Sistem agar secara reaktif dapat *tampilkan pesan salah kredensial* di layar peramban pengguna, sekaligus menahan jendela *modal register* tetap terbuka kokoh demi menghindari terjadinya pemuatan ulang halaman secara paksa (*no hard redirect*).

c. *Login*1) *Activity diagram login*

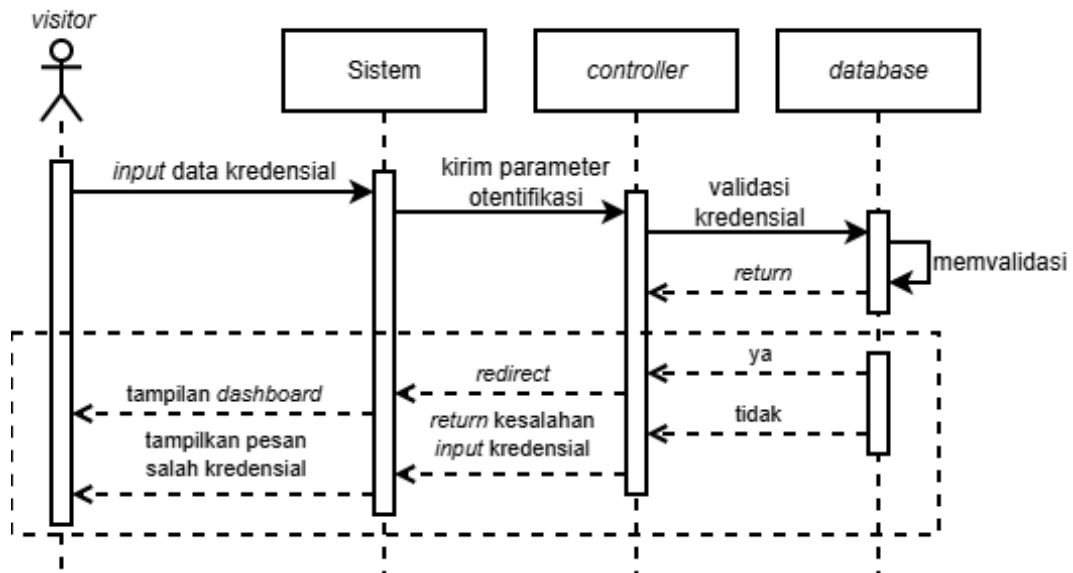
Gambar 5. 18 *Activity Diagram Login*
(Sumber: Data Pribadi)

Berdasarkan pemodelan visual yang disajikan pada Gambar 5. 18, diagram aktivitas tersebut menggambarkan aliran kerja (*workflow*) otentikasi pengguna terpusat yang melibatkan interaksi antara aktor *Visitor & Administrator* dengan *respons* Sistem di latar belakang *platform* SINEO. Alur data dikendalikan dari simpul awal (*initial state*) di sisi Sistem yang memproses penerimaan instruksi akses luar dan menayangkan modul halaman landas utama (*landing page*). Dari sisi pengguna, *Visitor & Administrator* menerima tampilan web SINEO dan menindaklanjutinya dengan menekan tombol navigasi operasional, seperti tombol "Masuk", "Mulai Monitoring", atau "Buka Dashboard Monitoring".

Aksi pemicuan tombol tersebut dikirimkan kembali ke sisi *back-end*, di mana Sistem menangkap *respons* permintaan untuk memproses penayangan jendela melayang berupa *Pop-up authModal* tanpa melakukan pengalihan halaman secara paksa (*no hard redirect*). Tampilan *Pop-up Modal Auth* tersebut sukses di-render di layar peramban pengguna, sehingga aktor dapat melakukan *input* kredensial berupa kombinasi alamat surel (*email*) dan kata sandi (*password*) yang terdaftar. Setelah data terisi, pengguna mengklik perintah aksi "Tekan *Login*".

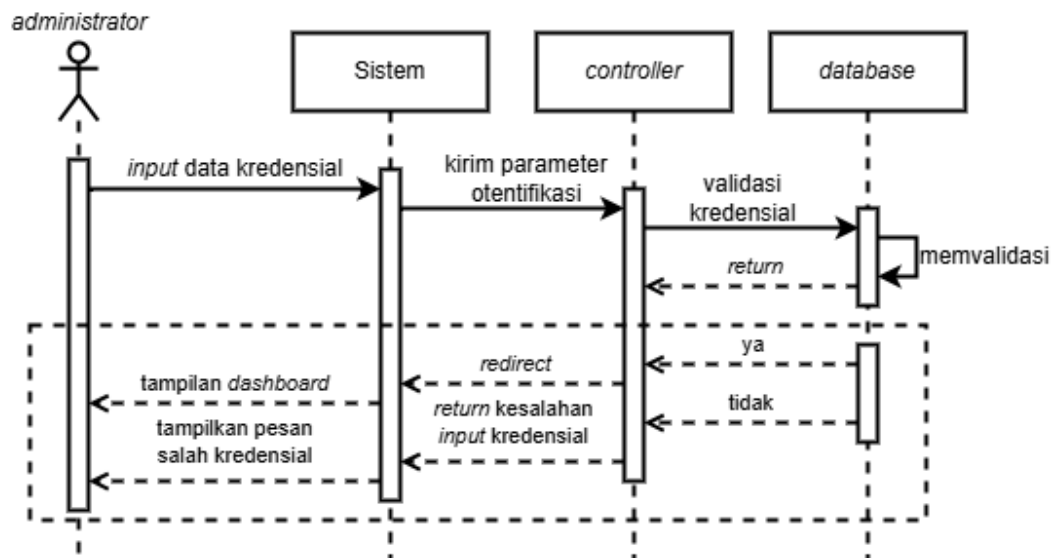
Aliran data kemudian berpindah secara instan ke sisi sistem untuk mengeksekusi proses memvalidasi data kredensial tersebut ke dalam basis data MySQL. Pada tahapan ini, alur logika melewati simpul keputusan (*decision node*) berbentuk belah ketupat untuk memeriksa keabsahan *records* akun. Apabila hasil keputusan validasi bernilai Tidak (kredensial salah), Sistem secara otomatis menolak hak akses dan mengembalikan alur ke fungsi pemrosesan *Pop-up authModal* awal seraya memunculkan pesan kesalahan validasi. Sebaliknya, jika hasil keputusan validasi bernilai Ya (kredensial cocok), Sistem dengan tanggap meloloskan hak otorisasi dan langsung mengarahkannya ke rute masing-masing tampilan *dashboard* yang sesuai dengan peran (*role*) pengguna. Aliran diagram aktivitas otentikasi ini dinyatakan selesai dan mencapai simpul penutup (*final state*) tepat setelah antarmuka tampilan *dashboard* utama SINEO siap diinteraksikan sepenuhnya oleh pengguna.

2) Sequence diagram login



Gambar 5. 19 Sequence Diagram Login (Visitor)
(Sumber: Data Pribadi)

Berdasarkan visualisasi aliran pesan yang dipaparkan pada Gambar 5. 19, skenario diagram urutan ini memetakan alur otentikasi hak akses umum ketika aktor *Visitor* masuk ke dalam pusat kendali *monitoring* menggunakan sistem keamanan bawaan. Alur kerja dimulai ketika aktor memasukkan data kredensial berupa alamat *email* dan kata sandi pada formulir digital yang disediakan oleh modal *login form*, yang kemudian mengirimkan parameter data tersebut via rute *HTTP POST Request* melalui fungsi *LoginRequest()* menuju objek *AuthController*. Di dalam server, objek *AuthController* mengolah masukan dengan mengirimkan kueri pencarian data *whereEmail()* kepada model entitas *User* untuk divalidasi ke dalam tabel basis data *users* pada MySQL. Setelah basis data mengonfirmasi kecocokan *records* dan mengembalikan token sesi aktif, *AuthController* menangkap status sukses tersebut lalu mengeksekusi fungsi pengalihan rute (*redirect*) untuk memuat halaman utama pada objek *dashboard.blade.php*.

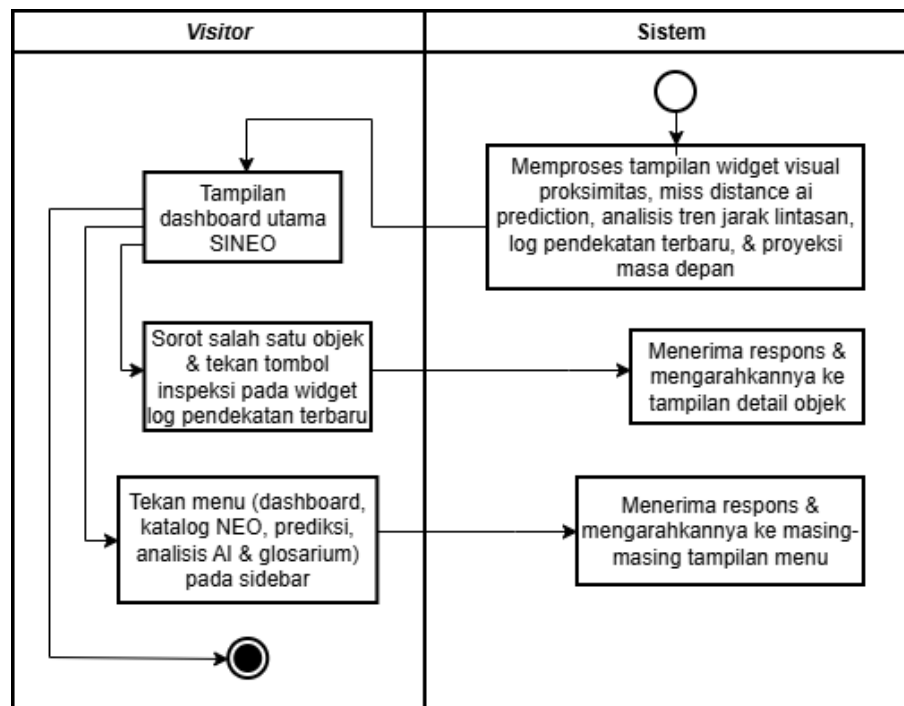


Gambar 5. 20 *Sequence Diagram Login (Administrator)*
(Sumber: Data Pribadi)

Berdasarkan visualisasi aliran pesan yang dipaparkan pada Gambar 5. 20, skenario diagram urutan ini memetakan alur otentikasi hak akses tingkat tinggi ketika aktor *Administrator* masuk ke dalam pusat kendali monitoring menggunakan sistem keamanan bawaan. Alur kerja dimulai ketika aktor memasukkan data kredensial berupa alamat email dan kata sandi pada formulir digital yang disediakan oleh modal *login form*, yang kemudian mengirimkan parameter data tersebut via rute *HTTP POST Request* melalui fungsi *LoginRequest()* menuju objek *AuthController*. Di dalam server, objek *AuthController* mengolah masukan dengan mengirimkan kueri pencarian data *whereEmail()* kepada model entitas *User* untuk divalidasi ke dalam tabel basis data *users* pada MySQL. Setelah basis data mengonfirmasi kecocokan *records* dan mengembalikan token sesi aktif, *AuthController* menangkap status sukses tersebut lalu mengeksekusi fungsi pengalihan rute (*redirect*) untuk memuat halaman utama pada objek *dashboard.blade.php*.

d. Melihat *dashboard*

1) *Activity diagram* melihat *dashboard*

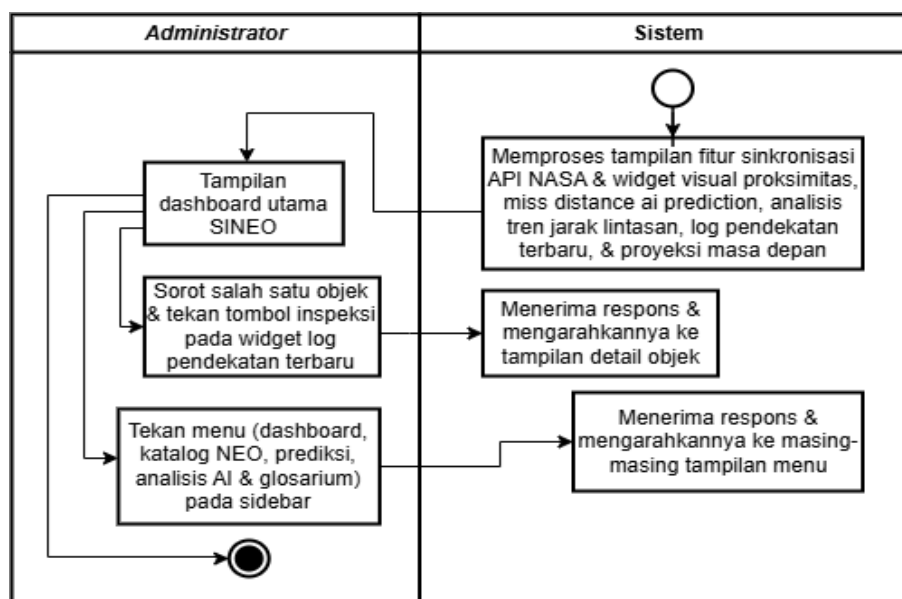


Gambar 5. 21 *Activity Diagram* Melihat *Dashboard Visitor*
(Sumber: Data Pribadi)

Berdasarkan alur kerja (*workflow*) yang dipetakan pada Gambar 5. 21, diagram aktivitas ini menggambarkan interaksi penjelajahan data astronomi oleh aktor *Visitor* di dalam halaman utama panel kendali SINEO. Aliran data diinisiasi oleh pihak sistem dari simpul awal (*initial state*), yang memproses penayangan seluruh komponen visual bawaan halaman utama, meliputi *widget* visual proksimitas, *miss distance AI prediction*, grafik analisis tren jarak lintasan, log pendekatan terbaru, serta blok proyeksi masa depan. Setelah seluruh komponen ter-render secara utuh, alur kerja bertransisi ke sisi aktor *Visitor* yang dihadapkan pada antarmuka tampilan *dashboard* Utama SINEO.

Setelah seluruh komponen antarmuka tersebut sukses di-render secara utuh, siklus bertransisi ke sisi aktor *Visitor* yang dihadapkan pada antarmuka "Tampilan *dashboard* utama SINEO". Dari halaman utama ini, *Visitor* diberikan dua opsi navigasi operasional reaktif yang dapat dipilih secara dinamis. Opsi pertama,

Visitor dapat melakukan penelusuran objek celestial secara individual dengan instruksi "Sorot salah satu objek & tekan tombol inspeksi pada *widget log* pendekatan terbaru". Aksi masukan ini dikirimkan menuju Sistem, yang akan langsung merespons dengan mengeksekusi perintah "Menerima *respons* & mengarahkannya ke tampilan detail objek". Dan opsi kedua, *Visitor* dapat melakukan perpindahan halaman menu dengan instruksi "Tekan menu (*dashboard*, katalog NEO, prediksi, analisis AI & glosarium AI) pada *sidebar*". Tindakan navigasi ini ditangkap oleh sistem untuk memicu perintah "Menerima *respons* & mengarahkannya ke masing-masing tampilan menu. Seluruh rangkaian aktivitas eksplorasi data oleh pengguna pada halaman utama ini dinyatakan selesai dan secara linear akan bermuara pada titik simpul penutup (*final state*) ketika *Visitor* telah selesai berinteraksi atau meninggalkan area halaman *monitoring* panel utama.

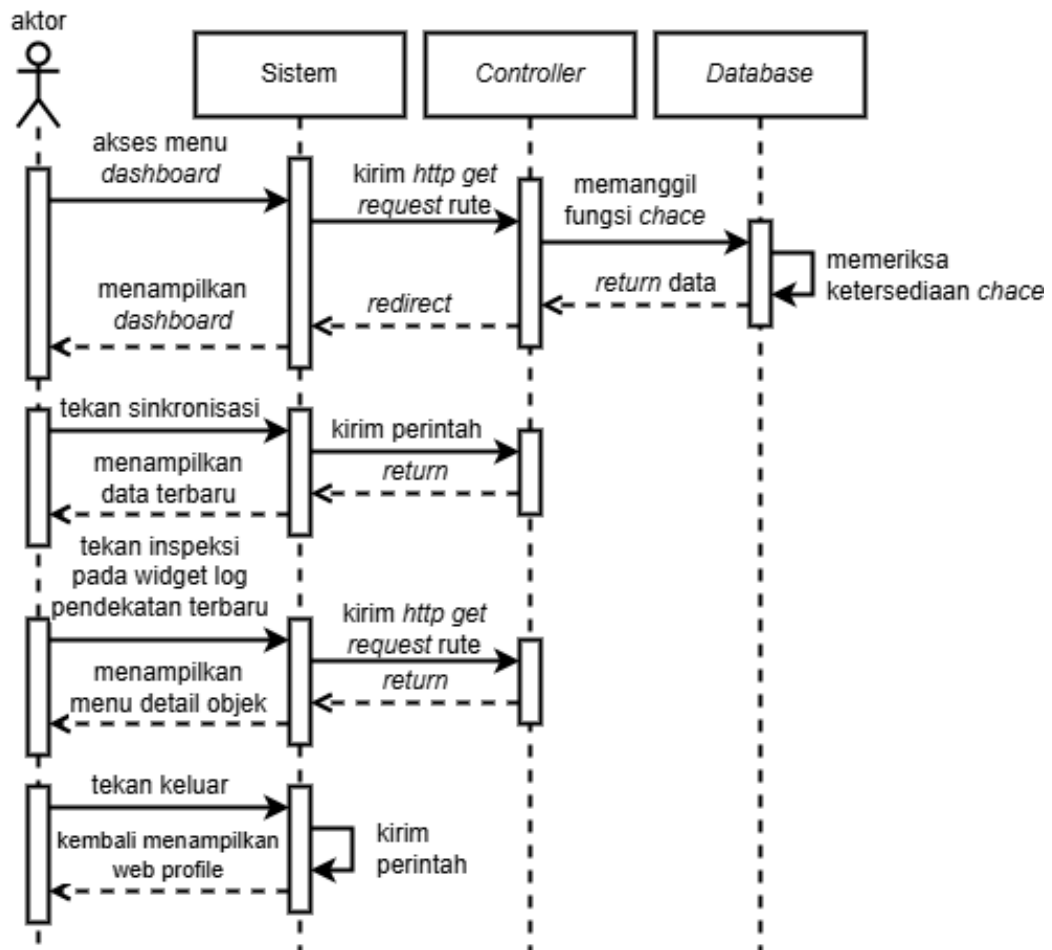


Gambar 5. 22 Activity Diagram Melihat Dashboard Administrator
(Sumber: Data Pribadi)

Berdasarkan pemodelan logika sistem yang disajikan pada Gambar 5. 22, diagram aktivitas tersebut memetakan hak akses operasional tingkat tinggi beserta alur kerja (*workflow*) yang dialami oleh aktor *Administrator* saat berinteraksi dengan panel kendali utama SINEO. Alur aktivitas dikendalikan dari simpul awal (*initial state*) di sisi sistem yang mengeksekusi instruksi pemuatan komponen fungsional kompleks. Secara spesifik, pihak *Sistem* merespons permintaan akses dengan

memproses tampilan fitur sinkronisasi API NASA serta mengompilasi jajaran *widget* analitik inti, yang meliputi visual proksimitas, *miss distance AI prediction*, analisis tren jarak lintasan, log pendekatan terbaru, dan proyeksi masa depan. Siklus data tersebut kemudian disalurkan ke sisi pengguna otoritas dalam bentuk penayangan tampilan *dashboard* Utama SINEO secara utuh.

Tepat setelah halaman utama termuat, alur kerja bercabang menjadi beberapa opsi tindakan kontrol yang dapat dieksekusi secara dinamis oleh *Administrator*. Opsi pertama, *Administrator* dapat memantau indikator operasional secara langsung pada bagian komponen UI *widget sync-nasa* & statusnya. Opsi kedua, *Administrator* dapat melakukan penelusuran spasial mendalam dengan menyorot salah satu objek asteroid serta menekan tombol inspeksi yang tertera pada *widget* log pendekatan terbaru. Aksi ini dikirim sebagai perintah masukan ke sisi *back-end*, di mana sistem akan menerima *respons* tersebut dan secara instan mengarahkan antarmuka menuju tampilan detail objek untuk kebutuhan inspeksi parameter fisis objek secara individual. Opsi ketiga, *Administrator* dapat memicu perpindahan navigasi halaman dengan menekan tombol menu fungsional (*Dashboard*, Katalog NEO, Prediksi, Analisis AI, atau Glosarium AI) pada area *sidebar*. Tindakan ini akan direspon oleh sistem dengan menerima *respons* perintah dan segera mengalihkan peramban menuju masing-masing tampilan menu anak yang dituju. Aliran keseluruhan diagram aktivitas khusus *administrator* ini dinyatakan selesai dan menuju simpul akhir (*final state*) ketika seluruh fungsi pemantauan data ataupun pemeliharaan basis data telah selesai dilaksanakan.

2) *Sequence diagram melihat dashboard*

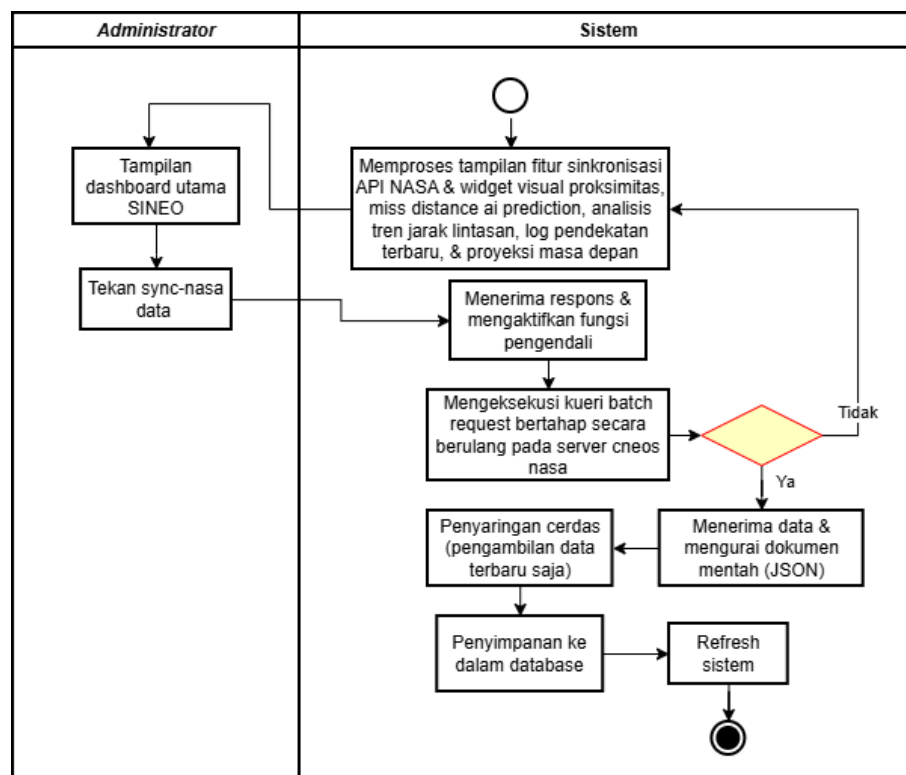
Gambar 5. 23 *Sequence Diagram Melihat Dashboard*
(Sumber: Data Pribadi)

Berdasarkan pemodelan rancangan interaksi terpadu pada Gambar 5. 23, skenario diagram urutan ini memetakan jalannya data operasional reaktif saat aktor membuka panel kendali sentral setelah proses inisiasi halaman berhasil dilakukan. Aktivitas diawali saat pengguna mengakses menu *dashboard* operasional yang memicu komponen tampilan *dashboard.blade.php* untuk mengirimkan pesan instruksi berupa HTTP *GET Request* menuju fungsi pengontrol *index()* di dalam objek *NeoController*. Menindaklanjuti permintaan tersebut, *NeoController* secara aktif mengirimkan parameter kueri *Cache::get('dashboard_stats')* kepada entitas kelas model *Cache* guna mendeteksi ketersediaan rekaman data historis astronomi di memori jangka pendek. Objek *Cache* kemudian memvalidasi data dan

mengembalikan larik ringkasan kalkulasi objek dekat bumi kepada pengontrol, di mana objek *NeoController* langsung meneruskan pasokan data aktual tersebut ke berkas tampilan *dashboard.blade.php* untuk dirender menjadi tombol *sync-data* (*Admin*) grafik analisis tren lintasan jarak dan *Chart.js* lainnya seperti visual proksimitas, *miss distance ai prediction*, log pendekatan terbaru, dan proyeksi masa depan.

e. Sinkronisasi data NASA API

1) *Activity diagram* sinkronisasi data NASA API



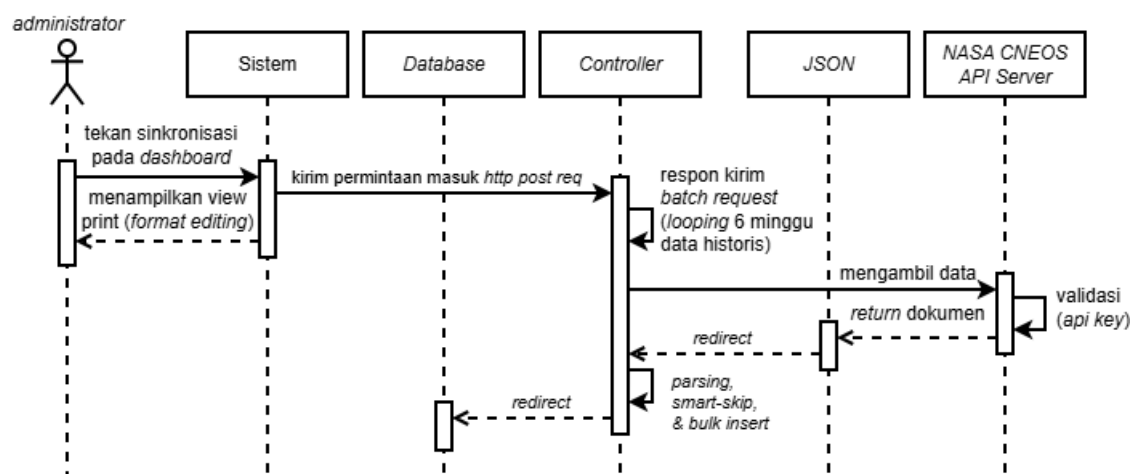
Gambar 5. 24 *Activity Diagram* Sinkronisasi Data NASA API
(Sumber: Data Pribadi)

Berdasarkan rancangan logis yang dipaparkan pada Gambar 5. 24, diagram aktivitas pada fitur Sinkronisasi Data NASA API memetakan urutan kerja (*workflow*) otomatisasi data yang dikendalikan secara khusus oleh aktor *Administrator*. Aliran aktivitas diinisiasi dari simpul awal (*initial state*) di sisi sistem yang menampilkan antarmuka utama pusat kendali. Dari sisi pengguna otoritas, *Administrator* menindaklanjuti dengan mengeklik komponen tombol

fungsional "Sync NASA Data" pada *widget dashboard* web. Aksi fisik ini dikirim sebagai permintaan masuk (*HTTP Request*) ke sisi *back-end* sistem. Pihak Sistem merespons instruksi tersebut dengan mengaktifkan fungsi pengendali *syncNasaData()* yang secara instan menetapkan batas waktu eksekusi server menjadi tak terbatas melalui perintah *set_time_limit(0)*.

Selanjutnya, sistem mengeksekusi *fasad* HTTP Laravel untuk mengirimkan kueri *batch requests* bertahap secara berulang (*looping* kalender mundur 6 minggu ke belakang) menuju server eksternal CNEOS NASA tanpa melalui verifikasi SSL. Setelah server NASA meloloskan permintaan, sistem bertugas menerima serta mengurai (*parsing*) dokumen mentah berformat JSON tersebut. Alur data kemudian melewati fungsi penyaringan cerdas (*smart skip*), di mana sistem akan mengidentifikasi dan mengabaikan *record* objek astronomi lama yang telah tersimpan untuk menghindari redundansi data. Tepat setelah proses penyaringan, sistem melakukan penyimpanan massal (*bulk insert*) secara legal menuju basis data relasional MySQL melalui model *Eloquent*. Aliran aktivitas manajemen siklus *database* ini resmi berakhir dan mencapai simpul penutup (*final state*) tepat setelah sistem me-*refresh* halaman web keseluruhan.

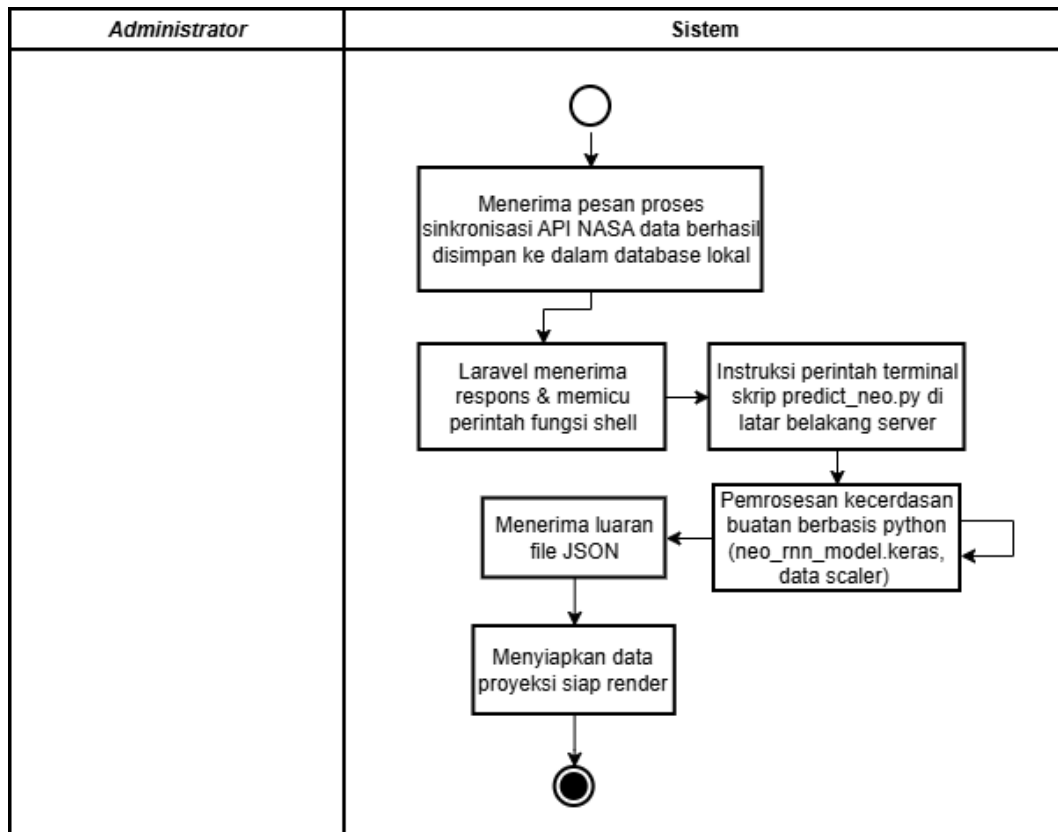
2) Sequence diagram sinkronisasi data NASA API



Gambar 5. 25 Sequence Diagram Sinkronisasi Data NASA AI
(Sumber: Data Pribadi)

Berdasarkan pemodelan operasional tingkat tinggi pada Gambar 5. 25, diagram urutan menjabarkan otomatisasi penarikan massal elemen orbital asteroid eksternal yang dipicu secara eksklusif oleh aktor *Administrator*. Alur pesan diawali dari klik fisik tombol "*Sync NASA Data*" pada objek tampilan *dashboard.blade.php* oleh *administrator*, yang mengirimkan permintaan masuk HTTP *POST Request* melalui fungsi *syncNasaData()* menuju objek pengontrol *NeoController*. Objek *NeoController* merespons pesan dengan mengonfigurasi batas waktu eksekusi server menjadi tak terbatas (*set_time_limit (0)*) dan memerintahkan komponen HTTP *Client* internal untuk mengeksekusi kueri permintaan data bertahap (*batch requests* via *looping* kalender mundur 6 minggu) menuju objek eksternal NASA CNEOS API Server. Setelah token divalidasi, NASA CNEOS API Server mengirimkan dokumen mentah berformat JSON data kembali ke kontroler, di mana *NeoController* melakukan pemecahan *array* (*parsing*) dan memanggil metode *NeoObject::bulkInsert()* untuk menyimpan rekaman data baru secara massal ke dalam *database* MySQL pada tabel *neo_objects*.

f. Memicu eksekusi inferensi AI

1) *Activity diagram* memicu eksekusi inferensi AI

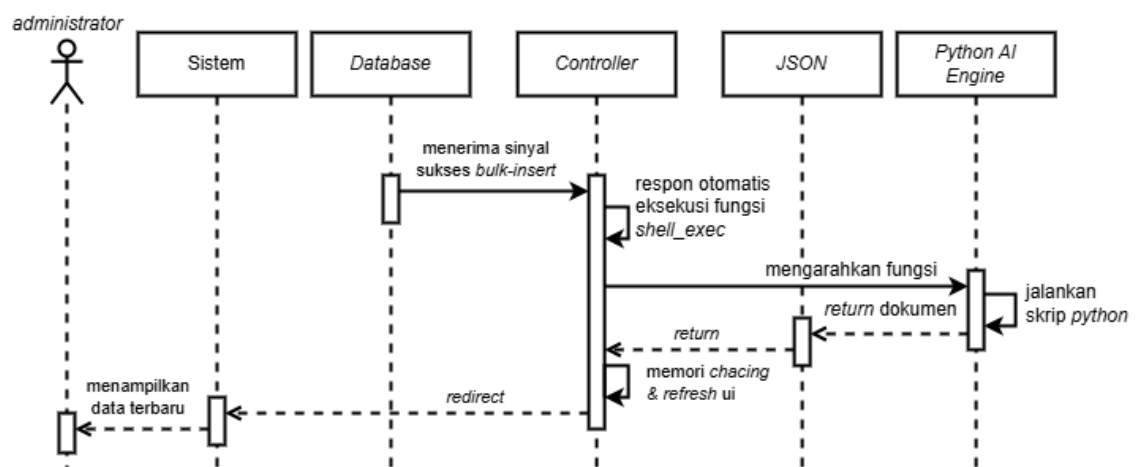
Gambar 5. 26 *Activity diagram* sinkronisasi data nasa api
(Sumber: Data Pribadi)

Berdasarkan pemodelan arsitektur sistem informasi *hybrid environment* yang disajikan pada Gambar 5. 26, diagram aktivitas pada fitur Memicu Eksekusi Inferensi AI menggambarkan jembatan interkoneksi latar belakang antara lingkungan web PHP Laravel 12 dengan lingkungan komputasi *Python*. Aliran aktivitas dikendalikan dari simpul awal (*initial state*) di bawah otoritas sistem yang dipicu secara otomatis (*include relation*) sesaat setelah proses sinkronisasi massal data mentah API NASA dinyatakan berhasil bermigrasi ke *database* lokal. Pihak sistem pada arsitektur *back-end* Laravel merespons peristiwa tersebut dengan memicu perintah fungsi internal *shell_exec()* atau *runAiInference()*. Melalui fungsi eksekusi *shell* ini, sistem mengirimkan instruksi perintah terminal mandiri untuk

menjalankan berkas skrip aktif mesin inferensi *Python* bernama *predict_neo.py* di latar belakang server.

Siklus pemrosesan kemudian berpindah secara instan menuju modul kecerdasan buatan berbasis *Python*, di mana skrip cerdas tersebut bekerja dengan memuat berkas model biner terkompilasi *neo_rnn_model.keras* serta *file* penormalan data *scaler*. Mesin AI menarik data runtun waktu (*rolling window*) sepanjang total 30 hari data objek teratas dari *database* MySQL sebagai basis memori historis awal. Memanfaatkan algoritma *Recursive Multi-step Forecasting*, skrip *Python* mengeksekusi komputasi peramalan berantai untuk memproyeksikan fluktuasi nilai *miss distance* sepekan (7 hari) ke depan. Hasil akhir kalkulasi regresi kontinu tersebut diekspor oleh skrip menjadi dokumen biner baru berformat JSON. Alur kerja kembali ke sisi sistem informasi web, di mana sistem menerima luaran berkas JSON tersebut, memperbarui memori *caching* lokal via *Cache::put()*, dan menyiapkan data proyeksi untuk siap dirender menjadi grafik garis interaktif pada antarmuka pengguna. Seluruh rangkaian aktivitas komputasi AI latar belakang ini dinyatakan selesai seutuhnya saat alur menyentuh simpul penutup (*final state*).

2) *Sequence diagram* memicu eksekusi inferensi AI



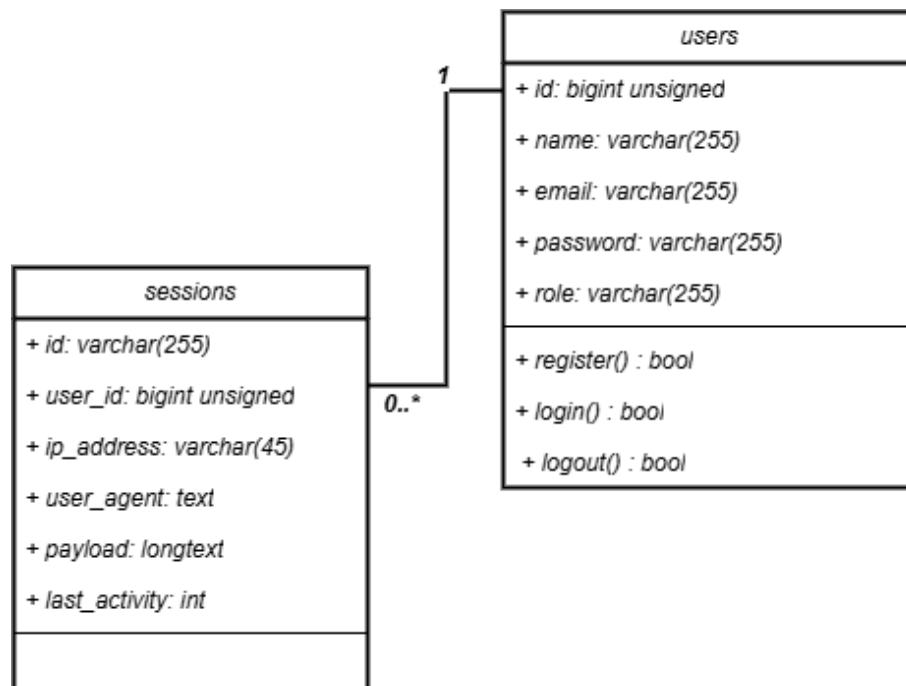
Gambar 5. 27 *Sequence Diagram* Memicu Eksekusi Inferensi AI
(Sumber: Data Pribadi)

Berdasarkan skema arsitektur koneksi antar-lingkungan server pada Gambar 5. 27, diagram urutan memetakan jembatan interkoneksi latar belakang otomatis antara

kerangka kerja Laravel web dengan lingkungan komputasi *Python*. Aliran pesan dikendalikan dari simpul awal di bawah otoritas sistem informasi, di mana tepat setelah objek pengontrol *NeoController* menerima sinyal sukses transaksi data massal dari basis data MySQL pada tabel *neo_objects*, kontroler secara otomatis memicu fungsi perintah *shell* latar belakang server berupa *shell_exec('python predict_neo.py')* mengarah ke objek eksternal *Python AI Engine*. Objek *Python AI Engine* aktif mengeksekusi skrip *predict_neo.py* untuk memuat model biner terkompilasi *neo_rnn_model.keras* serta melakukan kalkulasi peramalan berantai *Recursive Multi-step Forecasting* berbasis data historis 30 hari terakhir. Hasil kalkulasi regresi tersebut diekspor oleh skrip menjadi berkas biner baru *prediction_results.json* dan dikirimkan kembali ke *NeoController* untuk memperbarui memori lokal melalui perintah *Cache::put()*, serta memicu pemuatan ulang dokumen (*refresh layout*) di sisi antarmuka *dashboard.blade.php*.

B. Class diagram

Di samping melakukan dekonstruksi aliran aktivitas melalui diagram aktivitas, fase *Sprint* pada kerangka kerja *Scrum* ini juga menetapkan struktur statis sistem melalui penyusunan *Class Diagram*. Pemodelan struktural ini bertujuan untuk memetakan secara eksplisit seluruh kelas objek komponen model dan representasi data yang beroperasi di dalam arsitektur sistem *monitoring* SINEO, lengkap dengan spesifikasi visibilitas atribut murni (berbasis tipe data *riil* pada berkas kueri SQL), metode fungsi operasional (*methods*), serta derajat hubungan keterikatan (*multiplicity*) antar-entitas di dalam lingkungan *framework* Laravel 12.



Gambar 5. 28 *Class Diagram* Autentikasi SINEO
(Sumber: Data Pribadi)

Berdasarkan manifestasi struktur diagram kelas pada Gambar 5. 28, diagram kelas ini memetakan rancangan pemrograman berorientasi objek yang mengendalikan siklus keamanan dan manajemen otentikasi akun pengguna pada *platform* SINEO. Arsitektur diagram ini menempatkan kelas pengontrol utama, yaitu *AuthController*, sebagai komponen pusat yang memegang visibilitas metode publik untuk menangani alur interaksi logika pengguna. Di dalam kelas *AuthController*, didefinisikan rentetan metode esensial yang meliputi fungsi *openAuth()* dan *register()* untuk penanganan pendaftaran akun, fungsi *openAuth()* dan *login()* untuk pengenalan hak akses pengguna, serta metode fungsi *logout()* untuk terminasi sesi pangkalan data aktif pengguna, di mana kelas pengontrol ini mewarisi (*inheritance relationship*) sifat dasar dari kelas inti bawaan *framework* Laravel yaitu kelas *Controller*.

Guna menjamin validitas muatan data yang masuk sebelum diproses oleh pengontrol, kelas *AuthController* dikonfigurasi memiliki hubungan ketergantungan (*dependency relation*) yang kuat terhadap dua kelas penunjang *form* yaitu kelas *RegisterRequest* dan kelas *LoginRequest*. Kedua kelas *request* tersebut

ditugaskan secara modular untuk membungkus aturan validasi parameter masukan dari sisi antarmuka, di mana kelas *RegisterRequest* memuat metode publik *rules()* dan *authorize()* untuk menyaring string isian data dari pengguna, yang keduanya secara terstruktur diturunkan langsung dari kelas dasar cetak biru *FormRequest* milik arsitektur bawaan Laravel.

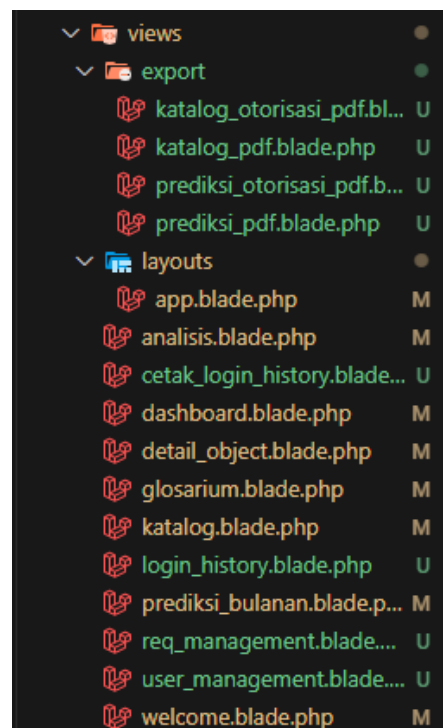
Poros diagram kelas ini ditutup secara terintegrasi oleh kehadiran kelas data utama, yaitu kelas model *User*. Kelas model *User* bertindak sebagai representasi cetak biru objek entitas fisik tabel *users* di dalam *database* MySQL, yang memuat enkapsulasi properti data sensitif pengguna seperti atribut pengenalan unik *id*, *name*, *email*, properti kata sandi tersandi *password*, properti penentu peran *role*, serta penanda waktu *created_at* dan *updated_at*. Hubungan relasi antara kelas pengontrol *AuthController* dengan kelas model *User* dimodelkan sebagai hubungan asosiasi terarah (*directed association*), yang menegaskan bahwa kelas pengontrol memegang kendali penuh untuk memicu operasi pemanggilan data, pembuatan *record* baru instansiasi *user* baru pada *database phpMyAdmin*, serta memverifikasi kesesuaian sandi akun pengguna melalui jembatan enkripsi bawaan sebelum memberikan izin akses menu halaman utama *platform* SINEO.

5.1.5.2.3. Implementation

Sistem antarmuka ini bertindak sebagai media interaksi utama (*front-end*) yang menyajikan hasil pemrosesan data historis dari *database* MySQL serta hasil inferensi prediksi dari model biner *Dual-Stacked* RNN (*neo_rnn_model.keras*) secara dinamis kepada pengguna. Secara fungsional, struktur pengembangan komponen antarmuka *monitoring* SINEO dirancang secara modular dan dibagi menjadi beberapa halaman utama yang didokumentasikan sebagai berikut:

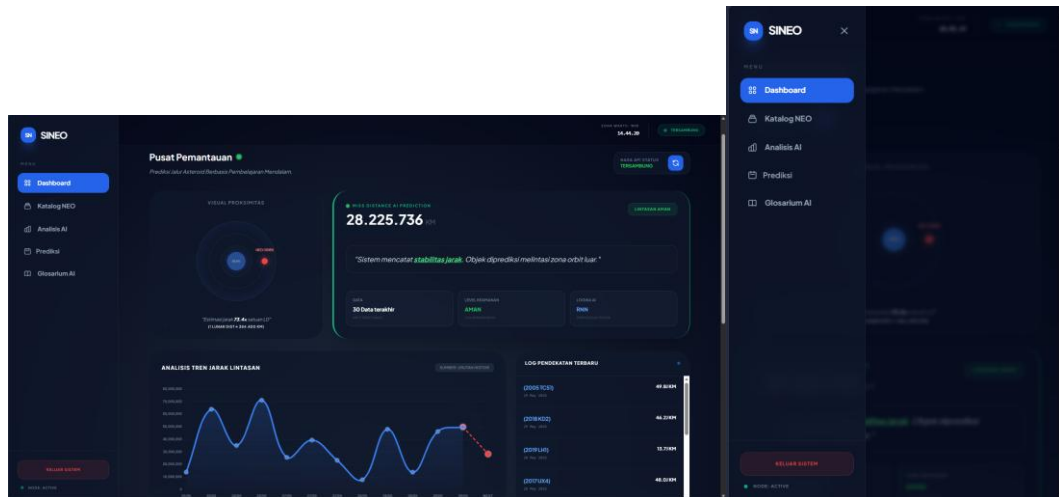
a. Struktur berkas komponen antarmuka pada sistem

Sebelum menjabarkan bentuk fisik dari masing-masing halaman, penataan letak berkas *views* (UI) di dalam lingkungan *editor Visual Studio Code* disajikan terlebih dahulu pada Gambar 5. 29. Pemisahan berkas ini ditujukan untuk mempermudah manajemen kode (*code maintainability*) dan memastikan pemisahan peran (*separation of concerns*) yang jelas antara semua halaman menu, dan komponen tata letak (*layouting*).



Gambar 5. 29 Struktur Berkas Komponen Views Antarmuka Web SINEO pada VS Code
(Sumber: Data Pribadi)

b. Komponen tata letak utama



Gambar 5. 30 Tampilan *Sidebar Navigation* pada *Desktop & Mobile*
(Sumber: Data Pribadi)

Berkas *app.blade.php* bertindak sebagai poros utama atau kerangka induk (*master layout*) dari sistem informasi SINEO. Berkas ini mengonfigurasi struktur dasar HTML5, pemanggilan *pustaka global* seperti *Tailwind CSS*, *Alpine.js*, dan *Chart.js*, serta menerapkan *font Plus Jakarta Sans* untuk menjamin konsistensi estetika visual naskah web.

Secara mekanis, skrip ini mengimplementasikan tata letak kolom ganda reaktif (*responsive sidebar-layout*) menggunakan utilitas *Flexbox Tailwind CSS*. Pada sisi kiri, terdapat komponen navigasi samping (*sidebar navigation*) permanen untuk *desktop* dan reaktif modal untuk perangkat seluler melalui kontrol direktif *x-data="{ mobileMenuOpen: false }"*. Sementara itu, komponen *header* bagian atas dilengkapi dengan indikator status koneksi API waktu nyata (*ping animation effect*) dan modul jam digital dinamis (*liveClock*) berbasis manipulasi DOM *JavaScript*. Seluruh halaman fungsional lain akan diinjeksikan secara otomatis ke dalam tubuh tata letak induk ini melalui instruksi direktif *@yield('content')*.

