UNIVERSITAS
NUSA MANDIRI

AR-RAUDHAH ACADEMIC SYSTEM



SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE UNTUK PENGELOLAAN DATA AKADEMIK SEKOLAH

SALMAN AL FARISI
DRA. HJ. SURYA INDRIANTI, MSI
NURUL KHASANAH, M.KOM

BAB IV

RANCANGAN SISTEM DAN PROGRAM USULAN

4.1 Analisa Kebutuhan *Software*

Analisa kebutuhan perangkat lunak bertujuan mengidentifikasi kebutuhan pengguna dalam penggunaan sistem informasi akademik berbasis web yang akan dirancang. Sistem ini dirancang untuk mempermudah penyampaian informasi sekolah, proses Penerimaan Peserta Didik Baru (PPDB), pengelolaan data siswa, serta pengelolaan dan akses *e-raport* secara online.

Berdasarkan hasil analisis, terdapat empat jenis pengguna yang akan menggunakan sistem, yaitu superadmin, admin, siswa, dan *guest* (pengunjung).

1. Kebutuhan Superadmin

   Superadmin memiliki hak akses penuh terhadap seluruh fitur yang tersedia dalam sistem, antara lain:

   a. Login ke sistem.
   b. Mengelola data admin.
   c. Mengelola data siswa.
   d. Mengelola data guru/organisasi.
   e. Mengelola informasi sekolah.
   f. Mengelola berita sekolah.
   g. Mengelola data kerja sama/PKL.
   h. Mengelola data PPDB.
   i. Mengelola data *e-raport*.
   j. Mengelola SPP
   k. Mengelola pengumuman PPDB

l. Logout dari sistem.

2. Kebutuhan Admin

a. Login ke sistem.

b. Melihat dashboard admin.

c. Melihat data siswa yang di kelola

d. Upload *e-raport* siswa.

e. Logout dari sistem.

3. Kebutuhan Siswa

a. Login menggunakan akun siswa.

b. Melihat profil siswa.

c. Melihat data diri serta upload foto/hapus.

d. Melihat *e-raport* yang telah diunggah.

e. Melihat tunggakan SPP

f. Logout dari sistem.

4. Kebutuhan *Guest* (Pengunjung)

a. Melihat profil sekolah.

b. Melihat visi dan misi sekolah.

c. Melihat struktur organisasi.

d. Melihat informasi kerja sama/PKL.

e. Melihat fasilitas sekolah.

f. Melihat berita sekolah.

g. Melihat informasi sekolah.

h. Melihat ekstrakurikuler

i. Melihat informasi PPDB.

j. Membuat akun PPDB.

k. Login ke akun PPDB.

l. Melakukan pendaftaran PPDB.

m. Melihat hasil seleksi menggunakan kode pendaftaran.

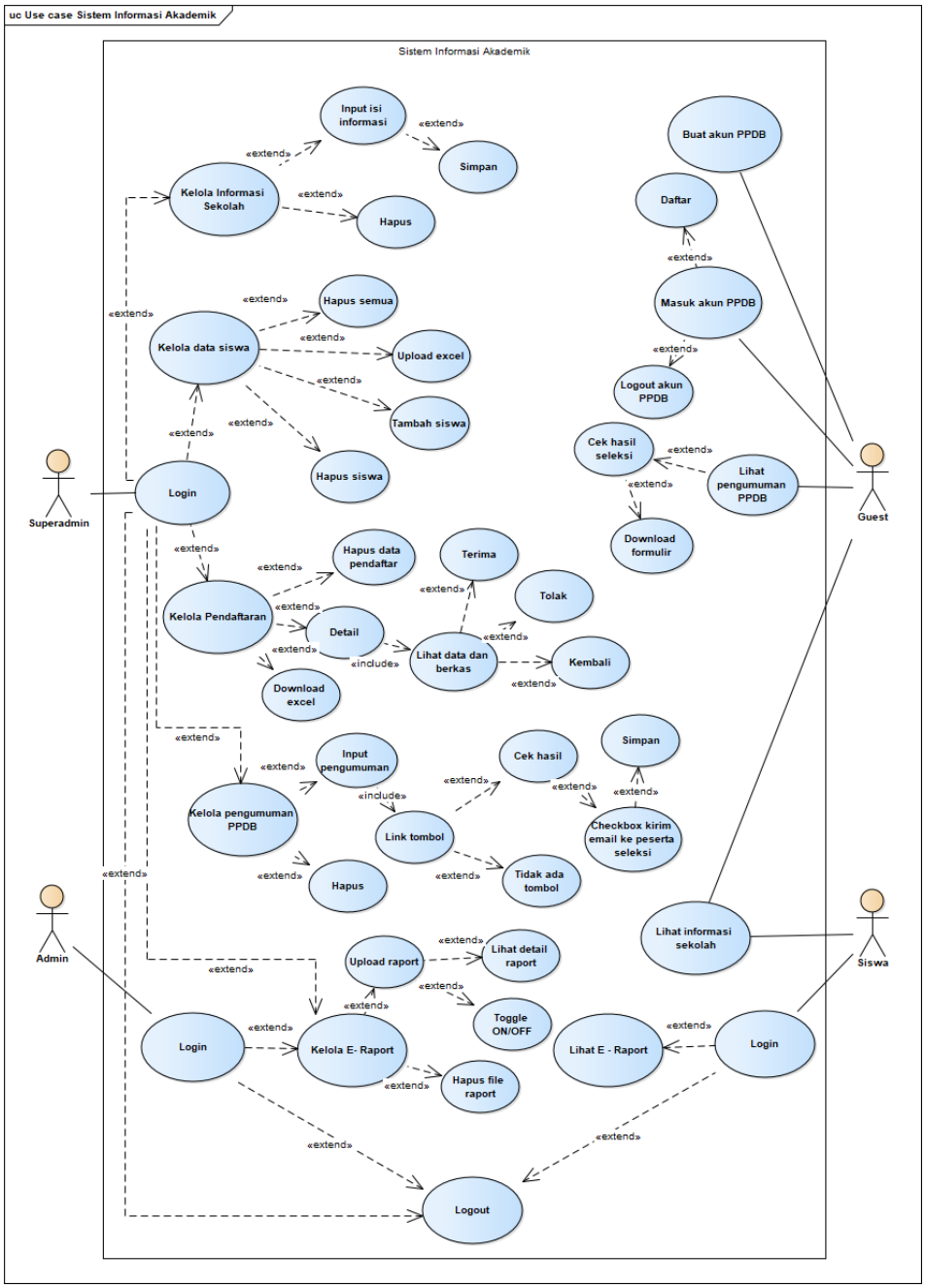
n. Logout akun PPDB.

4.2 Desain

Tahap desain bertujuan untuk menggambarkan rancangan sistem yang akan dibangun sehingga mempermudah proses implementasi sistem.

4.2.1 Desain Pemodelan Sistem

A. *Use Case Diagram*



Gambar IV.1 *Use case diagram* sistem informasi akademik

Tabel IV.1 Use diagram sistem informasi akademik

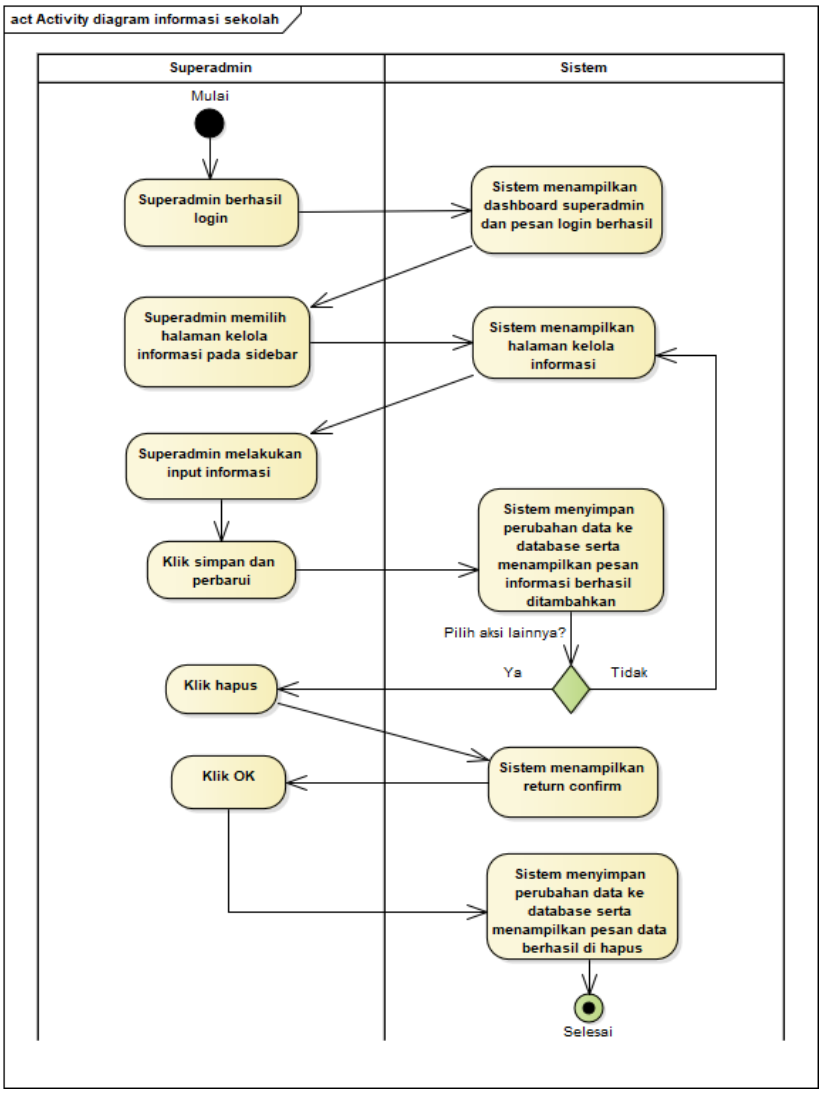
No	*Use Case* Name	Deskripsi	Aktor	*Precondition*	*Postcondition*
1.	Login	Proses pengguna masuk ke dalam sistem menggunakan akun yang valid	Superadmin, Admin, Siswa	Pengguna memiliki akun	Pengguna berhasil masuk ke sistem
2.	Kelola Informasi Sekolah	Mengelola informasi sekolah berupa teks.	Superadmin	Sudah login ke sistem	Informasi sekolah tersimpan atau dihapus
3.	Kelola Data Siswa	Mengelola data siswa dengan menambah, menghapus, atau mengunggah data melalui file Excel.	Superadmin	Sudah login ke sistem	Data siswa berhasil diperbarui
4.	Kelola Pendaftaran	Mengelola data pendaftar PPDB, melihat detail dan berkas pendaftar, serta menerima atau menolak pendaftaran.	Superadmin	Sudah login, data pendaftar tersedia	Status pendaftaran diperbarui

No	*Use Case* Name	Deskripsi	Aktor	*Precondition*	*Postcondition*
5.	Kelola Pengumuman PPDB	Membuat dan menghapus pengumuman PPDB yang akan ditampilkan kepada peserta PPDB.	Superadmin	Sudah login ke sistem	Pengumuman tersimpan dan dapat dilihat peserta PPDB
6.	Kelola *E-Raport*	Mengunggah, menghapus, mengatur status publikasi *e-raport* siswa dan melihat detail *e-raport*.	Superadmin, Admin	Sudah login ke sistem	Data *e-raport* tersimpan atau diperbarui
7.	Daftar	Proses pendaftaran PPDB bagi calon siswa.	*Guest*	Sudah login ke akun PPDB	Calon siswa mendapatkan kode pendaftaran
8.	Buat Akun PPDB	Membuat akun untuk bisa mendaftar pada sistem PPDB.	*Guest*	Belum memiliki akun	Akun PPDB berhasil dibuat
9.	Masuk Akun PPDB	Proses login ke sistem PPDB menggunakan akun yang telah dibuat.	*Guest*	Memiliki akun PPDB	Berhasil masuk ke akun PPDB
10.	Logout Akun PPDB	Keluar dari sesi akun PPDB.	*Guest*	Sedang login pada akun PPDB	Kembali ke halaman dashboard PPDB
11.	Lihat Pengumuman PPDB	Melihat informasi dan pengumuman terkait PPDB.	*Guest*	Pengumuman telah dipublikasikan	Pengumuman ditampilkan

No	*Use Case* Name	Deskripsi	Aktor	*Precondition*	*Postcondition*
12.	Cek Hasil Seleksi	Mengecek hasil seleksi penerimaan peserta didik baru menggunakan kode pendaftaran.	*Guest*	Seleksi telah diumumkan	Hasil seleksi ditampilkan
13.	Download Formulir	Mengunduh formulir yang diperlukan untuk daftar ulang di sekolah.	*Guest*	Calon siswa di nyatakan lulus	Formulir berhasil diunduh
14.	Lihat informasi sekolah	Melihat profil dan informasi sekolah yang dipublikasikan.	Siswa,*Guest*	Informasi sekolah tersedia	Informasi sekolah ditampilkan
15.	Lihat *e-raport*	Melihat *e-raport* yang telah dipublikasikan oleh sekolah.	Siswa	*E-raport* tersedia	*E-raport* ditampilkan
16.	Logout	Keluar dari sistem akademik.	Superadmin, Admin,Siswa	Sudah login ke sistem	Sistem kembali ke halaman login

B. *Activity Diagram*

1. *Activity Diagram* Informasi Sekolah

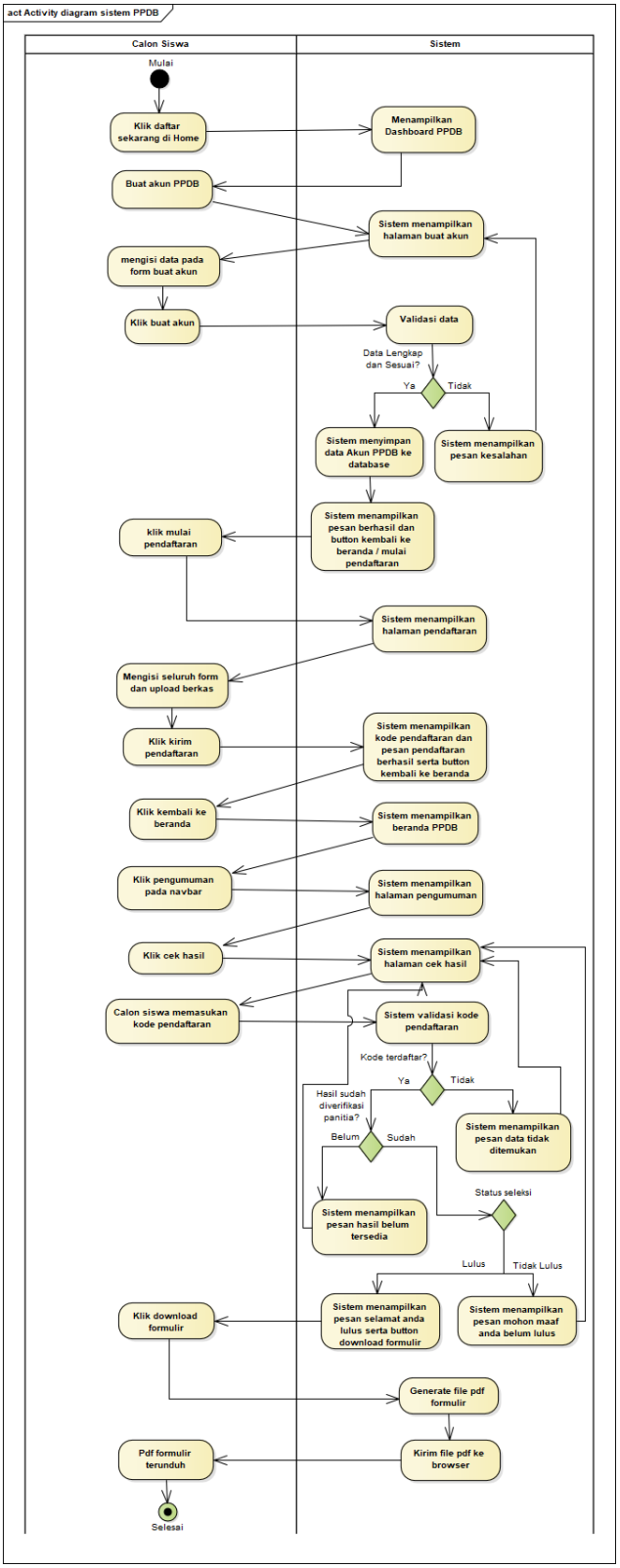


Gambar IV.2 *Activity diagram* informasi sekolah

Gambar IV.2 di atas merupakan Activity Diagram Informasi Sekolah yang melibatkan dua pihak, yaitu Superadmin dan Sistem. Proses diawali ketika superadmin berhasil login ke dalam sistem. Setelah login berhasil, sistem memuat dashboard superadmin beserta pesan bahwa proses login berhasil. Selanjutnya, superadmin membuka menu Kelola Informasi. Sistem selanjutnya menyajikan halaman kelola informasi yang berisi data informasi sekolah yang dapat dikelola pada tampilan tersebut, superadmin bisa melakukan input atau

perubahan informasi sekolah yang diperlukan. Setelah selesai melakukan input atau perubahan data, superadmin mengklik tombol Simpan dan Perbarui. Sistem kemudian menyimpan perubahan data ke dalam database dan menampilkan pesan bahwa informasi berhasil ditambahkan atau diperbarui. Jika superadmin memilih untuk melanjutkan pengelolaan data, maka sistem akan memuat kembali halaman kelola informasi. Namun, jika superadmin memilih untuk menghapus data informasi, maka superadmin mengklik tombol Hapus. Selanjutnya sistem menampilkan kotak konfirmasi penghapusan data (return confirm). Apabila superadmin menyetujui proses penghapusan dengan menekan tombol OK, maka sistem akan menghapus data dari basis data dan menampilkan notifikasi data telah dihapus. Setelah proses tersebut selesai, alur aktivitas berakhir.

2. *Activity Diagram* Sistem PPDB



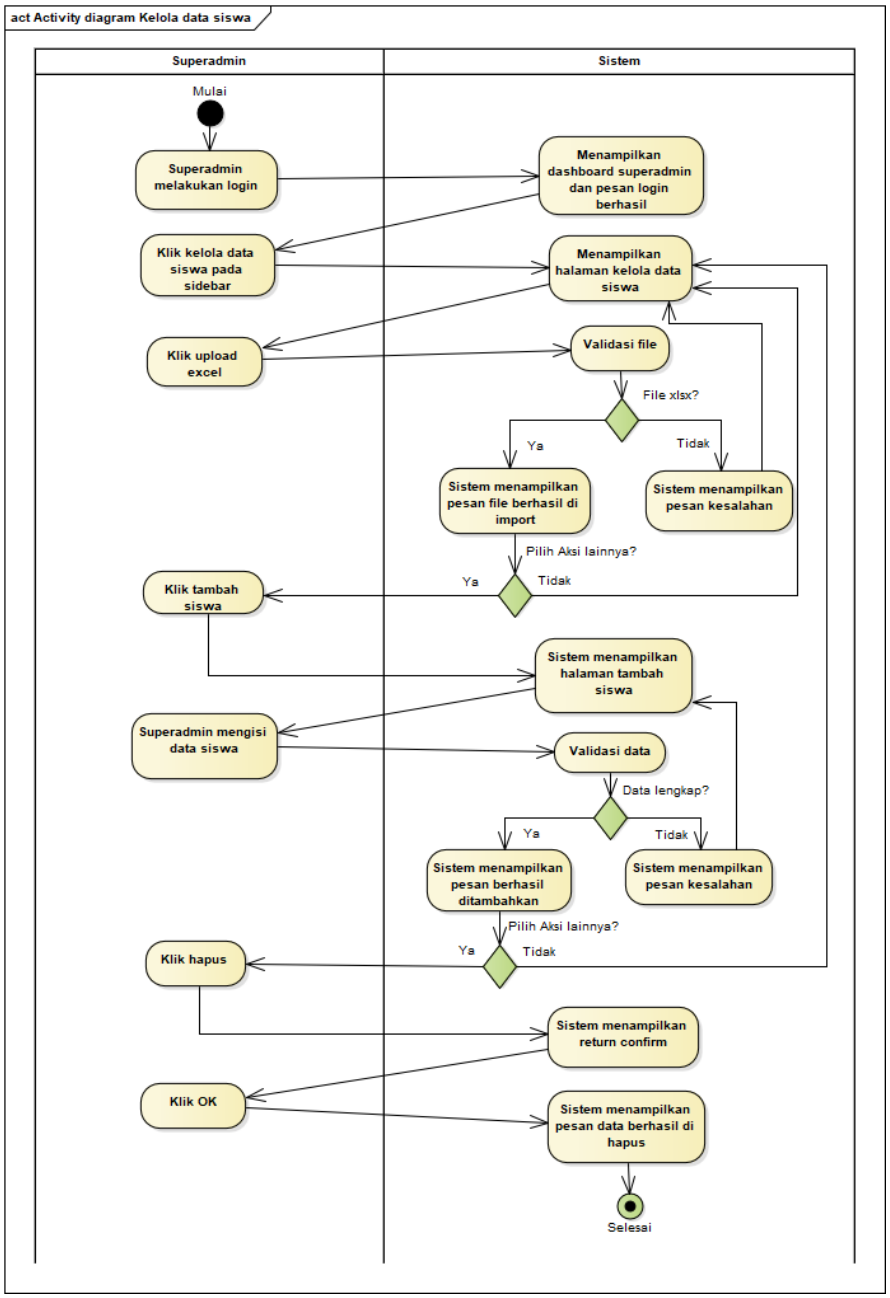
Gambar IV.3 *Activity diagram* sistem PPDB

Gambar IV.3 di atas merupakan *Activity Diagram* Sistem PPDB yang melibatkan dua pihak, yaitu Calon Siswa dan Sistem. Proses diawali ketika calon siswa menekan tombol Daftar Sekarang pada halaman beranda. Sistem kemudian menampilkan dashboard PPDB dan informasi maupun alur PPDB. Selanjutnya calon siswa membuat akun PPDB dengan mengisi data pada formulir pembuatan akun. Setelah seluruh data diisi, calon siswa menekan tombol Buat Akun. Sistem selanjutnya memvalidasi data yang telah diinput. Apabila data tersebut tidak lengkap atau tidak sesuai seperti email atau NISN yang sudah terdaftar, sistem akan menampilkan pesan kesalahan dan calon siswa diminta untuk memperbaiki data yang diinput. Namun apabila data telah lengkap dan sesuai, sistem akan menyimpan data akun PPDB ke dalam basis data dan menampilkan pesan bahwa akun berhasil dibuat beserta tombol untuk memulai pendaftaran. Setelah akun berhasil dibuat, calon siswa menekan tombol Mulai Pendaftaran dan sistem menampilkan halaman formulir pendaftaran. Pada tahap ini calon siswa mengisi seluruh data pendaftaran serta mengunggah berkas persyaratan yang diperlukan. Setelah seluruh data dan berkas diunggah, calon siswa menekan tombol Kirim Pendaftaran. Sistem kemudian menyimpan data pendaftaran dan menampilkan kode pendaftaran sebagai identitas peserta, serta menampilkan pesan bahwa pendaftaran berhasil dilakukan beserta tombol untuk kembali ke beranda.

