

BAB III

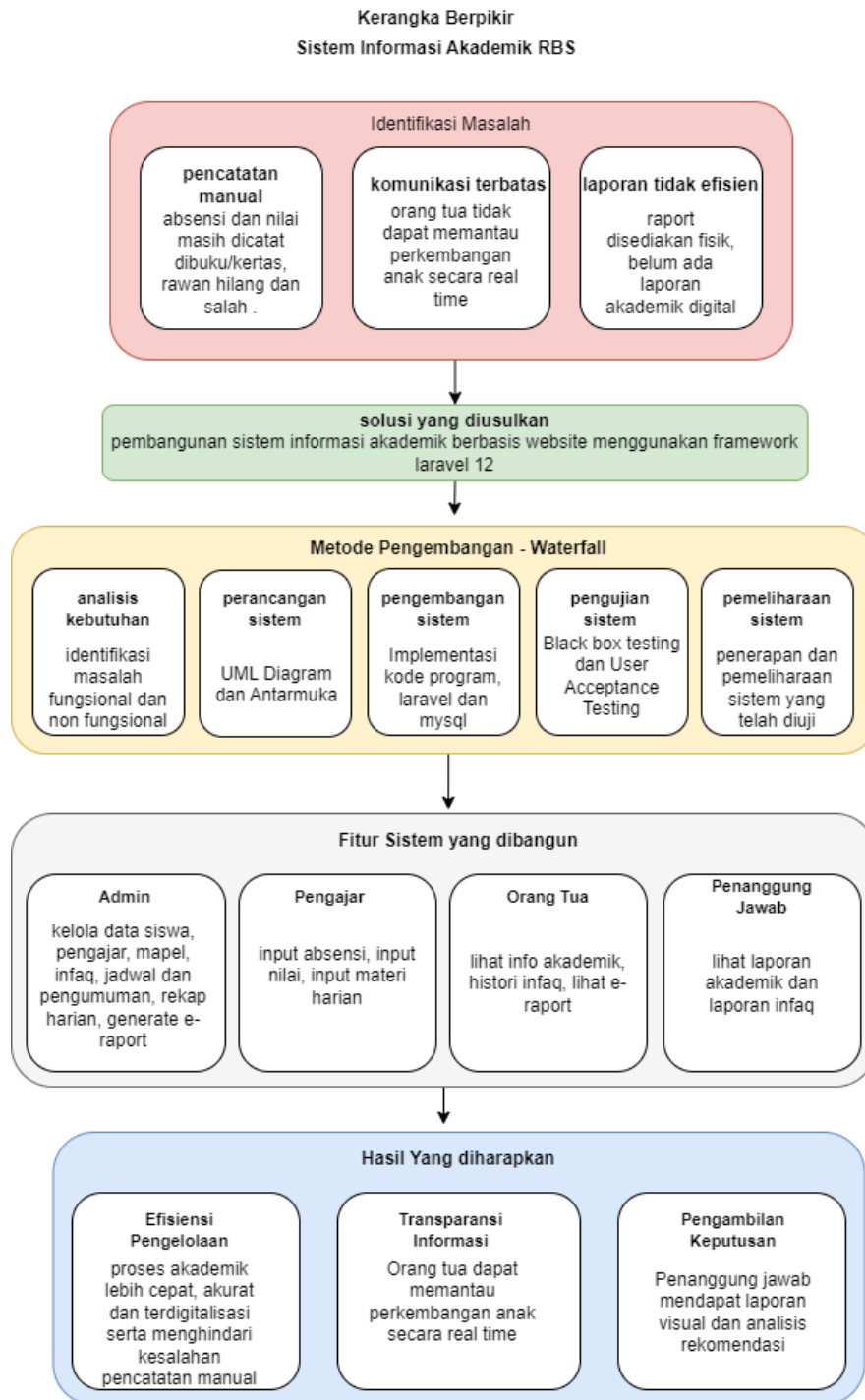
ANALISIS DAN PERANCANGAN

3.4.1 Kerangka Berpikir

Kerangka berpikir merupakan gambaran alur pemikiran peneliti dalam membangun sistem informasi akademik berbasis *website* di Ruang Belajar Sa'idah. Kerangka berpikir ini disusun berdasarkan permasalahan yang ditemukan di lapangan hingga menghasilkan solusi berupa sistem yang mampu mendukung pengelolaan data akademik secara terintegrasi.

Berdasarkan hasil observasi dan wawancara yang dilakukan di Ruang Belajar Sa'idah, diketahui bahwa pengelolaan data akademik masih dilakukan secara manual menggunakan buku besar, lembaran kertas, dan aplikasi perkantoran sederhana. Pengelolaan data tersebut meliputi data siswa, data pengajar, absensi, jadwal dan pengumuman, mata pelajaran, materi harian, nilai, Administrasi infaq, serta penyusunan *E-raport*. Kondisi ini menyebabkan proses pengelolaan data menjadi kurang efektif dan efisien, serta berpotensi menimbulkan kesalahan pencatatan, keterlambatan penyajian informasi, dan risiko kehilangan data.

Sebagai solusi terhadap permasalahan yang ditemukan, penelitian ini melakukan pengembangan sistem informasi akademik berbasis *website* dengan menerapkan metode *Waterfall* yang terdiri dari beberapa tahapan, yaitu analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan sistem. Sistem yang dikembangkan memanfaatkan *framework Laravel* sebagai teknologi pengembangan serta *MySQL* sebagai basis data penyimpanan informasi, dengan melibatkan empat jenis pengguna atau aktor, yaitu Admin, Pengajar, Orang Tua, dan Penanggung Jawab. Sistem ini diharapkan mampu mendukung proses pengelolaan data akademik agar menjadi lebih efektif, efisien, akurat, dan terintegrasi, sehingga dapat meningkatkan kualitas pelayanan akademik di Ruang Belajar Sa'idah.



Gambar 3. 1 Kerangka Berpikir

Sumber : Hasil Perancangan Penulis (2026)

3.4.2 Prosedur Penelitian

Prosedur penelitian dalam pengembangan sistem informasi akademik berbasis *website* pada Ruang Belajar Sa'idah dilakukan melalui beberapa tahapan yang terstruktur. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode *Waterfall*. Pemilihan metode *Waterfall*

didasarkan pada tahapn pengembangan yang sistematis dan terstruktur sehingga memudahkan proses pengembangan sistem.

Adapun tahapan yang dilakukan dalam penelitian ini meliputi analisis kebutuhan, perancangan sistem, implementasi, pengujian sistem dan pemeliharaan sistem.

1. Analisis Kebutuhan (*Requirement Analysis*)

Pada tahap ini dilakukan identifikasi kebutuhan sistem berdasarkan hasil observasi dan wawancara di Ruang Belajar Sa'idah. Analisis ini bertujuan untuk mengetahui permasalahan yang terjadi serta menemukan kebutuhan sistem yang akan dikembangkan, baik kebutuhan fungsional maupun non-fungsional.

2. Perancangan Sistem (*Design*)

Tahap perancangan sistem dilakukan dengan membuat desain yang meliputi perancangan alur sistem, struktur *Database*, serta rancangan antarmuka (*interface*). Perancangan ini bertujuan untuk memberikan gambaran awal mengenai sistem yang akan dibangun.

3. Implementasi (*Coding*)

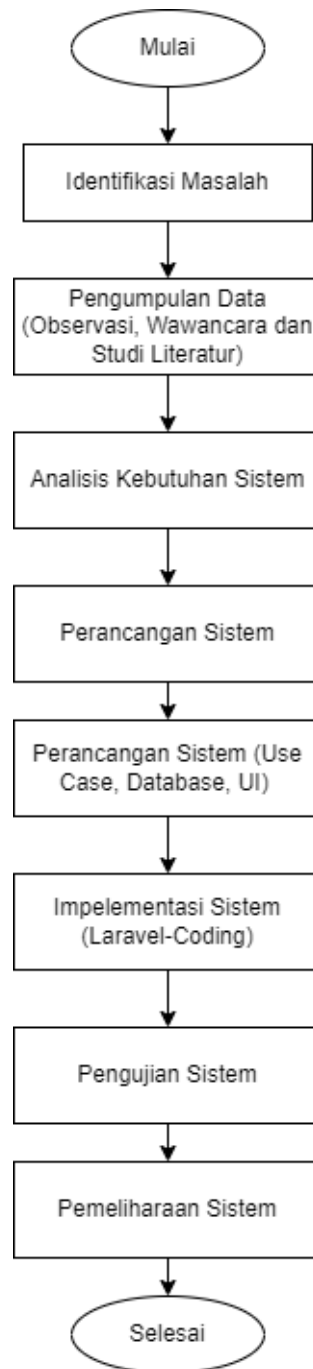
Pada tahap ini dilakukan proses pembangunan sistem berdasarkan desain yang telah dibuat. Sistem dikembangkan menggunakan bahasa pemrograman PHP dengan *framework Laravel* serta menggunakan *Database MySQL* sebagai media penyimpanan data.

4. Pengujian (*Testing*)

Tahap pengujian dilakukan untuk memastikan bahwa sistem yang telah dibangun dapat berjalan dengan baik sesuai dengan kebutuhan yang telah ditentukan. Pengujian dilakukan untuk menemukan kesalahan atau bug pada sistem agar dapat diperbaiki sebelum digunakan.

- a. Pemeliharaan (*Maintenance*)

Tahap pemeliharaan dilakukan setelah sistem diimplementasikan, dengan tujuan untuk memperbaiki kesalahan yang tidak terdeteksi sebelumnya serta melakukan pengembangan sistem.



Gambar 3. 2 Prosedur Penelitian
Sumber : Hasil Perancangan Penulis (2026)

3.3 Metode Penelitian

Metode penelitian yang diterapkan pada penelitian ini adalah metode kualitatif. Metode ini digunakan untuk memperoleh pemahaman secara mendalam mengenai kondisi serta kendala yang terdapat dalam proses pengelolaan data akademik dan administrasi di Ruang Belajar Sa'idah. Pengumpulan informasi dilakukan melalui kegiatan observasi langsung, wawancara dengan pihak terkait, serta kajian terhadap berbagai sumber literatur yang relevan.

Penelitian ini menggunakan pendekatan Rekayasa Perangkat Lunak (RPL) sebagai pendekatan dalam penyelesaian masalah, karena penelitian tidak hanya fokus pada identifikasi dan analisis permasalahan yang terjadi, tetapi mencakup proses perancangan serta pembangunan sistem informasi Ruang Belajar Sa'idah berbasis *website* sebagai bentuk solusi terhadap permasalahan tersebut.

Tahapan pengembangan sistem pada penelitian ini menggunakan metode *Waterfall*. Pemilihan metode *Waterfall* didasarkan pada karakteristiknya yang memiliki alur kerja secara berurutan dan terstruktur, sehingga setiap tahapan pengembangan dapat dilakukan secara sistematis, mulai dari analisis kebutuhan, perancangan, implementasi, pengujian, hingga tahap pemeliharaan sistem.

3.3.1 Teknik Pengumpulan Data

Proses pengumpulan data pada penelitian ini bertujuan untuk memperoleh informasi dan kebutuhan yang diperlukan sebagai dasar dalam pengembangan sistem informasi Ruang Belajar Sa'idah berbasis *website*. Teknik pengumpulan data yang diterapkan dalam penelitian ini terdiri dari beberapa metode, yaitu sebagai berikut:

1. Observasi

Observasi dilakukan secara langsung di Ruang Belajar Sa'idah. Melalui observasi, peneliti dapat mengetahui alur pengolahan data serta permasalahan yang terjadi pada proses pengelolaan data akademik dan administrasi.

2. Wawancara

Wawancara dilakukan secara langsung kepada pengelola Ruang Belajar Sa'idah. Tujuannya untuk mengetahui informasi mengenai kebutuhan sistem yang akan dikembangkan beserta kendala yang terjadi dalam proses pengelolaan data akademik dan administrasi.

3. Studi Literatur

Studi literatur digunakan untuk memperkuat dasar teori serta mendukung tahapan pengembangan sistem pada penelitian ini.

3.4 Analisis Kebutuhan (*Requirement Analysis*)

Tahap analisis kebutuhan menjadi tahapan awal dalam penelitian yang bertujuan untuk mengidentifikasi permasalahan yang terjadi serta menentukan kebutuhan sistem yang sesuai dengan kondisi di lapangan. Pada tahap ini dilakukan proses pengumpulan dan analisis informasi terkait kebutuhan fungsional maupun non fungsional sistem melalui kegiatan observasi serta wawancara dengan pengelola Ruang Belajar Sa'idah. Hasil dari tahap analisis

tersebut kemudian digunakan sebagai acuan dalam proses perancangan dan pembangunan sistem informasi akademik yang akan dikembangkan.

3.4.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional menjelaskan kebutuhan berbagai fitur, layanan yang dimiliki sistem untuk menjalankan proses sesuai kebutuhan pengguna. Berdasarkan hasil observasi dan wawancara dengan pihak Ruang Belajar Sa'idah, kebutuhan fungsional sistem informasi akademik yang dibangun adalah sebagai berikut:

a. Kebutuhan Admin

1. Sistem mampu mengelola data siswa termasuk pembentukan NIS secara otomatis dengan format angkatan ditambah tiga digit urutan.
2. Sistem mampu mengelola data pengajar beserta akun *Login* yang dibuat secara bersamaan.
3. Sistem mampu mengelola jadwal pembelajaran dan pengumuman dalam satu menu terintegrasi.
4. Sistem mampu mengelola mata pelajaran yang diajarkan di Ruang Belajar Sa'idah.
5. Sistem mampu mencatat pembayaran infaq siswa dengan pilihan nominal yang tersedia berdasarkan jenis infaq.
6. Sistem mampu melakukan koreksi terhadap data absensi dan nilai yang telah di *Input* pengajar.
7. Sistem mampu melakukan *generate*, penerbitan, dan penarikan kembali *E-raport* siswa.
8. Sistem mampu mengirimkan notifikasi email secara otomatis kepada orang tua saat raport diterbitkan dan infaq dicatat.
9. Sistem mampu mengirimkan rekap harian kepada orang tua yang berisi status kehadiran dan status infaq siswa.

b. Kebutuhan Pengajar

1. Sistem mampu menampilkan jadwal mengajar hari secara otomatis di *Dashboard*.
2. Sistem mampu menerima *Input* absensi siswa per kelas berdasarkan jadwal yang berlaku dengan tampilan *checkbox* horizontal.
3. Sistem mampu menerima *Input* nilai siswa berdasarkan kelas dan mata pelajaran dengan jenis nilai Mingguan, UTS, dan UAS.
4. Sistem mampu menghitung rata-rata nilai secara otomatis setelah nilai di *Input*.
5. Sistem mampu menerima *Input* materi harian beserta lampiran *file* dan mengirimkan notifikasi email kepada seluruh orang tua di kelas tersebut.

c. Kebutuhan Orang Tua

1. Sistem mampu menampilkan informasi jadwal pelajaran anak sesuai kelasnya.
2. Sistem mampu menampilkan rekap dan riwayat absensi anak.
3. Sistem mampu menampilkan nilai anak per mata pelajaran beserta rata-ratanya.
4. Sistem mampu menampilkan materi harian yang telah disampaikan pengajar.

5. Sistem mampu menampilkan dan mengunduh *E-raport* anak yang telah diterbitkan dalam format PDF.
6. Sistem mampu menampilkan histori pembayaran infaq anak beserta unduhan dalam format PDF.
7. Sistem mampu menerima notifikasi email berisi rekap harian, konfirmasi infaq, nilai dan pemberitahuan materi baru

d. Kebutuhan Penanggung Jawab/Pimpinan

1. Sistem mampu menampilkan laporan akademik berupa rekap absensi dan nilai dalam bentuk visualisasi diagram.
2. Sistem mampu menampilkan laporan infaq dalam bentuk diagram dengan filter berdasarkan bulan dan tahun.
3. Sistem mampu menampilkan analisis kondisi dan rekomendasi tindakan secara otomatis berdasarkan data laporan.
4. Sistem mampu mengunduh laporan akademik dan laporan infaq dalam format PDF.

e. Kebutuhan Umum

1. Sistem mampu mengautentikasi pengguna berdasarkan *username* dan *password* dengan hak akses yang berbeda untuk setiap peran.
2. Sistem mampu menampilkan notifikasi kesalahan yang spesifik saat *Login* gagal.
3. Sistem mampu mengirimkan tautan *Reset Password* melalui email kepada pengguna yang lupa *Password*.
4. Sistem mampu menampilkan pengumuman aktif di *Dashboard* seluruh pengguna.

3.4.2 Analisis Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah suatu kebutuhan yang menjelaskan standar kualitas dan karakteristik yang harus dimiliki oleh sistem agar dapat berjalan dengan baik, meskipun tidak berkaitan secara langsung. Kebutuhan non fungsional sistem informasi akademik Ruang Belajar Sa'idah meliputi kebutuhan perangkat keras, perangkat lunak, dan pengguna.

a. Kebutuhan Perangkat Keras (*Hardware*)

1. Perangkat Keras Pengembang

Tabel 3. 1 Kebutuhan Perangkat Keras (Perangkat Keras Pengembang)

No	Komponen	Spesifikasi yang Digunakan
1	Prosesor	Intel(R) Core(TM) i5-4210M CPU @ 2.60GHz, 2594 Mhz, 2 Core(s), 4 Logical Processor(s)
2	RAM	12 GB
3	Penyimpanan	SSD 128 GB dan HDD 320 GB
4	Koneksi Internet	Minimal 10 Mbps

5	Monitor	Resolusi 1366 x 768 piksel
---	---------	----------------------------

2. Perangkat Keras Pengguna

Tabel 3. 2 Kebutuhan Perangkat Keras (Perangkat Keras Pengguna)

No	Komponen	Spesifikasi Minimum
1	Prosesor	Komputer, Laptop, smartphone
2	RAM	2 GB
3	Koneksi Internet	Minimal 5 Mbps
4	Monitor	Resolusi 360 x 640 piksel (smartphone)

b. Kebutuhan Perangkat Lunak (*Software*)

Tabel 3. 3 Kebutuhan Perangkat Lunak

No	Komponen	Versi	Fungsi
1	XAMPP	8.2.4	Paket <i>server</i> lokal (Apache+MySQL)
2	PHP	8.2.4	Bahasa pemrograman <i>server-side</i>
3	<i>Laravel</i>	12.59.0	<i>Framework</i> PHP berbasis MVC
4	MySQL	10.4.28	Sistem manajemen basis data
5	Composer	2.8.11	Package manager PHP
6	Visual Studio Code	1.127.0	Editor kode program
7	Google Chrome	Terbaru	Browser pengujian
8	AdminLTE	3.16	<i>Template</i> antarmuka berbasis Bootstrap 4
9	<i>DomPDF</i>	3.1.2	<i>Library</i> pembuatan file PDF

c. Kebutuhan Pengguna (*Brainware*)

Pengguna sistem informasi akademik Ruang Belajar Sa'idah terdiri dari empat aktor dengan tingkat kemampuan teknik yang berbeda-beda sebagai berikut:

Tabel 3. 4 Kebutuhan Pengguna

No	Pengguna	Kemampuan yang dibutuhkan
1	Admin	Mampu mengoperasikan komputer dengan baik, memahami alur pengelolaan data akademik, dan mampu menggunakan aplikasi berbasis <i>Website</i> .
2	Pengajar	Mampu mengoperasikan perangkat komputer atau smartphone, memahami penggunaan formulir digital absensi, nilai, dan materi.
3	Orang Tua	Mampu mengakses internet menggunakan smartphone atau komputer, mampu memeriksa email untuk menerima notifikasi.
4	Penanggung Jawab/Pimpinan	Mampu mengoperasikan komputer, mampu membaca dan menginterpretasikan laporan dalam bentuk diagram dan tabel.

3.5 Perancangan Sistem (*Design*)

Perancangan sistem merupakan tahapan lanjutan yang dilakukan setelah kebutuhan sistem berhasil dianalisis. Pada tahap ini, dilakukan penyusunan desain sistem untuk menggambarkan proses kerja, fitur yang tersedia, serta hubungan interaksi antara pengguna dengan sistem yang akan dibangun. Proses perancangan tersebut menggunakan UML (*Unified Modeling Language*) sebagai media pemodelan untuk membantuk menggambarkan struktur dan alur sistem secara visual sebelum memasuki tahapan implementasi.

Perancangan sistem pada penelitian ini meliputi *Use case diagram*, *Activity Diagram*, *Sequence Diagram*, *Entity Relationship Diagram (ERD)*, *Class Diagram*, serta rancangan antarmuka sistem (*interface*).

3.5.1 *Use case diagram*

Use case diagram digunakan untuk memvisualisasikan hubungan antara aktor dengan sistem yang dikembangkan. Pada sistem informais akademik berbasis *website* di Ruang Belajar Sa'idah, terdapat empat aktor yang berinteraksi dengan sistem, yaitu Admin, Pengajar, Orang Tua, dan Penanggung Jawab. Setiap aktor memiliki hak akses serta fungsi yang berbeda sesuai dengan tugas dan tanggung jawab masing-masing. Adapun *use case diagram* dari sistem yang dirancang ditunjukkan pada gambar berikut.



Gambar 3. 3 Use case Diagram SIAKAD Ruang Belajar Sa'idah

Sumber : Hasil Perancangan Penulis (2026)

Gambar 3.3 menunjukkan Use case diagram sistem informasi akademik berbasis *website* pada Ruang Belajar Sa'idah. Admin memiliki hak akses untuk mengelola data siswa, data pengajar, data

mata pelajaran, jadwal dan pengumuman, Administrasi infaq, rekap harian, serta *E-raport*. Selain itu, Admin juga bertanggung jawab dalam pengelolaan data pengguna dan penyajian laporan sistem.

Pengajar memiliki akses untuk melakukan *Input* absensi, mengelola materi harian, serta memasukkan nilai siswa. Orang tua dapat melihat informasi akademik, informasi pembayaran infaq, dan *E-raport* siswa melalui sistem dan menerima notifikasi email. Sementara itu, Penanggung Jawab memiliki akses untuk melihat laporan akademik dan laporan Administrasi infaq sebagai sarana pemantauan dan evaluasi. Adapun calon orang tua memiliki hak akses untuk melihat *landing page* dan menggunakan *chatbot* FAQ.

Sebelum mengakses fitur yang tersedia, seluruh aktor diwajibkan melakukan autentikasi melalui halaman *Login*. Setelah proses *Login* berhasil, pengguna akan diarahkan ke halaman utama sesuai dengan hak akses yang dimiliki.

Tabel 3. 5 Deskripsi Aktor Usecase Diagram

No	Aktor	Deskripsi
1	Admin	Mengelola seluruh data sistem seperti data siswa, data pengajar, jadwal dan pengumuman, koreksi absensi, koreksi nilai, data infaq, kelola laporan dan <i>E-raport</i> siswa.
2	Pengajar	Mengelola absensi, nilai, materi harian, dan melihat jadwal pembelajaran.
3	Penanggung Jawab/Pimpinan	Melihat laporan dan memantau data akademik dan data infaq.
4	Orang Tua	Melihat nilai, absensi, jadwal, materi harian, dan <i>E-raport</i> siswa
5	Calon Orang Tua	Melihat <i>landing page</i> dan menggunakan <i>chatbot</i> FAQ

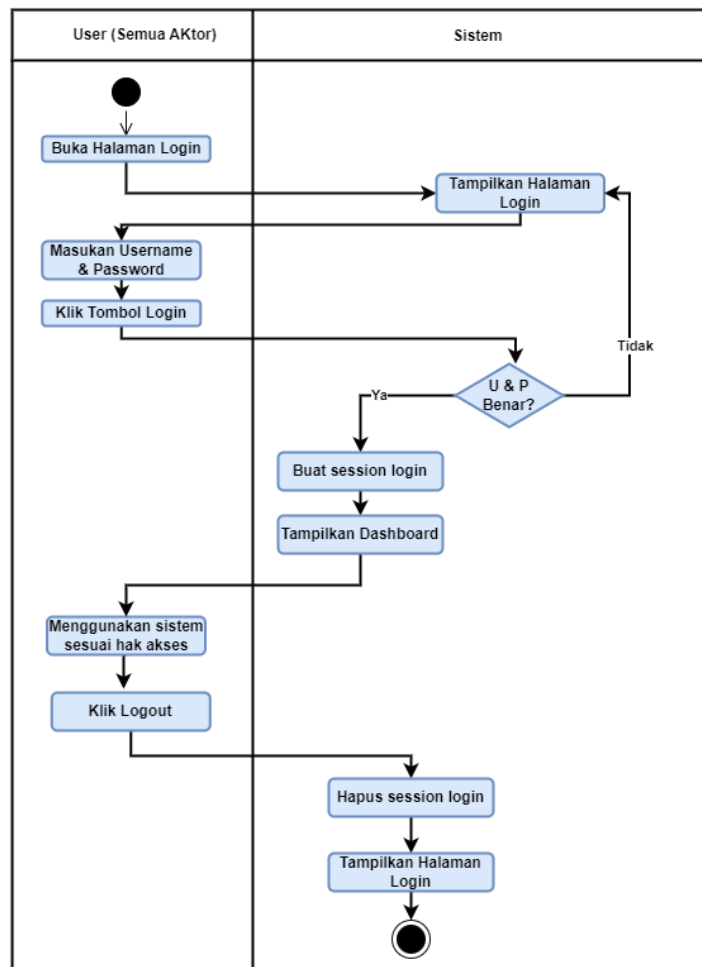
3.5.2 Activity Diagram

Activity Diagram digunakan untuk menggambarkan alur aktivitas atau proses yang terjadi pada sistem yang akan dikembangkan. Diagram ini menunjukkan urutan aktivitas mulai dari proses awal hingga proses akhir yang dilakukan oleh pengguna maupun sistem.

a. Activity Diagram Login

Activity Diagram Login menggambarkan alur proses autentikasi yang harus dilalui oleh seluruh pengguna sistem sebelum dapat mengakses fitur yang tersedia. Proses ini dirancang untuk memastikan keamanan sistem dengan memverifikasi identitas pengguna berdasarkan *username*, *Password*, dan status keaktifan akun. Berikut adalah *Activity Diagram* proses *Login* pada sistem.

