

BAB III

PELAKSANAAN KERJA PROFESI

3.1. Bidang Kerja

Dalam pelaksanaan kerja profesi di SMP Pembangunan Jaya Bintaro, praktikan ditempatkan di bawah bagian Staf IT Support, yang memiliki peran strategis dalam mendukung operasional sekolah melalui pemanfaatan teknologi informasi. Unit ini bertanggung jawab terhadap pemeliharaan sistem digital sekolah, pengelolaan aplikasi baik berbasis desktop maupun web, serta dukungan teknis terhadap infrastruktur server dan perangkat jaringan yang digunakan oleh pihak sekolah.

Praktikan memperoleh tanggung jawab utama sebagai pengembang aplikasi desktop, dengan fokus khusus pada pembuatan Aplikasi Buku Induk Digital. Aplikasi ini dirancang untuk menjawab tantangan administratif yang selama ini dihadapi oleh bagian Tata Usaha dalam mengelola data siswa. Sebelum sistem ini dikembangkan, proses pencarian data siswa baik yang masih aktif, sudah keluar, maupun yang telah lulus masih dilakukan secara manual menggunakan dokumen fisik dalam bentuk binder. Pendekatan ini tidak hanya menyulitkan pencarian data karena tidak sistematis, tetapi juga rawan terhadap kehilangan dan tidak lengkapnya informasi. Oleh karena itu, pengembangan aplikasi ini menjadi solusi digital yang bertujuan untuk meningkatkan efisiensi pencatatan dan pencarian data siswa.

Dalam pengembangan proyek tersebut, praktikan bertugas mulai dari tahap analisis kebutuhan pengguna, perancangan antarmuka, hingga proses pemrograman *prototyping* dari aplikasi. Adapun ruang lingkup kerja tidak mencakup tahap akhir seperti pengujian, implementasi, maupun pemeliharaan sistem. Aplikasi dirancang agar bersifat sederhana namun informatif, dan dapat digunakan oleh bagian Tata Usaha tanpa memerlukan pelatihan teknis yang mendalam.

Di luar tanggung jawab utamanya di bagian IT, praktikan juga mendapat tugas tambahan sebagai asisten pada Staf Tata Usaha. Dalam tugas ini, praktikan turut membantu dalam proses pelayanan administratif kepada siswa, guru, orang tua, dan staf sekolah lainnya. Kegiatan ini mencakup pengelolaan dokumen,

layanan informasi, serta penyediaan bantuan teknis apabila dibutuhkan dalam konteks operasional harian di sekolah.

Secara keseluruhan, kerja profesi ini memberikan pengalaman yang holistik, tidak hanya dalam hal pengembangan perangkat lunak tetapi juga dalam pemahaman terhadap kebutuhan riil di lingkungan pendidikan dan keterlibatan langsung dalam pelayanan administratif, yang menjadi bagian penting dari sistem pendukung sekolah.

3.2. Pelaksanaan Kerja

Dalam pengembangan sistem yang diberikan, praktikan melakukan diskusi dengan pembimbing untuk membuat lini masa kerja dalam upaya membantu praktikan dalam melakukan pekerjaannya secara sistematis.

Tabel 3.1 Tabel Pelaksanaan Kerja Profesi

Aktivitas	Februari	Maret				April			
	M4	M1	M2	M3	M4	M1	M2	M3	M4
Pengenalan Instansi									
Analisis Kebutuhan Fitur									
Perancangan Antarmuka									
Pengembangan Prototyping									

Berdasarkan tabel 3.1 dapat dilihat alur pengembangan aplikasi yang berlangsung kurang lebih selama dua bulan. Aplikasi masih akan terus mengalami perubahan untuk memenuhi kebutuhan dari penggunanya. Selama periode tersebut, praktikan secara rutin melakukan diskusi dengan pembimbing untuk mendapatkan arahan dalam membuat aplikasi dan memastikan aplikasi sesuai dengan kebutuhan pengguna.

Dalam proses pengembangan, institusi menerapkan beberapa tahapan untuk memastikan aplikasi berfungsi dengan baik dan tanpa hambatan. Dalam beberapa tahapan tersebut praktikan turut berproses pada fase perancangan dan fase pengembangan *prototyping*.

3.2.1. Deskripsi Proyek

Pengembangan Aplikasi Buku Induk Sekolah Digital berbasis desktop dilakukan untuk menyediakan sistem pencatatan data siswa yang terstruktur dan secara digital di SMP Pembangunan Jaya Bintaro. Sebelumnya, pencatatan data siswa termasuk siswa aktif, siswa yang telah lulus, maupun siswa yang telah keluar dilakukan dalam bentuk dokumen fisik. Metode ini menyulitkan proses pencarian data karena bergantung pada pencatatan manual dan penyimpanan arsip dalam binder. Permasalahan seperti tidak lengkapnya data, kesalahan pencatatan, serta keterbatasan akses terhadap informasi historis menjadi latar belakang utama dari proyek ini.

Aplikasi dirancang menggunakan Bahasa Pemrograman Python dan memanfaatkan pustaka antarmuka grafis Tkinter serta modul tambahan ttkbootstrap untuk keperluan desain antarmuka. Penyimpanan data dilakukan menggunakan SQLite, sehingga seluruh data tersimpan dalam satu basis data lokal yang dapat diakses melalui aplikasi. Proses kerja sistem dimulai ketika pengguna (staf administrasi sekolah) berinteraksi dengan antarmuka aplikasi. Pengguna dapat melakukan berbagai operasi seperti menambah data siswa baru melalui formulir terstruktur, melihat daftar siswa yang ada dengan fitur pencarian dan penyaringan, mengakses detail informasi setiap siswa, mengubah data siswa jika diperlukan, serta menghapus data siswa yang sudah tidak relevan. Selain itu, sistem dilengkapi dengan fitur *dashboard* untuk visualisasi statistik data siswa, serta fungsionalitas impor data dari berkas Excel maupun CSV dan ekspor data ke berkas Excel untuk kemudahan transfer dan pelaporan data. Setiap interaksi yang melibatkan perubahan data akan langsung tercatat dan diperbarui dalam basis data SQLite.

3.2.2. Perancangan

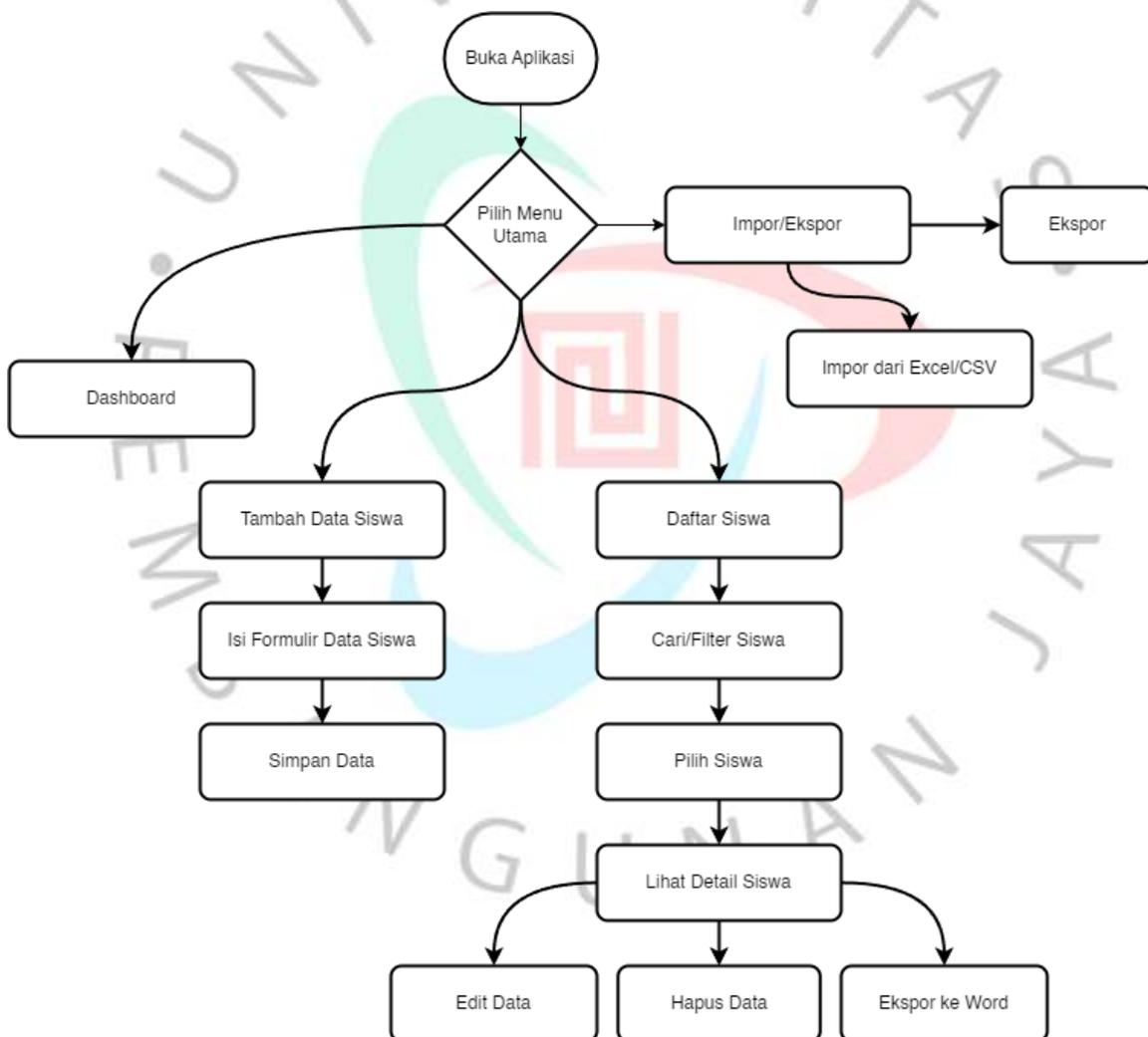
Pada fase perancangan aplikasi, dilakukan beberapa langkah untuk memastikan pengembangan aplikasi buku induk sekolah digital berbasis desktop dapat berjalan dengan terarah dan sesuai kebutuhan pengguna.

3.2.2.1. Analisis Kebutuhan

Langkah awal dimulai dengan arahan dari pembimbing untuk melakukan analisis mendalam terhadap kebutuhan sistem pengelolaan buku induk digital.

Analisis ini mencakup pembuatan diagram alir (*flowchart*), diagram aktivitas (*activity diagram*), dan diagram kelas (*class diagram*) yang bertujuan untuk memberikan gambaran menyeluruh tentang alur kerja, struktur data, dan komponen-komponen sistem. Pembuatan diagram ini mempermudah proses pengembangan di tahap selanjutnya.