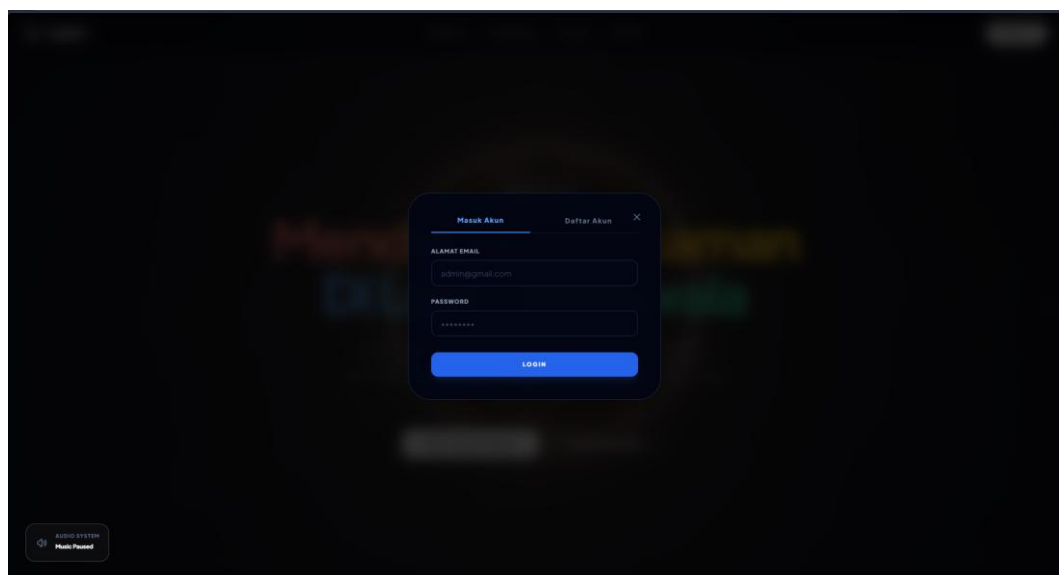
Selain mengonfigurasi kontrol navigasi seluler, berkas layout induk ini juga dimodifikasi pada level komponen *<body>* dengan menyuntikkan *global state Alpine.js* berupa *x-data="{ openAuth: ..., tabAuth: ..., }"*. Inisialisasi variabel global

ini dikonfigurasi secara hibrida agar dapat mendeteksi variabel kembalian *session error* dari Laravel (`$errors->any()`). Mekanisme ini menjamin bahwa seluruh komponen anak yang diinjeksi melalui direktif `@yield('content')` dapat berbagi sinyal data (*state sharing*) secara *real-time*.

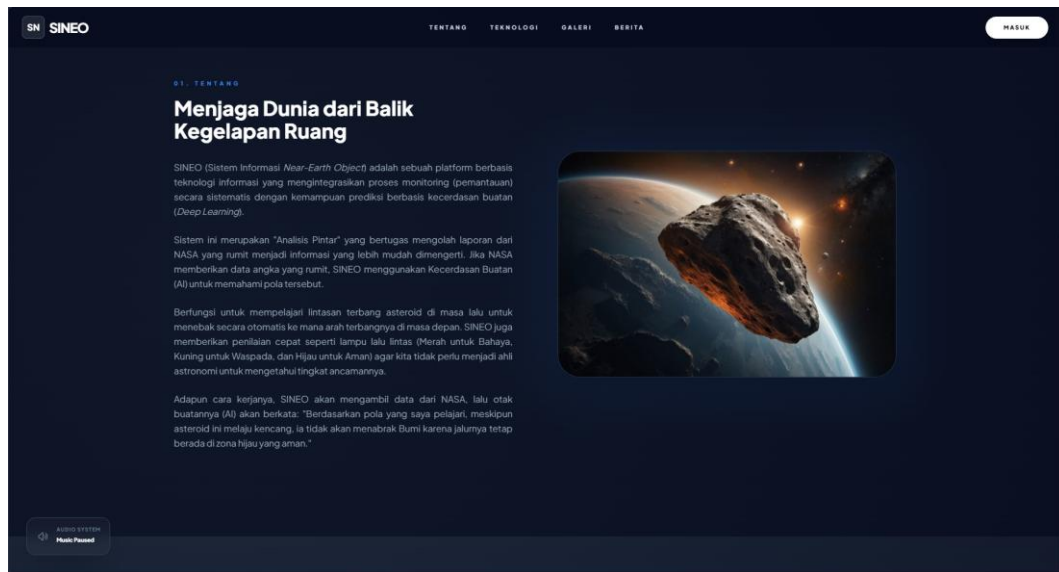
c. Halaman profil web / *landing page*



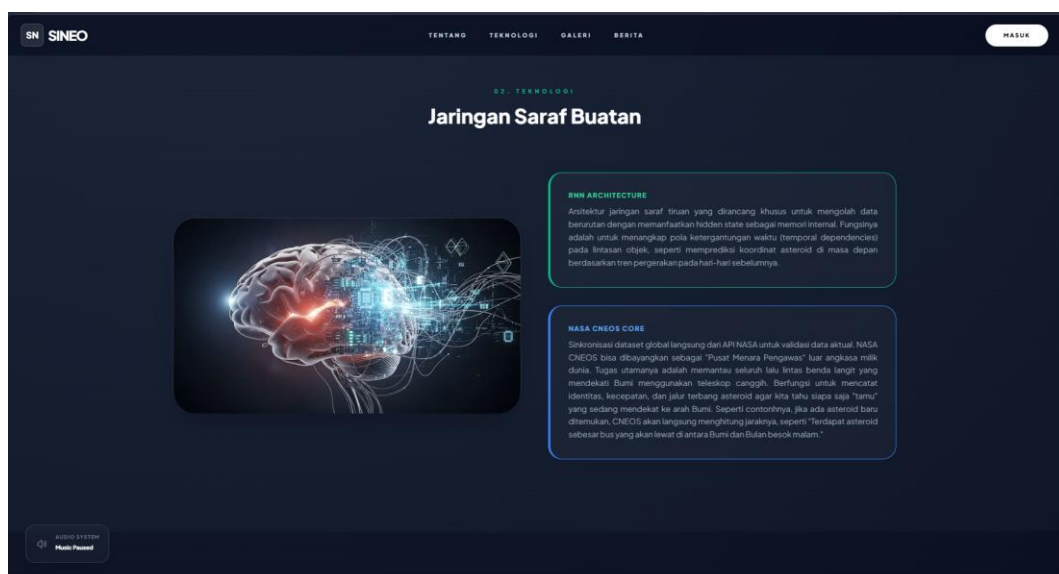
Gambar 5. 31 Tampilan *Landing Page* (Web Profile)
(Sumber: Data Pribadi)



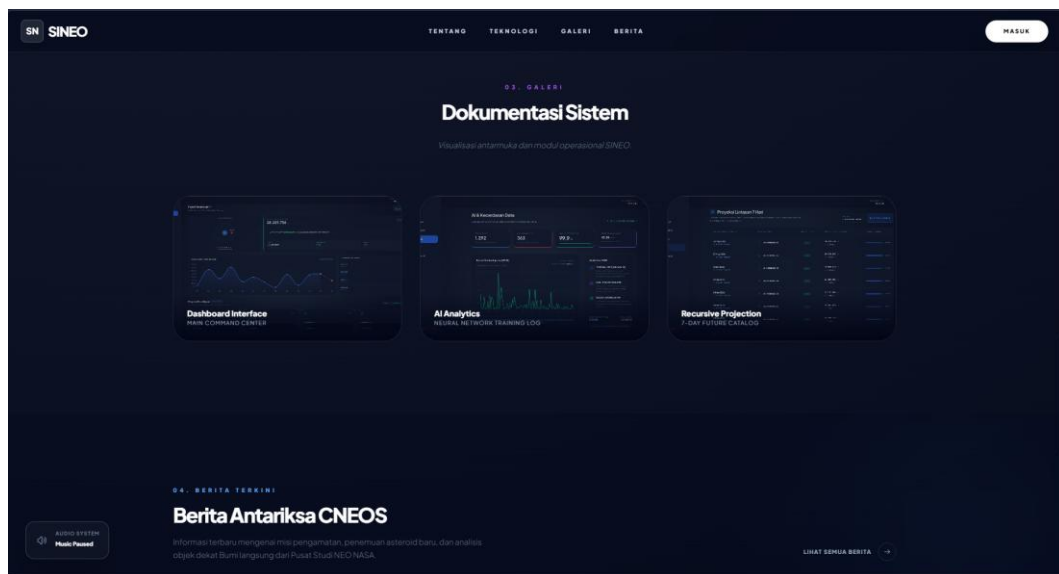
Gambar 5. 32 Tampilan *Landing Page* (Autentikasi)
(Sumber: Data Pribadi)



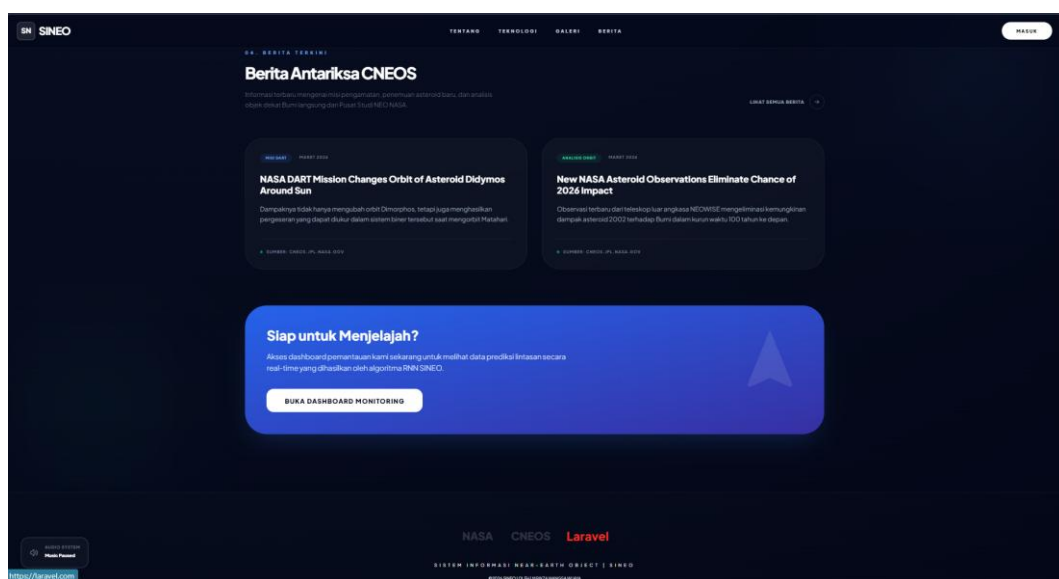
Gambar 5. 33 Tampilan *Landing Page* (Tentang SINEO)
(Sumber: Data Pribadi)



Gambar 5. 34 Tampilan *Landing Page* (Teknologi SINEO)
(Sumber: Data Pribadi)



Gambar 5. 35 Tampilan *Landing Page* (Dokumentasi SINEO)
(Sumber: Data Pribadi)



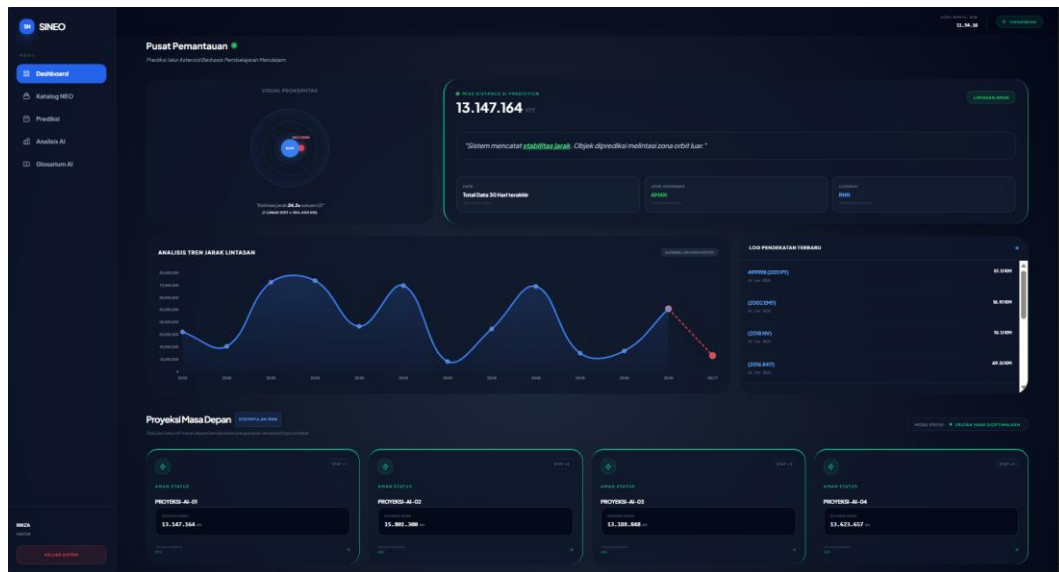
Gambar 5. 36 Tampilan *Landing Page* (Berita SINEO)
(Sumber: Data Pribadi)

Berkas *welcome.blade.php* bertindak sebagai *web profile* atau gerbang beranda utama (*landing page*) dari sistem informasi SINEO sebelum pengguna masuk ke dalam panel kendali operasional (*command center*). Secara estetika, halaman ini dikonfigurasi menggunakan latar belakang bertema ruang angkasa (*space background radial gradient*) yang dikombinasikan dengan efek animasi transisi *Animate On Scroll* (AOS) untuk memberikan pengalaman interaktif.

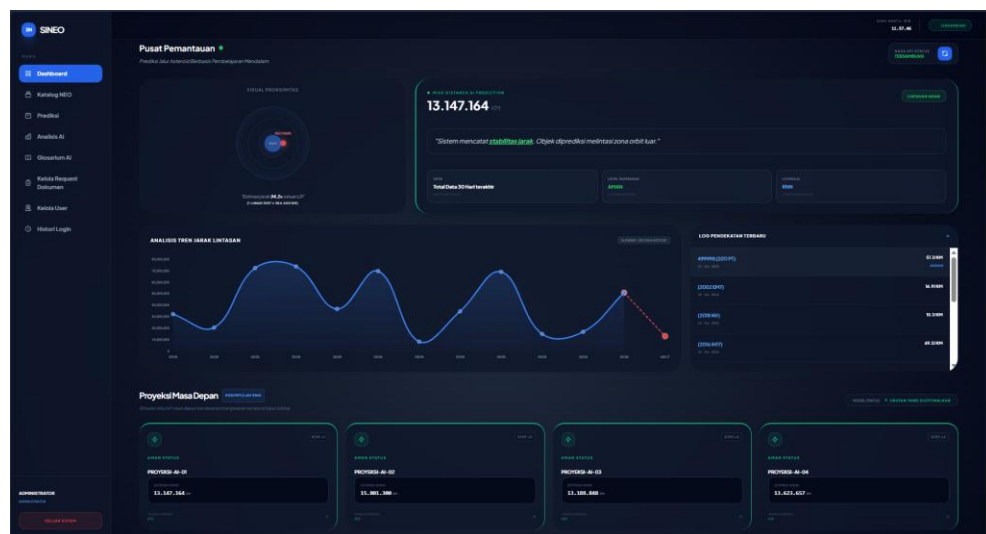
Fungsi krusial dari berkas ini adalah memetakan navigasi utama melalui tombol jembatan interaktif (*Call to Action*) yang terhubung langsung dengan rute *dashboard* sistem informasi. Selain itu, sebagai fitur sekunder, di dalam berkas ini ditanamkan kontrol pemutar audio latar belakang (*ambient background music*) reaktif menggunakan *Alpine.js* (*x-data="{isPlaying: false, ...}"*) yang memanipulasi elemen audio HTML5 secara instan tanpa mengganggu performa pemuatan dokumen awal.

Di samping berfungsi sebagai media edukasi publik, berkas *welcome.blade.php* kini bertindak sebagai *Gatekeeper* Autentikasi Utama bagi *users* melalui teknik *Glassmorphism Pop-up AuthModal*. Ketika tombol 'masuk' pada bilah navigasi diklik, sistem tidak melakukan dialihan halaman (*hard redirect*), melainkan mengubah status variabel reaktif *Modal* menjadi *true*. Jendela melayang akan muncul di atas lapisan video latar belakang dengan tingkat visibilitas dan dilengkapi fitur aksesibilitas berupa penutupan modal otomatis lewat klik area luar (*@click.away*) atau penekanan tombol *Escape* (*@keydown.escape.window*). Formulir di dalam modal ini diamankan secara mutlak menggunakan direktif *@csrf* untuk memproses pengiriman data kredensial via metode *POST* menuju *AuthController*.

d. Halaman *dashboard* utama sistem



Gambar 5. 37 Antarmuka Halaman *Dashboard* Utama Sistem Monitoring SINEO (Visitor)
(Sumber: Data Pribadi)



Gambar 5. 38 Antarmuka Halaman *Dashboard* Utama Sistem Monitoring SINEO (Admin)
(Sumber: Data Pribadi)

Halaman *dashboard.blade.php* merupakan komponen antarmuka sentral yang menyajikan ringkasan informasi eksekutif (*executive summary*) mengenai lalu lintas objek dekat Bumi secara *real-time*. Untuk *Administrator* sendiri, setelah

melakukan *login*, *dashboard* akan menampilkan tombol *Sync-Nasa* (sinkronisasi data API NASA. Sedangkan jika *Visitor*, tombol tersebut akan disembunyikan. Di dalam halaman ini, diimplementasikan beberapa metrik *widget card* berlatar belakang efek kaca (*glassmorphism*), antara lain visual proksimitas NEO terhadap bumi, prediksi AI *miss distance* (*window data*, *status level*, & logika AI).

Pada bagian bawahnya, diimplementasikan sebuah grafik analisis tren jarak lintasan (*line chart*) memanfaatkan objek *Chart.js* yang memetakan fluktuasi parameter jarak lintasan terdekat (*miss distance*) objek asteroid terhadap inti Bumi. Grafik ini diatur secara responsif menggunakan parameter *maintainAspectRatio: false* dan dilengkapi fungsi interaksi *tooltip* kustom untuk memunculkan label nilai angka ribuan desimal yang rapi (*toLocaleString('id-ID')*) saat kursor menyentuh titik koordinat tertentu. Kemudian pada sisi kanannya terdapat *widget* 10 log pendekatan terbaru yang ditampilkan dengan minimalis. Dan pada bagian bawah *widget* analitik, terdapat *widget* prediksi selama sepekan yang ditampilkan secara minimalis.

Komponen navigasi *sidebar* pada halaman *dashboard* juga memuat tombol aksi 'Keluar Sistem'. Demi memenuhi standar keamanan *framework* Laravel, tombol ini diintegrasikan dengan *JavaScript* *onclick="event.preventDefault(); document.getElementById('logout-form').submit();"* untuk memicu formulir tersembunyi ber-metode *POST*. Langkah arsitektur ini diterapkan untuk mencegah celah keamanan dan memastikan bahwa tabel *sessions* pada *database* MySQL langsung menghancurkan data otentikasi aktor secara bersih saat sesi diakhiri.

5.1.5.2.4. Review

Tahapan *Sprint Review* pada akhir *sprint* kedua ini ditujukan untuk mengevaluasi fungsionalitas dan interaksi antarmuka dari modul *Web Profile*, *Dashboard* Utama, Sistem Autentikasi, hingga fungsi Sinkronisasi Data NASA API menggunakan metode *Black Box Testing*. Pengujian *Black Box* dilakukan untuk memvalidasi perilaku fungsional komponen antarmuka reaktif dan pemrosesan data di *back-end* berdasarkan skenario masukan dan keluaran (*input-output*) dari sudut pandang

pengguna, tanpa menguji struktur kode internal *framework*. Proses peninjauan dilakukan secara hibrida melalui mekanisme *Client Validation Loop* secara terbuka melalui pengujian fungsionalitas (*Black Box Testing*) langsung oleh pengguna umum untuk mengevaluasi ketercapaian fungsi interaksi antarmuka dan otomatisasi sistem. Fokus utama evaluasi tertuju pada kepresisian renderan komponen visual *Web Profile*, keabsahan proses otentikasi hak akses akun yang dijembatani oleh *Glassmorphism Pop-up AuthModal* berbasis *Alpine.js*, efisiensi eksekusi *batch request* penarikan data NASA CNEOS API dengan penyaringan *smart-skip*, hingga keandalan otomatisasi pemicuan skrip inferensi *Python predict_neo.py* via fungsi *shell_exec()*.

Guna membuktikan ketercapaian seluruh fungsionalitas antarmuka, sistem pengamanan rute, dan otomatisasi sinkronisasi data pada *Sprint 2* secara empiris, dilakukan pengujian masukan dan keluaran (*input-output*) menggunakan metode *Black Box Testing*. Rangkuman hasil pengujian *Black Box* untuk seluruh fitur pada *Sprint 2* disajikan secara terperinci pada Tabel 5. 12 di bawah ini:

Tabel 5. 12 Hasil Pengujian *Black Box Testing Sprint 2*

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
1.	Akses rute utama beranda tanpa otentikasi sesi <i>login</i>	Mengakses URL melalui server lokal pada browser.	Sistem berhasil memuat <i>landing page</i> profil web beserta pemutar <i>ambient music</i> .	Halaman <i>welcome</i> ter- <i>render</i> responsif dan audio aktif.	Sesuai / Valid	21 April 2026
2.	Gerbang autentikasi melayang (<i>welcome.blade.php</i>)	Mengklik tombol "MASUK" pada bilah navigasi beranda utama.	Variabel reaktif <i>loginModal</i> berubah menjadi <i>true</i> . Sistem memunculkan jendela <i>pop-up</i> autentikasi melayang (<i>glassmorphism box</i>) di atas lapisan video tanpa memuat ulang halaman (<i>no hard redirect</i>).	Tombol reaktif merespons seketika, mengaktifkan <i>glassmorphism pop-up modal</i> berbasis <i>Alpine.js</i> muncul melayang tanpa <i>hard redirect</i> .	Sesuai / Valid	21 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
3.	Eksekusi pendaftaran akun baru dengan kredensial sah	Mengisi kredensial sah, dan menekan daftar.	Sistem memvalidasi data, melakukan <i>insert data</i> ke MySQL, dan memunculkan notifikasi sukses.	Data tersimpan aman di tabel <i>users phpMyAdmin</i> dan memunculkan pesan Registrasi berhasil! Silakan <i>login</i> menggunakan akun Anda.	Sesuai / Valid	21 April 2026
4.	Eksekusi pendaftaran dengan <i>email</i> yang sudah terdaftar	Mengisi formulir menggunakan <i>email</i> yang telah ada di <i>database</i> .	Sistem mendeteksi duplikasi via <i>FormRequest</i> , menolak pendaftaran, dan menampilkan notifikasi galat.	Jendela modal tetap terbuka seraya memunculkan pesan <i>Email</i> ini sudah terdaftar. Silakan gunakan <i>email</i> aktif lain.	Sesuai / Valid	21 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
5.	Validasi masukan kredensial salah	Memasukkan <i>email</i> atau kata sandi yang salah pada <i>form modal</i> melayang, lalu mengklik tombol " <i>Login</i> ".	Sistem menolak hak akses, melakukan <i>redirect back</i> , dan otomatis membuka kembali modal <i>login</i> melayang untuk menampilkan pesan <i>error validation</i> dari Laravel.	Laravel mengirimkan <i>error</i> pengondisian sesi, dan komponen <i>x-init Alpine.js</i> secara otomatis menahan modal tetap terbuka kokoh seraya menampilkan pesan "Kredensial yang Anda masukkan tidak cocok dengan data kami".	Sesuai / Valid	21 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
6.	Otorisasi Akses Sukses User (<i>Visitor/Admin</i>)	Memasukkan <i>email</i> dan kata sandi <i>User</i> yang sah pada <i>form modal</i> melayang, lalu mengklik tombol " <i>Login</i> ".	Sesi baru dibuat, server meregenerasi token keamanan guna mencegah <i>Session Fixation</i> , dan rute mengalihkan <i>User</i> secara murni menuju halaman <i>dashboard</i> .	Sistem berhasil memverifikasi peran akun pada <i>database</i> MySQL, menghancurkan sisa kuki <i>guest</i> , dan mengarahkan pengguna ke halaman panel kendali utama.	Sesuai / Valid	21 April 2026
7.	Sinkronisasi data via web UI (<i>dashboard.blade.php</i>)	Menekan tombol " <i>Sync NASA</i> " pada antarmuka <i>dashboard</i> utama pada posisi kanan atas.	Sistem mengirimkan <i>HTTP Request</i> bertahap (perulangan 6 minggu terakhir) ke API NASA CNEOS tanpa verifikasi SSL.	Sistem berhasil mengunduh seluruh data historis, mengabaikan duplikasi data lama, melakukan <i>bulk insert</i> ke <i>database</i> MySQL melalui <i>model Eloquent</i> .	Sesuai / Valid	21 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
8.	Pemicuan otomatis mesin kecerdasan buatan latar belakang	Sinyal sukses transaksi <i>bulk insert</i> database MySQL diteruskan.	Laravel merespons otomatis dengan memicu perintah fungsi terminal <i>shell_exec()</i> .	Skrip aktif Python berjalan mandiri tanpa memicu <i>I/O bottleneck</i> .	Sesuai / Valid	21 April 2026
9.	Mekanisme <i>Caching</i> Prediksi (<i>dashboard & prediksi</i>)	Mengakses atau memuat ulang (<i>refresh</i>) halaman <i>Dashboard</i> Utama dan Grafik Prediksi Lintasan.	Pembacaan data proyeksi tidak membebani <i>hardware server</i> karena sistem tidak mengambil data secara berulang dari <i>database</i> , melainkan dari memori jangka pendek.	Beban pemuatan ulang berjalan instan karena sistem berhasil mengambil data hasil inferensi RNN melalui fungsi <i>Cache::get('neo_predictions')</i> .	Sesuai / Valid	21 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
10.	Fungsi keluar sistem terproteksi	Mengklik tombol "Keluar Sistem" pada area bawah <i>sidebar navigation</i> .	<i>JavaScript</i> melakukan perintah <i>submit form POST</i> terproteksi token <i>@csrf</i> menuju <i>AuthController @logout</i> .	Sesi otentikasi di server dihancurkan secara bersih, baris data pada tabel <i>sessions</i> di-clear, dan pengguna dialihkan kembali ke halaman <i>landing page</i> .	Sesuai / Valid	21 April 2026

(Sumber: Data Pribadi)

Hasil peninjauan kelayakan fungsional berdasarkan parameter pengujian tugas akhir dirangkum ke dalam Tabel 5. 13 berikut:

Tabel 5. 13 Hasil *Review Sprint 2*

No.	Kebutuhan Sistem	Keterangan	Status
1.	Modul <i>Web Profile</i> dan <i>Dashboard</i>	Halaman profil web berhasil memuat elemen luar angkasa secara responsif, dan <i>dashboard</i> utama sukses me-render <i>widget</i> kartu ringkasan statistik objek dekat Bumi tanpa kendala latensi.	Valid
2.	Sistem Autentikasi Akun	Fungsi pendaftaran (<i>register</i>) dan masuk (<i>login</i>) yang dijembatani oleh <i>Glassmorphism Pop-up Modal</i> berbasis <i>Alpine.js</i> sukses memproses kredensial pengguna dan mengunci pengamanan rute.	Valid
3.	Sinkronisasi Data NASA API	Logika fungsi <i>syncNasaData()</i> pada kontroler berhasil mengeksekusi <i>batch request</i> 6 minggu ke belakang menggunakan <i>facade</i> HTTP Laravel, serta sukses menerapkan penyaringan <i>smart-skip</i> untuk mencegah duplikasi data di MySQL.	Valid

No.	Kebutuhan Sistem	Keterangan	Status
4.	Pemicuan Eksekusi Inferensi AI	Sistem sukses mengotomatisasikan pemanggilan skrip kecerdasan buatan <i>Python (predict_neo.py)</i> melalui fungsi perintah bawaan <i>shell_exec</i> sesaat setelah proses <i>bulk insert database</i> MySQL selesai dilakukan.	Valid

(Sumber: Data Pribadi)

Berdasarkan matriks evaluasi pada Tabel 5. 13, seluruh blok fungsionalitas yang dirancang pada *Sprint 2* dinyatakan bekerja secara valid. Tidak ada catatan evaluasi kritis dari setiap *item backlog* yang ada di *sprint* kedua.

5.1.5.3. Hasil *sprint 3*

5.1.5.3.1. *Daily scrum*

Pada *Sprint 3*, aktivitas *Daily Scrum* difokuskan pada penguatan aspek visualisasi data dan analisis mendalam bagi pengguna melalui modul Katalog, Detail Objek, Prediksi AI, Analisis AI, serta Glosarium AI. Tim melakukan koordinasi harian untuk memastikan setiap data historis yang ditarik dari API NASA CNEOS dapat ditransformasikan ke dalam representasi grafis yang komunikatif di halaman *dashboard* maupun tabel katalog. Hambatan teknis yang mencuat dalam siklus ini berkaitan dengan tantangan performa saat merender data tabular berskala masif, namun berhasil diatasi melalui implementasi fungsi paginasi Laravel 12 yang efisien. Selain itu, tim pengembang berfokus pada integrasi *Traffic Light System* pada modul detail objek sebagai instrumen edukasi literasi astronomi bagi pengguna awam, serta memastikan grafik *Chart.js* pada menu Analisis AI mampu mengekstraksi metrik konvergensi *Loss History* dari berkas *training_log.json* secara presisi. Seluruh fitur pada siklus ini telah melalui tahap validasi fungsional untuk memastikan bahwa interpretasi data fisis objek langit tidak mengalami distorsi saat dirender di peramban web.

Tabel 5. 14 Hasil *Daily Scrum Sprint 3*

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
22 April 2026	Validasi <i>refresh UI</i> pasca inferensi	Sinkronisasi data antara <i>Python</i> & <i>Laravel</i>	Finalisasi rancangan tabel Katalog NEO
23 April 2026	Implementasi tabel katalog & paginasi	<i>Overhead</i> memuat volume data besar	Pengoptimalan kueri SQL dengan paginasi <i>Laravel</i>
24 April 2026	<i>Coding</i> halaman detail objek	Logika <i>Traffic Light System</i> (TLS)	<i>Final touch</i> visualisasi & informasi detail objek
25 April 2026	Integrasi data fisis objek ke detail	Konversi satuan unit jarak (km/AU/LD)	Finalisasi rancangan tabel (data tabular) Prediksi AI
26 April 2026	Pengembangan halaman Prediksi AI	-	Finalisasi rancangan Analisis AI & Visualisasi grafik konvergensi <i>loss history</i> (<i>Recursive Forecasting</i>)
27 April 2026	Pengembangan modul Analisis AI	<i>Parsing</i> berkas log pelatihan JSON	Finalisasi rancangan Glosarium AI

(Sumber: Data Pribadi)

5.1.5.3.2. *Design*

A. *Activity dan sequence diagram*

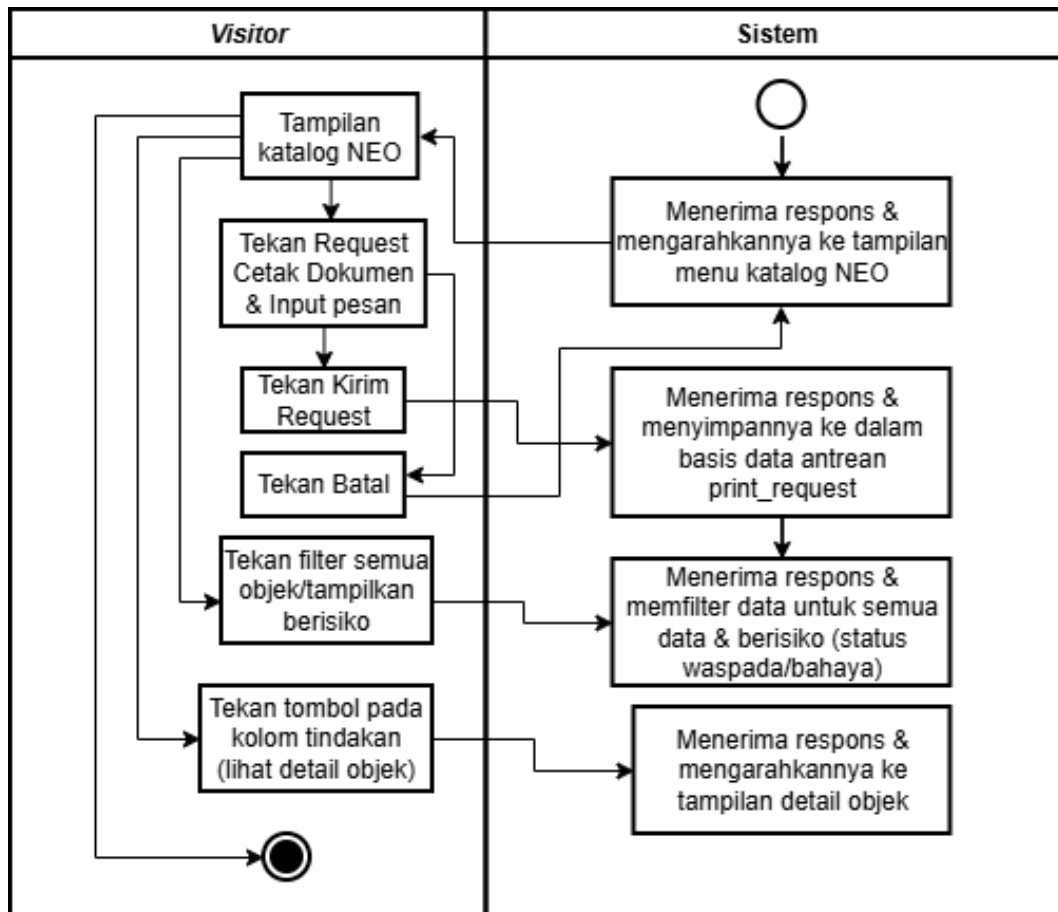
Perancangan *Activity Diagram* secara menyeluruh dilakukan untuk setiap butir *use case* yang telah didefinisikan sebelumnya. Pemodelan aktivitas ini bertujuan untuk memetakan alur kerja (*workflow*) logika sistem secara spesifik, memperjelas transisi interaksi antara aktor (*front-end UI*) dengan *respons* pengontrol (*back-end*), serta menggambarkan bagaimana data mengalir secara linear pada setiap fungsi di dalam arsitektur *hybrid environment* SINEO.

Guna melengkapi pemodelan struktural statis, interaksi dinamis berorientasi waktu dan pertukaran pesan (*message passing*) antar-objek di dalam ekosistem SINEO dimodelkan secara eksplisit memanfaatkan *sequence diagram*. Pemodelan *behavioral* ini bertujuan untuk menggambarkan urutan kronologis operasi koneksi hibrida (*hybrid connection*) yang melibatkan siklus hidup objek secara utuh. Seluruh rangkaian skenario interaksi dipetakan secara sinkron dan linear dengan *activity diagram* yang telah dijabarkan sebelumnya, mencakup fungsionalitas di sisi pengguna umum (*Visitor*), hak akses khusus (*Administrator*), hingga otomatisasi komputasi di latar belakang server (*back-end*).

Guna menyajikan dokumentasi sistem yang terstruktur dan komprehensif, urutan pertukaran pesan dinamis ini dijabarkan ke dalam skenario operasi sebagai berikut:

1) Melihat katalog NEO

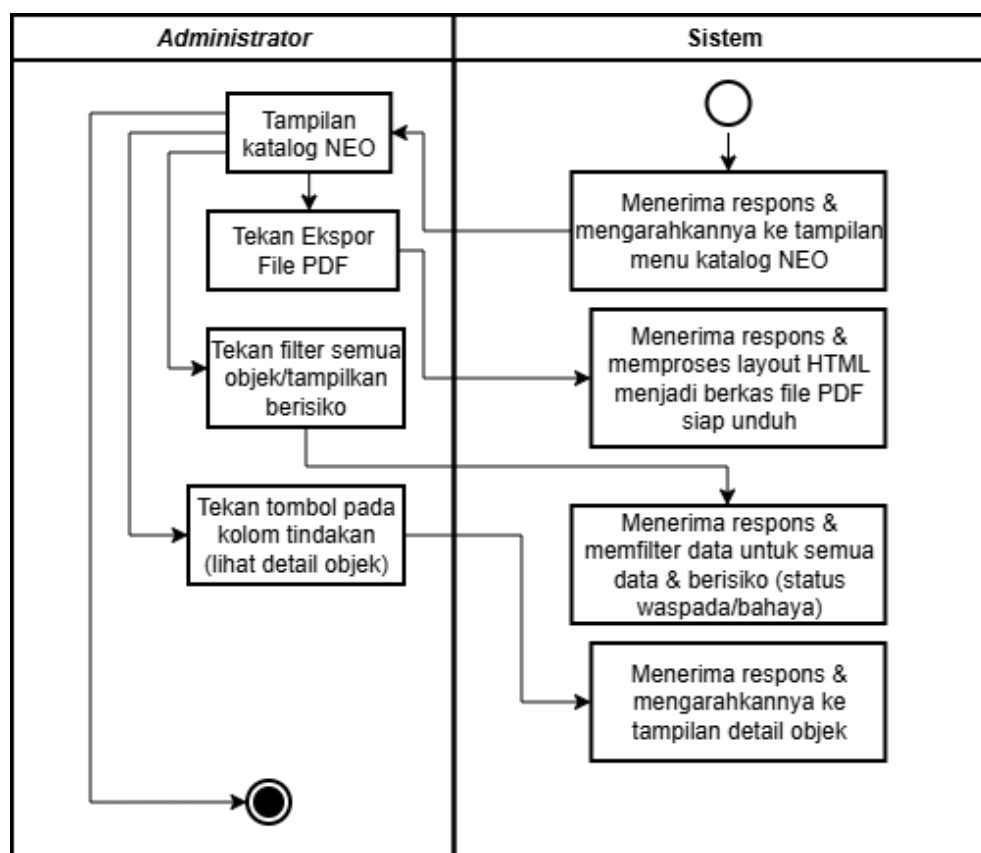
1) *Activity diagram* melihat katalog NEO



Gambar 5. 39 *Activity Diagram* Melihat Katalog NEO Visitor
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 39, diagram aktivitas tersebut diinisiasi oleh Sistem melalui simpul awal (*initial state*) dengan mengeksekusi perintah menerima respons dan mengarahkannya ke tampilan menu katalog NEO, sehingga dokumen data tersebut dapat disajikan secara utuh pada layar peramban. Setelah masuk ke dalam antarmuka tampilan katalog NEO, aktor *Visitor* dihadapkan pada tiga pilihan opsi tindakan reaktif yang dapat dieksekusi secara dinamis sesuai kebutuhan informasi. Pada opsi tindakan pertama, *Visitor* dapat melakukan pengajuan dokumen digital melalui instruksi tekan *request* cetak dokumen kemudian input pesan dan tekan batal untuk kembali atau kirim *request* untuk lanjut, di mana Sistem akan menangkap parameter tersebut

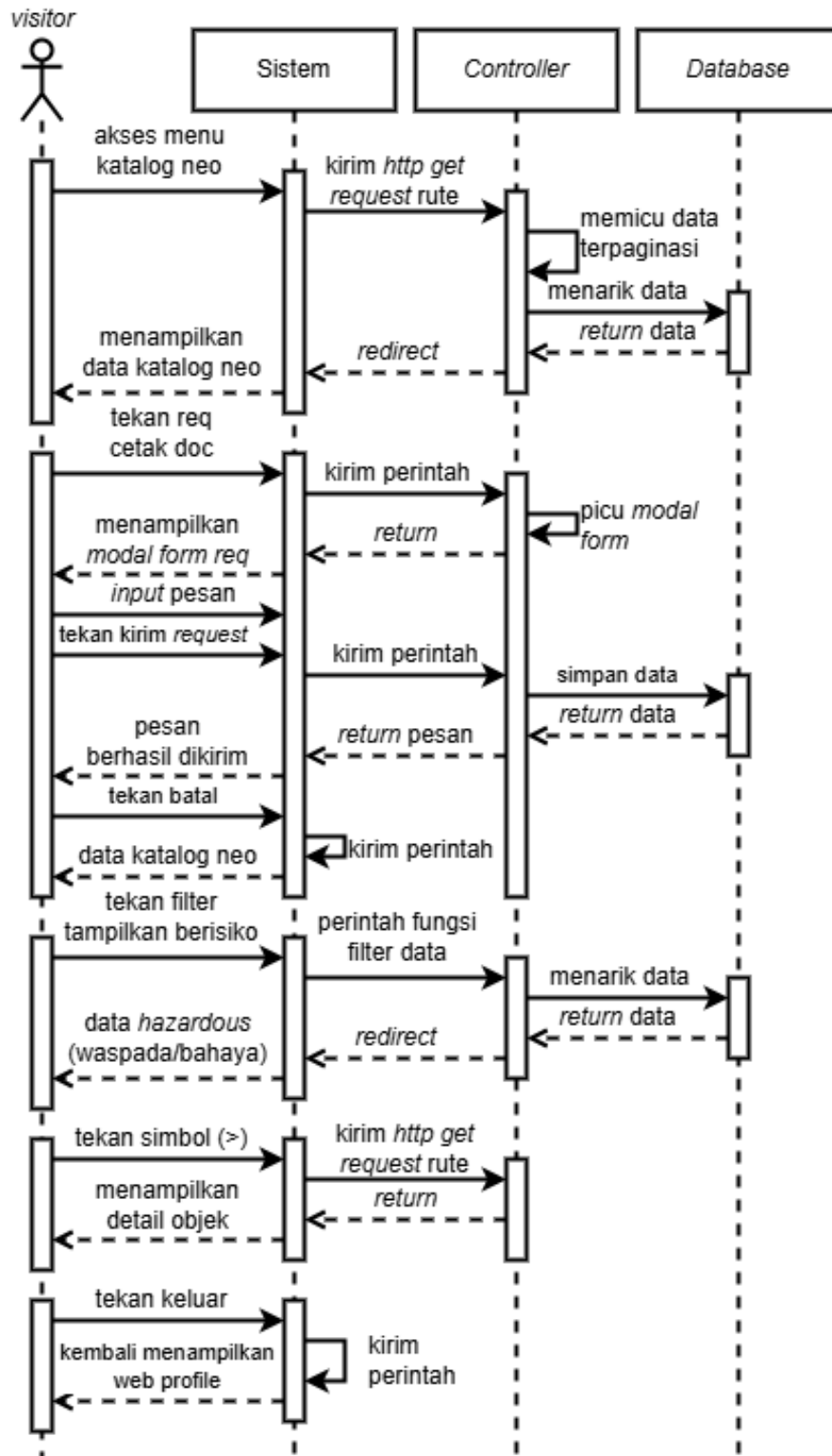
untuk menjalankan fungsi menerima *respons* dan menyimpannya ke dalam basis data antrean *print_request* secara aman. Pada opsi tindakan kedua, pengguna dapat memicu fungsi penyaringan data melalui instruksi tekan *filter* semua objek/tampilkan berisiko, yang ditanggapi secara otomatis oleh sistem dengan mengeksekusi perintah menerima respons dan memfilter data untuk semua data dan berisiko berdasarkan klasifikasi status waspada atau bahaya. Pada opsi tindakan ketiga, pengguna dapat melakukan penelusuran spasial asteroid secara individual dengan memilih instruksi tekan tombol pada kolom tindakan (lihat detail objek), di mana Sistem secara tanggap merespons lewat perintah menerima respons dan mengarahkannya ke tampilan detail objek. Seluruh rangkaian siklus aktivitas eksplorasi data katalog oleh *Visitor* ini akan berakhir dan menuju simpul penutup (*final state*) ketika pengguna selesai berinteraksi atau memutuskan untuk keluar dari halaman data tabular tersebut.



Gambar 5. 40 Activity Diagram Melihat Katalog NEO Admin
(Sumber: Data Pribadi)

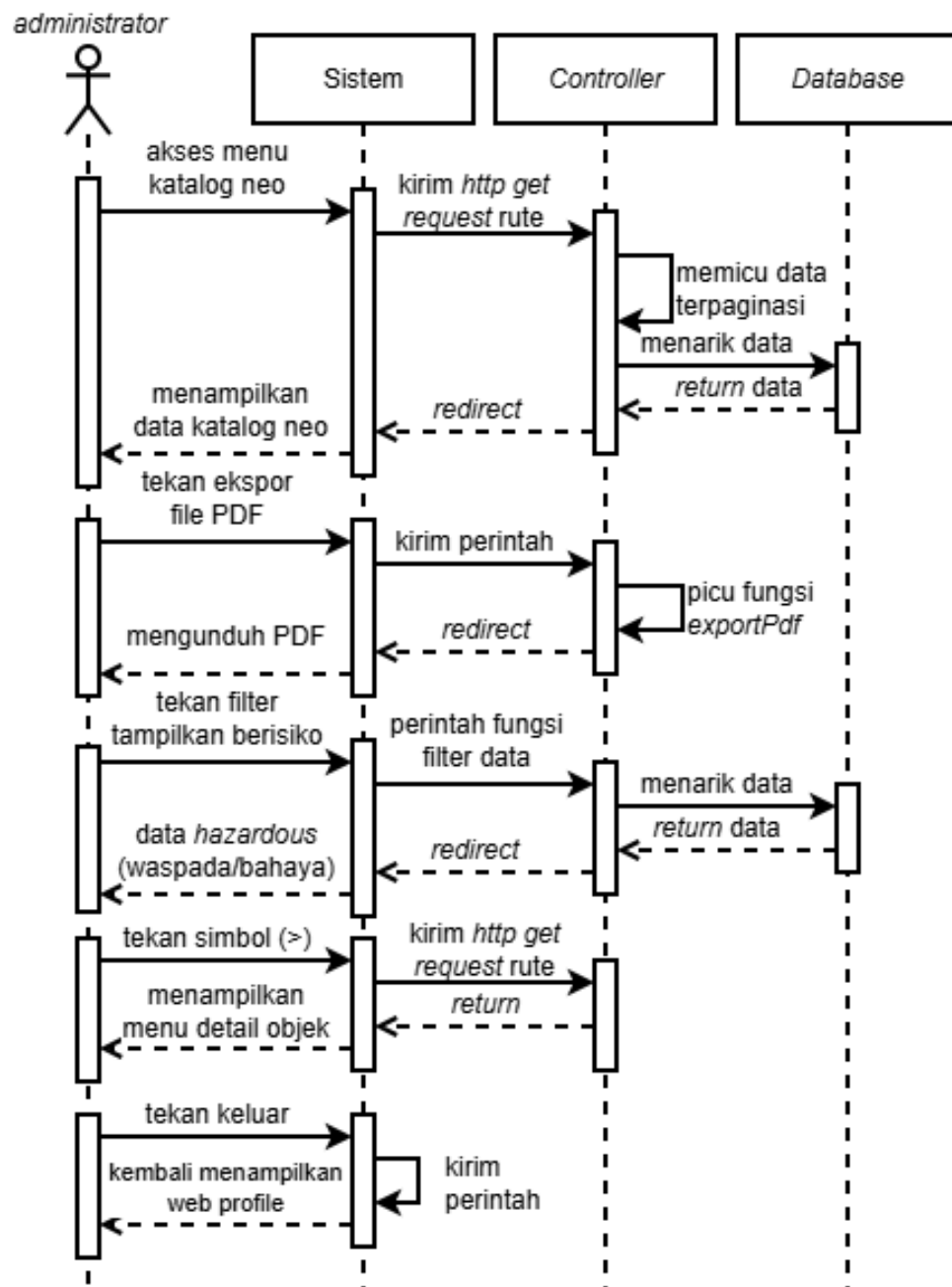
Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 40, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) di bawah otoritas sistem yang memproses perintah menerima *respons* dan mengarahkannya ke tampilan menu katalog NEO hingga memunculkan visualisasi tabel data spasial secara penuh bagi *Admin*. Tepat setelah halaman termuat sempurna, *Administrator* dapat mengontrol basis data melalui tiga opsi tindakan operasional utama yang tersedia pada antarmuka. Pada opsi tindakan pertama yang membedakannya dengan pengguna biasa, *Admin* dapat mengeksekusi penarikan dokumen secara murni melalui instruksi tekan ekspor *file* PDF, di mana Sistem langsung merespons secara hibrida dengan memicu fungsi menerima respons dan memproses *layout* HTML menjadi berkas *file* PDF siap unduh berukuran kertas A4. Pada opsi tindakan kedua, *Admin* dapat melakukan inspeksi populasi asteroid lewat instruksi tekan *filter* semua objek/tampilkan berisiko, yang direspon oleh Sistem dengan menjalankan perintah menerima *respons* dan memfilter data untuk semua data dan berisiko guna mengisolasi status objek berkategori waspada atau bahaya. Pada opsi tindakan ketiga, *Admin* dapat melakukan audit fisis mendalam objek *celestial* secara individual dengan instruksi tekan tombol pada kolom tindakan (lihat detail objek), di mana sistem akan memproses perintah menerima *respons* dan mengarahkannya ke tampilan detail objek secara instan. Seluruh rangkaian manajemen siklus katalog oleh *Administrator* ini dinyatakan selesai dan ditutup secara resmi pada simpul akhir (*final state*) setelah *Admin* selesai beraktivitas atau meninggalkan halaman manajemen tersebut.

2) *Sequence diagram* melihat katalog NEO



Gambar 5. 41 *Sequence Diagram* Melihat Katalog NEO (Visitor)
(Sumber: Data Pribadi)

Berdasarkan skema pemodelan kronologis lintas objek pada Gambar 5. 41, mengenai alur katalog *Near-Earth Objects* bagi *Visitor* menunjukkan rangkaian interaksi mandiri yang berfokus pada pengajuan permohonan dokumen katalog. Skenario diawali ketika aktor *Visitor* membuka antarmuka utama aplikasi melalui objek Katalog NEO View untuk menjelajahi sajian data asteroid hasil tarikan dari API NASA. Pada ruang antarmuka tersebut, *Visitor* dapat memberikan interaksi masukan berupa penekanan tombol ajukan cetak dokumen yang dibarengi dengan pengisian catatan alasan keperluan atau pesan deskripsi tertentu. Komponen antarmuka Katalog NEO View meneruskan *payload* kiriman data formulir tersebut menuju objek *Print Request Controller* menggunakan metode simpan atau *store*. Begitu data parameter *request* diterima, objek pengendali langsung melakukan operasi penulisan rekaman antrean data baru ke dalam *database* MySQL melalui instruksi pembuatan instansiasi baris data pada objek model *Print Request*. Sesaat setelah basis data MySQL berhasil mengunci dan menyimpan struktur baris data permohonan tersebut dengan status awal berupa antrean tertunda, sinyal konfirmasi sukses dikembalikan menuju objek pengendali. Sebagai sentuhan akhir dari siklus interaksi ini, yang kemudian menyajikan tanda pemberitahuan berhasil berwarna hijau di layar *Visitor* guna mengonfirmasi bahwa pengajuan cetak biner mereka telah aman masuk ke dalam sistem *monitoring admin*.