Activity Diagram - Login/Logout
Sistem Informasi Akademik Ruang Belajar Sa'idah



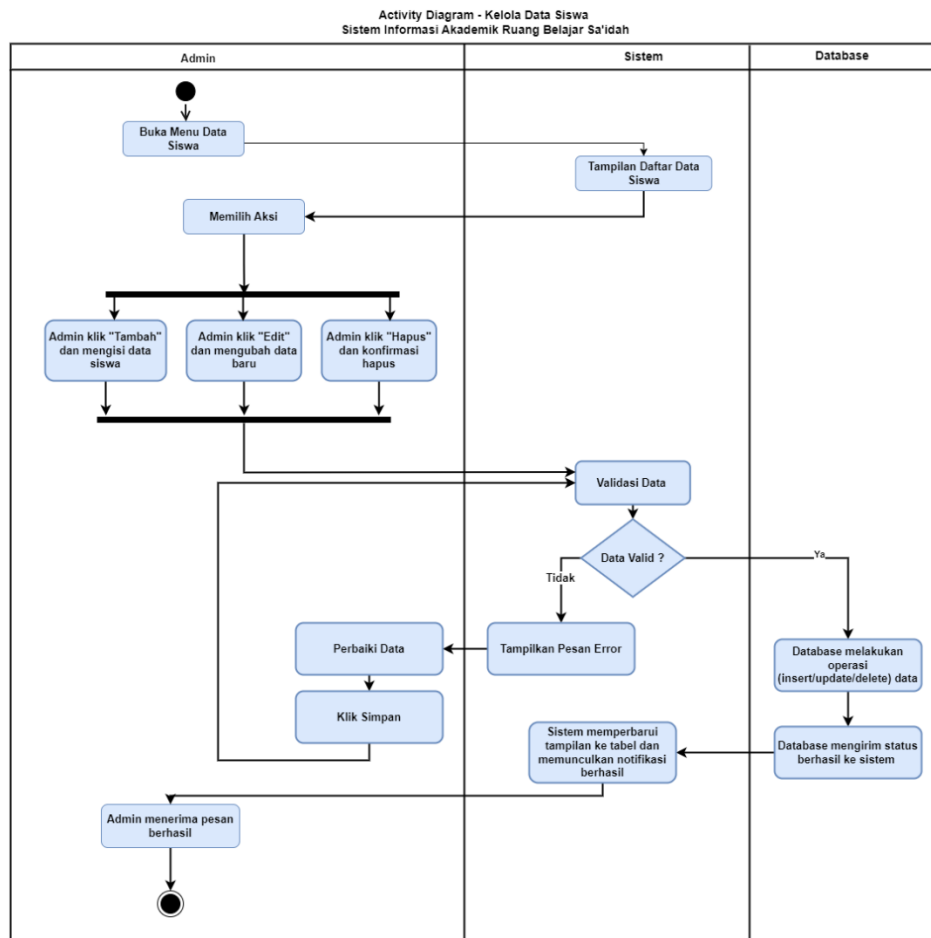
Gambar 3. 4 Activity Diagram Login

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.4 diatas, proses dimulai ketika pengguna mengakses halaman *Login* dan memasukkan *username* dan *Password*. Sistem kemudian memverifikasi apakah *username* yang dimasukkan terdaftar dalam *Database*. Apabila *username* tidak ditemukan, sistem akan menampilkan pesan kesalahan bahwa *username* tidak terdaftar. Jika *username* ditemukan, sistem melanjutkan verifikasi *Password*. Apabila *Password* tidak sesuai, sistem menampilkan pesan kesalahan *Password* salah. Sistem juga memeriksa status keaktifan akun, jika akun tidak aktif maka pengguna tidak dapat masuk dan diberikan notifikasi untuk menghubungi Admin. Jika seluruh verifikasi berhasil, sistem akan mengarahkan pengguna ke *Dashboard* yang sesuai berdasarkan *role* masing-masing.

b. Activity Diagram Kelola Data Siswa

Activity Diagram pengelolaan data siswa menggambarkan rangkaian aktivitas yang dilakukan oleh Admin. Pengelolaan data siswa mencakup beberapa aktivitas, seperti menambahkan data baru, pengubahan, dan penghapusan data berdasarkan data siswa beserta pembuatan akun *Login* untuk orang tua secara otomatis. Berikut adalah Activity Diagram proses pengelolaan data siswa.



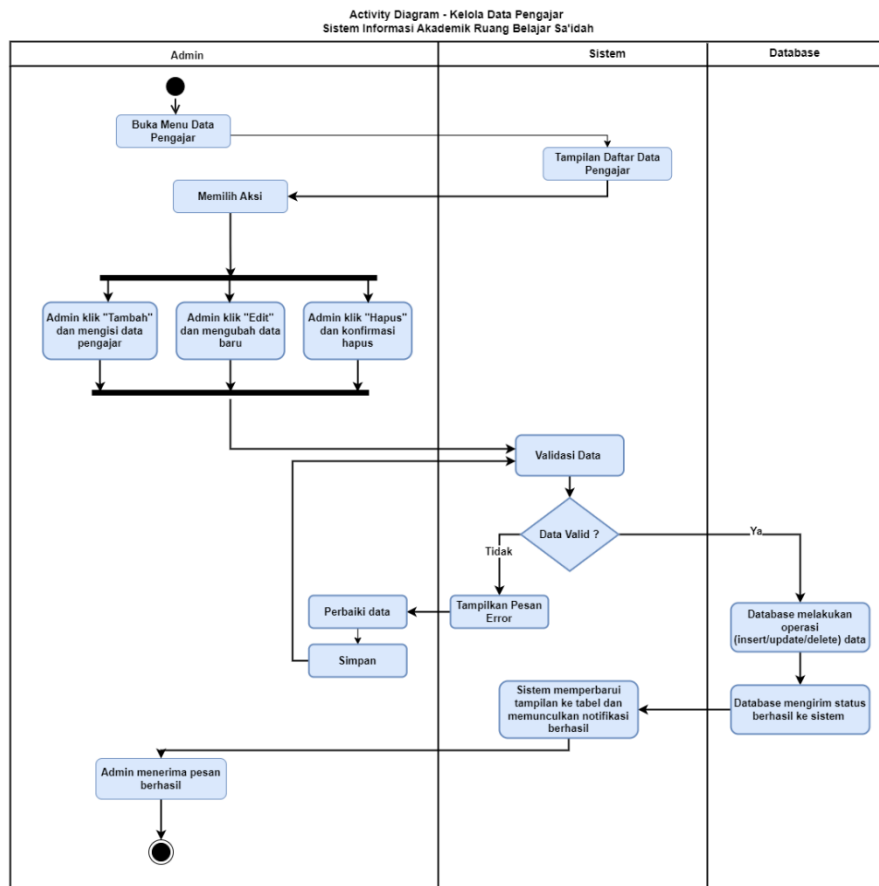
Gambar 3. 5 Activity Diagram Kelola Data Siswa

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.5 diatas, ketika Admin menambahkan data siswa baru, sistem secara otomatis men-*generate* Nomor Induk Siswa menggunakan format tahun berjalan ditambah tiga digit urutan, contohnya 2026001. Bersamaan dengan penyimpanan data siswa, sistem juga secara otomatis membuat akun *Login* untuk orang tua siswa yang bersangkutan. Email orang tua yang di*Input* pada data siswa akan digunakan sebagai tujuan pengiriman notifikasi sistem. Admin dapat mengubah data siswa kapan saja termasuk memperbarui email orang tua, serta dapat menghapus data siswa yang jika dihapus maka akun orang tua terkait juga akan terhapus sistem.

c. Activity Diagram Kelola Data Pengajar

Activity Diagram pengelolaan data pengajar menggambarkan rangkaian aktivitas yang dilakukan oleh Admin. Rangkaian aktivitas ini mencakup penambahan, pengubahan, dan penghapusan data pengajar beserta pengelolaan akun *Login* yang terintegrasi. Berikut adalah Activity Diagram proses pengelolaan data pengajar.



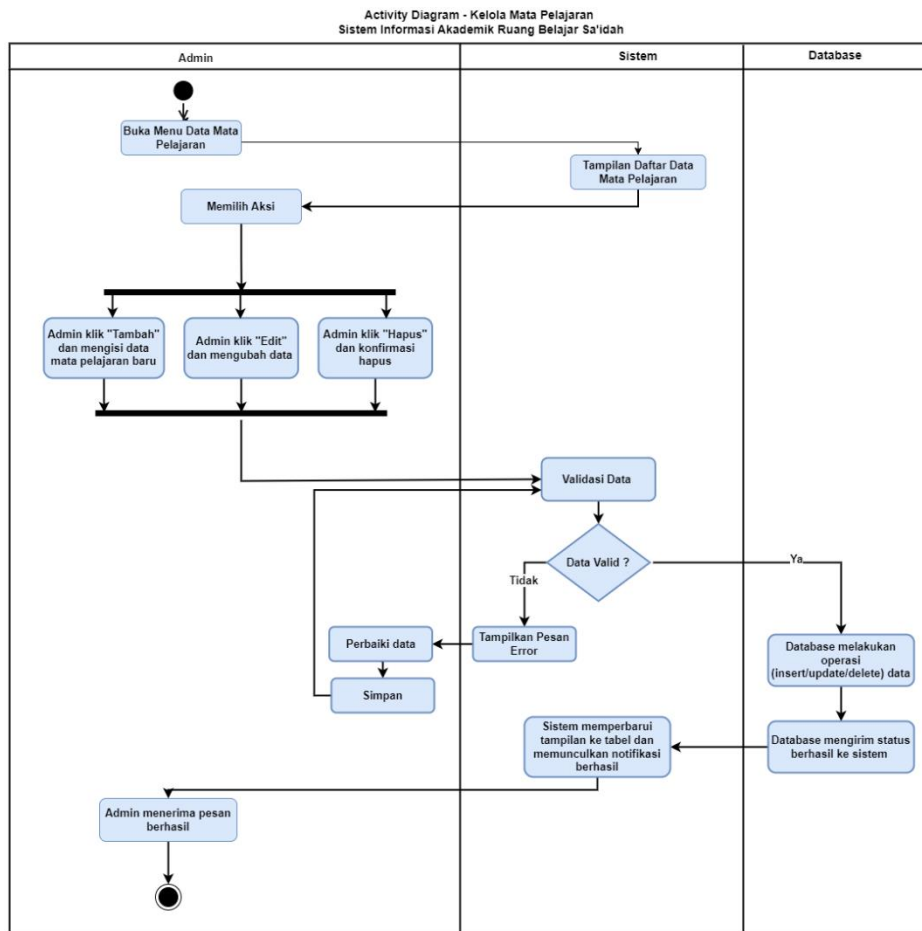
Gambar 3. 6 Activity Diagram Kelola Data Pengajar

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.6 diatas, Admin mengisi data pengajar meliputi NIK, nomor handphone, dan alamat. Saat menambahkan pengajar baru, Admin juga menentukan *username* dan *password* untuk akun *Login* pengajar tersebut. Sistem secara otomatis membuat akun *Login* pengajar bersamaan dengan penyimpanan data. Admin dapat memperbarui data pengajar termasuk mengganti *password* kapan saja, serta dapat menghapus data pengajar yang secara otomatis juga menghapus akun *Login* yang terkait.

d. Activity Diagram Kelola Mata Pelajaran

Activity Diagram kelola mata pelajaran menggambarkan aktivitas Admin dalam mengelola data mata pelajaran yang digunakan dalam kegiatan pembelajaran di Ruang Belajar Sa'idah.



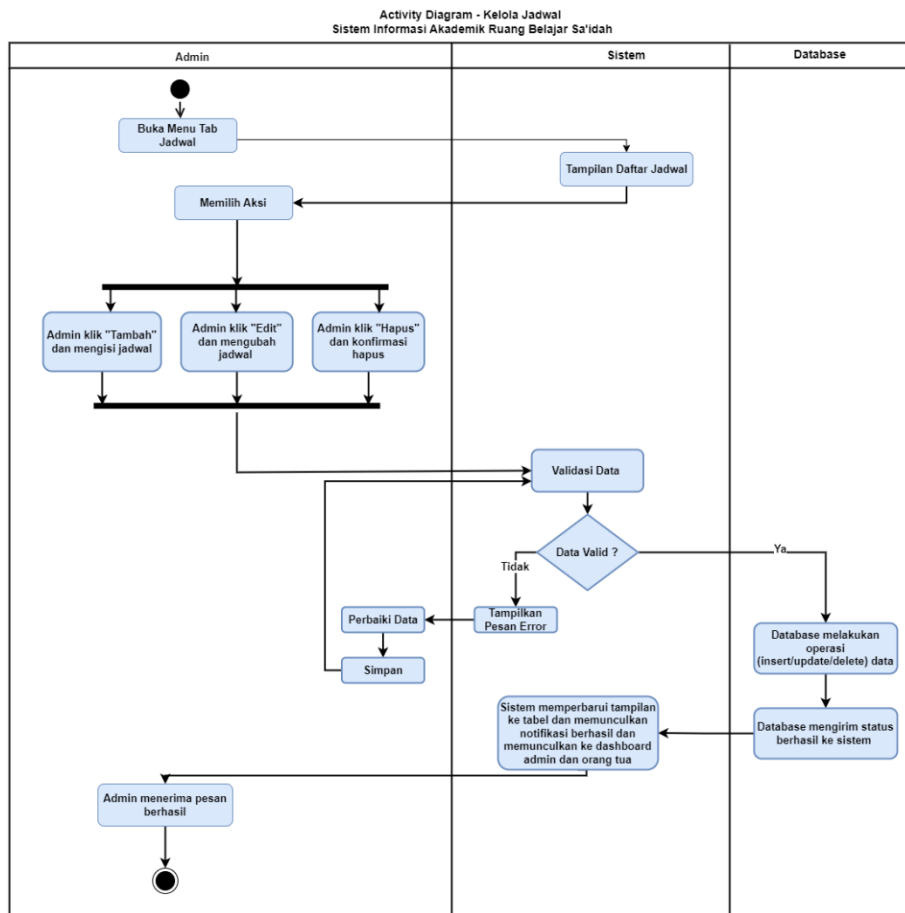
Gambar 3. 7 Activity Diagram Kelola Mata Pelajaran

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.7 diatas, menunjukkan alur pengelolaan data mata pelajaran pada sistem. Proses dimulai ketika Admin memilih menu mata pelajaran. Admin dapat menambahkan, mengubah, atau menghapus data mata pelajaran sesuai kebutuhan. Data yang telah dimasukkan akan divalidasi oleh sistem sebelum disimpan ke dalam basis data. Setelah berhasil disimpan, sistem akan menampilkan notifikasi bahwa proses pengelolaan data telah berhasil dilakukan.

e. Activity Diagram Kelola Jadwal dan Pengumuman

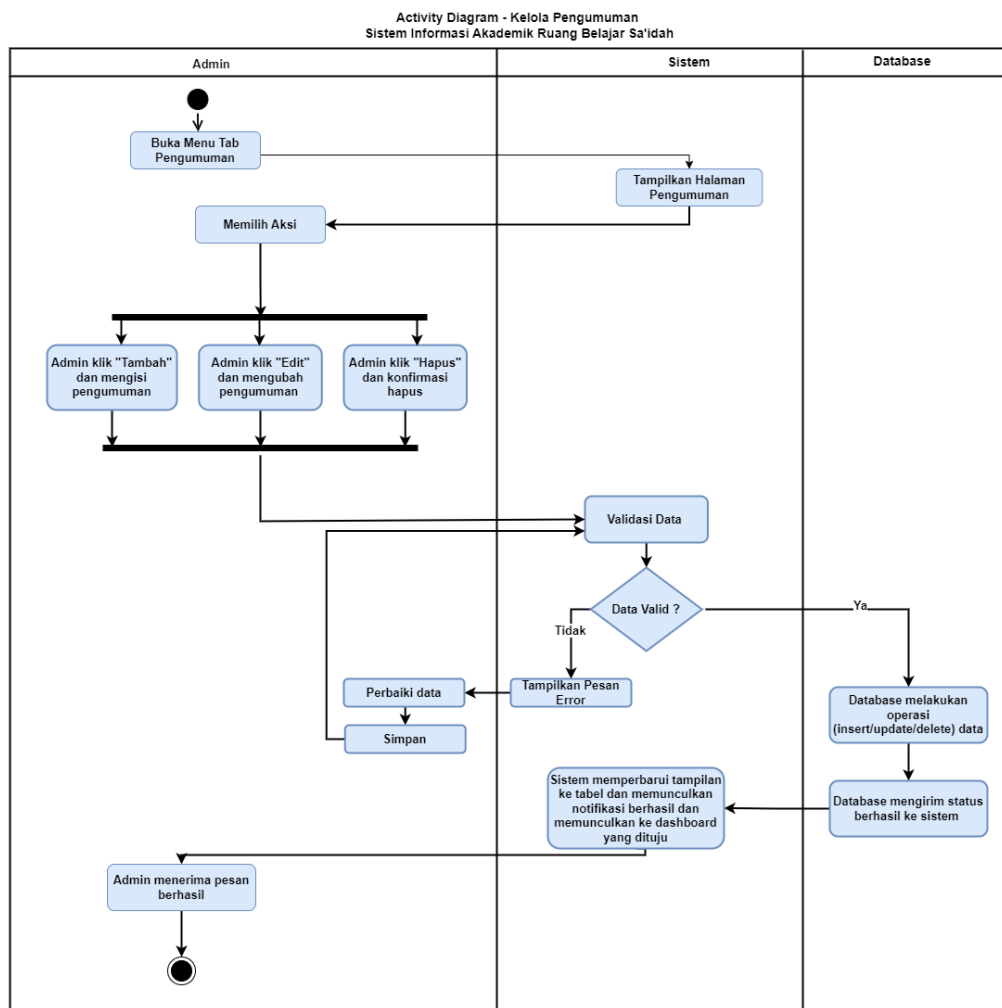
Activity Diagram kelola jadwal dan pengumuman alur proses pengelolaan jadwal mengajar dan pengumuman yang dilakukan oleh Admin. Kedua fitur ini digabungkan dalam satu menu dengan dua tab terpisah karena memiliki keterkaitan dalam memberikan informasi kepada seluruh pihak yang terlibat dalam kegiatan pembelajaran. Berikut adalah Activity Diagram proses pengelolaan jadwal dan pengumuman.



Gambar 3. 8 Activity Diagram Kelola Jadwal

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.8 diatas, pada pengelolaan jadwal, Admin mengisi data jadwal meliputi kelas, mata pelajaran, pengajar, hari, jam mulai, dan jam selesai. Sistem melakukan pengecekan duplikasi untuk mencegah jadwal yang identik dari pengajar yang sama di kelas dan waktu yang persis sama. Validasi ini tidak membatasi dua pengajar berbeda untuk mengajar di kelas yang sama pada waktu yang sama, karena sistem Ruang Belajar Sa'idah memungkinkan beberapa pengajar mengajar secara bersamaan. Jadwal yang tersimpan akan otomatis tampil di *Dashboard* pengajar dan orang tua.



Gambar 3. 9 Activity Diagram Kelola Pengumuman

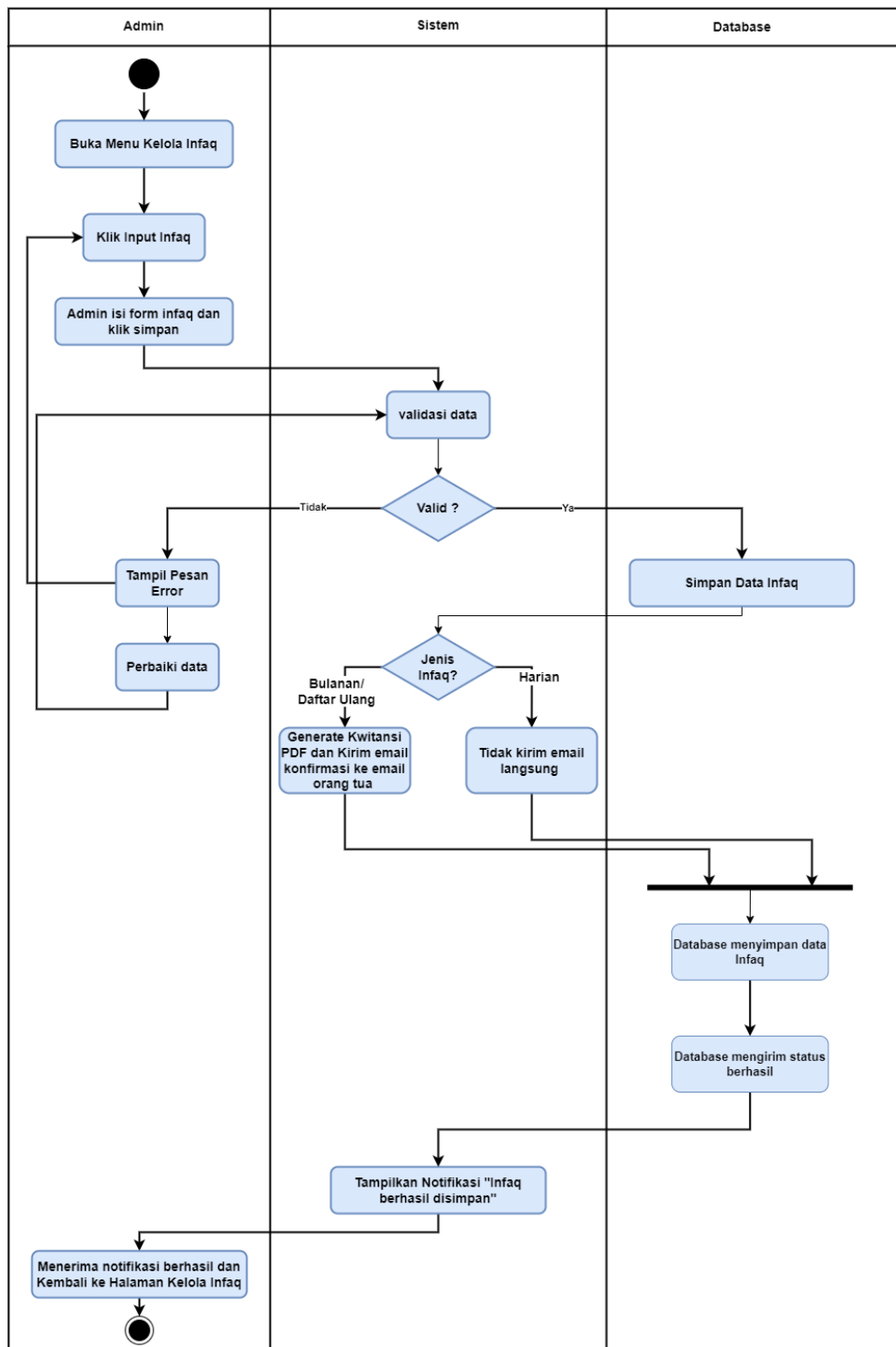
Sumber : Hasil Perancangan Penulis (2026)

Pada pengelolaan pengumuman berdasarkan Gambar 3.9 diatas, Admin mengisi judul, isi, tanggal mulai, tanggal selesai, jenis pengumuman yaitu Libur, Ulangan, atau Umum, serta dapat menentukan pengumuman ditujukan untuk kelas tertentu atau semua kelas. Pengumuman yang aktif akan otomatis muncul di *Dashboard* seluruh *role* yang relevan.

f. Activity Diagram Kelola Infaq

Activity Diagram kelola infaq menggambarkan proses pencatatan pembayaran infaq siswa yang dilakukan oleh Admin, termasuk penanganan notifikasi email berdasarkan jenis infaq yang dicatat. Berikut adalah Activity Diagram proses kelola infaq.

Activity Diagram - Kelola Infaq
Sistem Informasi Akademik Ruang Belajar Sa'idah



Gambar 3. 10 Activity Diagram Kelola Infaq

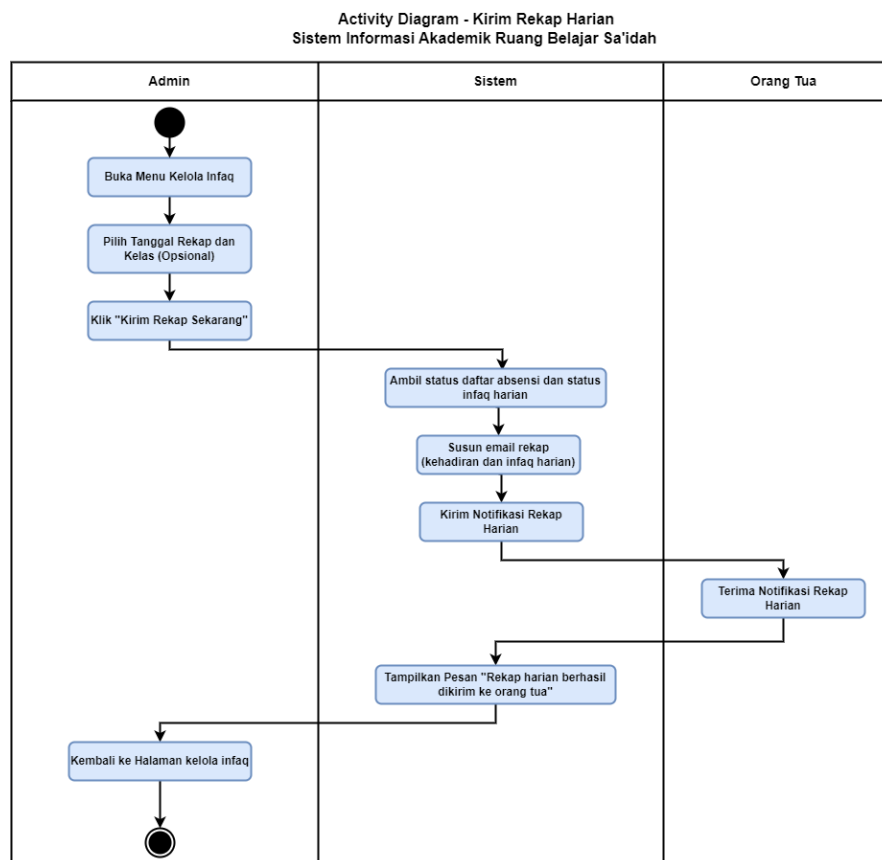
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.10 diatas, Admin memilih siswa yang akan dicatat infaqnya, kemudian memilih jenis infaq yang terdiri dari Harian, Bulanan, atau Daftar Ulang. Setelah jenis dipilih, sistem secara otomatis menampilkan pilihan nominal yang telah ditetapkan sesuai jenis infaqnya. Setelah data disimpan, sistem melakukan percabangan penanganan notifikasi berdasarkan jenis infaq. Untuk

infaq berjenis Harian, sistem tidak langsung mengirimkan email karena informasi pembayaran akan digabungkan bersama status kehadiran melalui fitur kirim rekap harian yang dipicu secara manual oleh Admin. Sedangkan untuk infaq berjenis Bulanan atau Daftar Ulang, sistem langsung mengirimkan email konfirmasi beserta lampiran kwitansi PDF ke alamat email_orangtua siswa yang bersangkutan pada saat itu juga.

g. Activity Diagram Rekap Harian

Activity Diagram kirim rekap harian menggambarkan alur proses pengiriman rekap gabungan kehadiran dan status infaq harian siswa yang dilakukan oleh Admin setelah seluruh data absensi dan infaq harian pada hari tersebut selesai dicatat. Berikut adalah Activity Diagram proses kirim rekap harian.