Selanjutnya calon siswa dapat melihat hasil seleksi dengan memilih menu Pengumuman pada *navigation bar*. Sistem menampilkan halaman pengumuman dan calon siswa menekan tombol Cek Hasil. Sistem kemudian menampilkan halaman pengecekan hasil seleksi. Pada halaman tersebut calon siswa memasukkan kode pendaftaran yang telah diperoleh sebelumnya. Setelah

kode pendaftaran dimasukkan, sistem melakukan validasi terhadap kode tersebut. Apabila kode pendaftaran tidak tercatat di dalam basis data, sistem menampilkan pemberitahuan bahwa data tidak ditemukan. Namun apabila kode pendaftaran ditemukan, sistem akan memeriksa status verifikasi hasil seleksi yang dilakukan oleh panitia. Jika hasil seleksi masih dalam proses verifikasi dan belum ditentukan oleh panitia, sistem akan menampilkan pesan bahwa hasil seleksi belum tersedia dan masih dalam proses verifikasi. Sebaliknya, jika hasil seleksi telah diverifikasi, sistem akan memeriksa status kelulusan calon siswa. Apabila calon siswa dinyatakan lulus, sistem akan menampilkan pesan ucapan selamat bahwa calon siswa telah lulus seleksi PPDB dan menyediakan tombol Download Formulir. Ketika tombol tersebut ditekan, sistem akan membuat (*generate*) file formulir dalam format PDF dan mengirimkan file tersebut ke browser sehingga formulir dapat diunduh oleh calon siswa. Namun apabila calon siswa dinyatakan tidak lulus, sistem akan menampilkan pesan bahwa calon siswa belum dinyatakan lulus pada seleksi PPDB tahun tersebut. Setelah proses pengecekan hasil seleksi maupun pengunduhan formulir selesai dilakukan, maka alur aktivitas berakhir.

3. *Activity Diagram* Kelola Data Siswa

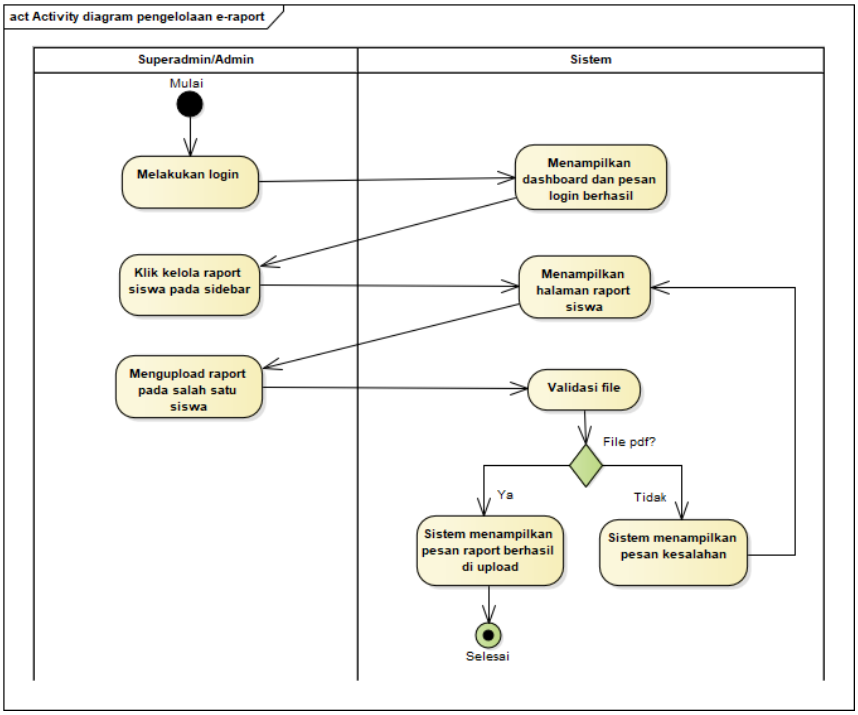


Gambar IV.4 *Activity diagram* kelola data siswa

Gambar IV.4 di atas merupakan *Activity Diagram* Kelola Data Siswa yang melibatkan dua pihak, yaitu Superadmin dan Sistem. dimulai ketika superadmin melakukan login dan menekan tombol kelola data siswa. Sistem kemudian memuat daftar data siswa yang tersimpan di dalam basis data. Superadmin kemudian dapat menentukan tindakan yang akan dilakukan seperti menambah,

mengupload Excel, atau menghapus data siswa. jika superadmin memilih tambah data siswa, maka sistem akan menampilkan form input data siswa. Setelah data diisi, sistem melakukan validasi data. Apabila data belum lengkap atau tidak memenuhi ketentuan, maka sistem akan menampilkan pesan kesalahan dan superadmin diminta melengkapi data tersebut. Namun apabila data sesuai, sistem akan menyimpan data siswa ke dalam database. Selain menambah data secara manual, superadmin juga dapat melakukan upload data siswa melalui file Excel. Sistem akan memproses file yang diunggah kemudian menyimpan data siswa secara otomatis ke dalam database apabila format data sesuai. Superdmin juga dapat menghapus data siswa yang tidak diperlukan. Setelah proses penghapusan berhasil dilakukan, sistem akan menampilkan notifikasi bahwa data berhasil dihapus. Seluruh pembaruan data yang dilakukan admin akan langsung tersimpan ke dalam database sehingga data siswa selalu terbarui secara otomatis.

4. *Activity Diagram* Pengelolaan *E-Raport*

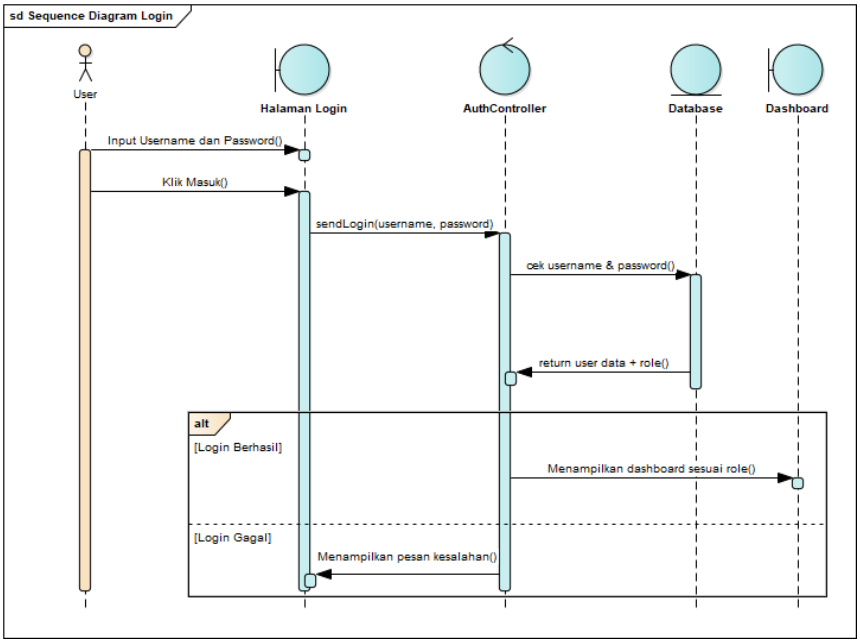


Gambar IV.5 *Activity diagram* pengelolaan e-raport

Gambar IV.5 dimulai ketika superadmin/admin login ke dalam sistem kemudian menekan tombol kelola raport siswa pada sidebar. Sistem akan menyajikan daftar siswa yang ada. Superadmin/admin kemudian memilih siswa yang akan ditambahkan rapor nya lalu melakukan upload file rapor dalam format PDF. Setelah file dipilih, sistem akan melakukan validasi file seperti format file yang diunggah. Apabila file tidak sesuai dengan ketentuan, maka sistem akan menampilkan notifikasi kesalahan dan admin diminta mengunggah ulang file rapor yang benar. Namun apabila file sesuai, sistem akan menyimpan file rapor ke database. Setelah proses upload berhasil, sistem akan menampilkan notifikasi bahwa rapor berhasil ditambahkan. File rapor yang telah diunggah kemudian dapat dilihat oleh siswa melalui akun masing-masing pada menu *e-raport* secara online.

C. *Sequence Diagram*

1. *Sequence diagram* Login



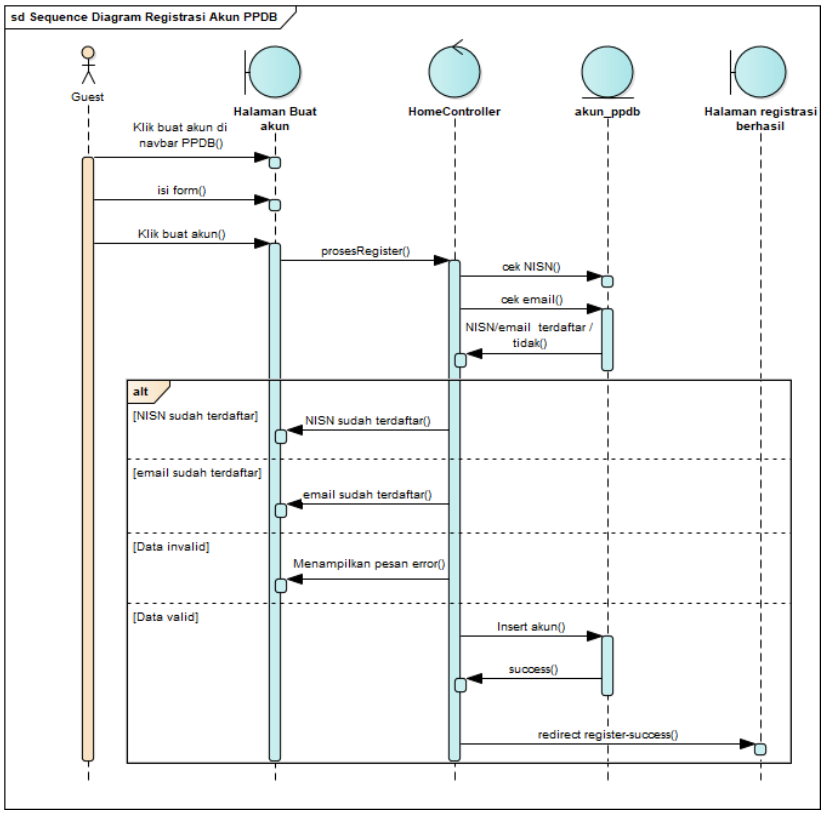
Gambar IV.6 *Sequence diagram* login

Gambar IV.6 merupakan *Sequence Diagram* Login yang menggambarkan alur proses autentikasi pengguna ke dalam sistem. Sequence diagram ini melibatkan empat objek, yaitu User sebagai aktor, Halaman Login sebagai *boundary*, Auth Controller sebagai *control*, dan database sebagai *entity*. Proses dimulai saat pengguna mengakses halaman login dan menginput username serta password pada formulir login. Setelah tombol masuk ditekan, sistem mengirimkan data username dan password ke Auth Controller untuk diproses. Selanjutnya Auth Controller memverifikasi kecocokan data username dan password ke database. Apabila username dan password yang dimasukkan tidak ditemukan pada database, database mengembalikan hasil bahwa data tidak ditemukan. Auth Controller kemudian mengirimkan pesan kesalahan kepada halaman login berupa

informasi bahwa username dan password tidak ditemukan, sehingga pengguna diminta untuk memasukkan data yang benar.

Namun apabila username dan password sesuai, Auth Controller akan menampilkan pesan bahwa login berhasil dan mengarahkan pengguna ke halaman dashboard seusai role nya, Setelah dashboard ditampilkan, proses login dinyatakan selesai.

2. *Sequence diagram* Registrasi Akun PPDB



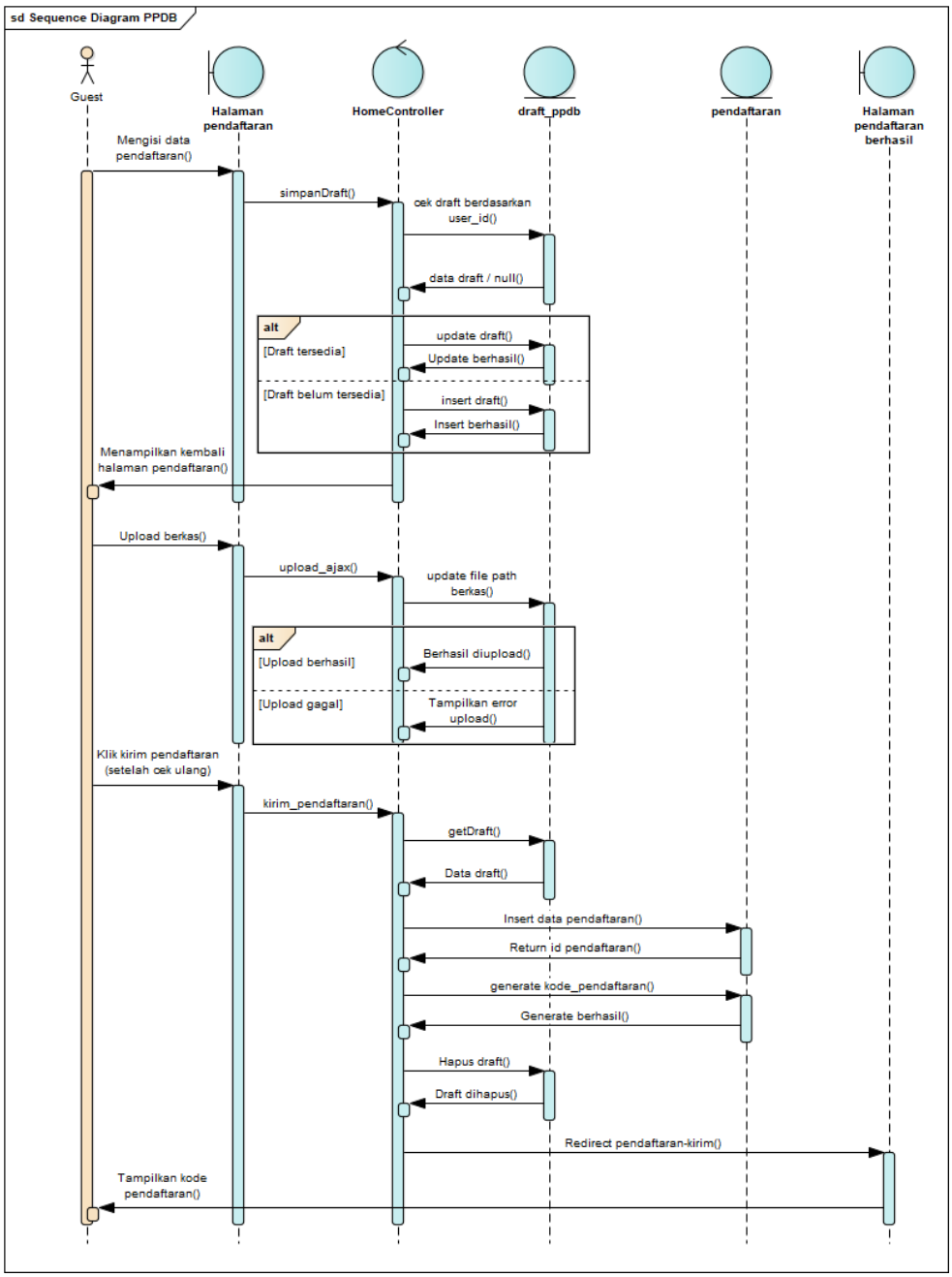
Gambar IV.7 *Sequence diagram* registrasi akun PPDB

Gambar IV.7 merupakan *Sequence Diagram* Registrasi Akun PPDB yang menggambarkan alur proses pembuatan akun PPDB oleh *Guest* (pengunjung). *Sequence diagram* ini melibatkan empat objek, yaitu *Guest* sebagai aktor, Halaman Buat Akun sebagai *boundary*, Home Controller sebagai *control*, dan

tabel akun_ppdb sebagai *entity*. Proses diawali ketika calon siswa mengisi formulir registrasi akun yang terdiri dari nama lengkap, NISN, email, jenis kelamin, password, dan konfirmasi password. Setelah seluruh data diisi, calon siswa menekan tombol Buat Akun. Sistem kemudian mengirimkan data registrasi ke Home Controller untuk dilakukan proses validasi. Tahap pertama yang dilakukan adalah pengecekan NISN ke tabel akun_ppdb. Database mengembalikan hasil apakah NISN tersebut sudah terdaftar atau belum. Apabila NISN telah terdaftar, sistem menampilkan pesan kesalahan bahwa NISN sudah digunakan sehingga calon siswa diminta menggunakan NISN yang berbeda atau menghubungi pihak sekolah. Apabila NISN belum terdaftar, sistem melanjutkan proses dengan melakukan pengecekan email pada tabel akun_ppdb. Database kemudian mengembalikan hasil apakah email tersebut sudah digunakan oleh akun lain atau belum. Jika email telah terdaftar, sistem menampilkan pesan kesalahan bahwa email sudah digunakan dan calon siswa diminta menggunakan email lain atau melakukan login apabila telah memiliki akun. Selanjutnya sistem melakukan validasi format password untuk memastikan password memenuhi ketentuan yang telah ditetapkan, yaitu mengandung huruf besar, huruf kecil, angka, dan memiliki panjang minimal delapan karakter. Apabila password tidak sesuai dengan ketentuan yang berlaku, sistem akan menampilkan notifikasi kesalahan dan meminta calon siswa memperbaiki password yang digunakan. Apabila seluruh data telah valid, sistem melakukan proses penyimpanan data akun ke dalam tabel akun_ppdb. Setelah data berhasil disimpan, database mengembalikan informasi bahwa proses penyimpanan berhasil beserta id akun yang baru dibuat. Sistem kemudian membuat session login secara otomatis sehingga calon siswa tidak perlu melakukan login ulang. Setelah proses registrasi selesai, sistem mengarahkan

calon siswa ke halaman Registrasi Berhasil yang berisi informasi bahwa akun PPDB telah berhasil dibuat dan siap digunakan untuk melakukan pendaftaran PPDB. Dengan demikian proses registrasi akun PPDB dinyatakan selesai.

3. *Sequence diagram* PPDB



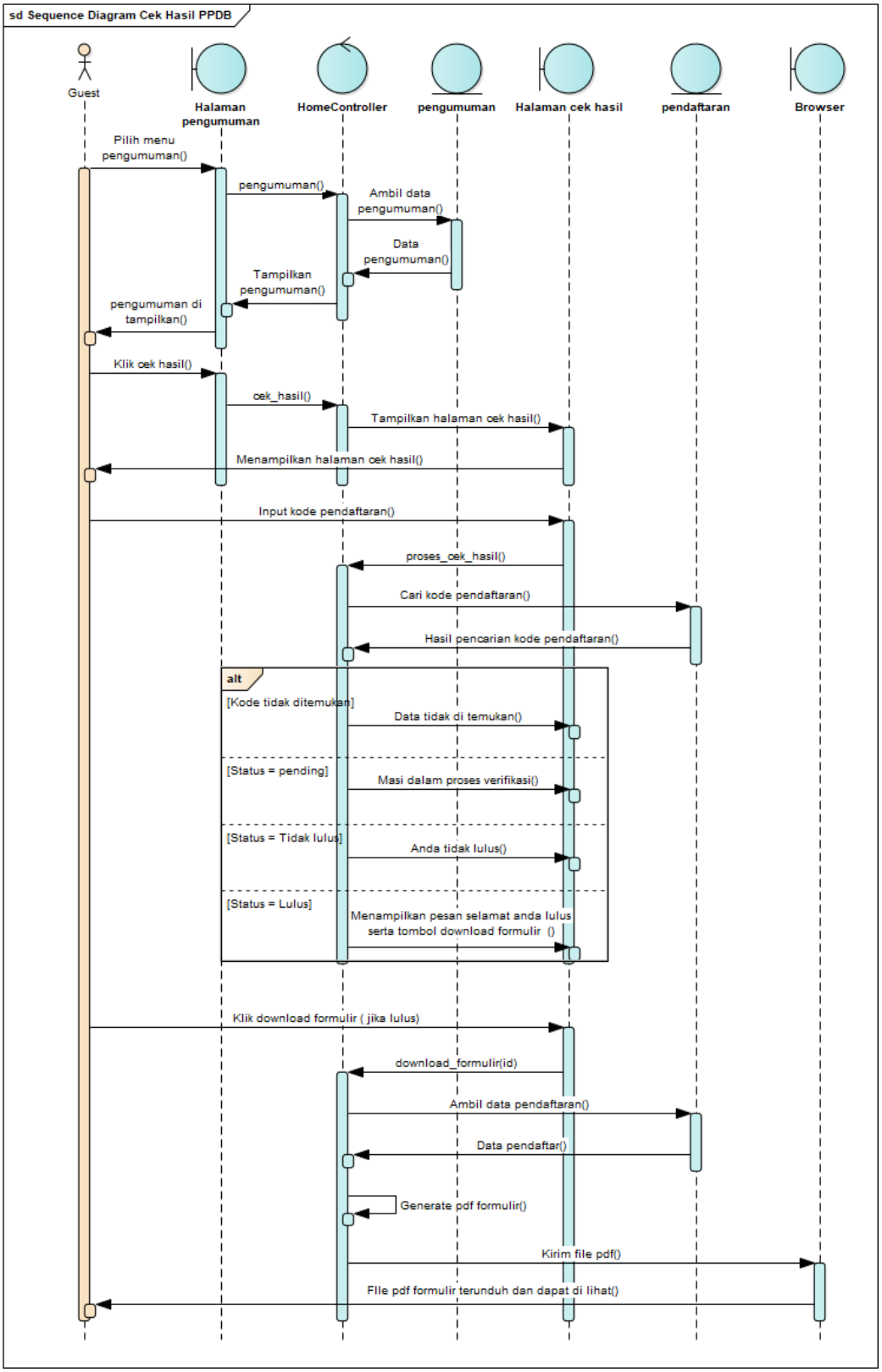
Gambar IV.8 *Sequence diagram* PPDB

Gambar IV.8 merupakan *Sequence Diagram* PPDB yang menggambarkan proses pengisian formulir pendaftaran oleh calon siswa hingga data pendaftaran berhasil dikirim ke sistem. *Sequence diagram* ini melibatkan Calon Siswa sebagai aktor, Form Pendaftaran PPDB sebagai *boundary*, Home Controller sebagai

control, serta tabel draft_ppdb dan pendaftaran sebagai *entity*. Proses dimulai ketika calon siswa mengisi formulir pendaftaran PPDB. Setiap data yang diinput akan disimpan sementara ke tabel draft_ppdb melalui Home Controller. Sistem terlebih dahulu memeriksa apakah data draft sudah tersedia. Jika data draft telah ada, sistem melakukan pembaruan data, sedangkan jika belum tersedia sistem akan membuat data draft baru.

Selanjutnya calon siswa mengunggah berkas persyaratan yang diperlukan. Sistem melakukan validasi terhadap file yang diunggah. Apabila file memenuhi ketentuan format dan ukuran, sistem menyimpan nama file ke tabel draft_ppdb. Namun jika file tidak sesuai ketentuan, sistem menampilkan notifikasi kesalahan dan meminta calon siswa mengunggah ulang berkas yang sesuai. Setelah seluruh data dan berkas lengkap, calon siswa menekan tombol Kirim Pendaftaran. Sistem mengambil seluruh data dari tabel draft_ppdb dan memeriksa kelengkapan data. Apabila data draft tidak ditemukan, sistem mengarahkan pengguna kembali ke formulir pendaftaran. Sebaliknya, apabila data tersedia, sistem menyimpan seluruh data ke tabel pendaftaran dan menghasilkan kode pendaftaran secara otomatis. Setelah kode pendaftaran berhasil dibuat, sistem menghapus data draft yang tersimpan pada tabel draft_ppdb dan menampilkan halaman konfirmasi pendaftaran yang berisi kode pendaftaran sebagai identitas peserta PPDB.