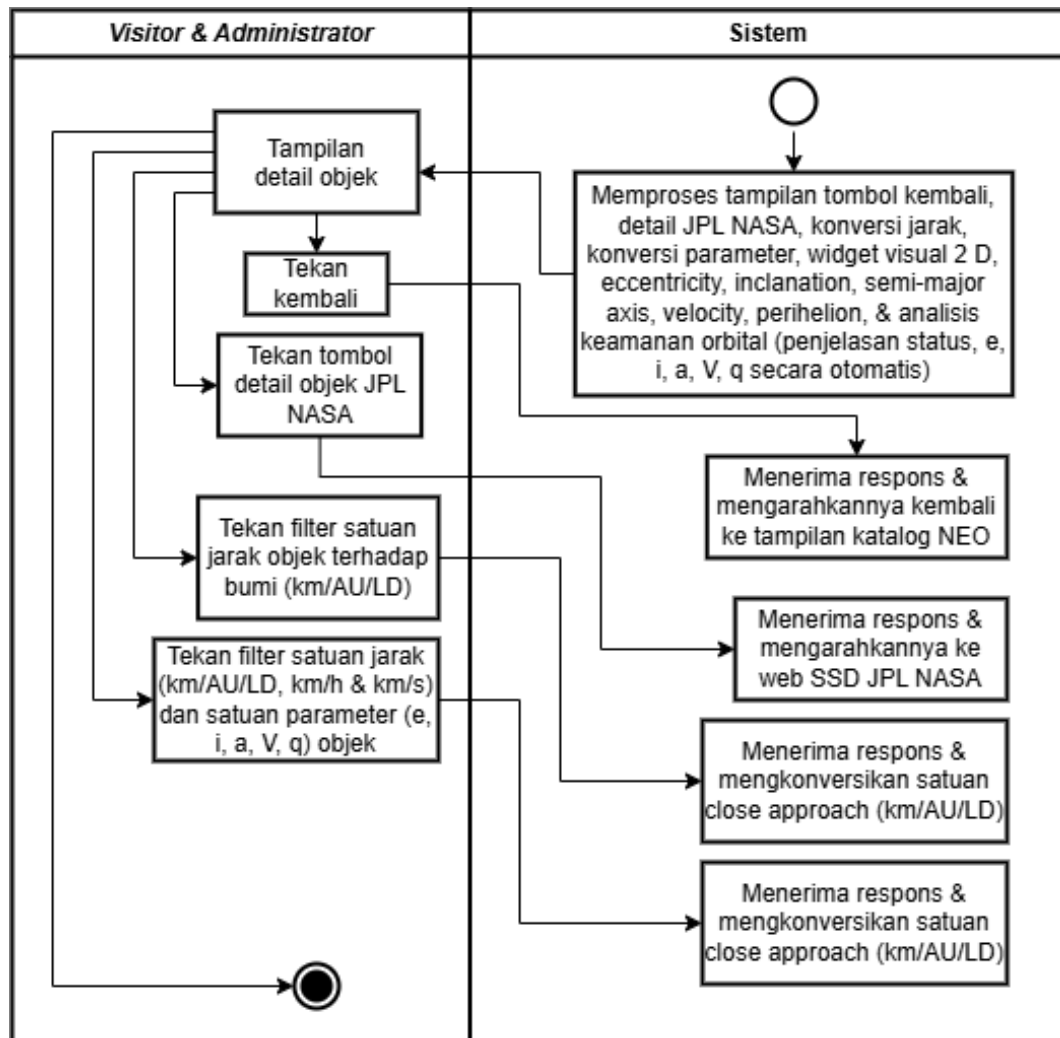
Gambar 5. 42 *Sequence Diagram* Melihat Katalog NEO (Admin)
(Sumber: Data Pribadi)

Berdasarkan manifestasi struktur diagram sekuensial pada Gambar 5. 42, alur ekspor laporan cetak katalog *Near-Earth Objects* oleh *Admin*, rangkaian interaksi diawali ketika aktor *Administrator* membuka antarmuka utama aplikasi melalui objek Katalog NEO View untuk menjelajahi sajian data asteroid hasil tarikan dari API NASA. Pada ruang antarmuka tersebut, *Admin* dapat memberikan interaksi

berupa penekanan tombol ekspor PDF langsung pada antarmuka Katalog NEO View. Komponen antarmuka ini menindaklanjutinya dengan mengirimkan permintaan pembuatan dokumen biner melalui pemanggilan fungsi *exportPdf* menuju objek *PrintRequestController* dengan menyertakan parameter jenis berupa katalog. Objek pengendali kemudian memicu instruksi optimasi server latar belakang secara internal melalui pengaturan batas waktu eksekusi serta batas memori guna mencegah terjadinya *timeout* sistem sewaktu memproses tumpukan data besar. Alur kerja dilanjutkan secara mandiri oleh kontroler dengan melakukan kueri penarikan seluruh data *riil* hasil sinkronisasi NASA secara menyeluruh dari *database* MySQL melalui perantara objek model *NeoObject*. Setelah objek model mengembalikan kumpulan data mentah asteroid, kontroler memproses data tersebut secara terstruktur dengan mengelompokkannya berdasarkan parameter bulan dan tahun kedatangan menggunakan bantuan komponen *carbon*. Siklus interaksi berlanjut saat objek pengendali membersihkan *buffer* memori secara paksa dan memuat seluruh *payload data* laporan terstruktur ke dalam objek komponen *DomPDF* melalui pemanggilan berkas templat visual *SINEO_katalog_lengkap_pdf*. Sebagai luaran akhir dari skenario pencetakan mandiri ini, objek *DomPDF* menggenerasikan file dokumen biner *final* berformat PDF dan mengirimkannya kembali ke lapisan Katalog NEO View, yang secara instan memicu browser pada komputer lokal milik *Admin* untuk memulai proses unduh otomatis ke dalam media penyimpanan lokal.

2) Membuka detail objek

1) *Activity diagram* membuka detail objek



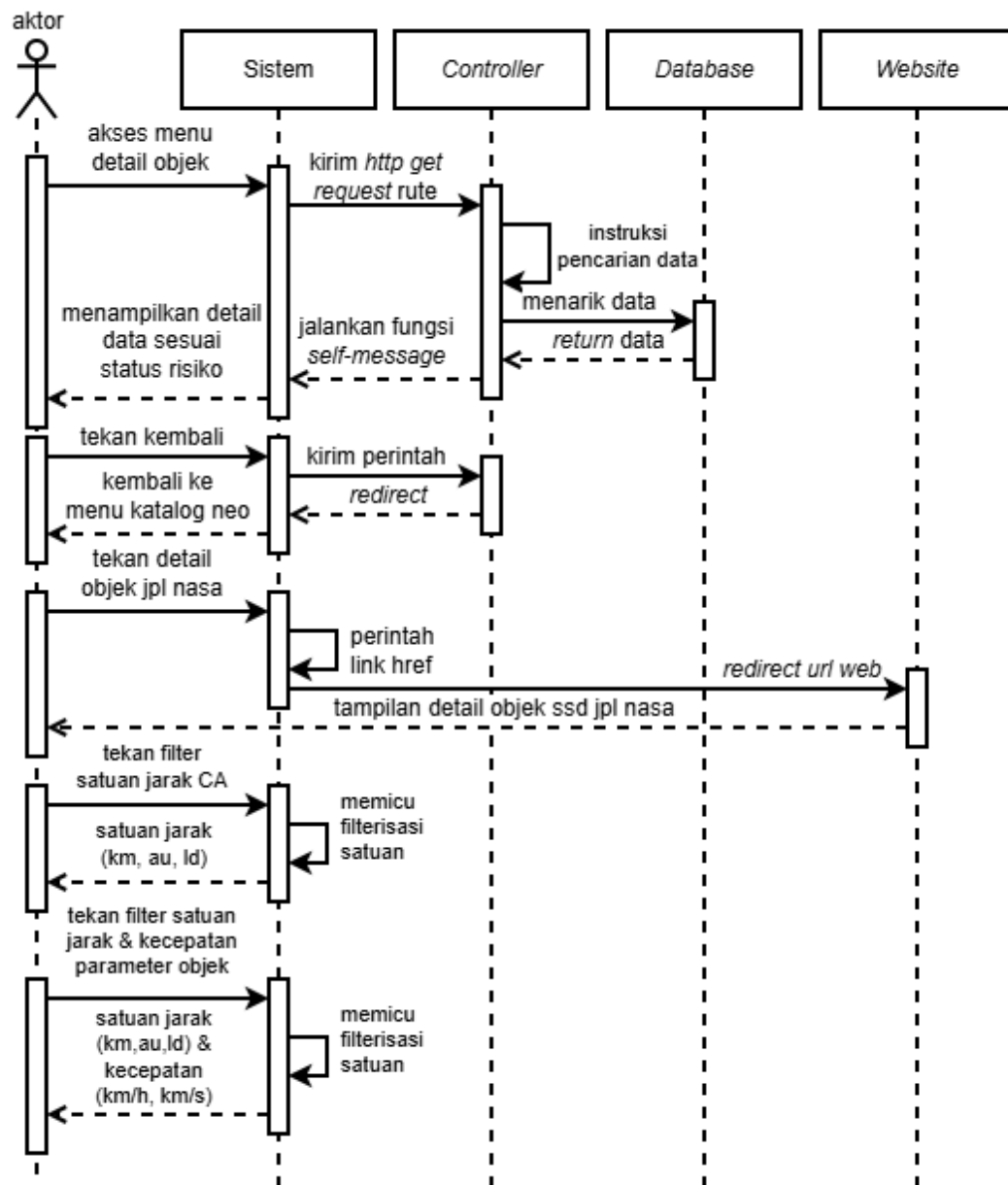
Gambar 5. 43 *Activity diagram* membuka detail objek
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 43, diagram aktivitas tersebut memetakan urutan logika interaksi mendalam antara aktor *Visitor & Administrator* dengan komponen sistem pada halaman spesifikasi individual objek dekat Bumi. Aliran aktivitas diawali dari simpul awal (*initial state*) di sisi sistem, di mana sistem secara aktif memproses pemuatan elemen antarmuka yang meliputi tombol kembali (katalog NEO), detail JPL NASA, konversi jarak, konversi parameter, *widget* visual 2D, parameter *eccentricity*, *inclination*, *semi-*

major axis, *velocity*, *perihelion*, serta modul analisis keamanan orbital yang memuat penjelasan status fisis ($e, i, a, V, q, \emptyset, H$) secara otomatis berbasis *database*. Seluruh luaran komponen tersebut kemudian disajikan secara utuh kepada pengguna dalam bentuk Tampilan detail objek. Setelah antarmuka termuat sempurna, alur kerja bertransisi menjadi empat opsi tindakan interaktif yang dapat dipilih oleh pengguna secara dinamis.

Opsi tindakan pertama, pengguna dapat mengeklik tombol tekan kembali, di mana sistem akan menerima *respons* perintah tersebut lalu mengarahkan rute navigasi peramban mundur ke tampilan utama katalog NEO. Opsi tindakan kedua, pengguna dapat mengeklik komponen tekan tombol detail objek JPL NASA. Aksi ini ditangkap sebagai instruksi jembatan keluar oleh sistem, yang langsung merespons dengan mengalihkan rute penjelajahan menuju halaman eksternal resmi web SSD JPL NASA (https://ssd.jpl.nasa.gov/tools/sbdb_lookup.html#/?sstr={id}). Opsi tindakan ketiga, pengguna dapat mengeklik fungsionalitas tekan *filter* satuan jarak objek terhadap bumi (km/AU/LD), yang direspon oleh sistem dengan menerima *respons* serta secara instan mengonversikan visualisasi metrik satuan jarak pendekatan terdekat (*close approach*) sesuai preferensi yang dipilih. Opsi tindakan keempat, pengguna dapat mengeklik fungsionalitas tekan *filter* satuan jarak (km/AU/LD, km/h & km/s) dan satuan parameter ($e, i, a, V, q, \emptyset, H$) objek, di mana sistem kembali menerima *respons* masukan tersebut untuk mengalkulasikan serta mengonversikan tampilan seluruh kombinasi metrik fisis dan parameter orbital secara reaktif. Akhirnya, apabila seluruh eksplorasi data, pengondisian *filter* satuan, ataupun pengalihan rute eksternal tersebut selesai dieksekusi, aliran diagram aktivitas detail objek ini akan bermuara secara linear pada simpul penutup atau *final state*.

2) *Sequence diagram* membuka detail objek



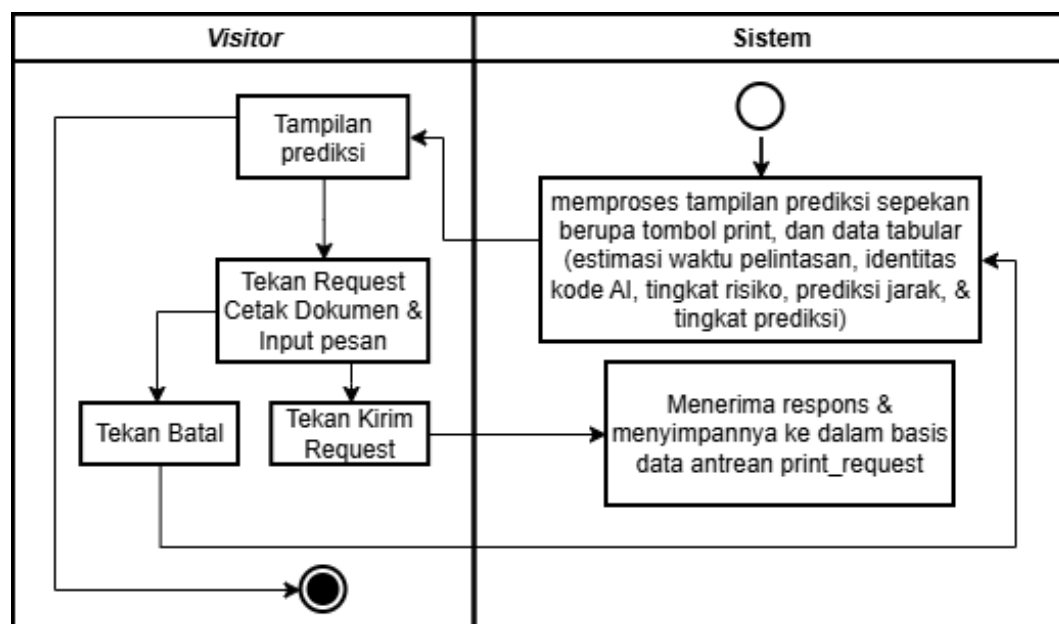
Gambar 5. 44 *Sequence Diagram* Membuka Detail Objek
(Sumber: Data Pribadi)

Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 44, diagram urutan ini menggambarkan aliran data operasional ketika aktor melakukan inspeksi mendalam karakteristik fisis asteroid secara individual menggunakan algoritma *Traffic Light System*. Alur pesan dimulai saat aktor mengeklik tombol inspeksi detail objek ikon panah (>) pada baris data tabel di dalam antarmuka

katalog.blade.php, yang kemudian mengirimkan sinyal pesan rute URL berupa fungsi *show(\$id)* mengarah ke objek pengontrol *NeoController*. Objek *NeoController* menindaklanjuti instruksi tersebut dengan mengirimkan parameter pencetakan kueri *NeoObject::findOrFail(\$id)* kepada model entitas *NeoObject* untuk menarik larik data parameter orbital lengkap (*e*, *i*, *a*, *V*, *q*, *miss_distance_km*) dari *database* MySQL. Tepat setelah data dikembalikan ke kontroler, *NeoController* mengeksekusi pesan ke diri sendiri (*self-message*) untuk mengalkulasikan ambang batas kritis proksimitas spasial dan secara reaktif meneruskan variabel data beserta kode warna tema (Hijau/Kuning/Merah) menuju komponen *detail_object.blade.php* untuk disajikan di peramban pengguna.

3) Memantau prediksi AI

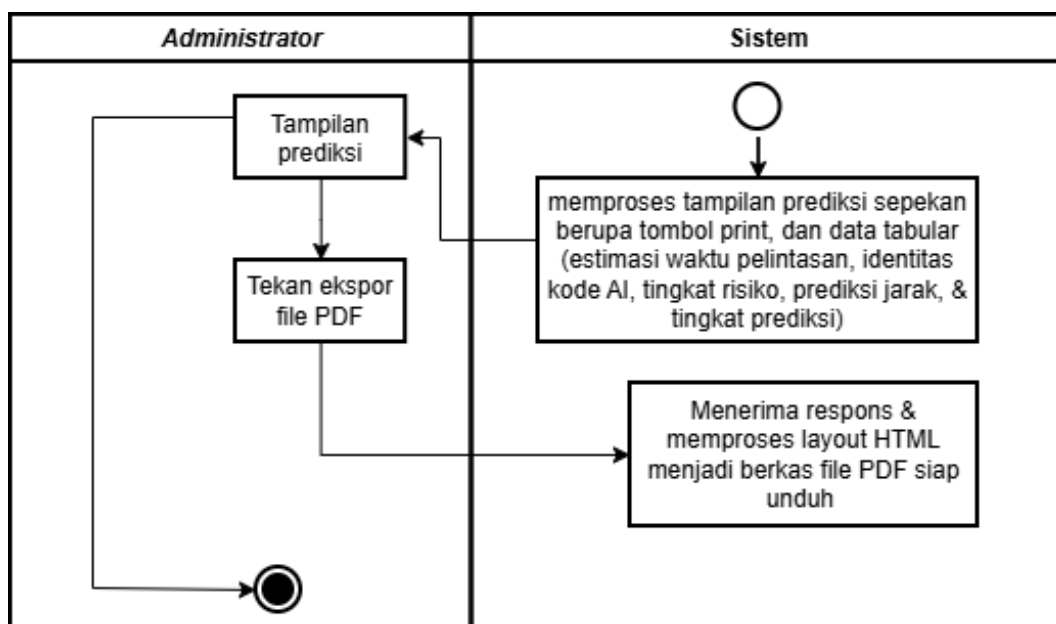
1) Activity diagram memantau prediksi AI



Gambar 5. 45 Activity Diagram Memantau Prediksi AI Visitor
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 45, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) oleh pihak sistem yang merespons rute masuk menu dengan memproses tampilan prediksi sepekan, dan data tabular yang memuat rincian estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, beserta tingkat prediksi secara dinamis. Hasil

pemetaan sekuensial tersebut kemudian disajikan secara utuh di layar peramban pengguna pada bagian tampilan prediksi. Karena adanya batasan hak akses cetak, aktor *Visitor* tidak dapat mengunduh dokumen secara langsung, melainkan diarahkan untuk mengeksekusi instruksi tekan *request* cetak dokumen kemudian input pesan, setelah itu tekan batal untuk kembali dan tekan kirim *request* untuk lanjut. Tindakan interaktif dari *Visitor* ini dikirimkan kembali menuju server, di mana pihak sistem akan memproses perintah tersebut lewat fungsi menerima *respons* & menyimpannya ke dalam basis data antrean *print_request* agar nantinya dapat ditinjau dan divalidasi oleh pihak *Admin*. Seluruh rangkaian aktivitas pemantauan kurva dan pengajuan kueri cetak prediksi oleh *Visitor* ini resmi berakhir dan mencapai simpul penutup (*final state*) tepat setelah halaman antarmuka kembali ke kondisi sedia untuk dijelajahi.

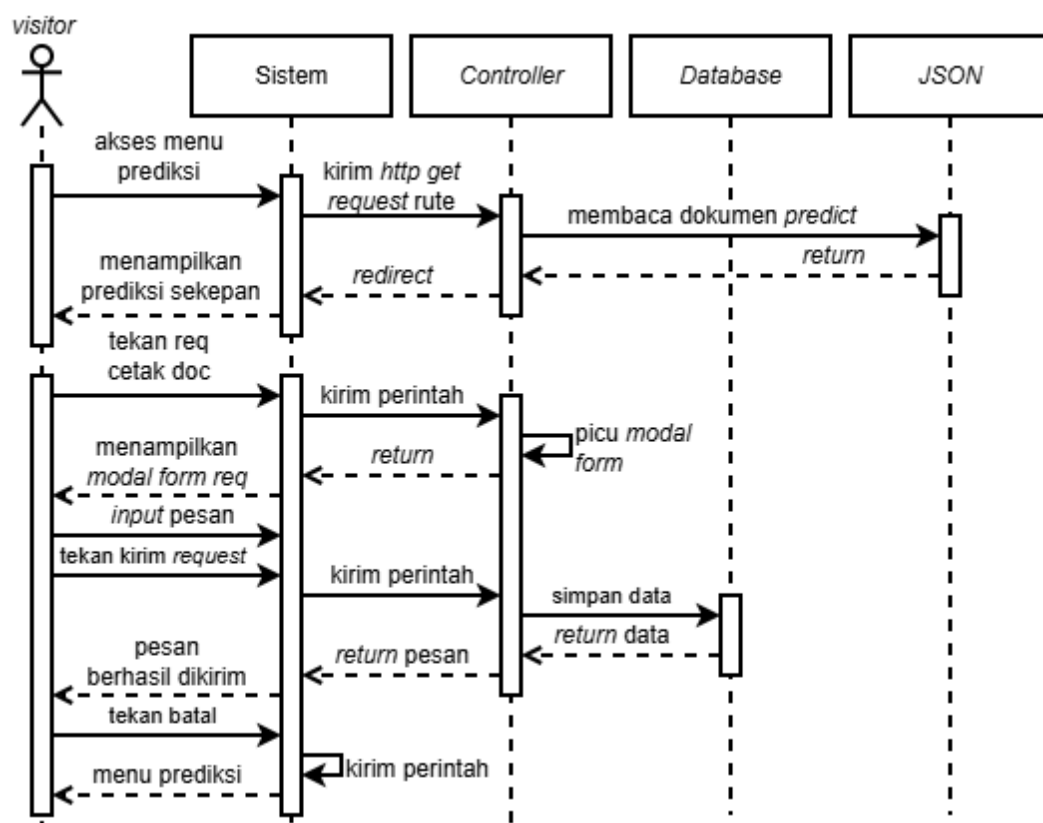


Gambar 5. 46 Activity Diagram Memantau Prediksi AI Admin
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 46, diagram aktivitas tersebut diawali dari simpul awal (*initial state*) di sisi sistem, di mana sistem secara otomatis menjalankan proses memproses tampilan prediksi sepekan berupa tombol ekspor *file* PDF, dan data tabular yang mencakup informasi estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, serta

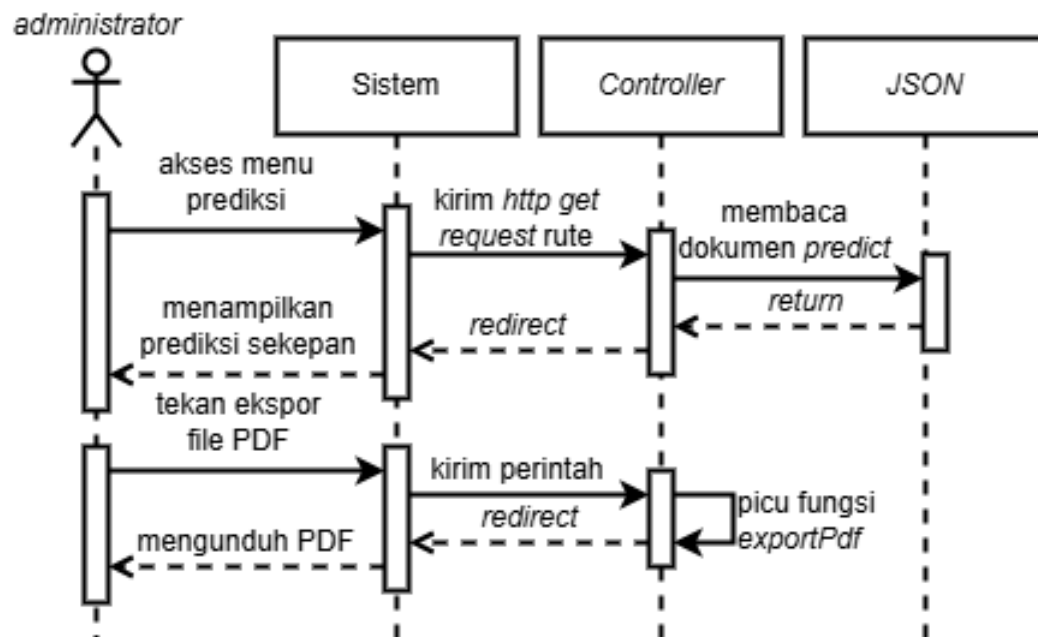
tingkat prediksi hasil kalkulasi model jaringan saraf rekuren. Setelah data ter-*render* sempurna, aktor *Administrator* akan menerima visualisasi pada halaman tampilan prediksi. Langkah berikutnya, *Admin* dapat menindaklanjuti data peramalan sekuensial tersebut dengan melakukan instruksi tekan ekspor *file* PDF. Aksi dari pengguna otoritas ini dikirimkan kembali sebagai instruksi masuk ke sisi *back-end*, di mana sistem bekerja menanggapi perintah tersebut melalui fungsi menerima *respons* & memproses *layout* HTML menjadi berkas *file* PDF siap unduh agar dapat disimpan menjadi arsip digital laporan instansi. Aliran diagram aktivitas khusus *administrator* ini dinyatakan selesai dan secara langsung mengarah menuju titik simpul penutup (*final state*) sesaat setelah proses pembuatan berkas PDF selesai diolah oleh server.

2) Sequence diagram memantau prediksi AI



Gambar 5. 47 Sequence Diagram Memantau Prediksi AI (Visitor)
(Sumber: Data Pribadi)

Berdasarkan pemodelan logis koneksi hibrida pada Gambar 5. 47, mengenai alur menu prediksi lintasan bagi *Visitor* menunjukkan rangkaian operasi yang berfokus pada visualisasi prediksi cerdas dan pengajuan permohonan dokumen. Interaksi dimulai ketika aktor *Visitor* memuat halaman antarmuka *Prediksi AI View* yang secara otomatis memicu pengiriman instruksi penarikan data menuju objek komponen *NeoController*. Objek pengendali ini segera memanggil fungsi penanganan inferensi cerdas dengan memanfaatkan data runtun waktu yang tersedia di dalam pangkalan data MySQL melalui perantara objek model *NeoObject*. Guna menghasilkan proyeksi lintasan masa depan, objek kontroler mengirimkan instruksi eksekusi terminal menuju skrip kecerdasan buatan berbasis *Python* yang bertugas memuat berkas biner model berarsitektur *Recurrent Neural Network* (RNN). Hasil kalkulasi peramalan kontinu sepekan ke depan tersebut diekspor kembali oleh *Python* ke dalam struktur berkas JSON, diterima oleh kontroler untuk disimpan ke dalam memori *caching*, dan dikirimkan kembali ke lapisan *Prediksi AI View* untuk disajikan menjadi data tabular terpaginasi di hadapan *Visitor*. Siklus ditutup ketika *Visitor* menekan tombol *request* cetak dokumen pada antarmuka, di mana *payload* formulir beserta pesan catatan dikirimkan menuju objek *PrintRequestController* untuk diproses menjadi rekaman baris antrean permohonan baru berstatus tertunda pada model *Print Request* di dalam basis data MySQL, sebelum akhirnya sistem mengembalikan pesan konfirmasi sukses berwarna hijau ke layar komputer *Visitor*.

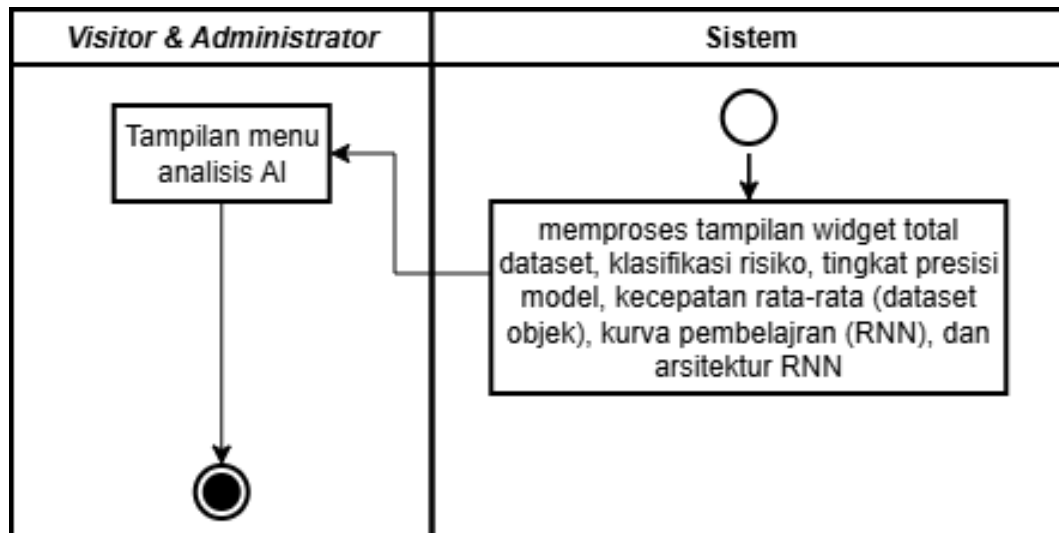


Gambar 5. 48 Sequence Diagram Memantau Prediksi AI (Admin)
(Sumber: Data Pribadi)

Berdasarkan pemodelan logis koneksi hibrida pada Gambar 5. 48, mengenai alur ekspor laporan prediksi lintasan objek dekat bumi oleh *Admin*, siklus interaksi diawali ketika aktor *Admin* memuat halaman antarmuka *prediksi.blade.php* yang secara otomatis memicu pengiriman instruksi penarikan data menuju objek komponen *NeoController*. Objek pengendali ini segera memanggil fungsi penanganan inferensi cerdas dengan memanfaatkan data runtun waktu yang tersedia di dalam pangkalan data MySQL melalui perantara objek model *NeoObject*. Guna menghasilkan proyeksi lintasan masa depan, objek kontroler mengirimkan instruksi eksekusi terminal menuju skrip kecerdasan buatan berbasis *Python* yang bertugas memuat berkas biner model berarsitektur *Recurrent Neural Network* (RNN). Hasil kalkulasi peramalan kontinu sepekan ke depan tersebut diekspor kembali oleh *Python* ke dalam struktur berkas JSON, diterima oleh kontroler untuk disimpan ke dalam memori *caching*, dan dikirimkan kembali ke lapisan *Prediksi Lintasan View Admin* juga dapat mengeksekusi tindakan penekanan tombol ekspor *file PDF* pada antarmuka *prediksi.blade.php*. Komponen antarmuka ini menindaklanjuti secara responsif dengan melempar parameter tipe berupa prediksi menuju objek komponen *PrintRequestController* melalui pemanggilan fungsi *exportPdf*.

Menyadari beban komputasi data besar, objek pengendali langsung melakukan optimalisasi kinerja server latar belakang dengan menonaktifkan batasan waktu eksekusi serta melonggarkan kapasitas alokasi memori sistem. Alur kerja berlanjut ketika kontroler melakukan pengecekan data lintasan secara runtun waktu ke dalam sistem penyimpanan *Cache* Laravel, yang apabila memori tersebut kosong, sistem akan beralih mengeksekusi pembacaan data cadangan terstruktur dari berkas *backup* JSON yang tersimpan di dalam direktori penyimpanan lokal. Tidak hanya memuat data proyeksi lintasan masa depan, kontroler juga melakukan operasi pembacaan nilai metrik akurasi latih model kecerdasan buatan secara murni dari berkas log pelatihan JSON untuk dikalkulasikan menjadi persentase tingkat presisi penalaran model RNN. Setelah seluruh variabel data performa sains dan jangkauan tanggal terkumpul utuh, kontroler membersihkan *buffer* memori secara berkala dan meneruskan *payload* tersebut ke dalam objek DomPDF dengan merujuk pada templat berkas visual *SINEO_prediksi_pdf*. Sebagai luaran final, objek *DomPDF* mengompilasi seluruh struktur data biner tersebut menjadi berkas dokumen PDF resmi dan mengirimkannya kembali ke lapisan antarmuka pengguna guna memicu browser lokal milik *Admin* agar mengeksekusi perintah unduh berkas secara otomatis ke dalam media penyimpanan komputer.

4) Memonitor analisis AI

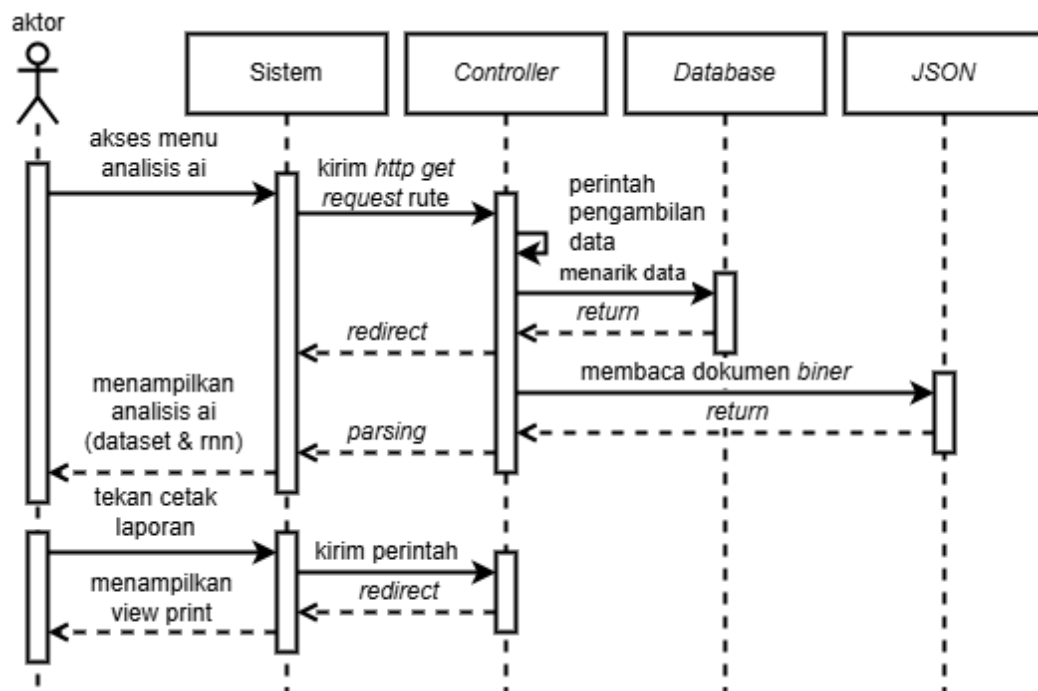
1) *Activity diagram* memonitor analisis AI

Gambar 5. 49 *Activity Diagram* Memonitor Analisis AI
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 49, diagram aktivitas tersebut memetakan urutan logika interaksi antara aktor *Visitor & Administrator* dengan komponen sistem pada halaman pemantauan log performa kecerdasan buatan. Aliran aktivitas dikendalikan dari simpul awal (*initial state*) di sisi sistem yang merespons rute masuk menu analisis dengan memproses data performa komputasi murni. Pihak sistem mengeksekusi instruksi pembacaan berkas log untuk memproses penayangan *widget* total *dataset*, klasifikasi risiko, tingkat presisi model, kecepatan rata-rata dari keseluruhan dataset objek, visualisasi kurva pembelajaran (*learning curve*) evaluasi penurunan nilai kesalahan jaringan saraf rekuren (RNN) selama 100 *epoch*, serta pemetaan skema arsitektur RNN bertumpuk yang tertanam di dalam aplikasi (*temporal input, dual-stacked RNN layer, dan recursive forecasting*). Seluruh luaran visual log pelatihan mendalam tersebut dikompilasi lalu disalurkan ke sisi pengguna dalam bentuk tampilan menu analisis AI yang informatif dan transparan. Aliran keseluruhan diagram aktivitas *monitoring* performa ini dirancang secara linear tanpa melibatkan input atau percabangan lanjutan dari pengguna, sehingga alur kerja langsung bermuara pada

simpul penutup atau *final state* tepat setelah keseluruhan statistik data *riil* dan grafik konvergensi *loss* sukses disajikan seutuhnya di layar peramban web.

2) *Sequence diagram* memonitor analisis AI



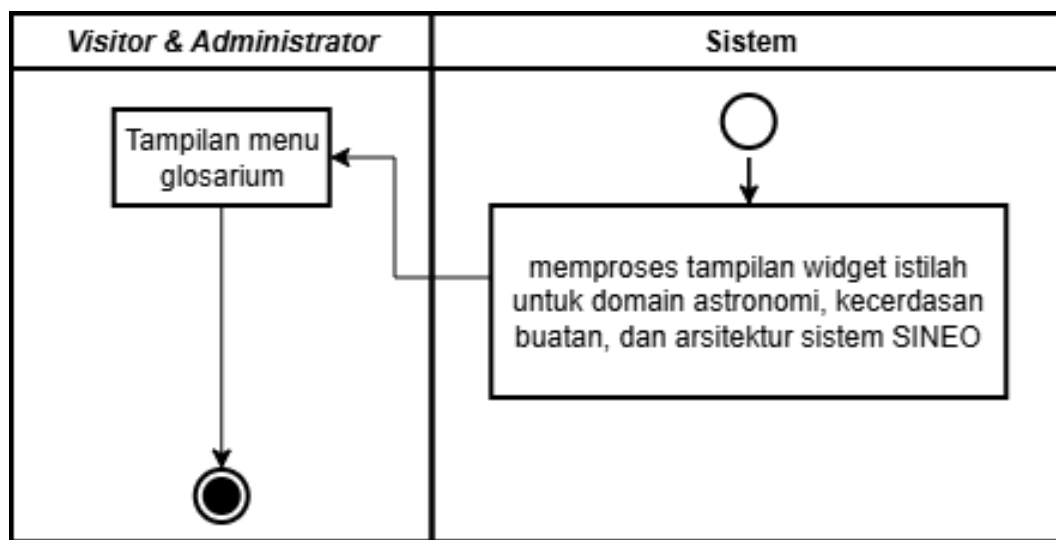
Gambar 5. 50 *Sequence Diagram* Memonitor Analisis AI
(Sumber: Data Pribadi)

Berdasarkan pemodelan visual yang tertera pada Gambar 5. 50, diagram urutan ini mendokumentasikan visualisasi evaluasi performa model kecerdasan buatan berbasis grafik konvergensi penurunan nilai *error (loss)* kepada aktor. Siklus interaksi berjalan ketika pengguna mengeklik menu Analisis pada bilah menu samping, di mana berkas komponen visual *analisis.blade.php* mengirimkan pesan permintaan rute menuju metode pengendali *showAnalysis()* pada objek pengontrol *NeoController*. Objek *NeoController* menangkap perintah tersebut lalu memuat rekaman riwayat latihan model melalui pembacaan berkas fisik log internal server bernama *training_log.json*. Dokumen teks log tersebut merespons dengan mengembalikan struktur data histori akurasi latihan *epoch* dari arsitektur jaringan saraf tiruan (*dual-stacked RNN*), untuk kemudian dikirimkan kembali oleh *NeoController* menuju objek tampilan *analisis.blade.php* agar dapat dipetakan menjadi grafik evaluasi performa konvergensi *loss* yang informatif, *chart* total

dataset, klasifikasi risiko, tingkat presisi model RNN, kecepatan rata-rata, informasi tentang model RNN pada sistem.

5) Membaca glosarium AI

1) *Activity diagram* membaca glosarium AI

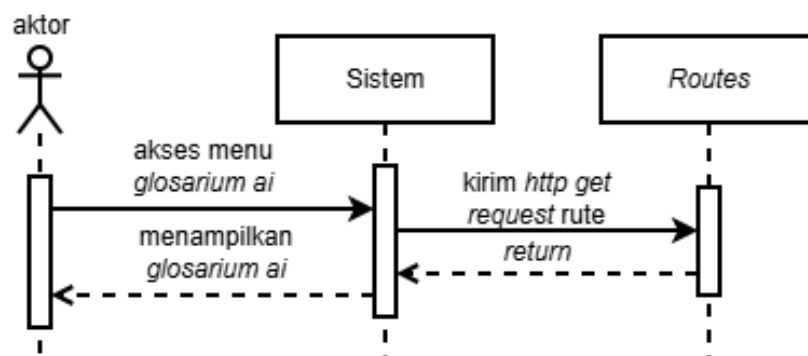


Gambar 5. 51 *Activity Diagram* Membaca Glosarium AI
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 51, diagram aktivitas tersebut memetakan urutan logika interaksi antara aktor *Visitor & Administrator* dengan komponen sistem pada modul kamus referensi edukatif aplikasi. Aliran aktivitas dikendalikan dari simpul awal (*initial state*) di sisi sistem yang merespons rute masuk menu glosarium dengan memproses penataan data dokumen teoretis secara terstruktur. Pihak *Sistem* mengeksekusi instruksi pemuatan kode antarmuka untuk memproses tampilan *widget* istilah yang mencakup tiga poros keilmuan utama, yaitu untuk domain astronomi, domain kecerdasan buatan, serta arsitektur sistem informasi SINEO. Hasil kompilasi jajaran kartu definisi komponen ilmiah tersebut kemudian dikirimkan dan disajikan ke sisi pengguna dalam bentuk tampilan menu glosarium yang terintegrasi rapi. Aliran diagram aktivitas ini berjalan secara linear dan independen tanpa melibatkan *input* atau percabangan fungsional lanjutan, sehingga alur kerja langsung bermuara pada

simpul penutup atau *final state* tepat setelah seluruh ringkasan istilah lintas disiplin ilmu tersebut sukses dirender seutuhnya di layar peramban web pengguna.

2) *Sequence diagram* membaca glosarium AI

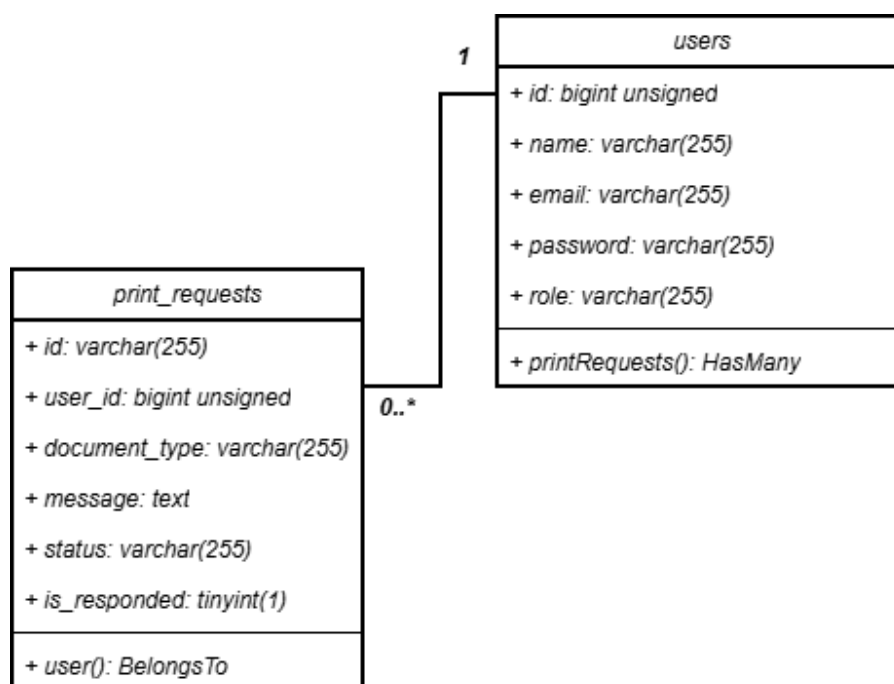


Gambar 5. 52 *Sequence Diagram* Membaca Glosarium AI
(Sumber: Data Pribadi)

Berdasarkan alur fungsional linear yang dipetakan pada Gambar 5. 52, diagram urutan menggambarkan penyajian dokumen referensi modul edukasi istilah ilmiah secara instan kepada aktor. Skenario operasional diawali saat aktor mengakses menu Glosarium AI untuk memeriksa kamus istilah astronomi, kecerdasan buatan, dan arsitektur sistem, di mana komponen antarmuka *glosarium.blade.php* meneruskan permintaan rute langsung menuju *routes* berkas tampilan *glosarium.blade.php* untuk langsung disajikan dalam bentuk tumpukan kartu visual teraktif terformat rata kanan-kiri (*justify*).

B. *Class diagram*

Di samping melakukan dekonstruksi aliran aktivitas melalui diagram aktivitas, fase *Sprint* pada kerangka kerja *Scrum* ini juga menetapkan struktur statis sistem melalui penyusunan *class diagram*. Pemodelan struktural ini bertujuan untuk memetakan secara eksplisit seluruh kelas objek komponen model dan representasi data yang beroperasi di dalam arsitektur sistem *monitoring* SINEO, lengkap dengan spesifikasi visibilitas atribut murni (berbasis tipe data *riil* pada berkas kueri SQL), metode fungsi operasional (*methods*), serta derajat hubungan keterikatan (*multiplicity*) antar-entitas di dalam lingkungan *framework* Laravel 12.



Gambar 5. 53 Class Diagram Print Request SINEO
(Sumber: Data Pribadi)

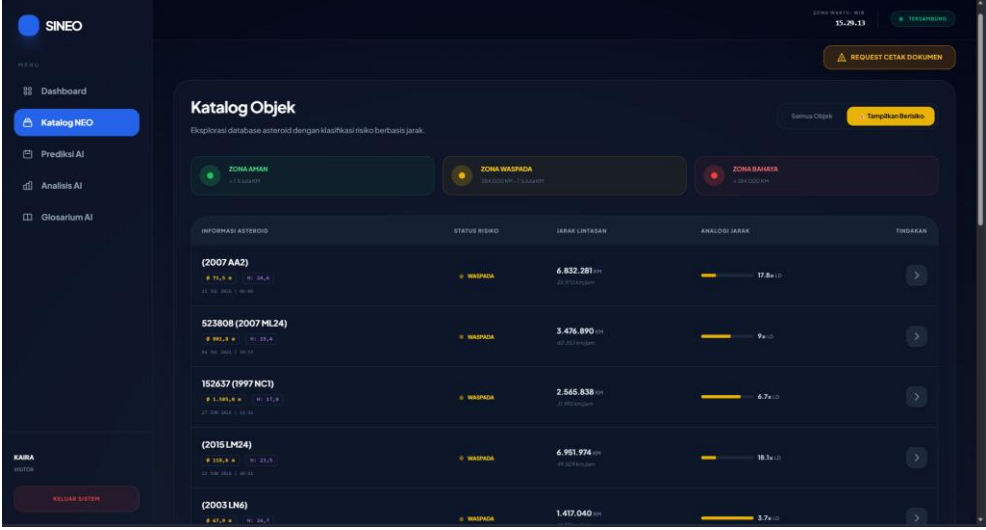
Berdasarkan manifestasi struktur diagram kelas pada Gambar 5. 53, diagram kelas ini memetakan rancangan pemrograman berorientasi objek yang mengendalikan siklus pengajuan, pencetakan dokumen katalog lengkap (*Admin*) pada *platform* SINEO. Arsitektur diagram ini menempatkan kelas pengontrol *PrintRequestController* sebagai komponen pusat yang bertindak sebagai pengendali utama aliran data. Di dalam kelas pengontrol ini, didefinisikan rentetan metode publik yang meliputi fungsi *store()* untuk menangani penyimpanan berkas permohonan baru dari entitas *Visitor*, serta fungsi tindakan mutlak berupa *exportPdf()* untuk mengeksekusi penarikan dokumen secara murni melalui instruksi tekan ekspor *file* PDF pada menu Katalog NEO dan Prediksi AI oleh *Admin*, di mana sistem langsung merespons secara hibrida dengan memicu fungsi menerima *respons* dan memproses *layout* HTML untuk masing-masing menu, menjadi berkas *file* PDF siap unduh berukuran kertas A4. Kelas pengontrol ini dirancang mewarisi (*inheritance relationship*) sifat dasar dari kelas komponen inti *framework* Laravel, yaitu kelas *controller*.

Guna menjamin validitas dan keamanan muatan data yang dikirimkan melalui *form* antarmuka, kelas *PrintRequestController* memiliki hubungan ketergantungan (*dependency relation*) yang kuat terhadap kelas utilitas *PrintRequest*. Kelas *PrintRequest* merupakan representasi model *Eloquent* objek entitas fisik tabel *print_requests* di dalam *database* MySQL, yang mengenkapsulasi properti data operasional seperti atribut *id*, kunci tamu *user_id*, string dokumen *document_type*, penanda status *status*, deskripsi alasan penolakan *rejection_reason*, serta jejak waktu *created_at* dan *updated_at*. Dari sudut pandang relasi basis data relasional *phpMyAdmin*, hubungan kelas model *User* terhadap kelas model *PrintRequest* memiliki nilai kardinalitas mutlak *one-to-many* (*1 ke 0..**). Penentuan derajat *multiplicity* ini menegaskan bahwa satu baris rekaman pada kelas model *User* dapat memiliki hubungan kepemilikan terhadap banyak (*0..**) baris rekaman dokumen pengajuan cetak, sedangkan satu (*1*) instansiasi baris data pada kelas model *PrintRequest* wajib terikat secara independen pada satu identitas pengguna yang valid guna menjaga akuntabilitas *log* audit keamanan sistem.