Gambar 3. 11 Activity Diagram Rekap Harian

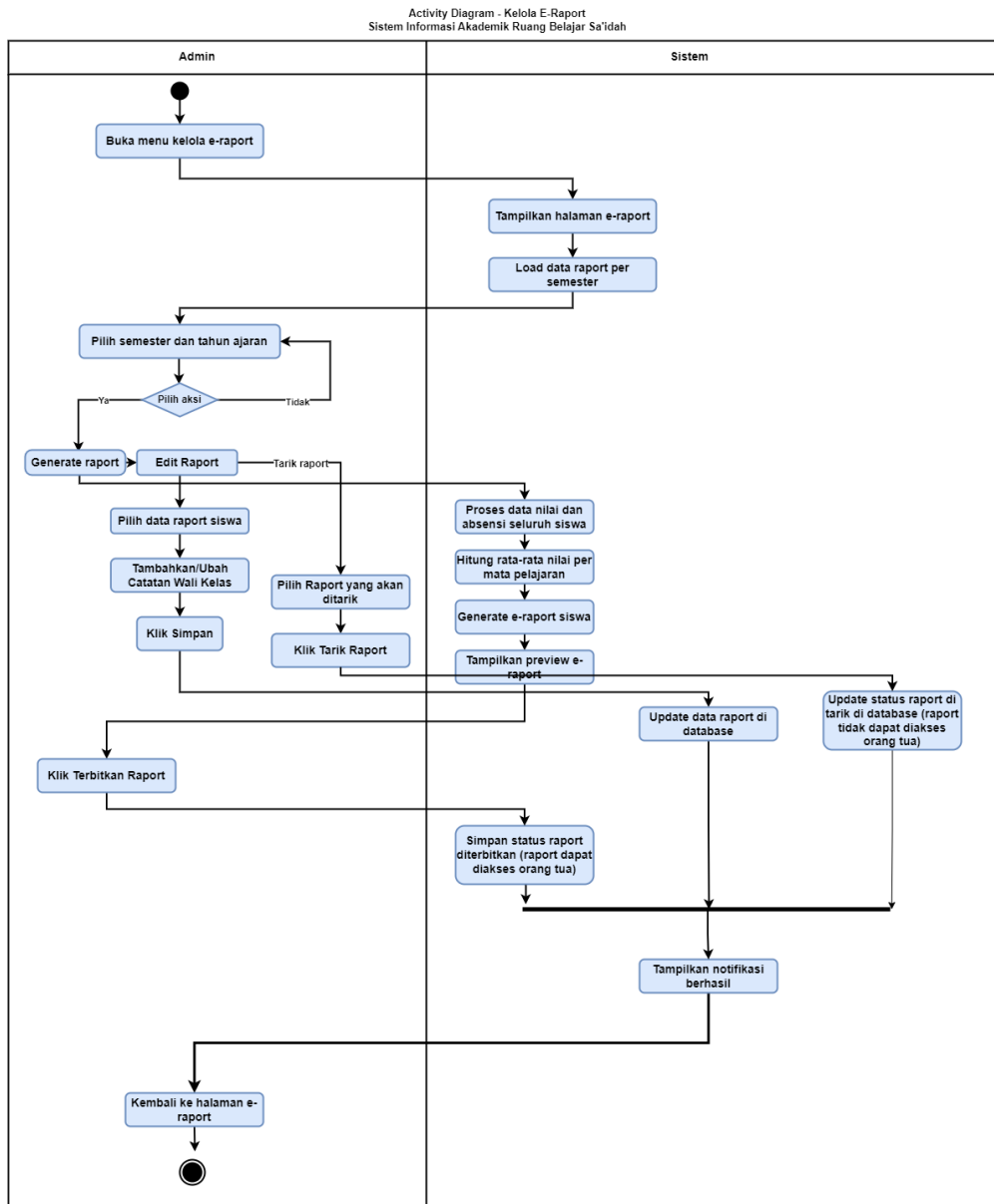
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.11 diatas, setelah seluruh data absensi dan infaq harian selesai dicatat, Admin membuka menu Kelola Infaq, memilih tanggal rekap serta kelas yang diinginkan, kemudian menekan tombol Kirim Rekap Sekarang. Sistem menampilkan konfirmasi terlebih dahulu sebelum memproses pengiriman. Setelah konfirmasi, sistem mengambil seluruh data absensi pada tanggal tersebut, kemudian untuk setiap siswa yang memiliki data absensi, sistem memeriksa apakah infaq harian pada tanggal yang sama sudah tercatat atau belum. Sistem menyusun satu email rekap yang berisi gabungan informasi status kehadiran dan status pembayaran infaq harian, kemudian

mengirimkannya ke alamat email_orangtua masing-masing siswa. Bagi siswa yang hadir namun belum membayar infaq, email rekap juga menyertakan pengingat agar pembayaran dilakukan pada pertemuan berikutnya.

h. Activity Diagram Kelola E-raport

Activity Diagram kelola E-raport menggambarkan proses pengelolaan raport elektronik yang dilakukan oleh Admin. Proses ini mencakup pembuatan, penambahan catatan, penerbitan, serta penarikan kembali raport yang telah diterbitkan. Berikut adalah Activity Diagram proses pengelolaan E-raport.



Gambar 3. 12 Activity Diagram Kelola E-raport

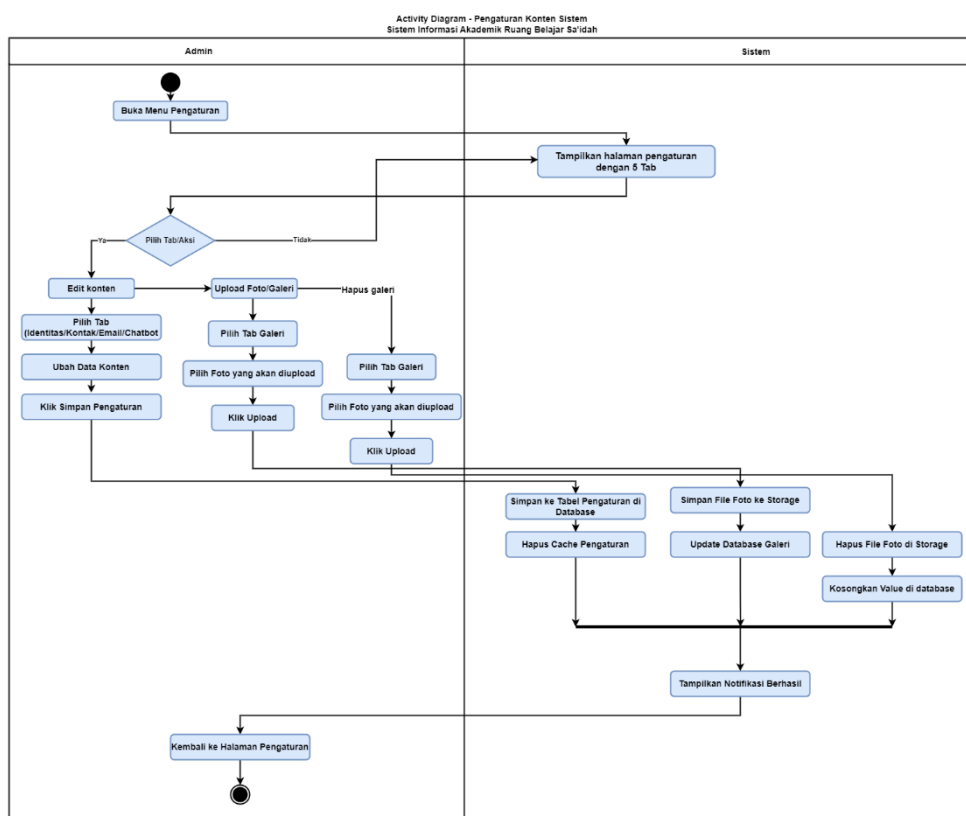
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.12 diatas, Admin terlebih dahulu melakukan proses *generate* raport berdasarkan semester, tahun ajaran, dan kelas yang dipilih. Sistem membuat data raport dalam status *Draft* untuk setiap siswa yang belum memiliki raport pada periode tersebut. Admin kemudian dapat

menambahkan catatan wali kelas pada masing-masing raport. Setelah raport siap, Admin menerbitkan raport sehingga statusnya berubah menjadi diterbitkan. Pada saat penerbitan, sistem secara otomatis mengirimkan notifikasi email beserta *file* PDF raport kepada orang tua masing-masing siswa. Orang tua yang sudah menerima email dapat langsung melihat dan mencetak raport tanpa perlu *Login* ke sistem. Admin juga dapat menarik kembali raport yang telah diterbitkan jika diperlukan perbaikan.

i. Activity Diagram Pengaturan Konten Sistem

Activity Diagram pengaturan konten sistem menggambarkan alur proses pengelolaan konten *Landing Page* dan *Template* email notifikasi yang dilakukan oleh Admin melalui empat tab pengaturan yang tersedia pada menu Pengaturan. Fitur ini memungkinkan Admin mengubah konten sistem secara langsung tanpa perlu mengubah kode program. Berikut adalah *Activity Diagram* proses pengaturan konten sistem.



Gambar 3. 13 Activity Diagram Pengaturan Konten Sistem

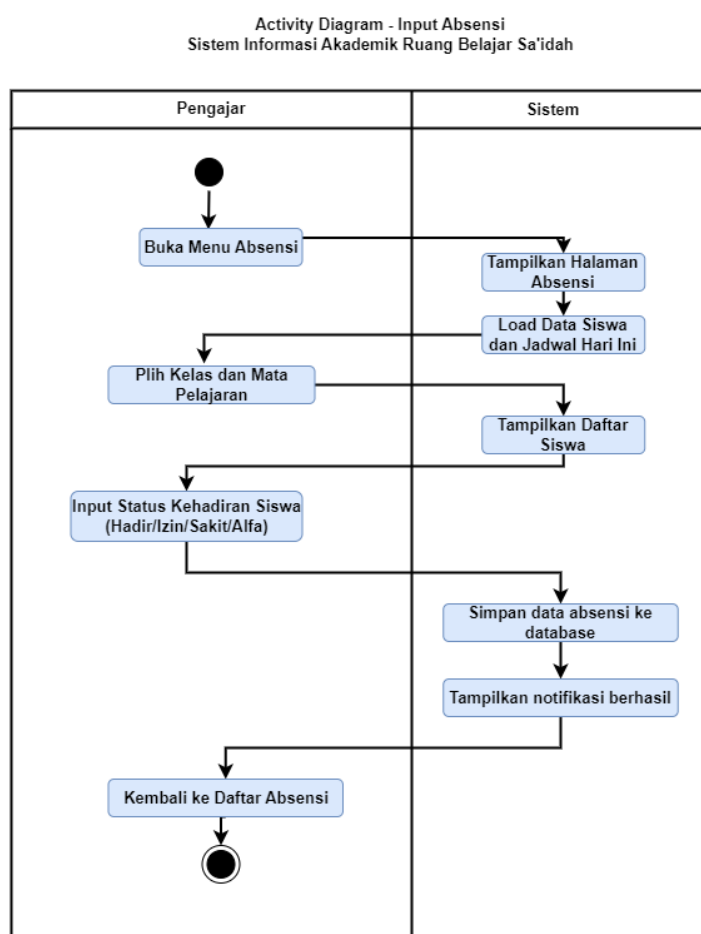
Sumber : Hasil Perancangan Sistem (2026)

Berdasarkan Gambar 3.13 diatas, Admin membuka menu Pengaturan yang menampilkan lima tab yaitu Identitas Lembaga, Kontak dan Lokasi, *Template* Email, Galeri Foto, dan *ChatBot*. Pada tab Identitas, Kontak, *Template* Email, dan *ChatBot*, Admin dapat mengubah konten yang diinginkan kemudian menekan tombol Simpan Pengaturan. Sistem menyimpan setiap perubahan ke tabel pengaturan pada *Database* dan menghapus *cache* sehingga *Landing Page* serta email notifikasi secara otomatis menggunakan konten terbaru. Pada tab Galeri Foto, Admin memilih foto yang ingin

diunggah pada salah satu dari enam slot yang tersedia, kemudian menekan tombol Upload. Sistem memvalidasi jenis dan ukuran *file*, menyimpannya ke dalam direktori penyimpanan, dan memperbarui data pada tabel pengaturan sehingga foto baru langsung tampil pada halaman utama. Admin juga dapat menghapus foto galeri yang sudah tidak diperlukan, dan sistem akan menghapus file dari penyimpanan serta mengosongkan data pada tabel pengaturan.

j. Activity Diagram *Input Absensi*

Activity Diagram *Input absensi* menggambarkan proses pencatatan kehadiran siswa yang dilakukan oleh Pengajar berdasarkan jadwal mengajar yang berlaku pada hari tersebut. Berikut adalah Activity Diagram proses *Input absensi*.



Gambar 3. 14 Activity Diagram *Input Absensi*

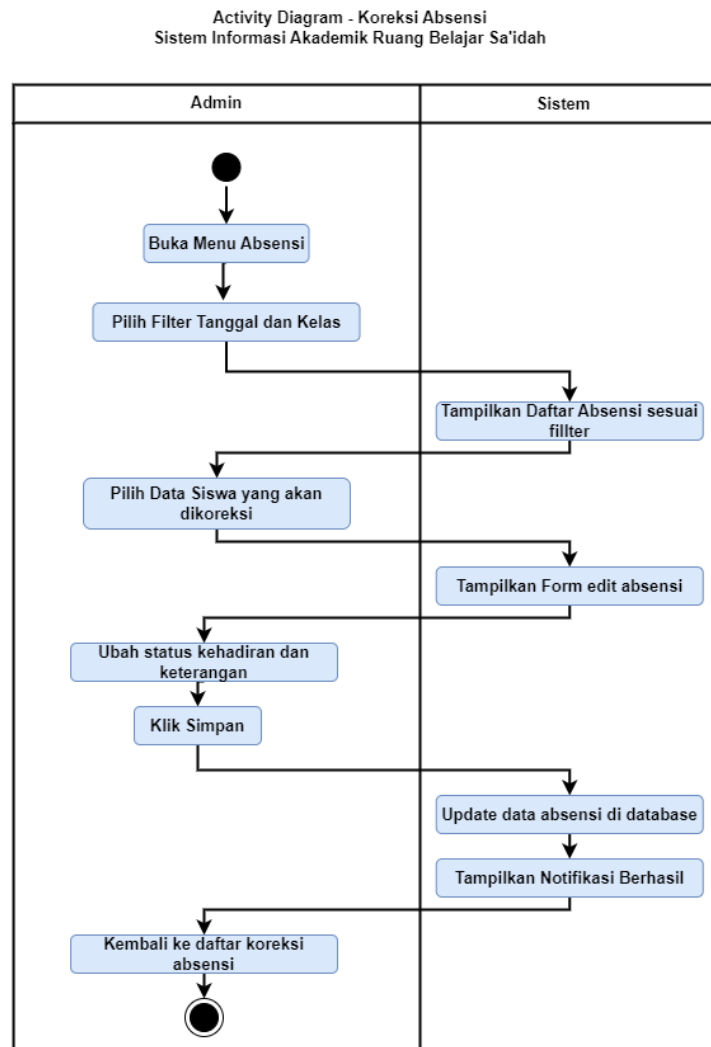
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.14 diatas, Pengajar membuka menu *Input Absensi* yang secara otomatis menampilkan daftar jadwal mengajar hari ini. Pengajar memilih jadwal yang sesuai dengan kelas dan mata pelajaran yang sedang berlangsung, kemudian sistem menampilkan daftar siswa di kelas tersebut secara otomatis. Pengajar mengisi status kehadiran setiap siswa menggunakan tampilan *checkbox* horizontal dengan empat pilihan yaitu Hadir, Izin, Sakit, dan Alfa. Tersedia pula tombol Hadir Semua untuk mempercepat proses *Input* apabila seluruh siswa hadir. Setelah seluruh siswa terisi statusnya, Pengajar menekan tombol Simpan dan sistem menyimpan data absensi ke dalam

Database. Pengajar juga dapat mengedit absensi yang telah diInput sebelumnya apabila terdapat kesalahan.

k. Activity Diagram Koreksi Absensi

Activity Diagram koreksi absensi menggambarkan alur proses perbaikan data kehadiran siswa yang dilakukan oleh Admin apabila terdapat kesalahan Input dari Pengajar. Berikut adalah Activity Diagram proses koreksi absensi.



Gambar 3. 15 Activity Diagram Koreksi Absensi

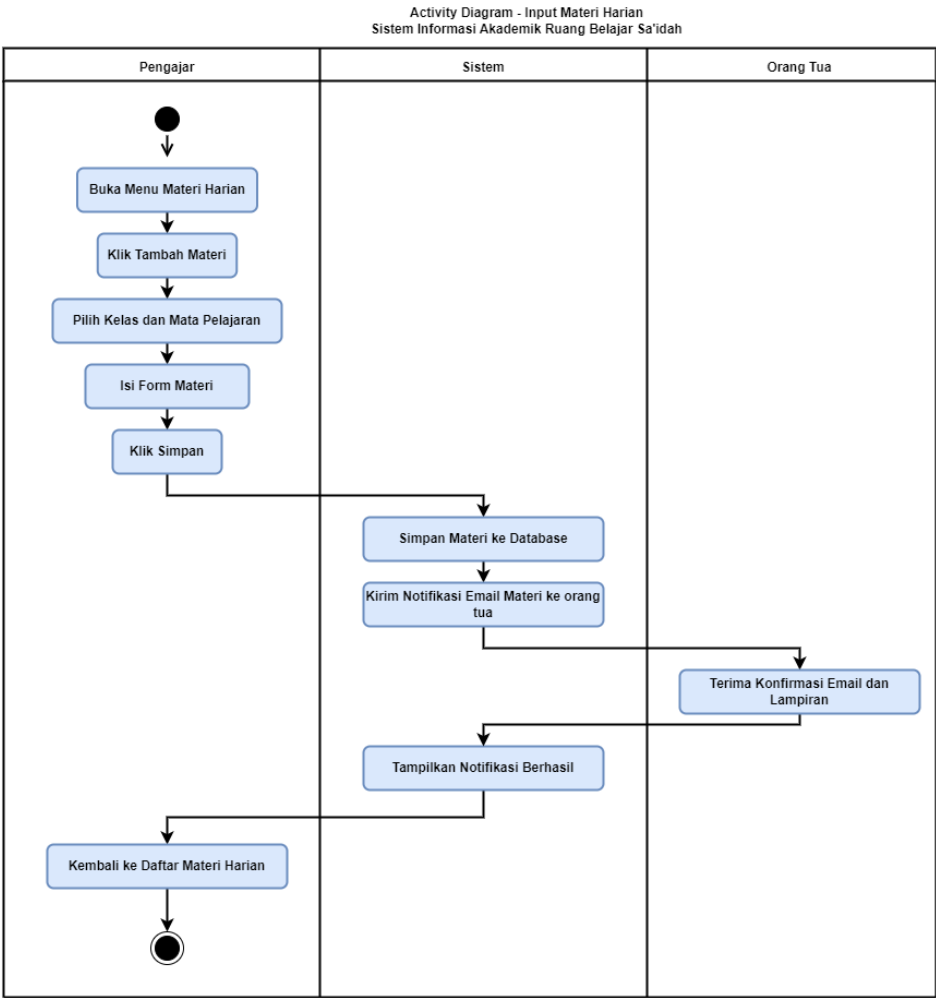
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.15 diatas, Admin membuka menu Koreksi Absensi dan dapat melakukan filter berdasarkan tanggal dan kelas untuk memudahkan pencarian data. Sistem menampilkan seluruh data absensi seluruh filter yang dipilih dalam bentuk tabel. Admin memilih data absensi yang perlu diperbaiki, mengubah status kehadiran dan keterangan yang diperlukan, kemudian menyimpan perubahan. Sistem memperbarui data absensi dan menampilkan pesan konfirmasi bahwa koreksi berhasil dilakukan.

l. Activity Diagram Input Materi Harian

Activity Diagram *Input* materi harian menggambarkan alur proses penambahan materi pembelajaran oleh Pengajar beserta proses pengiriman notifikasi email kepada seluruh orang tua siswa di kelas yang bersangkutan. Berikut adalah Activity Diagram proses *Input* materi harian.

Gambar 3. 16 Activity Diagram *Input* Materi Harian



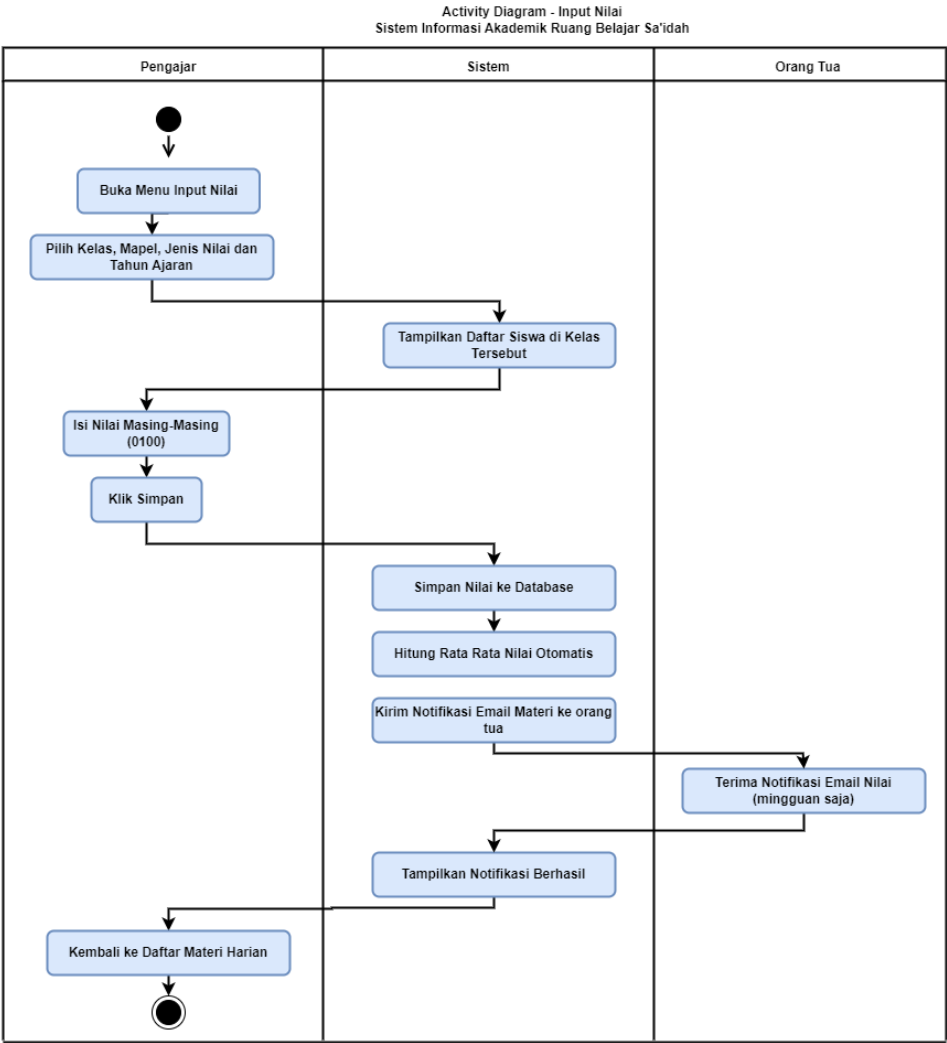
Sumber : Hasil

Perancangan Penulis (2026)

Berdasarkan Gambar 3.16 diatas, Pengajar memilih kelas dan mata pelajaran, kemudian mengisi judul, deskripsi, dan tanggal materi yang akan disampaikan. Pengajar juga dapat mengunggah *file* lampiran berupa PDF, gambar, atau dokumen lainnya. Setelah data tersimpan, sistem secara otomatis mengambil daftar siswa pada kelas tersebut yang memiliki alamat email_orangtua terdaftar, kemudian mengirimkan email notifikasi berisi informasi materi harian beserta *file* lampiran kepada setiap orang tua siswa di kelas tersebut.

m. Activity Diagram *Input* Nilai

Activity Diagram *Input* nilai menggambarkan alur proses peng*Input*an nilai akademik siswa oleh Pengajar beserta proses perhitungan rata-rata dan pengiriman notifikasi email kepada orang tua. Berikut adalah Activity Diagram proses *Input* nilai.



Gambar
3. 17
Activity

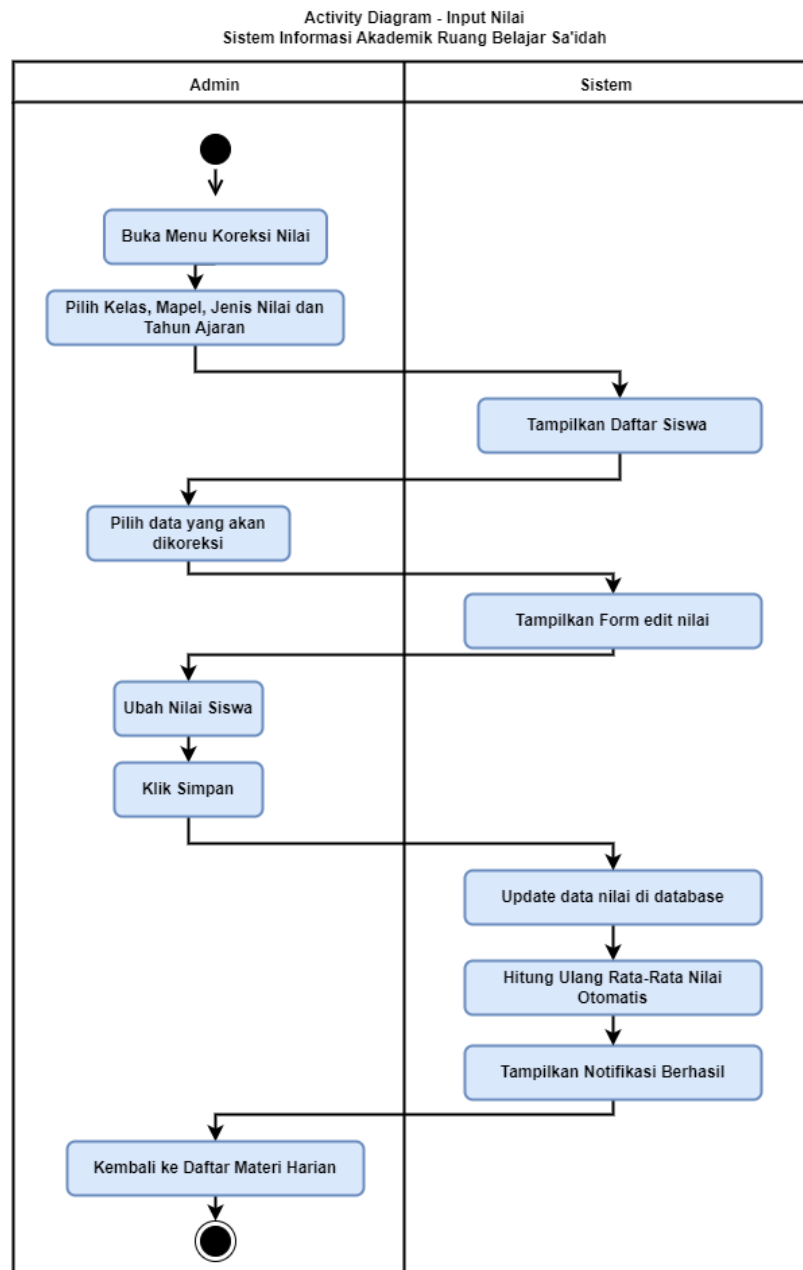
Diagram *Input* Nilai

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.17 diatas, Pengajar memilih kelas, mata pelajaran, jenis nilai yaitu Mingguan, UTS, atau UAS, serta semester dan tahun ajaran yang berlaku. Sistem menampilkan daftar siswa di kelas tersebut secara otomatis. Pengajar mengisi nilai masing-masing siswa dengan rentang nol hingga seratus yang mendukung satu angka desimal. Setelah nilai disimpan, sistem menghitung rata-rata nilai secara otomatis berdasarkan seluruh nilai siswa yang telah di*Input* untuk mata pelajaran, semester, dan tahun ajaran yang sama. Sistem kemudian mengirimkan email notifikasi berisi informasi mata pelajaran, jenis nilai, nilai yang diperoleh, dan predikat kepada orang tua siswa yang memiliki alamat email_orangtua terdaftar.

n. Activity Diagram Koreksi Nilai

Activity Diagram koreksi nilai menggambarkan alur proses perbaikan data nilai siswa yang dilakukan oleh Admin apabila terdapat kesalahan *Input* dari Pengajar. Berikut adalah Activity Diagram proses koreksi nilai.