4. *Sequence diagram* Cek Hasil PPDB

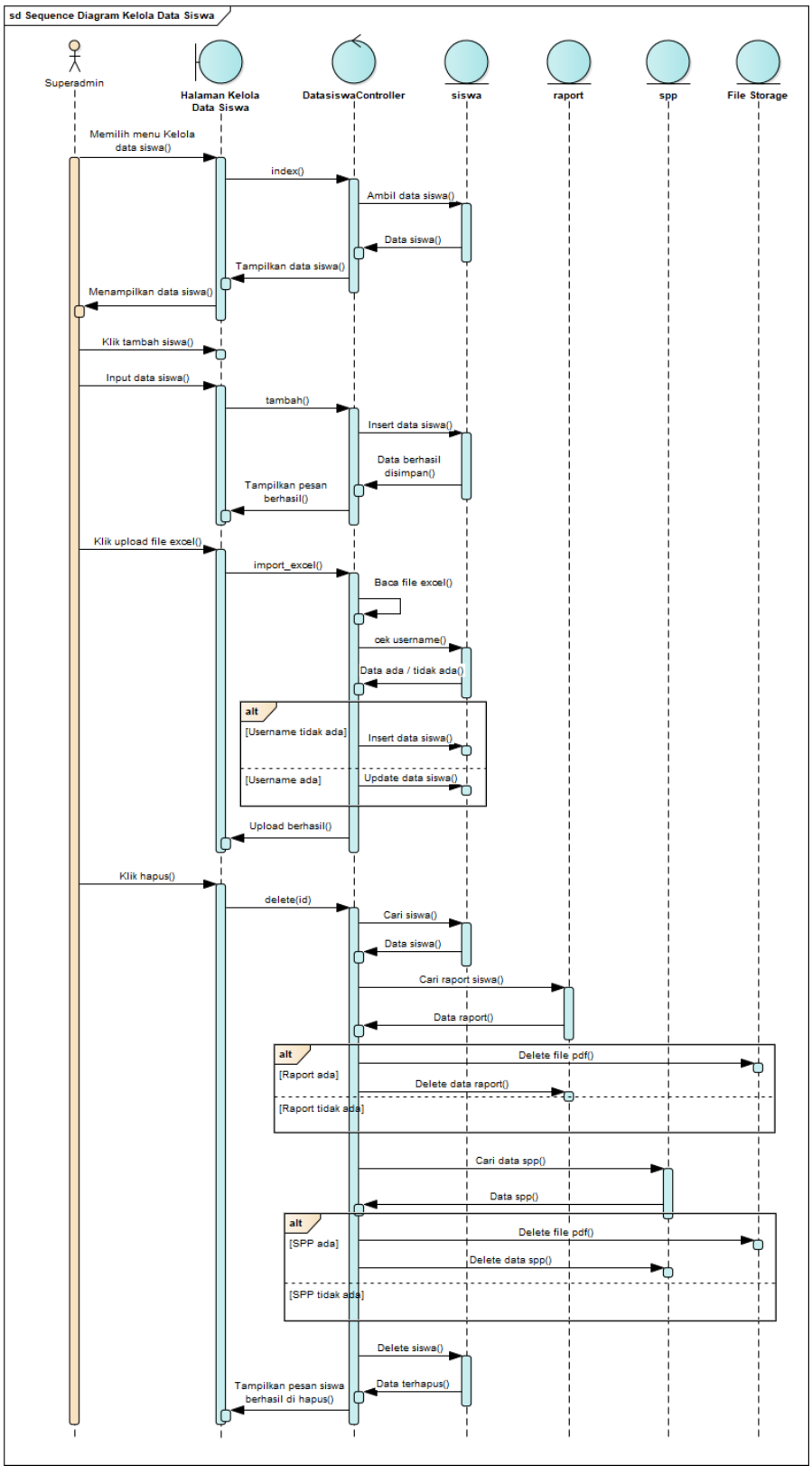


Gambar IV.9 *Sequence diagram* cek hasil PPDB

Gambar IV.9 merupakan *Sequence Diagram* Cek Hasil PPDB yang menggambarkan alur proses pengecekan hasil seleksi PPDB oleh calon siswa. *Sequence diagram* ini melibatkan Calon Siswa sebagai aktor, Halaman Pengumuman dan Halaman Cek Hasil sebagai *boundary*, Home Controller sebagai *control*, serta tabel pendaftaran sebagai *entity*. Proses diawali ketika calon siswa membuka halaman Pengumuman dan memilih tombol Cek Hasil. Sistem kemudian menampilkan Halaman Cek Hasil yang berisi formulir untuk memasukkan kode pendaftaran. Selanjutnya calon siswa menginput kode pendaftaran yang telah diperoleh pada saat proses pendaftaran PPDB dan menekan tombol Cek Hasil Setelah menerima kode pendaftaran yang dimasukkan oleh calon siswa, Home Controller melakukan proses pencarian data pada tabel pendaftaran berdasarkan kode pendaftaran tersebut. Database kemudian mengembalikan hasil pencarian kepada Home Controller untuk diproses lebih lanjut. Apabila kode pendaftaran yang diinput tidak ditemukan pada basis data, sistem akan menampilkan notifikasi data pendaftaran tidak ditemukan. Kondisi ini dapat terjadi apabila calon siswa memasukkan kode pendaftaran yang salah atau kode tersebut belum terdaftar pada sistem. Apabila data pendaftaran ditemukan, sistem selanjutnya memeriksa status hasil seleksi calon siswa. Jika status seleksi masih dalam tahap verifikasi atau belum ditentukan oleh panitia PPDB, sistem akan menampilkan informasi bahwa hasil seleksi masih dalam proses verifikasi sehingga calon siswa diminta untuk melakukan pengecekan kembali pada waktu yang telah ditentukan. Jika proses verifikasi telah selesai dan calon siswa dinyatakan tidak lulus seleksi, sistem akan menampilkan informasi bahwa calon siswa belum dinyatakan lulus pada seleksi PPDB.

Sebaliknya, apabila calon siswa dinyatakan lulus seleksi, sistem akan menampilkan pesan ucapan selamat bahwa calon siswa telah lulus PPDB serta menampilkan tombol Download Formulir. Ketika tombol tersebut dipilih, sistem mengambil data pendaftaran calon siswa dari tabel pendaftaran, kemudian menghasilkan formulir pendaftaran dalam format PDF yang sudah terisi secara otomatis berdasarkan data yang sebelumnya diinput pada saat proses pendaftaran. Selanjutnya file PDF dikirimkan ke browser sehingga dapat diunduh oleh calon siswa sebagai bukti hasil seleksi dan dokumen administrasi lanjutan. Setelah hasil seleksi ditampilkan atau formulir berhasil diunduh, proses pengecekan hasil PPDB dinyatakan selesai.

5. *Sequence diagram* Kelola Data Siswa



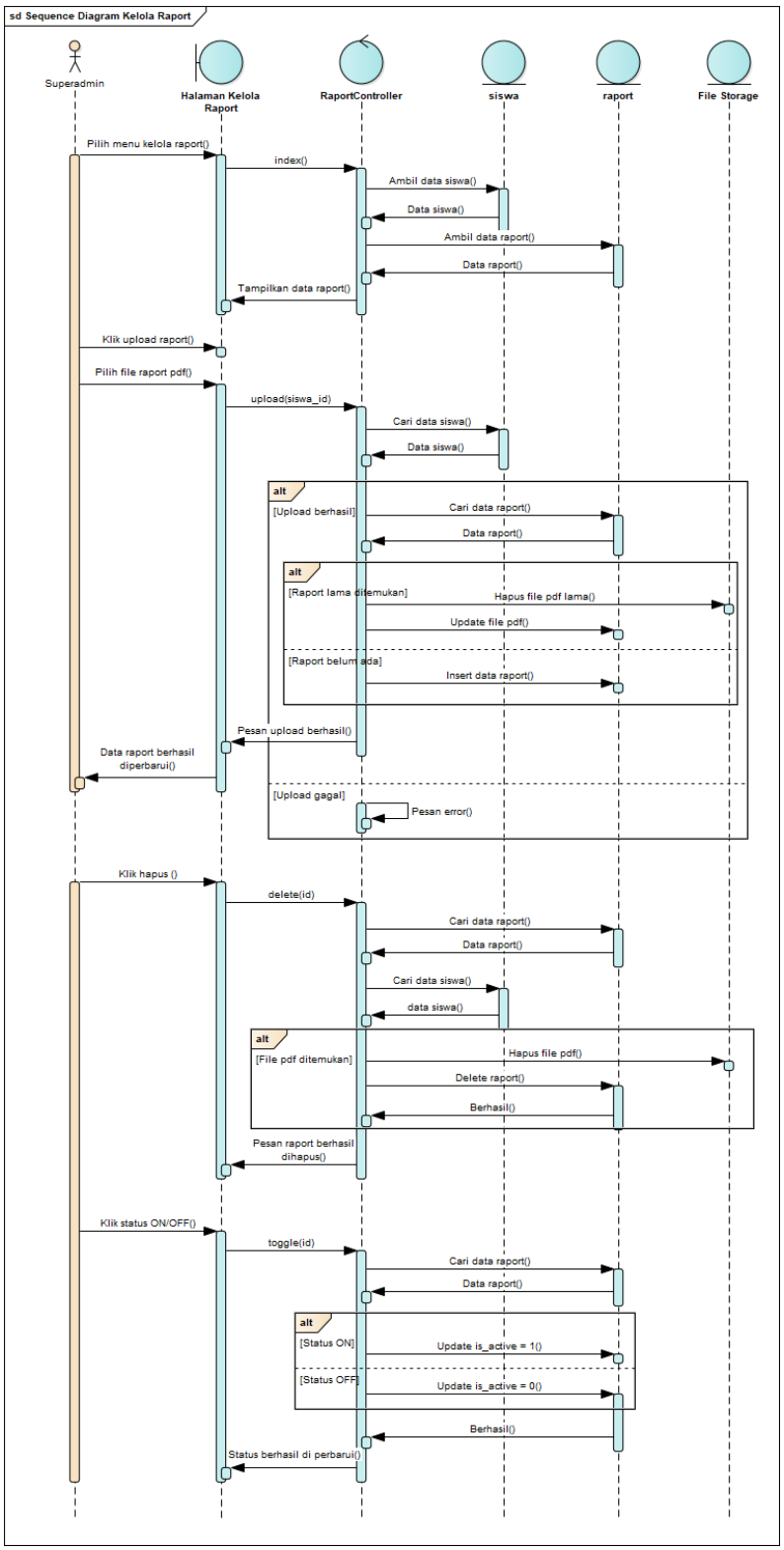
Gambar IV.10 *Sequence diagram* kelola data siswa

Gambar IV.10 merupakan *Sequence Diagram* Kelola Data Siswa yang menggambarkan proses pengelolaan data siswa yang dilakukan oleh Superadmin. *Sequence diagram* ini melibatkan Superadmin sebagai aktor, Halaman Data Siswa sebagai *boundary*, Datasiswa Controller sebagai *control*, serta tabel siswa, raport, dan spp sebagai *entity*. Pada modul ini Superadmin bisa melihat data siswa, menambahkan data siswa baru, mengimpor data siswa melalui file Excel, serta menghapus data siswa beserta seluruh data yang berkaitan dengannya. Proses diawali ketika Superadmin membuka menu Data Siswa. Halaman Data Siswa kemudian mengirimkan permintaan kepada Datasiswa Controller untuk menampilkan data siswa. Selanjutnya controller mengambil data dari tabel siswa dan mengembalikannya ke halaman sehingga daftar siswa dapat ditampilkan kepada Superadmin. Pada tahap ini Superadmin dapat melihat seluruh data siswa yang tersimpan di dalam sistem.

Selanjutnya Superadmin dapat menambahkan data siswa baru dengan memilih tombol Tambah Siswa dan mengisi formulir yang tersedia. Data yang telah diinput kemudian dikirimkan ke Datasiswa Controller untuk diproses. Controller selanjutnya menyimpan data tersebut ke tabel siswa. Setelah proses penyimpanan berhasil dilakukan, sistem menampilkan notifikasi data siswa berhasil ditambahkan dan data baru akan muncul pada daftar siswa. Selain menambahkan data secara manual, Superadmin juga dapat melakukan impor data siswa menggunakan file Excel. Proses dimulai ketika Superadmin mengunggah file Excel yang berisi data siswa. Datasiswa Controller kemudian membaca setiap baris data yang terdapat pada file tersebut. Untuk setiap data siswa, sistem melakukan pemeriksaan berdasarkan username yang digunakan. Apabila username belum terdaftar pada tabel siswa, sistem akan menambahkan data siswa

baru ke basis data. Sebaliknya, apabila username telah terdaftar, sistem akan memperbarui data siswa yang telah ada sesuai dengan informasi pada file Excel. Setelah seluruh data berhasil diproses, sistem menampilkan notifikasi proses impor data siswa telah berhasil dilakukan. Superadmin juga dapat menghapus data siswa yang tidak diperlukan lagi. Ketika tombol Hapus dipilih, Datasiswa Controller terlebih dahulu mengambil data siswa yang akan dihapus. Selanjutnya sistem memeriksa apakah siswa tersebut memiliki data rapor yang tersimpan pada tabel raport. Jika data rapor ditemukan, sistem akan menghapus data rapor tersebut. Setelah itu sistem melakukan pemeriksaan terhadap data SPP pada tabel spp. Jika data SPP ditemukan, sistem juga akan menghapus data tersebut. Setelah seluruh data yang memiliki hubungan dengan siswa berhasil dihapus, sistem menghapus data siswa dari tabel siswa. Proses ini dilakukan untuk menjaga konsistensi data sehingga tidak terdapat data rapor maupun data SPP yang tidak memiliki hubungan dengan data siswa. Setelah proses penghapusan selesai dilakukan, sistem menampilkan pesan bahwa data siswa beserta seluruh data terkait berhasil dihapus. Setelah seluruh proses pengelolaan data siswa selesai dilakukan, *sequence diagram* berakhir dan sistem menampilkan kondisi data terbaru sesuai dengan perubahan yang telah dilakukan oleh Superadmin.

6. *Sequence diagram* Kelola Rapor



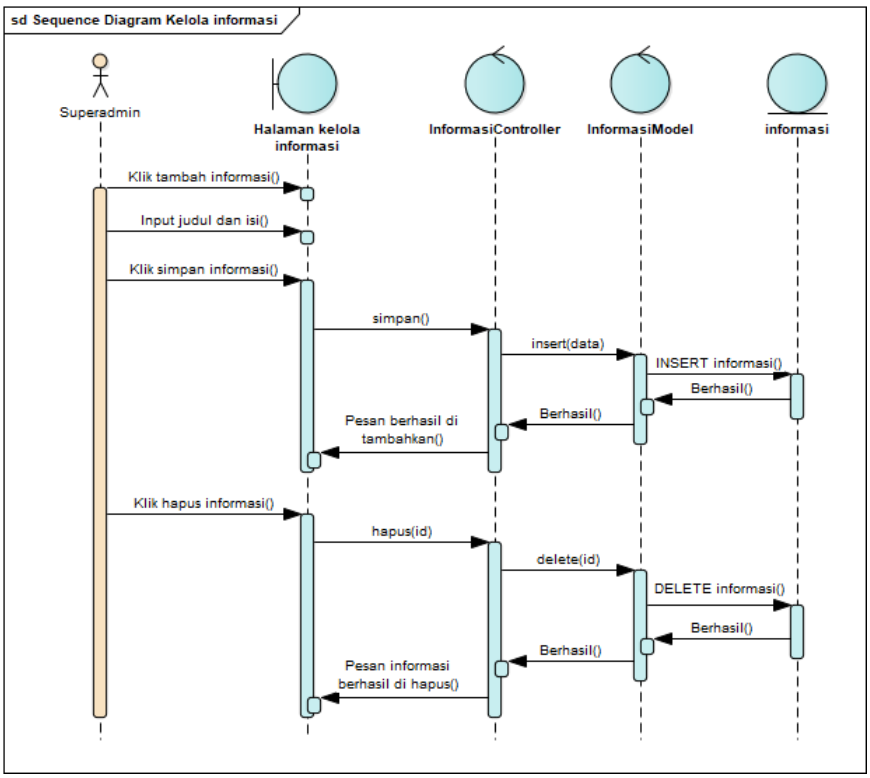
Gambar IV.11 *Sequence diagram* kelola rapor

Gambar IV.11 merupakan *Sequence Diagram* Kelola rapor yang menggambarkan proses pengelolaan file rapor siswa oleh Admin. *Sequence diagram* ini melibatkan Admin sebagai aktor, Halaman Kelola Raport sebagai *boundary*, Raport Controller sebagai *control*, serta tabel siswa dan tabel raport sebagai *entity*. Pada modul ini Admin dapat melihat data rapor siswa, mengunggah file rapor dalam format PDF, memperbarui file rapor yang telah ada, menghapus rapor, serta mengubah status publikasi rapor agar dapat diakses oleh siswa. Proses diawali ketika Admin membuka menu Kelola Raport. Halaman Kelola Raport kemudian mengirimkan permintaan kepada Raport Controller untuk menampilkan data siswa dan data rapor yang tersedia. Selanjutnya controller mengambil data dari tabel siswa dan tabel raport pada database. Data yang berhasil diperoleh kemudian dikirimkan kembali ke halaman sehingga daftar siswa beserta informasi rapor dapat ditampilkan kepada Admin. Untuk mengunggah rapor, Admin memilih siswa yang akan diberikan rapor kemudian mengklik tombol Upload Raport dan memilih file PDF yang akan diunggah. Halaman Kelola Raport mengirimkan data tersebut ke Raport Controller untuk diproses. Controller kemudian melakukan validasi file sesuai dengan ketentuan yang telah ditetapkan, yaitu hanya menerima file berformat PDF dengan ukuran maksimum yang diperbolehkan. Apabila proses upload gagal, sistem akan menampilkan pesan kesalahan kepada Admin. Namun apabila proses upload berhasil, sistem akan memeriksa apakah siswa tersebut telah memiliki data rapor sebelumnya. Jika data rapor sebelumnya ditemukan, sistem akan memperbarui data rapor yang telah ada dengan file yang baru diunggah. Sebelum melakukan pembaruan, sistem terlebih dahulu menghapus file rapor lama yang tersimpan pada server agar tidak terjadi duplikasi file. Sebaliknya, apabila siswa belum

memiliki data rapor, sistem akan menambahkan data rapor baru ke tabel raport. Setelah proses penyimpanan selesai dilakukan, sistem menampilkan notifikasi rapor berhasil diunggah.

Selain mengunggah rapor, Admin juga dapat menghapus data rapor siswa. Proses dimulai ketika Admin menekan tombol Hapus pada data rapor yang dipilih. Halaman Kelola Raport kemudian mengirimkan permintaan penghapusan kepada Raport Controller. Controller mengambil data rapor yang akan dihapus dari database dan memeriksa keberadaan file PDF pada server. Jika file ditemukan, sistem akan menghapus file tersebut dari penyimpanan server. Selanjutnya sistem menghapus data rapor dari tabel raport sehingga data rapor siswa tidak lagi tersedia pada sistem. Setelah proses penghapusan berhasil dilakukan, sistem menampilkan pesan bahwa rapor berhasil dihapus. Admin juga dapat mengubah status rapor melalui fitur aktif dan nonaktif. Proses dimulai ketika Admin mengubah status rapor pada halaman Kelola Raport. Halaman kemudian mengirimkan permintaan perubahan status kepada Raport Controller. Controller mengambil data rapor yang dipilih dan memperbarui nilai status pada database sesuai dengan pilihan Admin. Jika status diaktifkan maka raport dapat diakses oleh siswa melalui akun masing-masing. Sebaliknya, jika status dinonaktifkan maka raport tidak dapat diakses oleh siswa. Setelah proses pembaruan status selesai dilakukan, sistem menampilkan pesan bahwa status rapor berhasil diperbarui. Setelah seluruh proses pengelolaan rapor selesai dilakukan, *sequence diagram* berakhir dan sistem menampilkan data rapor terbaru sesuai dengan perubahan yang sudah dilakukan Admin.

7. *Sequence diagram* Kelola Informasi

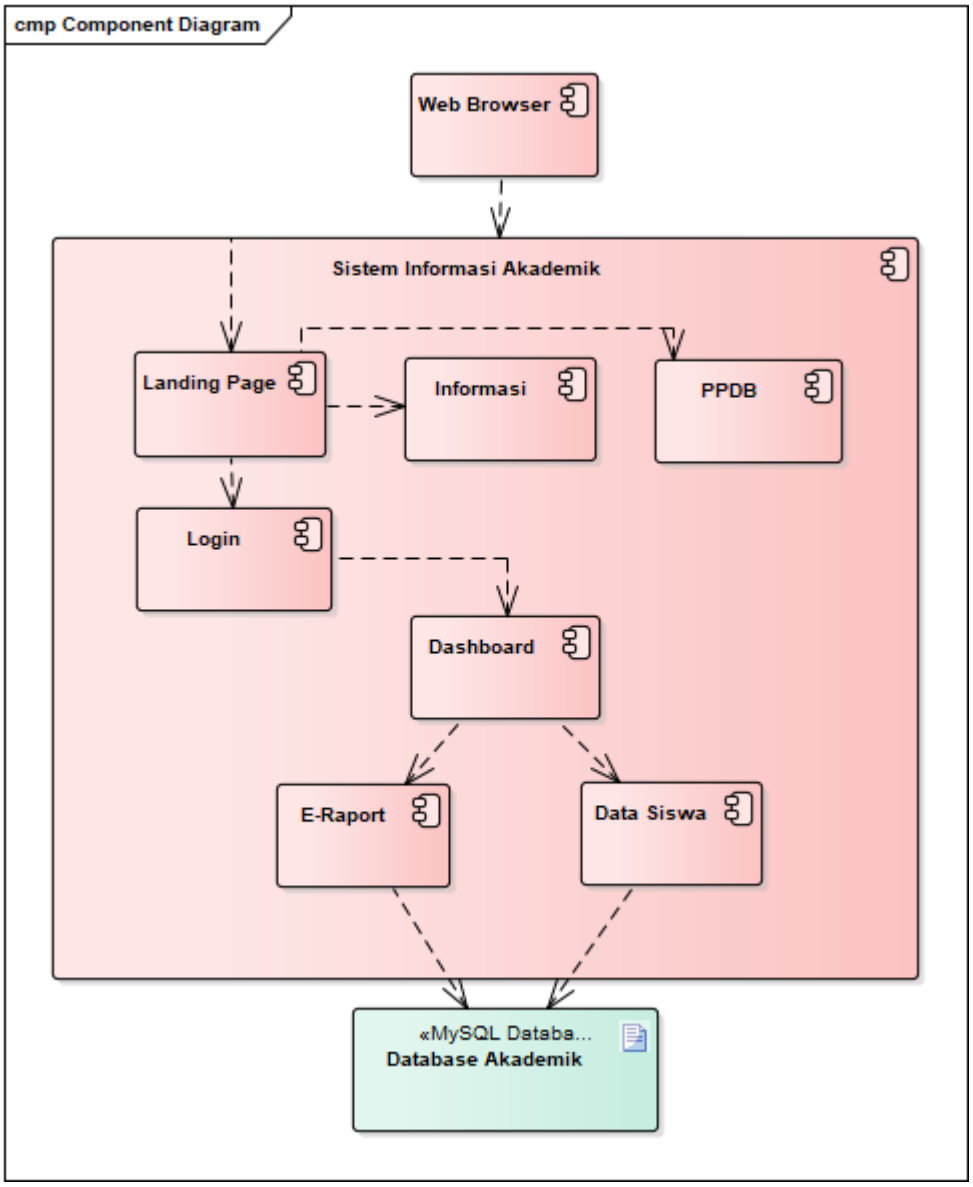


Gambar IV.12 *Sequence diagram* kelola informasi

Gambar IV.12 merupakan *Sequence Diagram* Kelola Informasi Sekolah yang menggambarkan proses pengelolaan informasi sekolah oleh Superadmin. *Sequence diagram* ini melibatkan Superadmin sebagai aktor, Halaman Kelola Informasi sebagai *boundary*, Informasi Controller dan Informasi Model sebagai *control*, serta tabel informasi sebagai *entity*. Pada modul ini Superadmin dapat melihat informasi yang telah dipublikasikan, menambahkan informasi baru, serta menghapus informasi yang sudah tidak diperlukan. Proses diawali ketika Superadmin membuka menu Kelola Informasi. Halaman Kelola Informasi kemudian mengirimkan permintaan kepada Informasi Controller untuk menampilkan seluruh data informasi yang tersedia. Selanjutnya controller memanggil Informasi Model untuk mengambil data dari tabel informasi pada database. Data yang berhasil diperoleh kemudian dikembalikan ke controller dan