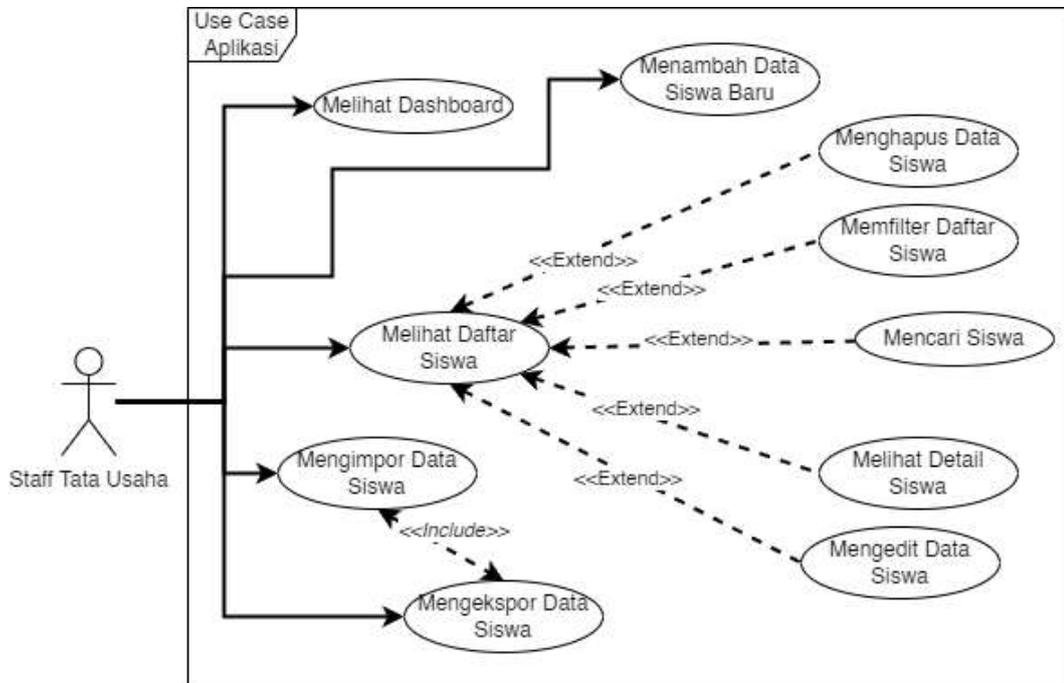
Selain itu, praktikan merancang alur tampilan antarmuka aplikasi sesuai dengan kebutuhan pengguna (staf administrasi sekolah) yang divisualisasikan pada Gambar 3.1. Upaya ini bertujuan untuk memastikan desain aplikasi selaras dengan ekspektasi dan kebutuhan pengguna.



Gambar 3.1 Alur Tampilan

Pada Gambar 3.1, dijelaskan alur tampilan aplikasi yang terdiri dari beberapa modul utama yang dapat diakses melalui *sidebar navigation*. Setiap pilihan menu akan menampilkan halaman yang sesuai di area konten utama. Selain membuat

alur tampilan aplikasi seperti yang disajikan pada Gambar 3.1, pada tahap selanjutnya praktikan mendeskripsikan interaksi antara pengguna dan sistem yang divisualisasikan pada Gambar 3.2.



Gambar 3.2 Use Case Diagram

Pada Gambar 3.2, praktikan memvisualisasikan interaksi antara pengguna (staf administrasi) dan sistem dalam bentuk *use case diagram*. *Use case diagram* ini menjelaskan fungsionalitas utama yang dapat dilakukan pengguna dalam sistem. Diagram tersebut akan mencakup *use case* seperti pengelolaan data siswa secara lengkap (memasukkan data siswa baru, melihat daftar siswa, melihat detail siswa, memperbaiki data siswa, dan menghapus data siswa), melihat ringkasan statistik melalui *dashboard*, melakukan impor data siswa dari berkas eksternal, dan mengekspor data siswa ke berkas Excel atau Word. Dalam menjelaskan setiap fungsi pada *use case diagram*, dijabarkan dalam bentuk tabel *use case description* sebagai berikut.

Tabel 3.2 Use Case Description: Melihat Dashboard

Use Case Name : Melihat Dashboard.	ID : UC1	Importance Level : Tinggi.
Primary Actor : Staff Tata Usaha		Use Case Type : Essential.
Stakeholder and Interest :		

- Staff Tata Usaha: Berkepentingan untuk mendapatkan ringkasan dan informasi statistik mengenai data siswa.
- Manajemen Sekolah: Membutuhkan visualisasi data.

Brief Description : Use case ini menjelaskan bagaimana Staff Tata Usaha dapat mengakses dan melihat halaman dashboard aplikasi yang menyajikan informasi ringkasan statistik

Trigger : Staff Tata Usaha memilih menu "Dashboard".

Type : *External Trigger.*

Relation Association : Staff Tata Usaha.

Relation Include : -

Relation Extend : -

Relation Generalization : -

Normal Flow of Events :

- Staff Tata Usaha navigasi ke Dashboard.
- Sistem mengambil serta menampilkan statistik dan grafik dari SQLite.
- Staff Tata Usaha dapat *refresh* data.

Subflows : -

Exceptional Flow :

- 1E. Kesalahan database/umum saat memuat data/grafik.
- 2E. Tidak ada data tahun valid untuk grafik.

Tabel 3.3 Use Case Description: Menambah Data Siswa Baru

Use Case Name : Menambah Data Siswa Baru.	ID : UC2	Importance Level : Tinggi.
Primary Actor : Staff Tata Usaha	Use Case Type : Essential.	
Stakeholder and Interest :		
<ul style="list-style-type: none"> • Staff Tata Usaha: Perlu memasukkan data siswa baru secara digital. • Manajemen Sekolah: Butuh data lengkap untuk administrasi. 		
Brief Description : Staff Tata Usaha memasukkan data siswa baru melalui formulir komprehensif ke sistem, menggantikan pencatatan manual.		
Trigger : Staff Tata Usaha memilih menu "Tambah siswa".		
Type : <i>External Trigger.</i>		
Association : Staff Tata Usaha.		
Include : -		

Extend : -

Generalization : -

Normal Flow of Events :

- Staff Tata Usaha memilih menu "Tambah Siswa".
 - Sistem menampilkan formulir kosong.
 - Staff Tata Usaha mengisi data pada field.
 - Staff Tata Usaha menekan "Simpan Data Siswa".
 - Sistem memvalidasi dan menyimpan data ke database SQLite; konfirmasi sukses lalu form diatur ulang.
-

Subflows : -

Exceptional Flow :

- 1E. Data tidak valid (kosong/format salah).
 - 2E. NIPD/NISN/NIK duplikat.
 - 3E. Pembatalan oleh pengguna.
-

Tabel 3.4 Use Case Description: Melihat Daftar Siswa

Use Case Name : Melihat Daftar Siswa.	ID : UC3	Importance Level : Tinggi.
Primary Actor : Staff Tata Usaha	Use Case Type : Essential.	

Stakeholder and Interest :

- Staff Tata Usaha: Perlu akses dan tinjauan cepat semua data siswa.
 - Manajemen Sekolah: Butuh akses mudah daftar siswa.
-

Brief Description : Staff Tata Usaha melihat daftar seluruh siswa dalam format tabel untuk tinjauan umum dan sebagai dasar operasi data.

Trigger : Staff Tata Usaha memilih menu "Daftar siswa".

Type : *External Trigger.*

Association : Staff Tata Usaha.

Include : -

Extend : -

Generalization : -

Normal Flow of Events :

- Staff Tata Usaha memilih menu "Daftar Siswa".
 - Sistem mengambil dan menampilkan tabel data siswa dari SQLite.
-

Subflows : -

Exceptional Flow :

- 1E. Daftar siswa kosong.
 - 2E. Kesalahan database saat memuat data.
-

Tabel 3.5: Use Case Description: Mengimpor Data Siswa

Use Case Name : Mengimpor Data Siswa.	ID : UC4	Importance Level : Tinggi.
Primary Actor : Staff Tata Usaha	Use Case Type : Essential.	
Stakeholder and Interest : <ul style="list-style-type: none">• Staff Tata Usaha: Perlu memasukkan data massal.• Manajemen Sekolah: Butuh memasukkan data dengan cepat.		
Brief Description : Staff Tata Usaha memasukkan data siswa massal dari berkas Excel/CSV ke sistem, dengan normalisasi header dan validasi per baris.		
Trigger : Kebutuhan input data siswa dalam jumlah besar dari sumber eksternal.		
Type : <i>External Trigger.</i>		
Association : Staff Tata Usaha.		
Include : -		
Extend : -		
Generalization : -		
Normal Flow of Events : <ul style="list-style-type: none">• Staff Tata Usaha memilih "Import/Export" lalu "Impor Data dari File".• Memilih berkas lalu konfirmasi impor.• Sistem membaca berkas (pandas), normalisasi header, validasi baris, dan INSERT data valid ke SQLite.• Sistem menampilkan ringkasan hasil impor.		
Subflows : -		
Exceptional Flow : <ul style="list-style-type: none">1E. Pembatalan pengguna.2E. Format/file kosong tidak didukung.3E. Kolom wajib tidak ditemukan.4E. Data tidak valid/duplikat dalam berkas (dilewati & dilaporkan).5E. Kesalahan koneksi database.		

Tabel 3.6 Use Case Description: Mengekspor Data Siswa

Use Case Name : Mengekspor Data Siswa.	ID : UC5	Importance Level : Medium.
Primary Actor : Staff Tata Usaha		Use Case Type : Essential.
Stakeholder and Interest :		
<ul style="list-style-type: none"> • Staff Tata Usaha: Perlu salinan data siswa untuk keperluan eksternal atau <i>backup</i>. • Manajemen Sekolah: Butuh data untuk pelaporan atau <i>backup</i>. 		
Brief Description : Staff Tata Usaha mengunduh seluruh data siswa dari sistem ke berkas Excel untuk backup atau pelaporan eksternal.		
Trigger : Kebutuhan salinan data siswa di luar aplikasi.		
Type : <i>External Trigger</i> .		
Association : Staff Tata Usaha.		
Include : -		
Extend : -		
Generalization : -		
Normal Flow of Events :		
<ul style="list-style-type: none"> • Staff Tata Usaha memilih "Import/Export" lalu "Ekspor Semua Data ke Excel". • Sistem mengambil seluruh data dari SQLite, menyiapkan dalam Excel (pandas). • Menampilkan dialog "Simpan Sebagai". • Staff Tata Usaha menentukan lokasi dan nama berkas, lalu klik "Simpan". • Sistem menyimpan berkas dan memberi konfirmasi. 		
Subflows : -		
Exceptional Flow :		
1E. Pembatalan pengguna.		
2E. Data siswa kosong.		
3E. Kesalahan database/umum saat ekspor.		

Tabel 3.7 Use Case Description: Menghapus Data Siswa

Use Case Name : Menghapus Data Siswa.	ID : UC6	Importance Level : High.
Primary Actor : Staff Tata Usaha		Use Case Type : Detail, Essential.

Stakeholder and Interest :

- Staff Tata Usaha: Perlu menghapus data tidak relevan/valid.
- Manajemen Sekolah: Butuh database bersih.

Brief Description : Staff Tata Usaha menghapus data siswa tertentu dari daftar, dengan konfirmasi untuk integritas data.

Trigger : Data siswa tidak lagi relevan/valid.**Type :** *External Trigger.*

Association : Staff Tata Usaha.**Include :** -**Extend :** Melihat Daftar Siswa.**Generalization :** -

Normal Flow of Events :

- Staff Tata Usaha di "Daftar Siswa", pilih siswa.
- Tekan "Hapus Siswa".
- Sistem tampilkan konfirmasi penghapusan (NIPD & nama siswa).
- Staff Tata Usaha konfirmasi.
- Sistem menjalankan SQL DELETE ke SQLite; konfirmasi sukses lalu daftar dimuat ulang.

Subflows : -

Exceptional Flow :

- 1E. Tidak ada siswa terpilih.
- 2E. Pembatalan pengguna.
- 3E. Siswa tidak ditemukan saat dihapus.
- 4E. Kesalahan database/umum.

Tabel 3.8 Use Case Description: Memfilter Data Siswa

Use Case Name : Memfilter Data Siswa.	ID : UC7	Importance Level : Medium.
Primary Actor : Staff Tata Usaha	Use Case Type : Supporting.	

Stakeholder and Interest :

- Staff Tata Usaha: Perlu menyaring daftar siswa berdasarkan kategori.