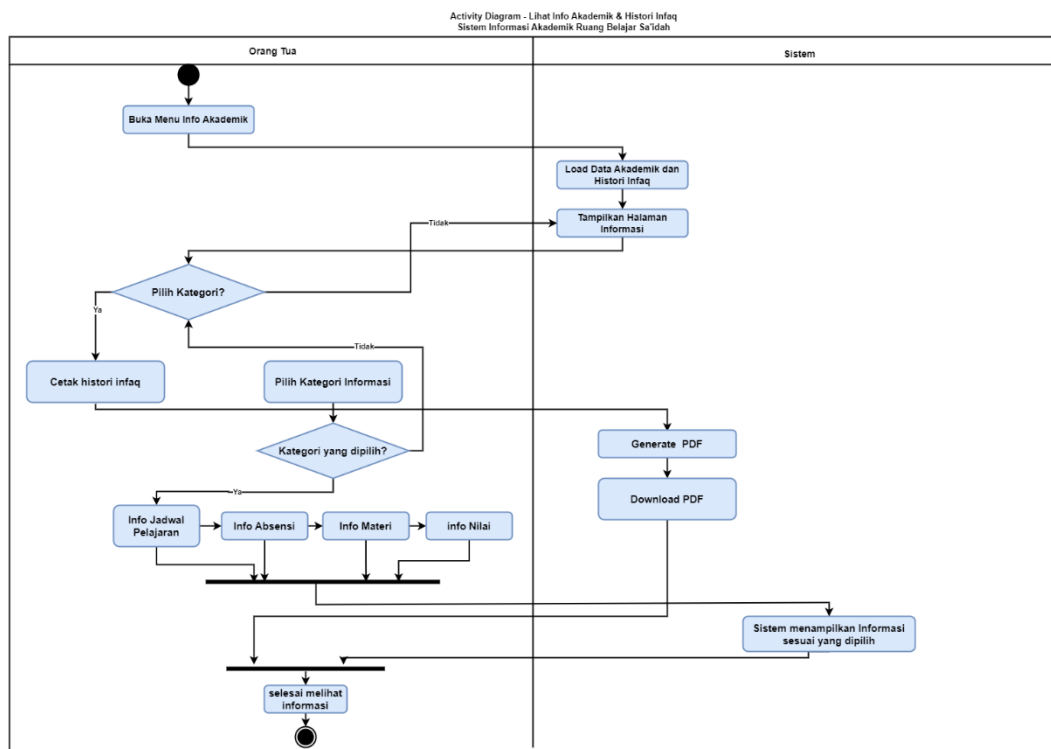
Gambar 3. 18 Activity Diagram Koreksi Nilai

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.18 diatas, Admin membuka menu Koreksi Nilai dan melakukan filter berdasarkan kelas, mata pelajaran, jenis nilai, semester, dan tahun ajaran. Sistem menampilkan tabel nilai seluruh siswa seluruh filter yang dipilih. Admin memilih data nilai yang perlu diperbaiki, mengubah nilai yang salah, kemudian menyimpan perubahan. Sistem memperbarui data nilai di *Database* dan menghitung ulang rata-rata nilai siswa secara otomatis setelah perubahan tersimpan.

o. Activity Diagram Lihat Informasi Akademik dan Infaq

Activity Diagram lihat informasi akademik dan infaq menggambarkan proses orang tua dalam mengakses informasi akademik siswa seperti jadwal, pengumuman, serta informasi pembayaran infaq melalui sistem.



Gambar
3. 19
Activity
Diagram
Lihat

Informasi Akademik dan Infaq

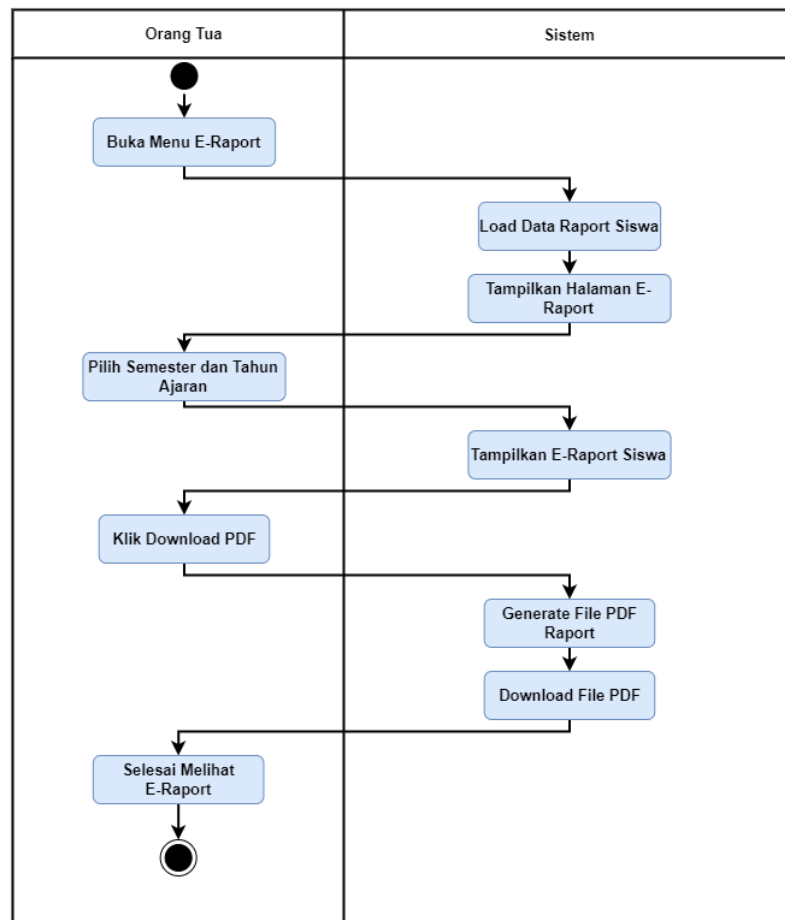
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.19 diatas, setelah *Login*, Orang Tua membuka menu Info Akademik yang menampilkan empat tab informasi. Pada tab Jadwal, Orang Tua dapat melihat jadwal pelajaran lengkap anaknya. Pada tab Absensi, Orang Tua dapat melihat rekap dan riwayat kehadiran anak. Pada tab Nilai, Orang Tua dapat melihat nilai mata pelajaran beserta rata-ratanya. Pada tab Materi Harian, Orang Tua dapat melihat materi yang telah disampaikan Pengajar beserta *file* lampiran yang dapat diunduh. Untuk melihat riwayat infaq, Orang Tua membuka menu Histori Infaq, dapat melakukan filter berdasarkan jenis infaq, dan mengunduh rekap histori infaq dalam format PDF.

p. Activity Diagram Lihat E-raport

Activity Diagram lihat *E-raport* menggambarkan alur proses pengaksesan dan pengunduhan raport elektronik oleh Orang Tua melalui sistem. Berikut adalah Activity Diagram proses lihat *E-raport*.

Activity Diagram - Lihat E-Raport
Sistem Informasi Akademik Ruang Belajar Sa'idah



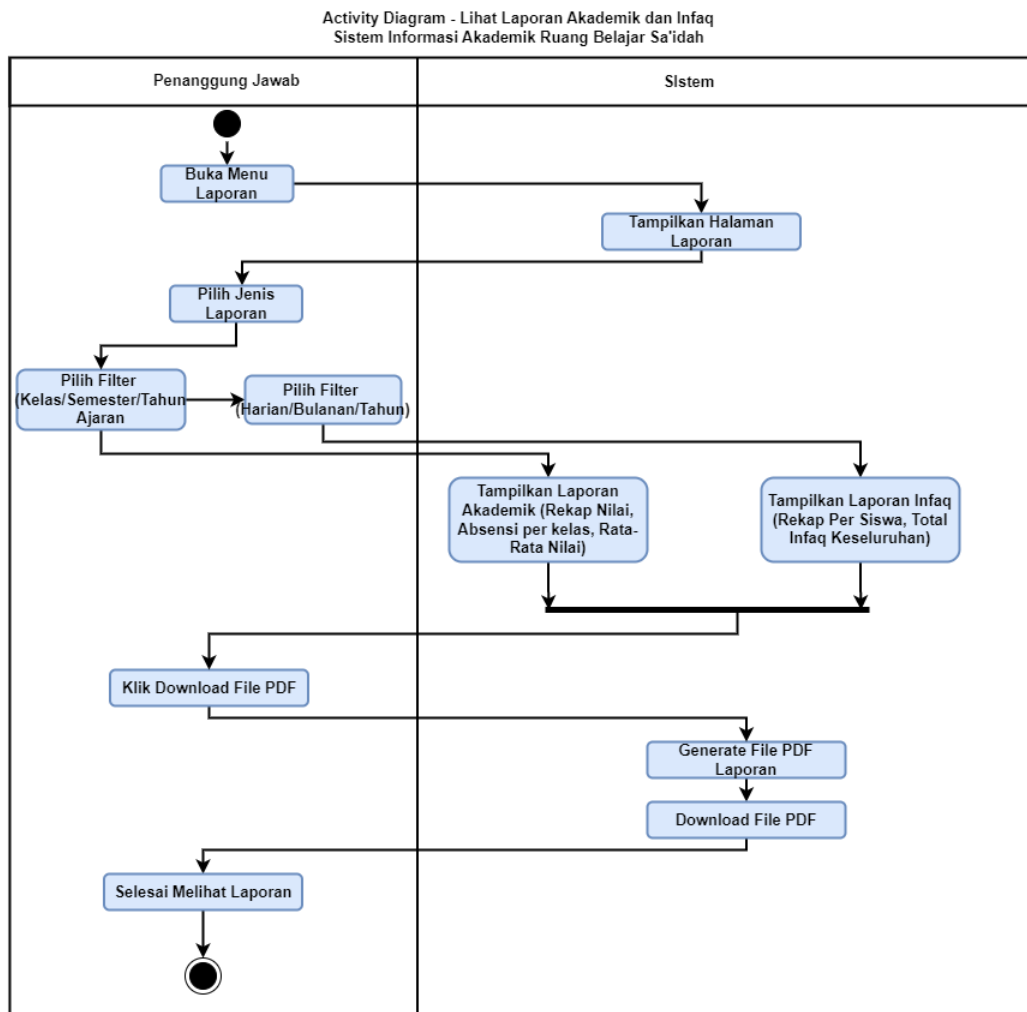
Gambar 3. 20 Activity Diagram Lihat E-raport

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.20 diatas, setelah *Login*, Orang Tua membuka menu *E-raport* yang menampilkan daftar raport yang telah diterbitkan untuk anaknya berdasarkan semester dan tahun ajaran. Orang Tua memilih raport yang ingin dilihat untuk mengakses Detail nilai per mata pelajaran, rekap absensi, dan catatan wali kelas. Orang Tua juga dapat mengunduh raport dalam format PDF langsung dari halaman tersebut. Raport yang belum diterbitkan atau berstatus *Draft* tidak dapat diakses oleh Orang Tua.

q. Activity Diagram Lihat Laporan Akademik dan Infaq

Activity Diagram lihat laporan akademik dan infaq menggambarkan proses pengaksesan dan analisis laporan yang dilakukan oleh Penanggung Jawab melalui sistem. Berikut adalah Activity Diagram proses lihat laporan akademik dan infaq.



Gambar 3. 21 Activity Diagram Lihat Laporan Akademik dan Infaq

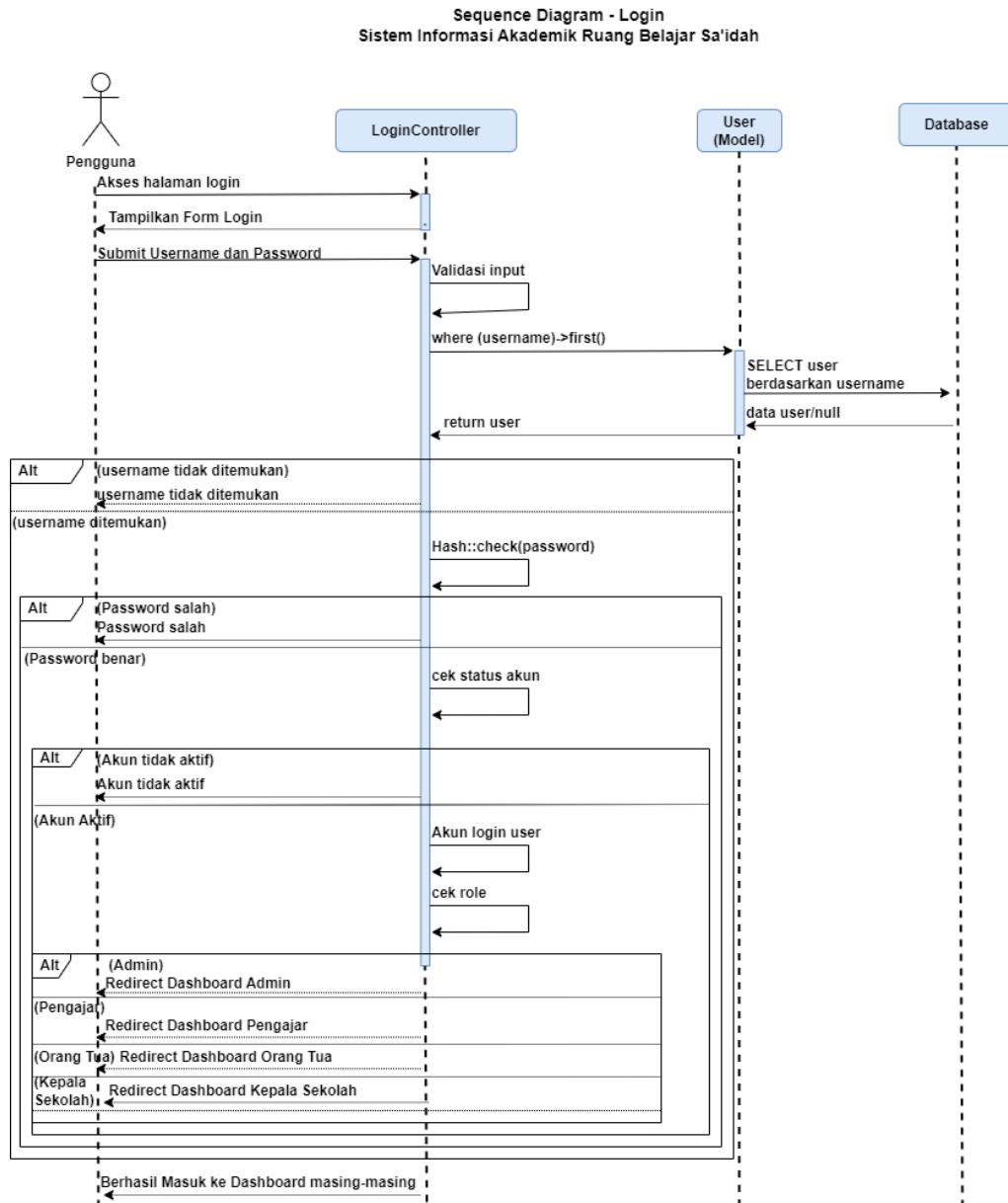
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.21 diatas, Penanggung Jawab dapat mengakses dua jenis laporan yaitu Laporan Akademik dan Laporan Infaq. Pada Laporan Akademik, Penanggung Jawab melakukan filter berdasarkan semester, tahun ajaran, dan kelas. Sistem menampilkan diagram donut rekap absensi keseluruhan, diagram batang nilai rata-rata per mata pelajaran, dan diagram batang absensi per kelas, dilengkapi analisis kondisi dan rekomendasi tindakan yang dihasilkan secara otomatis oleh sistem berdasarkan data yang tersedia. Pada Laporan Infaq, Penanggung Jawab melakukan filter berdasarkan bulan dan tahun. Sistem menampilkan diagram donut infaq per jenis, diagram batang per kelas, dan diagram garis tren infaq dua belas bulan terakhir juga dilengkapi analisis dan rekomendasi otomatis. Kedua laporan dapat diunduh dalam format PDF untuk keperluan arsip dan pelaporan.

3.5.3 Sequence Diagram

a. Sequence Diagram Login

Sequence Diagram Login menggambarkan interaksi antar objek dalam sistem pada saat proses autentikasi pengguna berlangsung. Berikut adalah Sequence Diagram proses Login.



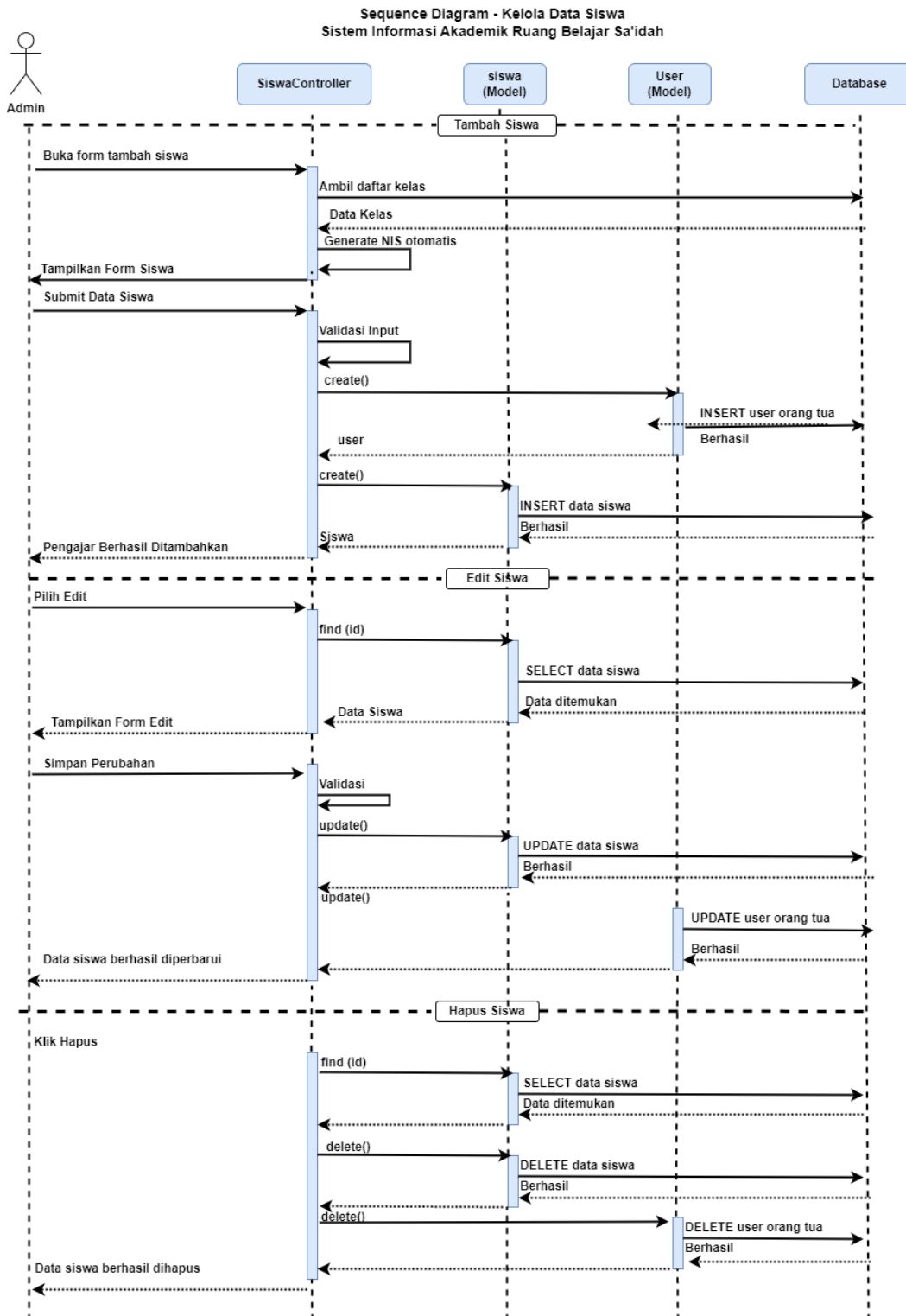
Gambar 3. 22 Sequence Diagram Login

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.22 diatas, pengguna mengirimkan *username* dan *Password* melalui *LoginController*. Sistem mencari data *user* pada *Database* melalui model *User* berdasarkan *username* yang dimasukkan. Apabila *username* tidak ditemukan, sistem menampilkan pesan kesalahan. Apabila ditemukan, sistem melakukan verifikasi *Password* menggunakan *Hash::check*, kemudian memeriksa status keaktifan akun. Jika seluruh verifikasi berhasil, sistem mencatat sesi pengguna dan mengarahkannya ke halaman *Dashboard* sesuai *role* yang dimiliki yaitu Admin, Pengajar, Orang Tua, atau Penanggung Jawab.

b. Sequence Diagram Kelola Data Siswa

Sequence Diagram kelola data Siswa menggambarkan interaksi antar objek pada saat Admin melakukan penambahan, perubahan, dan penghapusan data siswa beserta akun *Login* orang tuanya. Berikut adalah Sequence Diagram proses kelola data siswa.



Gambar 3. 23 Sequence Diagram Kelola Data Siswa

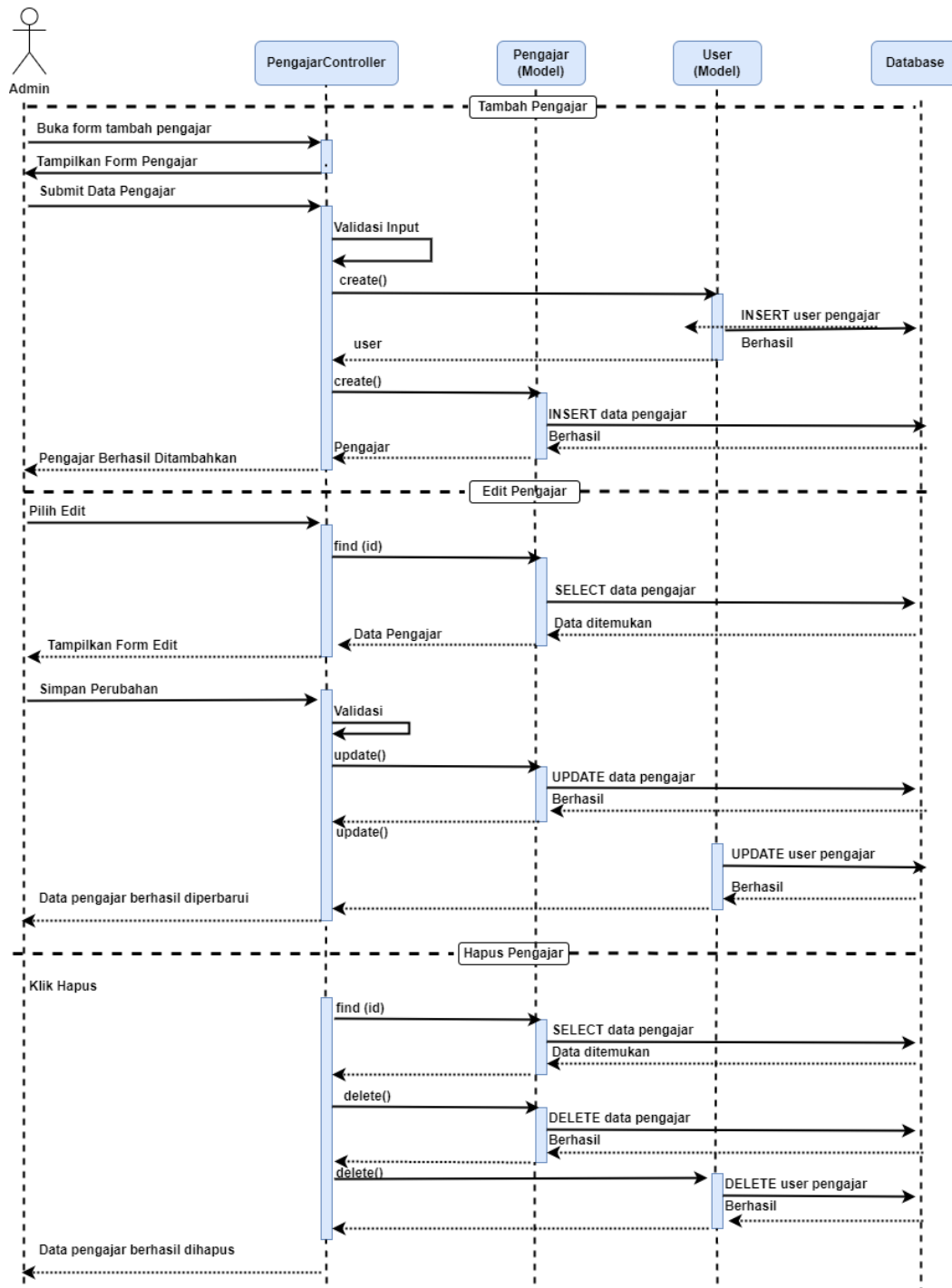
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.23 diatas, pada proses penambahan siswa, *SiswaController* terlebih dahulu mengambil data kelas dari *Database* dan men-*generate* NIS otomatis dengan format tahun berjalan ditambah tiga digit urutan. Setelah Admin mengisi dan mengirimkan *form*, sistem membuat akun *Login* orang tua melalui model *User*, kemudian menyimpan data siswa melalui model *Siswa* secara berurutan. Pada proses edit, sistem mengambil data siswa beserta akun orang tua yang terkait, menampilkan *form*, lalu memperbarui kedua data setelah Admin menyimpan perubahan. Pada proses hapus, sistem menghapus akun orang tua melalui model *User* terlebih dahulu, kemudian menghapus data siswa dari *Database*.

c. Sequence Diagram Kelola Data Pengajar

Sequence Diagram kelola data Pengajar menggambarkan interaksi antar objek pada saat Admin mengelola data pengajar beserta akun *Login* yang terintegrasi. Berikut adalah *Sequence* Diagram proses kelola data pengajar.

Sequence Diagram - Kelola Data Pengajar
Sistem Informasi Akademik Ruang Belajar Sa'idah



Gambar 3. 24 Sequence Diagram Kelola Data Pengajar

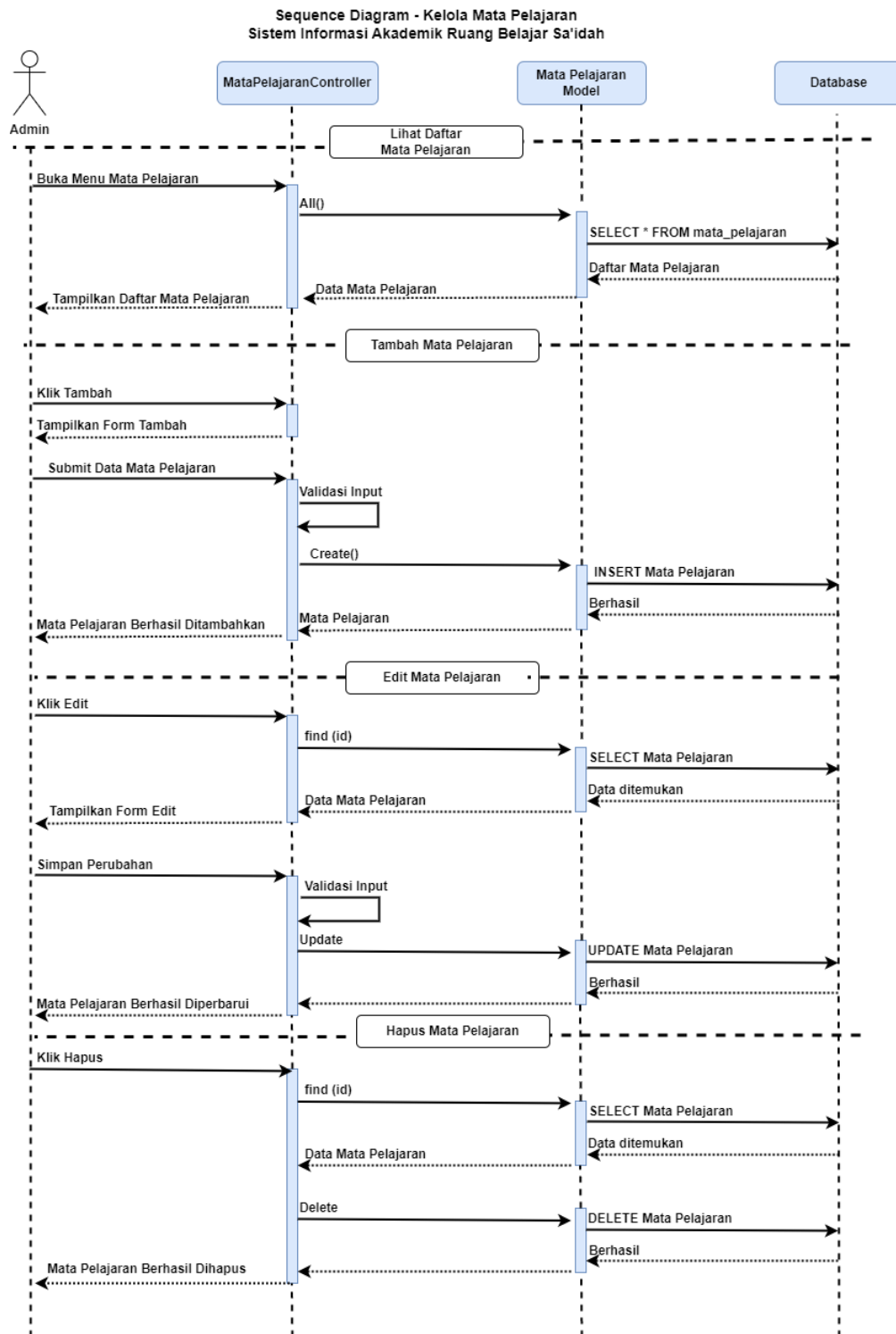
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.24 diatas, pada proses penambahan pengajar, Admin mengisi *form* data pengajar beserta *username* dan *Password* yang akan digunakan untuk *Login*. Sistem membuat akun *Login* melalui model *User* dengan *role* pengajar, kemudian menyimpan data pengajar melalui model *Pengajar* yang terhubung ke akun tersebut. Pada proses edit, sistem mengambil data pengajar beserta

akun *Login* terkait, lalu memperbarui keduanya setelah Admin menyimpan perubahan. Pada proses hapus, sistem menghapus data pengajar dan akun *Login* yang terkait secara bersamaan.

d. Sequence Diagram Kelola Mata Pelajaran

Sequence Diagram kelola mata pelajaran menggambarkan interaksi antar objek pada saat Admin mengelola data mata pelajaran yang tersedia pada sistem. Berikut adalah *Sequence Diagram* proses kelola mata pelajaran.