5.1.5.3.3. *Implementation*

Sistem antarmuka ini bertindak sebagai media interaksi utama (*front-end*) yang menyajikan hasil pemrosesan data historis dari *database* MySQL serta hasil inferensi prediksi dari model biner *Dual-Stacked* RNN (*neo_rnn_model.keras*) secara dinamis kepada pengguna. Secara fungsional, struktur pengembangan komponen antarmuka *monitoring* SINEO dirancang secara modular dan dibagi menjadi beberapa halaman utama yang didokumentasikan sebagai berikut:

a. Halaman katalog data asteroid



Katalog Objek
Eksplorasi database asteroid dengan klasifikasi risiko berbasis jarak.

RECENT: Dashboard, **Katalog NEO**, Prediksi AI, Analisis AI, Glosarium AI

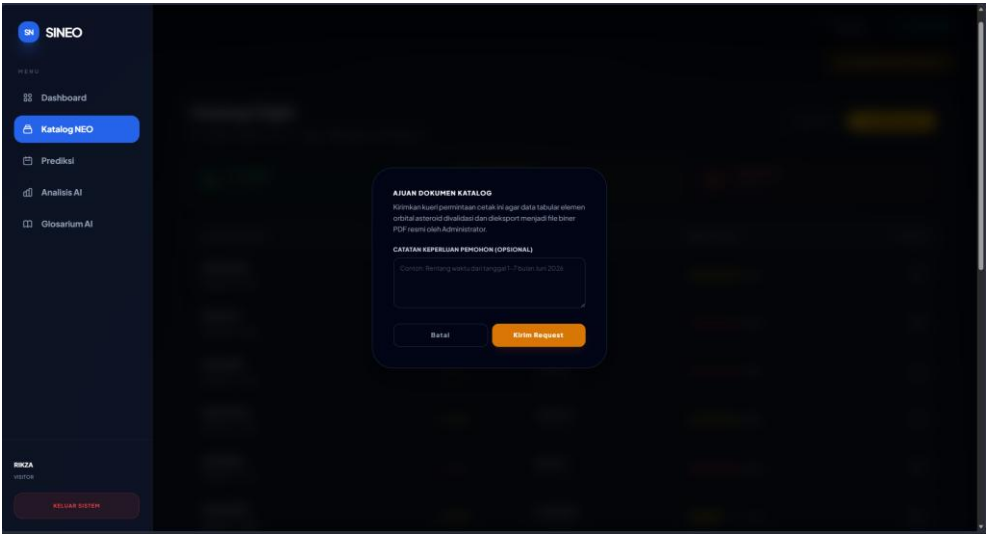
KARMA: VISITOR, KELUAR SISTEM

15.26.13 15:26:13 15:26:13 REQUEST CETAK DOKUMEN

Semua Objek Tampilkan Berlebih

INFORMASI ASTEROID	STATUS RISIKO	JARAK LINTASAN	ANALOGI JARAK	TINDAKAN
(2007 AA2) 17.1.18 - 17.1.18 17.1.18 - 17.1.18	WASPADA	6.832.281 km	17.1.18	>
523808 (2007 ML24) 17.1.18 - 17.1.18 17.1.18 - 17.1.18	WASPADA	3.476.890 km	17.1.18	>
152637 (1997 NC1) 17.1.18 - 17.1.18 17.1.18 - 17.1.18	WASPADA	2.565.838 km	17.1.18	>
(2015 LM24) 17.1.18 - 17.1.18 17.1.18 - 17.1.18	WASPADA	6.951.974 km	17.1.18	>
(2003 LN4) 17.1.18 - 17.1.18 17.1.18 - 17.1.18	WASPADA	1.417.040 km	17.1.18	>

Gambar 5. 54 Antarmuka Tabel Manajemen Katalog Data Historis Parameter Orbit *Near-Earth Objects (Visitor)*
(Sumber: Data Pribadi)

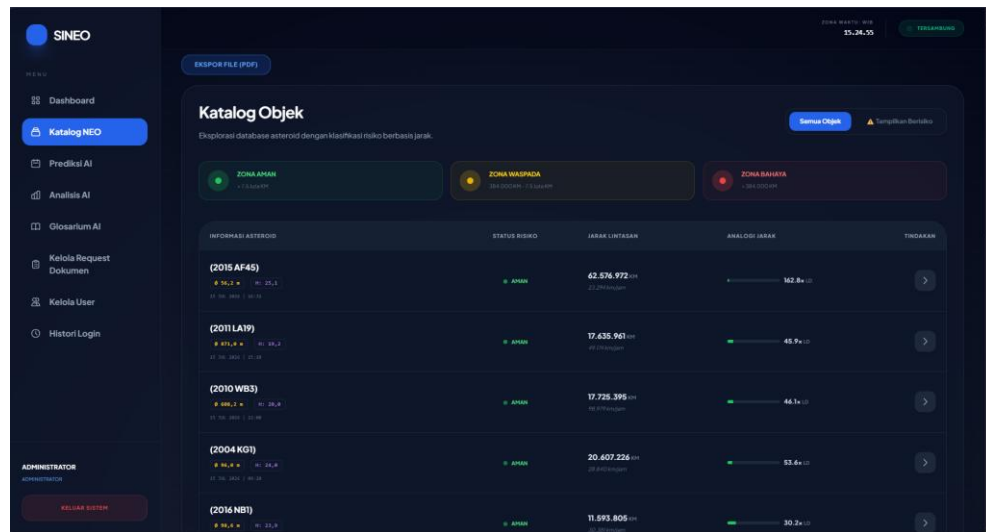


AJUAN DOKUMEN KATALOG
Kirimkan surat permintaan cetak ke agar data tabel elemen orbital asteroid divalidasi dan diekspor menjadi file berformat PDF resmi oleh Administrator.

CATATAN KEPERLUAN PENCIHOK (OPSIONAL)
Contoh: Berlangganan dari tanggal 1 Januari 2023.

Batal Kirim Request

Gambar 5. 55 Antarmuka *Modal Form Request* Katalog Data Historis Parameter Orbit *Near-Earth Objects (Visitor)*
(Sumber: Data Pribadi)



Gambar 5. 56 Antarmuka Tabel Manajemen Katalog Data Historis Parameter Orbit *Near-Earth Objects* (Admin)
(Sumber: Data Pribadi)

ID	NAMA ASTEROID	TANGGAL DETEKSI	JARAK LINTASAN	KECEPATAN	STATUS RISIKO
1	(2015 AF45)	15 Feb 2020	6.801.171 km	41.237 km/h	NEAPASPA
2	(2011 LA19)	15 Feb 2020	63.851.842 km	89.524 km/h	AMANK
3	(2012 UN95)	15 Feb 2020	20.881.028 km	18.232 km/h	AMANK
4	(1998 JF30)	15 Feb 2020	27.285.425 km	11.525 km/h	AMANK
5	(2017 KAS)	15 Feb 2020	51.712.839 km	65.467 km/h	AMANK
6	(2004 DP1)	15 Feb 2020	21.141.144 km	63.033 km/h	AMANK
7	(2004 DW4)	15 Feb 2020	16.715.522 km	44.852 km/h	AMANK
8	(2003 BR9)	15 Feb 2020	4.485.215 km	28.948 km/h	NEAPASPA
9	(2004 EL5)	15 Feb 2020	21.482.014 km	35.017 km/h	AMANK
10	(2009 DA10)	15 Feb 2020	2.813.389 km	32.021 km/h	NEAPASPA
11	(2004 C26)	15 Feb 2020	4.485.215 km	28.948 km/h	NEAPASPA
12	(2019 VNS)	15 Feb 2020	20.881.028 km	18.232 km/h	AMANK
13	(2009 C10)	15 Feb 2020	273.887 km	18.468 km/h	BAHAYA
14	(2009 BC11)	15 Feb 2020	4.485.215 km	28.948 km/h	NEAPASPA
15	(2019 DQ10)	15 Feb 2020	22.481.388 km	61.027 km/h	AMANK
16	(2007 CT26)	15 Feb 2020	15.513.153 km	28.858 km/h	AMANK
17	(2002 A1)	15 Feb 2020	47.281.581 km	19.225 km/h	AMANK
18	(2001 SC35)	15 Feb 2020	16.028.416 km	18.038 km/h	AMANK
19	(19222 (2001 TC4))	15 Feb 2020	16.028.422 km	18.038 km/h	AMANK
20	(2004 BT10)	15 Feb 2020	26.960.289 km	25.165 km/h	AMANK
21	(2008 BK2)	15 Feb 2020	1.482.385 km	25.412 km/h	NEAPASPA
22	(2014 CZ20)	15 Feb 2020	45.052.018 km	65.462 km/h	AMANK
23	(2004 DW4)	15 Feb 2020	1.346.027 km	40.225 km/h	AMANK
24	(2009 DV10)	15 Feb 2020	20.048.389 km	18.524 km/h	AMANK
25	(2004 BW4)	15 Feb 2020	2.961.487 km	21.853 km/h	NEAPASPA
26	(2005 DN1)	15 Feb 2020	461.323 km	28.455 km/h	BAHAYA
27	(2004 C26)	15 Feb 2020	4.485.215 km	28.948 km/h	AMANK
28	(2003 WF3)	15 Feb 2020	84.762.815 km	45.133 km/h	AMANK
29	(2001 DP2)	15 Feb 2020	60.881.228 km	84.423 km/h	AMANK
30	(2009 BT10)	15 Feb 2020	460.323 km	28.455 km/h	BAHAYA
31	(2004 GG1)	17 Feb 2020	24.884.423 km	25.342 km/h	AMANK
32	(2004 DW4)	17 Feb 2020	1.346.027 km	40.225 km/h	BAHAYA
33	(2002 A1)	17 Feb 2020	47.281.581 km	19.225 km/h	AMANK
34	(2015 VU1)	17 Feb 2020	70.371.352 km	100.027 km/h	AMANK
35	(2004 DW4)	17 Feb 2020	1.346.027 km	40.225 km/h	NEAPASPA

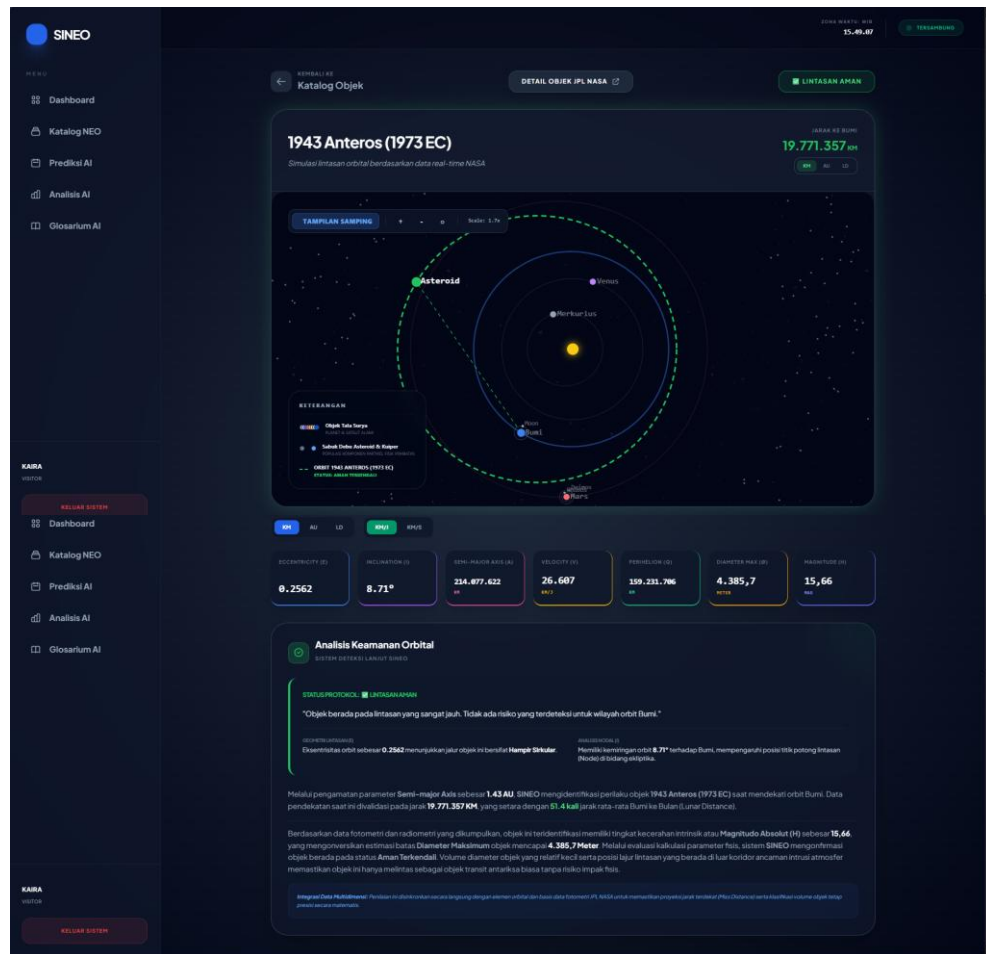
Gambar 5. 57 Tampilan Hasil Ekspor Dokumen *File* Katalog Data Historis Parameter Orbit *Near-Earth Objects*
(Sumber: Data Pribadi)

Halaman *catalog.blade.php* dikembangkan untuk menyajikan visualisasi data tabular dari seluruh objek asteroid yang telah ditarik secara berkala dari server NASA. Antarmuka ini dilengkapi dengan fitur kartu metrik informasi (*widget cards*) pada bagian atas halaman guna merangkum statistik data sains secara interaktif, serta fitur paginasi (*pagination*) bawaan Laravel 12 guna menjamin efisiensi pemuatan memori lokal. Tampilan tabel ini menyajikan representasi fisik

dari kolom *database neo_objects* secara rapi, meliputi nama objek, diameter ($\emptyset$), magnitudo (H), tingkat bahaya potensial, kecepatan relatif (V), dan estimasi jarak meleset (*miss distance*).

Bagi sisi *Visitor*, berkas antarmuka ini menyediakan elemen komponen modal pengajuan khusus yang memungkinkan pengguna mengirimkan permohonan cetak dokumen biner katalog menuju antrean pangkalan data. Sementara bagi sisi *Admin*, antarmuka ini telah diintegrasikan secara langsung dengan tombol ekspor PDF mandiri yang terhubung menuju fungsi *approve* pada *RequestManagementController*. Saat tombol tersebut dieksekusi oleh *Admin* demi kebutuhan internal seperti presentasi data hasil sinkronisasi NASA, sistem secara otomatis akan memicu optimalisasi performa server latar belakang, melakukan penarikan data *riil* dari *database*, mengelompokkannya secara terstruktur per bulan menggunakan *Carbon*, serta memanfaatkan objek *DomPDF* melalui *template katalog_otorisasi_pdf* untuk menghasilkan *file* biner PDF katalog lengkap yang langsung terunduh secara otomatis ke dalam penyimpanan lokal.

b. Halaman detail spesifikasi objek



Gambar 5. 58 Antarmuka Detail Parameter Orbit *Near-Earth Objects* (Sumber: Data Pribadi)

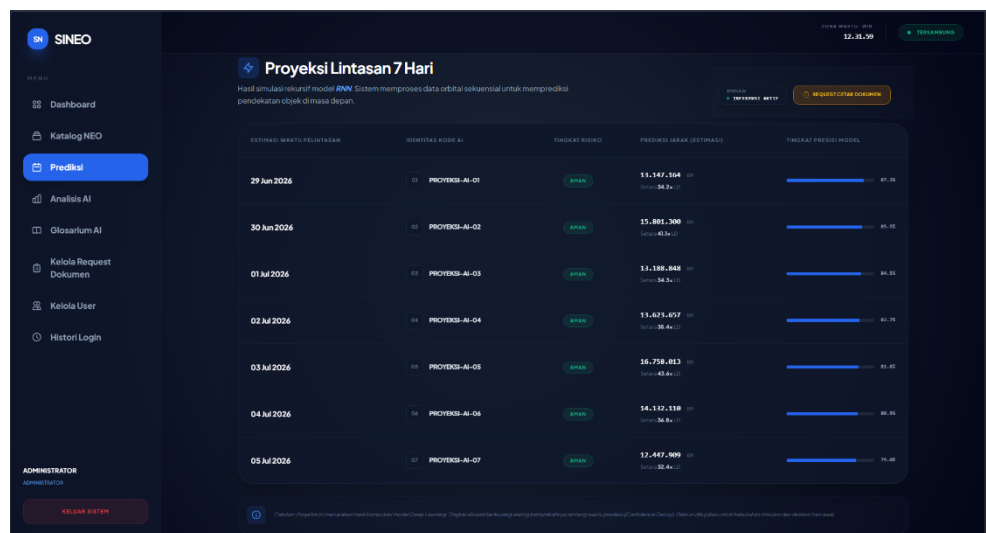
Halaman *detail_object.blade.php* dikembangkan khusus untuk menampilkan parameter dan orbital secara spesifik dan mendalam dari satu objek asteroid tertentu yang dipilih oleh pengguna melalui halaman katalog. Halaman ini memetakan seluruh atribut teknis yang bersumber dari *database* MySQL secara lengkap, termasuk nomor identifikasi unik (*neo_reference_id*), parameter fisis (e , i , a , V , q , $\emptyset$, H) untuk kebutuhan analisis mendalam karakteristik objek secara individual.

Di dalam kode program berkas ini, diimplementasikan sebuah sistem logika pengondisian warna tema berbasis tingkat risiko objek (*Traffic Light System*). Penentuan tingkat risiko ini tidak lagi sekadar mengukur parameter tunggal, melainkan mengintegrasikan status *Potentially Hazardous Asteroid* (PHA) orisinal

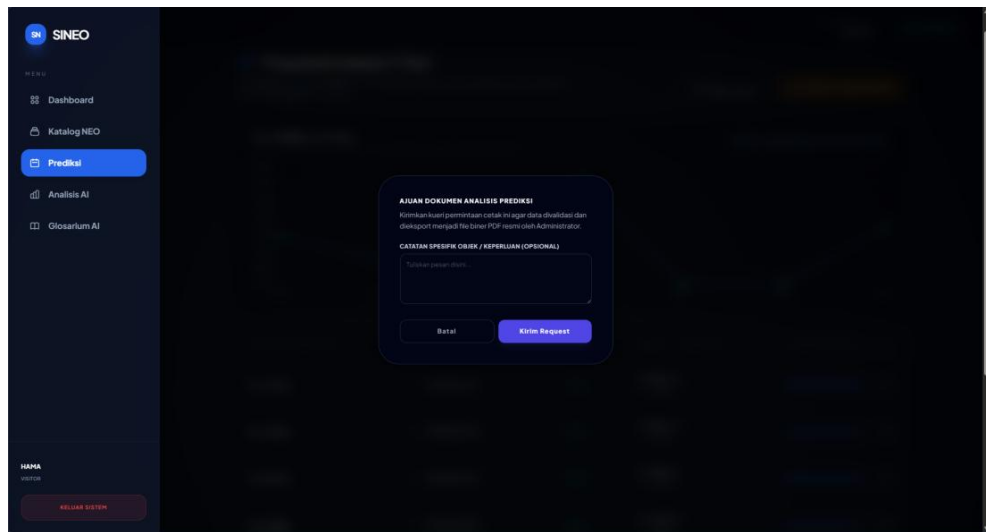
dari basis data NASA dengan kalkulasi jarak lintasan terdekat (*miss_distance_km*) serta dimensi fisis objek (*estimated_diameter_max*).

Sistem secara otomatis menetapkan variabel $\$themeColor = "red"$ untuk memicu peringatan berwarna merah membara (*shadow glow effect*) berlabel "Bahaya Kritis" hanya jika suatu objek tervalidasi sebagai PHA oleh NASA, melintas di dalam koridor proksimitas ekstrem memotong jarak orbit Bulan (< 384.400 KM, serta memiliki dimensi masif berkategori monster dengan diameter lebih dari 1 KM (1.000 meter). Sebaliknya, jika objek berstatus PHA secara umum atau merupakan objek non-PHA berdiameter lebih dari 50 meter yang memotong jarak orbit Bulan (384.400 KM/ 1 LD hingga $7.500.000$ KM), sistem akan mengalihkan warna tema menjadi kuning waspada (*yellow approach warning*). Di luar kondisi kritis tersebut, warna tema secara dinamis bergeser menjadi *emerald* (hijau) yang menandakan objek berada pada lintasan aman terkendali.

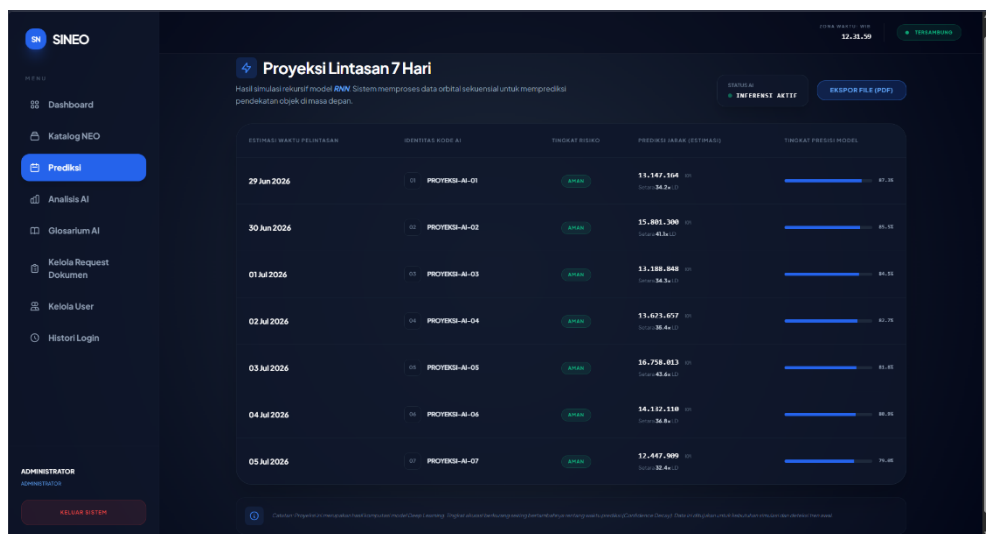
c. Halaman visualisasi grafik prediksi AI



Gambar 5. 59 Antarmuka Proyeksi Prediksi AI Berbasis RNN (*Visitor*)
(Sumber: Data Pribadi)



Gambar 5. 60 Antarmuka *Modal Form Request* Proyeksi Prediksi Jarak Berbasis RNN (*Visitor*)
(Sumber: Data Pribadi)



Gambar 5. 61 Antarmuka Proyeksi Prediksi Jarak Berbasis RNN (*Admin*)
(Sumber: Data Pribadi)

TANGGAL PELANTIKAN	IDENTITAS KODE AI	STATUS	PROJEKSI JARAK (ESTIMASI)	MODAL PREDIKSI MODEL
26 Jun 2026	Proyeksi-A1-01	AMAN	13 181 508 KM Selera 34.34 Jarak Bulan	87.8%
27 Jun 2026	Proyeksi-A1-02	AMAN	17 308 060 KM Selera 45.34 Jarak Bulan	85.8%
28 Jun 2026	Proyeksi-A1-03	AMAN	13 308 040 KM Selera 34.34 Jarak Bulan	84.8%
29 Jun 2026	Proyeksi-A1-04	AMAN	12 982 778 KM Selera 32.34 Jarak Bulan	82.7%
30 Jun 2026	Proyeksi-A1-05	AMAN	12 750 809 KM Selera 31.34 Jarak Bulan	81.8%
01 Jul 2026	Proyeksi-A1-06	AMAN	10 942 961 KM Selera 28.34 Jarak Bulan	80.8%
02 Jul 2026	Proyeksi-A1-07	AMAN	13 498 022 KM Selera 35.34 Jarak Bulan	79.8%

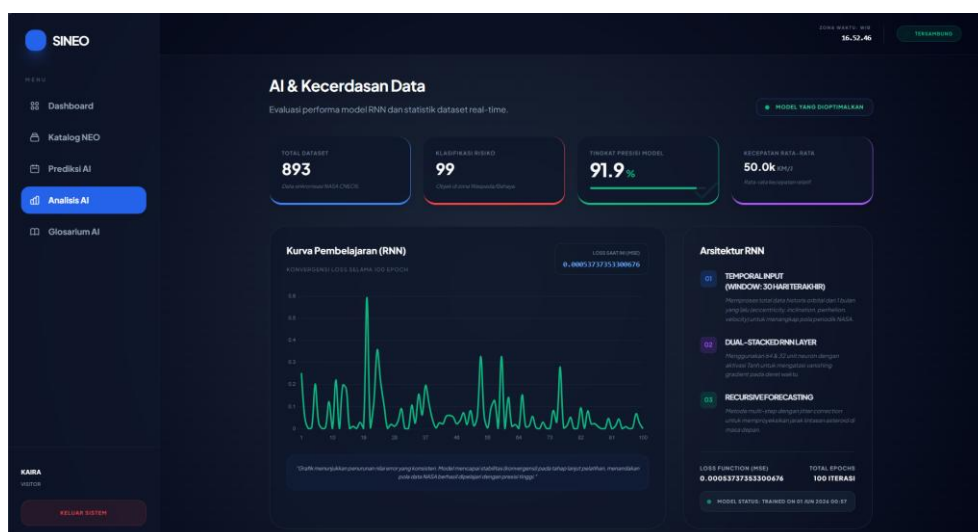
Gambar 5. 62 Tampilan Hasil Ekspor Dokumen *File*
Proyeksi Prediksi Jarak Berbasis RNN
(Sumber: Data Pribadi)

Halaman *prediksi_bulanan.blade.php* dikembangkan untuk memwadah komponen utama dari fungsi peramalan temporal pada sistem *monitoring* koordinat objek astronomi. Berkas antarmuka ini memuat visualisasi grafik garis dinamis (*line chart*) yang menyajikan data tabular ringkas proyeksi lintasan tujuh hari sepekan dan indikator *widget card* pemantau efektivitas konvergensi model kecerdasan buatan. Secara mekanis, skrip di dalam berkas ini membaca keluaran *Recursive Multi-step Forecasting* dari berkas hasil komputasi mesin inferensi *Python* yang disimpan di dalam memori penyimpanan *cache* lokal. Kehadiran halaman grafik prediksi ini memudahkan pengguna dalam melakukan analisis spasial temporal serta mitigasi dini terhadap tren pergerakan lintasan objek koordinat orbit di masa mendatang.

Bagi sisi *Visitor*, berkas antarmuka ini menyediakan elemen tombol operasional guna memicu kemunculan modal pengajuan dokumen analisis prediksi menuju baris antrean pangkalan data MySQL. Sementara bagi sisi *Admin*, antarmuka ini dirancang secara simetris menempatkan tombol ekspor PDF mandiri tepat di sebelah indikator status AI (inferensi aktif) guna memicu eksekusi fungsi *approve* pada kelas *RequestManagementController*. Saat tombol tersebut dieksekusi oleh *Admin* untuk memenuhi kebutuhan internal seperti presentasi data analisis, sistem

secara otomatis akan memicu optimalisasi batas waktu performa server latar belakang, melakukan pembacaan data ramalan sekuensial beserta metrik akurasi latih model RNN dari berkas log pelatihan JSON, serta mengompilasikan seluruh variabel laporan terstruktur tersebut menggunakan objek *DomPDF* melalui berkas templat *prediksi_otorisasi_pdf* untuk menghasilkan file biner PDF proyeksi lintasan resmi yang langsung terunduh secara otomatis ke dalam media penyimpanan lokal.

d. Halaman analisis AI



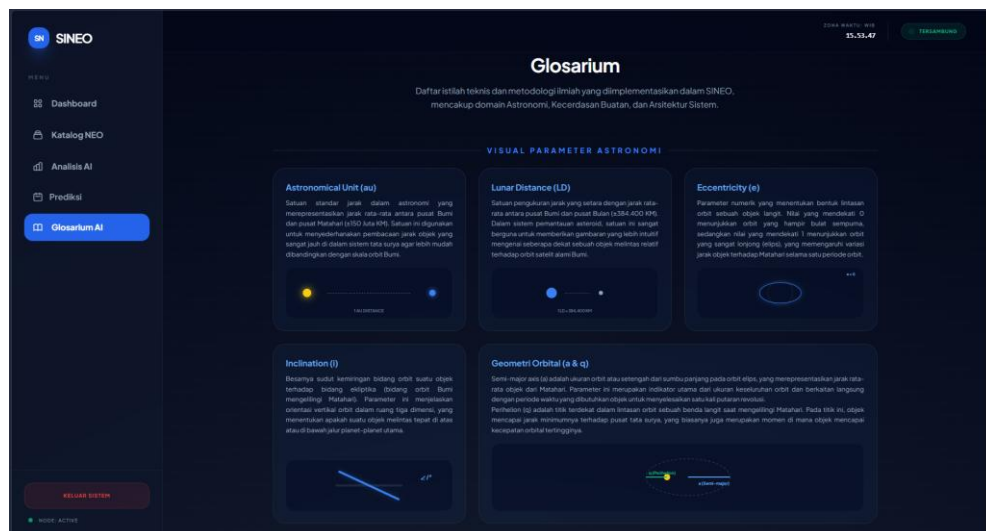
Gambar 5. 63 Antarmuka *Monitoring* Riwayat Penurunan *Loss (MSE)* Pelatihan Model RNN & Statistik Data *Real-Time* (Sumber: Data Pribadi)

Halaman *analisis.blade.php* merupakan halaman khusus yang dikembangkan untuk memonitor riwayat performa kecerdasan buatan. Antarmuka ini membaca berkas log biner *training_log.json* hasil pemrosesan eksekusi *Python* secara berkala. Halaman ini menyajikan indikator total data terintegrasi hasil sinkronisasi API NASA Portal (sejumlah total data rekaman data objek), jumlah objek berbahaya (*potentially hazardous objects*), serta *widget* ringkasan status operasional sistem, analisis RNN dan grafik penurunan nilai kesalahan *Mean Squared Error (MSE)* dari *epoch* 1 hingga *epoch* 100 secara transparan, serta mengonversikan nilai akhir

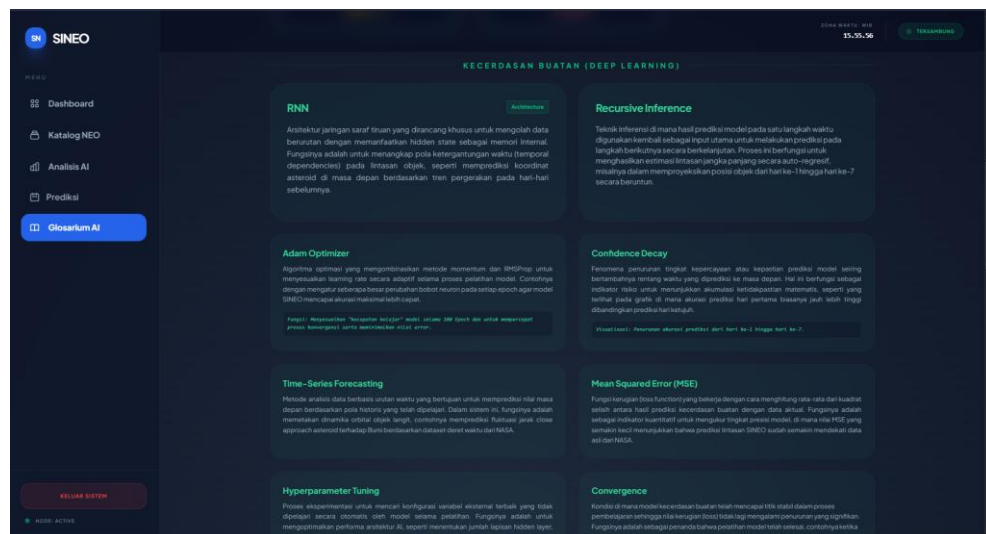
loss sebesar 0.000537 ke dalam visualisasi persentase akurasi model RNN (91.9%) untuk kedekatan prediksi pada *normalized space* bagi pengguna.

Di dalam skrip ini, konfigurasi *Chart.js* diatur secara spesifik untuk memantau nilai fluktuasi *Loss History*. Komponen titik koordinat grafik dimatikan (*pointRadius: 0*) dan hanya dimunculkan saat kursor melakukan aksi melayang di atas area grafik (*pointHoverRadius: 6*) untuk menjaga kinerja peramban web tetap ringan saat memuat ratusan data titik *epoch*. Modul ini menjadi bukti empiris keterbukaan performa kecerdasan data yang tertanam di dalam sistem SINEO.

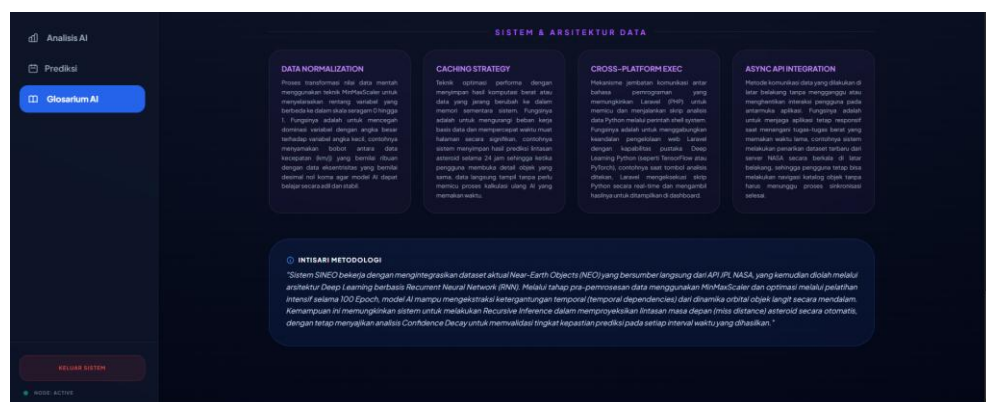
e. Halaman glosarium AI



Gambar 5. 64 Antarmuka Glosarium AI SINEO (Astronomi)
(Sumber: Data Pribadi)



Gambar 5. 65 Antarmuka Glosarium AI SINEO (Kecerdasan Buatan)
(Sumber: Data Pribadi)



Gambar 5. 66 Antarmuka Glosarium AI SINEO (Arsitektur Sistem)
(Sumber: Data Pribadi)

Halaman terakhir adalah *glosarium.blade.php* yang berfungsi sebagai modul edukasi sekunder di dalam sistem SINEO. Berkas ini merender jajaran kartu definisi yang memuat daftar penjelasan istilah ilmiah, simbol matematika, dan satuan astronomi yang digunakan di dalam sistem informasi (seperti *Astronomical Unit* (AU), Eksentrisitas, Inklinalasi, *Perihelion*, *Sliding Window*, hingga *Epoch*). Komponen UI ini bertujuan untuk menjembatani kesenjangan pemahaman pengguna umum maupun akademisi lintas disiplin ilmu agar mempermudah pemetaan interpretasi data fisis objek dekat Bumi yang disajikan oleh sistem *monitoring* SINEO.

5.1.5.3.4. Review

Tahapan *Sprint Review* pada akhir *sprint* ketiga ini ditujukan untuk mengevaluasi fungsionalitas dan akurasi visualisasi data dari modul Katalog NEO, Detail Objek (*Traffic Light System*), Prediksi AI, Analisis AI, hingga Glosarium AI menggunakan metode *Black Box Testing*. Pengujian *Black Box* dieksekusi guna memvalidasi bahwa setiap respons antarmuka reaktif dan pengolahan data di *server-side* berjalan sesuai dengan kriteria masukan dan keluaran (*input-output*) yang diharapkan pengguna, tanpa menguji struktur kode internal *controller*. Proses peninjauan dilakukan melalui mekanisme *Client Validation Loop* secara terbuka melalui pengujian fungsionalitas (*Black Box Testing*) langsung oleh pengguna umum. Fokus utama evaluasi tertuju pada efisiensi renderan data tabular berskala besar yang dikendalikan fungsi paginasi Laravel 12, ketepatan pengondisian warna indikator risiko spasial *Traffic Light System* (TLS) pada modul *Detail Objek*, hingga kepresisian penguraian log biner *training_log.json* menjadi grafik konvergensi *Loss History* berbasis *Chart.js*.

Guna membuktikan ketercapaian fungsionalitas visualisasi data, pengondisian risiko spasial, dan pe-renderan grafik analitik pada *Sprint 3* secara empiris, dilakukan pengujian masukan dan keluaran (*input-output*) menggunakan metode *Black Box Testing*. Rangkuman hasil pengujian *Black Box* untuk seluruh modul fitur pada *Sprint 3* disajikan secara terperinci pada Tabel 5. 15 di bawah ini:

Tabel 5. 15 Hasil Pengujian *Black Box Testing* Sprint 3

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
1.	Navigasi tabel katalog data (<i>katalog.blade.php</i>)	Membuka halaman Katalog NEO dengan menekan menu pada <i>sidebar</i> .	Menampilkan informasi asteroid, status, jarak, tombol detail objek, tombol <i>filter hazardous</i> menyaring baris tabel dengan status waspada & bahaya, dan tautan. Membagi baris data per bulan secara rapi pada hasil <i>file</i> PDF.	Data tersaji secara runtut berdasar <i>close_approach_date</i> , <i>filter hazardous</i> asteroid berfungsi responsif, dan perpindahan halaman berjalan lancar tanpa memutus koneksi data.	Sesuai / Valid	29 April 2026
2.	Pengajuan kueri cetak laporan digital oleh aktor <i>Visitor</i>	<i>Visitor</i> mengeklik tombol " <i>Request Cetak Dokumen</i> " pada menu katalog.	Jendela modal <i>form</i> pengisian catatan terbuka dan menyimpan ajuan berstatus <i>pending</i> ke <i>database</i> MySQL.	Tanda notifikasi pesan berhasil terkirim muncul di layar peramban.	Sesuai / Valid	29 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
3.	Penapisan <i>filter</i> risiko spasial objek dekat Bumi	Mengeklik komponen tombol penapisan status waspada/bahaya.	Sistem menyaring tumpukan data dan mengisolasi objek berdasarkan klasifikasi tingkat ancaman orbit (waspada & bahaya).	Tampilan baris data tabular menyusut secara dinamis sesuai <i>filter</i> .	Sesuai / Valid	29 April 2026
4.	Logika penilaian risiko spasial (<i>Traffic Light System</i>) (<i>detail_object.blade.php</i>)	Mengeklik tombol aksi "Detail" dengan <i>icon</i> tombol (>) pada baris asteroid untuk melihat lebih detail tentang objek asteroid yang ingin dilihat.	Pengontrol mengarahkan rute menuju <i>/detail/{id}</i> , visual 2D, menghitung jarak <i>close approach</i> (km, AU, & LD), kecepatan (km/s & km/h) tombol detail fisis objek, analisis keamanan orbital dan otomatis menyesuaikan tema warna berdasarkan status jaraknya.	Halaman <i>detail</i> menampilkan parameter fisis objek secara lengkap (<i>e, i, q, a, Ø, H</i>), kecepatan <i>V</i> (km/s & km/h), menampilkan jarak objek ke bumi (km, AU, & LD), mengarahkan detail objek ke JPL NASA, analisis keamanan orbital dan visual 2D dengan keterangannya.	Sesuai / Valid	29 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
5.	Prediksi berbasis langkah waktu	Mengakses halaman Prediksi AI selama 7 Hari untuk memantau simulasi rekursif AI masa depan.	Objek <i>Chart.js</i> memetakan <i>array</i> koordinat <i>future_predictions</i> dari model RNN ke dalam sumbu kronologis, menampilkan kurva volatilitas jarak lintasan, data tabular metrik estimasi waktu pelintasan, identitas, status, nilai batas <i>confidence score</i> , dan menyusun tabel ringkasan proyeksi jarak.	Antarmuka berhasil menyajikan kurva volatilitas jarak lintasan, data tabular tren fluktuasi kontinu masa depan hasil komputasi <i>Recursive Multi-step Forecasting</i> model RNN secara dinamis.	Sesuai / Valid	30 April 2026
6.	Pengajuan kueri cetak laporan digital oleh aktor <i>Visitor</i>	<i>Visitor</i> mengeklik tombol " <i>Request Cetak Dokumen</i> " pada menu prediksi.	Jendela modal <i>form</i> pengisian catatan terbuka dan menyimpanajuan berstatus <i>pending</i> ke <i>database</i> MySQL.	Tanda notifikasi pesan berhasil terkirim muncul di layar peramban.	Sesuai / Valid	30 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
7.	Modul ekspor dokumen lintas media (katalog & prediksi)	Mengeklik tombol fungsi "Ekspor <i>File</i> PDF" pada halaman Katalog NEO dan Prediksi.	Sistem memicu fungsi <i>exportPdf</i> dan mengeksekusi instruksi CSS untuk tata letak halaman menjadi format siap cetak PDF yang rapi dan informatif.	Berhasil menyajikan <i>file</i> dokumen yang sesuai kebutuhan dengan format PDF yang rapi dan informatif dan pas di kertas A4.	Sesuai / Valid	29-30 April 2026
8.	Statistik <i>dataset</i> & grafik <i>loss</i> RNN (<i>analisis.blade.php</i>)	Membuka halaman Analisis AI setelah proses pelatihan model selesai.	Komponen <i>Chart.js</i> mengurai <i>file training_log.json</i> , statistik <i>dataset</i> (total, risiko, & $\bar{v}$), menampilkan grafik konvergensi <i>loss</i> hingga <i>epoch</i> 100, dan <i>widget</i> akurasi model RNN hasil rumus RMSE yang dikalibrasi.	Menampilkan statistik <i>dataset</i> (total, risiko, & $\bar{v}$), grafik penurunan <i>Loss</i> (MSE) murni <i>ter-render</i> dengan sempurna hingga menyentuh nilai <i>error</i> akhir 0.000537 dengan visualisasi akurasi 91.9% (skala normalisasi) secara responsif.	Sesuai / Valid	30 April 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
9.	Modul glosarium AI & intisari metodologi	Mengakses menu Glosarium AI untuk memeriksa dokumentasi parameter ilmiah.	Sistem memuat tumpukan komponen kartu dinamis berisi definisi visual elemen astronomi (e , i , q , a , V , dll), kecerdasan buatan (RNN, dll), arsitektur sistem, serta blok teks berformat <i>justify</i> .	Seluruh definisi tersaji secara terstruktur (Astronomi, Kecerdasan Buatan, dan Arsitektur Sistem), mempermudah pengguna mengetahui istilah-istilah yang kurang awam pada sistem SINEO.	Sesuai / Valid	30 April 2026

(Sumber: Data Pribadi)

Hasil peninjauan kelayakan produk berdasarkan standar penulisan tugas akhir dirangkum ke dalam Tabel 5. 16 berikut:

Tabel 5. 16 Hasil *Review Sprint 3*

No.	Kebutuhan Sistem	Keterangan	Status
1.	Modul Katalog NEO dan Paginasi	Sistem sukses menyajikan data tabular berskala masif dengan kueri teroptimasi, serta berhasil memotong latensi pemuatan halaman menggunakan fungsi paginasi bawaan Laravel 12.	Valid
2.	Modul Detail Objek & <i>Traffic Light System</i>	Logika pengkondisian variabel <i>\$themeColor</i> di sisi <i>server-side</i> sukses menyeleksi <i>threshold</i> jarak lintasan (<i>miss_distance_km</i>) untuk merender warna peringatan risiko (Merah, Kuning, Hijau) secara tepat.	Valid
3.	Modul Halaman Prediksi AI	Berhasil menampilkan prediksi sepekan pada data tabular berdasarkan hasil <i>Recursive Multi-step Forecasting</i> dari skrip <i>Python</i> yang otomatis menampilkan data terbaru setiap kali Admin melakukan <i>sync-data</i> NASA.	Valid (Revisi)

No.	Kebutuhan Sistem	Keterangan	Status
4.	Modul Analisis AI	Sistem sukses mengurai <i>log</i> biner <i>training_log.json</i> , mengubah nilai <i>Mean Squared Error</i> (MSE) menjadi grafik konvergensi <i>Loss History</i> .	Valid
5.	Modul Glosarium AI	Berhasil menyajikan instrumen kamus istilah literasi astronomi, kecerdasan buatan, dan arsitektur sistem.	Valid

(Sumber: Data Pribadi)

Berdasarkan hasil peninjauan pada Tabel 5. 16, Pada blok fitur Modul Prediksi AI terdapat revisi penambahan pada UI. Dan untuk blok fitur lainnya pada *Sprint 3* dinyatakan memenuhi kualifikasi. Catatan evaluasi dari hasil *review* ini menyepakati pada *item backlog* Modul Prediksi AI disarankan menambahkan komponen *Chart.js* yang *me-render* kurva proyeksi lintasan runtun waktu 7 hari ke depan secara dinamis sebelum bagian data tabular prediksi, di mana objek yang diprediksi menyesuaikan otomatis dengan status risikonya (eksekusi konsep *Traffic Light System*) dan dapat memicu *pointHoverRadius* saat kursor berinteraksi dengan titiok objek pada kurva.

5.1.5.3.5. Retrospective

1. Daily scrum

Pada *Sprint retrospective*, aktivitas *Daily Scrum* difokuskan pada penambahan dan penguatan aspek visualisasi data dan analisis mendalam bagi pengguna pada modul Prediksi AI berdasarkan saran dari pengguna umum. Hambatan teknis yang mencuat dalam siklus ini berkaitan dengan tantangan performa saat merender kurva prediksi sepekan otomatis, namun berhasil diatasi melalui implementasi fungsi pada Laravel 12 yang efisien. Selain itu, tim pengembang berfokus pada integrasi *Traffic Light System* pada kurva prediksi sebagai instrumen edukasi literasi astronomi bagi pengguna awam. Seluruh fitur pada siklus ini telah melalui tahap validasi fungsional untuk memastikan bahwa interpretasi data fisis objek langit tidak mengalami distorsi saat dirender di peramban web.

Tabel 5. 17 Hasil *Daily Scrum Sprint Retrospective*

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
25 Juni 2026	-	-	Penambahan visualisasi kurva prediksi yang dikombinasikan dengan logika <i>Traffic Light System</i> (TLS)

(Sumber: Data Pribadi)

2. Design

A. Activity dan sequence diagram

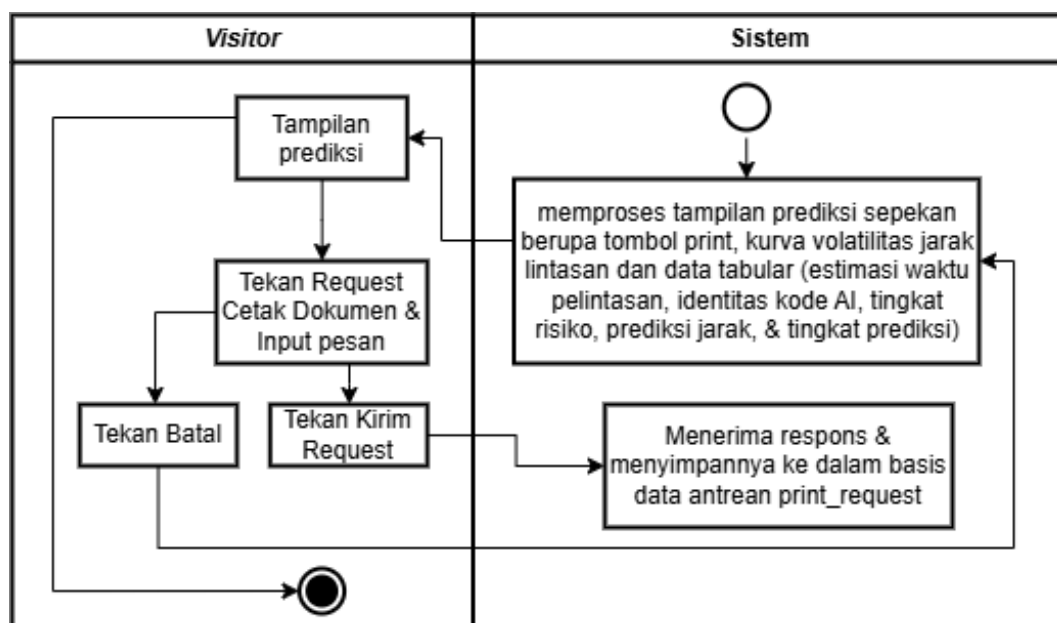
Perancangan *Activity Diagram* untuk pembaruan menu Prediksi AI. Pemodelan aktivitas ini bertujuan untuk memetakan alur kerja (*workflow*) terbaru sesuai dengan saran dari pengguna umum, untuk memperjelas transisi interaksi antara aktor (*front-end UI*) dengan *respons* pengontrol (*back-end*), serta menggambarkan

bagaimana data mengalir secara linear pada setiap fungsi di dalam arsitektur *hybrid environment* SINEO.

Guna melengkapi pemodelan struktural statis, interaksi dinamis berorientasi waktu dan pertukaran pesan (*message passing*) antar-objek di dalam ekosistem SINEO dimodelkan secara eksplisit memanfaatkan *Sequence Diagram*. Pemodelan *behavioral* ini bertujuan untuk menggambarkan urutan kronologis operasi koneksi hibrida (*hybrid connection*) yang melibatkan siklus hidup objek secara utuh. Rangkaian skenario interaksi dipetakan secara sinkron dan linear dengan *Activity Diagram* yang telah dijabarkan sebelumnya, mencakup fungsionalitas di sisi pengguna umum (*Visitor*), hak akses khusus (*Administrator*), hingga otomatisasi komputasi di latar belakang server (*back-end*).