Brief Description : Staff Tata Usaha menyaring daftar siswa berdasarkan kriteria pada modul "Daftar Siswa".

Trigger : Kebutuhan melihat kelompok siswa tertentu.

Type : *External Trigger.*

Association : Staff Tata Usaha.

Include : -

Extend : Melihat Daftar Siswa.

Generalization : -

Normal Flow of Events :

- Staff Tata Usaha di "Daftar Siswa".
- Pilih kriteria filter dari Combobox "Filter Status".
- Sistem menjalankan SQL SELECT dengan WHERE status; daftar diperbarui.

Subflows : -

Exceptional Flow :

1E. Tidak ada siswa yang cocok.

2E. Kesalahan database/umum.

Tabel 3.9 Use Case Description: Mencari Siswa

Use Case Name : Mencari Siswa.	ID : UC8	Importance Level : Medium.
Primary Actor : Staff Tata Usaha		Use Case Type : Supporting.
Stakeholder and Interest :		
<ul style="list-style-type: none"> • Staff Tata Usaha: Perlu menemukan siswa spesifik dengan cepat. 		
Brief Description : Staff Tata Usaha menemukan siswa tertentu di daftar menggunakan kata kunci (Nama, NIPD, atau NISN).		
Trigger : Kebutuhan menemukan siswa spesifik.		
Type : <i>External Trigger.</i>		
Association : Staff Tata Usaha.		
Include : -		
Extend : Melihat Daftar Siswa.		
Generalization : -		
Normal Flow of Events :		
<ul style="list-style-type: none"> • Staff Tata Usaha di "Daftar Siswa". • Masukkan kata kunci di field "Cari". • Tekan "Cari" atau Enter. • Sistem menjalankan SQL SELECT; daftar diperbarui. 		

Subflows : -
Exceptional Flow :
1E. Siswa tidak ditemukan.
2E. Kesalahan database/umum.

Tabel 3.10 Use Case Description: Melihat Detail Siswa

Use Case Name : Melihat Detail Siswa.	ID : UC9	Importance Level : Medium.
Primary Actor : Staff Tata Usaha	Use Case Type : Supporting.	
Stakeholder and Interest :		
<ul style="list-style-type: none"> Staff Tata Usaha: Perlu informasi lengkap siswa terpilih. 		
Brief Description : Staff Tata Usaha melihat seluruh informasi rinci siswa terpilih pada "Halaman Detail Siswa", awalnya bersifat <i>read-only</i> .		
Trigger : Kebutuhan memeriksa data lengkap siswa.		
Type : <i>External Trigger</i> .		
Association : Staff Tata Usaha.		
Include : -		
Extend : Melihat Daftar Siswa.		
Generalization : -		
Normal Flow of Events :		
<ul style="list-style-type: none"> Staff Tata Usaha di "Daftar Siswa", pilih siswa. Sistem mengambil dan menampilkan detail siswa dari SQLite di "Halaman Detail Siswa"; judul halaman diperbarui. 		
Subflows : -		
Exceptional Flow :		
1E. Tidak ada siswa terpilih.		
2E. Data detail tidak ditemukan.		
3E. Kesalahan database/umum.		

Tabel 3.11 Use Case Description: Mengedit Data Siswa

Use Case Name : Mengedit Data Siswa.	ID : UC10	Importance Level : High.
Primary Actor : Staff Tata Usaha	Use Case Type : Detail, Essential.	
Stakeholder and Interest :		

-
- Staff Tata Usaha: Perlu memperbarui data siswa yang ada.
 - Manajemen Sekolah: Butuh data terkini.
-

Brief Description : Staff Tata Usaha mengubah data siswa melalui "Halaman Detail Siswa" dengan mengaktifkan mode edit dan menyimpan perubahan.

Trigger : Perubahan data siswa atau koreksi kesalahan input.

Type : *External Trigger.*

Association : Staff Tata Usaha.

Include : -

Extend : Melihat Daftar Siswa.

Generalization : -

Normal Flow of Events :

- Staff Tata Usaha di "Halaman Detail Siswa".
 - Tekan "Edit Data Siswa"; field jadi dapat diedit, "Simpan perubahan" aktif; tombol "Edit Data Siswa" jadi "Batal Edit".
 - Staff Tata Usaha mengubah data.
 - Tekan "Simpan perubahan"; konfirmasi.
 - Sistem memvalidasi dan menjalankan SQL UPDATE ke SQLite; konfirmasi sukses, mode edit nonaktif.
-

Subflows : -

Exceptional Flow :

- 1E. Data input tidak valid (misal: duplikat NIK/NISN, format tanggal).
 - 2E. Pembatalan edit (tekan "Batal Edit" atau tidak simpan).
 - 3E. Tidak ada perubahan.
 - 4E. Kesalahan database/umum.
 - 5E. Ekspor ke Word dari detail siswa.
-

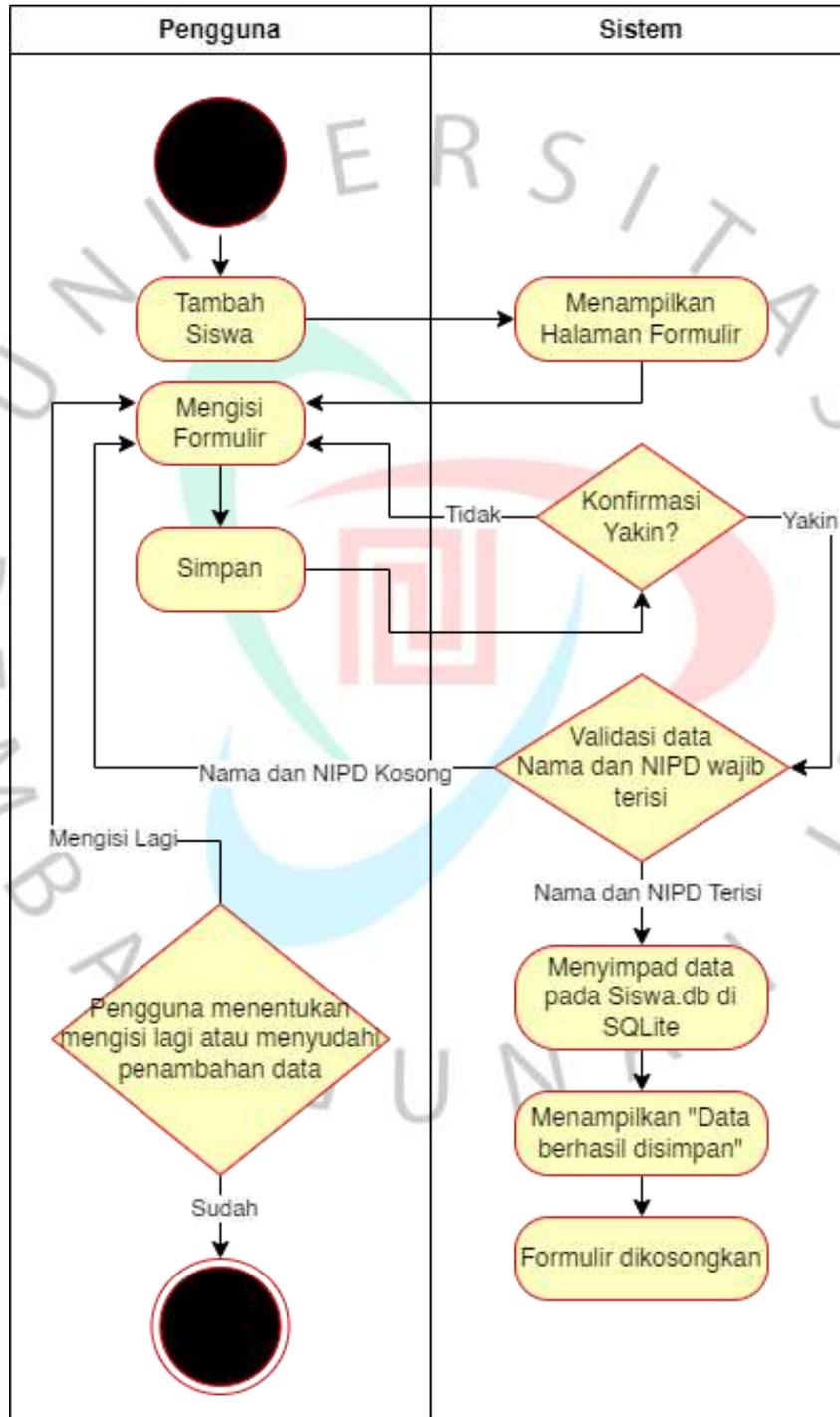
Setelah menguraikan fungsionalitas sistem melalui *Use Case Description* yang menjelaskan interaksi pengguna dengan aplikasi, langkah selanjutnya dalam perancangan sistem adalah mendefinisikan struktur data yang menjadi penyimpanan informasi. Basis data ini berperan dalam menyimpan, mengelola, dan memastikan integritas seluruh data siswa, mulai dari identitas pribadi, riwayat pendidikan, hingga informasi terkait orang tua/wali. Diagram basis data berikut akan memvisualisasikan entitas dan atribut operasional aplikasi.

siswa	
nipd	VARCCHAR(20)
nama_lengkap	VARCCHAR(255)
nim	VARCCHAR(20)
jenis_kelamin	VARCCHAR(15)
tempat_lahir	VARCCHAR(100)
tanggal_lahir	DATE
agama	VARCCHAR(50)
anak_ke	INTEGER
jumlah_saudara_kandung	INTEGER
jumlah_saudara_tiri	INTEGER
jumlah_saudara_angkat	INTEGER
status_yatm	VARCCHAR(20)
nik	VARCCHAR(20)
no_kk	VARCCHAR(30)
alamat	TEXT
rt_rw	VARCCHAR(10)
kelurahan	VARCCHAR(100)
kecamatan	VARCCHAR(100)
kota_kabupaten	VARCCHAR(100)
provinsi	VARCCHAR(100)
kode_pos	VARCCHAR(10)
nama_ayah	VARCCHAR(255)
pekerjaan_ayah	VARCCHAR(100)
no_telp_ayah	VARCCHAR(20)
alamat_tinggal_ayah	TEXT
nama_bu	VARCCHAR(255)
pekerjaan_bu	VARCCHAR(100)
no_telp_bu	VARCCHAR(20)
alamat_tinggal_bu	TEXT
nama_wali	VARCCHAR(255)
pekerjaan_wali	VARCCHAR(100)
no_telp_wali	VARCCHAR(20)
alamat_tinggal_wali	TEXT
status	VARCCHAR(20)
tahun_masuk	INTEGER
asal_sekolah	VARCCHAR(255)
tanggal_lulus_pindah_keluar	DATE
alasan_lulus_pindah_keluar	TEXT
melanjutkan_ke	TEXT
catatan_tambahan	TEXT