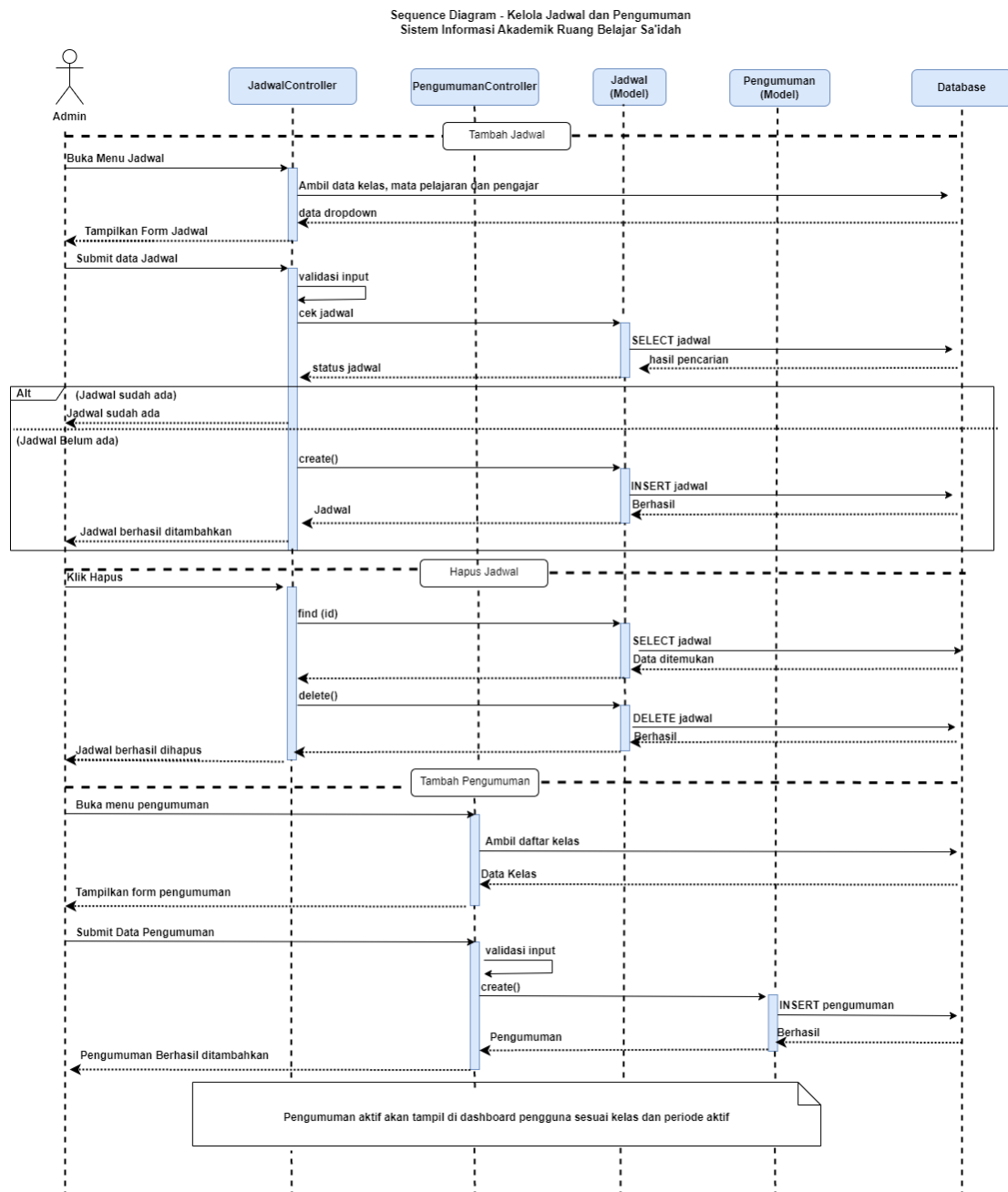
Gambar 3. 25 *Sequence* Diagram Kelola Mata Pelajaran

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.25 diatas, Admin membuka menu mata pelajaran dan sistem mengambil seluruh data dari tabel mata_pelajaran melalui model MataPelajaran untuk ditampilkan. Pada proses penambahan, Admin mengisi nama dan kode mata pelajaran, sistem melakukan validasi keunikan data, kemudian menyimpan ke *Database*. Pada proses edit, sistem mengambil data mata pelajaran yang dipilih, menampilkan form edit, lalu memperbarui data setelah Admin menyimpan. Pada proses hapus, sistem menghapus data mata pelajaran dari *Database* dan menampilkan konfirmasi keberhasilan.

e. *Sequence* Diagram Kelola Jadwal dan Pengumuman

Sequence Diagram kelola jadwal dan pengumuman menggambarkan interaksi antar objek pada saat Admin mengelola jadwal mengajar dan pengumuman dalam satu menu terintegrasi. Berikut adalah *Sequence* Diagram proses kelola jadwal dan pengumuman.



Gambar 3. 26 Sequence Diagram Kelola Jadwal dan Pengumuman

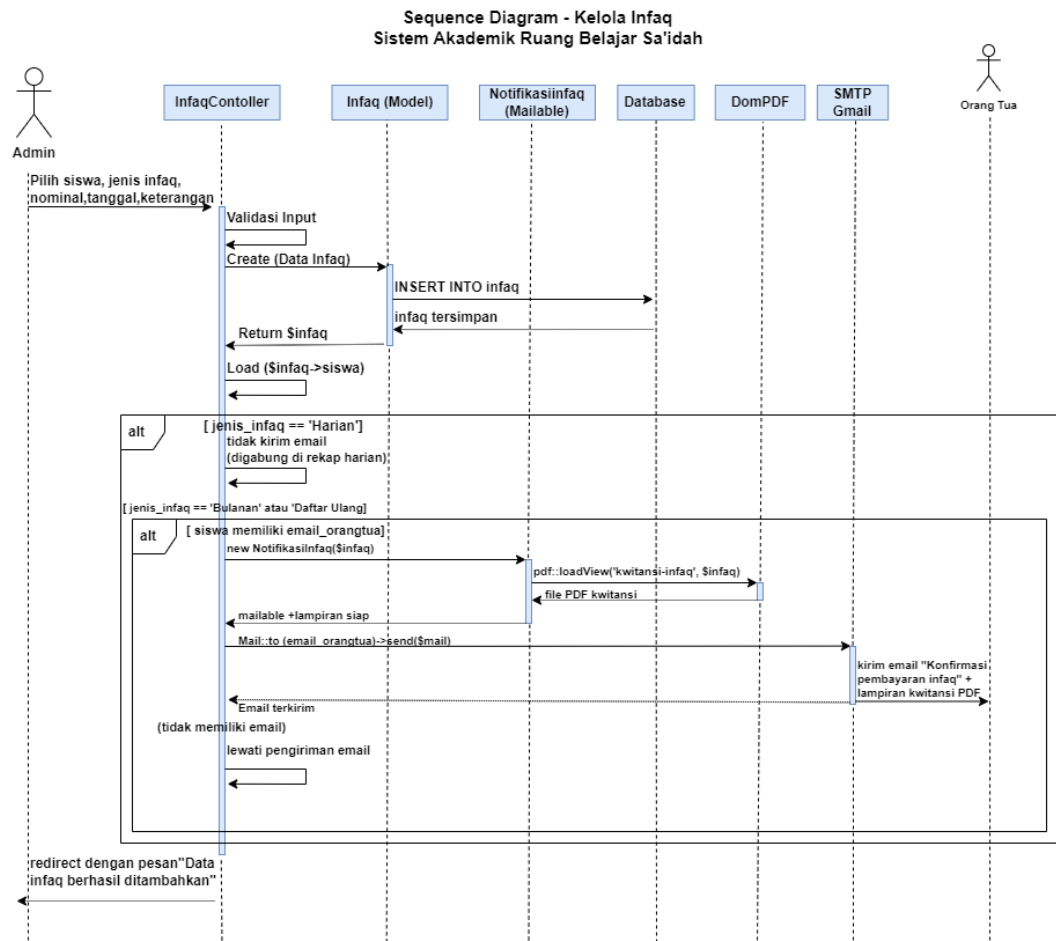
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.26 diatas, pada proses penambahan jadwal, *JadwalController* mengambil data kelas, mata pelajaran, dan pengajar dari *Database* untuk ditampilkan sebagai pilihan. Setelah Admin mengisi dan mengirimkan form, sistem memeriksa terlebih dahulu apakah terdapat jadwal yang identik pada *Database*. Apabila ditemukan duplikat, sistem menampilkan pesan kesalahan. Apabila tidak ada duplikat, sistem menyimpan jadwal baru dan jadwal tersebut otomatis tampil di *Dashboard* Pengajar maupun Orang Tua. Pada proes penambahan pengumuman, *PengumumanController* menyimpan data pengumuman beserta target kelas dan periode aktifnya,

sehingga pengumuman secara otomatis tampil di *Dashboard* seluruh pengguna yang relevan selama periode tersebut berlangsung.

f. Sequence Diagram Kelola Infaq

Sequence Diagram kelola infaq menggambarkan interaksi antar objek pada saat Admin mencatat pembayaran infaq siswa, termasuk perbedaan alur pengiriman notifikasi berdasarkan jenis infaq yang dicatat. Berikut adalah Sequence Diagram proses kelola infaq.



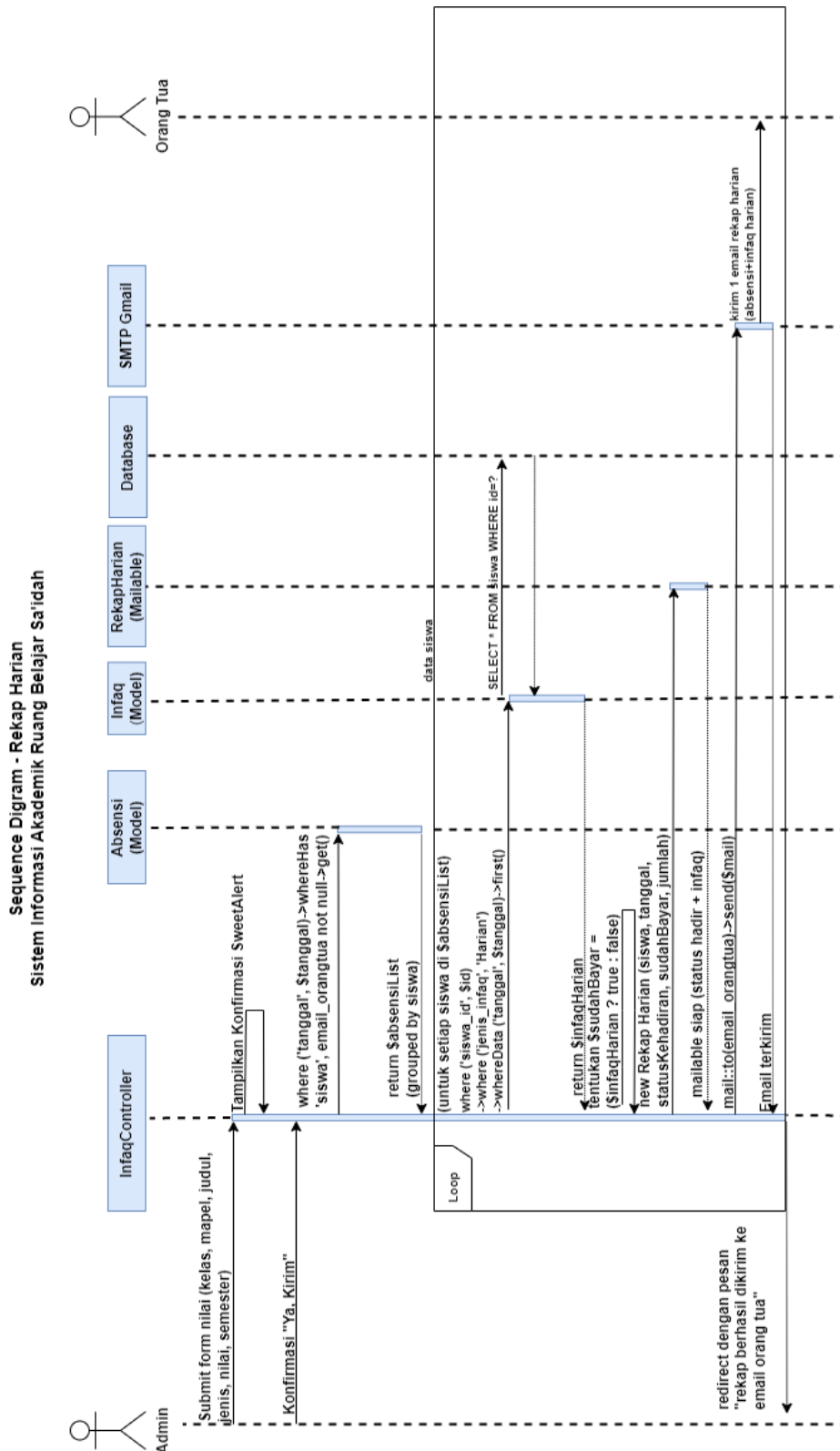
Gambar 3. 27 Sequence Diagram Kelola Infaq

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.27 diatas, Admin mengirimkan data infaq melalui *InfaqController* yang kemudian menyimpannya ke *Database* melalui model *Infaq*. Setelah data tersimpan, sistem memeriksa jenis infaq yang dicatat. Apabila berjenis *Harian*, sistem tidak mengirimkan email secara langsung karena informasinya akan digabungkan ke dalam rekap harian. Apabila berjenis *Bulanan* atau *Daftar Ulang*, sistem membuat objek *NotifikasiInfaq*, menghasilkan *file* PDF kwitansi menggunakan *DomPDF*, kemudian mengirimkan email beserta lampiran kwitansi tersebut melalui *SMTP Gmail* ke alamat *email_orangtua* siswa yang bersangkutan.

g. Sequence Diagram Kirim Rekap Harian

Sequence Diagram kirim rekap harian menggambarkan interaksi antar objek pada saat Admin mengirimkan email rekap gabungan kehadiran dan status infaq harian kepada orang tua siswa. Berikut adalah Sequence Diagram proses kirim rekap harian.



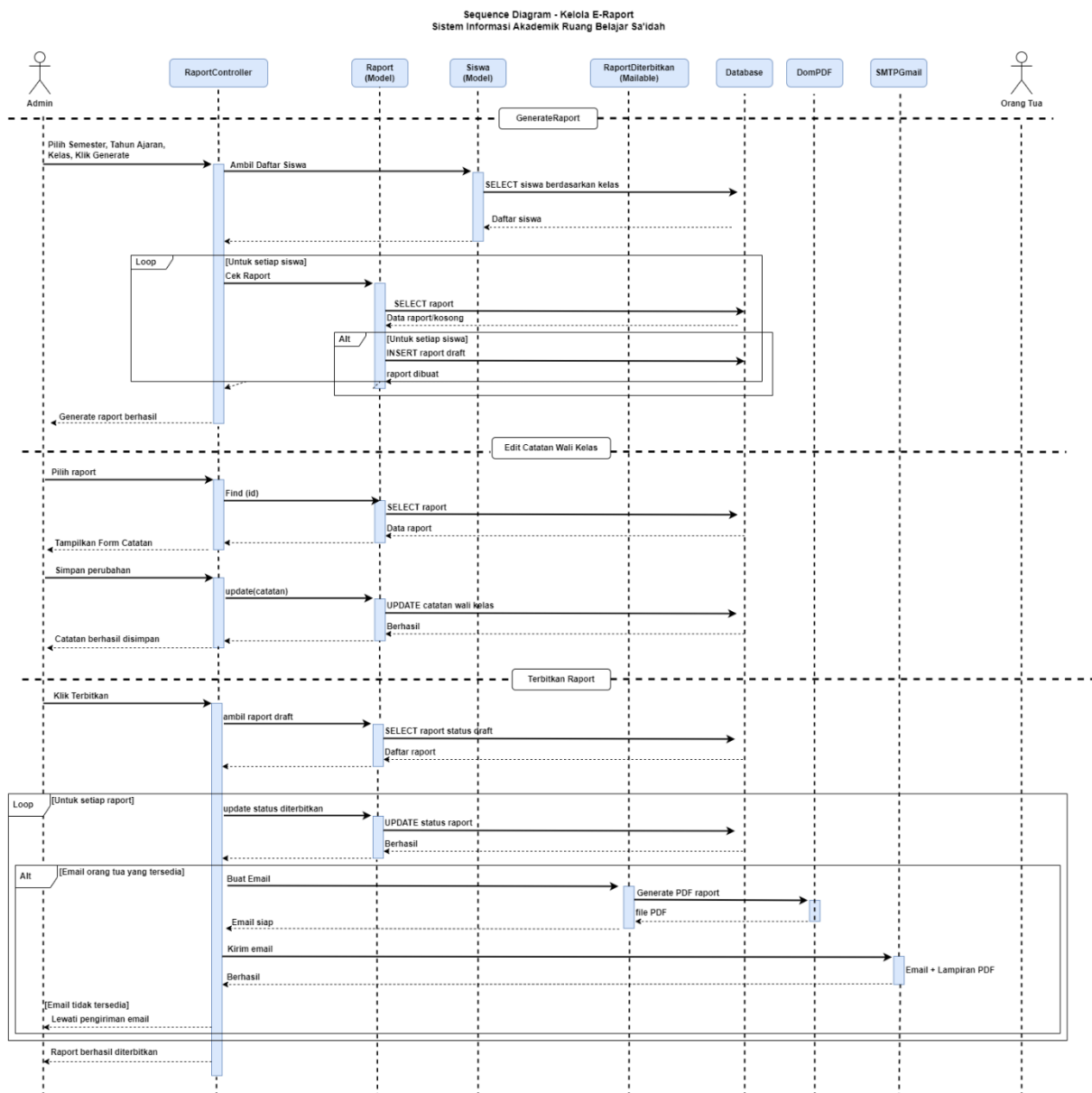
Gambar 3. 28 Sequence Diagram Rekap Harian

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.28 diatas, Admin memilih tanggal dan kelas kemudian mengkonfirmasi pengiriman melalui *InfaqController*. Sistem mengambil seluruh data absensi pada tanggal tersebut melalui model Absensi. Untuk setiap siswa yang memiliki data absensi, sistem memeriksa status infaq harian melalui model Infaq pada tanggal yang sama. Sistem kemudian membuat objek RekapHarian yang berisi gabungan status kehadiran dan status infaq harian, lalu mengirimkannya melalui STMP *Gmail* ke alamat email_orangtua masing-masing siswa. Apabila siswa hadir namun belum tercatat membayar infaq harian, email rekap juga menyertakan pengingat pembayaran.

h. Sequence Diagram Kelola E-raport

Sequence Diagram kelola *E-raport* menggambarkan interaksi antar objek pada saat Admin melakukan *generate*, penerbitan raport, serta pengiriman notifikasi email kepada orang tua siswa. Berikut adalah Sequence Diagram proses kelola *E-raport*.



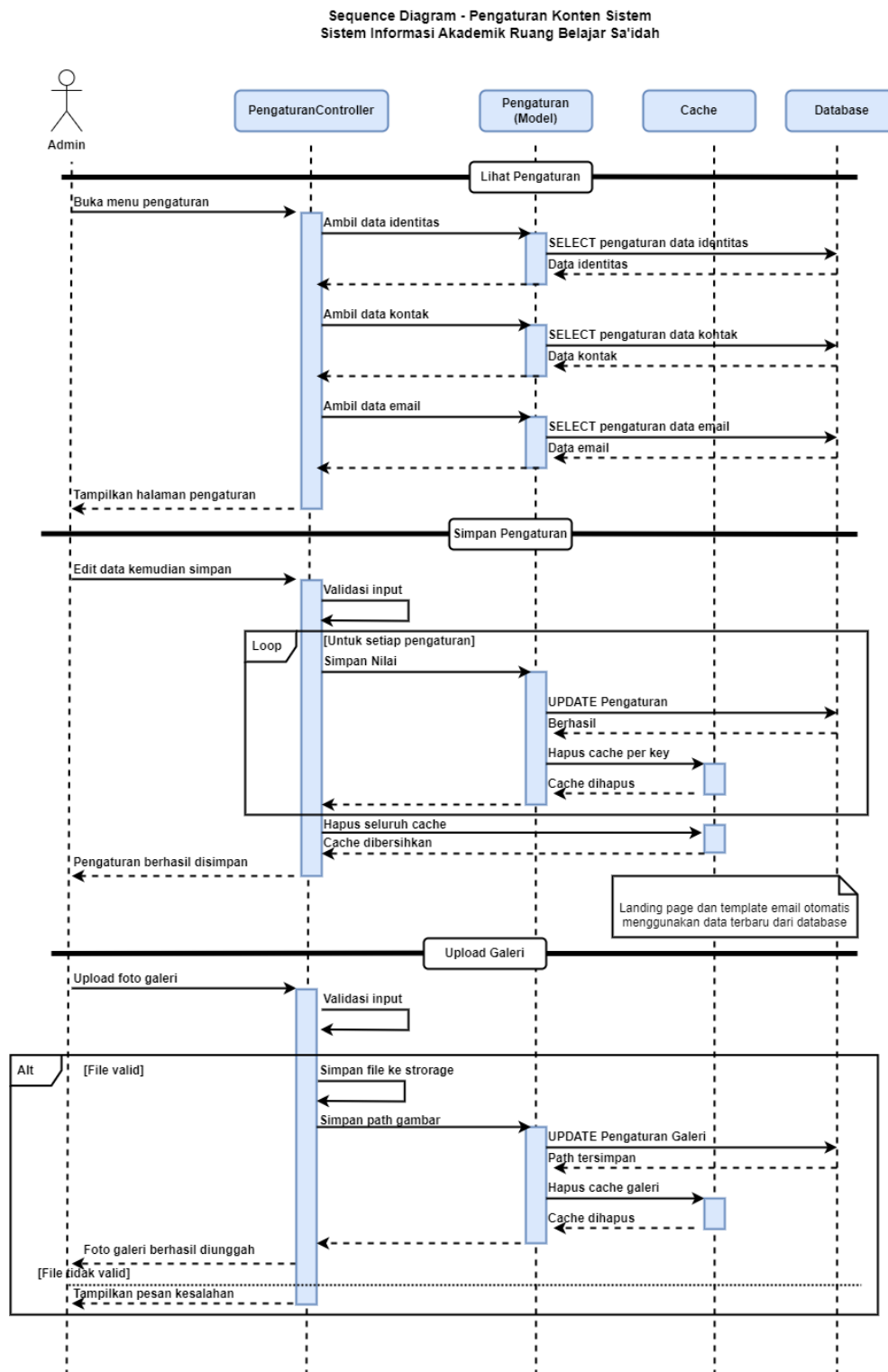
Gambar 3. 29 Sequence Diagram Kelola *E-raport*

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.29 diatas, pada proses *generate*, *RaportController* mengambil daftar siswa berdasarkan kelas dipilih, kemudian memeriksa setiap siswa apakah sudah memiliki data raport berstatus *Draft*. Pada proses penerbitan, sistem mengambil seluruh raport berstatus *Draft*, memperbarui statusnya menjadi *Diterbitkan*, kemudian untuk setiap siswa yang orang tuaya memiliki email_orangtua, sistem membuat objek *RaportDiterbitkan*, menghasilkan *file* PDF raport menggunakan *DomPDF*, dan mengirimkan email beserta lampiran PDF tersebut melalui SMTP *Gmail* kepada orang tua siswa.

i. Sequence Diagram Pengaturan Konten Sistem

Sequence Diagram pengaturan konten sistem menggambarkan interaksi antar objek pada saat Admin mengelola konten *Landing Page* dan *Template* email melalui menu Pengaturan. Berikut adalah *Sequence* Diagram proses pengaturan konten sistem.