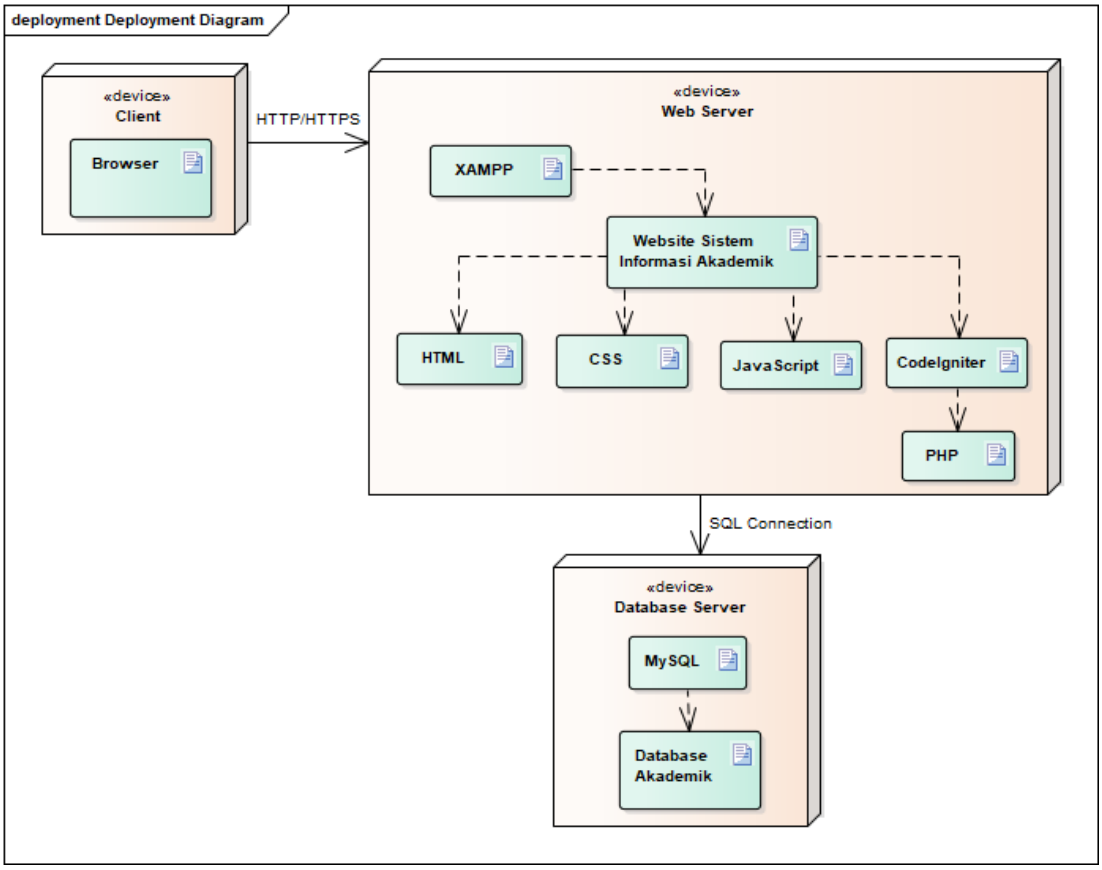
diteruskan ke halaman sehingga daftar informasi dapat ditampilkan kepada Superadmin. Untuk menambahkan informasi baru, Superadmin menekan tombol Tambah Informasi dan mengisi data berupa judul serta isi informasi. Setelah data diisi, halaman mengirimkan data tersebut kepada Informasi Controller untuk diproses. Controller kemudian meneruskan data ke Informasi Model untuk disimpan ke dalam tabel informasi. Setelah proses penyimpanan berhasil dilakukan, sistem menampilkan notifikasi informasi sukses ditambahkan dan data informasi yang baru akan muncul pada daftar informasi. Selain menambahkan informasi, Superadmin juga dapat menghapus informasi yang telah tersimpan. Proses dimulai ketika Superadmin memilih informasi yang akan dihapus dan menekan tombol Hapus. Halaman kemudian mengirimkan permintaan penghapusan kepada Informasi Controller. Controller meneruskan permintaan tersebut kepada Informasi Model untuk menghapus data dari tabel informasi berdasarkan identitas data yang dipilih. Setelah proses penghapusan berhasil dilakukan, sistem menampilkan pesan bahwa informasi berhasil dihapus dan daftar informasi akan diperbarui sesuai kondisi data terbaru. Setelah seluruh proses pengelolaan informasi selesai dilakukan, *sequence diagram* berakhir dan sistem menampilkan data informasi terkini sesuai perubahan yang dilakukan oleh Superadmin.

D. Component Diagram



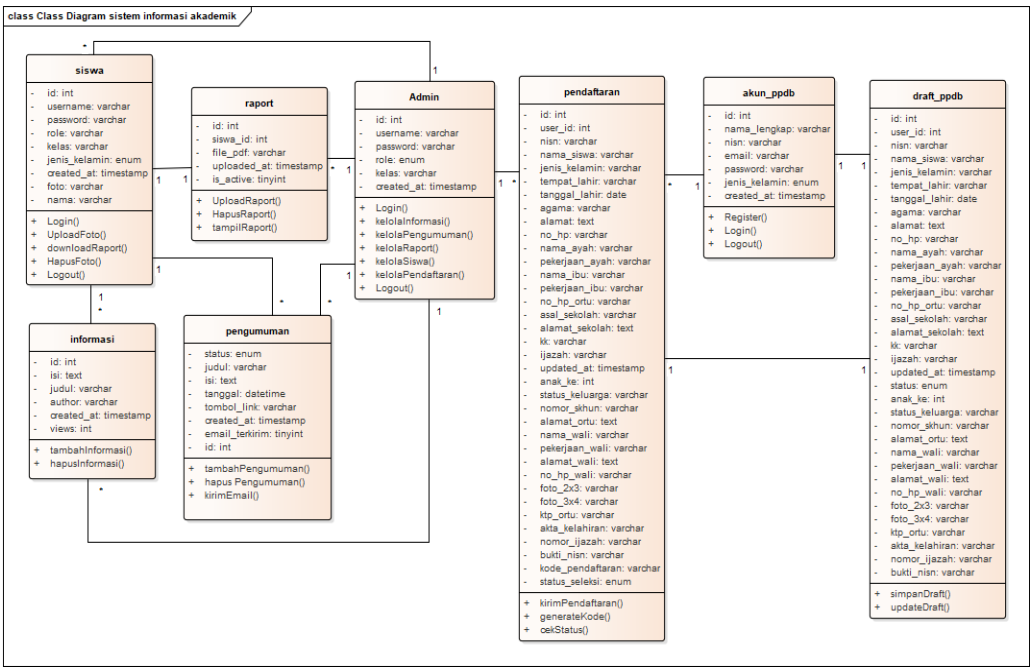
Gambar IV.13 *Component diagram*

E. Deployment Diagram



Gambar IV.14 *Deployment diagram*

F. *Class Diagram*



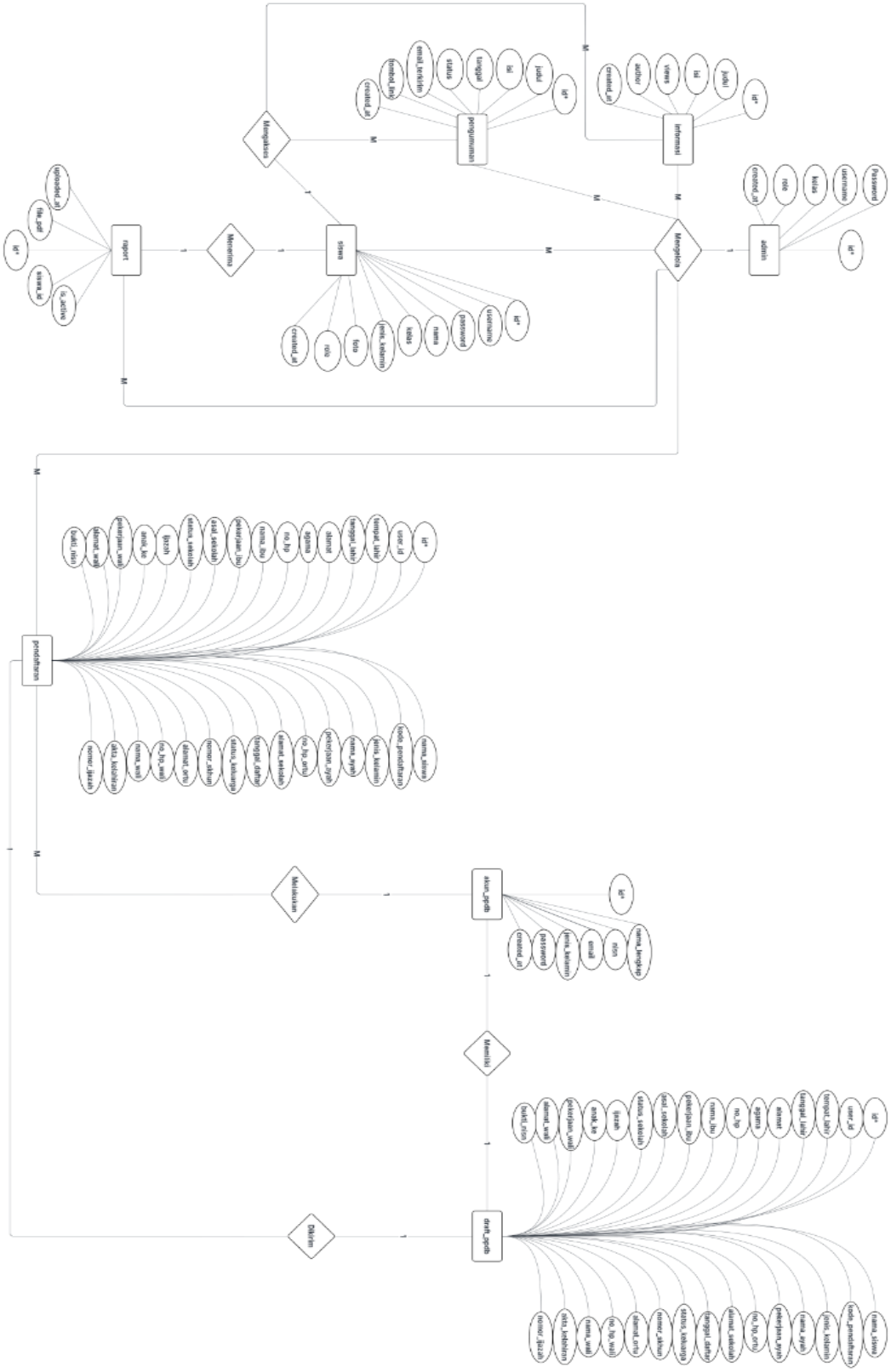
Gambar IV.15 *Class diagram*

Hubungan antar *class* pada Sistem Informasi Akademik menunjukkan keterkaitan data yang terdapat dalam sistem. *Class* akun_ppdb berhubungan dengan *class* draft_ppdb dengan kardinalitas (1:1), yang berarti satu akun PPDB dapat memiliki satu data draft pendaftaran yang aktif. Selanjutnya, *class* akun_ppdb juga berhubungan dengan *class* pendaftaran dengan kardinalitas (1: M), yang menunjukkan bahwa setiap akun PPDB dapat menghasilkan satu atau lebih data pendaftaran. Sementara itu, hubungan antara draft_ppdb dan pendaftaran juga bersifat (1:1) karena satu data draft hanya akan menghasilkan satu data pendaftaran. Selain itu, *class* Siswa memiliki hubungan dengan *class* raport dengan kardinalitas (1:1), yang berarti satu siswa hanya memiliki satu data raport yang diunggah. *Class* Siswa juga berhubungan dengan *class* Informasi dan Pengumuman dengan kardinalitas (1:M), yang menunjukkan bahwa seorang siswa dapat melihat banyak informasi dan banyak pengumuman yang tersedia

pada sistem. Selain itu, class Admin memiliki hubungan dengan beberapa *class* lainnya sebagai pengelola sistem. *Class* Admin berhubungan dengan *class* Pendaftaran dengan kardinalitas (1:M), yang berarti satu admin dapat mengelola banyak data pendaftaran. *Class* Admin juga berhubungan dengan *class* Siswa dengan kardinalitas (1:M), yang menunjukkan bahwa satu admin dapat mengelola banyak data siswa. Selanjutnya, *class* Admin berhubungan dengan *class* raport dengan kardinalitas (1:M) karena satu admin dapat mengunggah banyak file raport siswa. *Class* Admin juga berhubungan dengan *class* Informasi dan Pengumuman dengan kardinalitas (1:M), yang berarti satu admin dapat mengelola banyak informasi dan banyak pengumuman yang ditampilkan pada website.

4.2.2 Desain Pemodelan Data

1. *Entity Relationship Diagram* (ERD)



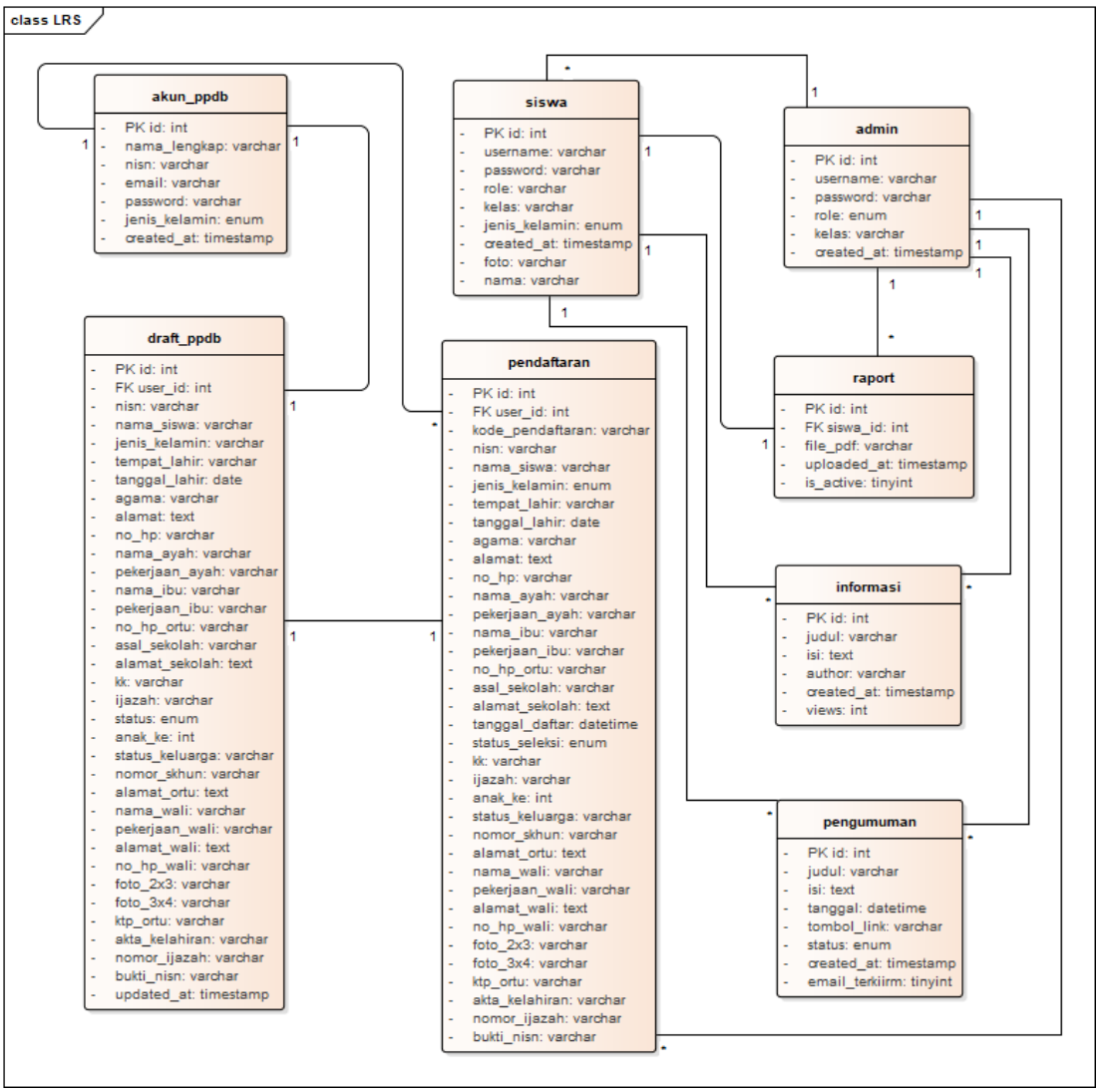
Gambar IV.16 *Entity Relantionship diagram* (ERD)

Berdasarkan *Entity Relationship Diagram* (ERD) yang telah dirancang, terdapat beberapa hubungan antar entitas pada sistem. Entitas admin berfungsi sebagai pengelola utama sistem. Admin memiliki hak akses untuk mengelola data siswa, data pendaftaran, data rapor, informasi, serta pengumuman. Hubungan tersebut bersifat one-to-many (1:M), yang berarti satu admin dapat mengelola banyak data pada masing-masing entitas tersebut. Entitas akun_ppdb digunakan sebagai akun yang dimiliki oleh calon peserta didik untuk mengakses sistem Penerimaan Peserta Didik Baru (PPDB). Setiap akun hanya memiliki satu data draft pendaftaran yang digunakan sebagai penyimpanan sementara selama proses pengisian formulir berlangsung. Mekanisme ini memungkinkan calon peserta didik melanjutkan pengisian formulir apabila proses sebelumnya belum selesai tanpa harus mengulangi pengisian data dari awal. Selama proses pengisian formulir, seluruh data akan disimpan pada entitas draft_ppdb. Data tersebut belum dianggap sebagai data pendaftaran yang sah hingga pengguna menekan tombol Kirim Pendaftaran. Setelah proses pengiriman dilakukan, data pada draft_ppdb akan dipindahkan ke entitas pendaftaran sebagai data pendaftaran resmi yang selanjutnya diproses oleh pihak sekolah. Hubungan antara akun_ppdb dan draft_ppdb bersifat one-to-one (1:1), yang menunjukkan bahwa setiap akun hanya memiliki satu draft aktif pada satu waktu. Sementara itu, hubungan antara draft_ppdb dan pendaftaran juga bersifat one-to-one (1:1) karena satu data draft hanya akan menghasilkan satu data pendaftaran. Adapun hubungan antara akun_ppdb dan pendaftaran bersifat one-to-many (1:M), yang berarti satu akun dapat melakukan lebih dari satu pendaftaran pada periode yang berbeda sesuai dengan ketentuan yang berlaku.sedangkan setiap data pendaftaran hanya berasal

dari satu akun PPDB. Adapun hubungan antara draft_ppdb dan akun_ppdb merupakan (1:1), yang berarti satu akun ppdb dapat memiliki 1 draft pendaftaran.

Entitas siswa merupakan data peserta didik yang telah diterima dan memiliki akses ke dalam sistem akademik. Melalui sistem tersebut, Siswa dapat mengakses berbagai informasi sekolah dan pengumuman yang telah dipublikasikan oleh pihak sekolah melalui website. Hubungan antara entitas siswa dengan entitas informasi dan pengumuman bersifat (1:M), yang berarti setiap siswa dapat melihat lebih dari satu informasi maupun pengumuman yang tersedia pada sistem. Selain itu, setiap siswa memiliki hubungan (1:1) dengan entitas rapor, yang menunjukkan bahwa satu siswa memiliki satu data rapor setiap semester.