Guna menyajikan dokumentasi sistem yang terstruktur dan komprehensif, urutan pertukaran pesan dinamis ini dijabarkan ke dalam skenario operasi sebagai berikut:

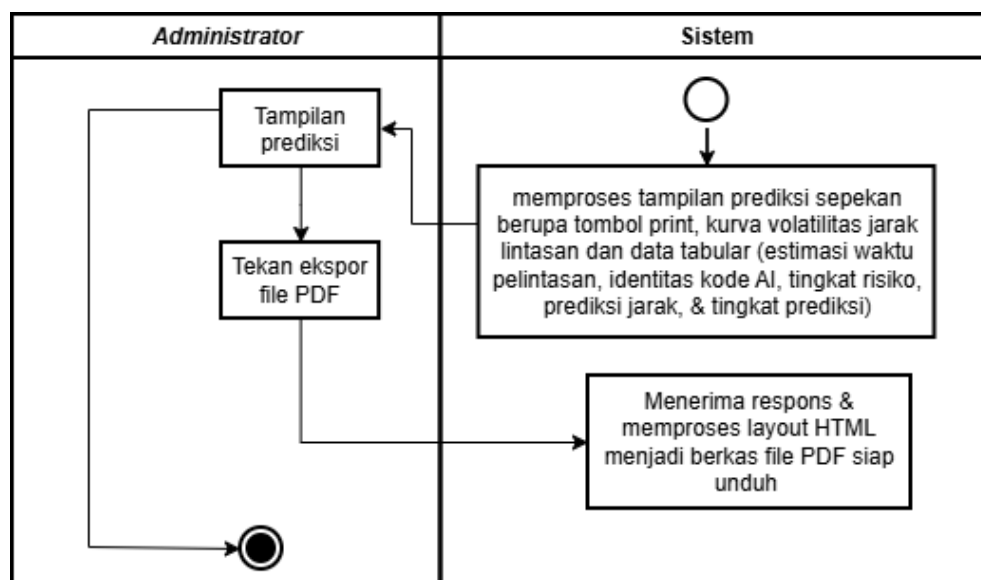
1) *Activity Diagram* Memantau Prediksi AI



Gambar 5. 67 *Activity Diagram* Memantau Prediksi AI Visitor
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 67, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) oleh pihak

sistem yang merespons rute masuk menu dengan memproses tampilan prediksi sepekan berupa tombol *print*, kurva volatilitas jarak lintasan dan data tabular yang memuat rincian estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, beserta tingkat prediksi secara dinamis. Hasil pemetaan sekuensial tersebut kemudian disajikan secara utuh di layar peramban pengguna pada bagian tampilan prediksi. Karena adanya batasan hak akses cetak, aktor *Visitor* tidak dapat mengunduh dokumen secara langsung, melainkan diarahkan untuk mengeksekusi instruksi tekan *request* cetak dokumen kemudian *input* pesan, setelah itu tekan batal untuk kembali dan tekan kirim *request* untuk lanjut. Tindakan interaktif dari *Visitor* ini dikirimkan kembali menuju server, di mana pihak sistem akan memproses perintah tersebut lewat fungsi menerima *respons* & menyimpannya ke dalam basis data antrean *print_request* agar nantinya dapat ditinjau dan divalidasi oleh pihak *Admin*. Seluruh rangkaian aktivitas pemantauan kurva dan pengajuan kueri cetak prediksi oleh *Visitor* ini resmi berakhir dan mencapai simpul penutup (*final state*) tepat setelah halaman antarmuka kembali ke kondisi sedia untuk dijelajahi.

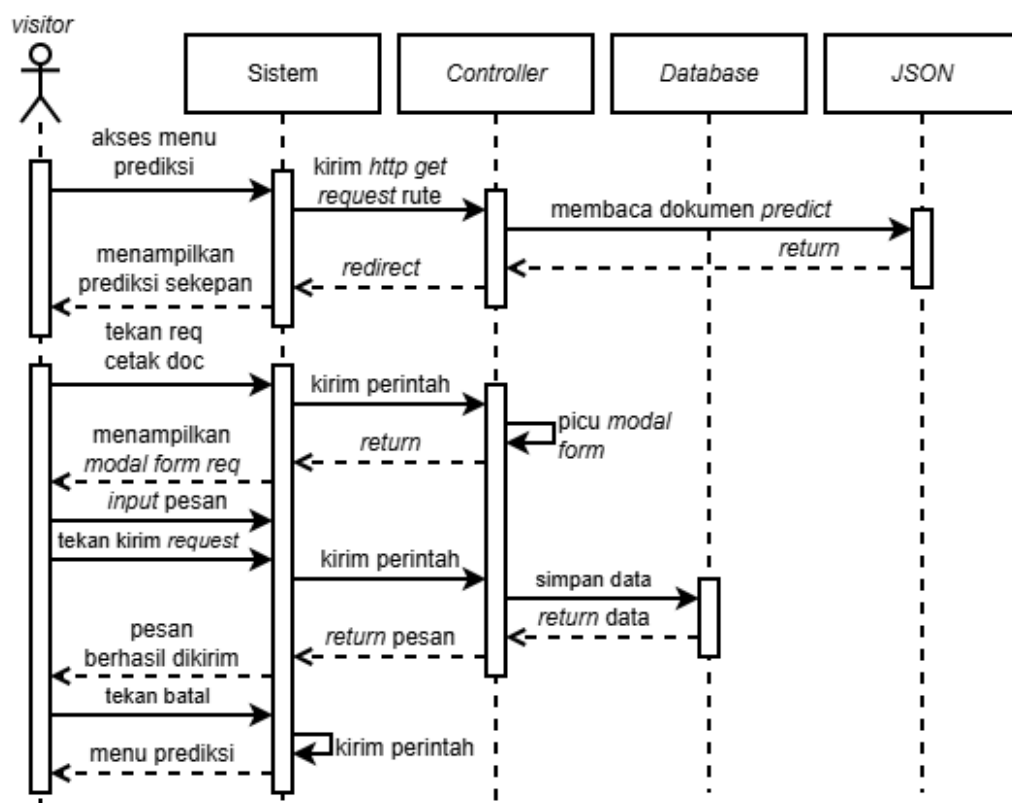


Gambar 5. 68 Activity Diagram Memantau Prediksi AI Admin
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 68, diagram aktivitas tersebut diawali dari simpul awal (*initial state*) di sisi sistem, di mana sistem secara otomatis menjalankan proses memproses tampilan prediksi

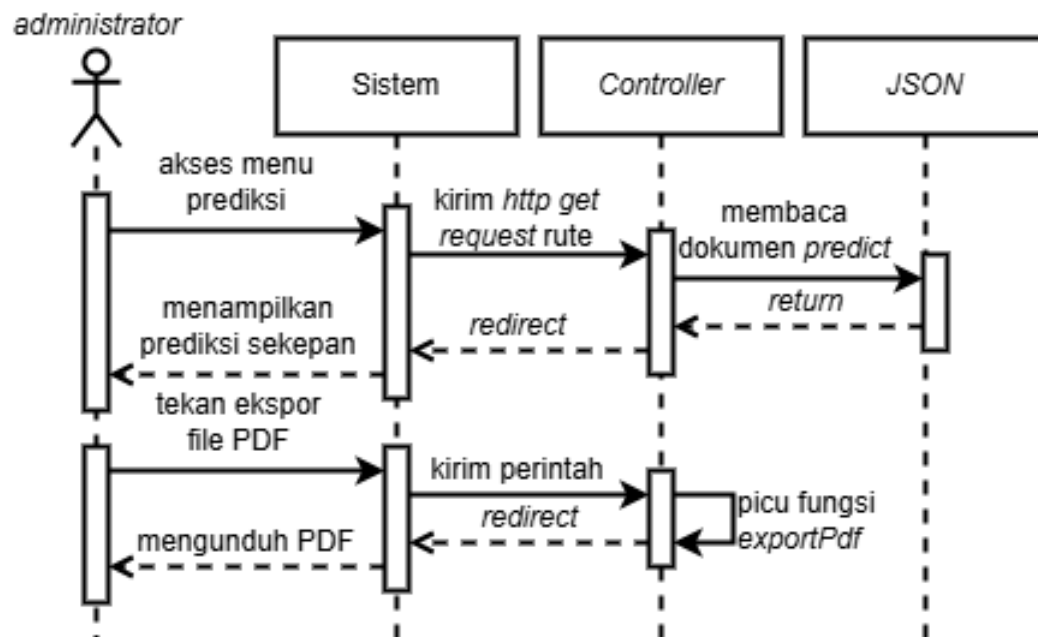
sepekan berupa tombol ekspor *file* PDF, kurva volatilitas jarak lintasan dan data tabular yang mencakup informasi estimasi waktu pelintasan, identitas kode AI, tingkat risiko, prediksi jarak, serta tingkat prediksi hasil kalkulasi model jaringan saraf rekuren. Setelah data ter-render sempurna, aktor *Administrator* akan menerima visualisasi pada halaman tampilan prediksi. Langkah berikutnya, *Admin* menindaklanjuti data peramalan sekuensial tersebut dengan melakukan instruksi tekan ekspor *file* PDF. Aksi dari pengguna otoritas ini dikirimkan kembali sebagai instruksi masuk ke sisi *back-end*, di mana sistem bekerja menanggapi perintah tersebut melalui fungsi menerima *respons* & memproses *layout* HTML menjadi berkas *file* PDF siap unduh agar dapat disimpan menjadi arsip digital laporan instansi. Aliran diagram aktivitas khusus *administrator* ini dinyatakan selesai dan secara langsung mengarah menuju titik simpul penutup (*final state*) sesaat setelah proses pembuatan berkas PDF selesai diolah oleh server.

2) Sequence Diagram Memantau Prediksi AI



Gambar 5. 69 Sequence Diagram Memantau Prediksi AI (Visitor)
(Sumber: Data Pribadi)

Berdasarkan pemodelan logis koneksi hibrida pada Gambar 5. 69, mengenai alur menu prediksi lintasan bagi *Visitor* menunjukkan rangkaian operasi yang berfokus pada visualisasi prediksi cerdas dan pengajuan permohonan dokumen. Interaksi dimulai ketika aktor *Visitor* memuat halaman antarmuka Prediksi Lintasan View yang secara otomatis memicu pengiriman instruksi penarikan data menuju objek komponen *NeoController*. Objek pengendali ini segera memanggil fungsi penanganan inferensi cerdas dengan memanfaatkan data runtun waktu yang tersedia di dalam pangkalan data MySQL melalui perantara objek model *NeoObject*. Guna menghasilkan proyeksi lintasan masa depan, objek kontroler mengirimkan instruksi eksekusi terminal menuju skrip kecerdasan buatan berbasis *Python* yang bertugas memuat berkas biner model berarsitektur *Recurrent Neural Network* (RNN). Hasil kalkulasi peramalan kontinu sepekan ke depan tersebut diekspor kembali oleh *Python* ke dalam struktur berkas JSON, diterima oleh kontroler untuk disimpan ke dalam memori *caching*, dan dikirimkan kembali ke lapisan Prediksi AI View untuk disajikan menjadi kurva volatilitas jarak lintasan dan data tabular terpaginasi di hadapan *Visitor*. Siklus ditutup ketika *Visitor* menekan tombol *request* cetak dokumen pada antarmuka, di mana payload formulir beserta pesan catatan dikirimkan menuju objek *PrintRequestController* untuk diproses menjadi rekaman baris antrean permohonan baru berstatus tertunda pada model *Print Request* di dalam basis data MySQL, sebelum akhirnya sistem mengembalikan pesan konfirmasi sukses berwarna hijau ke layar komputer *Visitor*.



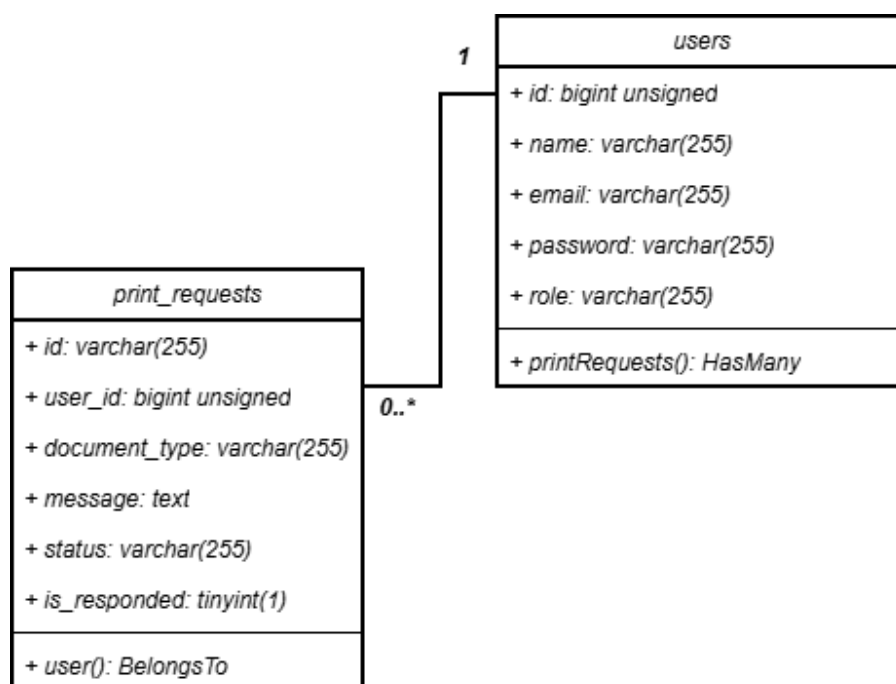
Gambar 5. 70 Sequence Diagram Memantau Prediksi AI (Admin)
(Sumber: Data Pribadi)

Berdasarkan pemodelan logis koneksi hibrida pada Gambar 5. 70, mengenai alur ekspor laporan prediksi lintasan objek dekat bumi oleh *Admin*, siklus interaksi diawali ketika aktor *Admin* memuat halaman antarmuka *prediksi.blade.php* yang secara otomatis memicu pengiriman instruksi penarikan data menuju objek komponen *NeoController*. Objek pengendali ini segera memanggil fungsi penanganan inferensi cerdas dengan memanfaatkan data runtun waktu yang tersedia di dalam pangkalan data MySQL melalui perantara objek model *NeoObject*. Guna menghasilkan proyeksi lintasan masa depan, objek kontroler mengirimkan instruksi eksekusi terminal menuju skrip kecerdasan buatan berbasis *Python* yang bertugas memuat berkas biner model berarsitektur *Recurrent Neural Network* (RNN). Hasil kalkulasi peramalan kontinu sepekan ke depan tersebut diekspor kembali oleh *Python* ke dalam struktur berkas JSON, diterima oleh kontroler untuk disimpan ke dalam memori *caching*, dan dikirimkan kembali ke lapisan *Prediksi Lintasan View Admin* juga dapat mengeksekusi tindakan penekanan tombol ekspor *file PDF* pada antarmuka *prediksi.blade.php*. Komponen antarmuka ini menindaklanjuti secara responsif dengan melempar parameter tipe berupa prediksi menuju objek komponen *PrintRequestController* melalui pemanggilan fungsi *exportPdf*.

Menyadari beban komputasi data besar, objek pengendali langsung melakukan optimalisasi kinerja server latar belakang dengan menonaktifkan batasan waktu eksekusi serta melonggarkan kapasitas alokasi memori sistem. Alur kerja berlanjut ketika kontroler melakukan pengecekan data lintasan secara runtun waktu ke dalam sistem penyimpanan *cache* Laravel, yang apabila memori tersebut kosong, sistem akan beralih mengeksekusi pembacaan data cadangan terstruktur dari berkas *backup* JSON yang tersimpan di dalam direktori penyimpanan lokal. Tidak hanya memuat data proyeksi lintasan masa depan, kontroler juga melakukan operasi pembacaan nilai metrik akurasi latih model kecerdasan buatan secara murni dari berkas log pelatihan JSON untuk dikalkulasikan menjadi persentase tingkat presisi penalaran model RNN. Setelah seluruh variabel data performa sains dan jangkauan tanggal terkumpul utuh, kontroler membersihkan *buffer* memori secara berkala dan meneruskan *payload* tersebut ke dalam objek *DomPDF* dengan merujuk pada templat berkas visual *SINEO_prediksi_pdf*. Sebagai luaran final, objek *DomPDF* mengompilasi seluruh struktur data biner tersebut menjadi berkas dokumen PDF resmi dan mengirimkannya kembali ke lapisan antarmuka pengguna guna memicu browser lokal milik *Admin* agar mengeksekusi perintah unduh berkas secara otomatis ke dalam media penyimpanan komputer.

B. Class diagram

Di samping melakukan dekonstruksi aliran aktivitas melalui diagram aktivitas, fase *Sprint* pada kerangka kerja *Scrum* ini juga menetapkan struktur statis sistem melalui penyusunan *class diagram*. Pemodelan struktural ini bertujuan untuk memetakan secara eksplisit seluruh kelas objek komponen model dan representasi data yang beroperasi di dalam arsitektur sistem *monitoring* SINEO, lengkap dengan spesifikasi visibilitas atribut murni (berbasis tipe data *riil* pada berkas kueri SQL), metode fungsi operasional (*methods*), serta derajat hubungan keterikatan (*multiplicity*) antar-entitas di dalam lingkungan *framework* Laravel 12.



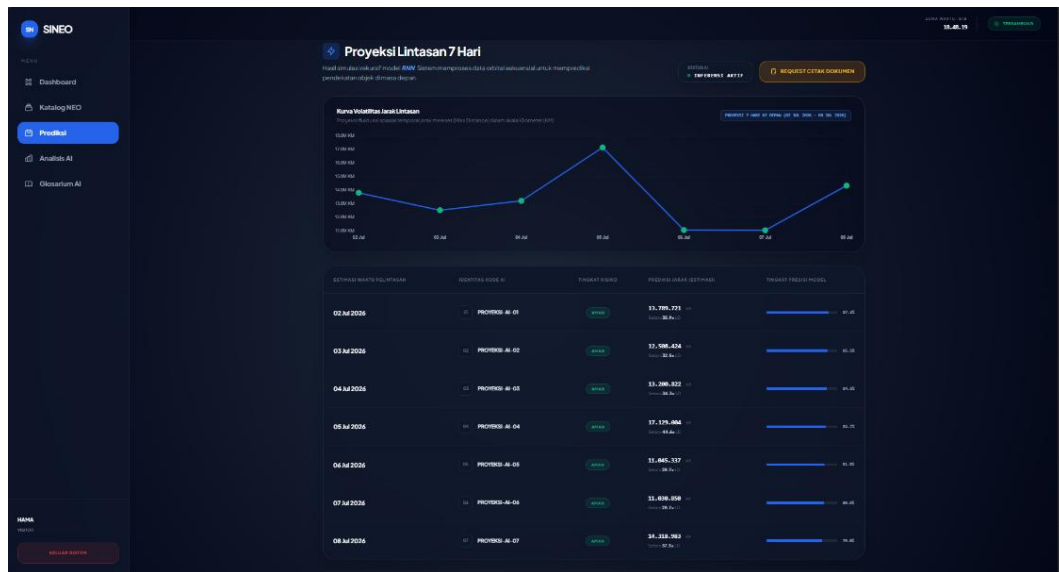
Gambar 5. 71 Class Diagram Print Request SINEO
(Sumber: Data Pribadi)

Berdasarkan manifestasi struktur diagram kelas pada Gambar 5. 71, diagram kelas ini memetakan rancangan pemrograman berorientasi objek yang mengendalikan siklus pengajuan, pencetakan dokumen katalog lengkap (*Admin*) pada *platform* SINEO. Arsitektur diagram ini menempatkan kelas pengontrol *PrintRequestController* sebagai komponen pusat yang bertindak sebagai pengendali utama aliran data. Di dalam kelas pengontrol ini, didefinisikan rentetan metode publik yang meliputi fungsi *store()* untuk menangani penyimpanan berkas permohonan baru dari entitas *Visitor*, serta fungsi tindakan mutlak berupa *exportPdf()* untuk mengeksekusi penarikan dokumen secara murni melalui instruksi tekan ekspor *file* PDF pada menu Katalog NEO dan Prediksi AI oleh *Admin*, di mana sistem langsung merespons secara hibrida dengan memicu fungsi menerima *respons* dan memproses *layout* HTML untuk masing-masing menu, menjadi berkas *file* PDF siap unduh berukuran kertas A4. Kelas pengontrol ini dirancang mewarisi (*inheritance relationship*) sifat dasar dari kelas komponen inti *framework* Laravel, yaitu kelas *controller*.

Guna menjamin validitas dan keamanan muatan data yang dikirimkan melalui *form* antarmuka, kelas *PrintRequestController* memiliki hubungan ketergantungan (*dependency relation*) yang kuat terhadap kelas utilitas *PrintRequest*. Kelas *PrintRequest* merupakan representasi model *Eloquent* objek entitas fisik tabel *print_requests* di dalam *database* MySQL, yang mengenkapsulasi properti data operasional seperti atribut *id*, kunci tamu *user_id*, string dokumen *document_type*, penanda status *status*, deskripsi alasan penolakan *rejection_reason*, serta jejak waktu *created_at* dan *updated_at*. Dari sudut pandang relasi basis data relasional *phpMyAdmin*, hubungan kelas model *User* terhadap kelas model *PrintRequest* memiliki nilai kardinalitas mutlak *one-to-many* (*1 ke 0..**). Penentuan derajat *multiplicity* ini menegaskan bahwa satu baris rekaman pada kelas model *User* dapat memiliki hubungan kepemilikan terhadap banyak (*0..**) baris rekaman dokumen pengajuan cetak, sedangkan satu (*1*) instansiasi baris data pada kelas model *PrintRequest* wajib terikat secara independen pada satu identitas pengguna yang valid guna menjaga akuntabilitas *log* audit keamanan sistem.

3. *Implementation*

Sistem antarmuka ini bertindak sebagai media interaksi utama (*front-end*) yang menyajikan hasil inferensi prediksi dari model biner *Dual-Stacked RNN* (*neo_rnn_model.keras*) secara dinamis kepada pengguna. Struktur pengembangan komponen antarmuka Prediksi AI sistem *monitoring* SINEO yang terbaru dirancang secara modular dan didokumentasikan sebagai berikut:



Gambar 5. 72 Antarmuka Proyeksi Prediksi Jarak Berbasis Jaringan Saraf Berulang (Visitor)
(Sumber: Data Pribadi)

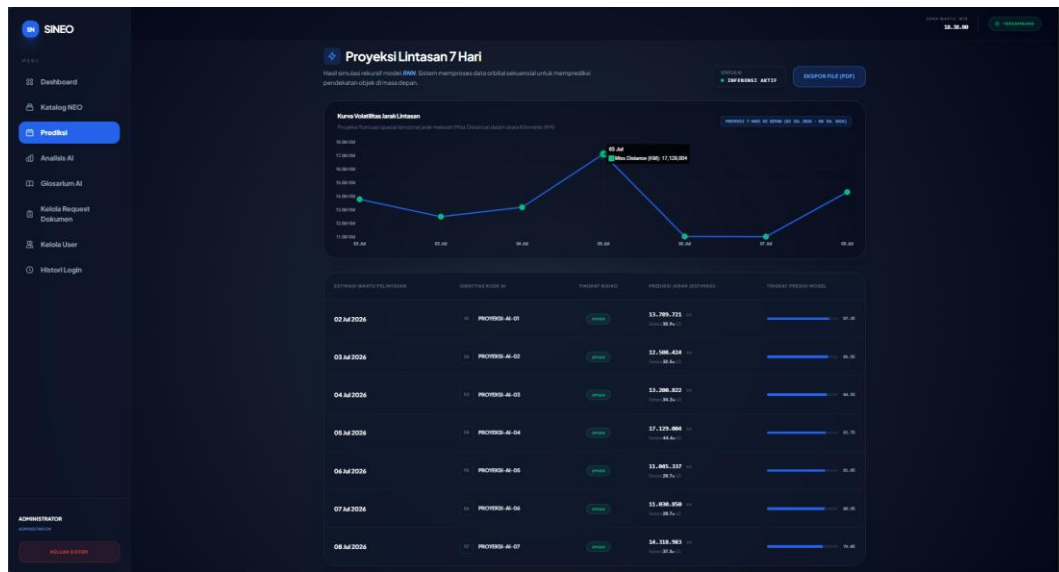
AJUAN DOKUMEN ANALISIS PREDIKSI

Kirimkan kueri permintaan cetak ini agar data divalidasi dan diekspor menjadi file biner PDF reams oleh Administrator.

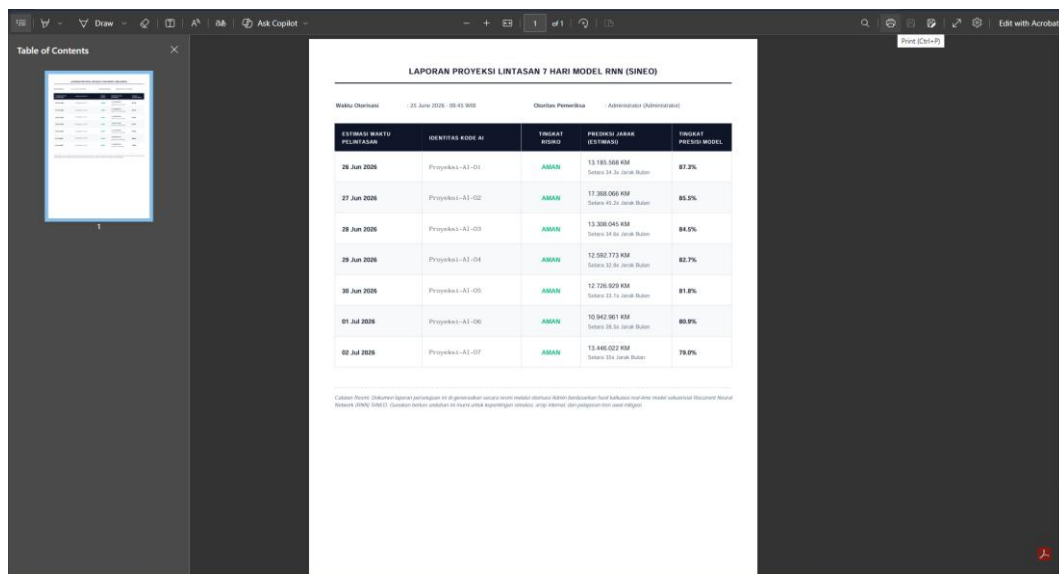
CATATAN SPESIFIK OBJEK / KEPERLUAN (OPSIONAL)

Tuliskan pesan khusus...

Gambar 5. 73 Antarmuka *Modal Form Request* Proyeksi Prediksi Jarak Berbasis Jaringan Saraf Berulang (Visitor)
(Sumber: Data Pribadi)



Gambar 5. 74 Antarmuka Proyeksi Prediksi Jarak Berbasis Jaringan Saraf Berulang (Admin) (Sumber: Data Pribadi)



Gambar 5. 75 Tampilan Hasil Ekspor Dokumen *File* Proyeksi Prediksi Jarak Berbasis Jaringan Saraf Berulang (Sumber: Data Pribadi)

Halaman *prediksi_bulanan.blade.php* dikembangkan untuk memwadhahi komponen utama dari fungsi peramalan temporal pada sistem *monitoring* koordinat objek astronomi. Berkas antarmuka ini memuat visualisasi grafik garis dinamis (*line chart*) yang menampilkan fluktuasi estimasi jarak lintasan terdekat (*miss distance*) objek asteroid terhadap Bumi untuk langkah waktu ke depan, serta didukung oleh

penyajian data tabular ringkas proyeksi lintasan tujuh hari sepekan dan indikator *widgit card* pemantau efektivitas konvergensi model kecerdasan buatan. Secara mekanis, skrip di dalam berkas ini membaca keluaran *Recursive Multi-step Forecasting* dari berkas hasil komputasi mesin inferensi *Python* yang disimpan di dalam memori penyimpanan *cache* lokal. Kehadiran halaman grafik prediksi ini memudahkan pengguna dalam melakukan analisis spasial temporal serta mitigasi dini terhadap tren pergerakan lintasan objek koordinat orbit di masa mendatang.

Bagi sisi *Visitor*, berkas antarmuka ini menyediakan elemen tombol operasional guna memicu kemunculan modal pengajuan dokumen analisis prediksi menuju baris antrean pangkalan data MySQL. Sementara bagi sisi *Admin*, antarmuka ini dirancang secara simetris menempatkan tombol ekspor PDF mandiri tepat di sebelah indikator status AI (inferensi aktif) guna memicu eksekusi fungsi *approve* pada kelas *RequestManagementController*. Saat tombol tersebut dieksekusi oleh *Admin* untuk memenuhi kebutuhan internal seperti presentasi data analisis, sistem secara otomatis akan memicu optimalisasi batas waktu performa server latar belakang, melakukan pembacaan data ramalan sekuensial beserta metrik akurasi latih model RNN dari berkas log pelatihan JSON, serta mengompilasikan seluruh variabel laporan terstruktur tersebut menggunakan objek *DomPDF* melalui berkas templat *prediksi_otorisasi_pdf* untuk menghasilkan *file* biner PDF proyeksi lintasan resmi yang langsung terunduh secara otomatis ke dalam media penyimpanan lokal.

4. Review

Aktivitas *Sprint Review* pada akhir siklus iterasi kelima dilaksanakan untuk mengevaluasi fungsionalitas visualisasi data astronomi dinamis sebelum masuk ke modul administratif final. Eksplorasi fungsional komponen visual ini didemonstrasikan secara terbuka, langsung oleh pengguna umum guna memastikan bahwa pembaruan visualisasi yang lebih mudah dipahami oleh orang awam sudah dilakukan. Hasil peninjauan kelayakan produk berdasarkan standar penulisan tugas akhir dirangkum ke dalam Tabel 5. 17 berikut:

Tabel 5. 18 Hasil *Review Sprint Retrospective*

No.	Kebutuhan Sistem	Keterangan	Status
1.	Modul Halaman Prediksi AI	Berhasil menampilkan kurva volatilitas jarak lintasan, prediksi sepekan pada data tabular berdasarkan hasil <i>Recursive Multi-step Forecasting</i> dari skrip <i>Python</i> yang otomatis menampilkan data terbaru setiap kali Admin melakukan <i>sync-data</i> NASA.	Valid

(Sumber: Data Pribadi)

Berdasarkan hasil peninjauan pada Tabel 5. 17, seluruh blok fungsionalitas dan visualisasi pada *Sprint Retrospective* dinyatakan bekerja secara valid dan berstatus setuju. Pembaruan atau penambahan yang informatif pada menu Prediksi AI sudah disetujui dan tidak ada catatan evaluasi kembali.

5.1.5.4. Hasil *sprint* 4

5.1.5.4.1. *Daily scrum*

Daily Scrum pada *Sprint* 4 ditujukan untuk mengonsolidasi seluruh modul administratif, sistem otorisasi dokumen, serta penguatan audit keamanan sistem. Tim melakukan koordinasi harian guna memastikan bahwa fitur Kelola *Request* Dokumen dapat bekerja secara hibrida, di mana sistem mampu mengintegrasikan data dari *database* MySQL dengan pustaka *DomPDF* untuk menghasilkan luaran dokumen fisik secara presisi. Kendala teknis yang dihadapi selama siklus ini mencakup integrasi *Google SMTP Server* untuk pengiriman notifikasi otomatis ke *Gmail Visitor*, serta upaya mitigasi *I/O bottleneck* saat Admin menarik data historis otentikasi pengguna secara penuh dari tabel *login_histories*. Melalui evaluasi harian yang disiplin, tim berhasil mengimplementasikan penguncian *status respons* pada model *PrintRequest* menggunakan *flag is_responded*, sehingga aksi operasional

Admin seperti pengiriman *email* tidak mengalami duplikasi yang tidak diinginkan. Selain itu, koordinasi harian ini juga memvalidasi ketepatan alur kerja CRUD pada manajemen pengguna dan memastikan seluruh jejak aktivitas akses pengguna terekam dengan akurasi *timestamp* yang valid untuk keperluan audit berkala.

Tabel 5. 19 Hasil *Daily Scrum Sprint 4*

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
01–02 Mei 2026	-	-	Implementasi <i>RequestManagement</i> <i>Controller</i> & Optimasi <i>DomPDF</i> dengan <i>queue</i> data

Tanggal	Apa yang dilakukan kemarin?	Apa hambatan yang dialami?	Apa yang dikerjakan hari ini
03 Mei 2026	Implementasi fitur Cetak Dokumen (PDF)	<i>Timeout</i> rendering berkas PDF masif	Uji coba pengiriman <i>email</i> otomatis
04 Mei 2026	Integrasi SMTP <i>Gmail Mailer</i>	Konfigurasi otentikasi Google <i>App Password</i>	<i>Coding</i> CRUD Kelola <i>User</i>
05 Mei 2026	<i>Coding</i> CRUD Kelola <i>User</i>	Konflik <i>routing</i> otorisasi Admin	Pengembangan fitur Histori <i>Login</i> & Implementasi <i>export</i> PDF untuk log audit
06 Mei 2026	Pengembangan fitur Histori <i>Login</i> & Tombol <i>export</i> PDF	<i>Parsing</i> data log aktivitas pengguna	<i>Integration</i> <i>Testing</i> antar modul (<i>Recursive Forecasting</i>)
07 Mei 2026	<i>Integration</i> <i>Testing</i> antar modul	<i>Bug</i> pada penguncian status <i>request</i>	Finalisasi <i>button</i> status <i>disabled/active</i>

(Sumber: Data Pribadi)

5.1.5.4.2. Design

A. Activity dan sequence diagram

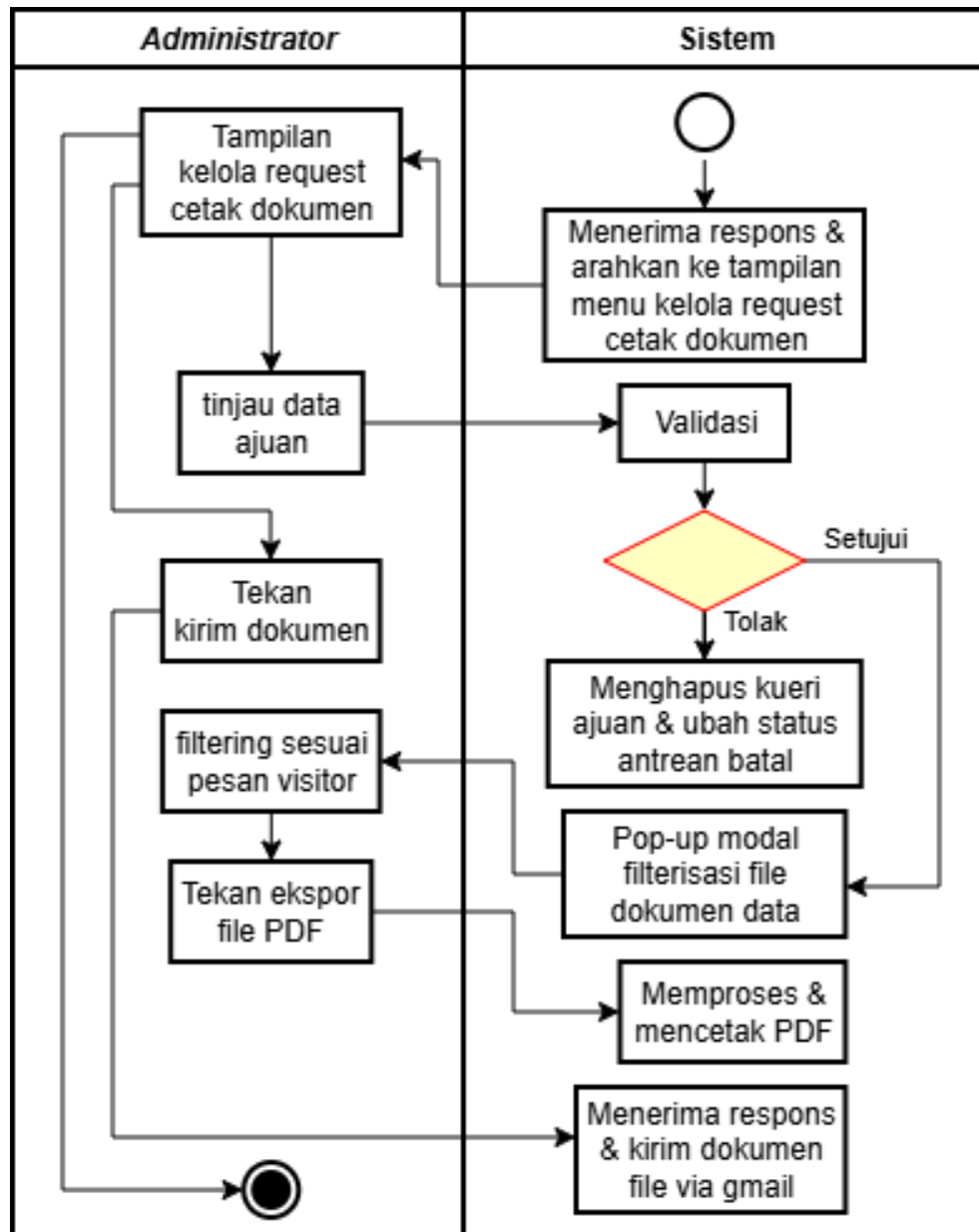
Perancangan *Activity Diagram* secara menyeluruh dilakukan untuk setiap butir *use case* yang telah didefinisikan sebelumnya. Pemodelan aktivitas ini bertujuan untuk memetakan alur kerja (*workflow*) logika sistem secara spesifik, memperjelas transisi interaksi antara aktor (*front-end UI*) dengan *respons* pengontrol (*back-end*), serta menggambarkan bagaimana data mengalir secara linear pada setiap fungsi di dalam arsitektur *hybrid environment* SINEO.

Guna melengkapi pemodelan struktural statis, interaksi dinamis berorientasi waktu dan pertukaran pesan (*message passing*) antar-objek di dalam ekosistem SINEO dimodelkan secara eksplisit memanfaatkan *Sequence Diagram*. Pemodelan *behavioral* ini bertujuan untuk menggambarkan urutan kronologis operasi koneksi hibrida (*hybrid connection*) yang melibatkan siklus hidup objek secara utuh. Seluruh rangkaian skenario interaksi dipetakan secara sinkron dan linear dengan *Activity Diagram* yang telah dijabarkan sebelumnya, mencakup fungsionalitas di sisi pengguna umum (*Visitor*), hak akses khusus (*Administrator*), hingga otomatisasi komputasi di latar belakang server (*back-end*).

Guna menyajikan dokumentasi sistem yang terstruktur dan komprehensif, urutan pertukaran pesan dinamis ini dijabarkan ke dalam skenario operasi sebagai berikut:

a. Kelola *request* cetak dokumen

1) *Activity diagram* kelola *request* cetak dokumen



Gambar 5. 76 *Activity Kelola Request Cetak Dokumen*
(Sumber: Data Pribadi)

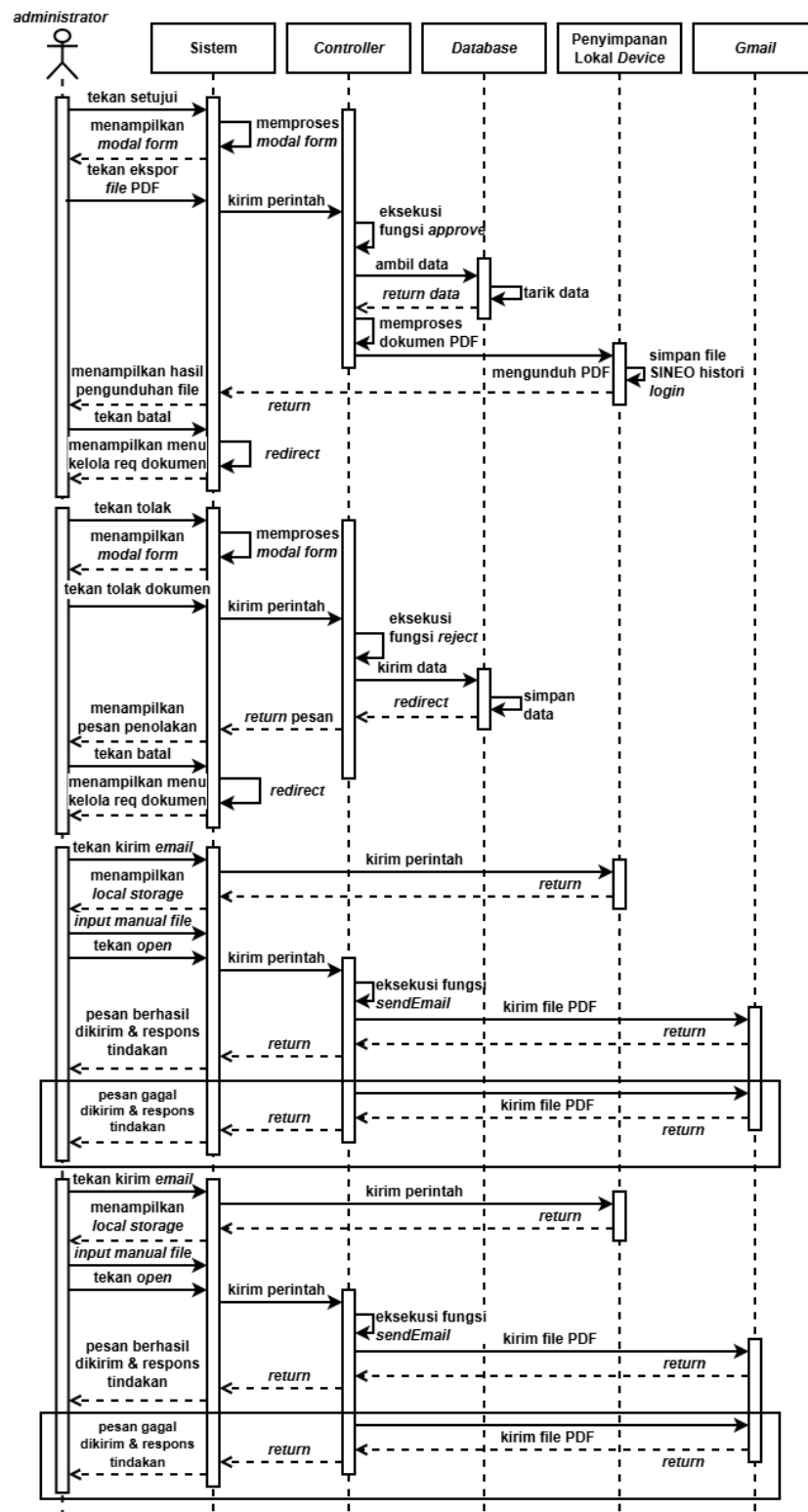
Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 76, diagram aktivitas tersebut diawali ketika seorang *Visitor* masuk ke dalam sistem SINEO dan menuju ke halaman pengajuan cetak dokumen. Di tahap awal ini, *Visitor* akan mengisi *form* permohonan dengan memasukkan parameter yang

diperlukan, seperti memilih jenis dokumen serta menentukan rentang waktu data yang diinginkan. Setelah semua data terisi dengan benar, *Visitor* menekan tombol ajukan (*request*), yang kemudian memicu sistem untuk merekam permohonan tersebut ke dalam basis data MySQL. Seketika itu juga, data pengajuan akan masuk ke dalam antrian verifikasi dengan status awal berupa *Pending*, dan sistem akan memperbarui tampilan halaman *Admin* untuk menunjukkan bahwa dokumen sedang menunggu persetujuan.

Siklus kemudian berpindah ke sisi *Administrator*, di mana sistem secara otomatis memperbarui daftar antrian verifikasi pada *dashboard* manajemen milik *Admin*. *Admin* kemudian dapat melihat detail dari setiap permohonan yang masuk dan dihadapkan pada dua pilihan keputusan otorisasi, yaitu menyetujui (*approve*) atau menolak (*reject*) permohonan tersebut. Jika *Admin* mendapati adanya ketidaksesuaian data atau parameter yang diminta tidak valid, *Admin* akan memilih opsi penolakan, yang kemudian memicu sistem untuk memunculkan modal dialog alasan penolakan. Setelah *Admin* memasukkan alasan tersebut, sistem akan memperbarui status ajuan di *database* menjadi *Rejected*, menyimpan alasan penolakannya, dan memperbarui halaman secara reaktif.

Sebaliknya, jika permohonan dirasa sudah sesuai dan valid, *Admin* akan menekan tombol persetujuan yang memicu sistem untuk memproses data tersebut menjadi sebuah dokumen biner. Sistem kemudian bekerja di latar belakang untuk melakukan kompilasi data dan mengekspornya menjadi *file* PDF resmi yang siap unduh. Begitu *file* PDF berhasil dibuat, sistem akan langsung mengirimkan *response* berupa *file download* ke browser *Admin* serta secara otomatis mengubah status ajuan di dalam *database* menjadi *Approved*. Sebagai penutup dari alur aktivitas ini, sistem yang telah diperbarui secara reaktif menggunakan *Alpine.js* akan langsung mengubah tampilan tindakan di layar *Admin*, menutup akses tombol otorisasi, dan memunculkan opsi baru untuk mengirimkan hasil cetak dokumen tersebut kepada *Visitor* melalui *gmail*.

2) Sequence diagram kelola request cetak dokumen



Gambar 5. 77 Sequence Diagram Kelola Request Cetak Dokumen
(Sumber: Data Pribadi)

Berdasarkan pemodelan peralihan media perangkat keras pada Gambar 5. 77, mengenai alur otorisasi persetujuan dokumen katalog dan prediksi oleh *Admin*, rangkaian interaksi diawali ketika *Admin* berinteraksi dengan antarmuka halaman *Request Management View* dengan mengeksekusi penekanan tombol setuju. Komponen antarmuka kemudian secara responsif mengirimkan *payload* parameter menuju objek *RequestManagementController* melalui pemanggilan fungsi *approve*. Objek pengendali menguji kondisi tipe dokumen, di mana jika bertipe katalog maka sistem melakukan kueri penyaringan data ke dalam *database* MySQL melalui model *NeoObject* berdasarkan *filter* tanggal, sedangkan jika bertipe prediksi sistem akan menarik data proyeksi sepekan dari memori *cache* atau *backup* JSON beserta metrik akurasi dari log pelatihan. Setelah variabel data laporan berhasil dihimpun dan diformat, kontroler mengirimkan perintah pembaruan status menjadi disetujui menuju model *PrintRequest* di dalam *database* MySQL. Siklus interaksi diakhiri dengan pembersihan *buffer* memori secara paksa oleh kontroler guna memuat *payload* data terstruktur ke dalam komponen *DomPDF* untuk *render* menjadi *file* biner PDF resmi, sebelum akhirnya dikembalikan ke lapisan *Request Management View* untuk memicu browser lokal milik *Admin* mengeksekusi perintah unduh berkas otomatis.

Sementara itu, mengenai alur saat *Admin* menekan tombol tolak pada baris antrean dokumen di dalam antarmuka *Request Management View*, yang secara instan membuka jendela modul *input* dan mengirimkan data *string* teks alasan menuju objek *RequestManagementController* melalui pemanggilan fungsi *reject*. Begitu *payload data* kiriman formulir diterima, objek pengendali langsung melakukan validasi struktur teks guna memastikan kolom alasan tidak kosong. Alur kerja dilanjutkan dengan memicu operasi pembaruan data secara searah ke dalam *database* MySQL melalui perantara objek model *PrintRequest*, di mana sistem mengubah status antrean menjadi ditolak sekaligus menyimpan teks argumen evaluasi ke dalam kolom catatan alasan penolakan di *database*. Konfirmasi sukses penguncian data dari MySQL kemudian diterima oleh kontroler, yang ditutup dengan pengembalian pesan *session flash* sukses menuju antarmuka *Request*

Management View agar sistem menampilkan notifikasi hijau pertanda operasi penolakan dan penyimpanan alasan evaluasi berhasil diselesaikan.