Gambar 3. 30 Sequence Diagram Pengaturan Konten Sistem

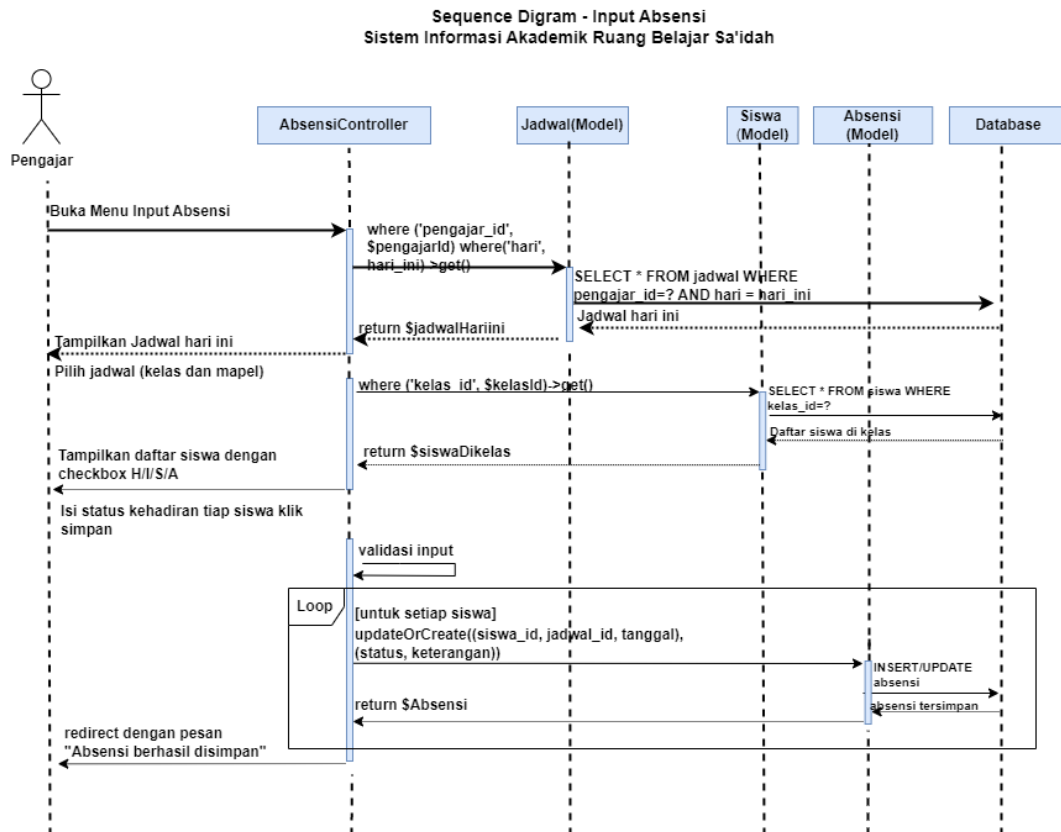
Sumber : Hasil perancangan Penulis (2026)

Berdasarkan Gambar 3.30 diatas, Admin membuka menu Pengaturan dan *PengaturanController* mengambil data dari tabel pengaturan yang dikelompokkan berdasarkan grup yaitu identitas, kontak, dan email. Pada proses penyimpanan konten, sistem melakukan perulangan untuk setiap *key* yang disubmit, memperbarui nilai pada tabel pengaturan melalui model Pengaturan menggunakan

metode *set()*, dan menghapus *cache* terkait agar perubahan langsung berlaku. Pada proses upload galeri, sistem memvalidasi *file*, menyimpannya ke direktori penyimpanan, kemudian memperbarui nilai *key* galeri pada tabel pengaturan dan menghapus *cache*-nya agar foto baru langsung tampil pada *Landing Page*.

j. Sequence Diagram Input Absensi

Sequence Diagram Input absensi menggambarkan interaksi antar objek pada saat Pengajar mencatat kehadiran siswa berdasarkan jadwal mengajar hari ini. Berikut adalah *Sequence Diagram* proses *Input* absensi.



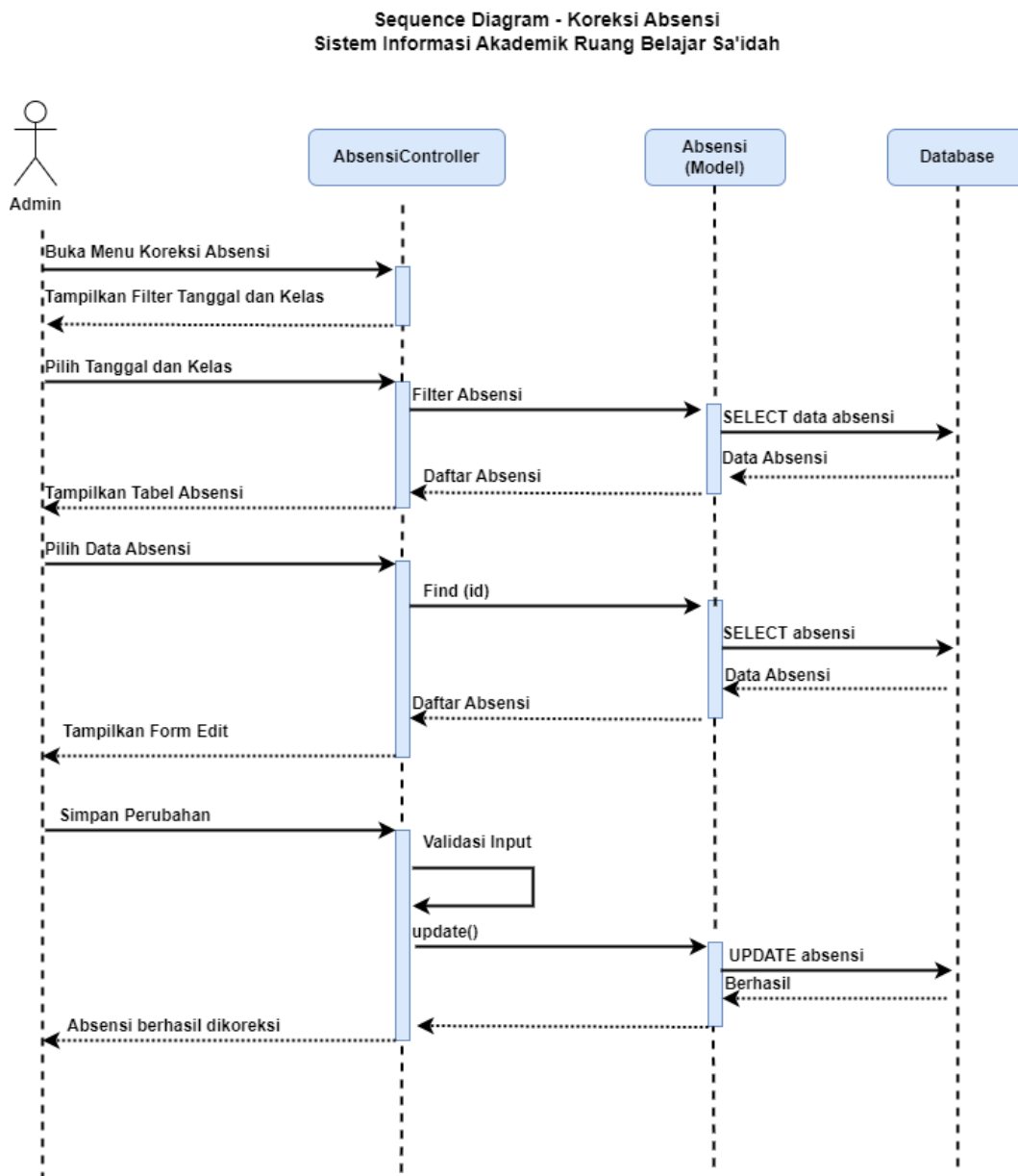
Gambar 3. 31 *Sequence Diagram Input Absensi*

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.31 diatas, Pengajar membuka menu *Input Absensi* dan *AbsensiController* mengambil jadwal hari ini milik Pengajar tersebut melalui model *Jadwal*. Setelah Pengajar memilih jadwal, sistem mengambil daftar siswa pada kelas yang bersangkutan melalui model *Siswa*. Pengajar mengisi status kehadiran setiap siswa kemudian menekan simpan. Sistem melakukan perulangan untuk setiap siswa, menyimpan atau memperbarui data absensi menggunakan metode *updateOrCreate* pada model *Absensi* sehingga absensi yang sama tidak tersimpan duplikat apabila Pengajar menyimpan ulang pada hari yang sama.

k. Sequence Diagram Koreksi Absensi

Sequence Diagram koreksi absensi menggambarkan interaksi antar objek pada saat Admin melakukan perbaikan data kehadiran siswa yang telah di*Input* oleh Pengajar. Berikut adalah *Sequence* Diagram proses koreksi absensi.



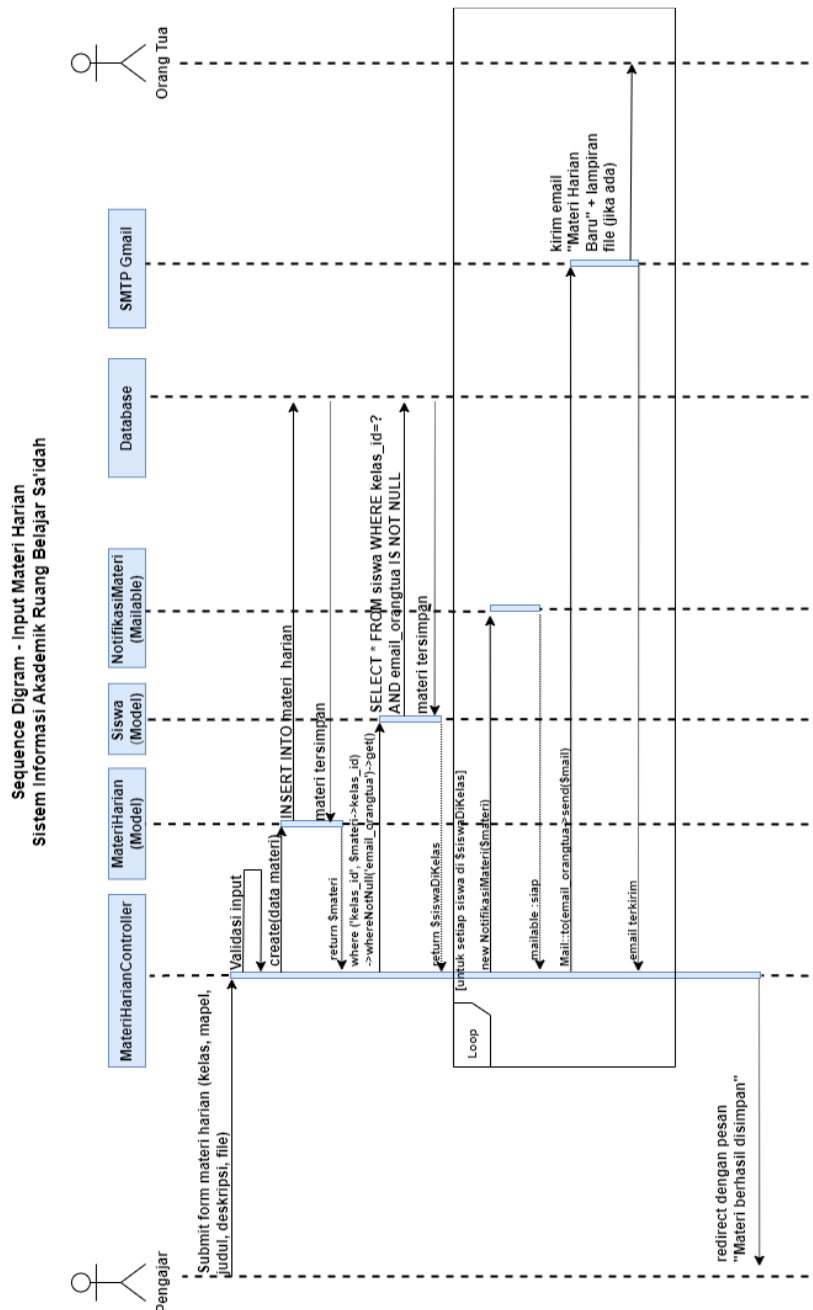
Gambar 3. 32 *Sequence* Diagram Koreksi Absensi

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.32 diatas, Admin membuka menu Koreksi Absensi dan memilih filter tanggal serta kelas. *AbsensiController* mengambil data absensi dari *Database* dengan melakukan *join* ke tabel jadwal dan siswa sesuai filter yang dipilih. Admin memilih data yang akan dikoreksi dan sistem mengambil Detail data absensi tersebut untuk ditampilkan pada form edit. Setelah Admin mengubah status atau keterangan dan menyimpan perubahan, sistem memperbarui data pada tabel absensi di *Database* dan menampilkan konfirmasi keberhasilan koreksi.

1. *Sequence* Diagram *Input* Materi Harian

Sequence Diagram Input materi harian menggambarkan interaksi antar objek mulai dari penyimpanan materi hingga pengiriman notifikasi email kepada orang tua seluruh siswa di kelas tersebut. Berikut adalah *Sequence Diagram* proses *Input* materi harian.



Gambar 3. 33 *Sequence Diagram Input* Materi Harian

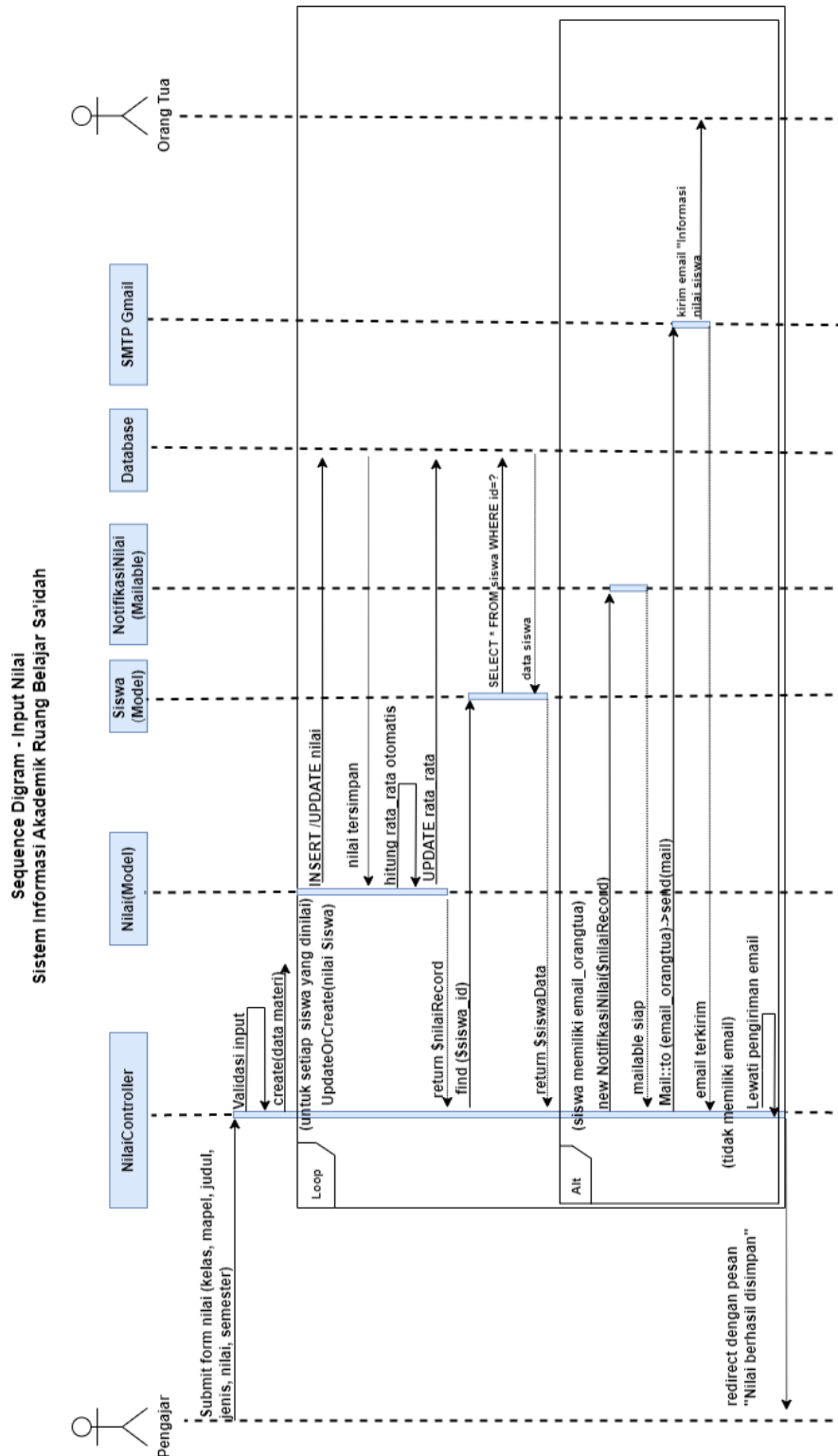
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.33 diatas, Pengajar mengirimkan data materi melalui *MateriHarianController* yang menyimpan data beserta file lampiran ke *Database* melalui model *MateriHarian*. Sistem kemudian mengambil daftar siswa pada kelas tersebut yang memiliki *email_orangtua* terdaftar melalui model *Siswa*. Sistem melakukan perulangan untuk setiap siswa,

membuat objek *NotifikasiMateri*, dan mengirimkannya beserta lampiran file melalui *SMTP Gmail* kepada orang tua masing-masing siswa.

m. Sequence Diagram *Input Nilai*

Sequence Diagram Input nilai menggambarkan interaksi antar objek mulai dari penyimpanan nilai, perhitungan rata-rata otomatis, hingga pengiriman notifikasi email kepada orang tua. Berikut adalah *Sequence Diagram* proses *Input* nilai.



Gambar 3. 34 *Sequence Diagram Input Nilai*

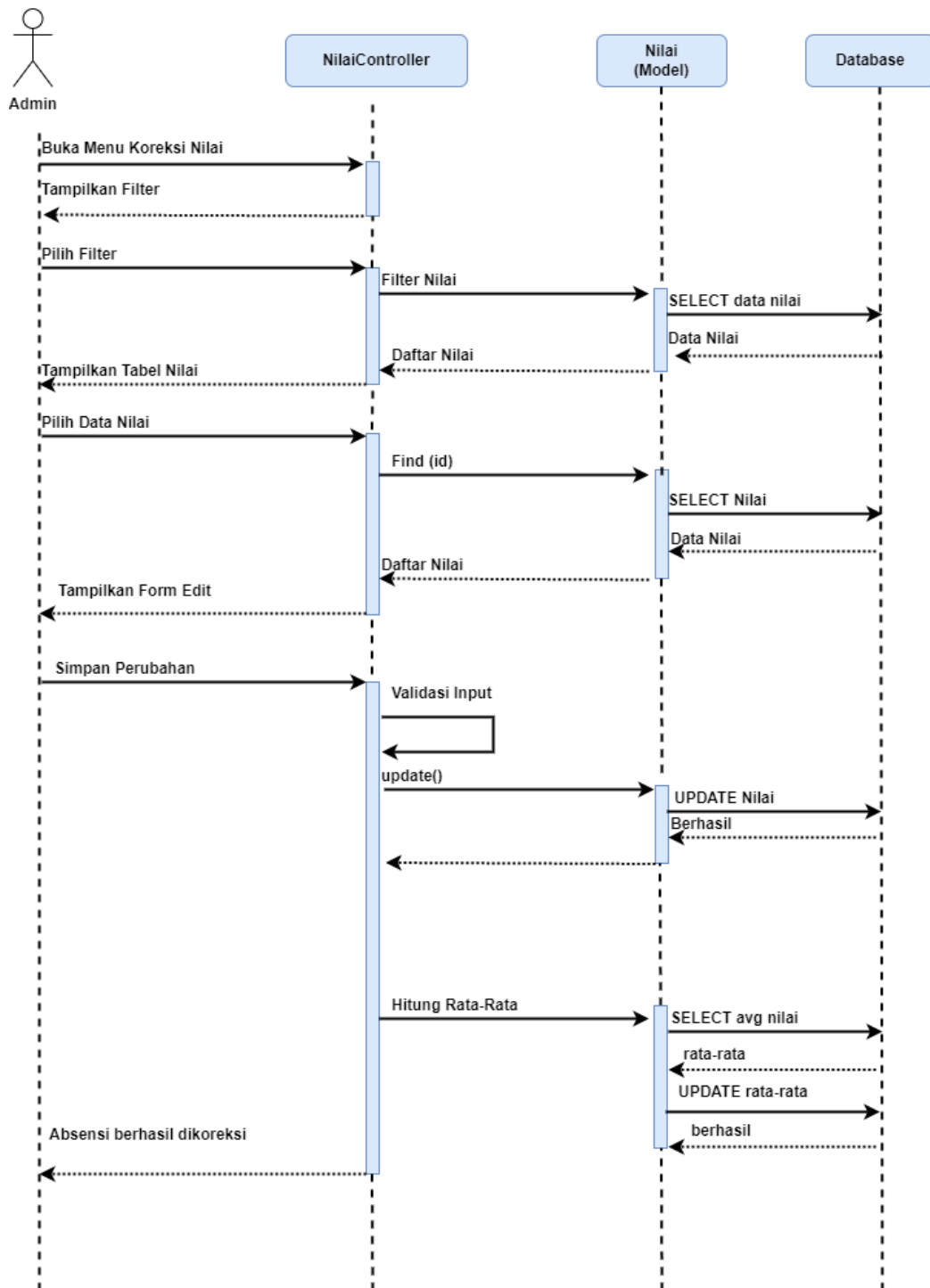
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.34 diatas, Pengajar mengirimkan data materi melalui *MateriHarianController* yang menyimpan data beserta *file* lampiran ke *Database* melalui model *MateriHarian*. Sistem kemudian mengambil daftar siswa pada kelas tersebut yang memiliki email_orangtua terdaftar melalui model *Siswa*. Sistem melakukan perulangan untuk setiap siswa, membuat objek *NotifikasiMateri*, dan mengirimkannya beserta lampiran file melalui *SMTP Gmail* kepada orang tua masing-masing siswa.

n. *Sequence Diagram Koreksi Nilai*

Sequence Diagram koreksi nilai menggambarkan interaksi antar objek pada saat Admin melakukan perbaikan data nilai siswa beserta pembaruan rata-rata secara otomatis. Berikut adalah *Sequence Diagram* proses koreksi nilai.

Sequence Diagram - Koreksi Nilai
Sistem Informasi Akademik Ruang Belajar Sa'idah



Gambar 3. 35 Sequence Diagram Koreksi Nilai

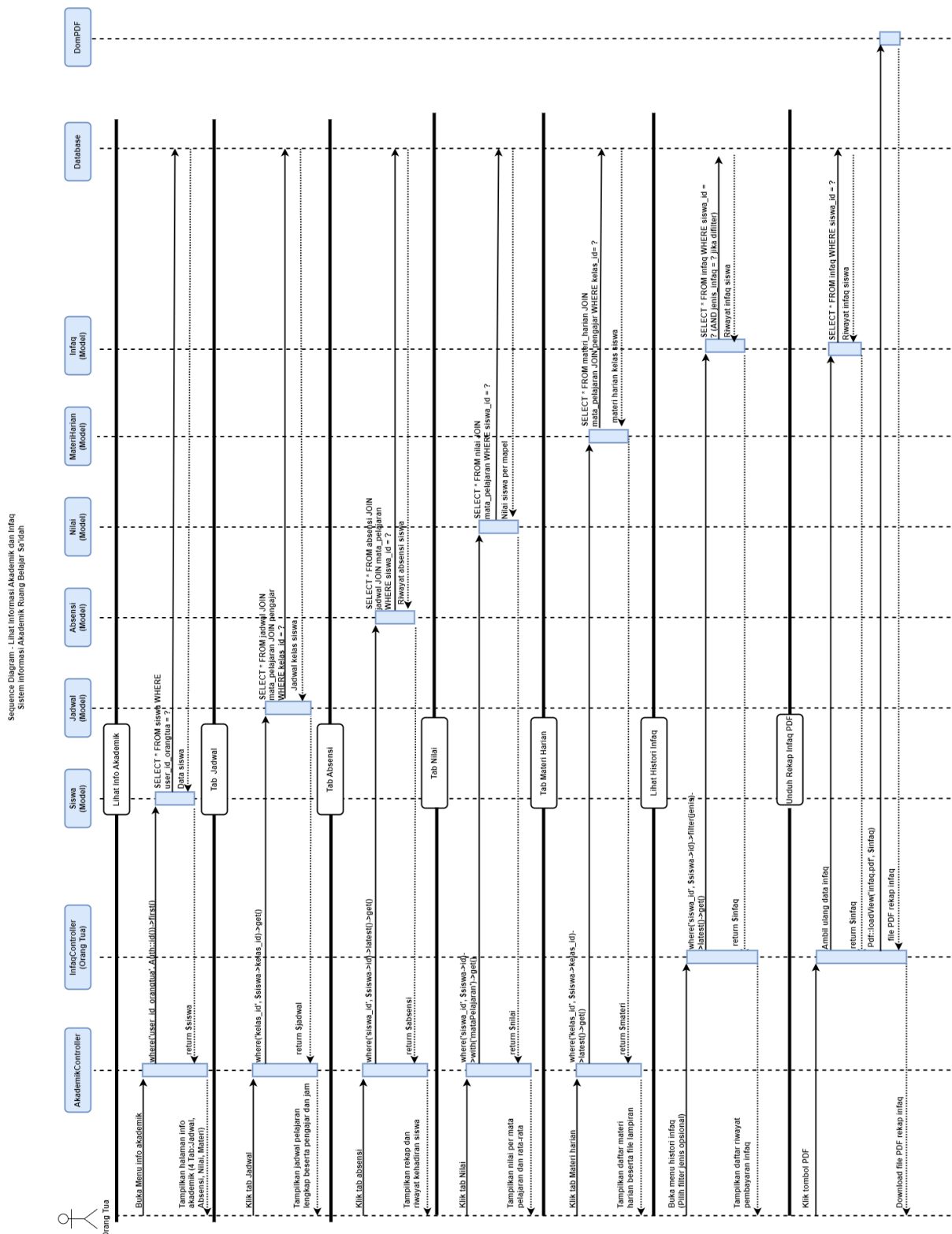
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.35 diatas, Admin membuka menu Koreksi Nilai dan memilih filter pencarian berdasarkan kelas, mata pelajaran, jenis nilai, semester, dan tahun ajaran. *NilaiController* mengambil data nilai dari *Database* dengan melakukan *join* ke tabel siswa dan mata_pelajaran. Admin memilih nilai yang akan dikoreksi dan sistem menampilkan form edit. Setelah Admin

mengubah nilai dan menyimpan perubahan, sistem memperbarui data di tabel nilai, kemudian menghitung ulang rata-rata nilai siswa tersebut untuk mata pelajaran, semester, dan tahun ajaran yang sama secara otomatis, dan memperbarui kolom rata-rata di *Database*.

o. Sequence Diagram Lihat Informasi Akademik dan Infaq

Sequence Diagram lihat informasi akademik dan histori infaq menggambarkan interaksi antar objek pada saat Orang Tua mengakses informasi perkembangan akademik dan riwayat pembayaran infaq anaknya. Berikut adalah *Sequence* Diagram proses lihat informasi akademik dan histori infaq.