2. *Logical Record Structure* (LRS)



Gambar IV.17 *Logical Record Structure* (LRS)

3. Spesifikasi File

a. Spesifikasi Tabel Siswa

Nama Database : ar_raudhah

Nama File : siswa

Akronim : tabel siswa

Tipe File : *File Master*

Panjang *Record* : 764 byte

Kunci *Field* : id

Tabel IV.2 Spesifikasi file tabel siswa

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Siswa	id	int	11	*Primary Key*
2.	Username	username	varchar	50	
3.	Password	password	varchar	255	
4.	Role	role	varchar	50	
5.	Kelas	kelas	varchar	50	
6.	Jenis Kelamin	jenis_kelamin	enum	-	
7.	Tanggal Dibuat	created_at	timestamp	-	
8.	Foto	foto	varchar	255	
9.	Nama	nama	varchar	100	

b. Spesifikasi tabel Raport

Nama Database : ar_raudhah

Nama File : raport

Akronim : tabel raport

Tipe File : *File Transaksi*

Panjang *Record* : 278 byte

Kunci *Field* : id

Tabel IV.3 Spesifikasi file tabel raport

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Raport	id	int	11	*Primary key*
2.	Id Siswa	siswa_id	int	11	*Foreign key*
3.	File Raport	file_pdf	varchar	255	
4.	Tanggal Upload	uploaded_at	timestamp	-	
5.	Status Aktif	is_active	tinyint	1	

c. Spesifikasi tabel Akun PPDB

Nama Database : ar_raudhah

Nama File : akun_ppdb

Akronim : tabel Akun PPDB

Tipe File : *File Master*

Panjang *Record* : 489 byte

Kunci *Field* : id

Tabel IV.4 Spesifikasi tabel akun PPDB

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Akun	id	int	11	*Primary Key*
2.	Nama Lengkap	nama_lengkap	varchar	100	
3.	NISN	nisn	varchar	30	
4.	Email	email	varchar	100	
5.	Password	password	varchar	255	
6.	Jenis kelamin	jenis_kelamin	enum	-	
7.	Tanggal Dibuatc	created_at	timestamp	-	

d. Spesifikasi tabel Draft PPDB

Nama Database : ar_raudhah

Nama File : draft_ppdb

Akronim : tabel Draft PPDB

Tipe File : *File Transaksi*

Panjang *Record* :1429 byte

Kunci *Field* : id

Tabel IV.5 Spesifikasi tabel draft PPDB

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Draft	id	int	11	*Primary key*
2.	Id User	user_id	int	11	*Foreign key*
3.	NISN	nisn	varchar	20	
4.	Nama Siswa	nama_siswa	varchar	100	
5.	Jenis Kelamin	jenis_kelamin	varchar	20	
6.	Tempat Lahir	tempat_lahir	varchar	100	
7.	Tanggal Lahir	tanggal_lahir	date	-	

No	Elemen Data	Nama Field	Type	Size	Keterangan
8.	Agama	agama	varchar	50	
9.	Alamat	alamat	text	-	
10.	No HP	no_hp	varchar	20	
11.	Nama Ayah	nama_ayah	varchar	100	
12.	Pekerjaan Ayah	pekerjaan_ayah	varchar	100	
13.	Nama Ibu	nama_ibu	varchar	100	
14.	Pekerjaan Ibu	pekerjaan_ibu	varchar	100	
15.	No Hp Ortu	no_hp_ortu	varchar	20	
16.	Asal Sekolah	asal_sekolah	varchar	150	
17.	Alamat Sekolah	alamat_sekolah	text	-	
18.	Kartu Keluarga	kk	varchar	255	
19.	Ijazah	ijazah	varchar	255	
20.	Status	status	enum	-	
21.	Updated At	uploaded_at	timestamp	-	
22.	Anak ke	anak_ke	int	2	
23.	Status Keluarga	status_keluarga	varchar	30	
24.	Nomor SKHUN	nomor_skhun	varchar	100	
25.	Alamat Orangtua	alamat_ortu	text	-	
26.	Nama Wali	nama_wali	varchar	150	
27.	Pekerjaan Wali	pekerjaan_wali	varchar	100	
28.	Alamat Wali	alamat_wali	text	-	
29.	Nomor HP Wali	no_hp_wali	varchar	20	
30.	Foto 2x3	foto_2x3	varchar	255	
31.	Foto 3x4	foto_3x4	varchar	255	
32.	KTP OrangTua	ktp_ortu	varchar	255	
33.	Akta Kelahiran	akta_kelahiran	varchar	255	

No	Elemen Data	Nama Field	Type	Size	Keterangan
34.	Nomor Ijazah	nomor_ijazah	varchar	100	
35.	Bukti NISN	bukti_nisn	varchar	255	

e. Spesifikasi tabel Informasi

Nama Database : ar_raudhah

Nama File : informasi

Akronim : tabel Informasi

Tipe File : *File Master*

Panjang *Record* : 363 byte

Kunci *Field* : id

Tabel IV.6 Spesifikasi tabel informasi

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Informasi	id	Int	11	*Primary key*
2.	Judul	judul	varchar	255	
3.	Isi Informasi	isi	text	-	
4.	Author	author	varchar	100	
5.	Tanggal Dibuat	created_at	timestamp	-	
6.	Jumlah Dilihat	views	int	11	

f. Spesifikasi tabel Pengumuman

Nama Database : ar_raudhah

Nama File : pengumuman

Akronim : tabel Pengumuman

Tipe File : *File Master*

Panjang *Record* : 515 byte

Kunci *Field* : id

Tabel IV.7 Spesifikasi tabel pengumuman

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Pengumuman	id	int	11	*Primary key*
2.	Judul	judul	varchar	255	
3.	Isi Pengumuman	isi	text	-	
4.	Tanggal	tanggal	datetime	-	
5.	Tombol link	tombol_link	varchar	255	
6.	Status	status	enum	-	
7.	Created at	created_at	timestamp	-	
8.	Email Terkirim	email_terkirim	tinyint	1	

g. Spesifikasi tabel Admin

Nama Database : ar_raudhah

Nama File : admin

Akronim : tabel Admin

Tipe File : *File Master*

Panjang *Record* : 359 byte

Kunci *Field* : id

Tabel IV.8 Spesifikasi tabel admin

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Admin	id	int	11	*Primary key*
2.	Username	username	varchar	50	
3.	Password	password	varchar	255	
4.	Role	role	enum	-	
5.	Kelas	kelas	varchar	50	
6.	Created at	created_at	timestamp	-	

h. Spesifikasi tabel Pendaftaran

Nama Database : ar_raudhah

Nama File : pendaftaran

Akronim : tabel Pendaftaran

Tipe File : *File Transaksi*

Panjang *Record* : 1459 byte

Kunci *Field* : id

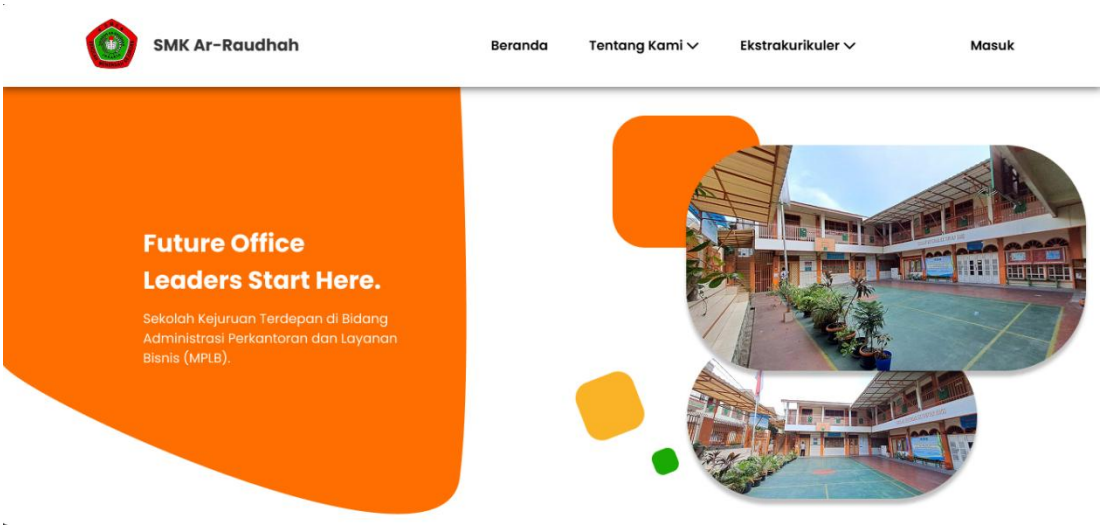
Tabel IV.9 Spesifikasi tabel pendaftaran

No	Elemen Data	Nama Field	Type	Size	Keterangan
1.	Id Pendaftaran	id	int	11	*Primary key*
2.	Id User	user_id	int	11	*Foreign key*
3.	NISN	nisn	varchar	20	
4.	Nama Siswa	nama_siswa	varchar	100	
5.	Jenis Kelamin	jenis_kelamin	varchar	20	
6.	Tempat Lahir	tempat_lahir	varchar	100	
7.	Tanggal Lahir	tanggal_lahir	date	-	
8.	Agama	agama	varchar	50	
9.	Alamat	alamat	text	-	
10.	No HP	no_hp	varchar	20	

No	Elemen Data	Nama Field	Type	Size	Keterangan
11.	Nama Ayah	nama_ayah	varchar	100	
12.	Pekerjaan Ayah	pekerjaan_ayah	varchar	100	
13.	Nama Ibu	nama_ibu	varchar	100	
14.	Pekerjaan Ibu	pekerjaan_ibu	varchar	100	
15.	No Hp Ortu	no_hp_ortu	varchar	20	
16.	Asal Sekolah	asal_sekolah	varchar	150	
17.	Alamat Sekolah	alamat_sekolah	text	-	
18.	Kartu Keluarga	kk	varchar	255	
19.	Ijazah	ijazah	varchar	255	
20.	Status Seleksi	status_seleksi	enum	-	
21.	Kode Pendaftaran	kode_pendaftaran	varchar	30	
22.	Anak ke	anak_ke	int	2	
23.	Status Keluarga	status_keluarga	varchar	30	
24.	Nomor SKHUN	nomor_skhun	varchar	100	
25.	Alamat Orangtua	alamat_ortu	text	-	
26.	Nama Wali	nama_wali	varchar	150	
27.	Pekerjaan Wali	pekerjaan_wali	varchar	100	
28.	Alamat Wali	alamat_wali	text	-	
29.	Nomor HP Wali	no_hp_wali	varchar	20	
30.	Foto 2x3	foto_2x3	varchar	255	
31.	Foto 3x4	foto_3x4	varchar	255	
32.	KTP OrangTua	ktp_ortu	varchar	255	
33.	Akta Kelahiran	akta_kelahiran	varchar	255	
34.	Nomor Ijazah	nomor_ijazah	varchar	100	
35.	Bukti NISN	bukti_nisn	varchar	255	
36.	Tanggal Daftar	tanggal_daftar	datetime	-	

4.2.3 Desain *User Interface*

a. Tampilan *Landing Page*



Gambar IV.18 Tampilan *landing page*

b. Tampilan Halaman Login





Gambar IV.19 Tampilan halaman login

c. Tampilan Halaman Informasi



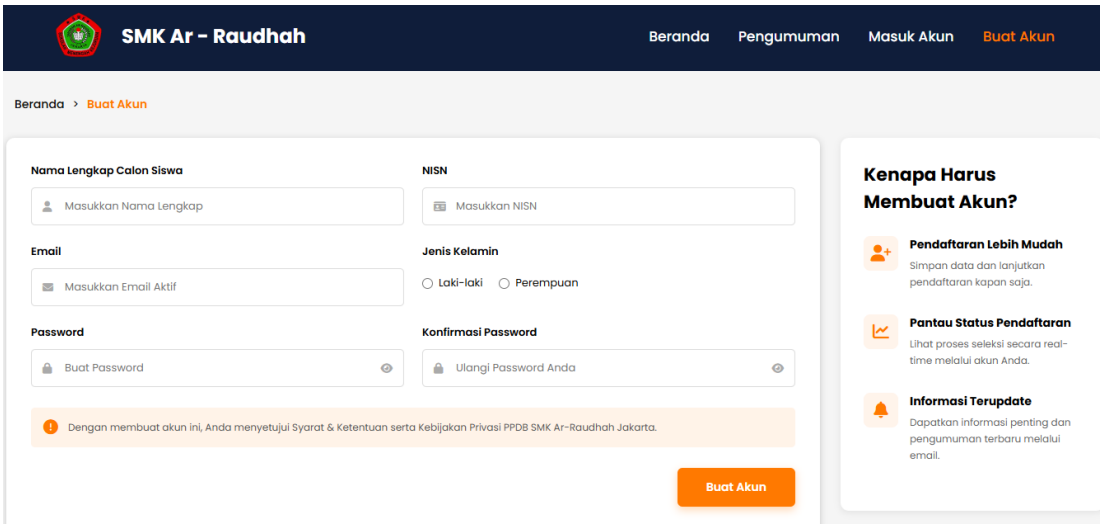
Gambar IV.20 Tampilan halaman informasi

d. Tampilan Halaman Dashboard PPDB



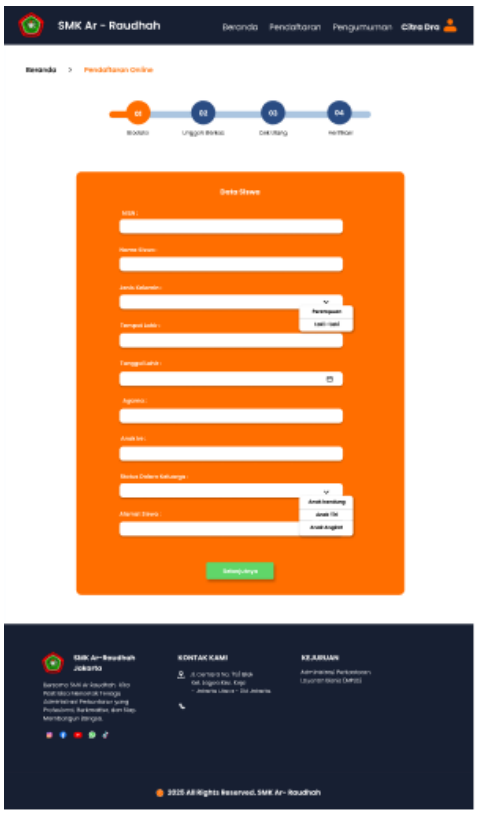
Gambar IV.21 Tampilan halaman dashboard PPDB

e. Tampilan Halaman Buat akun PPDB



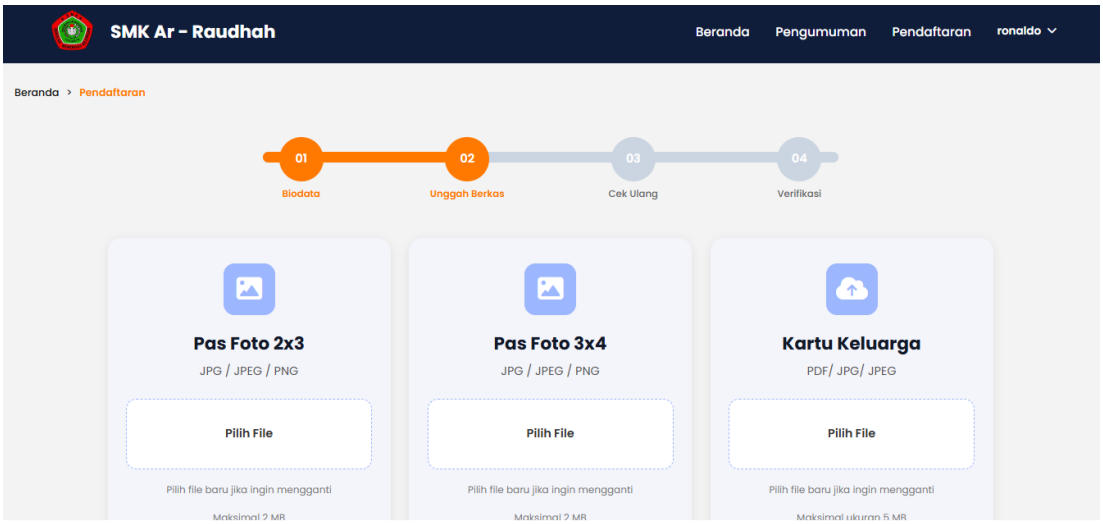
Gambar IV.22 Tampilan halaman buat akun PPDB

f. Tampilan Halaman Pendaftaran



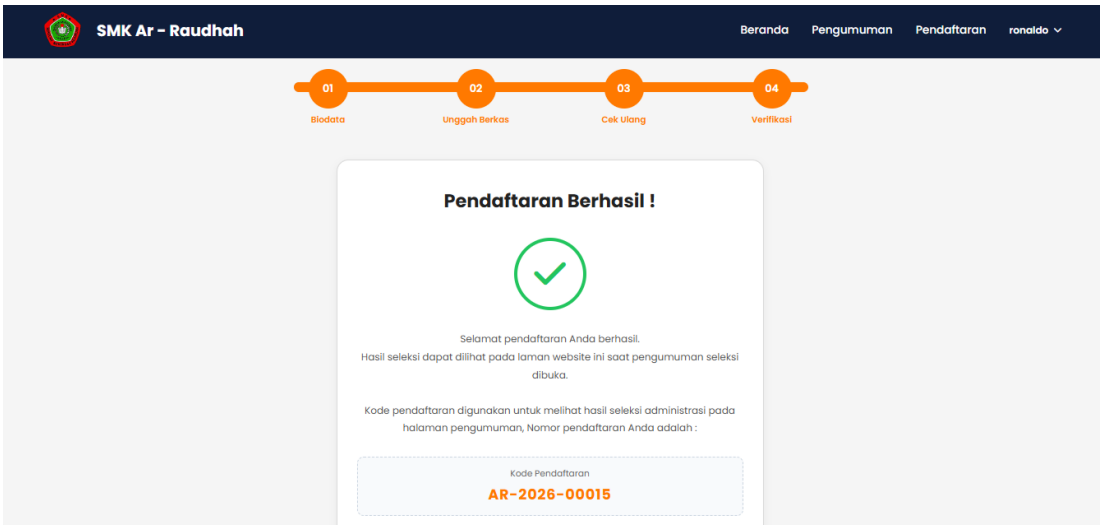
Gambar IV.23 Tampilan halaman pendaftaran

g. Tampilan Halaman Pendaftaran Unggah berkas



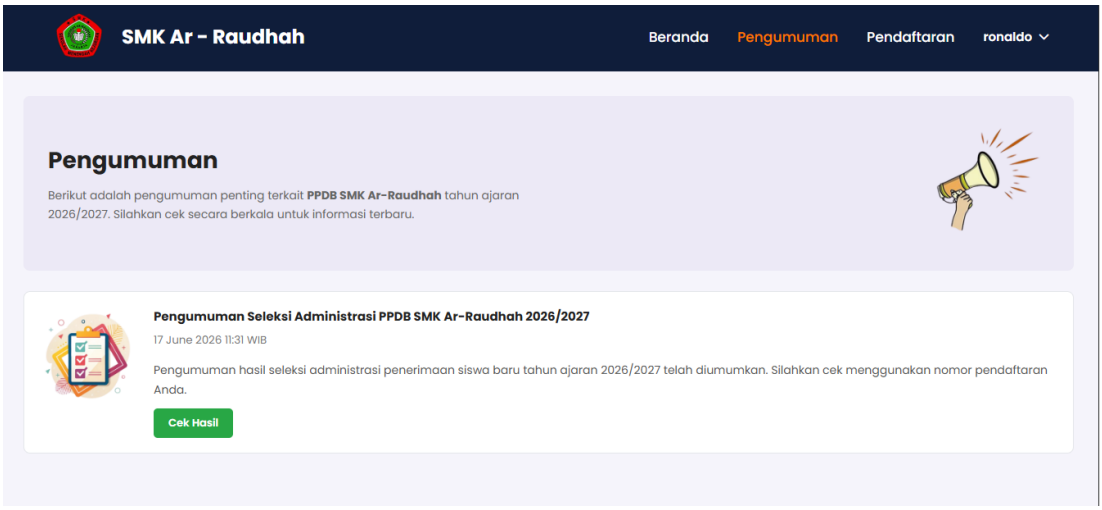
Gambar IV.24 Tampilan halaman unggah berkas

h. Tampilan Halaman Pendaftaran Berhasil



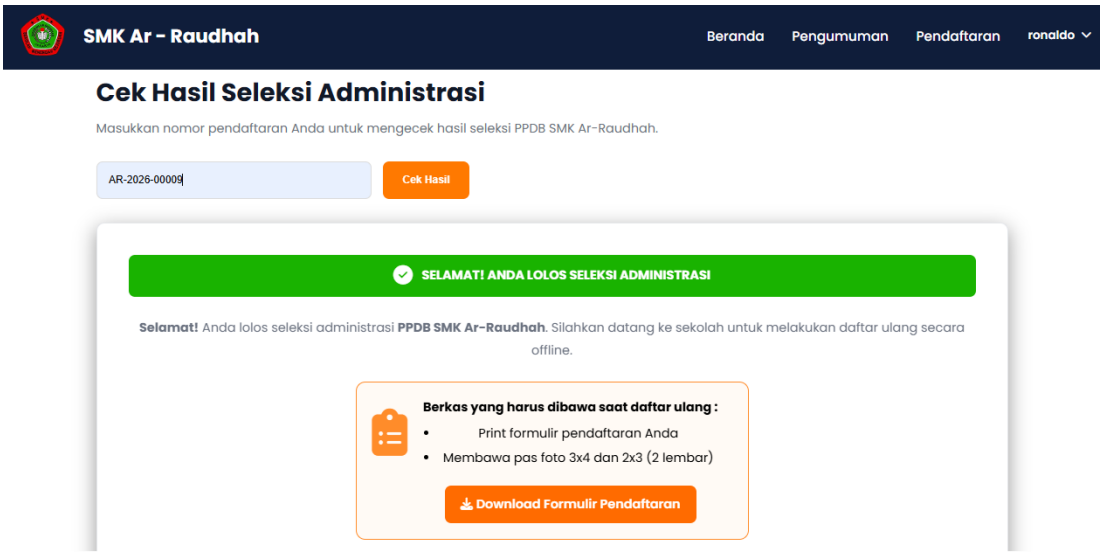
Gambar IV.25 Tampilan halaman pendaftaran berhasil

i. Tampilan Halaman Pengumuman PPDB



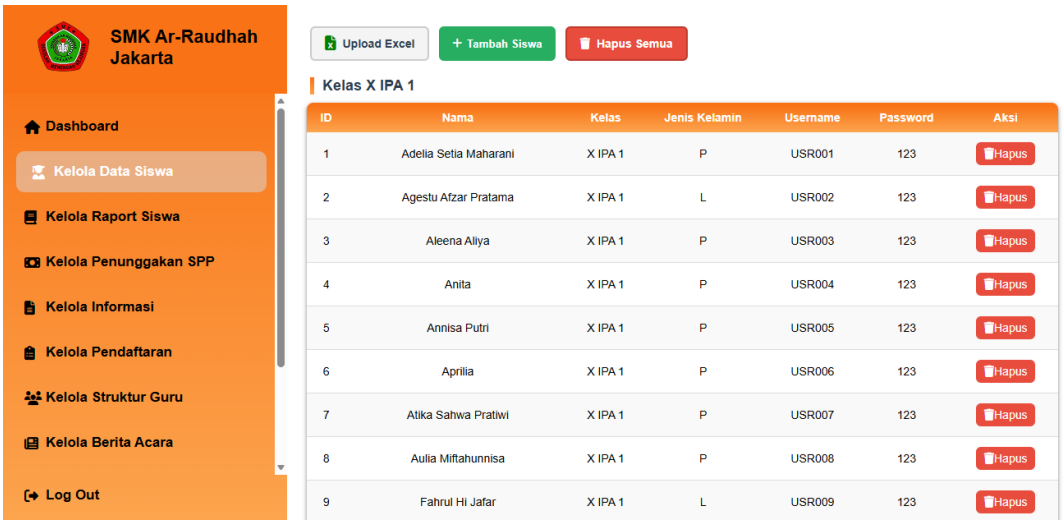
Gambar IV.26 Tampilan halaman pengumuman PPDB

j. Tampilan Halaman Cek Hasil Seleksi PPDB



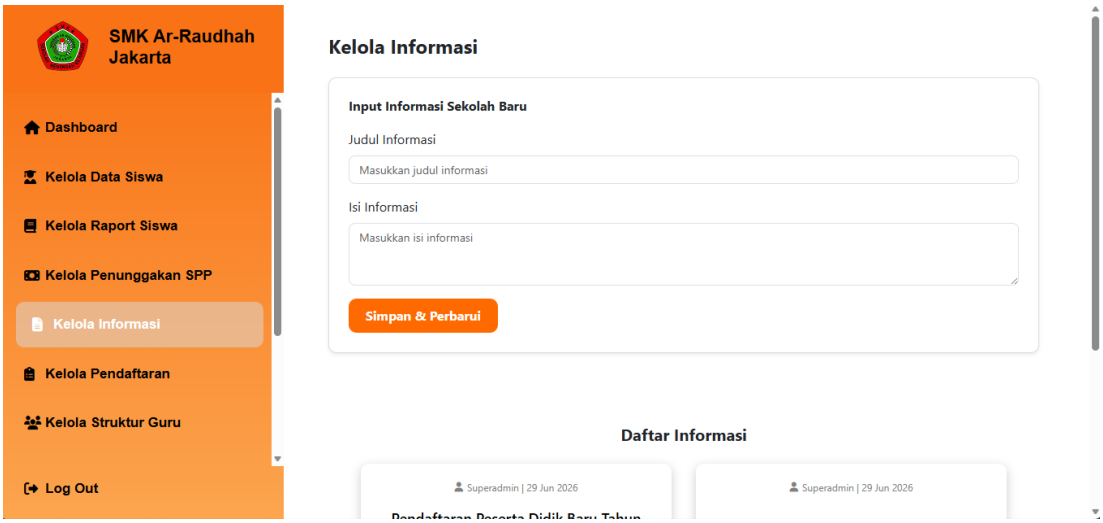
Gambar IV.27 Tampilan halaman cek hasil seleksi PPDB

k. Tampilan Halaman Kelola Data Siswa (Superadmin)