Gambar 3.3 Diagram Database

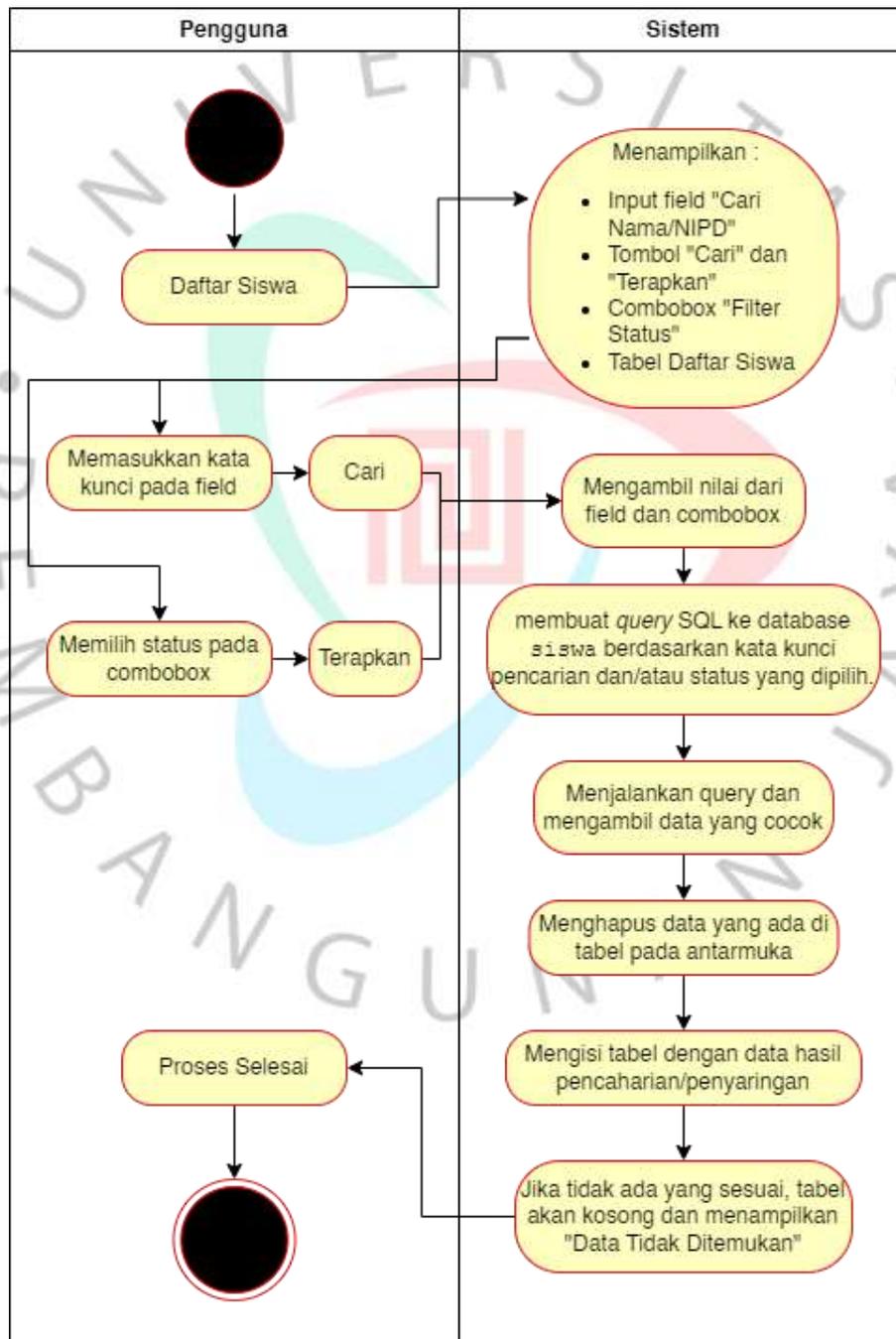
Setelah menguraikan struktur penyimpanan data melalui diagram basis data, fokus selanjutnya adalah memvisualisasikan bagaimana data-data tersebut diproses dan bagaimana pengguna berinteraksi dengan sistem melalui serangkaian aktivitas. Bagian ini akan menjelaskan alur kerja operasional aplikasi

untuk setiap fungsionalitas utama. Diagram aktivitas akan memberikan gambaran yang jelas mengenai langkah-langkah yang terlibat, baik dari sisi pengguna (Staff Tata Usaha) maupun respons sistem, dalam menjalankan proses bisnis pengelolaan data siswa secara digital. Pemahaman terhadap alur ini krusial untuk memastikan bahwa setiap fitur aplikasi berfungsi secara koheren dan efisien.



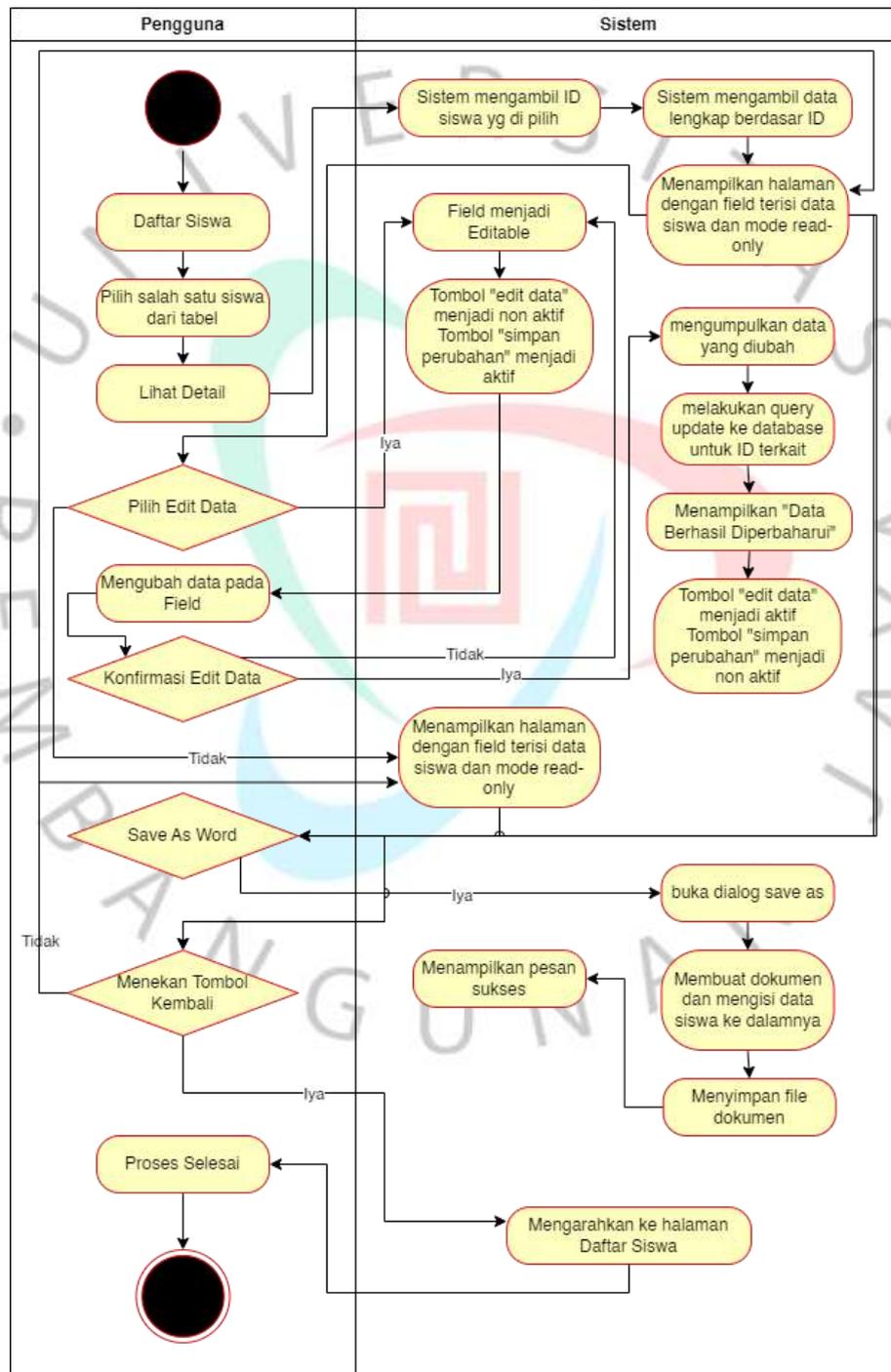
Gambar 3.4 Activity Diagram Input Data Siswa Baru

Gambar 3.4 merupakan *activity diagram* yang menggambarkan alur proses saat data siswa baru di *input* ke dalam sistem melalui modul "Tambah Siswa". Proses dimulai ketika pengguna memilih menu untuk menambahkan siswa. Sistem akan menampilkan formulir dengan berbagai *field* yang harus diisi, mencakup data pribadi, data kependudukan, data orang tua/wali, dan keterangan pendidikan siswa. Setelah semua data terisi, pengguna menyimpan data tersebut, dan sistem akan memvalidasi serta memasukkannya ke dalam basis data SQLite.



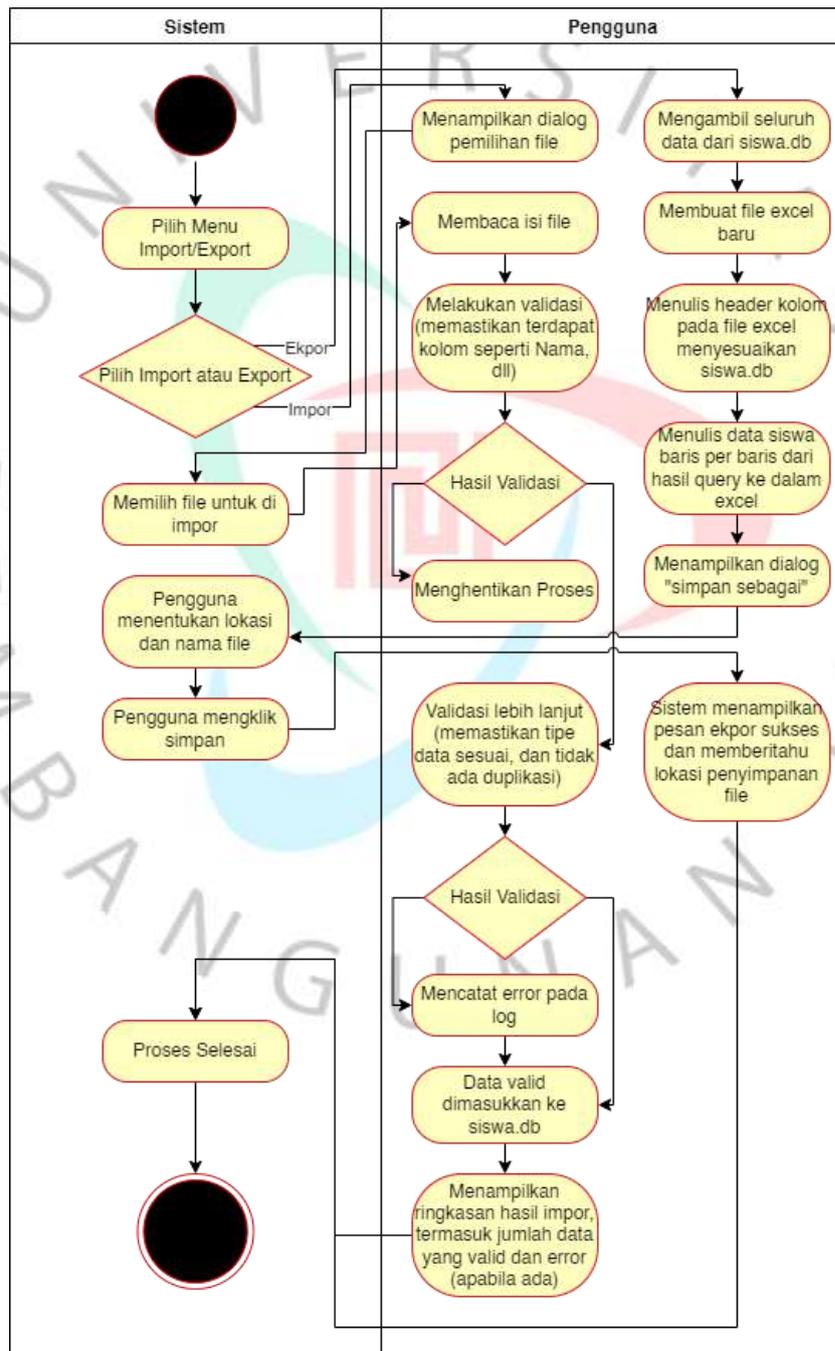
Gambar 3.5 Activity Diagram Pencarian dan Penyaringan Data Siswa

Pada Gambar 3.5 merupakan *activity diagram* yang menjelaskan alur proses ketika pengguna melakukan pencarian atau penyaringan data siswa pada modul "Daftar Siswa". Pengguna dapat memasukkan kata kunci pencarian (nama atau NIPD) atau memilih kriteria filter (status siswa). Sistem kemudian akan melakukan *query* ke basis data berdasarkan kriteria yang diberikan dan menampilkan hasil yang relevan dalam bentuk tabel pada antarmuka pengguna.