Gambar 3. 36 Sequence Diagram Lihat Informasi Akademik dan Infaq

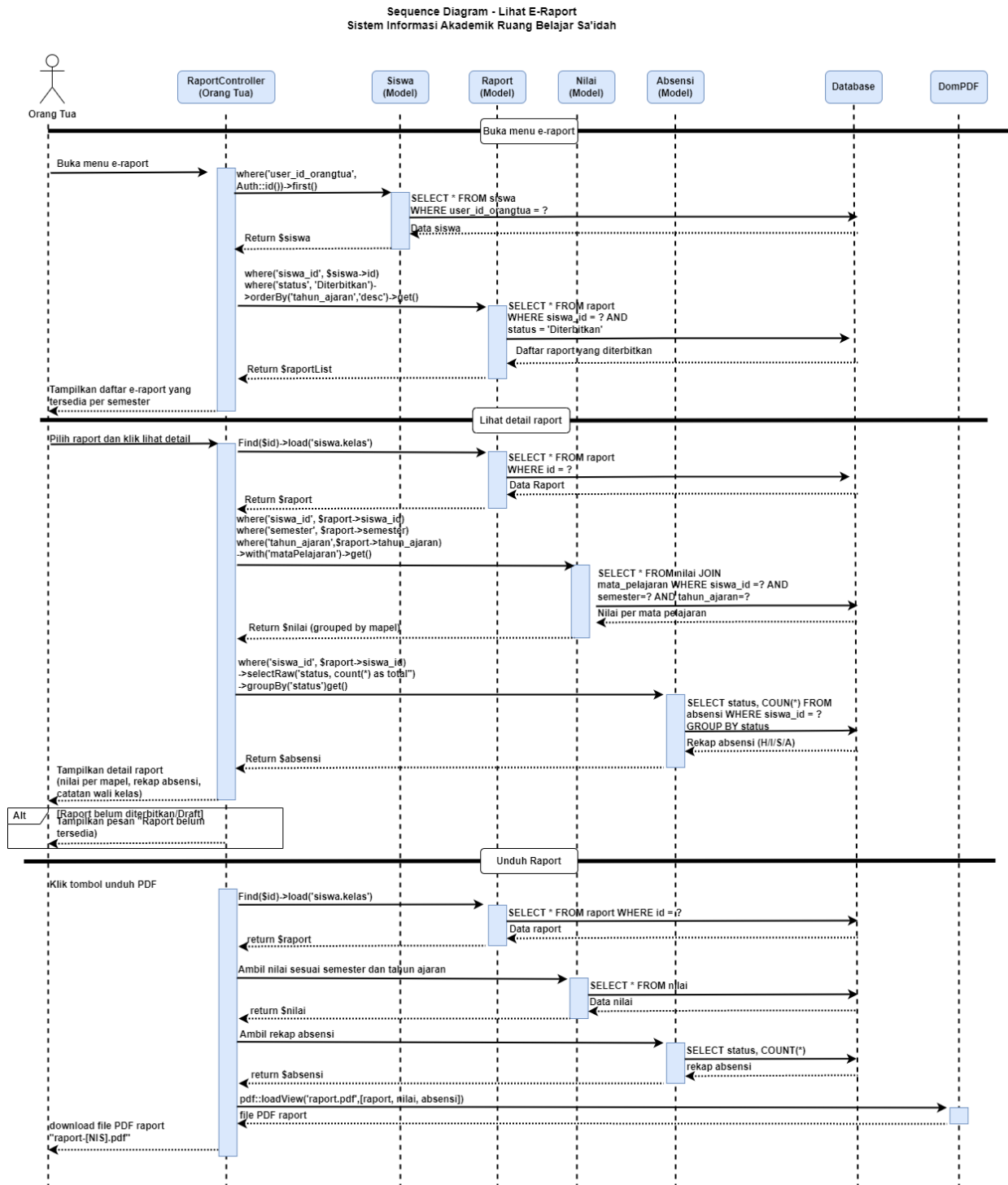
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.36 diatas, Orang Tua membuka menu info Akademik melalui *AkademikController* yang mengambil data siswa milih orang tua tersebut dari *Database*. Sistem kemudian mengambil dan menampilkan data sesuai tab yang dipilih, yaitu jadwal dari tabel jadwal, absensi dari tabel absensi, nilai beserta rata-ratanya dari tabel nilai, dan materi harian beserta *file*

lampirannya dari tabel materi_harian. Pada menu Histori Infaq, *InfaqController* mengambil seluruh riwayat siswa dari tabel Infaq berdasarkan filter jenis yang dipilih, dan Orang Tua dapat mengunduh rekap infaq dalam format PDF yang dihasilkan oleh *DomPDF*.

p. Sequence Diagram Lihat E-raport

Sequence Diagram lihat *E-raport* menggambarkan interaksi antar objek pada saat Orang Tua mengakses dan mengunduh raport elektronik anaknya melalui sistem. Berikut adalah *Sequence* Diagram proses lihat *E-raport*.



Gambar 3. 37 Sequence Diagram Lihat E-raport

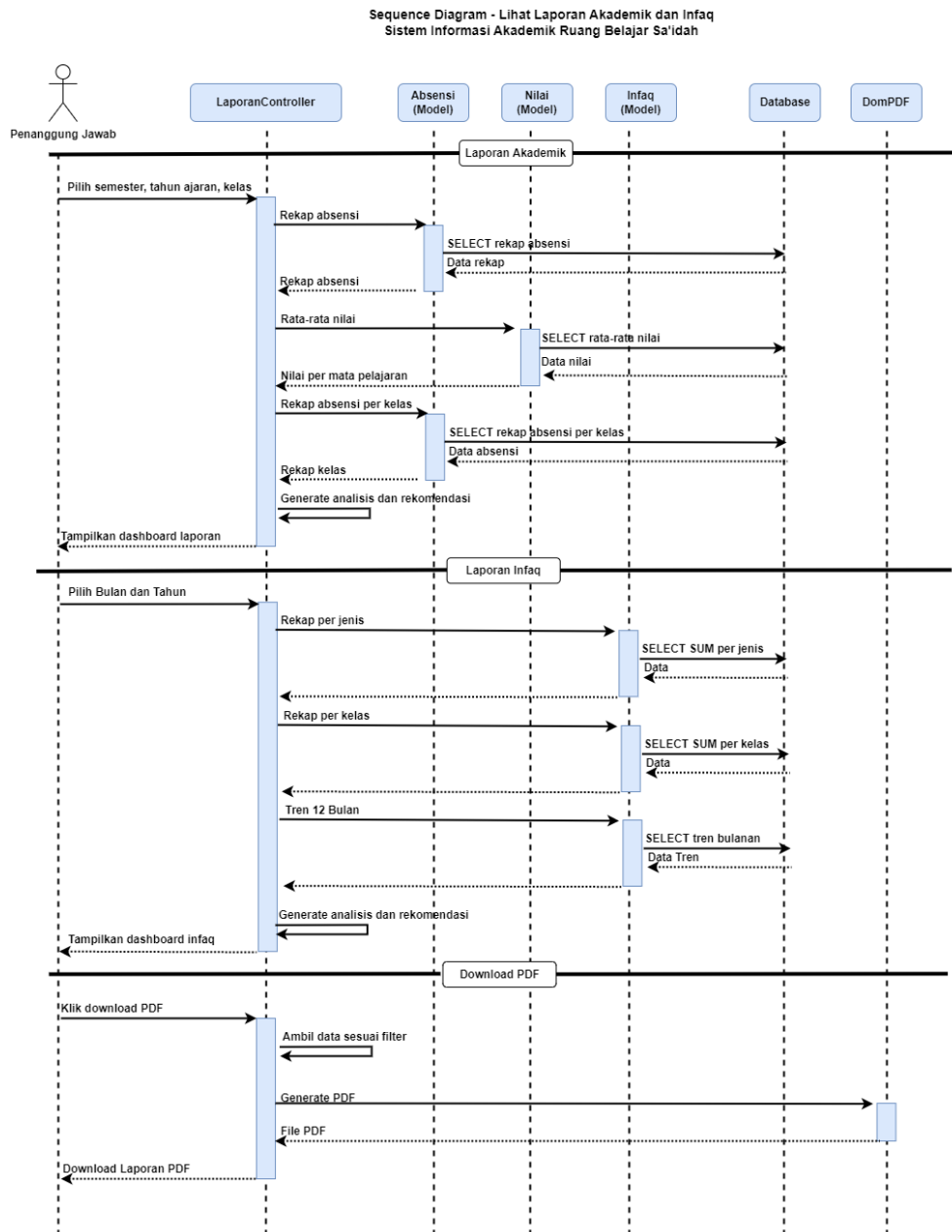
Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.37 diatas, Orang Tua membuka menu *E-raport* dan *RaportController* mengambil daftar raport yang berstatus Diterbitkan milik siswa terkait dari tabel raport. Orang Tua memilih raport yang ingin dilihat dan sistem mengambil Detail data raport beserta nilai per mata pelajaran dari tabel nilai dan rekap absensi dari tabel absensi. Apabila Orang Tua memilih untuk

mengunduh, sistem menghasilkan *file* PDF raport menggunakan *DomPDF* berdasarkan data yang telah diambil, kemudian mengirimkan *file* tersebut sebagai unduhan ke perangkat Orang Tua.

q. Sequence Diagram Lihat Laporan Akademik dan Infaq

Sequence Diagram lihat laporan akademik dan infaq menggambarkan interaksi antar objek pada saat Penanggung Jawab mengakses laporan beserta visualisasi diagram dan analisis otomatis yang tersedia pada sistem. Berikut adalah *Sequence Diagram* proses lihat laporan akademik dan infaq.



Gambar 3. 38 Sequence Diagram Lihat Laporan Akademik dan Infaq

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.38 diatas, pada laporan akademik, Penanggung Jawab memilih filter dan *LaporanController* mengambil data rekap absensi per status dari model Absensi, nilai rata-rata per mata pelajaran dari model Nilai, dan absensi per kelas dari *Database* secara berurutan. Sistem kemudian menghasilkan analisis kondisi dan rekomendasi tindakan secara otomatis berdasarkan data yang

3.5.4 Class Diagram

```

classDiagram
    class User {
        id: int
        name: string
        email: string
        password: string
        foto: string
        role: string
        isActive: boolean
        login(): boolean
        logout(): void
        resetPassword(email: string): boolean
        updateProfile(name: string, foto: string): void
    }
    class Pengaturan {
        id: int
        key: string
        value: string
        label: string
        get(key: string): string
        set(key: string, value: string): void
    }
    class Pengajar {
        id: int
        userId: int
        nama: string
        nik: string
        nohp: string
        alamat: string
        InputNilai(siswaid: int, skor: float, jenis: string): void
        InputAbsensi(siswaid: int, jadwaid: int, status: string)
        UploadMateri(kelasid: int, judul: string, file: string)
        viewJadwal(): List<Jadwal>
    }
    class MataPelajaran {
        id: int
        kodeMapel: string
        namaMapel: string
        kategori: string
        getJadwalTerbuka(): List<Jadwal>
    }
    class Kelas {
        id: int
        namaKelas: string
        tingkat: string
        getDaftarSiswa(): List<Siswa>
        getJadwalPelajaran(): List<Jadwal>
    }
    class MateriHarian {
        id: int
        pengajarid: int
        kelasid: int
        tanggal: date
        judul: string
        fileLampiran: string
        downloadFile(): string
    }
    class Jadwal {
        id: int
        kelasid: int
        mataPelajaranid: int
        pengajarid: int
        hari: string
        jamMulai: time
        jamSelesai: time
        broadcast(): void
    }
    class Siswa {
        id: int
        userOrangtua: int
        kelasid: int
        nama: string
        nis: string
        tanggalLahir: date
        jenisKelamin: string
        emailOrangtua: string
        viewRapor(): List<Rapor>
        viewAbsensi(): List<Absensi>
        viewNilai(): List<Nilai>
    }
    class Pengumuman {
        id: int
        judul: string
        isi: string
        jenis: string
        status: string
        tanggalMulai: date
        broadcast(): void
    }
    class Absensi {
        id: int
        siswaid: int
        jadwaid: int
        pengajarid: int
        tanggal: date
        status: string
        keterangan: string
        koreksiAbsensi(statusBaru: string): void
    }
    class Infq {
        id: int
        siswaid: int
        adminid: int
        jenisInfq: string
        tanggal: date
        jumlah: float
        broadcast(): void
    }
    class Nilai {
        id: int
        siswaid: int
        mataPelajaranid: int
        pengajarid: int
        jenisNilai: string
        nilai: float
        semester: int
        tahunAjaran: string
        hitungRataRata(): float
    }
    class Rapor {
        id: int
        siswaid: int
        semester: int
        tahunAjaran: string
        catatanWaliKelas: string
        status: string
        broadcast(): void
    }

    User "1" -- "0..*" Siswa : memiliki
    User "1" -- "0..*" Pengajar : memiliki
    Pengaturan "1" -- "0..*" Pengajar : memiliki
    Pengajar "1" -- "0..*" MataPelajaran : mengajarkan
    MataPelajaran "1" -- "0..*" Jadwal : menghasilkan
    Pengajar "1" -- "0..*" Jadwal : mengajarkan
    Kelas "1" -- "0..*" Siswa : mendaftarkan
    Kelas "1" -- "0..*" Jadwal : mendapatkan
    MateriHarian "0..*" Jadwal : ditunjukkan
    Jadwal "0..*" Absensi : mencatat kehadiran
    Jadwal "0..*" Infq : menghasilkan
    Siswa "0..*" Rapor : memperoleh
    Siswa "0..*" Nilai : memperoleh
    Pengumuman "0..*" Absensi : broadcast
    Absensi "0..*" Nilai : broadcast
    Infq "0..*" Nilai : broadcast
    Rapor "0..*" Nilai : broadcast
    
```

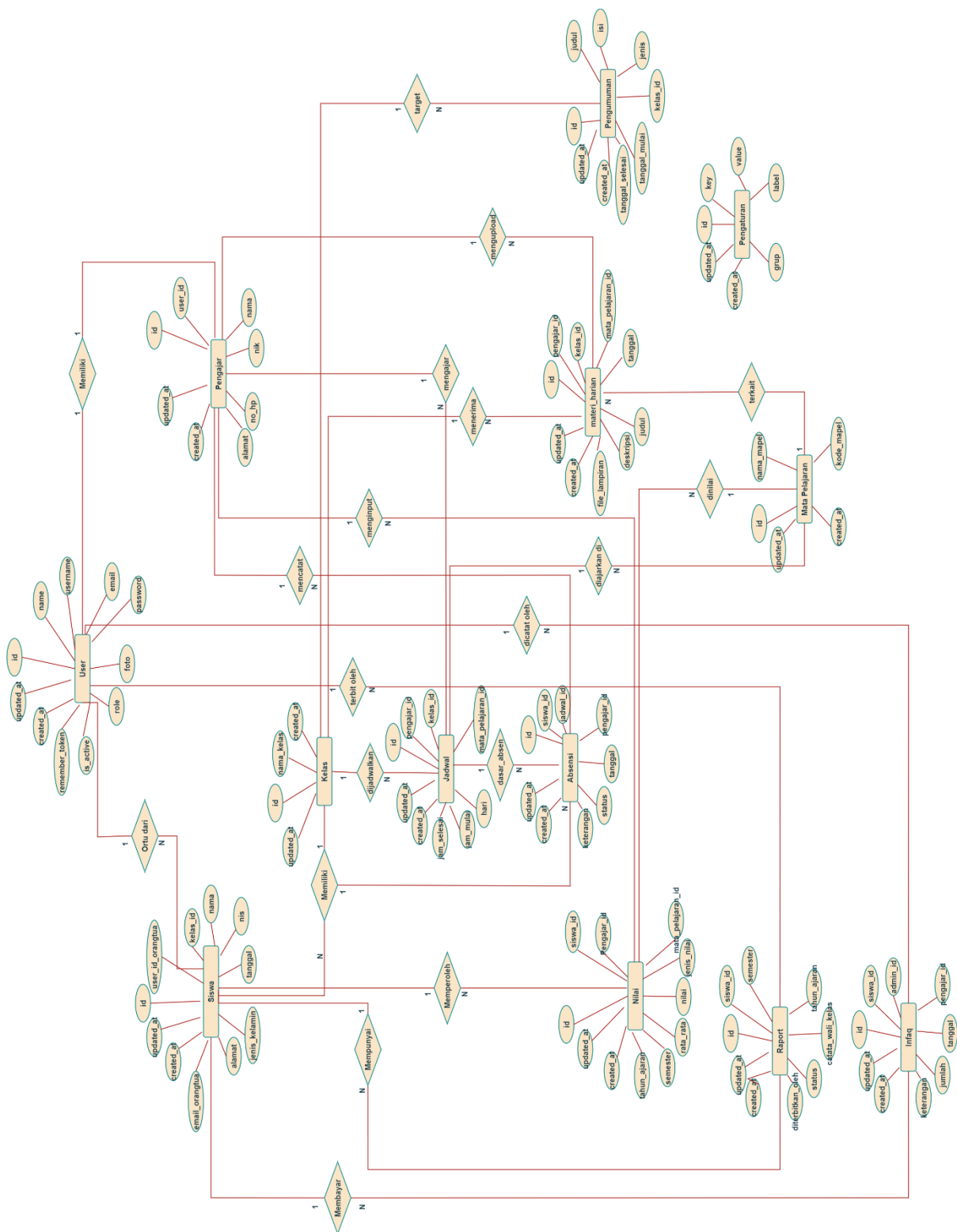
Sumber : Hasil Perancangan Penulis (2026)

53

serta berelasi dengan kelas Siswa melalui relasi *hasMany* sebagai orang tua. Kelas Siswa sebagai sentral berelasi dengan kelas Absensi, Nilai, Infaq, dan Raport melalui relasi *hasMany*. Kelas Jadwal menghubungkan kelas Pengajar, Kelas, dan MataPelajaran melalui relasi *belongsTo* pada masing-masing kelas tersebut, sekaligus berelasi dengan kelas Absensi melalui relasi *hasMany*. Kelas Pengaturan merupakan kelas independen yang memiliki dua *method* statis yaitu *get()* untuk mengambil nilai konten dari *Database* dengan mekanisme *cache*, dan *set()* untuk menyimpan nilai sekaligus menghapus *cache* terkait agar perubahan langsung berlaku. Lima kelas *Mailable* yaitu NotifikasiMateri, NotifikasiInfaq, RaportDiterbitkan, RekapHarian, dan NotifikasiNilai masing-masing bertanggung jawab menyusun dan mengirimkan email notifikasi sesuai kejadian yang memicunya, serta memanfaatkan kelas Pengaturan melalui *method get()* untuk mengambil teks salam, penutup, tanda tangan, dan isi email yang dapat diubah oleh Admin melalui menu Pengaturan.

3.5.5 Entity Relationship diagram

Entity Relationship Diagram (ERD) menggambarkan struktur basis data sistem informasi akademik Ruang Belajar Sa'idah beserta relasi antar entitas menggunakan *notasi Chen* yang terdiri dari persegi panjang sebagai entitas, *elips* sebagai atribut, dan belah ketupat sebagai relasi. Berikut adalah ERD dari sistem yang dibangun.



Gambar 3. 40 Entity Relationship Diagram SIKAD Ruang Belajar Sa'idah

Sumber : Hasil Perancangan Penulis (2026)

Berdasarkan Gambar 3.40 diatas, basis data sistem terdiri dari tiga belas entitas utama yang saling berelasi. Entitas *Users* menyimpan data akun seluruh pengguna sistem dan memiliki relasi satu ke satu dengan entitas pengajar, serta relasi satu ke banyak dengan entitas siswa dalam kapasitas sebagai orang tua. Entitas kelas memiliki relasi satu ke banyak dengan entitas siswa, jadwal, materi_harian, dan

pengumuman. Entitas siswa sebagai entitas sentral memiliki relasi satu ke banyak dengan entitas absensi, nilai, infaq, dan raport. Entitas mata_pelajaran berelasi dengan entitas jadwal, nilai, dan materi_harian. Entitas jadwal yang menghubungkan pengajar, kelas, dan mata_pelajaran memiliki relasi dengan entitas absensi. Entitas pengaturan berdiri sebagai entitas independen yang menyimpan konten diamis dalam bentuk pasangan *key-value* dan tidak memiliki relasi langsung dengan entitas lain, namun digunakan oleh seluruh fitur notifikasi email dan *Landing Page* untuk mengambil konten yang dapat dikelola oleh Admin.

3.5.6 Perancangan Basis Data

Perancangan basis data merupakan tahap untuk menerjemahkan pemodelan *entity relationship* digaram (ERD) ke dalam bentuk struktur tabel yang akan diterapkan pada sistem manajemen basis data menggunakan *MySQL*. Pada tahap ini ditentukan nama tabel, nama kolom, tipe data yang dibutuhkan sistem. Perancangan basis data yang baik menjadi fondasi utama dalam memastikan data tersimpan secara terstruktur, konsisten, dan dapat diakses dengan efisien oleh seluruh modul sistem.

Basis data sistem informasi akademik Ruang Belajar Sa'idah diberi nama ***siakad_rbs*** dan terdiri dari tiga belas tabel utama beserta dua tabel pendukung bawaan *Laravel*. Seluruh tabel dibuat secara otomatis melalui mekanisme migrasi yang disediakan oleh *framework Laravel*. Berikut adalah struktur rancangan masing-masing tabel pada basis data sistem.

a. Tabel Users

Menyimpan data akun pengguna sistem (Admin, Pengajar, Orang Tua, dan Penanggung Jawab) untuk kebutuhan autentikasi dan otorisasi.

Tabel 3. 6 Perancangan Basis Data Tabel *Users*

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik pengguna
2	name	varchar(100)	Nama lengkap pengguna
3	username	varchar(50)	Username untuk <i>Login</i>
4	email	varchar(100)	Alamat email unik
5	<i>Password</i>	varchar(255)	Hash Password <i>Login</i>
6	foto	varchar(255)	Nama file/path foto
7	role	enum('admin', 'pengajar', 'orangtua', 'penanggungjawab')	Hak akses/peran dalam sistem
8	is_active	boolean	Status keaktifan akun
9	remember_token	varchar(100)	Token untuk fitur remember me
10	created_at	timestamp	Waktu data dibuat
11	updated_at	timestamp	Waktu data terakhir diubah

b. Tabel Pengajar

Menyimpan profil lengkap data guru/pengajar yang berelasi langsung dengan akun pengguna (*Users*).

Tabel 3. 7 Perancangan Basis Data Tabel Pengajar

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik pengajar
2	user_id	bigint(20)	Relasi ke tabel pengguna
3	nama	varchar(100)	Nama lengkap pengajar
4	nik	varchar(20)	Nomor induk kependudukan
5	no_hp	varchar(15)	Nomor telepon aktif
6	alamat	text	Alamat tempat tinggal
7	created_at	timestamp	Waktu data dibuat
8	updated_at	timestamp	Waktu data terakhir diubah

c. Tabel Kelas

Menyimpan data kelas yang tersedia di lingkungan Ruang Belajar Sa'idah.

Tabel 3. 8 Perancangan Basis Data Tabel Kelas

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik kelas
2	nama_kelas	varchar(50)	Nama kelas
3	tingkat	varchar(20)	Tingkat pendidikan kelas
4	created_at	timestamp	Waktu data dibuat
5	updated_at	timestamp	Waktu data terakhir diubah

d. Tabel Siswa

Menyimpan informasi identitas siswa, kelas siswa tersebut, dan orang tua yang memantau perkembangannya.

Tabel 3. 9 Perancangan Basis Data Tabel Siswa

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik siswa
2	user_orangtua	bigint(20)	Id akun orang tua
3	kelas_id	bigint(20)	Id kelas siswa
4	nama	varchar(100)	Nama lengkap siswa
5	nis	varchar(20)	Nomor induk siswa
6	tanggal_lahir	date	Tanggal lahir siswa
7	jenis_kelamin	enum('L','P')	Jenis kelamin (L/P)
8	alamat	text	Alamat tempat tinggal siswa
9	email_orangtua	varchar(255)	Kontak email orang tua aktif

10	created_at	timestamp	Waktu data dibuat
11	updated_at	timestamp	Waktu data terakhir diubah

e. Tabel Mata_Pelajaran

Menyimpan daftar mata pelajaran yang diajarkan.

Tabel 3. 10 Perancangan Basis Data Tabel Mata_Pelajaran

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik mata pelajaran
2	kode_mapel	varchar(10)	Kode unik mata pelajaran
3	nama_mapel	varchar(100)	Nama lengkap mata pelajaran
4	kategori	varchar(50)	Kategori (“wajib/pilihan”)
5	deskripsi	text	Penjelasan materi pembelajaran
6	created_at	timestamp	Waktu data dibuat
7	updated_at	timestamp	Waktu data terakhir diubah

f. Tabel Jadwal

Mengelola Detail penjadwalan proses belajar mengajar harian, mencocokkan pengajar, kelas, mata pelajaran, serta waktu pelaksanaan.

Tabel 3. 11 Perancangan Basis Data Tabel Jadwal

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik jadwal
2	kelas_id	bigint(20)	Id kelas
3	mata_pelajaran_id	bigint(20)	Id mata pelajaran
4	pengajar_id	bigint(20)	Id pengajar
5	hari	enum(‘senin’, ‘selasa’, ‘rabu’, ‘kamis’, ‘Jum’at’)	Hari
6	jam_mulai	time	Jam mulai kelas
7	jam_selesai	time	Jam mulai selesai
8	created_at	timestamp	Waktu data dibuat
9	updated_at	timestamp	Waktu data terakhir diubah

g. Tabel Absensi

Mencatat kehadiran harian siswa dalam sesi jadwal pelajaran tertentu.

Tabel 3. 12 Perancangan Basis Data Tabel Absensi

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik absensi
2	siswa_id	bigint(20)	Id siswa
3	jadwal_id	bigint(20)	Id jadwal pelajaran terkait

No	Nama Kolom	Tipe Data	Deskripsi
4	pengajar_id	bigint(20)	Id pengajar yang mengisi absen
5	tanggal	date	Tanggal pencatatan absensi
6	status	enum('Hadir', 'Izin', 'Sakit', 'Alfa')	Status presensi siswa
7	keterangan	varchar(255)	Keterangan tambahan (misal alasan izin)
8	dikoreksi_oleh	bigint(20)	Id admin/pengajar yang mengoreksi data
9	created_at	timestamp	Waktu data dibuat
10	updated_at	timestamp	Waktu data terakhir diubah

h. Tabel Nilai

Menyimpan perolehan nilai siswa untuk berbagai tipe ujian (Mingguan, UTS, UAS).

Tabel 3. 13 Perancangan Basis Data Tabel Nilai

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik nilai
2	siswa_id	bigint(20)	Id siswa penerima nilai
3	mata_pelajaran_id	bigint(20)	Id mata pelajaran terkait
4	pengajar_id	bigint(20)	Id pengajar yang mengInput nilai
5	jenis_nilai	enum('Mingguan', 'UTS', 'UAS')	Kategori pengisian nilai
6	nilai	decimal(5,1)	Besaran nilai numerik
7	rata-rata	decimal(5,1)	Rata-rata berkala siswa
8	semester	tinyint(4)	Semester berjalan
9	tahun_ajaran	varchar(10)	Tahun ajaran aktif
10	dikoreksi_oleh	bigint(20)	Id admin yang melakukan penyesuaian
11	created_at	timestamp	Waktu data dibuat
12	updated_at	timestamp	Waktu data terakhir diubah

i. Tabel Materi_Harian

Mengelola modul ajar, dokumen rangkuman, atau berkas pendukung pembelajaran harian yang diunggah pengajar.

Tabel 3. 14 Perancangan Basis Data Tabel Materi_Harian

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik materi harian
2	pengajar_id	bigint(20)	Id pengajar pengunggah materi
3	kelas_id	bigint(20)	Id target kelas
4	mata_pelajaran_id	bigint(20)	Id mata pelajaran
5	tanggal	date	Tanggal publikasi materi
6	judul	varchar(200)	Judul pokok bahasan materi
7	deskripsi	text	Deskripsi singkat materi
8	file_lampiran	varchar(255)	Path/URL berkas materi
9	created_at	timestamp	Waktu data dibuat
10	updated_at	timestamp	Waktu data terakhir diubah

j. Tabel Infaq

Mencatat pembayaran infaq dari siswa yang dikelola oleh admin.

Tabel 3. 15 Perancangan Basis Data Tabel Infaq

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik transaksi infaq
2	siswa_id	bigint(20)	Id siswa
3	admin_id	bigint(20)	Id admin penerima transaksi
4	jenis_infaq	Enum('Harian', 'Bulanan', 'Daftar Ulang')	Kategori infaq
5	tanggal	date	Tanggal pembayaran infaq
6	jumlah	decimal(10,2)	Nominal uang yang dibayarkan
7	keterangan	varchar(255)	Catatan tambahan transaksi
8	created_at	timestamp	Waktu data dibuat
9	updated_at	timestamp	Waktu data terakhir diubah

k. Tabel Raport

Menggabungkan berkas raport evaluasi berkala siswa pada tiap akhir semester sebelum diterbitkan.

Tabel 3. 16 Perancangan Basis Data Tabel Raport

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik raport
2	siswa_id	bigint(20)	Id siswa
3	semester	tinyint(4)	Semester penilaian
4	tahun_ajaran	varchar(10)	Rentang tahun ajaran terkait

No	Nama Kolom	Tipe Data	Deskripsi
5	catatan_wali_kelas	text	Catatan motivasi/evaluasi wali kelas
6	status	Enum('Draft', 'Diterbitkan', 'Ditarik')	Status ketersediaan berkas
7	diterbitkan_oleh	bigint(20)	Id admin kelas penerbit
8	created_at	timestamp	Waktu data dibuat
9	updated_at	timestamp	Waktu data terakhir diubah

l. Tabel Pengumuman

Mengelola informasi pengumuman untuk seluruh pengguna sistem.