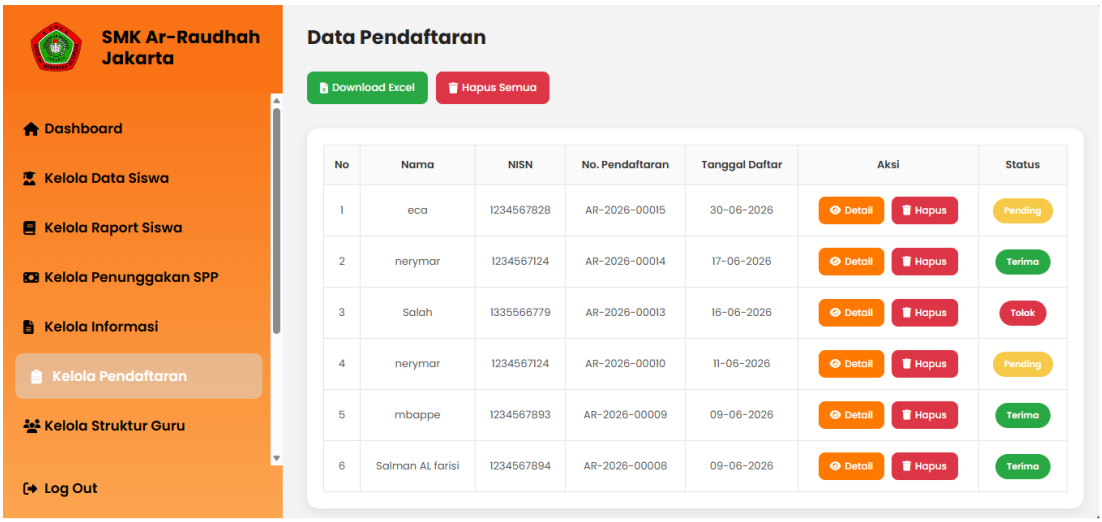
Gambar IV.28 Halaman kelola data siswa

l. Tampilan Halaman Kelola Informasi (Superadmin)



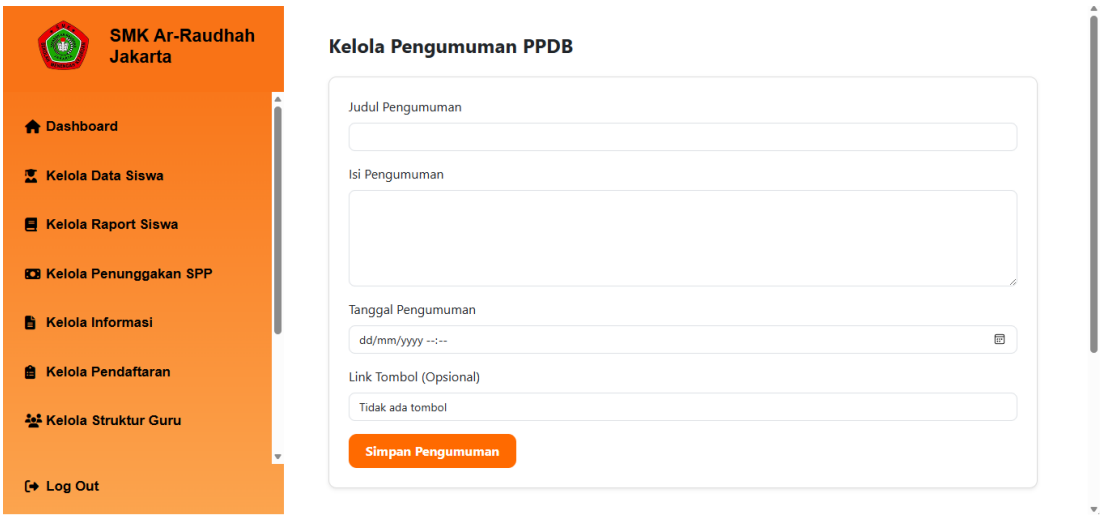
Gambar IV.29 Tampilan halaman kelola informasi

m. Tampilan Halaman Kelola Pendaftaran (Superadmin)



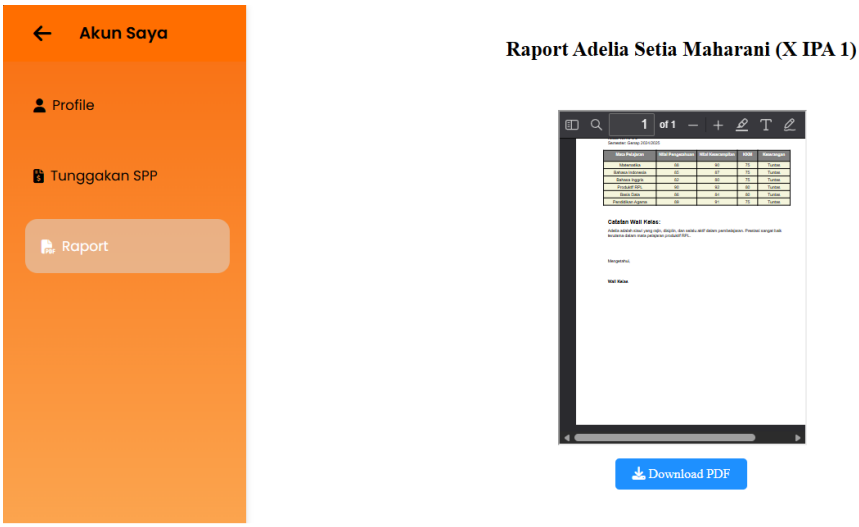
Gambar IV.30 Tampilan halaman kelola pendaftaran

n. Tampilan Halaman Kelola Pengumuman PPDB (Superadmin)



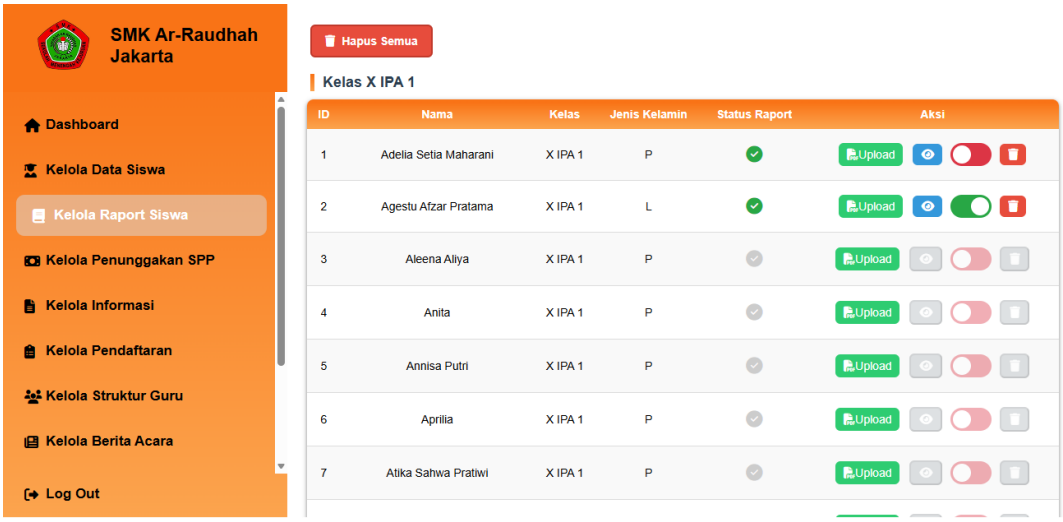
Gambar IV.31 Tampilan halaman kelola pengumuman PPDB

o. Tampilan Halaman Raport (Siswa)



Gambar IV.32 Tampilan halaman raport

p. Tampilan Halaman Kelola *Raport (*Superadmin)



Gambar IV.33 Tampilan halaman kelola raport

4.3 Code Generation

a. Halaman Landing Page

```
<!-- Hero Section -->
<section class="hero">
    <!-- Kiri -->
    <div class="hero-left">
        <div class="images">
            <img src="<?= base_url('assets/img/herosection1.png') ?>"
alt="foto Sekolah">
        </div>
    </div>

    <!-- Kanan -->
    <div class="hero-right">

        <div class="images" data-aos="zoom-in-up" data-aos-
duration="1500">
            <img src="<?= base_url('assets/img/sekolah.png') ?>"
alt="foto Sekolah">
        </div>
    </div>
</section>
```

b. Halaman Login

```
class Auth extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
        $this->load->model('Auth_model');
        $this->load->library('session');
        $this->load->helper(['url', 'form']);
    }

    public function login()
    {

        if ($this->session->userdata('logged_in')) {
            redirect('admin/dashboard');
        }

        if ($this->input->post()) {
```

```
            $username = $this->input->post('username');
            $password = $this->input->post('password');

            $user = $this->Auth_model->login($username, $password);

            if ($user) {

                // SET SESSION LOGIN
                $this->session->set_userdata([
                    'logged_in' => TRUE,
                    'username'  => $user->username,
                    'nama'      => isset($user->nama) ? $user->nama :
$user->username,
                    'role'      => $user->role,
                    'user'      => $user
                ]);


                if ($user->role == 'siswa') {
                    $this->session->set_flashdata('success', 'Selamat
Datang Kembali ' . $user->nama);
                    redirect('home');
                } else {
                    $this->session->set_flashdata('success', 'Selamat
Datang Kembali ' . $user->username);
                    redirect('admin/dashboard');
                }
                // Arahkan sesuai role
                if ($user->role == 'superadmin') {
                    redirect('admin/dashboard');
                } elseif ($user->role == 'admin') {
                    redirect('admin/dashboard');
                } elseif ($user->role == 'siswa') {
                    redirect('home');
                }
            } else {
                // Jika username/password salah
                $this->session->set_flashdata('error', 'Username atau
Password salah!');
                redirect('auth/login');
            }
        }

        $this->load->view('auth/login');
    }
```

c. Halaman Informasi

```php
<?php
class Informasi_model extends CI_Model
{

    protected $table = 'informasi';

    public function getAll()
    {
        $this->db->order_by('created_at', 'DESC');
        return $this->db->get('informasi')->result();
    }

    public function getById($id)
    {
        return $this->db->get_where('informasi', ['id' => $id])->row();
    }

    public function insert($data)
    {
        return $this->db->insert('informasi', $data);
    }

    public function incrementViews($id)
    {
        $this->db->set('views', 'views+1', FALSE);
        $this->db->where('id', $id);
        $this->db->update('informasi');
    }

    public function delete($id)
    {
        return $this->db->where('id', $id)->delete($this->table);
    }
}
```

d. Halaman Dashboard PPDB

```php
<body>

    <?php $this->load->view('layouts/navbar_ppdb'); ?>


    <!-- HERO -->

    <section class="hero">
```

```
        <div class="back-home">
            <a href="<?= base_url('home'); ?>" class="btn-back">
                <i class="fas fa-arrow-left"></i>
                <span>Kembali ke Beranda Utama</span>
            </a>
        </div>
        <div class="hero-content">
            <h1>
                SELAMAT DATANG DI PPDB <br>
                SMK AR-RAUDHAH <br>
                JAKARTA <br>
                TAHUN AJARAN 2026/2027
            </h1>
        </div>
    </section>

    <!-- ALUR -->

    <section class="section">
        <div class="section-title">Alur Pendaftaran</div>

        <div class="alur-container">
            <div class="alur-line"></div>

            <div class="alur-item">
                <div class="alur-number">1</div>
                <div class="alur-icon">
                    <i class="fas fa-user-plus"></i>
                </div>
                <h3>Buat Akun</h3>
                <p>
                    Buat akun PPDB menggunakan email aktif.
                </p>
            </div>

            <div class="alur-item">
                <div class="alur-number">2</div>
                <div class="alur-icon">
                    <i class="fas fa-file-signature"></i>
                </div>
                <h3>Lengkapi Formulir</h3>
                <p>
                    Lengkapi data diri peserta beserta dokumen
pendukung.
                </p>
            </div>
```

```
            <div class="alur-item">
                <div class="alur-number">3</div>
                <div class="alur-icon">
                    <i class="fas fa-clipboard-check"></i>
                </div>
                <h3>Seleksi</h3>
                <p>
                    Verifikasi data dan proses seleksi administrasi.
                </p>
            </div>

            <div class="alur-item">
                <div class="alur-number">4</div>
                <div class="alur-icon">
                    <i class="fas fa-bullhorn"></i>
                </div>
                <h3>Pengumuman</h3>
                <p>
                    Cek hasil pengumuman kelulusan secara online.
                </p>
            </div>
        </div>

        <!-- INFO -->

        <div class="info-wrapper">

            <div class="info-card">
                <h3>
                    <i class="fas fa-calendar-alt"></i>
                    Jadwal PPDB
                </h3>

                <div class="jadwal-item">
                    <span>Pendaftaran Online</span>
                    <span>1 Juni - 5 Juli 2026</span>
                </div>

                <div class="jadwal-item">
                    <span>Seleksi & Verifikasi</span>
                    <span>10 Juni - 22 Juni 2026</span>
                </div>

                <div class="jadwal-item">
                    <span>Pengumuman Hasil</span>
                    <span>30 Juli 2026</span>
                </div>
            </div>
```

```
            <div class="info-card">
                <h3>
                    <i class="fas fa-folder-open"></i>
                    Persyaratan
                </h3>

                <ul>
                    <li>Upload ijazah/STL</li>
                    <li>Upload Kartu Keluarga</li>
                    <li>Upload KTP wali</li>
                    <li>Upload NISN</li>
                    <li>Upload Akte Kelahiran</li>
                </ul>
            </div>

            <div class="info-card">
                <h3>
                    <i class="fas fa-info-circle"></i>
                    Informasi Penting
                </h3>

                <ul>
                    <li>Semua proses pendaftaran dilakukan secara
online melalui website.</li>
                    <li>jika dinyatakan lolos seleksi administrasi,
lakukan daftar ulang secara offline di sekolah sesuai jadwal.</li>
                    <li>Gunakan data yang valid dan dapat dipertanggung
jawabkan.</li>
                </ul>
            </div>

        </div>
    </section>

    <!-- LANGKAH -->

    <section class="section">
        <div class="section-title">Langkah - Langkah PPDB</div>

        <div class="langkah-box">
            <div class="langkah-wrapper">

                <div class="langkah-item">
                    <div class="langkah-circle">1</div>
                    <h4>Pendaftaran Akun</h4>
                    <p>
```

```
                        Calon siswa membuat akun untuk bisa memulai
pendaftaran.
                    </p>
                </div>

                <div class="arrow">→</div>

                <div class="langkah-item">
                    <div class="langkah-circle">2</div>
                    <h4>Pendaftaran</h4>
                    <p>
                        Calon siswa melengkapi
                        formulir pendaftaran dan
                        unggah berkas yang
                        dibutuhkan melalui
                        website ini.
                    </p>
                </div>

                <div class="arrow">→</div>


                <div class="langkah-item">
                    <div class="langkah-circle">3</div>
                    <h4>Verifikasi Administrasi</h4>
                    <p>
                        Panitia akan memverifikasi
                        data dan berkas pendaftaran
                        Anda secara online.
                    </p>
                </div>

                <div class="arrow">→</div>

                <div class="langkah-item">
                    <div class="langkah-circle">4</div>
                    <h4>Pengumuman Hasil</h4>
                    <p>
                        Cek hasil seleksi administrasi
                        melalui menu pengumuman
                        (lolos / tidak lolos).
                    </p>
                </div>

                <div class="arrow">→</div>

                <div class="langkah-item">
                    <div class="langkah-circle">5</div>
```

```
                    <h4>Daftar Ulang (Offline)</h4>
                    <p>
                        Bagi yang lolos, datang ke
                        sekolah untuk daftar ulang
                        dengan membawa berkas
                        persyaratan asli/ print out.
                    </p>
                </div>
            </div>

            <div class="catatan">
                <strong>* Catatan:</strong>
                Pastikan data dan berkas yang diunggah benar. Kesalahan
data dapat menyebabkan proses verifikasi gagal.
            </div>
        </div>
    </section>

    <!-- FOOTER -->

    <footer>
        <?php $this->load->view('layouts/footer'); ?>
    </footer>



    <?php if ($this->session->flashdata('success')): ?>
        <script
src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>

        <script>
            const successMessage = "<?=
$this->session->flashdata('success'); ?>";
            const namaUser = "<?=
$this->session->userdata('ppdb_nama'); ?>";

            Swal.fire({
                icon: 'success',
                title: 'Login Berhasil',
                text: 'Selamat datang kembali, ' + namaUser,
                showConfirmButton: false,
                timer: 2500,
                timerProgressBar: true,
                customClass: {
                    popup: 'swal-custom-popup',
                    title: 'swal-custom-title',
                    htmlContainer: 'swal-custom-text'
                }
```

```
            });
        </script>
    <?php endif; ?>

    <script>
        const toggle = document.getElementById('dropdownToggle');
        const menu = document.getElementById('dropdownMenu');

        toggle.addEventListener('click', function() {

            menu.classList.toggle('show');

            toggle.classList.toggle('active');

        });

        // klik luar dropdown = tutup
        document.addEventListener('click', function(e) {

            if (!document.querySelector('.dropdown-
user').contains(e.target)) {

                menu.classList.remove('show');

                toggle.classList.remove('active');
            }

        });
    </script>

</body>
```

e. Halaman Buat Akun

```
 public function register_ppdb()
    {
        $this->load->view('ppdb/register');
    }
    public function proses_register()
    {
        $password = $this->input->post('password');
        $konfirmasi = $this->input->post('konfirmasi_password');
        $nisn = $this->input->post('nisn');

        // cek NISN
        $cek_nisn = $this->db
            ->where('nisn', $nisn)
```

```
            ->get('akun_ppdb')
            ->row();

        if ($cek_nisn) {

            $this->session->set_flashdata(
                'nisn_exist',
                'NISN sudah terdaftar.'
            );

            redirect('ppdb/register');
        }

        // cek password
        if (
            !preg_match(
                '/^(?=.*[a-z])(?=.*[A-Z])(?=.*\d).{8,}$/',
                $password
            )
        ) {

            $this->session->set_flashdata(
                'error',
                'Password harus mengandung huruf besar, huruf kecil dan
angka'
            );

            redirect('ppdb/register');
        }

        // cek email sudah terdaftar atau belum
        $email = $this->input->post('email');

        $cek_email = $this->db
            ->where('email', $email)
            ->get('akun_ppdb')
            ->row();

        if ($cek_email) {

            $this->session->set_flashdata(
                'email_exist',
                'Email sudah terdaftar. Silakan gunakan email lain atau
login.'
            );

            redirect('ppdb/register');
        }
```

```
        $data = [
            'nama_lengkap' => $this->input->post('nama_lengkap'),
            'nisn' => $this->input->post('nisn'),
            'email' => $this->input->post('email'),
            'jenis_kelamin' => $this->input->post('jenis_kelamin'),
            'password' => password_hash($password, PASSWORD_DEFAULT)
        ];

        $this->db->insert('akun_ppdb', $data);

        $user_id = $this->db->insert_id();

        // session login
        $this->session->set_userdata([
            'ppdb_login' => true,
            'ppdb_id'    => $user_id,
            'ppdb_nama'  => $data['nama_lengkap'],
            'ppdb_email' => $data['email']
        ]);

        redirect('ppdb/register-success');
    }
```

f. Halaman Pendaftaran

```
public function pendaftaran()
    {
        $this->cekLogin();
        $data['draft'] = $this->getDraft();

        $this->load->view(
            'ppdb/pendaftaran',
            $data
        );
    }

    public function pendaftaran_orangtua()
    {
        $this->cekLogin();
        if ($this->input->post()) {

            $this->simpanDraft(
                $this->input->post()
            );
        }
```

```
        $data['draft'] = $this->getDraft();

        $this->load->view(
            'ppdb/pendaftaran_orangtua',
            $data
        );
    }

    public function pendaftaran_sekolah()
    {
        $this->cekLogin();
        if ($this->input->post()) {

            $this->simpanDraft(
                $this->input->post()
            );
        }
        $data['draft'] = $this->getDraft();

        $this->load->view(
            'ppdb/pendaftaran_sekolah',
            $data
        );
    }
```

g. Halaman Pendaftaran Unggah Berkas

```
public function upload_berkas_view()
    {
        $this->cekLogin();
        if ($this->input->post()) {

            $post = $this->input->post();

            $post['user_id'] =
                $this->session->userdata('ppdb_id');

            $cek = $this->db
                ->where(
                    'user_id',
                    $post['user_id']
                )
                ->get('draft_ppdb')
                ->row();

            if ($cek) {
```

```
                $this->db
                    ->where(
                        'user_id',
                        $post['user_id']
                    )
                    ->update(
                        'draft_ppdb',
                        $post
                    );
            } else {

                $this->db->insert(
                    'draft_ppdb',
                    $post
                );
            }
        }

        $data['draft'] = $this->getDraft();

        $this->load->view(
            'ppdb/upload_berkas',
            $data
        );
    }

    public function upload_berkas()
    {
        $this->cekLogin();
        $draft = $this->getDraft();

        if (
            empty($draft['foto_2x3']) ||
            empty($draft['foto_3x4']) ||
            empty($draft['kk']) ||
            empty($draft['ktp_ortu']) ||
            empty($draft['akta_kelahiran']) ||
            empty($draft['bukti_nisn']) ||
            empty($draft['ijazah'])
        ) {

            $this->session->set_flashdata(
                'error',
                'Silakan upload semua berkas terlebih dahulu.'
            );

            redirect('ppdb/upload-berkas-view');
        }
```

```
        redirect('ppdb/cek-ulang');
    }
```

h. Halaman Pendaftaran Berhasil

```
public function kirim_pendaftaran()
    {
        $this->cekLogin();
        $draft = $this->getDraft();

        if (!$draft) {

            redirect('ppdb/pendaftaran');
        }
        $data = [

            'user_id' => $this->session->userdata('ppdb_id'),


            'nisn'            => $draft['nisn'],
            'nama_siswa'      => $draft['nama_siswa'],
            'jenis_kelamin'   => $draft['jenis_kelamin'],
            'tempat_lahir'    => $draft['tempat_lahir'],
            'tanggal_lahir'   => $draft['tanggal_lahir'],
            'agama'           => $draft['agama'],
            'alamat'          => $draft['alamat'],
            'no_hp'           => $draft['no_hp'],
            'anak_ke'         => $draft['anak_ke'],
            'status_keluarga' => $draft['status_keluarga'],

            'nama_ayah'       => $draft['nama_ayah'],
            'pekerjaan_ayah'  => $draft['pekerjaan_ayah'],

            'nama_ibu'        => $draft['nama_ibu'],
            'pekerjaan_ibu'   => $draft['pekerjaan_ibu'],

            'no_hp_ortu'      => $draft['no_hp_ortu'],

            'alamat_ortu'     => $draft['alamat_ortu'],

            'nama_wali'       => $draft['nama_wali'],
            'pekerjaan_wali'  => $draft['pekerjaan_wali'],
            'alamat_wali'     => $draft['alamat_wali'],
            'no_hp_wali'      => $draft['no_hp_wali'],

            'asal_sekolah'    => $draft['asal_sekolah'],
            'alamat_sekolah'  => $draft['alamat_sekolah'],
```

```
            'nomor_ijazah'    => $draft['nomor_ijazah'],
            'nomor_skhun'     => $draft['nomor_skhun'],

            'foto_2x3'        => $draft['foto_2x3'],
            'foto_3x4'        => $draft['foto_3x4'],
            'kk'              => $draft['kk'],
            'ktp_ortu'        => $draft['ktp_ortu'],
            'akta_kelahiran'  => $draft['akta_kelahiran'],
            'ijazah'          => $draft['ijazah'],
            'bukti_nisn'      => $draft['bukti_nisn'],
        ];
        // Generate kode pendaftaran
        // Simpan dulu data
        $data['kode_pendaftaran'] = '';

        $this->db->insert('pendaftaran', $data);

        $id = $this->db->insert_id();

        $kode = 'AR-' .
            date('Y') .
            '-' .
            str_pad(
                $id,
                5,
                '0',
                STR_PAD_LEFT
            );

        $this->db
            ->where('id', $id)
            ->update(
                'pendaftaran',
                [
                    'kode_pendaftaran' => $kode
                ]
            );

        $this->db
            ->where(
                'user_id',
                $this->session->userdata('ppdb_id')
            )
            ->delete('draft_ppdb');

        $this->session->set_userdata(
            'kode_pendaftaran',
            $kode
```

```
        );

        redirect('ppdb/pendaftaran-kirim');
    }

    public function pendaftaran_kirim()
    {
        $data['kode'] = $this->session->userdata(
            'kode_pendaftaran'
        );

        $this->load->view(
            'ppdb/pendaftaran_kirim',
            $data
        );
    }
```