Gambar 3.6 Activity Diagram Pengelolaan Detail Siswa

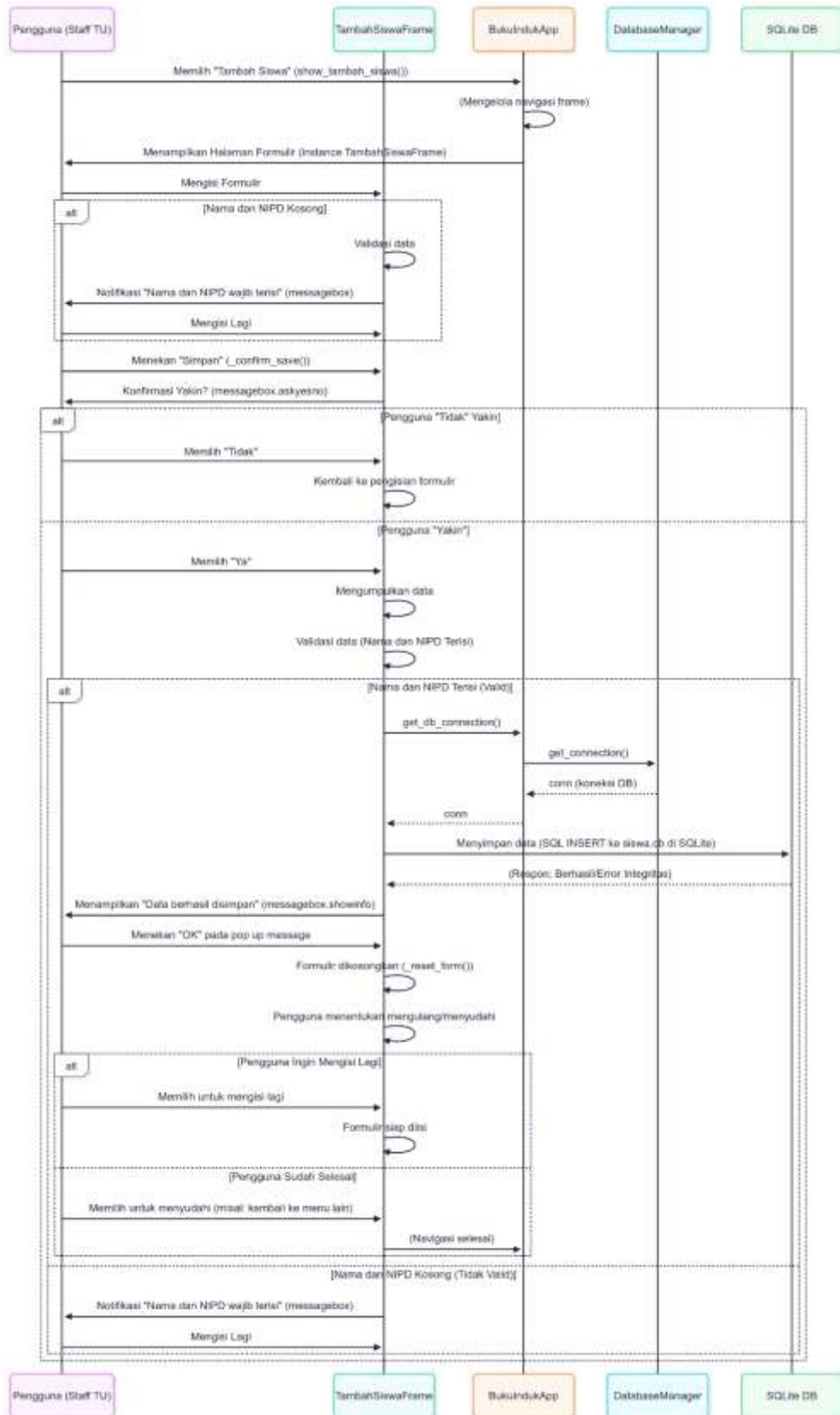
Gambar 3.6 merupakan *activity diagram* pada proses pengelolaan detail siswa. Proses ini dimulai ketika pengguna perlu memilih satu siswa dari daftar untuk melihat detailnya. Sistem akan menampilkan semua informasi terkait siswa tersebut dalam tampilan detail. Pengguna memiliki opsi untuk mengubah data siswa. Jika ada perubahan data, pengguna dapat menyimpan perubahan tersebut ke basis data. Pengguna juga dapat mengekspor data detail siswa tersebut ke dalam format dokumen Microsoft Word untuk keperluan arsip atau pelaporan.



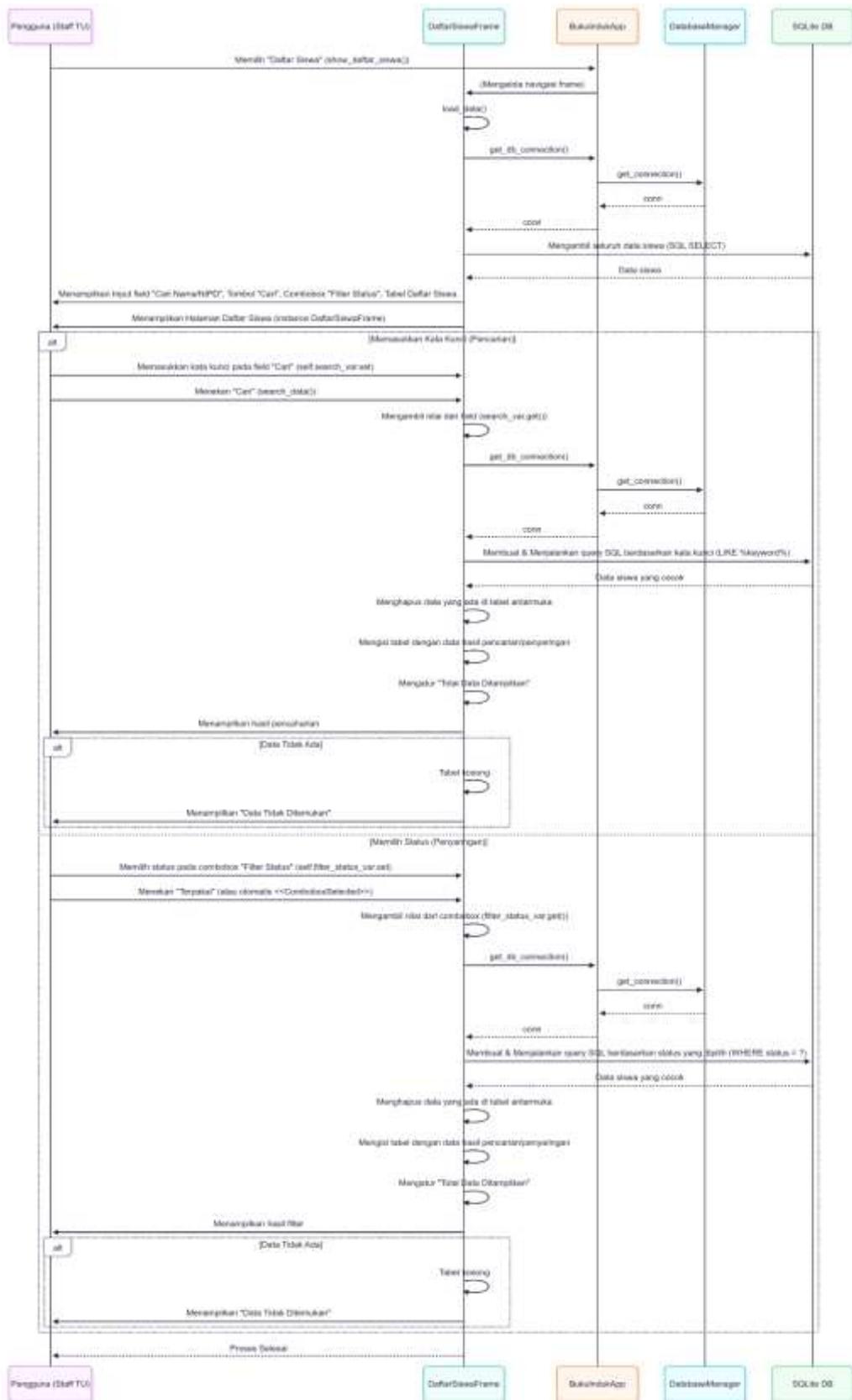
Gambar 3.7 Activity Diagram Impor dan Ekspor Data Siswa

Gambar 3.7 ini menunjukkan bagaimana sistem menangani pemasukan dan pengeluaran data siswa dalam jumlah besar, yaitu melalui fitur impor dan ekspor. Saat kita ingin mengimpor data, misalnya memasukkan banyak data siswa sekaligus, pengguna perlu memilih berkas (seperti Excel atau CSV) yang sudah berisi data tersebut. Sistem akan memeriksa apakah format berkasnya sudah benar dan apakah isi datanya sudah sesuai. Apabila sudah sesuai, data siswa tersebut akan langsung masuk ke dalam database aplikasi. Jika ada data yang salah, sistem akan memunculkan *message box*. Sementara itu, apabila pengguna perlu mengekspor data. Pengguna perlu memilih opsi ekspor. Sistem akan mengambil semua data siswa yang ada, lalu menyiapkannya dalam bentuk berkas Excel. Setelah itu, pengguna dapat memilih mau disimpan di mana dan dengan nama apa berkas tersebut.