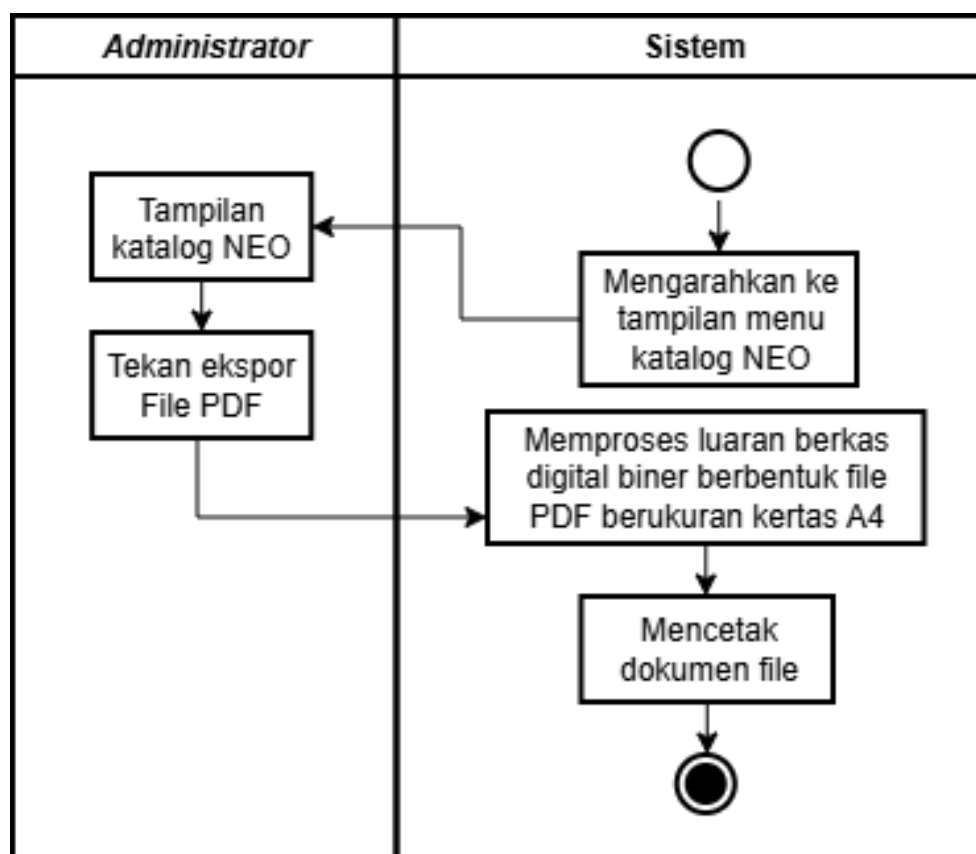
Adapun mengenai alur eksekusi kirim *email* dokumen hasil otorisasi persetujuan oleh *Admin*, siklus aktivitas berfokus pada pendistribusian *file* biner PDF yang telah diunggah manual oleh *Admin* ke alamat *email Visitor*. Skenario berjalan ketika *Admin* menekan tombol kirim *email* pada antarmuka *Request Management View*, yang seketika memicu penyerahan *file* dokumen lokal berformat khusus PDF menuju fungsi *sendEmail* pada objek *RequestManagementController*. Objek pengendali melakukan validasi ekstensi serta ukuran berkas secara ketat, lalu melakukan kueri pencarian data profil akun penerima menuju *database MySQL* melalui model *PrintRequest* untuk mendapatkan alamat *Gmail Visitor* yang bersangkutan. Setelah jalur pengiriman tervalidasi aman, kontroler menginstansiasi objek *DocumentDeliveryMail* dengan menyertakan *path* berkas fisik serta nama dokumen asli, kemudian memerintahkan subsistem *Mailer Service* untuk melakukan pengiriman data keluar melalui *Google SMTP Server* menggunakan otentikasi sandi aplikasi. Setelah server *Google* berhasil meneruskan lampiran PDF tersebut ke kotak masuk *Visitor*, kontroler memperbarui nilai bendera penanda *flag respons* menjadi benar pada model *PrintRequest* di *database MySQL*, lalu mengembalikan status sukses hijau ke layar antarmuka *Admin* agar tombol kirim *email* otomatis berubah menjadi sudah merespons (tidak aktif) guna mencegah terjadinya pengiriman ganda.

Terakhir, mengenai alur eksekusi kirim alasan notifikasi penolakan oleh *Admin* menggambarkan mekanisme pemberitahuan instan tanpa lampiran berkas fisik menuju *Gmail Visitor*. Proses diawali saat *Admin* menekan tombol kirim alasan pada baris dokumen yang telah berstatus ditolak di dalam antarmuka *Request Management View*, yang secara langsung mengirimkan *request POST* menuju objek *RequestManagementController* melalui fungsi *sendEmail*. Objek pengendali segera menarik data permohonan spesifik beserta informasi akun *user* terkait melalui model *PrintRequest* di dalam *database MySQL*, guna mengambil *string* catatan alasan penolakan yang telah disimpan pada tahapan evaluasi sebelumnya.

Setelah teks alasan berhasil dimuat ke dalam memori server, kontroler membungkus variabel nama *visitor*, tipe dokumen, serta deskripsi alasan ke dalam kelas *mailable RejectionNotificationMail* untuk ditembakkan langsung menuju *Google SMTP Server*. Sesaat setelah sistem *SMTP Google* sukses menyalurkan *email* pemberitahuan tersebut ke *Gmail Visitor*, kontroler memperbarui status kolom penanda *is_responded* menjadi benar pada model *PrintRequest* di *database MySQL*, sebelum akhirnya mengembalikan pesan notifikasi berhasil ke antarmuka *Request Management View* demi mengubah tampilan tombol operasional menjadi abu-abu bertuliskan sudah merespons secara dinamis.

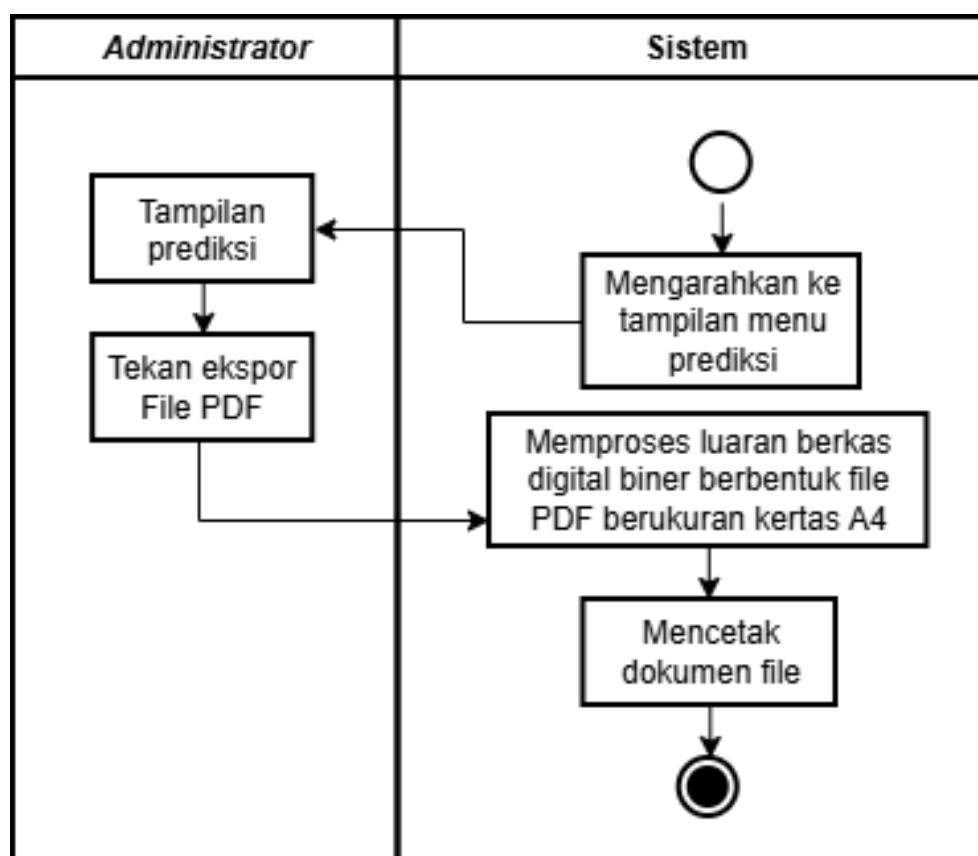
b. Mencetak dokumen

1) *Activity diagram* mencetak dokumen



Gambar 5. 78 *Activity Diagram* Mencetak Dokumen Katalog NEO
(Sumber: Data Pribadi)

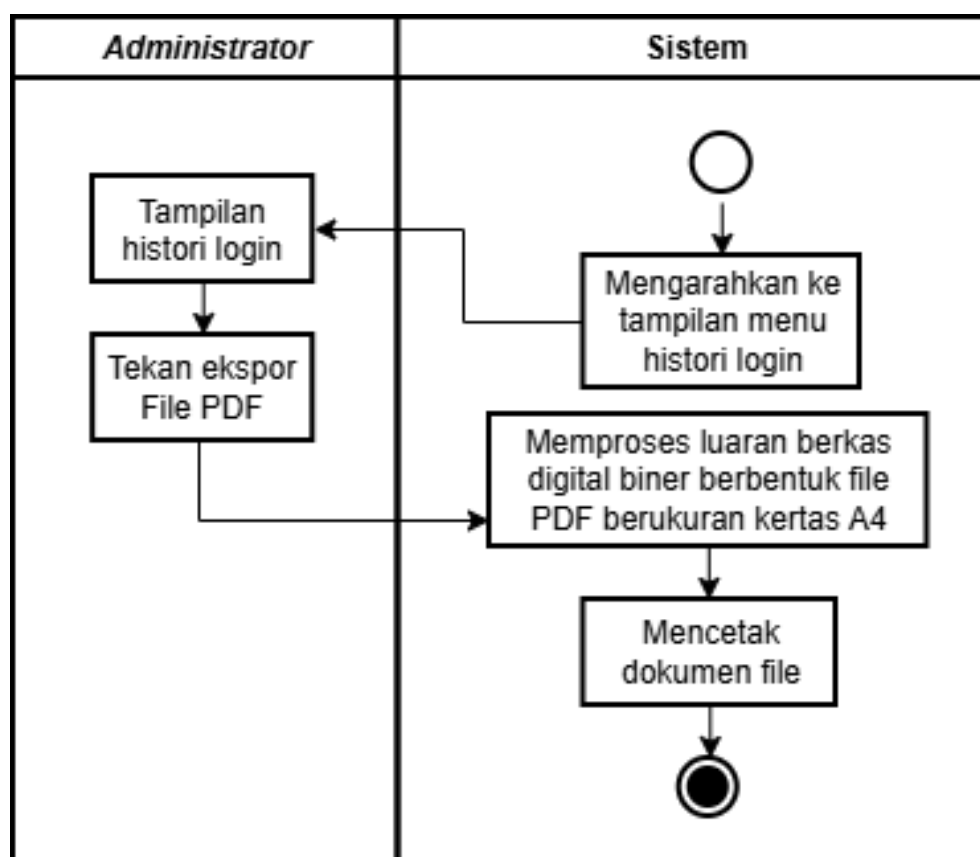
Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 78, diagram aktivitas tersebut diinisiasi dari simpul awal (*initial state*) di sisi sistem yang menjalankan perintah mengarahkan ke tampilan menu katalog NEO. Pengguna menerima penayangan struktur tabel tersebut pada tampilan katalog NEO, lalu mengeklik komponen fungsional tekan ekspor *File* PDF. Sistem menangkap tindakan tersebut dengan memicu proses menghasilkan luaran berkas digital biner berbentuk *file* PDF berukuran kertas A4. *Output* pengondisian server secara tanggap mengeksekusi instruksi mencetak dokumen *file* PDF bersih yang pas di kertas ukuran A4. Aliran keseluruhan diagram aktivitas cetak manajemen katalog khusus *administrator* ini dinyatakan selesai seutuhnya saat menyentuh simpul penutup (*final state*).



Gambar 5. 79 Activity Diagram Mencetak Dokumen Prediksi
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 79, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) di bawah

otoritas Sistem yang memproses perintah mengarahkan ke tampilan menu prediksi guna menyajikan grafik tren fluktuasi sekuensial pada peramban web. Setelah layar ter-render sempurna pada tampilan prediksi, *Admin* mengeklik pilihan instruksi tekan ekspor *file* PDF. Permintaan HTTP dikirimkan langsung menuju server, di mana sistem meresponsnya dan memproses luaran berkas digital biner berbentuk *file* PDF berukuran kertas A4. Setelah berkas siap, sistem akan langsung mengeksekusi fungsi mencetak dokumen *file* biner PDF ke repositori lokal. Rangkaian aktivitas cetak laporan proyeksi sepekan ini resmi berakhir saat alur menyentuh simpul penutup (*final state*).

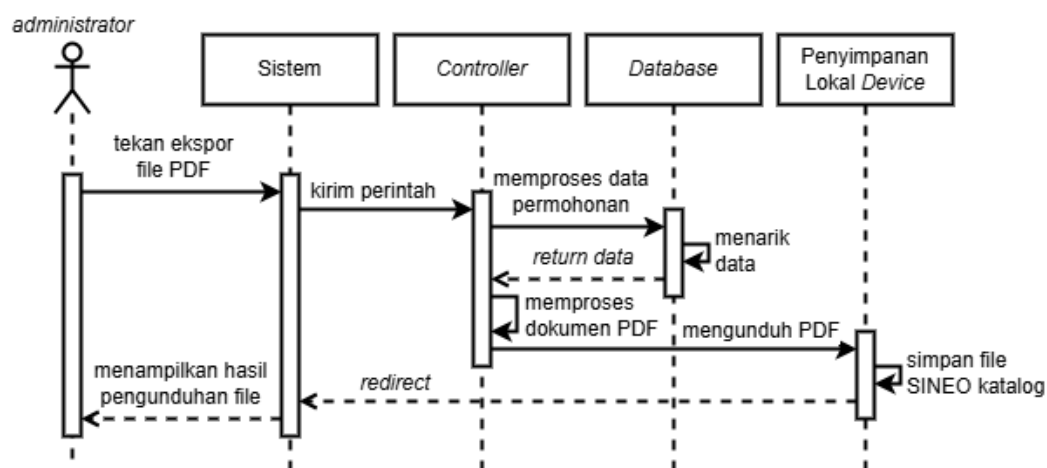


Gambar 5. 80 Activity Diagram Mencetak Dokumen Histori Login
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 80, diagram aktivitas tersebut diawali dari simpul hidup awal (*initial state*) pada sisi sistem yang mengeksekusi perintah mengarahkan ke tampilan menu histori *login* untuk menyajikan ringkasan data otentikasi kepada pengguna. Dari area halaman

tersebut, *Administrator* menindaklanjutinya dengan mengeklik tombol tekan ekspor *file* PDF. Aksi masukan ini dikirimkan ke sisi *backend*, di mana sistem bekerja memproses luaran berkas digital biner berbentuk *file* PDF berukuran kertas A4. Hasil olahan tersebut kemudian secara responsif memicu modul mencetak dokumen *file* berupa salinan arsip cetak PDF murni ke penyimpanan lokal. Rangkaian aktivitas dokumentasi log ini dinyatakan selesai dan ditutup secara resmi tepat pada simpul penutup (*final state*).

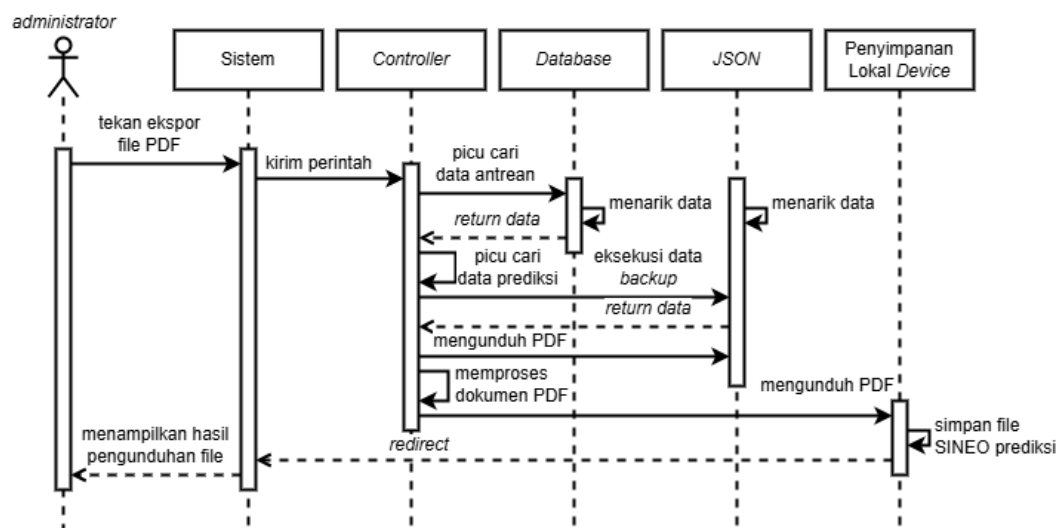
2) *Sequence diagram* cetak dokumen



Gambar 5. 81 *Sequence Diagram* Mencetak Dokumen (Katalog NEO)
(Sumber: Data Pribadi)

Berdasarkan pemodelan peralihan media perangkat keras pada Gambar 5. 81, mengenai alur otorisasi persetujuan dan ekspor dokumen PDF katalog berdasarkan permohonan *Visitor* oleh *Admin*, rangkaian interaksi diawali ketika *Admin* berinteraksi dengan antarmuka *Request Management View* dengan menekan tombol setuju. Komponen antarmuka kemudian mengirimkan parameter pengenalan unik serta *payload data* rentang tanggal melalui pemanggilan fungsi *approve* menuju objek *RequestManagementController*. Objek pengendali mengidentifikasi bahwa permohonan tersebut bertipe katalog, lalu melakukan kueri pencarian data *Near-Earth Objects riil* dari *database MySQL* melalui model *NeoObject* berdasarkan *filter* batasan tanggal yang diajukan. Setelah koleksi objek asteroid dikembalikan oleh model, kontroler mengelompokkan data secara runtun waktu menggunakan komponen *Carbon*, mengemas seluruh variabel ke dalam parameter laporan, serta

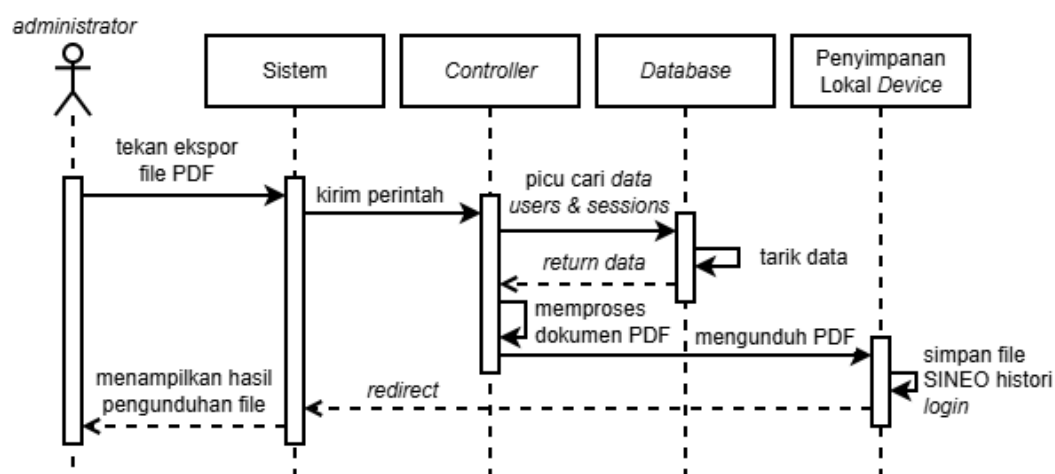
melakukan pembaruan status data permohonan menjadi disetujui pada model *PrintRequest* di *database*. Siklus interaksi berlanjut saat objek pengendali membersihkan *buffer* memori secara paksa dan meneruskan kumpulan *payload* laporan tersebut ke dalam objek *DomPDF* dengan merujuk pada templat berkas visual *katalog_otorisasi_pdf*. Sebagai luaran akhir dari skenario otorisasi katalog ini, objek *DomPDF* mengompilasi seluruh data biner menjadi *file* dokumen PDF utuh dan mengirimkannya kembali ke lapisan *Request Management View*, yang secara instan memicu browser pada komputer lokal *Admin* untuk memulai proses unduh otomatis berkas ke dalam penyimpanan lokal.



Gambar 5. 82 *Sequence Diagram* Mencetak Dokumen (Prediksi AI)
(Sumber: Data Pribadi)

Berdasarkan pemodelan peralihan media perangkat keras pada Gambar 5. 82, mengenai alur otorisasi persetujuan dan ekspor cetak dokumen prediksi lintasan model RNN berdasarkan permohonan *Visitor* oleh *Admin*, siklus interaksi dimulai saat *Admin* menekan tombol setuju untuk request bertipe prediksi pada antarmuka halaman *Request Management View*. Lapisan antarmuka ini secara responsif meneruskan instruksi ekspor data biner menuju objek komponen *RequestManagementController* lewat pemanggilan fungsi *approve* bersama pengenalan data unik. Objek pengendali segera mengenali kondisi tipe dokumen sebagai prediksi, lalu memicu pencarian data ramalan runtun waktu sepekan ke depan secara langsung ke dalam memori *cache* Laravel, yang apabila memori

tersebut kosong, sistem akan secara otomatis beralih mengeksekusi pembacaan data cadangan terstruktur dari berkas *backup* JSON di dalam direktori penyimpanan lokal server. Tidak hanya memuat proyeksi lintasan, kontroler juga melakukan operasi pembacaan nilai metrik akurasi latih model dari berkas log pelatihan JSON untuk dikompilasikan menjadi nilai persentase presisi model penalaran. Setelah seluruh variabel laporan siap, kontroler mengubah status pengajuan menjadi disetujui pada model *PrintRequest* di *database* MySQL, membersihkan seluruh *buffer* memori, dan memuat *payload* data terstruktur ke dalam objek *DomPDF* melalui pemanggilan berkas templat visual *prediksi_otorisasi_pdf*. Sebagai hasil akhir dari skenario otorisasi ini, objek *DomPDF* menggenerasikan *file* biner PDF resmi dan menyerahkannya kembali ke lapisan antarmuka pengguna guna mengeksekusi perintah unduh berkas laporan secara otomatis pada browser lokal milik *Admin*.



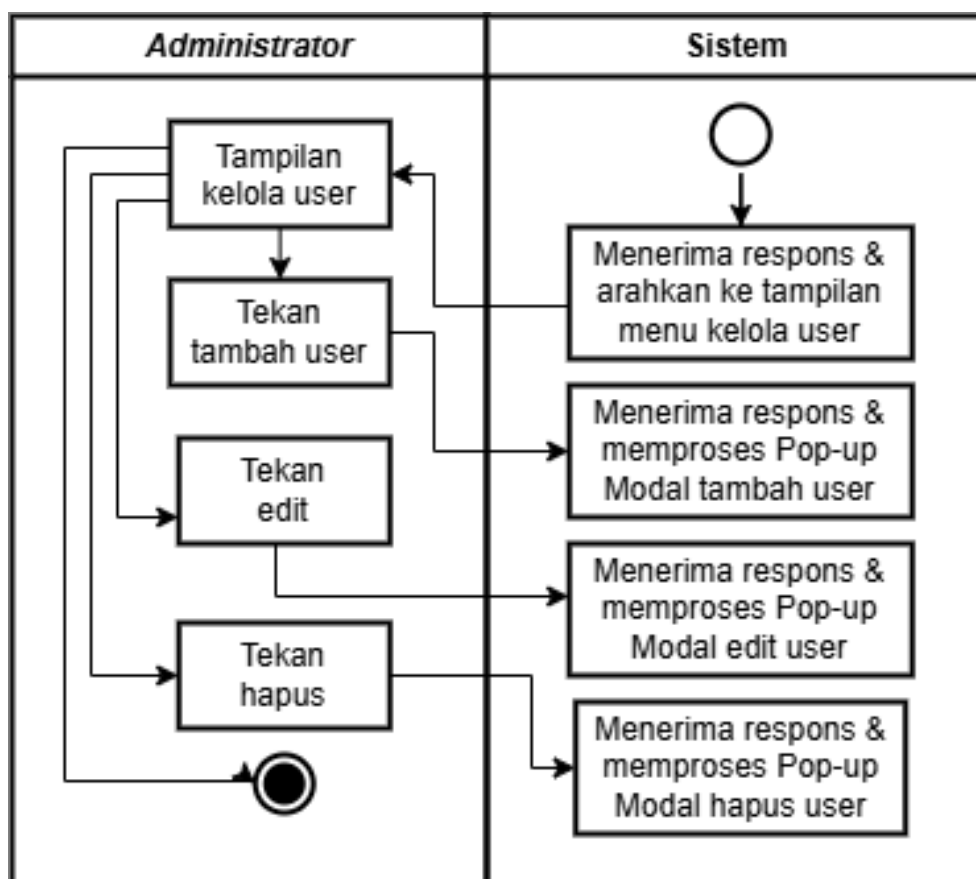
Gambar 5. 83 Sequence Diagram Mencetak Dokumen (Histori Login)
(Sumber: Data Pribadi)

Berdasarkan pemodelan peralihan media perangkat keras pada Gambar 5. 83, mengenai alur cetak dokumen laporan histori *login* oleh *Admin*, rangkaian operasi diawali ketika aktor *Admin* berinteraksi langsung dengan antarmuka halaman utama rekaman aktivitas dengan mengeksekusi tindakan penekanan tombol cetak laporan berupa ekspor PDF. Komponen antarmuka ini menindaklanjutinya secara responsif dengan mengirimkan permintaan pembuatan dokumen laporan audit keamanan menuju objek *LoginHistoryController* melalui pemanggilan fungsi *exportPdf*.

Objek pengendali kemudian langsung memproses instruksi tersebut dengan melakukan kueri penarikan data log aktivitas otentikasi masuk secara penuh tanpa pembatasan halaman dari *database* MySQL melalui perantara objek model *LoginHistory*, sekaligus memuat data relasi nama serta *email* dari model *User*. Setelah pangkalan data mengembalikan seluruh koleksi data log audit operasional yang sah, kontroler mengemas data tersebut ke dalam *array* laporan dan meneruskannya ke dalam objek *DomPDF* untuk dikompilasikan berdasarkan basis struktur templat *cetak_login_history*. Sebagai luaran akhir dari skenario pemantauan keamanan ini, objek *DomPDF* secara instan menggenerasikan struktur *file* biner tersebut menjadi dokumen PDF utuh dan mengirimkannya kembali ke lapisan antarmuka pengguna, guna memicu browser pada komputer lokal milik *Admin* agar mengeksekusi perintah unduh otomatis *file* laporan ber-*timestamp* ke dalam media penyimpanan lokal.

c. Kelola user

1) Activity diagram kelola user

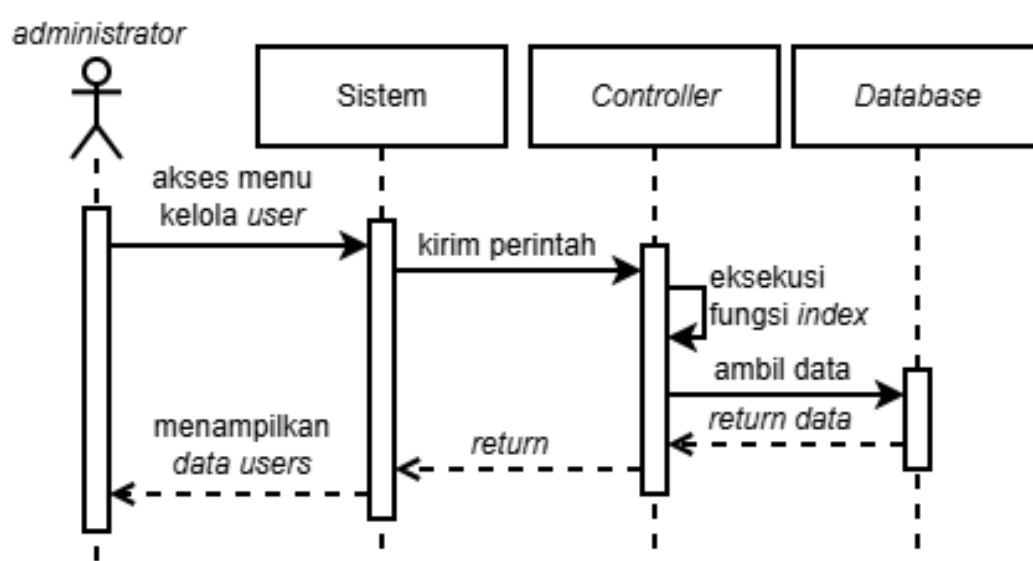


Gambar 5. 84 Activity Diagram Kelola User
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 84, diagram aktivitas tersebut diawali dari simpul hidup awal (*initial state*) oleh pihak sistem yang merespons dengan memicu perintah menerima *respons* & arahkan ke tampilan menu kelola *user*, yang kemudian memunculkan antarmuka pusat berupa tampilan kelola *user* di peramban *Admin*. Dari halaman kendali utama ini, *Admin* diberikan tiga jalur opsi tindakan manipulasi data reaktif yang saling bercabang. Jalur pertama, *Admin* dapat mengeklik perintah tekan tambah *user* yang langsung direspon oleh sistem melalui instruksi menerima *respons* & memproses *Pop-up Modal* tambah *user*. Jalur kedua, *Admin* dapat memilih perintah tekan edit yang ditanggapi secara dinamis oleh sistem lewat instruksi menerima *respons* &

memproses *Pop-up Modal* edit user. Jalur ketiga, *Admin* dapat mengeklik perintah tekan hapus, di mana sistem secara tanggap mengeksekusi instruksi menerima *respons* & memproses *Pop-up Modal* hapus user. Seluruh percabangan jalur manajemen pengelolaan akun ini secara linear akan bermuara kembali pada tampilan utama atau langsung menuju simpul penutup (*final state*) ketika aktivitas *Admin* dinyatakan selesai.

2) Sequence diagram kelola user



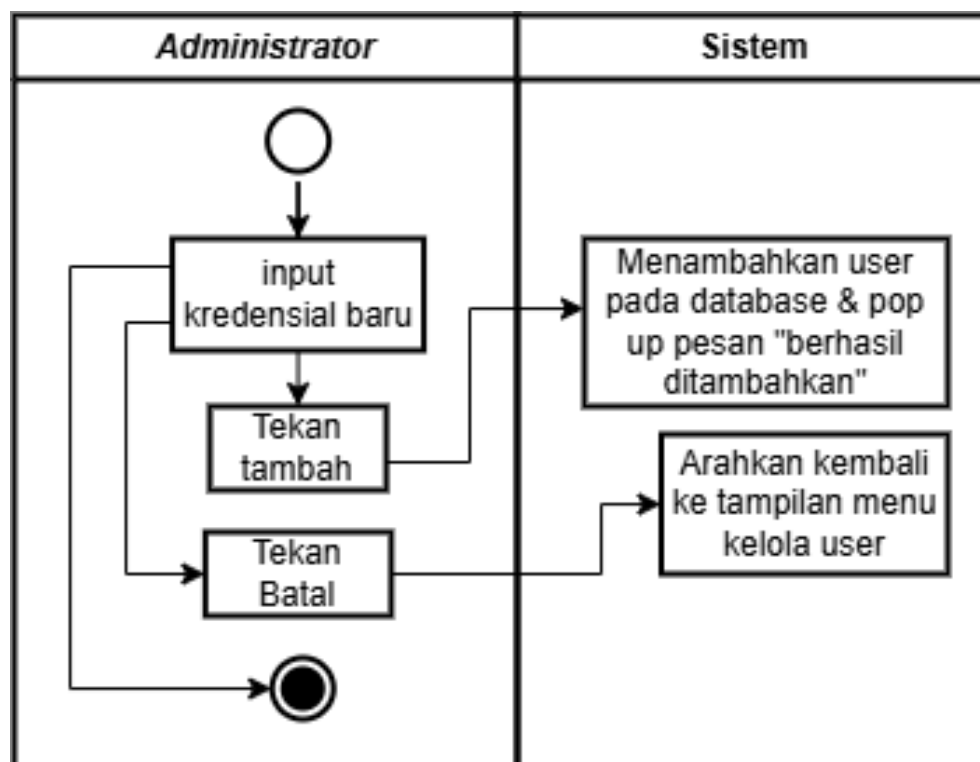
Gambar 5. 85 Sequence Diagram Kelola User
(Sumber: Data Pribadi)

Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 85, mengenai alur kelola data user oleh *Admin*, siklus interaksi diawali ketika aktor *Admin* berinteraksi dengan antarmuka halaman *Manajemen User View* untuk melakukan pembukaan data secara menyeluruh. Komponen antarmuka ini menindaklanjutinya dengan mengirimkan permintaan pemuatan data menuju objek *UserController* melalui pemanggilan fungsi *index*. Objek pengendali kemudian meneruskan instruksi tersebut untuk melakukan kueri pembacaan data akun pengguna secara terintegrasi ke dalam pangkalan data MySQL melalui perantara objek model *User*. Sesaat setelah basis data mengembalikan seluruh koleksi rekaman data *user* yang valid, kontroler memproses data tersebut untuk kemudian

merendernya kembali ke dalam bentuk tabel informasi pada antarmuka *Manajemen User View* agar dapat dipantau langsung oleh *Admin*.

d. Tambah user

1) *Activity diagram* tambah user

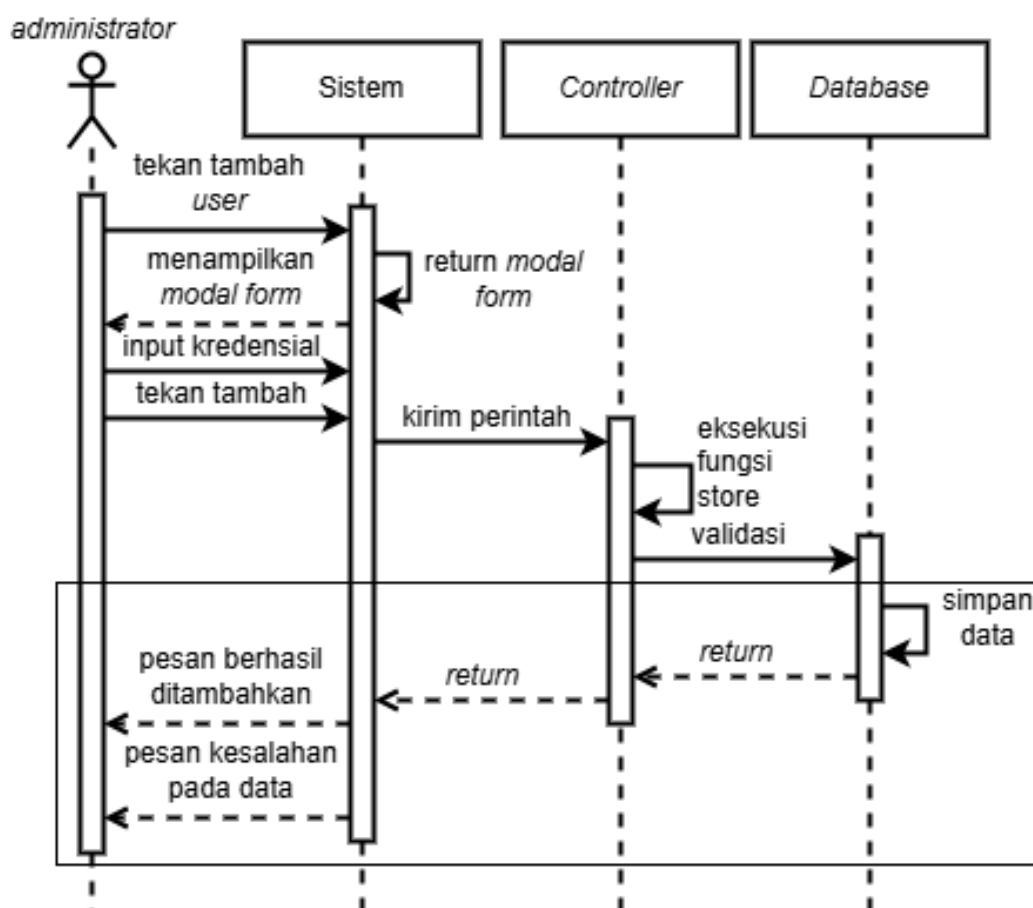


Gambar 5. 86 *Activity diagram* Tambah User
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 86, diagram aktivitas tersebut bermula ketika *Admin* membuka jendela modal aktif dan melaksanakan instruksi *input* kredensial baru. Setelah data terisi, *Admin* dihadapkan pada dua pilihan opsi tindakan pengondisian menu. Opsi tindakan pertama, jika *Admin* memutuskan untuk menyimpan data tersebut, *Admin* akan mengeklik perintah tekan tambah, di mana sistem langsung bekerja merespons secara permanen melalui fungsi menambahkan user pada database & pop up pesan "berhasil ditambahkan" sebelum menuju titik simpul penutup. Opsi tindakan kedua, apabila *Admin* ingin membatalkan transaksi pengisian, *Admin* cukup mengeklik perintah tekan batal, yang ditanggapi secara reaktif oleh sistem melalui instruksi

arahkan kembali ke tampilan menu kelola *user*. Aliran diagram aktivitas penambahan *user* ini resmi berakhir dan mencapai simpul akhir (*final state*) setelah halaman web dikembalikan ke kondisi sedia.

2) *Sequence diagram* tambah *user*



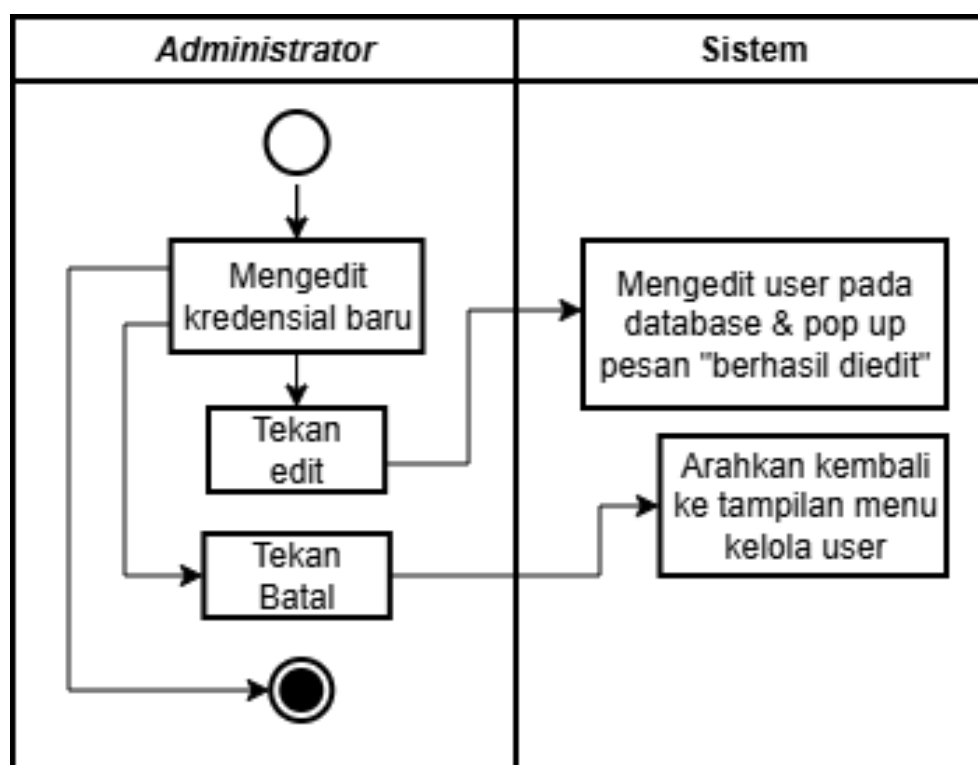
Gambar 5. 87 *Sequence Diagram* Tambah *User*
(Sumber: Data Pribadi)

Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 87, mengenai alur tambah data *user* oleh *Admin*, rangkaian interaksi dimulai ketika *Admin* menekan tombol tambah *user* pada antarmuka *Manajemen User View* untuk memunculkan modal formulir data baru. Setelah *Admin* mengisi seluruh data kredensial beserta hak akses pengguna dan menekan tombol simpan, komponen antarmuka secara responsif mengirimkan *payload data* tersebut menuju objek antarmuka *UserManagementController* melalui pemanggilan fungsi *store*. Objek pengendali mengesahkan seluruh parameter masukan melalui proses validasi sistem, lalu

memerintah objek model *User* untuk melakukan penulisan rekaman data baru ke dalam *database* MySQL. Sesaat setelah basis data berhasil mengunci baris data baru tersebut, pangkalan data mengembalikan sinyal konfirmasi sukses menuju kontroler, yang kemudian ditutup dengan pelemparan kembali pesan *session flash* sukses ke lapisan *Manajemen User View* agar antarmuka memunculkan notifikasi hijau tanda operasi penambahan pengguna berhasil diselesaikan seutuhnya.

e. Edit user

1) *Activity diagram* edit user

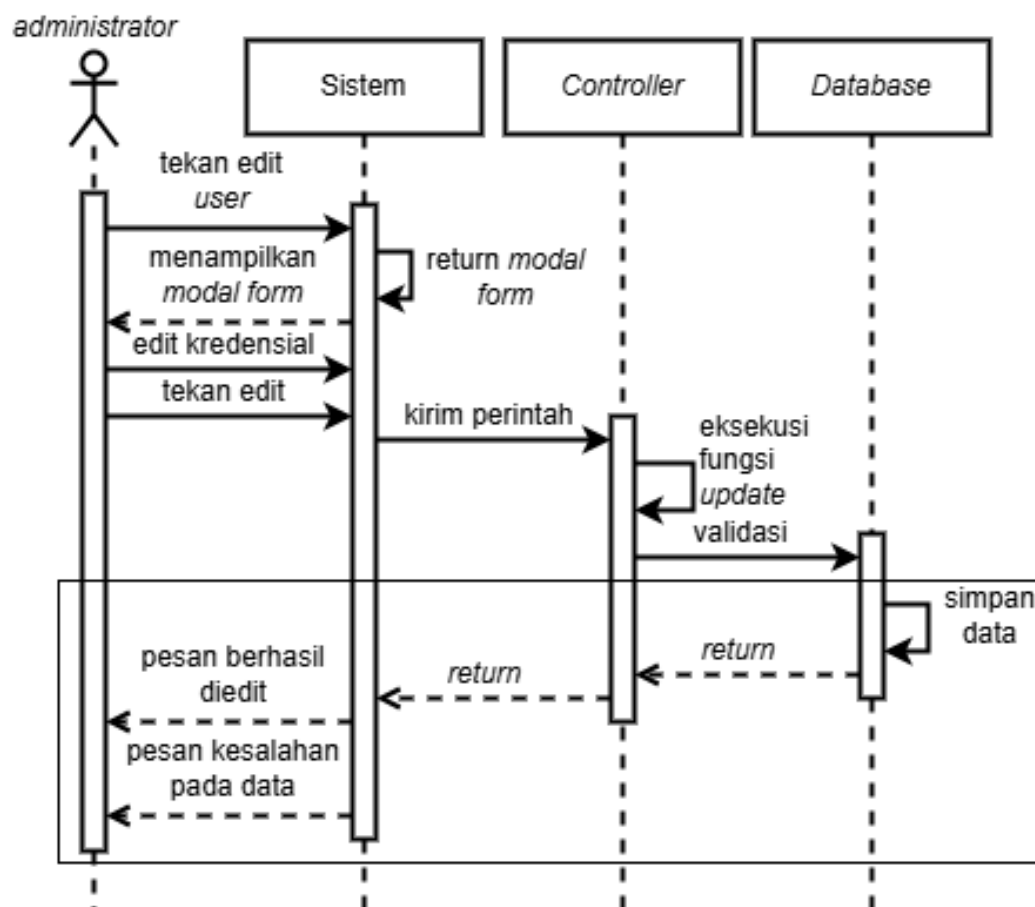


Gambar 5. 88 *Activity diagram* Edit User
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 88, diagram aktivitas tersebut diawali ketika *Admin* masuk ke dalam komponen modal penyuntingan data dan mengeksekusi tindakan mengedit kredensial baru. Tepat setelah perubahan parameter disesuaikan, *Admin* diberikan dua opsi pilihan tombol untuk mengeksekusi instruksi akhir. Opsi pilihan pertama, *Admin* dapat memicu penyimpanan pembaruan dengan mengklik perintah tekan edit, yang direspon

secara responsif oleh sistem melalui operasi server berupa mengedit *user* pada *database* & *pop up* pesan "berhasil diedit". Opsi pilihan kedua, *Admin* dapat menggagalkan perubahan data tersebut dengan mengeklik komponen tekan batal, di mana sistem secara otomatis menjalankan perintah arahkan kembali ke tampilan menu kelola *user*. Seluruh rangkaian manajemen pembaruan data pengguna ini dinyatakan selesai seutuhnya saat aliran data menyentuh simpul penutup (*final state*).

2) Sequence diagram edit user



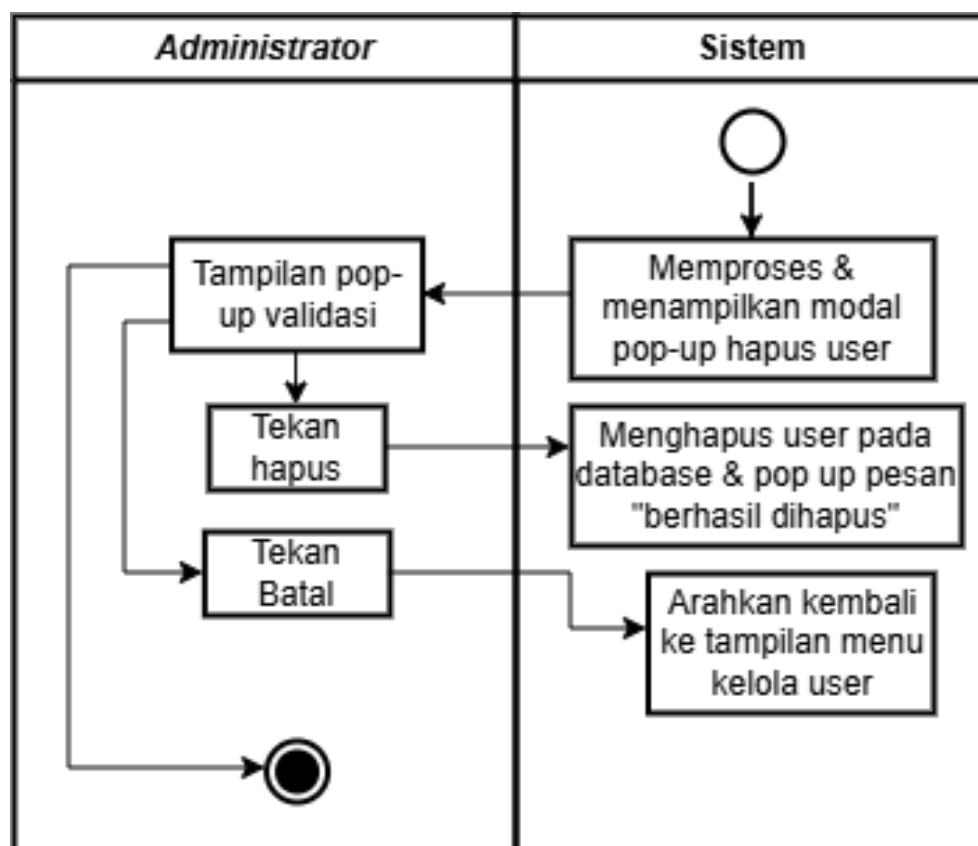
Gambar 5. 89 Sequence Diagram Edit User
(Sumber: Data Pribadi)

Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 89, mengenai alur ubah data user oleh *Admin* menggambarkan skenario pembaruan informasi akun pengguna secara berkala. Proses diawali ketika *Admin* mengeksekusi penekanan tombol *edit user* pada baris pengguna terpilih di

antarmuka *Manajemen User View*, yang memicu pengiriman parameter identitas unik menuju fungsi edit pada *UserManagementController* guna memuat data spesifik dari model *User* di *database* dan menampilkannya ke dalam formulir ubah data. Begitu *Admin* memperbarui informasi pada formulir tersebut dan menekan tombol simpan, antarmuka menyerahkan *payload data* baru menuju fungsi *update* pada objek *UserManagementController*. Komponen pengendali melakukan validasi ulang sebelum akhirnya memerintahkan model *User* untuk memperbarui baris data terkait di dalam *database* MySQL. Siklus manipulasi ini diakhiri dengan pengiriman konfirmasi sukses dari pangkalan data menuju kontroler, yang ditutup dengan pengembalian pesan sukses ke lapisan *Manajemen User View* agar antarmuka memperbarui tampilan tabel secara dinamis.

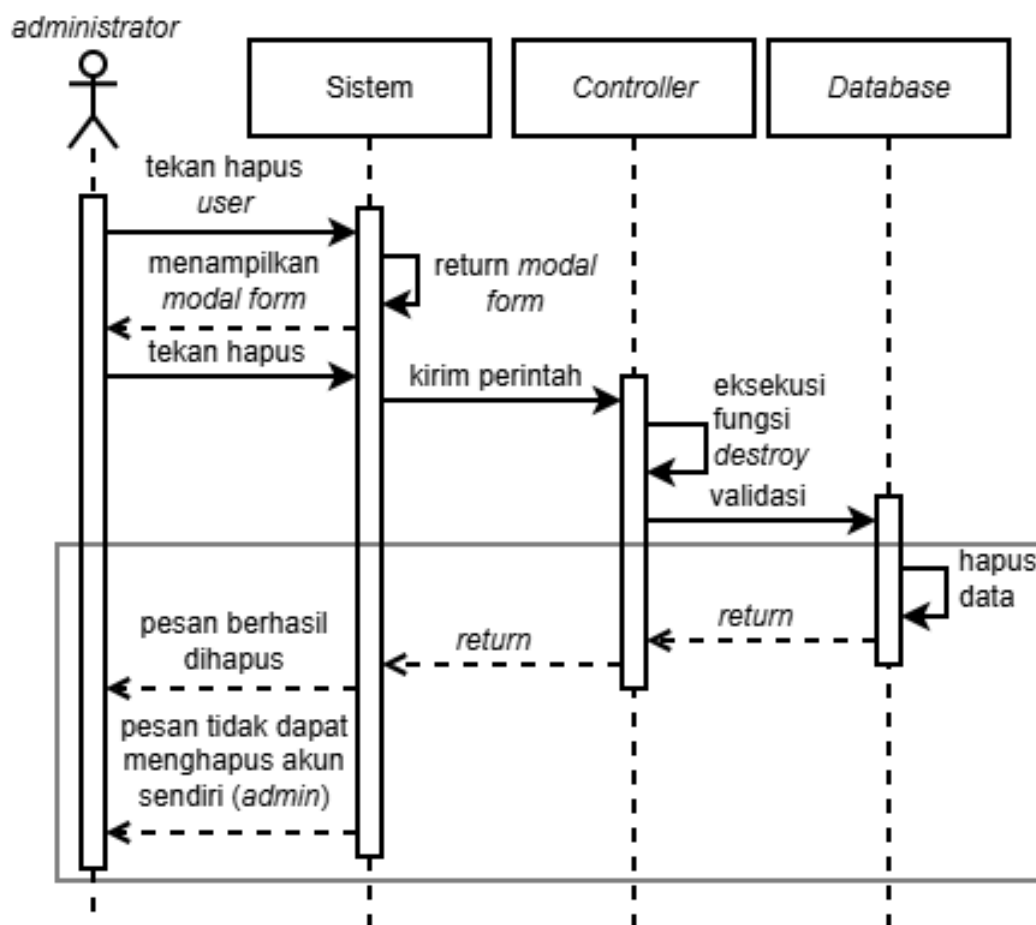
f. Hapus user

1) Activity diagram hapus user



Gambar 5. 90 Activity Diagram Hapus User
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 90, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) di bawah otoritas sistem yang langsung memproses & menampilkan modal *pop-up* hapus *user* di layar *browser*. *Admin* kemudian menerima konfirmasi tersebut dalam bentuk antarmuka tampilan *pop-up* validasi dan diberikan dua opsi tindakan keputusan. Opsi tindakan pertama, jika *Admin* setuju untuk melenyapkan akun tersebut, *Admin* mengeklik perintah tekan hapus, yang ditangkap oleh *backend* sistem untuk memproses fungsi menghapus *user* pada *database* & *pop up* pesan "berhasil dihapus". Opsi tindakan kedua, jika *Admin* membatalkan penghapusan, *Admin* mengeklik perintah tekan batal yang akan direspon oleh sistem lewat instruksi arahkan kembali ke tampilan menu kelola *user*. Aliran manajemen penghapusan *records data* ini dinyatakan selesai dan ditutup secara resmi tepat pada simpul akhir (*final state*).