Tabel 3. 17 Perancangan Basis Data Tabel Pengumuman

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik pengumuman
2	judul	varchar(200)	Judul pengumuman
3	isi	text	Konten lengkap pengumuman
4	jenis	enum('Libur', 'Ulangan', 'Umum')	Kategori informasi
5	kelas_id	bigint(20)	Id kelas target khusus
6	tanggal_mulai	date	Batas awal pengumuman
7	tanggal_selesai	date	Batas akhir pengumuman
8	created_at	timestamp	Waktu data dibuat
9	updated_at	timestamp	Waktu data terakhir diubah

m. Tabel Pengaturan

Menyimpan konfigurasi umum aplikasi berupa pasangan *key-value* global.

Tabel 3. 18 Perancangan Basis Data Tabel Pengaturan

No	Nama Kolom	Tipe Data	Deskripsi
1	id	bigint(20)	Id unik pengaturan
2	key	bigint(20)	Kunci unik konfigurasi
3	value	bigint(20)	Nilai atau isi dari konfigurasi yang disimpan
4	label	bigint(20)	Nama label penjelas untuk dibaca di halaman admin
5	grup	date	Tanggal publikasi materi
6	created_at	timestamp	Waktu data dibuat

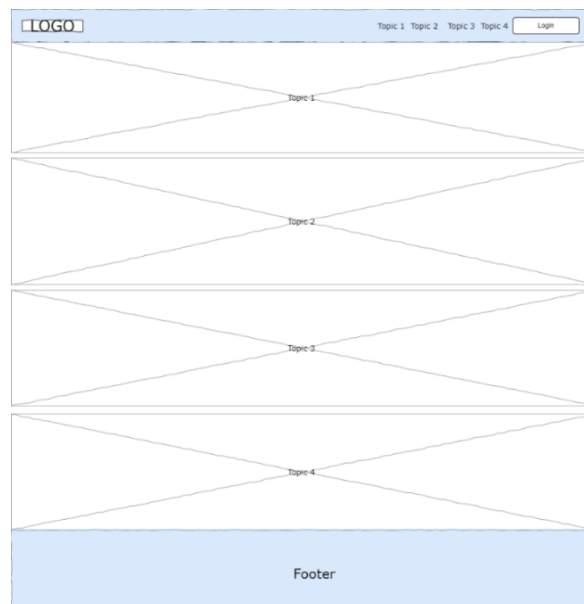
No	Nama Kolom	Tipe Data	Deskripsi
7	updated_at	timestamp	Waktu data terakhir diubah

3.5.7 Perancangan Antarmuka (*Interface*)

Perancangan antarmuka merupakan tahap penyusunan desain tampilan sistem yang digunakan sebagai pedoman pada proses pembangunan aplikasi. Pada tahap ini, rancangan visual dibuat menggunakan aplikasi draw.io dalam bentuk *wireframe* untuk menggambarkan susunan elemen dan komponen yang terdapat pada setiap halaman sistem. Tujuan dari perancangan antarmuka adalah memberikan gambaran awal mengenai tampilan serta mempermudah pengguna dan pengembang dalam memahami desain sistem sebelum memasuki tahapan implementasi. Berikut adalah rancangan antarmuka dari masing-masing halaman pada sistem informasi akademik Ruang Belajar Sa'idah.

a. Halaman *Landing Page*

Menampilkan halaman utama sistem yang dapat diakses publik tanpa login, berisi informasi, program, galeri formulir, lokasi dan *chatbot FAQ*.

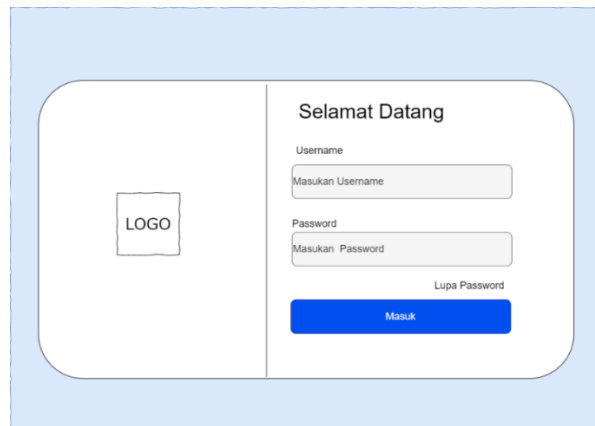


Gambar 3. 41 Antarmuka Halaman *Landing Page*

Sumber : Hasil Perancangan Penulis (2026)

b. Halaman *Login*

Menampilkan halaman login dengan desain dua panel yang dilengkapi fitur tampilkan atau sembunyikan password dan tautan lupa password.

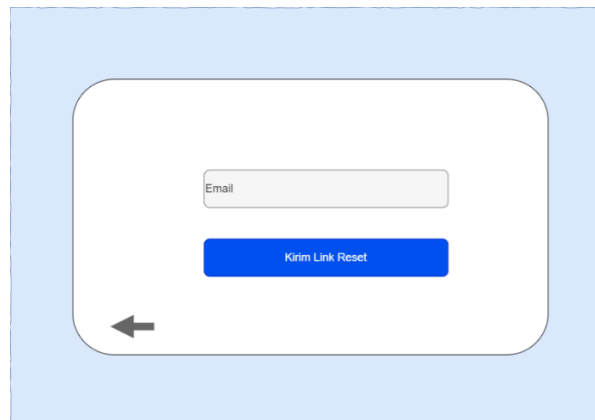


Gambar 3. 42 Antarmuka Halaman *Login*

Sumber : Hasil Perancangan Penulis (2026)

c. Halaman *Lupa Password*

Meampilkan halaman lupa password.

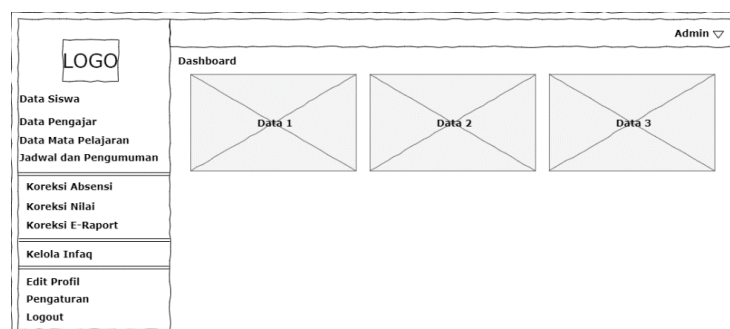


Gambar 3. 43 Antarmuka Halaman *Lupa Password*

Sumber : Hasil Perancangan Penulis (2026)

d. Halaman *Dashboard Admin*

Menampilkan dashboard admin yang menyajikan total siswa, pengajar, kelas, dan infaq secara keseluruhan.

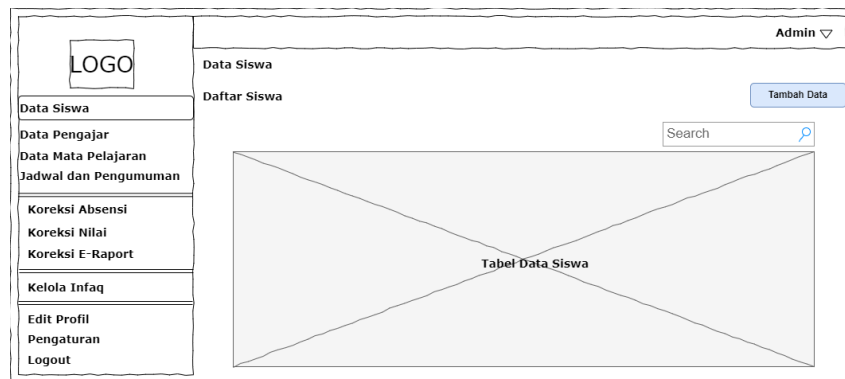


Gambar 3. 44 Antarmuka *Dashboard Admin*

Sumber : Hasil Perancangan Penulis (2026)

e. Halaman *Kelola Data Siswa*

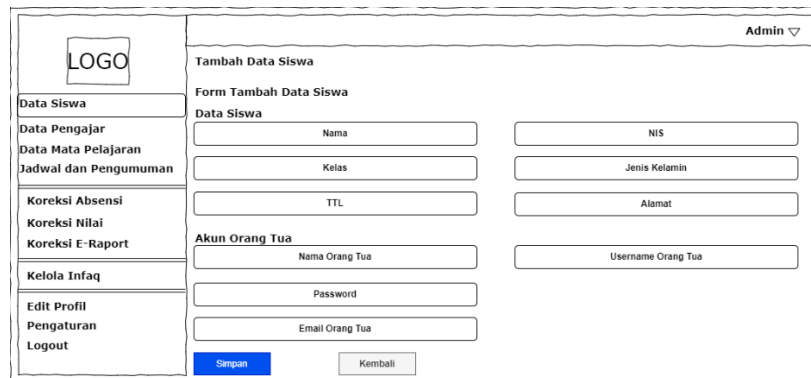
Menampilkan halaman kelola data siswa menyajikan seluruh daftar siswa.



Gambar 3. 45 Antarmuka Kelola Data Siswa

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman tambah data siswa.



Gambar 3. 46 Antarmuka Tambah Data Siswa

Sumber : Hasil Perancangan Penulis (2026)

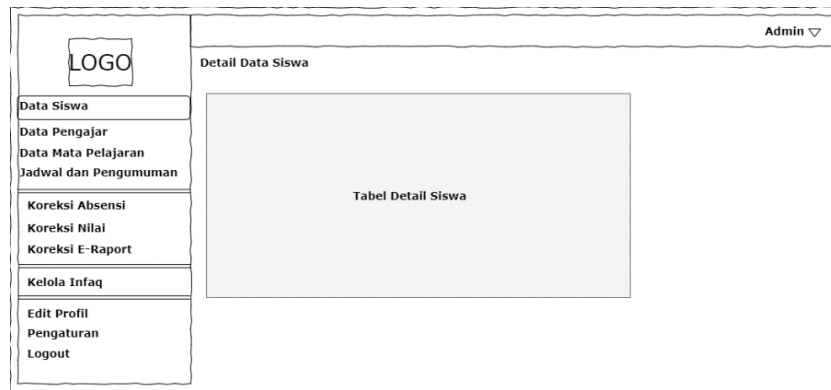
Menampilkan halaman edit siswa.



Gambar 3. 47 Antarmuka Edit Data Siswa

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman detail siswa.

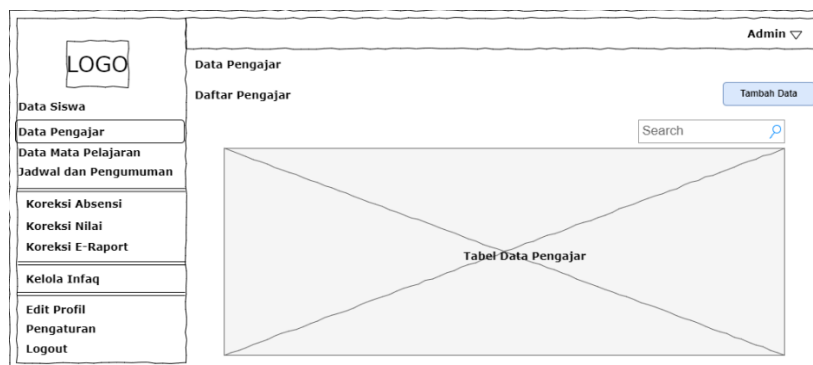


Gambar 3. 48 Antarmuka Detail Data Siswa

Sumber : Hasil Perancangan Penulis (2026)

f. Halaman Kelola Data Pengajar

Menampilkan halaman kelola data pengajar menyajikan seluruh daftar pengajar.



Gambar 3. 49 Antarmuka Kelola Data Pengajar

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman tambah data pengajar.

Gambar 3. 50 Antarmuka Tambah Data Pengajar

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman edit pengajar.

Gambar 3. 51 Antarmuka Edit Data Pengajar

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman detail pengajar.

Gambar 3. 52 Antarmuka Detail Data Pengajar

Sumber : Hasil Perancangan Penulis (2026)

g. Halaman Kelola Mata Pelajaran

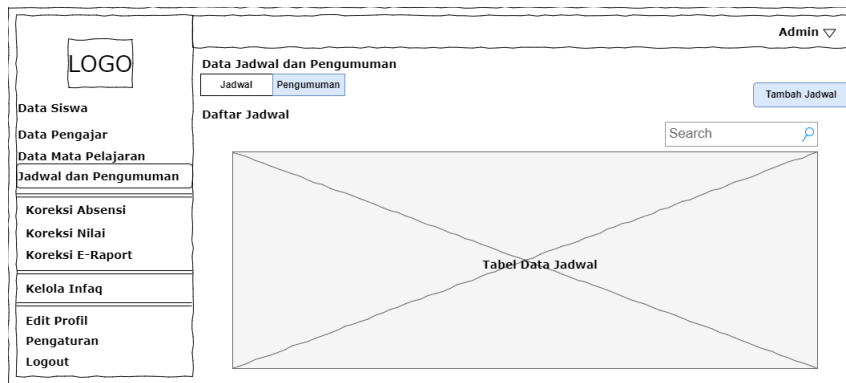
Menampilkan halaman kelola mapel, menyajikan seluruh daftar mapel.

Gambar 3. 53 Antarmuka Kelola Mata Pelajaran

Sumber : Hasil Perancangan Penulis (2026)

h. Halaman Jadwal dan Pengumuman

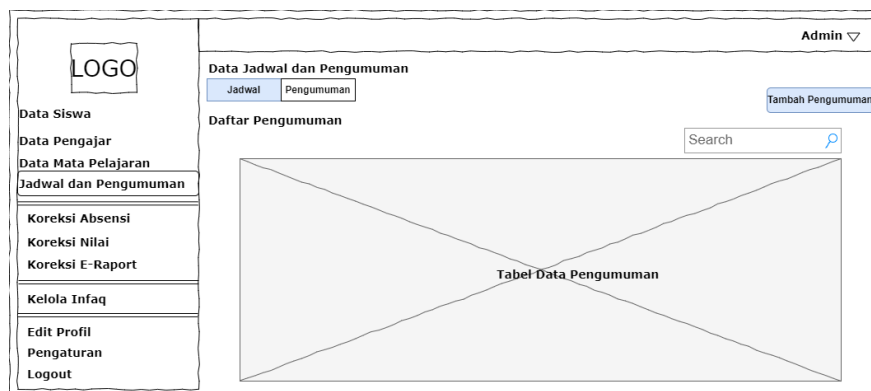
Menampilkan halaman jadwal dengan menyajikan daftar seluruh jadwal.



Gambar 3. 54 Antarmuka Tambah Jadwal

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman pengumuman dengan menyajikan seluruh daftar pengumuman.

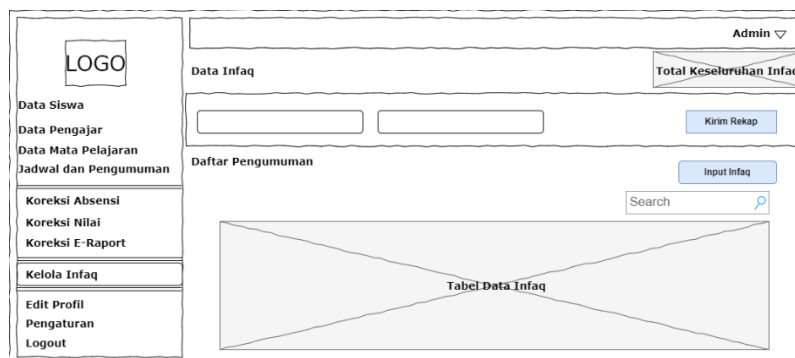


Gambar 3. 55 Antarmuka Tambah Pengumuman

Sumber : Hasil Perancangan Penulis (2026)

i. Halaman Kelola Infaq

Menampilkan kelola data infaq dan menyajikan seluruh daftar infaq.

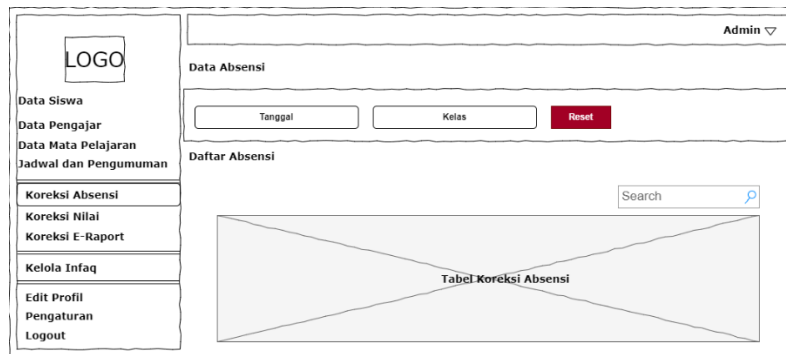


Gambar 3. 56 Antarmuka Kelola Infaq

Sumber : Hasil Perancangan Penulis (2026)

j. Halaman Koreksi Absensi

Menampilkan halaman keseluruhan daftar absensi siswa.

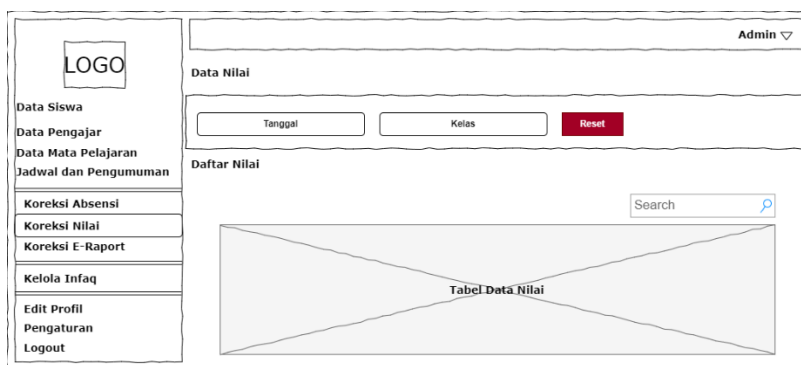


Gambar 3. 57 Antarmuka Koreksi Absensi

Sumber : Hasil Perancangan Penulis (2026)

k. Halaman Koreksi Nilai

Menampilkan daftar keseluruhan nilai siswa.

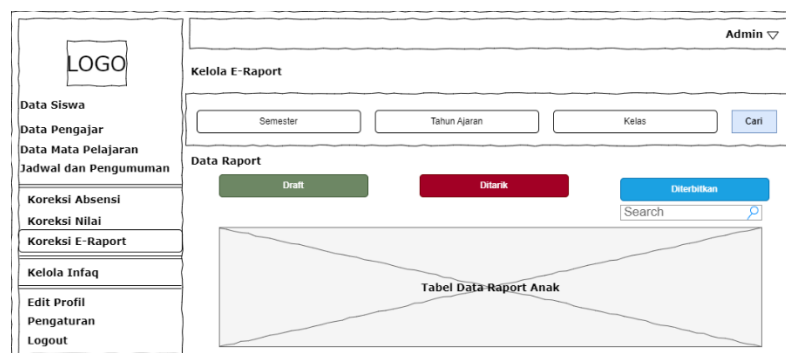


Gambar 3. 58 Antarmuka Koreksi Nilai

Sumber : Hasil Perancangan Penulis (2026)

l. Halaman Kelola E-raport

Menampilkan daftar kelola *e-raport* siswa.



Gambar 3. 59 Antarmuka Kelola E-raport

Sumber : Hasil Perancangan Penulis (2026)

m. Halaman Pengaturan Konten Sistem

Menampilkan halaman pengaturan konten sistem dan 5 tab untuk *landing page*.

Gambar 3. 60 Antarmuka Pengaturan Konten Sistem

Sumber : Hasil Perancangan Penulis (2026)

n. Halaman *Dashboard* Pengajar

Gambar 3. 61 Antarmuka *Dashboard* Pengajar

Sumber : Hasil Perancangan Penulis (2026)

o. Halaman *Input Absensi*

Menampilkan halaman input absensi siswa.

Gambar 3. 62 Antarmuka *Input Absensi*

Sumber : Hasil Perancangan Penulis (2026)

p. Halaman *Input Nilai*

Menampilkan halaman input nilai siswa.

Gambar 3. 63 Antarmuka *Input Nilai*

Sumber : Hasil Perancangan Penulis (2026)

q. Halaman *Input Materi Harian*

Menampilkan halaman input materi harian siswa.

Gambar 3. 64 Antarmuka *Input Materi Harian*

Sumber : Hasil Perancangan Penulis (2026)

r. Halaman *Dashboard Orang Tua*

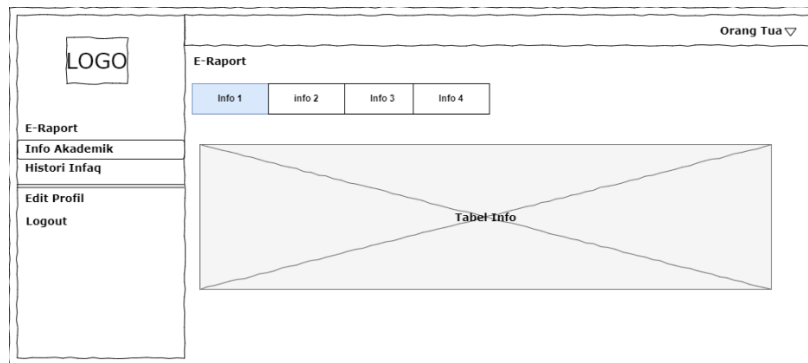
Menampilkan halaman informasi data dan jadwal anak di *dashboard* orang tua.

Gambar 3. 65 Antarmuka *Dashboard Orang Tua*

Sumber : Hasil Perancangan Penulis (2026)

s. Halaman *Info Akademik*

Menampilkan halaman informasi akademik siswa.

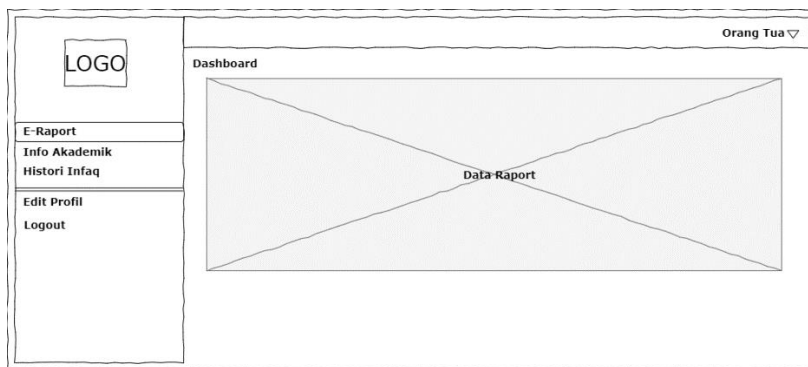


Gambar 3. 66 Antarmuka Info Akademik

Sumber : Hasil Perancangan Penulis (2026)

t. Halaman Raport Orang Tua

Menampilkan halaman dan hasil *e-raport* siswa.

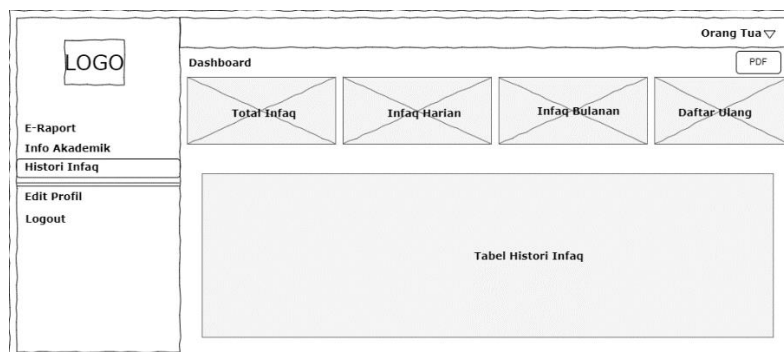


Gambar 3. 67 Antarmuka Lihat E-raport

Sumber : Hasil Perancangan Penulis (2026)

u. Halaman Histori Infaq

Menampilkan halaman informasi infaq siswa.

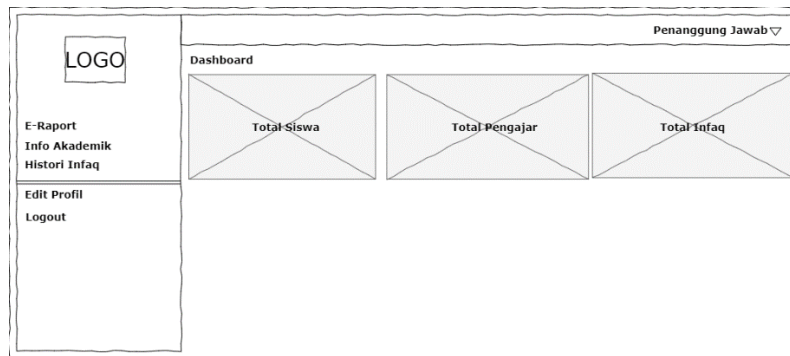


Gambar 3. 68 Antarmuka Histori Infaq

Sumber : Hasil Perancangan Penulis (2026)

v. Halaman *Dashboard* Penanggung Jawab

Menampilkan halaman informasi total siswa, pengajar, infaq.

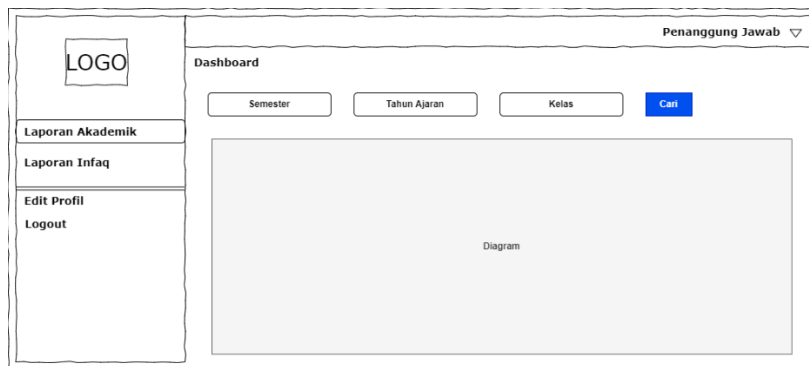


Gambar 3. 69 Antarmuka *Dashboard* Penanggung Jawab

Sumber : Hasil Perancangan Penulis (2026)

w. Halaman Laporan Akademik dan Infaq

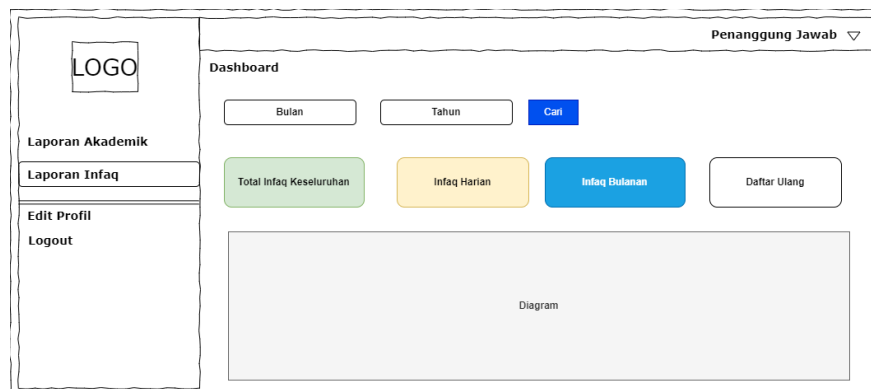
Menampilkan halaman laporan akademik.



Gambar 3. 70 Antarmuka Laporan Akademik

Sumber : Hasil Perancangan Penulis (2026)

Menampilkan halaman laporan infaq.



Gambar 3. 71 Antarmuka Laporan Infaq

Sumber : Hasil Perancangan Penulis (2026)