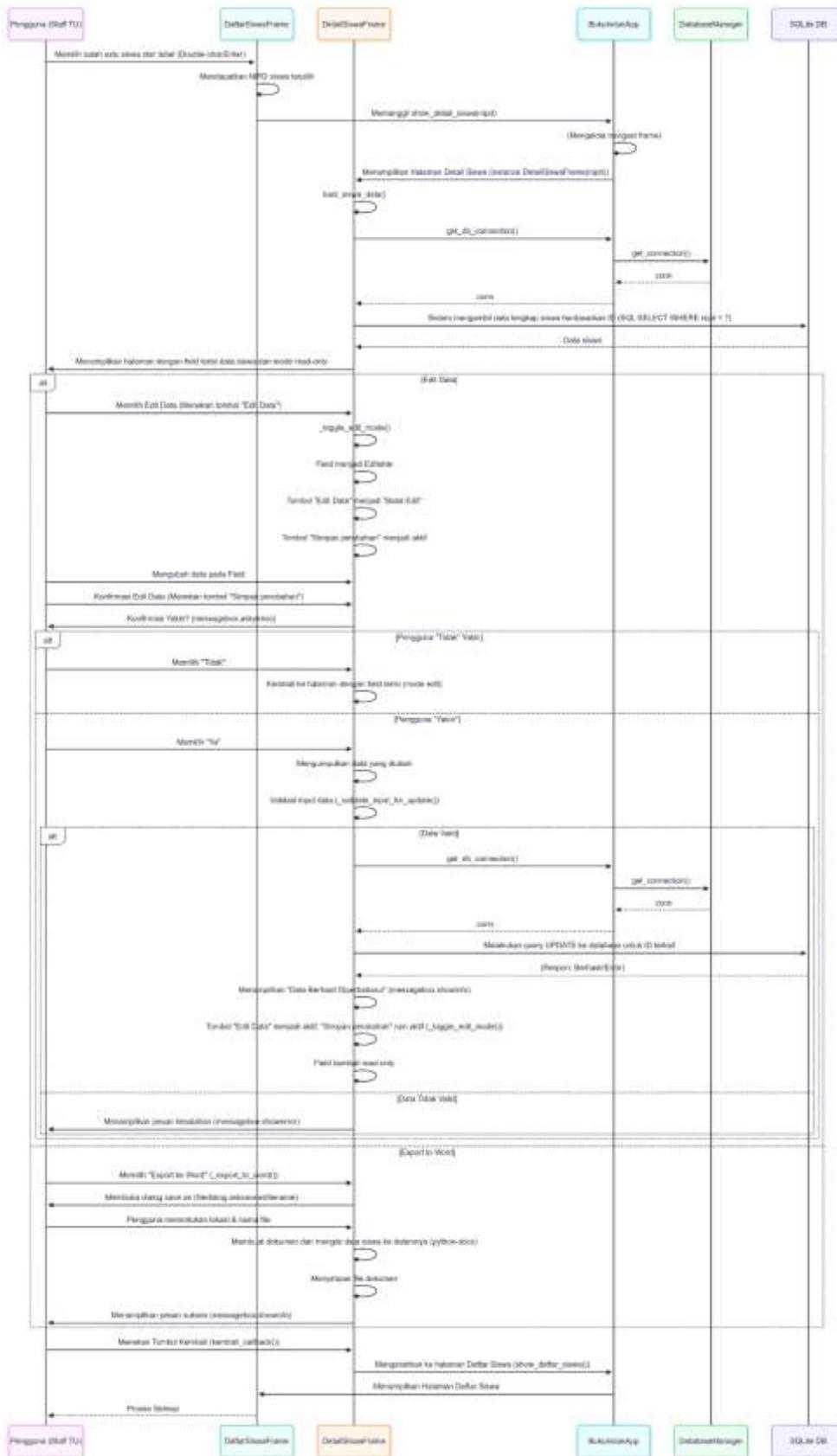
Setelah memetakan alur proses bisnis dan aktivitas sistem secara umum melalui *Activity Diagram*, langkah berikutnya adalah menganalisis dan memvisualisasikan detail interaksi antar komponen dalam sistem. Bagian ini akan menyajikan *Sequence Diagram* untuk setiap fungsionalitas kunci yang telah diidentifikasi. *Sequence Diagram* ini akan memperjelas urutan pesan yang dipertukarkan antara aktor (Staff Tata Usaha) dan berbagai objek internal sistem, seperti antarmuka pengguna, logika aplikasi, dan basis data, seiring waktu. Dengan demikian, *Sequence Diagram* akan memberikan gambaran yang lebih rinci tentang bagaimana setiap aktivitas dalam sistem dieksekusi melalui kolaborasi antar objek.



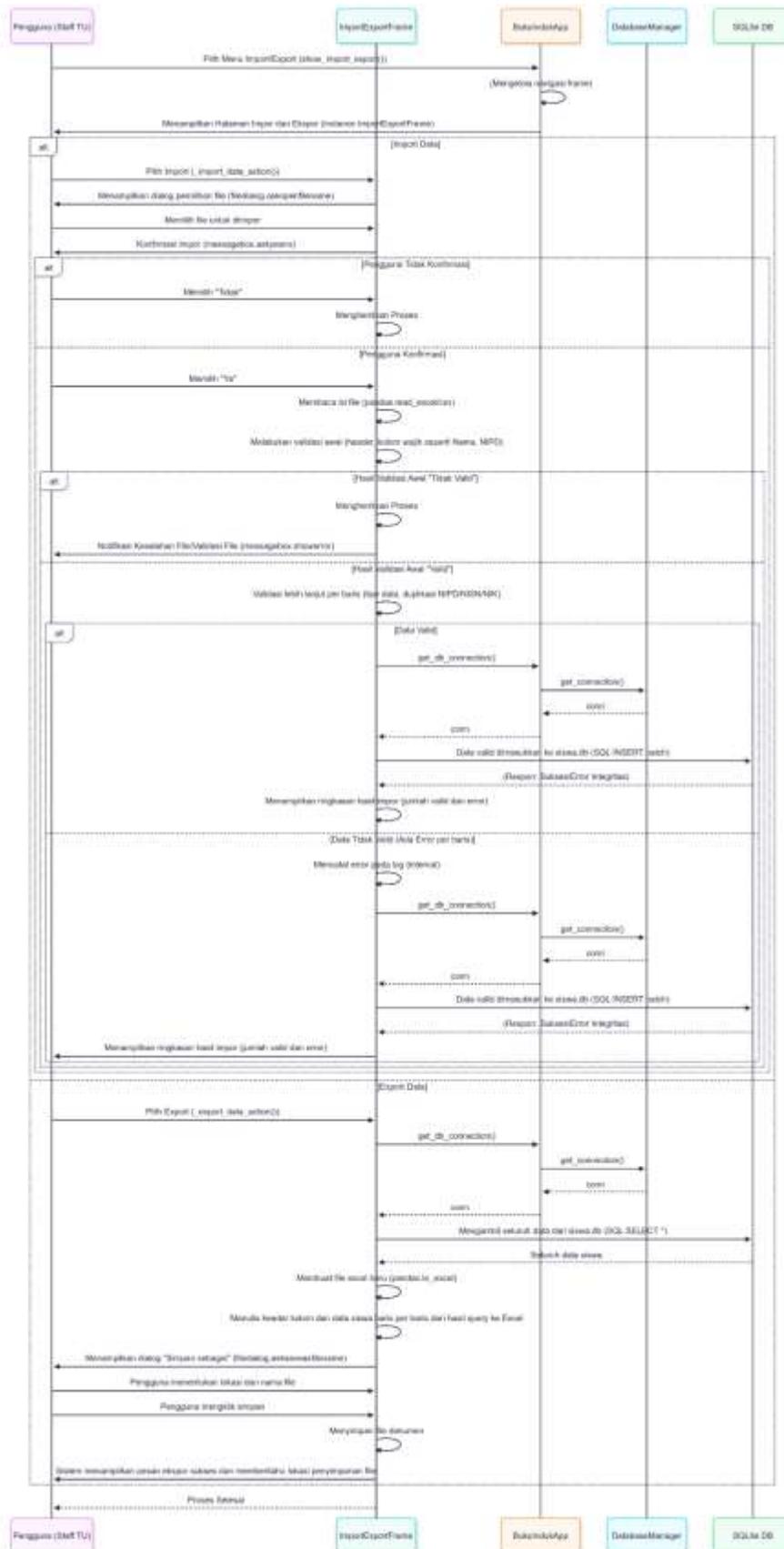
Gambar 3.8 Sequence Diagram: Input Data Siswa Baru



Gambar 3.9 Sequence Diagram: Pencarian dan Penyaringan Data Siswa



Gambar 3.10 Sequence Diagram: Pengelolaan Detail Siswa



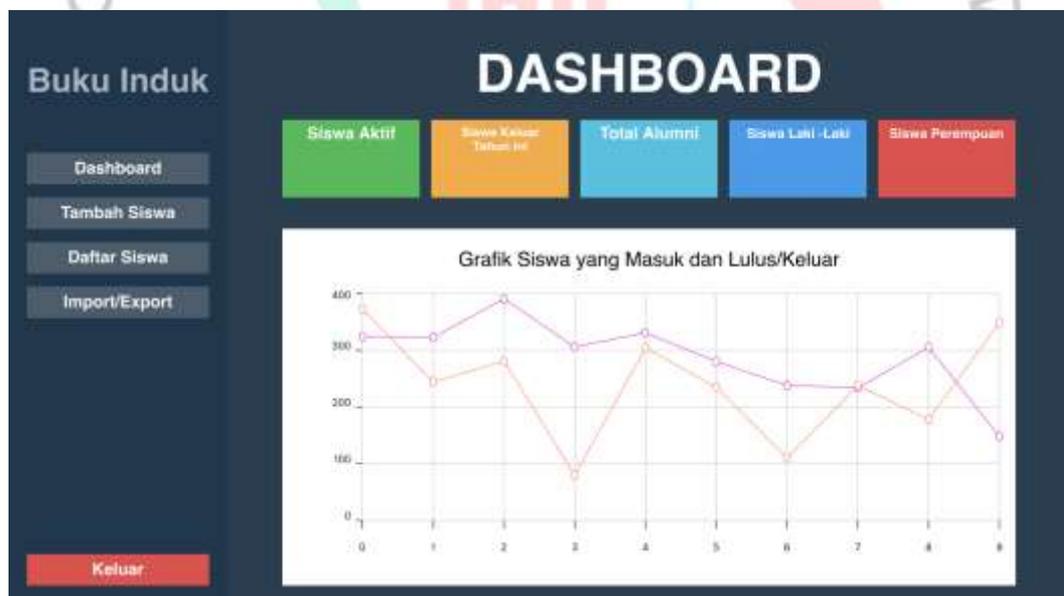
Gambar 3.11 Sequence Diagram: Ekspor Impor Data

Secara keseluruhan, *Sequence Diagram* yang telah dipaparkan memberikan gambaran yang transparan mengenai interaksi dinamis antara Staf Tata Usaha dan komponen-komponen internal aplikasi. Pemahaman tentang alur pesan ini untuk memastikan setiap fungsionalitas sistem bekerja sesuai harapan dan meminimalkan potensi masalah integrasi antar modul. Diagram-diagram ini menjadi panduan penting dalam fase implementasi dan pengujian, memastikan bahwa kode yang ditulis merefleksikan perilaku sistem yang telah dirancang.

3.2.2.2. Perancangan Antarmuka

Tahap perancangan antarmuka pengguna dilakukan dengan membuat *mockup* untuk setiap fungsionalitas utama sistem. *Mockup* ini bertujuan untuk memberikan representasi visual awal dari aplikasi "Buku Induk Digital Sekolah", sehingga memudahkan pemahaman alur kerja sistem dan interaksi pengguna dengan fitur-fitur yang dikembangkan. Pertimbangan utama dalam perancangan *mockup* adalah kemudahan penggunaan, kejelasan informasi, dan konsistensi desain antar halaman.

3.2.2.2.1. Halaman Dashboard



Gambar 3.12 Mockup Halaman Dashboard

Gambar 3.12 menampilkan mockup untuk Halaman Dashboard. Halaman ini berfungsi sebagai tampilan awal setelah pengguna berhasil masuk ke dalam sistem, menyajikan ringkasan informasi penting secara visual. Pada bagian atas halaman, ditampilkan judul "DASHBOARD" dengan ukuran font yang menonjol. Di bawahnya, terdapat beberapa kartu (card) statistik yang menampilkan data

agregat seperti jumlah "Siswa Aktif", "Siswa Keluar Tahun Ini", "Total Alumni", "Siswa Laki-Laki", dan "Siswa Perempuan". Setiap kartu statistik dirancang dengan warna latar yang berbeda untuk membedakan informasi dan menampilkan angka statistik dengan jelas.

Bagian utama halaman Dashboard dialokasikan untuk menampilkan "Grafik Siswa yang Masuk dan Lulus/Keluar". Grafik ini bertujuan untuk memvisualisasikan tren pendaftaran siswa baru dan siswa yang telah menyelesaikan studi atau keluar dari sekolah selama periode waktu tertentu. Di sisi kiri halaman, terdapat panel navigasi vertikal utama yang berisi menu "Buku Induk" dan tombol-tombol navigasi seperti "Dashboard", "Tambah Siswa", "Daftar Siswa", "Import/Export", serta tombol "Keluar" di bagian paling bawah panel. Panel navigasi ini dirancang agar selalu terlihat dan memudahkan pengguna untuk berpindah antar modul utama aplikasi.