2) *Sequence diagram hapus user*

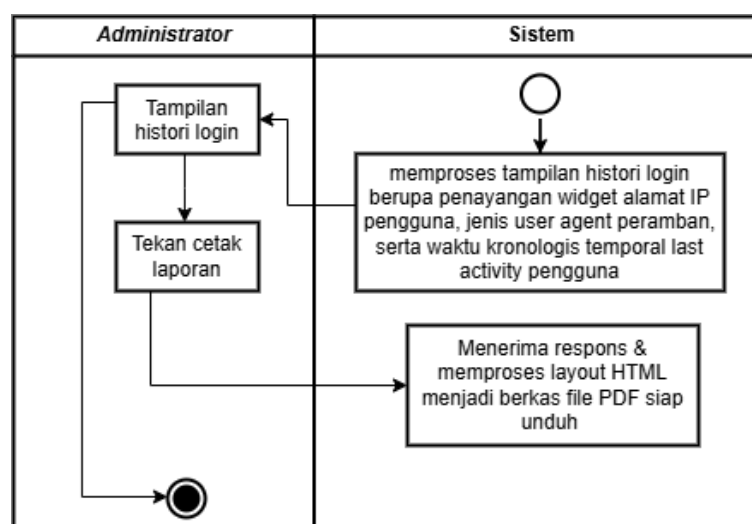
Gambar 5. 91 *Sequence Diagram Hapus User*
(Sumber: Data Pribadi)

Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 91, mengenai alur hapus data *user* oleh *Admin* menunjukkan mekanisme pembersihan akun pengguna dari sistem. Skenario aktivitas berjalan ketika *Admin* memilih salah satu target pengguna dan mengeksekusi penekanan tombol hapus pada antarmuka *Manajemen User View*, yang secara instan memicu pemanggilan fungsi *destroy* menuju objek *UserManagementController* dengan menyertakan parameter pengenalan unik *user* terkait. Objek pengendali langsung menindaklanjuti permintaan tersebut dengan mengirimkan instruksi penghapusan data secara searah ke dalam *database MySQL* melalui perantara objek model *User*. Setelah pangkalan data *MySQL* berhasil membuang baris rekaman data yang dituju dan mengembalikan status sukses, kontroler menangkap sinyal tersebut dan melemparkan kembali pesan

session flash berhasil menuju antarmuka *Manajemen User View*, sehingga sistem secara otomatis memperbarui struktur tabel dan menampilkan notifikasi sukses berwarna hijau di layar komputer *Admin*.

g. Histori login

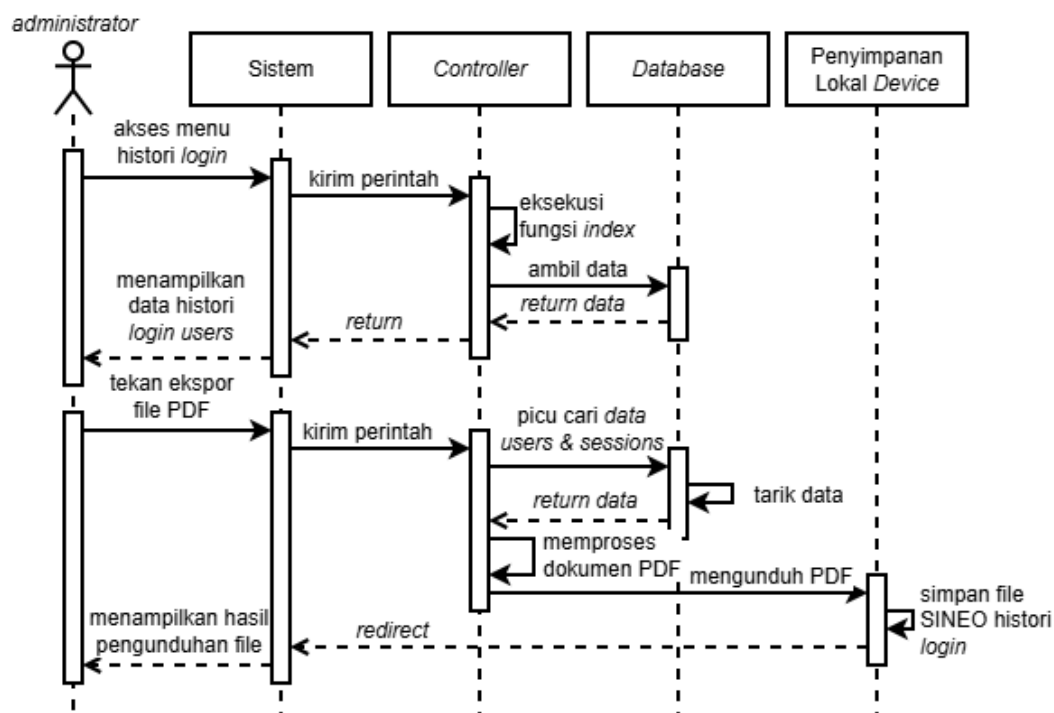
1) Activity diagram histori login



Gambar 5. 92 Activity Diagram Kelola Histori Login
(Sumber: Data Pribadi)

Berdasarkan pemodelan alur kerja (*workflow*) yang disajikan pada Gambar 5. 92, diagram aktivitas tersebut dikendalikan dari simpul awal (*initial state*) di sisi sistem yang memproses kueri kustom berupa memproses tampilan histori login berupa penayangan *widget* alamat IP pengguna, jenis *user agent* peramban, serta waktu kronologis temporal *last activity* pengguna. Hasil kompilasi data jejak audit digital tersebut disalurkan ke sisi pengguna otoritas dalam bentuk halaman tampilan histori login. Dari halaman visual tersebut, *Admin* menindaklanjutinya dengan mengeklik perintah operasional tekan cetak laporan. Sinyal perintah ekspor ini diteruskan kembali menuju server, di mana pihak Sistem bekerja menanggapi *request* lewat fungsi menerima *respons* & memproses *layout* HTML menjadi berkas *file* PDF siap unduh. Aliran keseluruhan diagram aktivitas khusus pemantauan log keamanan ini dinyatakan selesai seutuhnya saat menyentuh simpul penutup (*final state*).

2) Sequence diagram histori login



Gambar 5. 93 Sequence Diagram Histori Login
(Sumber: Data Pribadi)

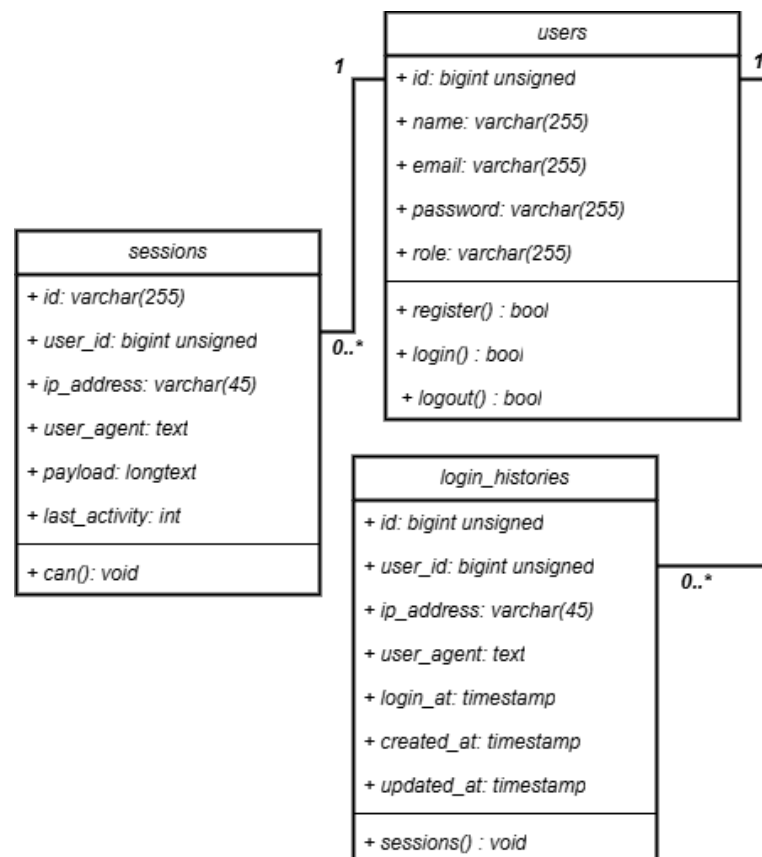
Berdasarkan rancangan interaksi terpadu yang disajikan pada Gambar 5. 93, mengenai alur pemantauan histori login oleh Admin menjamin aspek transparansi dan audit keamanan di dalam sistem. Proses dimulai saat Admin membuka menu rekaman aktivitas pada antarmuka *Login History View*, yang seketika memicu pengiriman sinyal permintaan data menuju objek *LoginHistoryController* melalui pemanggilan fungsi *index*. Objek pengendali memproses permintaan tersebut dengan melakukan kueri pembacaan data log aktivitas otentikasi masuk dari *database MySQL* menggunakan perantara objek model *LoginHistory*, sekaligus memuat data relasi informasi nama serta *email* dari model *User*. Begitu *database MySQL* mengembalikan seluruh kumpulan data log audit operasional yang sah, kontroler merendernya kembali ke dalam bentuk tabel runtun waktu di dalam antarmuka *Login History View*, sehingga Admin dapat langsung memantau setiap

riwayat alamat IP, perangkat, serta waktu otentikasi pengguna secara *riil* dan komprehensif.

B. Class diagram

Di samping melakukan dekonstruksi aliran aktivitas melalui diagram aktivitas, fase *Sprint* pada kerangka kerja *Scrum* ini juga menetapkan struktur statis sistem melalui penyusunan *class diagram*. Pemodelan struktural ini bertujuan untuk memetakan secara eksplisit seluruh kelas objek komponen model dan representasi data yang beroperasi di dalam arsitektur sistem *monitoring* SINEO, lengkap dengan spesifikasi visibilitas atribut murni (berbasis tipe data *riil* pada berkas kueri SQL), metode fungsi operasional (*methods*), serta derajat hubungan keterikatan (*multiplicity*) antar-entitas di dalam lingkungan *framework* Laravel 12.

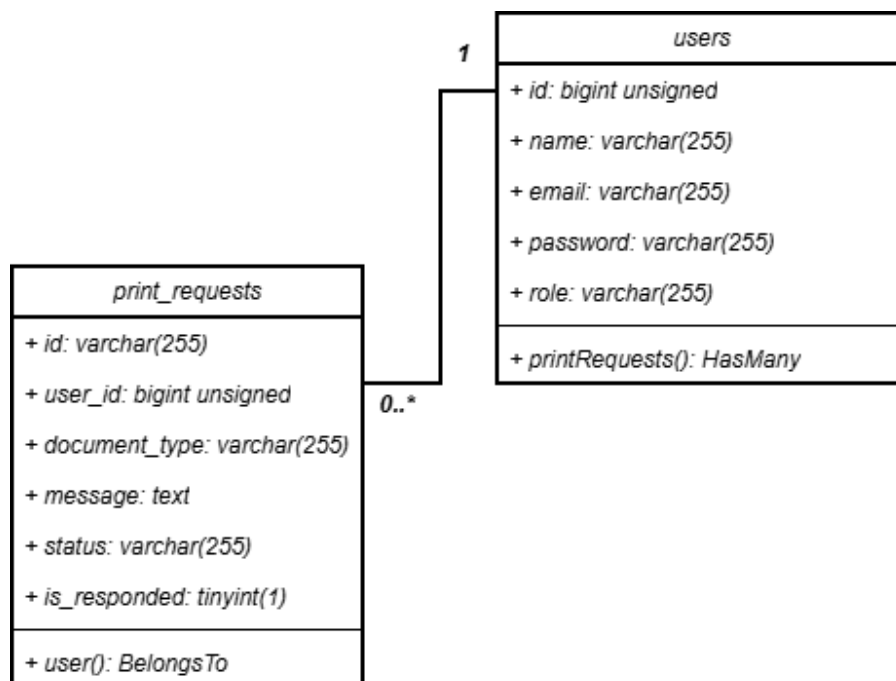
1) Class diagram histori login



Gambar 5. 94 Class Diagram Histori Login SINEO
(Sumber: Data Pribadi)

Berdasarkan manifestasi struktur diagram kelas pada Gambar 5. 94, diagram kelas ini memetakan rancangan pemrograman berorientasi objek yang mengendalikan pencatatan jejak digital otentikasi masuk pengguna (*audit trail*) demi menjaga integritas keamanan pada *platform* SINEO. Arsitektur diagram ini menempatkan kelas pengontrol *LoginHistoryController* sebagai komponen pusat yang bertindak sebagai pengendali utama aliran *data log* aktivitas. Di dalam kelas pengontrol ini, didefinisikan metode publik berupa fungsi *index()* yang bertanggung jawab untuk memicu penarikan data histori otentikasi dari pangkalan data, melakukan pengurutan rekaman secara kronologis terbalik, serta mengalirkan data tersebut menuju komponen antarmuka *Admin* dalam format terpaginasi. Kelas pengontrol ini dirancang mewarisi (*inheritance relationship*) sifat dasar dari kelas komponen inti bawaan *framework* Laravel, yaitu kelas *controller*.

Guna menjamin ketertelusuran dan akurasi *log* audit, kelas *LoginHistoryController* dikonfigurasi memiliki hubungan ketergantungan (*dependency relation*) yang kuat terhadap kelas utilitas model *Eloquent LoginHistory*. Kelas model *LoginHistory* bertindak sebagai representasi cetak biru objek entitas fisik tabel *login_histories* di dalam *database* MySQL, yang bertugas mengemas properti data sensitif rekaman otentikasi masuk meliputi atribut *id*, kunci tamu *user_id*, *string* alamat protokol internet *ip_address*, informasi peramban dan perangkat *user_agent*, serta penanda waktu otomatis *created_at* dan *updated_at*. Dari sudut pandang relasi basis data relasional *phpMyAdmin*, hubungan antara kelas model *User* terhadap kelas model *LoginHistory* dimodelkan menggunakan nilai kardinalitas mutlak *one-to-many* (1 ke 0..*). Penentuan derajat *multiplicity* ini menegaskan bahwa satu baris rekaman identitas pada kelas model *User* dapat memiliki keterkaitan hubungan terhadap banyak (0..*) baris data rekaman histori aktivitas *login* dari waktu ke waktu, sedangkan satu (1) instansiasi baris *data log* pada kelas model *LoginHistory* wajib terikat secara independen pada satu identitas akun pengguna yang valid untuk keperluan akuntabilitas forensik digital sistem.

2) *Class diagram print request*

Gambar 5. 95 *Class Diagram Print Request SINEO*
(Sumber: Data Pribadi)

Berdasarkan manifestasi struktur diagram kelas pada Gambar 5. 95, diagram kelas ini memetakan rancangan pemrograman berorientasi objek yang mengendalikan siklus pengajuan, otorisasi, pencetakan dokumen katalog dan prediksi, hingga fungsi pengiriman notifikasi *email* otomatis pada *platform* SINEO. Arsitektur diagram ini menempatkan kelas pengontrol *RequestManagementController* sebagai komponen pusat yang bertindak sebagai pengendali utama aliran data. Di dalam kelas pengontrol ini, didefinisikan rentetan metode publik yang meliputi fungsi *index()* untuk menampilkan daftar antrean berkas bagi pihak *Administrator*, serta fungsi tindakan mutlak berupa *approve()* dan *reject()* untuk mengeksekusi konfirmasi status persetujuan dokumen secara hibrida melalui menu *Kelola Req Dokumen*. Kelas pengontrol ini dirancang mewarisi (*inheritance relationship*) sifat dasar dari kelas komponen inti *framework* Laravel, yaitu kelas *controller*.

Guna menjamin validitas dan keamanan muatan data yang dikirimkan melalui *form* antarmuka, kelas *RequestManagementController* memiliki hubungan

ketergantungan (*dependency relation*) yang kuat terhadap kelas utilitas *PrintRequest*. Kelas *PrintRequest* merupakan representasi model *Eloquent* objek entitas fisik tabel *print_requests* di dalam *database* MySQL, yang mengenkapsulasi properti data operasional seperti atribut *id*, kunci tamu *user_id*, string dokumen *document_type*, penanda status status, deskripsi alasan penolakan *rejection_reason*, serta jejak waktu *created_at* dan *updated_at*. Dari sudut pandang relasi basis data relasional *phpMyAdmin*, hubungan kelas model *User* terhadap kelas model *PrintRequest* memiliki nilai kardinalitas mutlak *one-to-many* (*1 ke 0..**). Penentuan derajat *multiplicity* ini menegaskan bahwa satu baris rekaman pada kelas model *User* dapat memiliki hubungan kepemilikan terhadap banyak (*0..**) baris rekaman dokumen pengajuan cetak, sedangkan satu (*1*) instansiasi baris data pada kelas model *PrintRequest* wajib terikat secara independen pada satu identitas pengguna yang valid guna menjaga akuntabilitas log audit keamanan sistem.

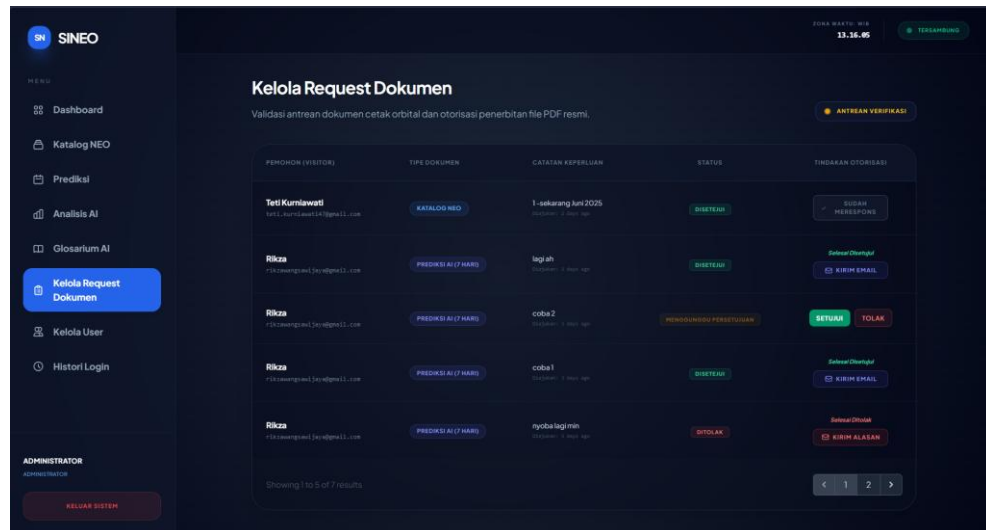
Integrasi hilir pada modul administrasi ini diselesaikan melalui jembatan dependensi terarah menuju komponen pengiriman lintas media yang melingkupi *file mailable* *DocumentDeliveryMail* dan *file RejectionNotificationMail*. Kedua *file mailable* tersebut ditugaskan secara modular untuk menyusun struktur muatan pesan elektronik, di mana *file DocumentDeliveryMail* memuat metode *content()* untuk merelasikan berkas lampiran luaran pustaka *DomPDF* dan metode *envelope()* untuk mendefinisikan subjek *email* tujuan, yang keduanya diturunkan secara hierarkis dari kelas induk bawaan *framework* yaitu *file Mailable*. Hubungan dependensi dari kelas pengontrol *RequestManagementController* menuju kedua *file mailable* tersebut memastikan bahwa sesaat setelah status *flag is_responded* bernilai benar (*true*) pada *database*, sistem secara otomatis mengeksekusi gerbang *Google SMTP Server* untuk mendistribusikan berkas PDF atau notifikasi pembatalan menuju kotak masuk *Gmail* pengguna secara *real-time*.

5.1.5.4.3. *Implementation*

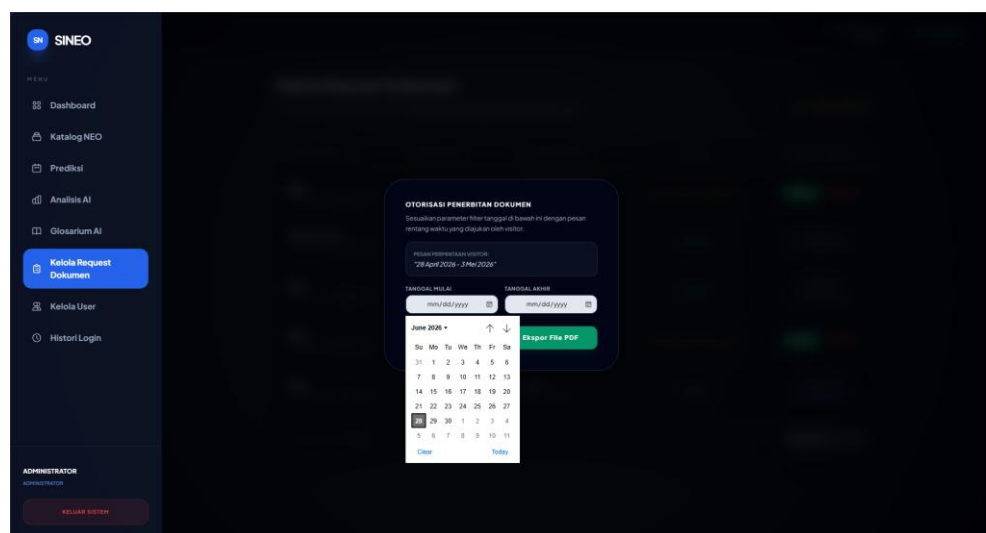
Sistem antarmuka ini bertindak sebagai media interaksi utama (*front-end*) yang menyajikan berbagai manajemen dari *user Visitor* yang dikelola langsung oleh *Admin*. Secara fungsional, struktur pengembangan komponen antarmuka

monitoring SINEO dirancang secara modular dan dibagi menjadi beberapa halaman utama yang didokumentasikan sebagai berikut:

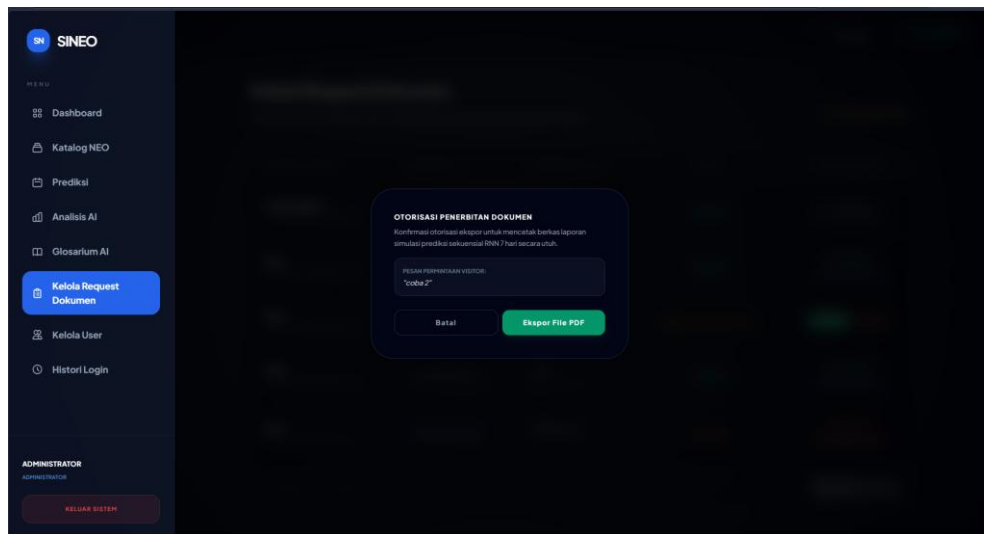
a. Halaman kelola *request* dokumen



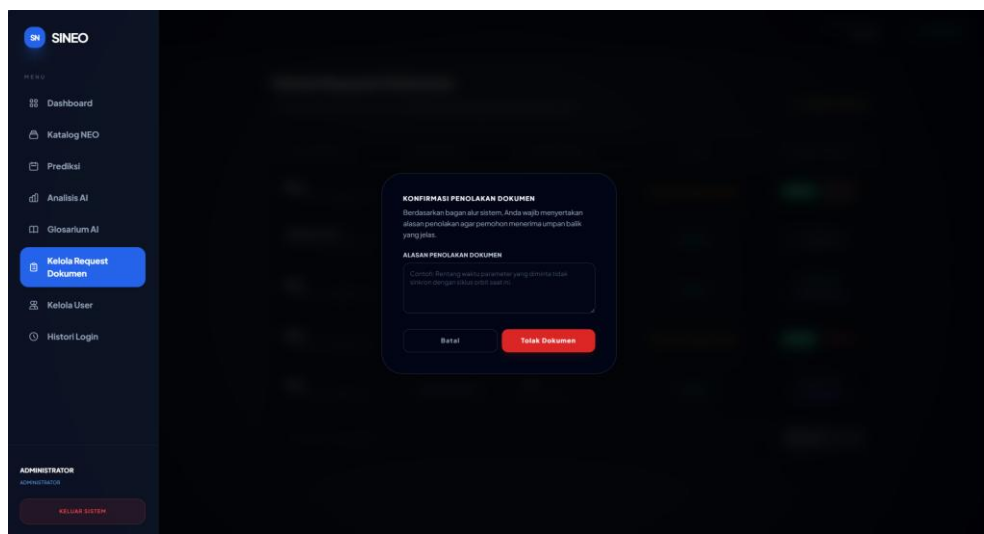
Gambar 5. 96 Antarmuka Kelola *Request* Dokumen SINEO
(Sumber: Data Pribadi)



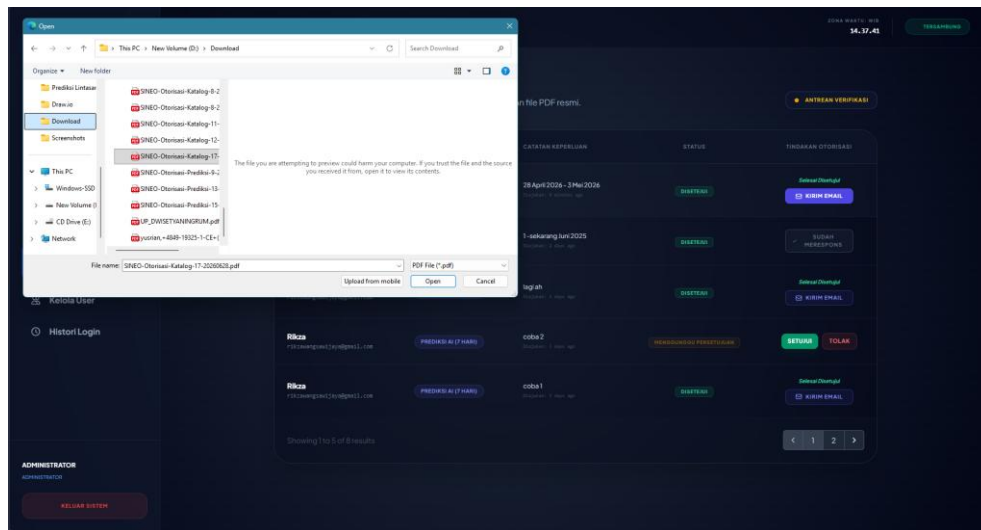
Gambar 5. 97 Antarmuka *Modal Form* Setujui Kelola
Request Dokumen SINEO (Ajuan Katalog NEO)
(Sumber: Data Pribadi)



Gambar 5. 98 Antarmuka *Modal Form* Setujui Kelola Request Dokumen SINEO (Ajuan Prediksi AI)
(Sumber: Data Pribadi)



Gambar 5. 99 Antarmuka *Modal Form* Tolak Kelola Request Dokumen SINEO (Ajuan Katalog/Prediksi)
(Sumber: Data Pribadi)



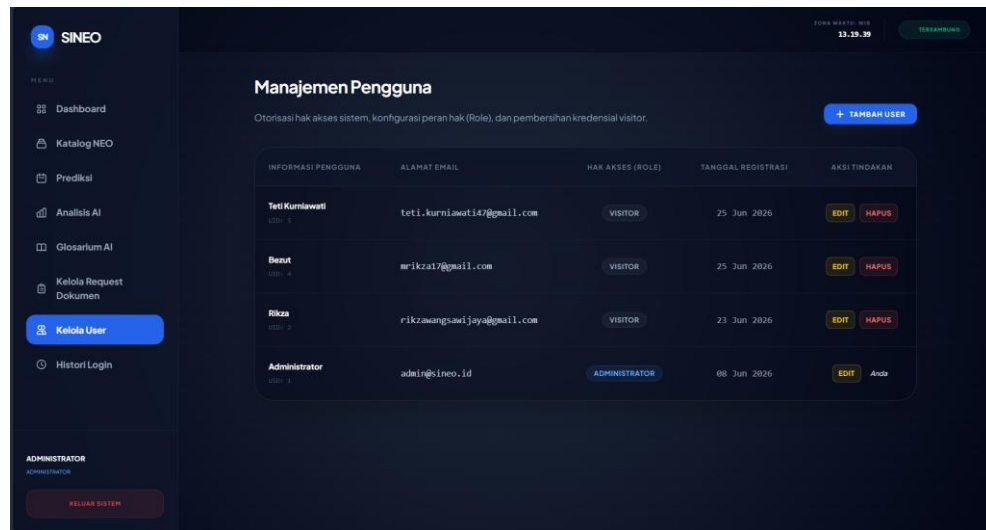
Gambar 5. 100 Antarmuka Kirim Gmail (Setuju/Tolak)
Kelola *Request* Dokumen SINEO
(Sumber: Data Pribadi)

Halaman *req_management.blade.php* dikembangkan khusus sebagai pusat kendali administratif sistem SINEO yang hanya dapat diakses secara eksklusif oleh pengguna dengan peran hak akses *Administrator*. Antarmuka ini dirancang untuk menampilkan visualisasi data tabular terintegrasi dari seluruh antrean permohonan cetak berkas biner yang diajukan oleh *Visitor* secara berkala. Guna menjaga efisiensi konsumsi memori lokal server, tabel utama pada halaman ini dilengkapi dengan fungsi paginasi (*pagination*) dinamis bawaan Laravel 12 yang membatasi pemuatan data sebanyak lima baris pengajuan per halaman, yang diurutkan secara runtun waktu terbalik (*descending*) berdasarkan waktu pembuatan. Representasi visual tabel ini merekam data fisik secara transparan dari relasi pangkalan data MySQL meliputi nama lengkap pemohon, tipe dokumen sains yang diajukan (katalog atau prediksi), pesan catatan deskripsi keperluan khusus dari pihak *Visitor*, serta label indikator status pengkondisian antrean dengan visualisasi warna kontras dinamis guna mempermudah pemantauan siklus validasi berkas.

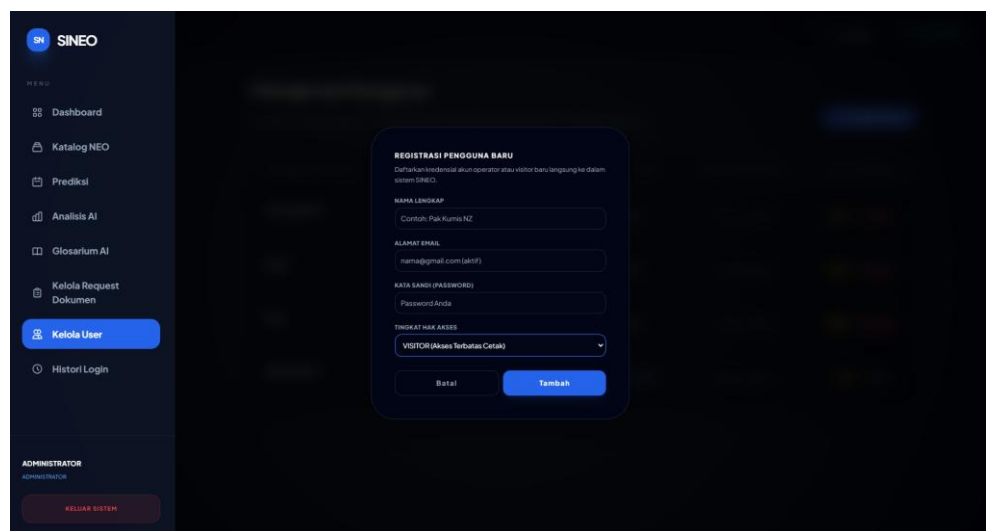
Di dalam arsitektur operasional terbarunya, antarmuka kelola permohonan ini menyediakan empat aksi interaktif mandiri yang terikat penuh pada fungsi penanganan kelas *RequestManagementController*. Aksi pertama diakomodasi melalui tombol setuju yang memicu fungsi *approve* untuk memproses dokumen

secara kondisional, di mana sistem akan membuka elemen modal parameter *filter* tanggal untuk tipe katalog guna menarik data *riil Near-Earth Objects* dari objek model *NeoObject*, sedangkan untuk tipe prediksi sistem akan memuat data proyeksi sepekan ke depan dari penyimpanan *cache* dan *backup* JSON beserta metrik persentase akurasi murni latih model RNN dari log pelatihan. Setelah status baris data permohonan berhasil diperbarui menjadi disetujui pada model *PrintRequest* di *database* MySQL, kontroler langsung membersihkan *buffer* memori secara paksa dan memuat seluruh *payload* laporan ke dalam objek *DomPDF* melalui *template* visual otorisasi khusus untuk menghasilkan *file* berkas biner PDF resmi yang langsung terunduh secara otomatis ke penyimpanan lokal komputer *Admin*.

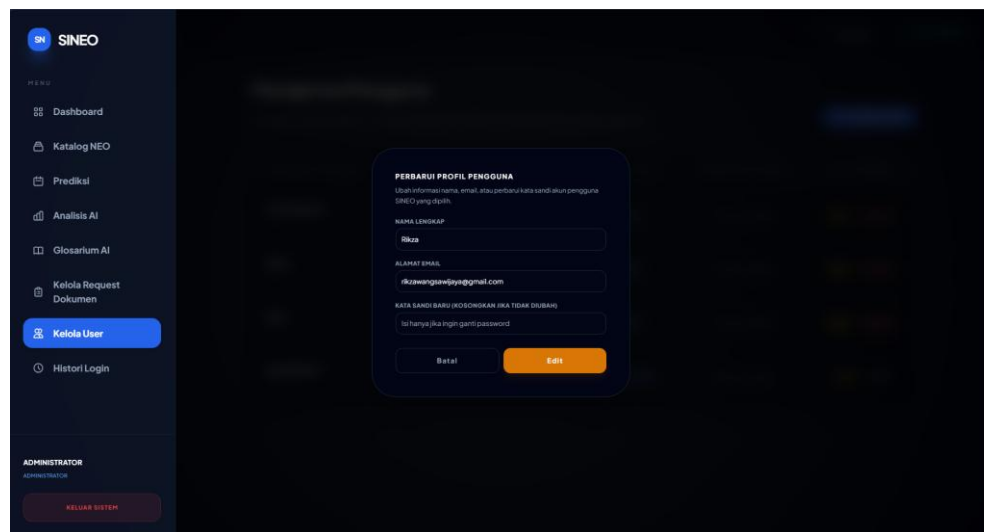
Aksi kedua diwadahi oleh tombol tolak yang memicu fungsi *reject* untuk menolak pengajuan yang dianggap tidak valid oleh *Administrator*, di mana sistem akan menyajikan modal formulir *input string* teks alasan evaluasi yang wajib diisi. Begitu data dikirimkan, kontroler melakukan validasi ketat dan memperbarui status baris data permohonan menjadi ditolak sekaligus mengunci *string* teks argumen evaluasi tersebut ke dalam kolom catatan pangkalan data MySQL sebelum melemparkan kembali pesan *session flash* sukses berwarna hijau di layar. Selanjutnya, aksi ketiga dan keempat diakomodasi secara dinamis melalui tombol kirim *email* dan tombol kirim alasan yang terhubung langsung menuju fungsi *sendEmail* untuk mendistribusikan notifikasi keluar ke alamat *Gmail Visitor* memanfaatkan *Google SMTP Server* dengan otentikasi sandi aplikasi yang sah. Tombol kirim *email* bertugas memvalidasi pengunggahan manual berkas fisik PDF hasil cetak *database* oleh *Admin* untuk diteruskan lewat kelas *mailable DocumentDeliveryMail*, sedangkan tombol kirim alasan bertugas menarik *string* catatan alasan penolakan secara instan dari *database* untuk ditembakkan lewat kelas *mailable RejectionNotificationMail*, di mana kedua alur ini secara cerdas akan langsung memperbarui status *is_responded* menjadi benar di *database* guna mengunci fungsi tombol antarmuka menjadi abu-abu tidak aktif secara permanen demi menghindari terjadinya duplikasi respon pengiriman berkas.

b. Halaman kelola *user*

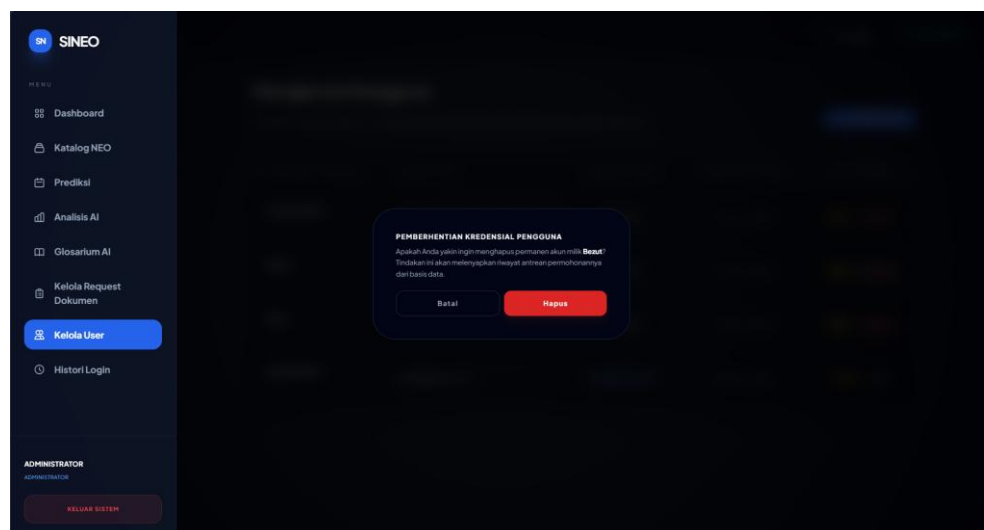
Gambar 5. 101 Antarmuka Kelola User SINEO
(Sumber: Data Pribadi)



Gambar 5. 102 Antarmuka Kelola Tambah User SINEO
(Sumber: Data Pribadi)



Gambar 5. 103 Antarmuka Kelola Edit *User* SINEO
(Sumber: Data Pribadi)



Gambar 5. 104 Antarmuka Kelola Hapus *User* SINEO
(Sumber: Data Pribadi)

Halaman *user_management.blade.php* dikembangkan sebagai modul administratif khusus di dalam sistem SINEO yang berfungsi untuk memfasilitasi peran *Administrator* dalam mengelola seluruh data akun pengguna yang terdaftar. Antarmuka ini dirancang dengan menyajikan visualisasi data tabular terintegrasi yang memuat informasi fisik secara rapi dari struktur tabel pangkalan data MySQL meliputi nama lengkap pengguna, alamat *email* aktif, status verifikasi akun, penanda waktu pembuatan akun (*timestamp*), serta visualisasi label peran hak akses

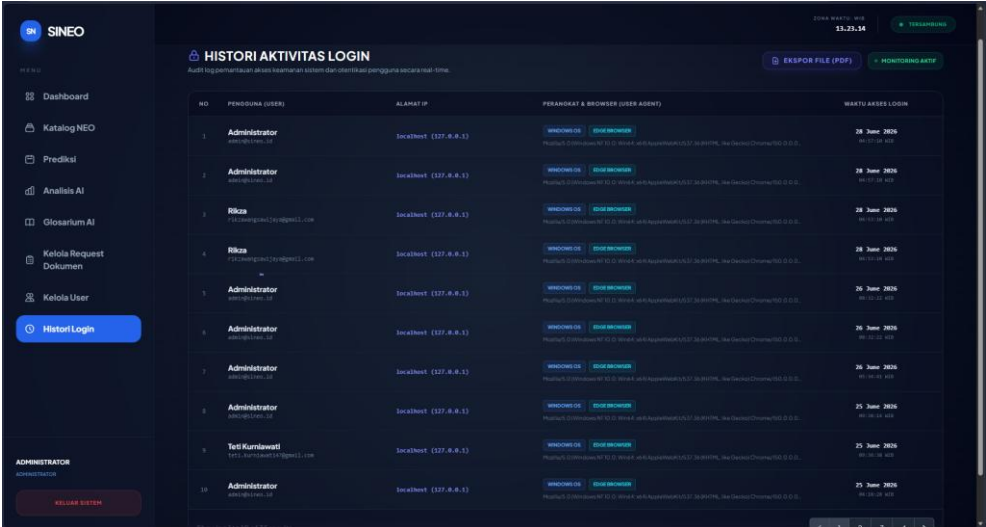
(role) baik sebagai *Administrator* maupun *Visitor*. Guna menjaga kestabilan pemuatan memori lokal pada server saat menangani volume data yang besar, tabel utama pada halaman manajemen ini dilengkapi dengan fungsi paginasi (*pagination*) dinamis bawaan Laravel 12. Selain itu, antarmuka ini menyediakan tombol operasional utama berupa tambah *user* baru pada sudut kanan atas halaman, serta tombol aksi edit dan hapus pada setiap baris data pengguna guna mendukung eksekusi siklus hidup manajemen data yang bersifat interaktif.

Secara arsitektural, fungsionalitas pengolahan data pada antarmuka manajemen pengguna ini terikat penuh pada fungsi penanganan di dalam kelas *UserManagementController* melalui pemanfaatan operasi CRUD secara terisolasi. Operasi penambahan data diawali ketika *Admin* mengeksekusi penekanan tombol tambah *user* baru yang secara responsif memicu kemunculan elemen modal formulir input data kredensial baru. Begitu formulir disubmit, data dilemparkan menuju fungsi *store* pada kontroler untuk divalidasi secara ketat sebelum diperintahkan menuju model *User* untuk mengunci rekaman baris baru di dalam *database* MySQL. Skenario yang serupa juga diterapkan pada proses modifikasi data, di mana penekanan tombol edit akan melempar parameter identitas unik pengguna menuju fungsi *edit* untuk menarik data spesifik dari *database*, merendernya ke dalam formulir modal ubah data, dan mengeksekusi pembaruan baris data ke pangkalan data melalui fungsi *update* setelah *Admin* menekan tombol simpan perubahan.

Sementara itu, operasi pembersihan hak akses atau penghapusan data pengguna diakomodasi secara aman melalui tombol hapus pada tabel antarmuka. Aktivitas ini secara instan memicu fungsi *destroy* pada objek *UserManagementController* dengan menyertakan parameter pengenalan unik user terkait ke dalam kueri penghapusan *database* MySQL setelah *Admin* melewati jendela konfirmasi keamanan sistem. Seluruh siklus manipulasi data pengguna pada antarmuka ini diakhiri dengan mekanisme pengembalian sinyal konfirmasi sukses dari pangkalan data menuju kontroler, yang kemudian diteruskan kembali dalam bentuk pesan *session flash* sukses berwarna hijau di bagian atas layar antarmuka *Manajemen*

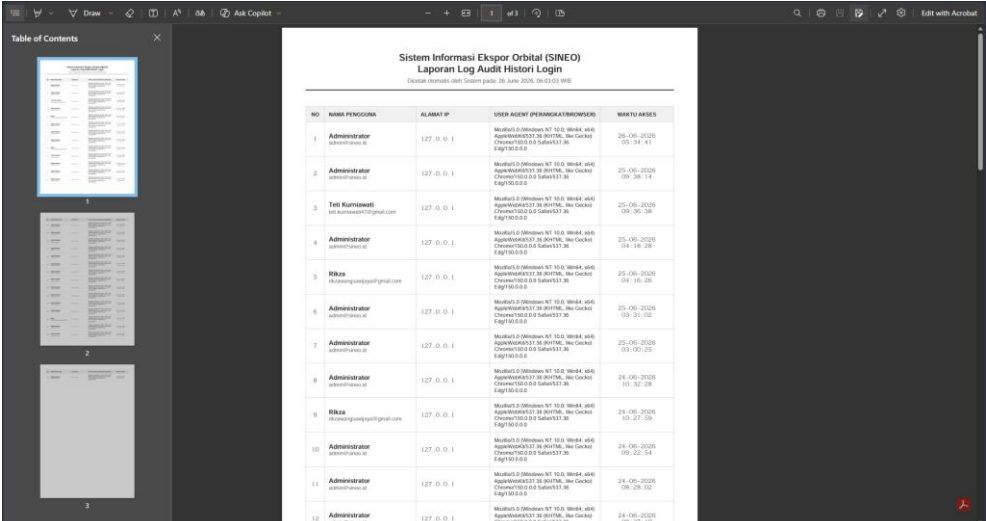
User View secara dinamis pertanda operasi tambah, ubah, maupun hapus data pengguna telah berhasil diselesaikan secara utuh oleh sistem.

c. Halaman histori *login*



NO	PENGGUNA (USER)	ALAMAT IP	PERANGKAT & BROWSER (USER AGENT)	WAKTU ASSES LOGIN
1	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
2	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
3	Rika rika.singapore@gmail.com	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
4	Rika rika.singapore@gmail.com	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
5	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
6	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
7	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
8	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
9	Teti Kurniawati tetikurniawati@gmail.com	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB
10	Administrator administrator@neo.id	localhost (127.0.0.1)	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28 June 2025 00:10:14 WIB

Gambar 5. 105 Antarmuka Kirim Gmail (Setuju/Tolak)
Kelola *Request* Dokumen SINEO
(Sumber: Data Pribadi)



NO	PENGGUNA (USER)	ALAMAT IP	USER AGENT (PERANGKAT & BROWSER)	WAKTU ASSES
1	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
2	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
3	Teti Kurniawati tetikurniawati@gmail.com	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
4	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
5	Rika rika.singapore@gmail.com	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
6	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
7	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
8	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
9	Rika rika.singapore@gmail.com	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
10	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
11	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14
12	Administrator administrator@neo.id	127.0.0.1	Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0	28-06-2025 00:10:14

Gambar 5. 106 Antarmuka Kirim Gmail (Setuju/Tolak)
Kelola *Request* Dokumen SINEO
(Sumber: Data Pribadi)

Halaman *login_history.blade.php* dikembangkan sebagai modul audit keamanan dan pemantauan aktivitas otentikasi digital yang dapat diakses secara transparan

oleh *Administrator* di dalam ekosistem SINEO. Antarmuka ini dirancang untuk menyajikan visualisasi data tabular kronologis yang memuat rekaman fisik secara rapi dari struktur tabel pangkalan data MySQL meliputi nama lengkap pengguna, alamat *email* aktif, alamat IP (*IP Address*) perangkat yang digunakan, informasi *user agent* browser, serta penanda waktu otentikasi masuk secara presisi (*timestamp*). Guna memastikan performa pemuatan antarmuka tetap stabil dan terhindar dari kendala *overhead* memori lokal server akibat lonjakan *data log* yang masif, tabel utama pada modul audit ini telah dilengkapi dengan fungsi paginasi (*pagination*) otomatis bawaan Laravel 12. Kehadiran halaman histori *login* ini memperkuat aspek akuntabilitas sistem, sehingga *Administrator* dapat melakukan deteksi dini terhadap potensi anomali akses ilegal atau aktivitas otentikasi tidak sah dari pihak luar.