i. Halaman Pengumuman PPDB

```
public function pengumuman()
    {
        $data['pengumuman'] = $this->db
            ->where('tanggal <=', date('Y-m-d H:i:s'))
            ->order_by('tanggal', 'DESC')
            ->get('pengumuman')
            ->result();

        $this->load->view(
            'ppdb/pengumuman',
            $data
        );
    }
```

j. Halaman Cek Hasil Seleksi PPDB

```
  public function cek_hasil()
    {
        $data['hasil'] = null;
        $data['not_found'] = false;

        $this->load->view(
            'ppdb/cek_hasil',
            $data
        );
    }
```

```php
public function proses_cek_hasil()
{
    $kode = trim(
        $this->input->post('kode_pendaftaran')
    );

    $hasil = $this->db
        ->where('kode_pendaftaran', $kode)
        ->get('pendaftaran')
        ->row();

    $data['hasil'] = $hasil;
    $data['not_found'] = empty($hasil);

    $this->load->view(
        'ppdb/cek_hasil',
        $data
    );
}


public function download_formulir($id)
{
    $ppdb = $this->db
        ->where('id', $id)
        ->get('pendaftaran')
        ->row();

    if (!$ppdb) {
        show_404();
    }

    $pdf = new \setasign\Fpdi\Fpdi();

    function titik($pdf, $x, $y)
    {
        $pdf->SetXY($x, $y);
        $pdf->SetFont('Arial', 'B', 6);
        $pdf->SetTextColor(255, 0, 0);

        $pdf->Write(
            0,
            "($x,$y)"
        );
    }

    $pdf->AddPage();
```

```
        $template = FCPATH .
'assets/formulir/formulir_PPDB_Ar_Raudhah.pdf';

        $pageCount = $pdf->setSourceFile($template);

        $tplId = $pdf->importPage(1);

        $pdf->useTemplate($tplId, 0, 0, 210);


        // FONT
        $pdf->SetFont('Arial', '', 10);
        $pdf->SetTextColor(0, 0, 0);

        // POSISI SETELAH TANDA ":"
        $x = 112;

        /* =======================================================
     * DATA SISWA
     * ======================================================= */

        $pdf->SetXY($x, 80);
        $pdf->Write(0, $ppdb->nisn);

        $pdf->SetXY($x, 85);
        $pdf->Write(0, $ppdb->nama_siswa);

        $pdf->SetXY($x, 90);
        $pdf->Write(0, $ppdb->jenis_kelamin);

        $pdf->SetXY($x, 95);
        $pdf->Write(0, $ppdb->tempat_lahir);

        $pdf->SetXY($x, 100);
        $pdf->Write(0, date('d-m-Y', strtotime($ppdb->tanggal_lahir)));

        $pdf->SetXY($x, 104);
        $pdf->Write(0, $ppdb->agama);

        $pdf->SetXY($x, 109);
        $pdf->Write(0, $ppdb->anak_ke);

        $pdf->SetXY($x, 114);
        $pdf->Write(0, $ppdb->status_keluarga);

        $pdf->SetXY($x, 117);
        $pdf->MultiCell(70, 4, $ppdb->alamat);
```

```
    /* ======================================================
   * SEKOLAH ASAL
   * ====================================================== */

    $pdf->SetXY($x, 138);
    $pdf->Write(0, $ppdb->asal_sekolah);

    $pdf->SetXY($x, 141);
    $pdf->MultiCell(70, 4, $ppdb->alamat_sekolah);

    $pdf->SetXY($x, 148);
    $pdf->Write(0, $ppdb->nomor_ijazah);

    $pdf->SetXY($x, 152);
    $pdf->Write(0, $ppdb->nomor_skhun);

    /* ======================================================
   * DATA ORANG TUA
   * ====================================================== */

    $pdf->SetXY($x, 167);
    $pdf->Write(0, $ppdb->nama_ayah);

    $pdf->SetXY($x, 171);
    $pdf->Write(0, $ppdb->nama_ibu);

    $pdf->SetXY($x, 176);
    $pdf->Write(0, $ppdb->no_hp_ortu);

    $pdf->SetXY($x, 179);
    $pdf->MultiCell(70, 4, $ppdb->alamat_ortu);

    $pdf->SetXY($x, 186);
    $pdf->Write(0, $ppdb->pekerjaan_ayah);

    $pdf->SetXY($x, 190);
    $pdf->Write(0, $ppdb->pekerjaan_ibu);

    $pdf->SetXY($x, 195);
    $pdf->Write(0, $ppdb->nama_wali);

    $pdf->SetXY($x, 198);
    $pdf->MultiCell(70, 4, $ppdb->alamat_wali);

    $pdf->SetXY($x, 205);
    $pdf->Write(0, $ppdb->pekerjaan_wali);

    $pdf->SetXY($x, 210);
```

```
        $pdf->Write(0, $ppdb->no_hp_wali);


        $pdf->Output(
            'I',
            'Formulir_' . $ppdb->kode_pendaftaran . '.pdf'
        );
    }
```

k. Halaman Kelola Data Siswa

```
class Datasiswa extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->library('session');
        $this->load->helper(['url', 'form']);
        $this->load->database();
    }

    public function index()
    {
        $user = $this->session->userdata('user');
        $role = $user->role;

        if ($role == 'admin') {

            $kelas_akses = $user->kelas;

            // admin hanya lihat kelas sendiri
            $kelas_list = [$kelas_akses];
            $siswa = $this->db->where('kelas',
$kelas_akses)->get('siswa')->result();
        } else {

            // superadmin lihat semua kelas
            $kelas_list =
$this->db->select('kelas')->group_by('kelas')->get('siswa')->result_arr
ay();
            $kelas_list = array_column($kelas_list, 'kelas');

            $siswa = $this->db->get('siswa')->result();
        }

        $data['kelas_list'] = $kelas_list;
        $data['siswa'] = $siswa;
```

```
        $this->load->view('admin/datasiswa', $data);
    }

    public function import_excel()
    {
        // Cek apakah file dipilih
        if (!isset($_FILES['excel']) || $_FILES['excel']['error'] != 0)
{
            $this->session->set_flashdata('flash', [
                'type' => 'error',
                'msg'  => 'Silakan pilih file Excel terlebih dahulu!'
            ]);
            redirect('admin/datasiswa');
        }

        $file = $_FILES['excel'];

        // Validasi ekstensi
        $allowed = ['xls', 'xlsx'];
        $ext = strtolower(pathinfo($file['name'], PATHINFO_EXTENSION));

        if (!in_array($ext, $allowed)) {
            $this->session->set_flashdata('flash', [
                'type' => 'error',
                'msg'  => 'File harus berformat Excel (.xls
atau .xlsx)!'
            ]);
            redirect('admin/datasiswa');
        }

        try {

            require 'vendor/autoload.php';

            $reader =
\PhpOffice\PhpSpreadsheet\IOFactory::createReaderForFile($file['tmp_nam
e']);
            $spreadsheet = $reader->load($file['tmp_name']);
            $sheet = $spreadsheet->getActiveSheet()->toArray(null,
true, true, true);

            // Validasi format header Excel
            $header = [
                'A' => trim($sheet[1]['A']),
                'B' => trim($sheet[1]['B']),
                'C' => trim($sheet[1]['C']),
                'D' => trim($sheet[1]['D']),
                'E' => trim($sheet[1]['E']),
```

```
            ];

            $header_benar = [
                'A' => 'Nama',
                'B' => 'Kelas',
                'C' => 'Jenis Kelamin',
                'D' => 'Username',
                'E' => 'Password',
            ];

            if ($header != $header_benar) {
                $this->session->set_flashdata('flash', [
                    'type' => 'error',
                    'msg'  => 'Format file Excel tidak sesuai. Gunakan
template yang benar.'
                ]);
                redirect('admin/datasiswa');
            }

            for ($i = 2; $i <= count($sheet); $i++) {

                $nama     = trim($sheet[$i]['A']);
                $kelas    = trim($sheet[$i]['B']);
                $jk_excel = strtolower(trim($sheet[$i]['C']));
                $username = trim($sheet[$i]['D']);
                $password = trim($sheet[$i]['E']);

                $jk = ($jk_excel == 'l' || $jk_excel == 'laki-laki')
                    ? 'L'
                    : (($jk_excel == 'p' || $jk_excel == 'perempuan')
                        ? 'P'
                        : null);

                // Validasi jenis kelamin
                if ($jk === null) {
                    $this->session->set_flashdata('flash', [
                        'type' => 'error',
                        'msg'  => 'Jenis kelamin pada baris ke-' . $i .
' harus diisi dengan L, P, Laki-laki, atau Perempuan.'
                    ]);
                    redirect('admin/datasiswa');
                }

                if (empty($username)) {
                    $this->session->set_flashdata('flash', [
                        'type' => 'error',
                        'msg'  => 'Username pada baris ke-' . $i . '
tidak boleh kosong.'
```

```
                    ]);
                    redirect('admin/datasiswa');
                }

                $siswa = $this->db->get_where('siswa', [
                    'username' => $username
                ])->row();

                if (
                    empty($nama) ||
                    empty($kelas) ||
                    empty($jk_excel) ||
                    empty($username) ||
                    empty($password)
                ) {
                    $this->session->set_flashdata('flash', [
                        'type' => 'error',
                        'msg'  => 'Data pada baris ke-' . $i . ' belum
lengkap.'
                    ]);
                    redirect('admin/datasiswa');
                }

                if (!$siswa) {

                    $this->db->insert('siswa', [
                        'nama' => $nama,
                        'kelas' => $kelas,
                        'jenis_kelamin' => $jk,
                        'username' => $username,
                        'password' => $password
                    ]);
                } else {

                    $this->db->where('id', $siswa->id)->update('siswa',
[
                        'nama' => $nama,
                        'kelas' => $kelas,
                        'jenis_kelamin' => $jk,
                        'password' => $password
                    ]);
                }
            }

            $this->session->set_flashdata('flash', [
                'type' => 'success',
                'msg' => 'Data siswa berhasil diimport!'
            ]);
```

```
            $this->load->model('Aktivitas_model');
            $this->Aktivitas_model->tambah('Kelola Data Siswa',
'Mengimport Excel');
        } catch (Exception $e) {

            $this->session->set_flashdata('flash', [
                'type' => 'error',
                'msg' => 'File Excel tidak valid atau rusak!'
            ]);
        }

        redirect('admin/datasiswa');
    }
 public function delete($id)
    {

        $siswa = $this->db->get_where('siswa', ['id' => $id])->row();
        // hapus raport siswa
        $raport = $this->db->get_where('raport', ['siswa_id' =>
$id])->row();
        if ($raport) {
            $filePath = './uploads/raport/' . $raport->file_pdf;
            if ($raport->file_pdf && file_exists($filePath))
unlink($filePath);
            $this->db->delete('raport', ['siswa_id' => $id]);
        }

        // hapus spp siswa
        $spp = $this->db->get_where('spp', ['siswa_id' => $id])->row();
        if ($spp) {
            $filePath = './uploads/spp/' . $spp->file_pdf;
            if ($spp->file_pdf && file_exists($filePath))
unlink($filePath);
            $this->db->delete('spp', ['siswa_id' => $id]);
        }

        // hapus siswa
        $this->db->delete('siswa', ['id' => $id]);

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
        $this->Aktivitas_model->tambah('Kelola Data Siswa', 'Menghapus
siswa: ' . $siswa->nama);

        $this->session->set_flashdata('flash', ['type' => 'danger',
'msg'  => 'Siswa dan semua file terkait berhasil dihapus !!']);
        redirect('admin/datasiswa');
```

```
    }

    public function delete_all()
    {
        // hapus semua file raport
        foreach ($this->db->get('raport')->result() as $r) {
            $file = './uploads/raport/' . $r->file_pdf;
            if ($r->file_pdf && file_exists($file)) unlink($file);
        }
        // hapus semua file spp
        foreach ($this->db->get('spp')->result() as $s) {
            $file = './uploads/spp/' . $s->file_pdf;
            if ($s->file_pdf && file_exists($file)) unlink($file);
        }

        $this->db->query('SET FOREIGN_KEY_CHECKS=0');
        $this->db->query('TRUNCATE TABLE raport');
        $this->db->query('TRUNCATE TABLE spp');
        $this->db->query('TRUNCATE TABLE siswa');
        $this->db->query('SET FOREIGN_KEY_CHECKS=1');

        $this->session->set_flashdata('flash', ['type' => 'danger',
'msg'  => 'Semua data siswa & file berhasil dihapus !!']);

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
        $this->Aktivitas_model->tambah('Kelola Data Siswa', 'Menghapus
Seluruh siswa');

        redirect('admin/datasiswa');
    }

    public function tambah()
    {
        $data = [
            'nama'          => $this->input->post('nama'),
            'kelas'         => $this->input->post('kelas'),
            'jenis_kelamin' => $this->input->post('jenis_kelamin'),
            'username'      => $this->input->post('username'),
            'password'      => $this->input->post('password')
        ];

        $this->db->insert('siswa', $data);

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
        $this->Aktivitas_model->tambah('Kelola Data Siswa',
'Menambahkan siswa: ' . $data['nama']);
```

```
        $this->session->set_flashdata('flash', [
            'type' => 'success',
            'msg'  => 'Siswa baru berhasil ditambahkan!'
        ]);
        redirect('admin/datasiswa');
    }
}
```

l. Halaman Kelola Informasi

```
class Informasi extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->model('Informasi_model');
        $this->load->library('session');
        $this->load->helper('text');
    }

    public function index()
    {
        $data['informasi'] = $this->Informasi_model->getAll();
        $this->load->view('admin/informasi', $data);
    }

    public function simpan()
    {
        $judul = $this->input->post('judul');
        $isi   = $this->input->post('isi');

        $data = [
            'judul' => $judul,
            'isi'   => $isi,
            'author' => 'Superadmin'
        ];

        $this->Informasi_model->insert($data);

        $this->session->set_flashdata('flash', [
            'type' => 'success',
            'msg' => 'Informasi berhasil ditambahkan!'
        ]);

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
```

```
        $this->Aktivitas_model->tambah('Kelola Informasi', 'Menambah
Informasi ');

        redirect('admin/informasi');
    }

    // Hapus informasi
    public function hapus($id)
    {
        if ($this->Informasi_model->delete($id)) {
            $this->session->set_flashdata('flash', [
                'type' => 'danger',
                'msg' => 'Informasi berhasil di hapus !!!'
            ]);
        } else {
            $this->session->set_flashdata('error', 'Informasi gagal
dihapus.');
        }

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
        $this->Aktivitas_model->tambah('Kelola Informasi', 'Menghapus
Informasi ');

        redirect('admin/informasi');
    }
}
```

m. Halaman Kelola Pendaftaran

```
class Pendaftaran extends CI_Controller
{

    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        $data['pendaftaran'] = $this->db
            ->order_by('id', 'DESC')
            ->get('pendaftaran')
            ->result();

        $this->load->view('admin/pendaftaran', $data);
```

```
    }

    public function pendaftaran()
    {
        $data['total_pendaftar'] = $this->db->count_all('pendaftaran');

        $data['total_pending'] = $this->db
            ->where('status_seleksi', 'Menunggu')
            ->count_all_results('pendaftaran');

        $data['total_diterima'] = $this->db
            ->where('status_seleksi', 'Diterima')
            ->count_all_results('pendaftaran');

        $data['total_ditolak'] = $this->db
            ->where('status_seleksi', 'Ditolak')
            ->count_all_results('pendaftaran');

        $this->load->view('admin/pendaftaran', $data);
    }

 public function detail($id)
    {
        $data['siswa'] = $this->db
            ->where('id', $id)
            ->get('pendaftaran')
            ->row();

        if (!$data['siswa']) {
            show_404();
        }

        $this->load->view('admin/pendaftaran_detail', $data);
    }


    public function update_status($id)
    {
        $status = $this->input->post('status_seleksi');

        $this->db->where('id', $id);
        $this->db->update('pendaftaran', [
            'status_seleksi' => $status
        ]);

        $this->session->set_flashdata('success', 'Status seleksi
berhasil diperbarui');
```

```
        redirect('admin/pendaftaran');
    }

    public function export_excel()
    {
        $data = $this->db->get('pendaftaran')->result();

        $spreadsheet = new Spreadsheet();
        $sheet = $spreadsheet->getActiveSheet();

        $header = [
            'NISN',
            'Nama Siswa',
            'Jenis Kelamin',
            'Tempat Lahir',
            'Tanggal Lahir',
            'Agama',
            'Alamat',
            'No HP',
            'Anak Ke',
            'Status Keluarga',

            'Nama Ayah',
            'Pekerjaan Ayah',

            'Nama Ibu',
            'Pekerjaan Ibu',

            'No HP Ortu',
            'Alamat Ortu',

            'Nama Wali',
            'Pekerjaan Wali',
            'Alamat Wali',
            'No HP Wali',

            'Asal Sekolah',
            'Alamat Sekolah',
            'Nomor Ijazah',
            'Nomor SKHUN',

            'Foto 2x3',
            'Foto 3x4',
            'KK',
            'KTP Ortu',
            'Akta Kelahiran',
            'Ijazah',
            'Bukti NISN',
```

```
        'Status Seleksi'
    ];
    $col = 'A';

    foreach ($header as $h) {
        $sheet->setCellValue($col . '1', $h);
        $col++;
    }

    $row = 2;

    foreach ($data as $siswa) {

        $sheet->setCellValue('A' . $row, $siswa->nisn);
        $sheet->setCellValue('B' . $row, $siswa->nama_siswa);
        $sheet->setCellValue('C' . $row, $siswa->jenis_kelamin);
        $sheet->setCellValue('D' . $row, $siswa->tempat_lahir);
        $sheet->setCellValue('E' . $row, $siswa->tanggal_lahir);
        $sheet->setCellValue('F' . $row, $siswa->agama);
        $sheet->setCellValue('G' . $row, $siswa->alamat);
        $sheet->setCellValue('H' . $row, $siswa->no_hp);
        $sheet->setCellValue('I' . $row, $siswa->anak_ke);
        $sheet->setCellValue('J' . $row, $siswa->status_keluarga);

        $sheet->setCellValue('K' . $row, $siswa->nama_ayah);
        $sheet->setCellValue('L' . $row, $siswa->pekerjaan_ayah);

        $sheet->setCellValue('M' . $row, $siswa->nama_ibu);
        $sheet->setCellValue('N' . $row, $siswa->pekerjaan_ibu);

        $sheet->setCellValue('O' . $row, $siswa->no_hp_ortu);
        $sheet->setCellValue('P' . $row, $siswa->alamat_ortu);

        $sheet->setCellValue('Q' . $row, $siswa->nama_wali);
        $sheet->setCellValue('R' . $row, $siswa->pekerjaan_wali);
        $sheet->setCellValue('S' . $row, $siswa->alamat_wali);
        $sheet->setCellValue('T' . $row, $siswa->no_hp_wali);

        $sheet->setCellValue('U' . $row, $siswa->asal_sekolah);
        $sheet->setCellValue('V' . $row, $siswa->alamat_sekolah);
        $sheet->setCellValue('W' . $row, $siswa->nomor_ijazah);
        $sheet->setCellValue('X' . $row, $siswa->nomor_skhun);

        // FOTO 2X3
        $sheet->setCellValue('Y' . $row, 'Lihat Foto 2x3');
        $sheet->getCell('Y' . $row)
            ->getHyperlink()
```

```
                ->setUrl(base_url('uploads/ppdb/' . $siswa->foto_2x3));

            // FOTO 3X4
            $sheet->setCellValue('Z' . $row, 'Lihat Foto 3x4');
            $sheet->getCell('Z' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' . $siswa->foto_3x4));

            // KK
            $sheet->setCellValue('AA' . $row, 'Lihat KK');
            $sheet->getCell('AA' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' . $siswa->kk));

            // KTP ORTU
            $sheet->setCellValue('AB' . $row, 'Lihat KTP');
            $sheet->getCell('AB' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' . $siswa->ktp_ortu));

            // AKTA
            $sheet->setCellValue('AC' . $row, 'Lihat Akta');
            $sheet->getCell('AC' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' .
$siswa->akta_kelahiran));

            // IJAZAH
            $sheet->setCellValue('AD' . $row, 'Lihat Ijazah');
            $sheet->getCell('AD' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' . $siswa->ijazah));

            // BUKTI NISN
            $sheet->setCellValue('AE' . $row, 'Lihat Bukti NISN');
            $sheet->getCell('AE' . $row)
                ->getHyperlink()
                ->setUrl(base_url('uploads/ppdb/' .
$siswa->bukti_nisn));

            $sheet->setCellValue('AF' . $row, $siswa->status_seleksi);

            $row++;
        }
        $filename = 'Data_Pendaftaran.xlsx';

        header('Content-Type: application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet');
```

```
        header('Content-Disposition: attachment; filename="' .
$filename . '"');

        $writer = new Xlsx($spreadsheet);
        $writer->save('php://output');
    }

    public function hapus($id)
    {
        $this->db
            ->where('id', $id)
            ->delete('pendaftaran');

        $this->session->set_flashdata(
            'success',
            'Data pendaftaran berhasil dihapus.'
        );

        redirect('admin/pendaftaran');
    }

    public function hapus_semua()
    {
        $this->db->empty_table('pendaftaran');

        $this->session->set_flashdata(
            'success',
            'Semua data pendaftaran berhasil dihapus.'
        );

        redirect('admin/pendaftaran');
    }
}
```