3.2.2.2.2. Halaman Tambah Data Siswa



Gambar 3.13 Halaman Tambah Data Siswa

Gambar 3.13 merupakan *mockup* Halaman Tambah Data Siswa. Halaman ini dirancang untuk memfasilitasi proses input data siswa baru ke dalam sistem secara sistematis. Judul halaman "Tambah Data Siswa" ditampilkan dengan jelas di bagian atas. Di bawah judul, terdapat serangkaian field input yang terstruktur untuk berbagai kategori data siswa, seperti "Nama", "NIPD", "NISN", "Jenis Kelamin" (menggunakan dropdown), "Tempat Lahir", "Tanggal Lahir", "Agama",

"Anak Keberapa", "Jumlah Saudara Kandung", "Jumlah Saudara Tiri", dan "Status Anak" (menggunakan dropdown).

Setiap field input terdiri dari label yang deskriptif di sebelah kiri dan area input di sebelah kanan. Tata letak ini bertujuan untuk memudahkan pengguna dalam mengisi informasi yang diperlukan. Di sisi kanan atas formulir, terdapat tombol "Simpan" untuk menyimpan data yang telah diinput ke database dan tombol "Reset" untuk mengosongkan semua field input. Panel navigasi utama di sisi kiri tetap konsisten tersedia untuk akses ke modul lain. Desain halaman ini mengutamakan kelengkapan data dan efisiensi proses entri.

3.2.2.2.3. Halaman Daftar Siswa



Gambar 3.14 Mockup Halaman Daftar Siswa

Gambar 3.14 adalah *mockup* Halaman Daftar Siswa. Halaman ini berfungsi untuk menampilkan daftar siswa yang tersimpan dalam database dan menyediakan fungsionalitas pencarian serta penyaringan data. Judul halaman "Daftar Siswa" terletak di bagian atas. Di bawahnya, terdapat fitur "Cari (Nama/NIPD/NISN)" yang terdiri dari label, kolom input teks untuk kata kunci pencarian, dan tombol "Cari". Di sebelah kanan fitur pencarian, terdapat opsi "Filter" dengan komponen dropdown yang memungkinkan pengguna untuk menyaring data siswa berdasarkan kriteria tertentu (misalnya status siswa).

Bagian utama halaman ini akan menampilkan data siswa dalam format tabel, dengan header kolom yang terlihat jelas seperti "NAMA", "NIPD", dan "NISN".

Meskipun konten tabel belum terisi pada *mockup*, area ini dirancang untuk menampilkan data secara terstruktur sehingga mudah dibaca dan dikelola oleh pengguna. Panel navigasi di sisi kiri juga tetap ditampilkan untuk konsistensi antarmuka.

3.2.2.2.4. Halaman Detail Siswa



Gambar 3.15 Halaman Detail Siswa

Gambar 3.15 menampilkan *mockup* untuk Halaman Detail Siswa. Halaman ini bertujuan untuk menyajikan informasi lengkap mengenai satu siswa tertentu yang dipilih dari Halaman Daftar Siswa, serta menyediakan opsi untuk mengedit data dan mengekspor data siswa. Judul "Detail Siswa" ditampilkan di bagian atas. Serupa dengan Halaman Tambah Data Siswa, halaman ini menampilkan field-field data siswa seperti "Nama", "NIPD", "NISN", "Jenis Kelamin", dan seterusnya, dalam format label dan area tampilan data (yang pada awalnya bersifat *read-only*).

Di bagian bawah area detail data, terdapat tombol-tombol aksi: "Edit Data" untuk mengubah field menjadi dapat diedit, "Simpan perubahan" untuk menyimpan modifikasi data, dan "Export ke Word" untuk mengekspor informasi siswa ke dalam format dokumen. Panel navigasi sisi kiri tetap tersedia untuk memudahkan perpindahan antar modul. Desain ini memungkinkan pengguna untuk melihat data secara komprehensif sekaligus melakukan manajemen data individual siswa.

3.2.2.2.5. Halaman Impor dan Ekspor



Gambar 3.16 Mockup Halaman Import/Export

Gambar 3.16 merupakan *mockup* untuk Halaman Import/Export. Halaman ini menyediakan fungsionalitas bagi pengguna untuk melakukan impor data siswa secara massal dari berkas eksternal atau mengekspor keseluruhan data siswa dari sistem ke dalam sebuah berkas. Judul halaman "Import/Export" ditempatkan secara jelas di bagian atas.

Konten utama halaman ini terdiri dari dua tombol aksi utama: "Import Data" dan "Export Data". Tombol "Import Data" akan memicu proses untuk memilih berkas (misalnya, format Excel atau CSV) yang berisi data siswa untuk dimasukkan ke dalam database sistem. Tombol "Export Data" akan memulai proses untuk mengambil semua data siswa dari database dan menyimpannya dalam format berkas yang dapat diunduh (misalnya, Excel). Desain halaman ini dibuat sederhana dan langsung pada tujuan untuk memudahkan pengguna dalam melakukan operasi transfer data. Seperti halaman lainnya, panel navigasi di sisi kiri tetap ada.

3.2.3. Pengembangan Prototyping

Tahapan prototyping sistem dimulai setelah proses perancangan detail, termasuk perancangan antarmuka pengguna, selesai dilakukan. Fokus utama pada tahap ini adalah menerjemahkan rancangan konseptual dan visual menjadi sebuah aplikasi desktop yang fungsional dan interaktif. Implementasi sistem "Buku Induk Digital Sekolah" ini dibagi menjadi dua bagian utama: pengembangan

antarmuka pengguna (front-end) dan logika aplikasi serta manajemen data (back-end), yang diuraikan sebagai berikut.

1. Teknologi dan Kerangka Kerja

Aplikasi ini dibangun menggunakan bahasa Python. Antarmuka pengguna dikembangkan dengan pustaka Tkinter, dan menggunakan *framework* ttkbootstrap untuk tampilan visual modern dan komponen widget yang baik. Dalam manajemen data, aplikasi ini menggunakan sistem database SQLite. Logika back-end untuk operasi data (CRUD), validasi, serta proses impor/ekspor juga diimplementasikan dengan Python, didukung oleh pustaka pandas untuk penanganan berkas eksternal.

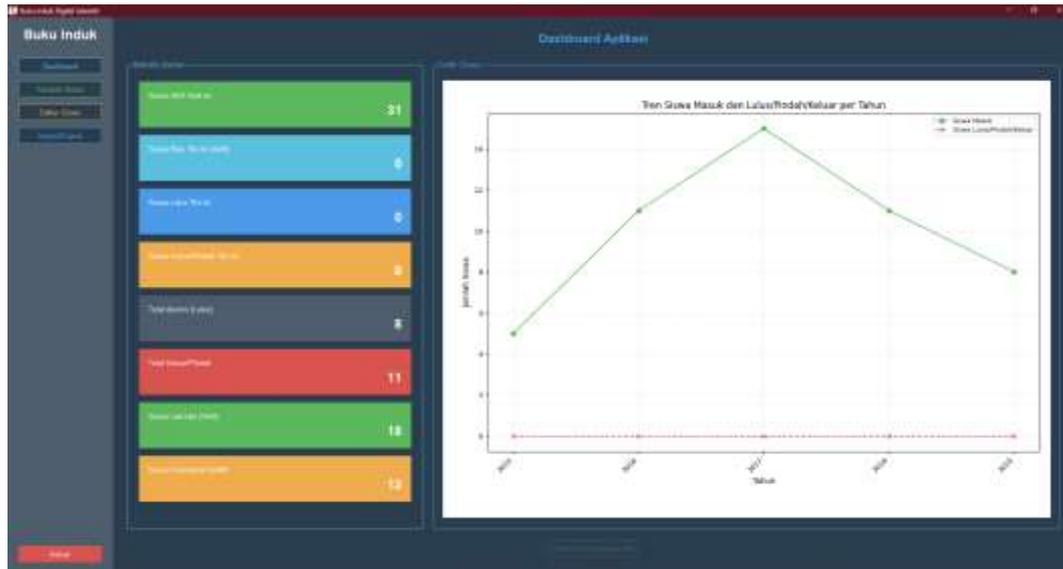
2. Struktur Aplikasi

Aplikasi disusun secara modular, dengan setiap fungsionalitas utama diimplementasikan sebagai kelas Python tersendiri. Dengan menggunakan satu kelas utama yang berfungsi sebagai inti aplikasi yang mengatur navigasi antar modul dan menyediakan akses terpusat ke database.

3. Integrasi Front-End dan Back-End

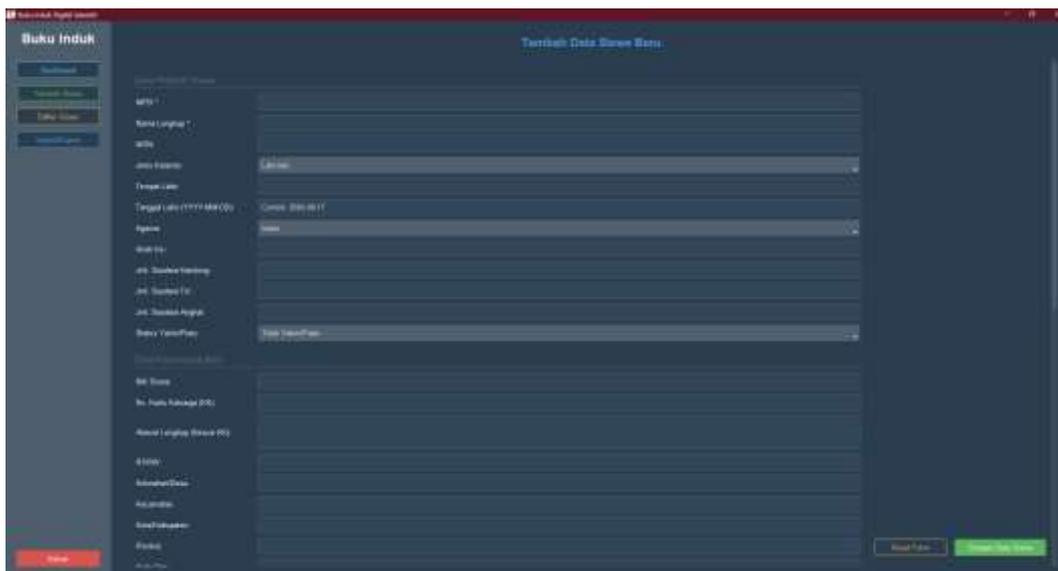
Integrasi kedua sisi dicapai melalui mekanisme *event binding*, di mana aksi pengguna pada antarmuka memicu fungsi back-end. Data mengalir dua arah dari input pengguna ke database, dan dari database ke tampilan antarmuka. Manajemen state antarmuka dan umpan balik kepada pengguna diimplementasikan untuk mencerminkan hasil operasi dan status sistem.

Proses implementasi sistem menghasilkan serangkaian antarmuka pengguna yang fungsional dan intuitif, dirancang untuk memenuhi kebutuhan pengelolaan data siswa secara digital. Setiap halaman utama aplikasi memiliki peran spesifik dalam alur kerja aplikasi "Buku Induk Digital Sekolah".