Secara fungsional, manipulasi dan penarikan data log keamanan pada antarmuka ini dikendalikan penuh oleh fungsi penanganan di dalam kelas *LoginHistoryController*. Selain memfasilitasi pemuatan data *riil* ke dalam tabel visual melalui fungsi *index*, berkas antarmuka ini juga menyediakan fitur pencetakan laporan mandiri berupa tombol ekspor *file* PDF pada sudut kanan atas halaman untuk kebutuhan dokumentasi fisik atau laporan berkala manajemen keamanan instansi. Ketika tombol ekspor tersebut dieksekusi oleh *Administrator*, antarmuka akan mengirimkan permintaan pembuatan dokumen laporan audit menuju fungsi *exportPdf* pada objek *LoginHistoryController*, yang secara instan memicu kueri penarikan seluruh *data log* aktivitas masuk tanpa batasan halaman dari *database* MySQL dengan memuat relasi data dari model *User*. Setelah pangkalan data mengembalikan seluruh koleksi data log audit operasional yang sah, kontroler mengemas data tersebut ke dalam *array* laporan dan meneruskannya ke dalam objek *DomPDF* untuk dikompilasikan berdasarkan basis struktur templat *cetak_login_history*. Siklus operasi ini diakhiri dengan penggenerasian *file* biner PDF utuh yang dikirimkan kembali ke lapisan antarmuka pengguna, guna memicu browser pada komputer lokal milik *Admin* agar mengeksekusi perintah unduh otomatis berkas laporan keamanan ber-*timestamp* tersebut ke dalam media penyimpanan lokal.

5.1.5.4.4. Review

Tahapan *Sprint Review* pada akhir *sprint* keempat ini ditujukan untuk mengevaluasi fungsionalitas modul administratif tingkat tinggi, sistem otorisasi dokumen, serta penguatan audit keamanan web menggunakan metode *Black Box Testing*. Pengujian *Black Box* dieksekusi untuk memvalidasi interaksi fungsional antara antarmuka administratif dengan pemrosesan *back-end* berdasarkan skenario masukan dan keluaran (*input-output*) dari sudut pandang *Administrator*, tanpa memeriksa arsitektur internal *controller* secara mendetail. Proses peninjauan dilakukan melalui mekanisme *Client Validation Loop* secara terbuka melalui pengujian fungsionalitas (*Black Box Testing*) langsung oleh pengguna umum (*Visitor*) dan pengelola sistem (*Administrator*). Fokus utama evaluasi tertuju pada keandalan sistem dalam mengunci status respons (*flag is_responded*) pada pengajuan dokumen cetak, integrasi pengiriman notifikasi persetujuan/penolakan secara *real-time* via Google SMTP *Gmail Mailer*, ketepatan kompilasi *layout* HTML menjadi dokumen biner PDF berukuran A4 via pustaka *DomPDF*, keabsahan fungsionalitas CRUD pada manajemen pengguna (Kelola *User*), hingga kemampuan sistem dalam menangkap serta mengeksport parameter *IP User Agent* pada log aktivitas *Histori Login*.

Guna membuktikan ketercapaian fungsionalitas sistem otorisasi administratif, pencetakan dokumen, dan rekam jejak audit keamanan pada *Sprint 4* secara empiris, dilakukan pengujian masukan dan keluaran (*input-output*) menggunakan metode *Black Box Testing*. Rangkuman hasil pengujian *Black Box* untuk seluruh fitur manajemen data pada *Sprint 4* disajikan secara terperinci pada Tabel 5. 20 di bawah ini:

Tabel 5. 20 Hasil Pengujian *Black Box Testing Sprint 4*

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
1.	Modul kelola <i>request</i> dokumen (aksi setuju / <i>approve</i>)	<i>Admin</i> mengeklik tombol "Setujui" pada antrean <i>request</i> dokumen dari <i>Visitor</i> .	Sistem menampilkan modal parameter <i>filter</i> tanggal untuk tipe katalog atau memuat <i>cache/json</i> untuk tipe prediksi, memperbarui status menjadi <i>approved</i> , menggenerasikan berkas biner via <i>DomPDF</i> , dan mengunduh <i>file</i> PDF secara otomatis pada browser lokal <i>Admin</i> .	Sistem berhasil melakukan validasi <i>filter</i> tanggal, mengubah status basis data menjadi disetujui, dan mengunduh berkas biner PDF katalog/prediksi secara otomatis.	Sesuai / Valid	08 Juni 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
2.	Modul kelola <i>request</i> dokumen (aksi tolak / <i>reject</i>)	<i>Admin</i> mengeklik tombol "Tolak" pada antrean <i>request</i> dan mengisi <i>string</i> alasan pada modal evaluasi.	Sistem memvalidasi <i>input</i> teks alasan penolakan, mengubah status baris permohonan menjadi <i>rejected</i> , menyimpan alasan ke <i>database</i> , serta memunculkan notifikasi <i>session flash</i> sukses berwarna hijau.	Sistem berhasil mengunci alasan evaluasi, memperbarui status menjadi ditolak, dan memicu notifikasi sukses di antarmuka <i>Admin</i> .	Sesuai / Valid	08 Juni 2026
3.	Modul kelola <i>request</i> dokumen (aksi kirim <i>gmail</i> berkas)	<i>Admin</i> mengeklik tombol "Kirim <i>Email</i> " pada <i>request</i> yang disetujui dan mengunggah berkas PDF secara manual.	Sistem memvalidasi ekstensi serta ukuran berkas, mengirimkan <i>email</i> lampiran PDF ke <i>Gmail Visitor</i> melalui <i>Google SMTP Server</i> via <i>DocumentDeliveryMail</i> , dan memperbarui <i>flag is_responded</i> menjadi benar (<i>true</i>).	Berkas PDF berhasil divalidasi, terkirim utuh ke alamat <i>Gmail Visitor</i> , dan tombol aksi otomatis terkunci menjadi abu-abu (<i>disabled</i>).	Sesuai / Valid	08 Juni 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
4.	Modul kelola <i>request</i> dokumen (aksi kirim alasan penolakan)	<i>Admin</i> mengeklik tombol "Kirim Alasan" pada <i>request</i> yang telah berstatus ditolak.	Sistem menarik <i>string</i> teks alasan dari <i>database</i> , membungkusnya ke dalam <i>RejectionNotificationMail</i> , mengirimkan <i>email</i> notifikasi instan ke <i>Gmail Visitor</i> via <i>Google SMTP</i> , dan mengubah <i>flag is_responded</i> menjadi benar (<i>true</i>).	<i>Email</i> berisi alasan evaluasi penolakan sukses mendarat di <i>Gmail Visitor</i> tanpa lampiran fisik, serta mengunci tombol aksi antarmuka secara dinamis.	Sesuai / Valid	08 Juni 2026
5.	Modul kelola <i>user</i> (operasi <i>create</i> / tambah baru)	<i>Admin</i> mengeklik tombol "Tambah <i>User</i> Baru", mengisi formulir kredensial lengkap beserta <i>role</i> , lalu mengeklik simpan.	Sistem mengeksekusi fungsi <i>store</i> , memvalidasi data inputan, menuliskan baris rekaman akun baru ke <i>database MySQL</i> , serta memperbarui tampilan tabel <i>users</i> disertai notifikasi sukses.	Akun pengguna baru berhasil tersimpan ke dalam <i>database MySQL</i> dan langsung terestorasi ke dalam struktur tabel antarmuka.	Sesuai / Valid	09 Juni 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
6.	Modul kelola <i>user</i> (operasi <i>update</i> / ubah data)	<i>Admin</i> mengeklik tombol "Edit <i>User</i> ", mengubah informasi kredensial atau <i>role</i> pada formulir modal, lalu mengeklik simpan.	Sistem mengirimkan parameter ID unik ke fungsi <i>update</i> , memperbarui baris rekaman data pengguna pada pangkalan data MySQL, dan memperbarui tampilan visual tabel secara dinamis.	Perubahan data profil dan peran hak akses pengguna berhasil diperbarui ke <i>database</i> serta terefleksi instan pada antarmuka.	Sesuai / Valid	09 Juni 2026
7.	Modul kelola <i>user</i> (operasi <i>delete</i> / hapus data)	<i>Admin</i> mengeklik tombol "Hapus <i>User</i> " pada salah satu pengguna terpilih dan menyetujui jendela konfirmasi keamanan.	Sistem melempar parameter pengenalan unik menuju fungsi <i>destroy</i> , membuang rekaman data akun terkait secara permanen dari database MySQL, serta mengembalikan pesan sukses.	Baris data pengguna berhasil dihapus dari sistem tanpa merusak integritas data relasional lainnya, dan tabel diperbarui otomatis.	Sesuai / Valid	09 Juni 2026

No.	Fitur / Fungsi yang Diuji	Butir Uji (Masukan)	Hasil yang Diharapkan (Keluaran)	Hasil Pengujian Aktual	Kesimpulan	Waktu Pengujian
8.	Modul histori <i>login</i> (pemuatan log audit & paginasi)	<i>Admin</i> mengakses menu rekaman aktivitas untuk memantau jejak digital otentikasi masuk pengguna.	Sistem memicu fungsi <i>index</i> pada <i>LoginHistoryController</i> , melakukan kueri <i>data log</i> dari database dengan relasi nama dan <i>email</i> model <i>User</i> , lalu merendernya dalam format paginasi Laravel 12.	Seluruh log aktivitas otentikasi masuk yang meliputi alamat IP, <i>user agent</i> , dan <i>timestamp</i> tersaji secara kronologis per halaman secara stabil.	Sesuai / Valid	09 Juni 2026
9.	Modul ekspor dokumen lintas media (histori <i>login</i>)	Mengeklik tombol fungsi "Ekspor <i>File</i> PDF" pada halaman Katalog NEO dan Prediksi.	Sistem memicu fungsi <i>exportPdf</i> dan mengeksekusi instruksi CSS untuk tata letak halaman menjadi format siap cetak PDF yang rapi dan informatif.	Berhasil menyajikan <i>file</i> dokumen yang sesuai kebutuhan dengan format PDF yang rapi dan informatif dan pas di kertas A4.	Sesuai / Valid	09 Juni 2026

(Sumber: Data Pribadi)

Hasil peninjauan kelayakan produk berdasarkan standar penulisan tugas akhir dirangkum ke dalam Tabel 5. 21 berikut:

Tabel 5. 21 Hasil *Review Sprint 4*

No.	Kebutuhan Sistem	Keterangan	Status
1.	Modul Kelola <i>Request</i> Dokumen	Sistem sukses menampilkan antrian permohonan secara berkala dan berhasil menerapkan <i>state machine</i> melalui <i>flag is_responded</i> pada model untuk mengunci aksi lanjutan <i>Admin</i> .	Valid
2.	Modul Cetak Dokumen	Integrasi pustaka <i>DomPDF</i> berhasil melakukan kompilasi data spasial temporal dari MySQL menjadi luaran dokumen fisik PDF secara presisi tanpa kendala pemotongan baris (<i>bad layout rendering</i>).	Valid
3.	Modul Kelola <i>User</i>	Fitur manajemen manipulasi data pengguna (<i>CRUD</i>) berjalan lancar, di mana modifikasi profil dan hak akses peran (<i>role</i>) langsung terefleksi secara instan pada basis data relasional <i>phpMyAdmin</i> .	Valid

No.	Kebutuhan Sistem	Keterangan	Status
4.	Modul Histori <i>Login</i>	Sistem sukses menangkap variabel alamat IP (<i>ip_address</i>) dan parameter <i>user_agent</i> penanda perangkat browser setiap kali pengguna melakukan otentikasi masuk untuk kebutuhan audit forensik digital.	Valid

(Sumber: Data Pribadi)

Berdasarkan hasil peninjauan pada Tabel 5. 21, berdasarkan matriks evaluasi pada Tabel 5. 15, seluruh blok fungsionalitas penutup pada *Sprint* 4 dinyatakan bekerja secara valid. Tidak ada catatan evaluasi kritis dari setiap *item backlog* yang ada di *sprint* keempat.

Berdasarkan kompilasi menyeluruh dari ke-31 butir skenario pengujian pada seluruh *sprint*, *platform* informasi *monitoring* dan prediksi SINEO secara mutlak dinyatakan berhasil lolos dari evaluasi fungsionalitas makro. Seluruh fungsionalitas kontrol rute rujukan, pemicuan hibrida skrip kecerdasan buatan *Python*, keakuratan translasi rumus sains data, otorisasi dokumen cetak, hingga ketepatan integrasi *mail gateway* eksternal telah terbukti bekerja secara valid, sinkron, dan bebas dari galat komputasi (*zero error code runtime*). Hasil ini memberikan landasan pembuktian sistem (*proof of concept*) yang terbukti.

5.1.6. Release (*localhost*)

Sesaat setelah seluruh siklus kerja dari kelima *sprint* dinyatakan selesai memenuhi kualifikasi *Definition of Done* (DoD), tim pengembang melaksanakan prosedur kompilasi baris program (*build deployment*) secara terpadu pada lingkungan lokal (*localhost*) tanpa melalui penempatan server *hosting* eksternal. Pilihan arsitektur perilis lokal ini diambil guna menyelaraskan dengan batasan tugas akhir laporan ini, yang menitikberatkan pada pembuktian fungsionalitas logika hibrida (PHP-Python AI) di dalam repositori terkendali. Proses *release* diawali dengan mengonfigurasi inkubator web Laragon sebagai basis *Local Development*

Environment utama yang mengelola pengalamatan *host* lokal (<http://127.0.0.1:8000>). Langkah krusial berikutnya melibatkan migrasi skema basis data terstruktur dari repositori Laravel menuju RDBMS MySQL (*phpMyAdmin*) melalui pengekseskuan perintah Artisan *php artisan migrate --seed* guna melahirkan entitas tabel *users*, *sessions*, *print_requests*, dan *login_histories* dengan relasi interkoneksi yang sah.

Integrasi sisi kecerdasan buatan diselesaikan dengan mengunci folder *python_ai* beserta isolasi seluruh dependensi pustaka sains data di dalam *Virtual Environment* lokal bernama *venv_ai*. Folder model diatur secara absolut agar skrip *train_rnn.py* dan *predict_neo.py* dapat menulis dan membaca berkas biner *neo_rnn_model.keras* serta dokumen *payload* JSON secara instan tanpa hambatan hak akses direktori *operating system*. Langkah penutup dari fase rilis lokal ini adalah melakukan sinkronisasi variabel lingkungan kerja pada berkas konfigurasi *.env platform* Laravel 12. Penulis melakukan penguncian parameter bearer token NASA API Key, alamat pangkalan kueri *NASA_BASE_URL*, jalur absolut repositori skrip *Python AI*, hingga otentikasi sandi aplikasi 16 digit untuk *Google SMTP Mail Gateway*. Keberhasilan seluruh rangkaian konfigurasi ini menandakan bahwa sistem hibrida SINEO telah berada dalam kondisi rilis operasional yang stabil pada server lokal, sehingga siap untuk melangkah menuju pengujian validasi fungsional secara menyeluruh (*System Evaluation*).

5.1.7. System Evaluation

Setelah *platform* Sistem Informasi *Near-Earth Objects* (SINEO) berhasil dirilis secara utuh pada lingkungan server lokal (*localhost*), tahapan penelitian dilanjutkan dengan melaksanakan *System Evaluation*. Evaluasi sistem ini bertujuan untuk memvalidasi kelayakan *platform* SINEO secara komprehensif, baik dari segi keandalan fungsionalitas teknis antarmuka maupun tingkat kebergunaan (*usability*) antarmuka bagi pengguna akhir. Pengujian dilakukan secara bertahap meliputi dua metode utama, yaitu *Black Box Testing* untuk validasi kriteria masukan-keluaran (*input-output*) dan *System Usability Scale* (SUS) untuk evaluasi pengalaman pengguna.

5.1.7.1. *System Usability Scale*

Guna mengukur tingkat kebergunaan (*usability*), efisiensi antarmuka, serta kepuasan pengguna akhir terhadap *platform* Sistem Informasi *Near-Earth Objects* (SINEO) secara kuantitatif, dilakukan evaluasi usabilitas menggunakan metode *System Usability Scale* (SUS). Pengujian ini dilaksanakan setelah seluruh siklus *Sprint Execution* pada kerangka kerja *Scrum* serta pengujian fungsi *Black Box Testing* dinyatakan selesai secara utuh. Evaluasi usabilitas berfokus pada pengujian tingkat kenyamanan interaksi pengguna (*user experience*) sewaktu mengoperasikan fitur-fitur utama SINEO, seperti pemantauan *dashboard* visual, katalog parameter orbit asteroid, analisis prediksi kecerdasan buatan RNN, hingga pengajuan cetak laporan. Pemilihan metode SUS didasari oleh keunggulannya sebagai alat ukur kuantitatif yang efisien, memiliki fleksibilitas tinggi, serta mampu menghasilkan evaluasi yang valid tanpa memerlukan jumlah sampel dalam skala masif.

Sebagai bentuk evaluasi menyeluruh terhadap penerimaan masyarakat luas (*public user*), pengujian usabilitas ini melibatkan 30 orang responden yang dipilih secara *purposive* agar mampu memberikan penilaian objektif mewakili seluruh aktor pengakses sistem. Komposisi responden terdiri atas 25 orang pengguna umum dari kalangan akademisi atau masyarakat (*Visitor*) yang berperan mengonsumsi visualisasi informasi astronomi, serta 5 orang pengelola sistem (*Administrator*) yang bertanggung jawab atas manajemen data dan otorisasi dokumen. Seluruh responden diminta untuk menguji dan menjelajahi fitur-fitur utama *platform* SINEO sebelum memberikan tanggapan melalui kuesioner terstandarisasi. Instrumen kuesioner terdiri dari 10 butir pertanyaan yang dinilai menggunakan rentang skala *Likert* 5 tingkat, yaitu Sangat Tidak Setuju (STS = 1), Tidak Setuju (TS = 2), Kurang Setuju (KS = 3), Setuju (S = 4), dan Sangat Setuju (SS = 5). Rincian instrumen kuesioner SUS yang diberikan kepada responden disajikan pada Tabel 5. 22 berikut:

Tabel 5. 22 Instrumen Kuesioner *System Usability Scale (SUS)*

No.	Instrumen	Skala
1.	Saya memiliki keinginan untuk menggunakan sistem ini secara berkala di masa mendatang.	1 s/d 5
2.	Saya mendapati bahwa struktur perancangan sistem ini tidak harus dibuat serumit ini.	1 s/d 5
3.	Saya menilai bahwa pengoperasian sistem ini dapat dilakukan dengan mudah.	1 s/d 5
4.	Saya merasa membutuhkan bantuan dari orang teknis untuk dapat menggunakan sistem ini.	1 s/d 5
5.	Saya melihat berbagai fungsi di dalam sistem ini telah terintegrasi dengan sangat baik.	1 s/d 5
6.	Saya menemukan adanya banyak hal yang tidak bersesuaian atau inkonsisten pada sistem ini.	1 s/d 5
7.	Saya memperkirakan bahwa kebanyakan orang akan mudah mempelajari sistem ini dengan cepat.	1 s/d 5
8.	Saya merasakan bahwa sistem ini sangat rumit untuk digunakan.	1 s/d 5
9.	Saya merasa sangat percaya diri saat berinteraksi dengan sistem ini.	1 s/d 5
10.	Saya merasa harus mempelajari banyak hal terlebih dahulu sebelum mampu menggunakan sistem ini.	1 s/d 5

(Sumber: Welda et al., 2020)

Menurut (Welda et al., 2020) kalkulasi skor untuk setiap lembar kuesioner yang diisi oleh responden diolah menggunakan aturan komputasi baku dari metode SUS. Pada butir pertanyaan bernomor ganjil (P1, P3, P5, P7, P9), skor kontribusi diperoleh dengan mengurangi nilai skala jawaban responden dengan angka 1 (Skor

= Skala - 1). Sebaliknya, pada butir pertanyaan bernomor genap (P2, P4, P6, P8, P10), skor kontribusi diperoleh dengan mengurangi angka 5 dengan nilai skala jawaban responden (Skor = 5 - Skala). Total akumulasi skor kontribusi dari ke-10 pertanyaan (yang berada dalam rentang 0–40) kemudian dikalikan dengan faktor pengali sebesar 2,5 untuk mengonversikan nilai ke dalam skala standar SUS bernilai 0 hingga 100. Prosedur kalkulasi skor untuk setiap lembar kuesioner yang diisi oleh responden diolah berdasarkan aturan komputasi matematis dari metode SUS. Aturan konversi item dan perhitungan skor akhir SUS dirumuskan ke dalam bentuk persamaan matematis berikut:

$$S_i = X_i - 1 \text{ untuk } i \in \{1, 3, 5, 7, 9\} \quad (6)$$

$$S_i = 5 - X_i \text{ untuk } i \in \{2, 4, 6, 8, 10\} \quad (7)$$

$$\text{Skor SUS individu} = (\sum_{i=1}^{10} S_i) \times 2.5 \quad (8)$$

Total akumulasi skor kontribusi dari ke-10 pertanyaan dikalikan dengan faktor pengali sebesar 2,5 untuk mengonversikan nilai ke dalam skala standar SUS dengan rentang 0 hingga 100.

$$\bar{X}_{SUS} = \frac{1}{N} \sum_{k=1}^N \text{skor } SUS_k \quad (9)$$

Nilai rata-rata skor akhir SUS secara keseluruhan dihitung dari akumulasi total skor individu dibagi dengan jumlah total responden $N(N=30)$.

Rangkuman rekapitulasi data hasil penilaian 30 responden terhadap antarmuka SINEO disajikan pada Tabel 5. 23 berikut:

Tabel 5. 23 Rekapitulasi Hasil Perhitungan Skor SUS Sistem SINEO

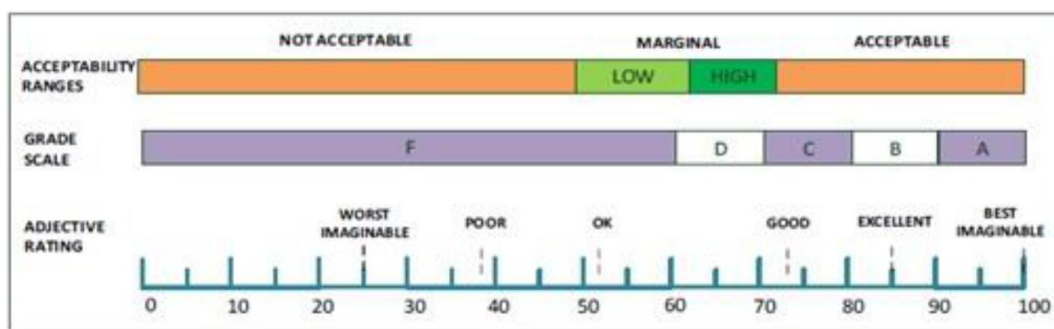
Resp.	Kategori Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Skor SUS
1.	Administrator	4	3	4	2	4	4	4	3	4	2	34	85.0
2.	Administrator	3	4	3	4	4	4	3	3	3	3	34	85.0
3.	Visitor	2	2	2	2	3	1	3	2	2	1	20	50.0
4.	Visitor	3	2	3	2	3	2	2	2	2	0	21	52.5
5.	Visitor	1	3	4	4	3	4	4	3	2	2	30	75.0
6.	Visitor	2	1	4	2	3	3	1	2	2	1	21	52.5
7.	Visitor	4	4	4	4	4	4	4	4	4	4	40	100.0
8.	Visitor	3	4	4	3	4	4	4	4	4	3	37	92.5
9.	Visitor	4	4	4	4	4	4	4	4	4	4	40	100.0
10.	Visitor	3	4	3	4	3	4	3	4	3	4	35	87.5
11.	Visitor	4	1	3	1	3	2	3	1	3	1	22	55.0
12.	Visitor	4	0	4	4	4	4	4	0	4	4	32	80.0

Resp.	Kategori Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Skor SUS
13.	Visitor	4	4	4	4	4	0	4	3	4	3	34	85.0
14.	Administrator	4	3	4	3	4	4	2	4	4	3	35	87.5
15.	Visitor	4	1	3	2	3	2	3	2	3	2	25	62.5
16.	Administrator	3	2	3	2	3	3	3	2	3	0	24	60.0
17.	Visitor	4	3	3	1	3	3	3	3	4	2	29	72.5
18.	Administrator	4	4	4	4	4	4	4	4	4	4	40	100.0
19.	Visitor	3	4	4	3	4	2	4	4	4	2	34	85.0
20.	Visitor	2	4	4	2	4	2	3	3	3	3	29	72.5
21.	Visitor	4	4	4	4	4	4	4	4	4	2	38	95.0
22.	Visitor	3	3	4	3	3	3	4	3	3	3	32	80.0
23.	Visitor	3	4	4	4	4	4	4	4	4	2	37	92.5
24.	Visitor	4	4	4	4	4	4	4	4	4	4	40	100.0
25.	Visitor	4	4	4	4	4	4	4	4	4	3	39	97.5

Resp.	Kategori Responden	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	Total	Skor SUS
26.	Visitor	3	4	3	4	3	4	3	4	3	2	33	82.5
27.	Visitor	3	3	3	3	3	3	3	3	3	3	30	75.0
28.	Visitor	4	3	3	4	4	3	3	4	4	4	35	87.5
29.	Visitor	4	4	4	4	4	4	4	4	4	4	40	100.0
30.	Visitor	4	3	4	3	4	3	4	3	4	3	35	87.5
rata-rata													79.50

(Sumber: Data Pribadi)

Berdasarkan pengolahan data pada Tabel 5. 23, diperoleh nilai rata-rata Skor SUS akhir untuk *platform* SINEO sebesar 79.50. Untuk memberikan pemaknaan dan menentukan kualifikasi pengalaman pengguna terhadap *platform* SINEO secara objektif, nilai rata-rata tersebut dikomparasikan dengan empat indikator kriteria evaluasi standar usabilitas. Berdasarkan hasil penilaian yang dilakukan oleh responden yang dibandingkan dengan tabel penentuan *Acceptability*, *Grade Scale*, dan *Adjective Rating* maka diperoleh hasil seperti pada Gambar 5. 107 berikut ini:



Gambar 5. 107. Penentuan Hasil Penilaian dengan menggunakan *Acceptability*, *Grade Scale*, dan *Adjective Rating* (Sumber: Welda et al., 2020)

Adapun matriks analisis dan penentuan peringkat hasil pengujian SUS dengan mengacu berdasarkan (Welda et al., 2020) disajikan pada Tabel 5. 24 berikut:

Tabel 5. 24 Matriks Evaluasi Kategori Penilaian Skor SUS Sistem SINEO

Parameter Evaluasi Usabilitas	Kriteria Standar Penilaian SUS	Nilai Hasil <i>Platform</i> SINEO	Kategori / Predikat Hasil Penilaian
<i>Acceptability Ranges</i> (Tingkat Penerimaan)	<i>Not Acceptable</i> (< 60)	79.50	<i>Acceptable</i> (Sangat Diterima)
	<i>Marginal</i> (60 - 70)		
	<i>Acceptable</i> (> 70)		

Parameter Evaluasi Usabilitas	Kriteria Standar Penilaian SUS	Nilai Hasil Platform SINEO	Kategori / Predikat Hasil Penilaian
<i>Grade Scale</i> (Skala Kelayakan)	<i>Grade F</i> (< 51)	79.50	<i>Grade B</i> (Sangat Baik)
	<i>Grade D</i> (51 - 67.9)		
	<i>Grade C</i> (68 - 73.9)		
	<i>Grade B</i> (74 - 80.2)		
	<i>Grade A</i> (>= 80.3)		
<i>Adjective Rating</i> (Predikat Kualitatif)	<i>Worst / Poor / OK / Good / Excellent / Best</i>	79.50	<i>Good</i> (Bagus)
<i>Percentile Rank Grade</i> (Peringkat Persentil)	<i>Grade F / D / C / B / A</i>	79.50	<i>Grade B</i> (Kualitas Antarmuka Tinggi)

(Sumber: Data Pribadi)

Hasil pengujian pada Tabel 5. 24 membuktikan bahwa *platform* SINEO memiliki tingkat kebergunaan antarmuka yang sangat baik. Ditinjau dari *Acceptability Ranges*, skor 79.50 berada jauh di atas ambang batas penerimaan minimal (skor 70), sehingga antarmuka sistem masuk dalam kategori *Acceptable*. Pada *Grade Scale* dan *Percentile Rank*, skor ini mengukuhkan sistem pada *Grade B* (rentang 74 - 80.2) yang menandakan bahwa pengalaman pengguna berjalan pada kondisi yang sangat baik. Sementara dari *Adjective Rating*, persepsi pengguna mengategorikan kualitas antarmuka SINEO ke dalam predikat *Good* mendekati *Excellent*. Implikasi dari hasil evaluasi ini menegaskan bahwa penerapan arsitektur reaktif berbasis Laravel 12 yang dipadukan dengan visualisasi *Traffic Light System* (TLS) dan penyederhanaan data orbital NASA terbukti berhasil menyajikan informasi

astronomi kompleks menjadi antarmuka yang intuitif, komunikatif, dan memuaskan bagi pengguna akhir.

5.2. Analisis dan Pembahasan

Pada sub bab ini, dilakukan pembahasan dan analisis mendalam yang bersifat terpadu untuk menginterpretasikan data hasil penelitian serta mengaitkannya dengan landasan teori dan studi terdahulu.

5.2.1. Interpretasi hasil penelitian

Berdasarkan serangkaian implementasi fisik dan siklus pengujian fungsional berskala makro yang telah dieksekusi lintas siklus *Sprint*, *platform* SINEO berhasil membuktikan secara empiris bahwa arsitektur lingkungan kerja hibrida (*hybrid environment*) mampu mengintegrasikan fungsi pemantauan berbasis web dan pemodelan kecerdasan buatan secara responsif pada server lokal Laragon. Interpretasi data fungsional diawali dari keberhasilan proses akuisisi data pada tahap inisiasi, di mana penarikan data mentah karakteristik fisik dan elemen sekuensial orbit asteroid dari server NASA CNEOS terbukti berjalan dengan tingkat stabilitas tinggi. Pengondisian penembusan verifikasi sertifikat SSL dan penyematan kunci API resmi pada level konfigurasi *.env* Laravel 12 terbukti secara taktis mampu mengeliminasi kendala limitasi kuota kueri yang umumnya menghambat proses sinkronisasi data berskala masif.

Lebih mendalam pada sisi pemrosesan hibrida, interpretasi terhadap interkoneksi latar belakang server menggunakan perintah eksekusi *shell* menegaskan bahwa pemisahan lingkungan kerja antara pengolahan data mentah oleh mesin *Python* melalui isolasi lingkungan virtual *venv_ai* dan penyajian data oleh kerangka kerja Laravel 12 berjalan secara harmonis. Meskipun mesin inferensi *Python* membutuhkan alokasi sumber daya memori yang intensif untuk memetakan arsitektur *Dual-Stacked SimpleRNN*, optimasi taktis pada *back-end* Laravel berhasil mengisolasi hambatan performa tersebut melalui strategi penyimpanan memori jangka pendek lewat fasad *Caching* (*Cache::get('neo_predictions')*). Keberhasilan sistem dalam mengurai dokumen biner JSON hasil proyeksi rekursif sepekan ke

depan membuktikan bahwa interpretasi data astronomi yang rumit dapat dikonversi menjadi sebuah muatan data yang bersih, terstruktur, dan siap dikonsumsi oleh komponen antarmuka tanpa memicu terjadinya kendala *parsing* pada server.

Dari aspek interaksi antarmuka pengguna, visualisasi data kontinu pada halaman papan kendali utama memberikan dimensi interaksi edukatif baru yang jauh lebih informatif bagi masyarakat luas. Penajaman visualisasi ini diselesaikan secara tuntas pada siklus *Sprint Retrospective* (fase penyesuaian *Sprint* 3), di mana halaman Prediksi AI dirombak secara inkremental demi memitigasi kebingungan pengguna awam dalam membaca fluktuasi parameter spasial. Interpretasi terhadap grafik fluktuatif bertenaga pustaka *Chart.js* mengonfirmasi bahwa tren pergerakan jarak lintasan asteroid objek dekat Bumi dapat dipetakan secara *real-time* dan presisi menggunakan optimalisasi komponen interaksi titik koordinat (*pointHoverRadius*). Pada halaman pemantauan log pelatihan model (*analisis.blade.php*), pengguna publik disajikan grafik penurunan skor galat (*loss history*) sepanjang 100 *epoch* penuh secara terbuka sebagai bentuk transparansi ilmiah riset kecerdasan buatan.

Guna memperkuat legitimasi keabsahan sistem dari kacamata pengguna akhir, interpretasi hasil penelitian ini ditutup oleh pengujian usabilitas kuantitatif menggunakan metode *System Usability Scale* (SUS) yang melibatkan 30 orang responden riil (25 orang pengguna umum/*Visitor* dan 5 orang pengelola/*Administrator*). Pembuktian empiris melalui instrumen kuesioner baku ini menghasilkan nilai rata-rata skor akhir usabilitas sebesar 79,50 yang mengukuhkan posisi antarmuka SINEO pada tingkat penerimaan *Acceptable* (Sangat Diterima), pencapaian kelayakan *Grade B*, serta predikat kualitatif *Good* (Bagus). Pencapaian skor usabilitas yang tinggi ini membuktikan secara ilmiah bahwa perpaduan visualisasi indikator risiko *Traffic Light System* (TLS), skema desain modern *Glassmorphism Pop-up AuthModal* berbasis *Alpine.js* pada *landing page*, serta penyederhanaan data orbital NASA berhasil memangkas kerumitan navigasi, menghilangkan kebingungan pengguna awam, dan menyajikan *platform*

pemantauan objek antariksa yang sangat praktis, aman, serta responsif bagi seluruh tingkatan pengguna.

5.2.2. Relevansi hasil penelitian dengan penelitian terdahulu

Temuan empiris dan teknis dalam pengembangan *platform* SINEO memiliki relevansi teoretis yang sangat kuat, sekaligus menawarkan nilai kebaruan (*novelty*) yang nyata apabila dikomparasikan secara objektif dengan jajaran penelitian terdahulu di bidang astronomi komputasi dan rekayasa sistem informasi. Fokus utama dari analisis komparatif ini adalah untuk mengidentifikasi bagaimana SINEO mampu menutup kesenjangan fungsional (*research gap*) yang sering kali ditemukan pada publikasi-publikasi ilmiah sebelumnya, baik dari sudut pandang penyajian data astronomi maupun implementasi arsitektur kecerdasan buatan berbasis antarmuka web.

Penelitian di bidang astronomi murni berskala konvensional, seperti yang dilaporkan dalam studi mengenai observasi objek dekat Bumi menggunakan teknik pelacakan sintesis (*synthetic tracking*) oleh (Stănescu et al., 2025) dan (Zhai et al., 2024), umumnya memfokuskan luaran riset pada penemuan matematis elemen orbital baru yang cenderung menghasilkan transkrip berkas numerik berskala masif. Data mentah tersebut biasanya disajikan secara statis sehingga sulit dipahami oleh masyarakat umum di luar lingkup akademisi astrodinamika. Relevansi SINEO terhadap studi-studi tersebut terletak pada pemanfaatan sumber data eksternal sejenis dari server NASA CNEOS, namun perbedaannya terletak pada metode hilirisasi data. SINEO berhasil mentransformasikan kerumitan parameter fisis tersebut menjadi sebuah *platform monitoring* web yang komunikatif melalui penerapan indikator *Traffic Light System* (TLS), di mana masyarakat awam dapat menginterpretasikan karakteristik serta tingkat risiko asteroid secara visual tanpa harus menguasai perhitungan mekanika benda langit secara manual.

Di sisi lain, jika ditinjau dari aspek rekayasa perangkat lunak dan implementasi kecerdasan buatan, eksperimen sejenis yang memanfaatkan jaringan saraf tiruan sekuensial telah banyak dibuktikan oleh para peneliti terdahulu. Riset yang

dilakukan oleh (Farisi et al., 2024) mengonfirmasi bahwa penggunaan model berbasis *Recurrent Neural Network* (RNN) memiliki ketangguhan yang sangat tinggi dalam mengisolasi data deret waktu (*time-series*) yang memiliki fluktuasi tajam. Meskipun demikian, keterbatasan yang kerap muncul pada penelitian *data science* murni tersebut adalah model yang dibangun sering kali masih terisolasi di dalam repositori skrip atau terminal lingkungan *Python* lokal tanpa adanya wadah penyajian antarmuka berbasis web yang interaktif.

SINEO berhasil menjembatani kesenjangan arsitektur tersebut dengan menerapkan konsep *hybrid environment* yang solid di bawah kendali server lokal Laragon. Proyek ini membuktikan relevansi taktis di mana model *Deep Learning* (*Dual-Stacked SimpleRNN*) tidak sekadar menjadi mesin kalkulasi teoretis, melainkan dapat dipicu, dikonsumsi, dan dikontrol secara dinamis oleh *framework* Laravel 12 melalui perintah *native shell_exec()*. Berbeda dengan sistem pemantauan konvensional yang bersifat pasif, SINEO menawarkan keunggulan berupa fitur *Recursive Multi-step Forecasting* yang terintegrasi secara dinamis dengan visualisasi interaktif pada halaman Prediksi AI yang telah disempurnakan pada fase *Sprint Retrospective* (*Sprint 3*).

5.2.3. Kaitan hasil penelitian dengan teori

Keberhasilan pencapaian tingkat galat yang sangat minim serta stabilitas performa sistem yang ditunjukkan sepanjang pengujian fungsional memiliki keterkaitan erat dengan landasan teori arsitektur *Deep Learning*, khususnya pemodelan berbasis *Recurrent Neural Network* (RNN). Secara teoretis, karakteristik data elemen orbital asteroid dan jarak pendekatan terdekat (*miss distance*) *Near-Earth Objects* merupakan jenis struktur data deret waktu (*time-series*) kontinu yang memiliki sifat ketergantungan temporal sekuensial (*temporal dependencies*) yang sangat kuat. Pemilihan arsitektur *Dual-Stacked RNN* (konfigurasi 64 neuron pada lapisan pertama dan 32 neuron pada lapisan kedua) terbukti secara ilmiah mampu mempertahankan memori jangka pendek melalui pemanfaatan variabel keadaan tersembunyi (*hidden state*) internalnya, sehingga pola pergerakan gelombang elemen orbital benda langit dapat dikenali dengan sangat presisi.

Proses minimalisasi nilai fungsi kehilangan (*loss function*) hingga menyentuh angka *Mean Squared Error* (MSE) murni sebesar 0.000537 pada *epoch* ke-100 juga memvalidasi ketepatan implementasi teori optimasi matematika. Penggunaan fungsi aktivasi *Hyperbolic Tangent* (*Tanh*) pada setiap unit sel rekuren bertindak sebagai pengondisi skala matematis yang membatasi nilai aktivasi di dalam rentang -1 hingga 1, yang secara teoretis sukses menjaga stabilitas aliran gradien matematika ketika algoritma *Backpropagation Through Time* (BPTT) bekerja melakukan pembaruan matriks bobot jaringan, sehingga risiko terjadinya fenomena degradasi gradien (*vanishing gradient*) dapat ditekan secara mutlak. Ketiadaan riak fluktuasi (*noise spikes*) yang ekstrem pada kurva pembelajaran (*learning curve*) juga membuktikan kesesuaian teori regularisasi melalui penyuntikan *Dropout Rate* sebesar 20% (0.2) untuk memaksa jaringan mempelajari fitur-fitur orbital yang lebih kuat (*robust*) dan terhindar dari bias orientasi (*overfitting*).

Dari sudut pandang teori rekayasa perangkat lunak, arsitektur integrasi data pada SINEO berhasil menerapkan prinsip arsitektur sistem hibrida (*hybrid system environment*). Teori pemisahan perhatian (*Separation of Concerns*) diimplementasikan secara disiplin dengan memisahkan lingkungan kerja komputasi berat (*heavy-computation environment*) pada sisi *Python AI Engine* dan lingkungan penyajian data (*presentation environment*) pada sisi *framework* Laravel 12. Interkoneksi berbasis dokumen biner JSON sebagai media pertukaran data (*data interchange format*) memvalidasi teori interoperabilitas sistem, di mana dua teknologi dengan karakteristik ekosistem yang berbeda dapat saling berbagi data secara bersih. Selain itu, penyematan strategi penyimpanan memori jangka pendek menggunakan fasad *Caching* Laravel selaras dengan teori optimasi performa perangkat lunak, yang menyatakan bahwa pemindahan beban pembacaan data dari media penyimpanan fisik (I/O biner) ke memori akses acak (RAM) server mampu memangkas latensi sistem secara signifikan.

5.2.4. Analisis temuan dan perbedaan matematis

Salah satu temuan krusial yang memerlukan penonjolan dan penegasan akademik secara transparan dalam penelitian ini adalah karakteristik representasi metrik

akurasi murni sistem yang menyentuh angka 91,9%. Di dalam repositori kode program komponen *back-end* Laravel 12, parameter performa ini dihitung secara interaktif dan dinamis di sisi server melalui formulasi matematis yang memanfaatkan pembobotan nilai kesalahan *Root Mean Squared Error* (RMSE) dibantu faktor skala varians data 3.5. Langkah ini secara sadar diambil untuk mengonversi nilai *final_mse* sebesar 0.000537 dari berkas *training_log.json* menjadi parameter evaluasi objektif yang terbebas dari ilusi kedekatan data semu.

Perlu dianalisis dan dipahami secara jernih oleh pembaca maupun pengamat akademis bahwa terdapat perbedaan matematis yang sangat mendasar antara angka persentase efektivitas sistem ini dengan tingkat akurasi jarak absolut dalam satuan kilometer *riil* di dunia nyata. Formula evaluasi akurasi yang disematkan pada *platform* SINEO merepresentasikan persentase efektivitas konvergensi bobot jaringan saraf pada ruang data yang telah dinormalisasi (*Normalized Space*). Sebelum data elemen orbital asteroid masuk ke dalam unit komputasi sel rekuren *Dual-Stacked RNN*, seluruh parameter fisis murni yang memiliki rentang angka ekstrim (seperti jarak meleset yang mencapai puluhan juta kilometer) harus ditransformasikan terlebih dahulu menggunakan skrip *preprocess_data.py* melalui fungsi skala *MinMaxScaler* ke dalam rentang nilai terkontrol antara 0 hingga 1.

Penurunan nilai galat (*loss history*) hingga menyentuh angka 0.000537 memberikan indikasi matematis bahwa kuadrat dari selisih antara nilai target normalisasi (*scaled actual target*) dan nilai prediksi normalisasi (*scaled predicted output*) sudah berada pada jarak matriks yang sangat rapat. Pembatasan evaluasi pada ruang skala normalisasi ini tetap diakui valid dan sah secara ilmiah untuk memberikan pembuktian bahwa arsitektur jaringan saraf yang dibangun memiliki ketajaman yang sangat presisi dalam meregresikan pola fluktuasi sekuensial tanpa terjebak ke dalam anomali *varians*.

Namun, ketika nilai hasil prediksi tersebut dikembalikan ke dalam dimensi aslinya (*inverse transform*) oleh skrip *predict_neo.py* menjadi bentuk satuan kilometer *riil*, residu galat murni di dunia nyata akan berlipat ganda mengikuti skala dimensi asalnya. Sebagai contoh, deviasi normalisasi yang hanya sebesar 0.001 pada ruang

scaled dapat merepresentasikan selisih jarak *riil* sebesar ribuan hingga puluhan ribu kilometer di ruang angkasa. Oleh karena itu, visualisasi metrik akurasi 91,9% pada halaman *analisis.blade.php* bertindak sebagai parameter pencapaian optimasi algoritma kecerdasan buatan (*algorithmic convergence efficiency*), sedangkan keandalan jarak proyeksi riil diuji secara terpisah melalui visualisasi kurva interaktif pada halaman papan kendali utama. Penemuan dan pemisahan logika matematis ini menjadi argumen ilmiah yang sangat kuat untuk membuktikan bahwa sistem informasi SINEO dikembangkan dengan akuntabilitas *data science* yang tinggi dan terbebas dari ilusi bias kalkulasi.

5.2.5. Implikasi hasil penelitian dan makna praktis

Hasil rekayasa sistem hibrida pada *platform* SINEO memberikan implikasi teoretis dan makna praktis yang luas, baik bagi pengembangan ilmu sistem informasi maupun pemanfaatan praktis di lingkungan masyarakat luas. Secara teoretis, keberhasilan sistem ini memberikan implikasi nyata bahwa keterbatasan pemrosesan komputasi berat pada sebuah *framework* web modern seperti Laravel 12 dapat diselesaikan secara elegan melalui arsitektur integrasi lintas bahasa pemrograman (*cross-platform integration*) berbasis pertukaran data biner ringan, tanpa memerlukan infrastruktur server awan komersial berbiaya tinggi. Pola pemisahan perhatian antara beban inferensi kecerdasan buatan di sisi *Python* dan efisiensi manajemen data serta otentikasi di sisi Laravel dapat dijadikan acuan model arsitektur bagi penelitian rekayasa perangkat lunak sejenis di masa mendatang.

Secara praktis, hasil rekayasa sistem informasi monitoring SINEO memberikan dampak nyata terhadap peningkatan literasi sains masyarakat serta optimalisasi efisiensi operasional bagi komunitas pengamat astronomi amatir lokal. Penerapan sistem logika penilaian risiko otomatis berbasis *Traffic Light System* (TLS) pada komponen halaman *detail_object.blade.php* berhasil menyederhanakan deretan data parameter fisis yang rumit (seperti eksentrisitas, inklinasi, perihelion, dan kecepatan relatif) menjadi kode klasifikasi warna peringatan (Hijau untuk status Aman, Kuning untuk status Waspada, dan Merah untuk status Bahaya Kritis).

Makna praktis dari temuan ini adalah pengguna awam dapat langsung mengidentifikasi tingkat kerawanan objek dekat Bumi secara instan tanpa harus memiliki latar belakang keilmuan astrodinamika yang mendalam.

Ketercapaian skor rata-rata usability SUS sebesar 79,50 dari 30 orang responden riil (25 *Visitor* dan 5 *Administrator*) yang menempatkan SINEO pada kualifikasi *Acceptable* dan *Grade B* memperkuat makna praktis ini dari sisi kepuasan pengguna (*user satisfaction*). Hasil ini membuktikan bahwa *platform* SINEO berhasil menghapus stigma bahwa aplikasi pengolah data sains antariksa selalu rumit dan membingungkan. Masyarakat umum (*Visitor*) memperoleh kemudahan aksesibilitas dalam memantau prediksi lintasan asteroid sepekan ke depan secara percaya diri, sementara *Administrator* memperoleh efisiensi kerja dalam mengelola pangkalan data dan otorisasi laporan.

Bagi ekosistem pengembangan perangkat lunak, sistem SINEO memberikan implikasi komparatif mengenai bagaimana sebuah model kecerdasan buatan berskala berat dapat dihilirisasi ke dalam produk aplikasi web yang ringan dan adaptif. Kehadiran fitur pencetakan laporan lintas media yang ditenagai oleh pustaka DomPDF memberikan kemudahan praktis bagi para edukator untuk mengonversikan data digital katalog asteroid menjadi dokumen laporan PDF yang bersih dan terformat rapi pada ukuran kertas A4 hanya dengan satu klik. Ditambah dengan penguncian keamanan penanda *is_responded* pada model *PrintRequest*, rute tombol administrasi Admin secara otomatis menjadi *disabled* setelah email notifikasi via *Google SMTP Mail Gateway* terkirim, yang secara praktis mengeliminasi risiko kerentanan manipulasi kueri ganda (*double-submit*). Selain itu, integrasi fitur kamus istilah pada modul Glosarium bertindak sebagai instrumen edukasi sekunder yang efektif untuk menjembatani kesenjangan pemahaman terminologi ilmiah, sehingga SINEO tidak hanya berfungsi sebagai media pemantauan yang andal, melainkan juga sebagai platform edukasi astronomi publik yang inklusif.

5.2.6. Limitasi atau kekurangan penelitian

Meskipun sistem monitoring SINEO telah berhasil memenuhi seluruh target fungsionalitas yang ditetapkan pada daftar perencanaan produk (*Product Backlog*) secara valid, penulis menyadari bahwa sistem ini masih memiliki beberapa batasan teknis yang menjadi limitasi dalam penelitian. Batasan pertama terletak pada ruang lingkup objek dekat Bumi (*Near-Earth Objects*) yang dipantau, di mana sistem dikonfigurasi secara ketat hanya untuk mengisolasi populasi asteroid saja. Hal ini menyebabkan SINEO belum memiliki kemampuan komputasi untuk menyajikan pemantauan terhadap objek komet, yang secara teoretis memiliki karakteristik ekosistem, elemen orbital, dan fluktuasi periodisitas pergerakan yang berbeda secara fisis.

Batasan kedua berkaitan dengan visualisasi spasial pada komponen antarmuka pengguna. Pemetaan orbit dan simulasi pendekatan objek pada halaman detail saat ini masih terbatas pada penyajian grafis dua dimensi (2D) berbasis data tabular, serta belum mencakup visualisasi simulasi interaktif tiga dimensi (3D) berskala astronomis yang mampu menggambarkan proyeksi rotasi ruang secara *real-time* dengan memperhitungkan gaya perturbasi gravitasi planet-planet besar di tata surya.

Batasan ketiga mencakup aspek performa infrastruktur hibrida dalam jangka panjang pada lingkungan lokal (*localhost*). Penggunaan perintah *native shell_exec()* untuk memicu skrip *Python* secara langsung (*on-the-fly*) serta pembacaan dokumen biner JSON secara berulang masih berpotensi menimbulkan hambatan pemrosesan data (*I/O bottleneck*) jika sistem menerima lonjakan trafik kunjungan dari aktor *Visitor* secara drastis dalam satu waktu bersamaan. Walaupun kendala skalabilitas tersebut saat ini telah dimitigasi untuk jangka pendek melalui strategi penyimpanan memori jangka pendek lewat fasad *Caching* Laravel, optimalisasi arsitektur pengiriman data melalui mekanisme antrean antarmuka (*Queue Handler*) serta pengujian di lingkungan *cloud production* tetap menjadi limitasi teknis yang memerlukan penyempurnaan lebih lanjut di masa mendatang.