n. Halaman Kelola Pengumuman PPDB

```
class Pengumuman extends CI_Controller

{

    public function __construct()
    {
        parent::__construct();

        $this->load->library('email');
    }
    public function index()
```

```php
{
    $data['pengumuman'] = $this->db
        ->order_by('id', 'DESC')
        ->get('pengumuman')
        ->result();

    $this->load->view('admin/pengumuman', $data);
}


public function simpan_pengumuman()
{
    $data = [
        'judul'       => $this->input->post('judul'),
        'isi'         => $this->input->post('isi'),
        'tanggal'     => $this->input->post('tanggal'),
        'tombol_link' => $this->input->post('tombol_link'),
        'email_terkirim'  => 0
    ];

    $this->db->insert('pengumuman', $data);

    $pengumuman_id = $this->db->insert_id();

    // jika admin mencentang kirim email
    if ($this->input->post('kirim_email') == '1') {
        $this->kirim_email_pengumuman();

        $this->db
            ->where('id', $pengumuman_id)
            ->update(
                'pengumuman',
                [
                    'email_terkirim' => 1
                ]
            );
    }

    $this->session->set_flashdata(
        'flash',
        [
            'type' => 'success',
            'msg'  => 'Pengumuman berhasil ditambahkan'
        ]
    );

    redirect('admin/pengumuman');
}
```

```php
public function hapus_pengumuman($id)
{
    $this->db
        ->where('id', $id)
        ->delete('pengumuman');

    $this->session->set_flashdata(
        'flash',
        [
            'type' => 'success',
            'msg'  => 'Pengumuman berhasil dihapus'
        ]
    );

    redirect('admin/pengumuman');
}

private function kirim_email_pengumuman()
{
    $peserta = $this->db
        ->select('
    p.kode_pendaftaran,
    p.nama_siswa,
    a.email
')
        ->from('pendaftaran p')
        ->join(
            'akun_ppdb a',
            'a.id = p.user_id'
        )
        ->where('a.email IS NOT NULL')
        ->get()
        ->result();

    foreach ($peserta as $row) {


        $pesan = '
    <h3>Pengumuman Hasil Seleksi PPDB</h3>

    <p>Assalamu\'alaikum Wr. Wb.</p>

    <p>
    Pengumuman hasil seleksi PPDB SMK Ar-Raudhah Jakarta
    telah resmi dibuka.
    </p>
```

```
        <table>
            <tr>
                <td>Nama Peserta</td>
                <td>: ' . $row->nama_siswa . '</td>
            </tr>

            <tr>
                <td>Kode Pendaftaran</td>
                <td>: <b>' . $row->kode_pendaftaran . '</b></td>
            </tr>
        </table>

        <br>

        <a href="' . site_url('ppdb/cek-hasil') . '"
        style="
            background:#ff7900;
            color:white;
            padding:12px 20px;
            text-decoration:none;
            border-radius:6px;
        ">
            Cek Hasil Seleksi
        </a>

        <br><br>

        <p>
        Simpan kode pendaftaran Anda karena diperlukan
        untuk melihat hasil seleksi.
        </p>

        <p>
        Terima kasih.
        </p>
        ';

            $this->email->from(
                'salmansalfarisi8@gmail.com',
                'PPDB SMK Ar-Raudhah'
            );

            $this->email->to(
                'salmansalfarisi8@gmail.com'
            );
$this->email->subject(
                'Pengumuman Hasil Seleksi PPDB'
            );
```

```
            $this->email->message($pesan);

            if (!$this->email->send()) {

                log_message(
                    'error',
                    $this->email->print_debugger()
                );
            }
        }
    }
}
```

o. Halaman Kelola *Raport*

```
class Raport extends Admin_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function index()
    {
        $user = $this->session->userdata('user');
        $role = $user->role;

        if ($role == 'admin') {
            $kelas_akses = $user->kelas;

            $kelas_list = [$kelas_akses];
            $siswa = $this->db->where('kelas',
$kelas_akses)->get('siswa')->result();
        } else {
            $kelas_list =
$this->db->select('kelas')->group_by('kelas')->get('siswa')->result_arr
ay();
            $kelas_list = array_column($kelas_list, 'kelas');

            $siswa = $this->db->get('siswa')->result();
        }

        $raport = $this->db->get('raport')->result();

        $data = [
```

```
            'siswa'      => $siswa,
            'raport'     => $raport,
            'kelas_list' => $kelas_list
        ];

        $this->load->view('admin/raport', $data);
    }

    public function upload($siswa_id)
    {
        $config['upload_path']   = './uploads/raport/';
        $config['allowed_types'] = 'pdf';
        $config['max_size']      = 2048;
        $config['file_name']     = 'raport_' . $siswa_id . '_' .
time();

        $this->load->library('upload', $config);

        // ambil data siswa sebelum upload
        $siswa = $this->db->get_where('siswa', ['id' =>
$siswa_id])->row();

        if ($this->upload->do_upload('pdf')) {
            $file = $this->upload->data('file_name');

            // hapus file lama kalau ada
            $old = $this->db->get_where('raport', ['siswa_id' =>
$siswa_id])->row();
            if ($old && file_exists('./uploads/raport/' .
$old->file_pdf)) {
                unlink('./uploads/raport/' . $old->file_pdf);
            }

            // update / insert
            $cek = $this->db->get_where('raport', ['siswa_id' =>
$siswa_id])->row();
            if ($cek) {
                $this->db->where('id', $cek->id)->update('raport',
['file_pdf' => $file]);
            } else {
                $this->db->insert('raport', ['siswa_id' => $siswa_id,
'file_pdf' => $file]);
            }

            $this->session->set_flashdata('flash', ['type' =>
'success', 'msg' => 'Raport berhasil diupload.']);
        } else {
```

```
            $error = strip_tags($this->upload->display_errors());

            if (strpos($error, 'filetype') !== false || strpos($error,
'allowed') !== false) {
                $pesan = 'File yang diupload harus berformat PDF
(.pdf).';
            } elseif (strpos($error, 'exceeds') !== false ||
strpos($error, 'size') !== false) {
                $pesan = 'Ukuran file maksimal 2 MB.';
            } else {
                $pesan = $error;
            }

            $this->session->set_flashdata('flash', [
                'type' => 'error',
                'msg'  => $pesan
            ]);
        }
        // Catat aktivitas
        if ($siswa) {
            $this->load->model('Aktivitas_model');
            $this->Aktivitas_model->tambah('Kelola Raport Siswa',
'Mengupload PDF raport siswa: ' . $siswa->nama);
        }

        redirect('admin/raport');
    }

    public function delete($id)
    {
        $row = $this->db->get_where('raport', ['id' => $id])->row();

        // ambil data siswa terkait sebelum dihapus
        $siswa = $row ? $this->db->get_where('siswa', ['id' =>
$row->siswa_id])->row() : null;

        if ($row && file_exists('./uploads/raport/' . $row->file_pdf))
{
            unlink('./uploads/raport/' . $row->file_pdf);
        }

        $this->db->where('id', $id)->delete('raport');
        $this->session->set_flashdata('flash', ['type' => 'danger',
'msg' => 'Raport berhasil dihapus.']);

        // Catat aktivitas
        if ($siswa) {
            $this->load->model('Aktivitas_model');
```

```
            $this->Aktivitas_model->tambah('Kelola Raport Siswa',
'Menghapus PDF raport siswa: ' . $siswa->nama);
        }

        redirect('admin/raport');
    }

    public function delete_all()
    {
        $raport = $this->db->get('raport')->result();
        foreach ($raport as $r) {
            if ($r->file_pdf && file_exists('./uploads/raport/' .
$r->file_pdf)) {
                unlink('./uploads/raport/' . $r->file_pdf);
            }
        }
        $this->db->empty_table('raport');
        $this->session->set_flashdata('flash', ['type' => 'danger',
'msg' => 'Semua raport berhasil dihapus.']);

        // Catat aktivitas
        $this->load->model('Aktivitas_model');
        $this->Aktivitas_model->tambah('Kelola Raport Siswa',
'Menghapus seluruh file raport');

        redirect('admin/raport');
    }

    public function toggle($id)
    {
        $raport = $this->db->get_where('raport', ['id' => $id])->row();

        // ambil data siswa terkait
        $siswa = $raport ? $this->db->get_where('siswa', ['id' =>
$raport->siswa_id])->row() : null;

        if ($raport) {
            $is_active = $this->input->post('is_active') ? 1 : 0;
            $this->db->where('id', $id)->update('raport', ['is_active'
=> $is_active]);
            $this->session->set_flashdata('flash', [
                'type' => 'success',
                'msg' => 'Status raport berhasil diperbarui.'
            ]);
        }

        // Catat aktivitas
        if ($siswa) {
```

```
            $this->load->model('Aktivitas_model');
            $status = isset($is_active) && $is_active ? 'ON' : 'OFF';
            $this->Aktivitas_model->tambah('Kelola Raport Siswa',
'Mengubah status raport siswa (' . $status . '): ' . $siswa->nama);
        }

        redirect('admin/raport');
    }
}
```

4.4 Testing

1. Testing Form Login

Tabel IV.10 Hasil pengujian *black box testing form* login

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Login dengan data valid	Username dan password sesuai dengan data yang terdaftar	Sistem mengarahkan pengguna ke dashboard sesuai dengan hak akses yang dimiliki.	Dashboard berhasil ditampilkan	Berhasil
2	Login dengan username salah	Username salah, password benar	Sistem menampilkan pemberitahuan "Username atau Password salah"	Pesan error ditampilkan	Berhasil
3	Login dengan password salah	Username benar, password salah	Sistem menampilkan pesan "Username atau Password salah"	Pesan error ditampilkan	Berhasil
4	Login dengan kolom kosong	Username dan password dikosongkan	Sistem menampilkan pesan validasi input tidak boleh kosong	Validasi berhasil ditampilkan	Berhasil
5	Login menggunakan akun yang tidak terdaftar	Username dan password tidak ada di database	Sistem menampilkan pesan error	Pesan error ditampilkan	Berhasil

2. *Testing Form* Buat Akun PPDB

Tabel IV.11Hasil pengujian *black box testing form* buat akun PPDB

No.	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Membuat akun dengan data valid	Semua data diisi benar	Akun berhasil dibuat	sistem menampilkan pesan akun berhasil di buat	Berhasil
2	Email sudah terdaftar	Menggunakan email yang sama	Sistem menampilkan pesan email sudah terdaftar	Pesan tampil	Berhasil
3	NISN sudah terdaftar	Menggunakan NISN yang sama	Sistem menampilkan pesan NISN sudah terdaftar	Pesan tampil	Berhasil
4	Data kosong	Seluruh data di kosongkan	Sistem menampilkan validasi	Validasi berhasil	Berhasil
5	Password dan konfirmasi berbeda	Password tidak sama	Sistem menampilkan pesan kesalahan	Pesan tampil	Berhasil

3. *Testing Form* Pendaftaran

Tabel IV.12 Hasil pengujian *black box testing form* pendaftaran

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Mengisi formulir lengkap	Semua data benar	Data berhasil disimpan	Data tersimpan	Berhasil
2	Data kosong	Seluruh data kosong	Sistem menampilkan validasi	Validasi berhasil	Berhasil
3	Upload berkas valid	File sesuai ketentuan	Upload berhasil	Sistem menampilkan pesan file berhasil di upload	Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
4	Upload format tidak didukung	Upload .exe	Sistem menolak file	Sistem menampilkan pesan upload gagal	Berhasil
5	Kirim pendaftaran	Klik Kirim Pendaftaran	Data masuk tabel pendaftaran dan kode pendaftaran dibuat	Kode pendaftaran tampil	Berhasil

4. *Testing* Cek Hasil Seleksi *PPDB*

Tabel IV.13 Hasil pengujian black box testing cek hasil seleksi PPDB

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Kode pendaftaran benar	Kode terdaftar	Status seleksi ditampilkan	Status tampil	Berhasil
2	Kode tidak ditemukan	Kode tidak terdaftar	Sistem menampilkan pesan data tidak ditemukan	Pesan tampil	Berhasil
3	Kolom kosong	Kode tidak diisi	Validasi muncul	Validasi tampil	Berhasil

5. *Testing Form* Kelola Informasi

Tabel IV.14 Hasil pengujian black box testing form kelola informasi

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Tambah informasi	Data valid	Informasi tersimpan	Sistem menampilkan pesan informasi berhasil ditambahkan	Berhasil
2	Hapus informasi	Klik hapus	Data informasi dihapus	Sistem menampilkan notifikasi	Berhasil

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
				informasi berhasil di hapus	
3	Kolom kosong	Judul dan isi dikosongkan	Validasi muncul	Validasi tampil	Berhasil

6. *Testing* Kelola Pengumuman PPDB

Tabel IV.15 Hasil pengujian black box testing kelola pengumuman PPDB

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Tambah Pengumuman	Data valid	Pengumuman tersimpan	Sistem menampilkan pesan pengumuman berhasil ditambahkan	Berhasil
2	Hapus pengumuman	Klik hapus	Pengumuman dihapus	Sistem menampilkan pesan pengumuman berhasil dihapus	Berhasil
3	Kolom kosong	Kolom tidak diisi	Validasi muncul	Validasi tampil	Berhasil
4	Mengatur tanggal publish pengumuman	Menentukan tanggal publish lebih dari tanggal saat ini	Pengumuman belum ditampilkan pada website sebelum tanggal publish	Pengumuman tampil sesuai tanggal yang telah ditentukan	Berhasil
5	Membuat pengumuman tanpa tombol	Opsi tombol dipilih tidak ada	Pengumuman ditampilkan tanpa tombol aksi	Pengumuman tampil tanpa tombol	Berhasil
6.	Membuat pengumuman dengan tombol cek hasil	Opsi tombol dipilih cek hasil	Pengumuman menampilkan tombol cek hasil yang mengarah	Tombol berhasil ditampilkan dan berfungsi dengan baik	Berhasil

			ke halaman pengecekan hasil seleksi		
7	Mengirim email pemberitahuan kepada peserta	Checkbox Kirim email dicentang kemudian klik simpan	Sistem mengirim email pemberitahuan kepada seluruh peserta PPDB	Email berhasil dikirim kepada peserta yang mendaftar	Berhasil

7. Testing *E-Raport*

Tabel IV.16 Hasil pengujian black box testing *e-raport*

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Upload file PDF	File PDF valid	Raport berhasil diupload	Sistem menampilkan pesan raport berhasil di upload	Berhasil
2	Upload file selain PDF	Upload JPG/DOCX	Sistem menolak upload	Sistem menampilkan pesan file yang di upload harus berformat pdf	Berhasil
3	Hapus raport	Klik hapus	Raport berhasil terhapus	Sistem menampilkan pesan raport berhasil di hapus	Berhasil
4	Siswa melihat *e-raport*	Klik raport pada menu akun saya	File PDF ditampilkan	Raport tampil	Berhasil
5	Download *e-raport*	Klik tombol download	File berhasil diunduh	File berhasil diunduh	Berhasil

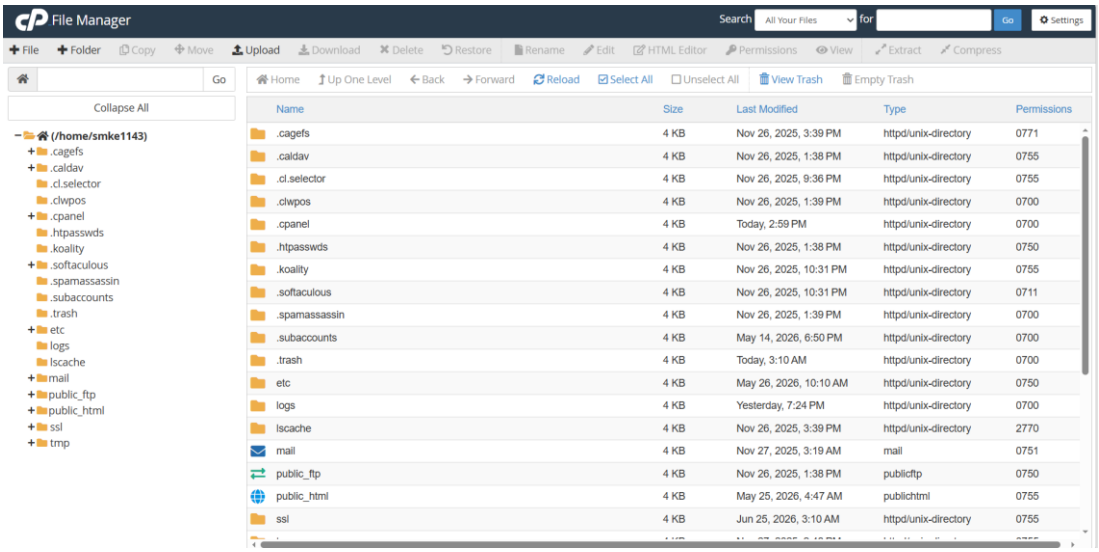
8. *Testing Form* Kelola Data Siswa

Tabel IV.17 Hasil pengujian black box testing form kelola data siswa

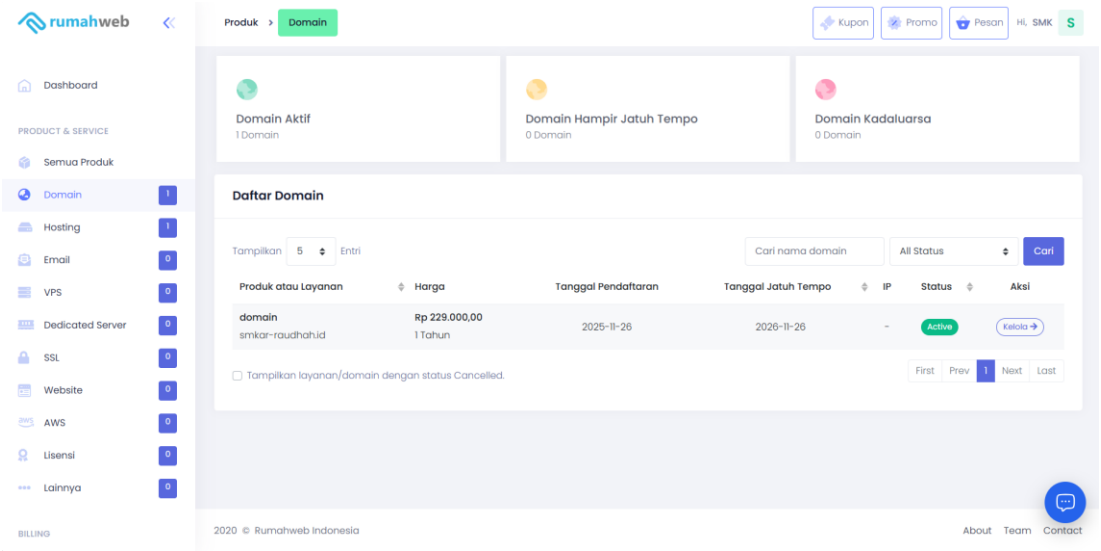
No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
1	Tambah data siswa	Data valid	Data berhasil ditambahkan	Sistem menampilkan pesan siswa berhasil di tambahkan	Berhasil
2	Hapus data siswa	Klik hapus	Data berhasil dihapus	Sistem menampilkan pesan siswa berhasil di hapus	Berhasil
3	Upload Excel valid	Format sesuai template	Import berhasil	Sistem menampilkan pesan import excel berhasil	Berhasil
4	Upload Excel tidak valid	Format salah	Sistem menolak upload	Sistem menampilkan pesan Format file Excel tidak sesuai. Gunakan template yang benar.	Berhasil
5.	Upload file selain excel	Upload PDF	Sistem menolak upload	Sistem menampilkan pesan file harus berformat excel	Berhasil

4.5 Support

4.5.1 Publikasi *Web*



Gambar IV.34 Bukti publikasi web



Gambar IV.35 Bukti beli domain

4.5.2 Spesifikasi Hardware dan Software

Tabel IV.18 Spesifikasi hardware dan software

Kebutuhan	Keterangan
Sistem Operasi	Microsoft Windows 11
Processor	AMD Ryzen 5 3500U
RAM	8 GB
Harddisk / SSD	239 GB
Monitor	14 inci HD (1366 × 768), Anti-Glare, LED, 220 nits
Keyboard	Full-size Island-style Spill-resistant Keyboard
Browser	Microsoft Edge, Google Chrome, Mozilla Firefox
Web Server	XAMPP (Apache, PHP, MySQL)
Bahasa Pemrograman	PHP , CSS , JavaScript
Framework	CodeIgniter 3
Database	MySQL
Code Editor	Visual Studio Code

Software Pendukung	Microsoft Office 365, PDF Reader
Koneksi Internet	Wi-Fi

4.6 Spesifikasi Dokumen Sistem Usulan

1. Nama Dokumen : Formulir Pendaftaran Peserta Didik Baru (PPDB)
   Fungsi : Mendaftarkan calon peserta didik baru melalui sistem PPDB berbasis web.
   Sumber : Calon Siswa
   Media : Digital
   Tujuan : Menyimpan data pendaftaran calon peserta didik ke dalam database.
   Frekuensi : Setiap calon siswa melakukan pendaftaran.

2. Nama Dokumen : Data Siswa
   Fungsi : Menyimpan dan mengelola data identitas siswa yang aktif di sekolah.
   Sumber : Superadmin
   Media : Digital
   Tujuan : Menyimpan data siswa sebagai data utama pada sistem informasi akademik.
   Frekuensi : Setiap terdapat penambahan atau penghapusan data siswa.

3. Nama Dokumen : *E-Raport* Siswa

   Fungsi : menyimpan dan menyajikan hasil belajar siswa dalam bentuk dokumen PDF.

   Sumber : Superadmin/Admin

   Media : Digital

   Tujuan : Memudahkan siswa dan wali murid dalam melihat dan mengunduh *e-raport* secara daring.

   Frekuensi : Setiap Akhir Semester

4. Nama Dokumen : Informasi Sekolah

   Fungsi : Menyampaikan informasi sekolah kepada siswa, orang tua, dan masyarakat melalui website.

   Sumber : Superadmin

   Media : Digital

   Tujuan : Menyediakan informasi sekolah yang dapat diakses secara online.

   Frekuensi : Setiap terdapat informasi terbaru.

REPUBLIK INDONESIA
KEMENTERIAN HUKUM

SURAT PENCATATAN CIPTAAN

Dalam rangka pelindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan	:	EC002026131424, 3 Agustus 2026
Pencipta		
Nama	:	**SALMAN AL FARISI, Surya Indrianti dkk**
Alamat	:	Jalan Cemara Blok I GG III. No 75, Lagoa, Koja, Jakarta Utara, DKI Jakarta, Koja, Kota Adm. Jakarta Utara, DKI Jakarta, 14269
Kewarganegaraan	:	Indonesia
Pemegang Hak Cipta		
Nama	:	**SALMAN AL FARISI, Surya Indrianti dkk**
Alamat	:	Jalan Cemara Blok I GG III. No 75, Lagoa, Koja, Jakarta Utara, DKI Jakarta, Koja, Kota Adm. Jakarta Utara, DKI Jakarta, 14269
Kewarganegaraan	:	Indonesia
Jenis Ciptaan	:	**Program Komputer**
Judul Ciptaan	:	**AR-RAUDHAH ACADEMIC SYSTEM SISTEM INFORMASI AKADEMIK BERBASIS WEBSITE UNTUK PENGELOLAAN DATA AKADEMIK SEKOLAH**
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	:	3 Agustus 2026, di Kota Adm. Jakarta Timur
Jangka waktu pelindungan	:	Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	:	001391069

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.
Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM
DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL
u.b
Direktur Hak Cipta dan Desain Industri

Agung Damarsasongko,SH.,MH.
NIP. 196912261994031001

Balai Besar Sertifikasi Elektronik

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang ditertibkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	SALMAN AL FARISI	Jalan Cemara Blok I GG III. No 75, Lagoa, Koja, Jakarta Utara, DKI Jakarta Koja, Kota Adm. Jakarta Utara
2	Surya Indrianti	Jl. Cemara Blok Ind No. 75, Koja, Jakarta Utara 14270 Koja, Kota Adm. Jakarta Utara
3	Nurul Khasanah	Gebang II, RT 03/RW 02, Plumbon, Temon, Kulon Progo, Yogyakarta, 55654 Temon, Kab. Kulon Progo

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	SALMAN AL FARISI	Jalan Cemara Blok I GG III. No 75, Lagoa, Koja, Jakarta Utara, DKI Jakarta Koja, Kota Adm. Jakarta Utara
2	Surya Indrianti	Jl. Cemara Blok Ind No. 75, Koja, Jakarta Utara 14270 Koja, Kota Adm. Jakarta Utara
3	Nurul Khasanah	Gebang II, RT 03/RW 02, Plumbon, Temon, Kulon Progo, Yogyakarta, 55654 Temon, Kab. Kulon Progo

Balai Besar Sertifikasi Elektronik

Disclaimer:
1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.
2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang ditertibkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara.
3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.