Gambar 3.17 Tampilan Halaman Dashboard

Gambar 3.17 menampilkan antarmuka Halaman Dashboard yang berfungsi sebagai pusat informasi visual utama aplikasi. Halaman ini menyajikan ringkasan statistik kunci seperti jumlah siswa aktif, siswa yang keluar pada tahun berjalan, total alumni, serta komposisi siswa laki-laki dan perempuan. Untuk memberikan gambaran dinamika populasi siswa, sebuah grafik tren yang memvisualisasikan data siswa masuk dan lulus/keluar per tahun juga ditampilkan secara menonjol. Seluruh data statistik dan grafik ini dihasilkan melalui query ke database SQLite yang mengambil dan mengagregasi informasi siswa secara *real-time* saat halaman dimuat atau ketika pengguna melakukan *refresh*. Terdapat integrasi dengan pustaka matplotlib untuk visualisasi data tren dalam bentuk grafik yang disematkan langsung ke dalam antarmuka Tkinter. Panel navigasi di sisi kiri halaman, yang konsisten di seluruh aplikasi, memastikan akses cepat ke modul-modul fungsional lainnya.



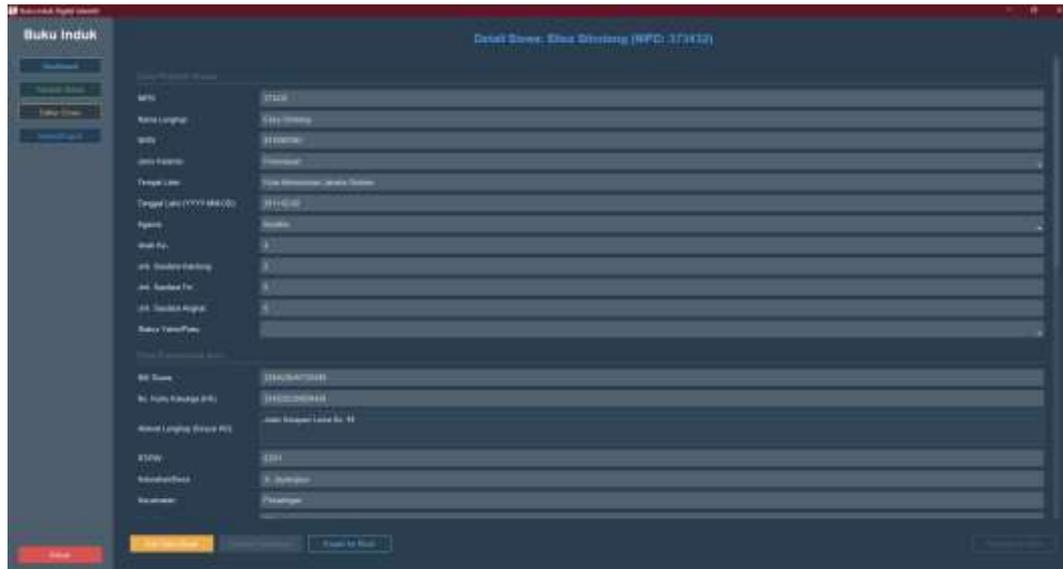
Gambar 3.18 Tampilan Halaman Tambah Data Siswa

Antarmuka Halaman Tambah Data Siswa, seperti terlihat pada Gambar 3.18, dirancang untuk memfasilitasi proses *input* data siswa baru secara sistematis dan komprehensif. Halaman ini menyediakan formulir yang terstruktur dalam beberapa seksi, mencakup kategori informasi mulai dari data pribadi hingga keterangan pendidikan siswa. Untuk mengakomodasi banyaknya field, area formulir dilengkapi dengan fitur scrollable. Setiap field terdiri dari label deskriptif dan komponen *input* yang sesuai, seperti Entry untuk isian teks, Combobox untuk pilihan terbatas dari daftar, dan area Text untuk *input* narasi yang lebih panjang. Setelah pengisian data, pengguna dapat memanfaatkan tombol "Simpan" untuk menyimpan informasi ke database atau tombol "Reset" untuk mengosongkan formulir. Integrasi dengan *back-end* terjadi saat tombol "Simpan" ditekan; data dari semua field akan dikumpulkan, divalidasi di sisi *front-end*, dan jika valid, dikirim untuk dieksekusi sebagai perintah SQL INSERT ke tabel siswa dalam database SQLite. Sistem memberikan umpan balik visual kepada pengguna mengenai status keberhasilan atau kegagalan proses penyimpanan data.

The screenshot shows a web interface titled "Daftar Rombongan Siswa". It features a search bar with the text "Cari Nama/NIPD/NISN" and a "Cari" button. Below the search bar is a table with multiple columns. The visible columns include "Nama", "NIPD", "NISN", "Jenis Kelamin", "Status Siswa", and "Tanggal Masuk". The table contains numerous rows of student data. On the left side, there is a sidebar menu with options like "Beranda", "Daftar Siswa", "Tambah Siswa", and "Hapus Siswa". At the bottom, there are buttons for "Tambah", "Cari", and "Filter Status".

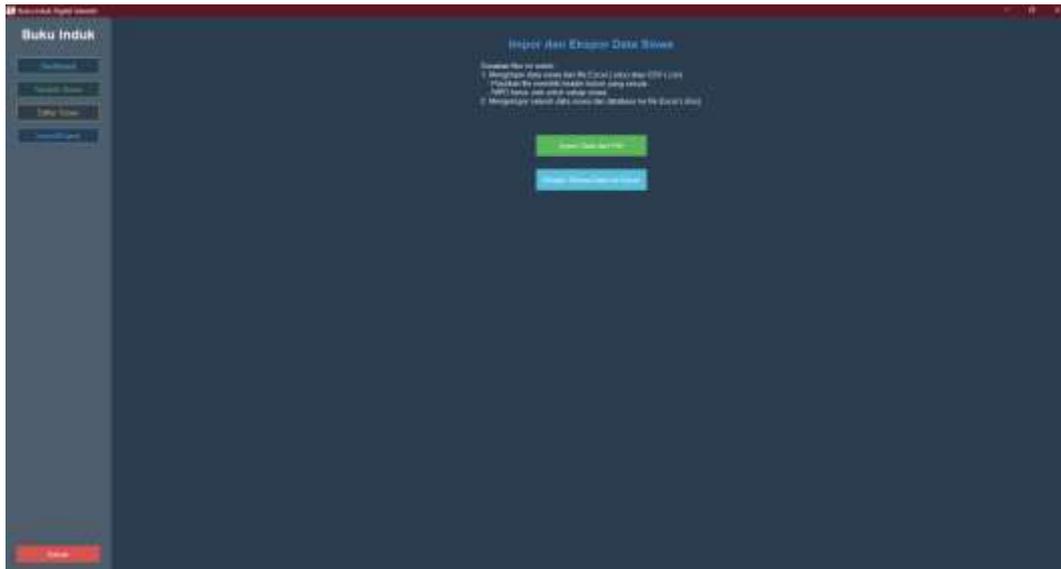
Gambar 3.19 Tampilan Halaman Daftar Siswa

Gambar 3.19 menunjukkan implementasi Halaman Daftar Siswa, yang berfungsi sebagai antarmuka utama untuk menampilkan dan mengelola daftar siswa yang tersimpan dalam database. Data siswa disajikan dalam format tabel menggunakan komponen `ttkbootstrap.Treeview`, menampilkan kolom-kolom kunci seperti "NAMA", "NIPD", dan "NISN" untuk identifikasi cepat. Halaman ini dilengkapi dengan fitur pencarian interaktif yang memungkinkan pengguna mencari siswa berdasarkan Nama, NIPD, atau NISN melalui input field dan tombol "Cari". Selain itu, terdapat opsi "Filter" berbentuk dropdown untuk menyaring daftar siswa berdasarkan kriteria tertentu, seperti status siswa. Integrasi dengan database terjadi secara dinamis; saat halaman dimuat, atau ketika fitur pencarian dan filter digunakan, sistem akan mengeksekusi query SQL `SELECT` untuk mengambil data siswa yang relevan. Hasil query tersebut kemudian diperbarui dan ditampilkan dalam `Treeview`. Pengguna dapat memilih satu siswa dari tabel untuk melakukan aksi lebih lanjut, seperti melihat detail atau menghapus data, yang akan memicu callback ke fungsi terkait di logika aplikasi.



Gambar 3.20 Tampilan Halaman Detail Siswa

Halaman Detail Siswa, yang mock-upnya terlihat pada Gambar 3.20, dirancang untuk menyajikan informasi lengkap mengenai satu siswa tertentu yang telah dipilih dari Halaman Daftar Siswa. Saat halaman ini dimuat, NIPD siswa yang dipilih digunakan sebagai kunci untuk melakukan query SELECT ke database, dan semua data terkait siswa tersebut kemudian ditampilkan dalam berbagai field input yang awalnya bersifat *read-only*. Pengguna memiliki opsi untuk mengubah data siswa dengan menekan tombol "Edit Data", yang akan mengubah status field menjadi dapat diedit dan mengaktifkan tombol "Simpan perubahan". Jika perubahan dilakukan dan disimpan, data yang telah dimodifikasi akan dikumpulkan, divalidasi, dan query SQL UPDATE dijalankan untuk memperbarui record siswa di database berdasarkan NIPD. Selain itu, halaman ini menyediakan fitur "Export ke Word", yang mengintegrasikan data siswa yang sedang ditampilkan dengan pustaka python-docx untuk menghasilkan dokumen Microsoft Word.



Gambar 3.21 Tampilan Halaman Import/Export

Gambar 3.21, menyediakan fungsionalitas untuk manajemen data siswa secara massal. Halaman ini dirancang dengan sederhana, menampilkan dua tombol aksi utama: "Import Data" dan "Export Data". Fitur "Import Data" memungkinkan pengguna memilih berkas eksternal (Excel atau CSV) yang berisi data siswa. Setelah berkas dipilih, logika back-end yang terintegrasi dengan pustaka pandas akan membaca berkas tersebut, melakukan normalisasi header dan pemetaan kolom terhadap skema database, serta memvalidasi setiap baris data sebelum melakukan operasi INSERT secara batch ke dalam tabel siswa di database SQLite. Sistem akan memberikan ringkasan hasil proses impor, termasuk jumlah data yang berhasil dan gagal diimpor. Untuk fitur "Export Data", sistem akan menjalankan query SELECT * untuk mengambil seluruh data siswa dari database, kemudian menggunakan pandas untuk menyusun data tersebut dan menyimpannya ke dalam berkas Excel yang lokasinya ditentukan oleh pengguna.

3.3. Kendala yang Dihadapi

Selama proses pelaksanaan kerja profesi dan pengembangan Aplikasi Buku Induk Sekolah Digital, praktikan menghadapi beberapa kendala teknis dan konseptual. Kendala-kendala ini memerlukan analisis dan upaya penyelesaian agar tidak menghambat progres pengembangan sistem. Berikut merupakan penjabaran kendala utama yang dihadapi praktikan selama proses pelaksanaan kerja profesi.

1. **Pengelolaan Operasi CRUD (Create, Read, Update, Delete)**
Implementasi fitur CRUD untuk data siswa pada setiap modul memerlukan logika yang cukup kompleks untuk memastikan integritas data dan interaksi yang benar dengan database SQLite. Praktikan menghadapi tantangan dalam merancang query SQL yang optimal dan menangani berbagai skenario pembaruan serta penghapusan data dengan aman, terutama terkait relasi atau dependensi data yang mungkin ada.
2. **Manajemen Integrasi dan Navigasi Antar Halaman**
Aplikasi ini terdiri dari beberapa halaman atau frame yang berbeda. Mengelola transisi antar frame ini, memastikan data dapat dilewatkan dengan benar dan menjaga konsistensi state aplikasi menjadi sebuah tantangan. Diperlukan mekanisme callback dan referensi antar objek kelas yang terstruktur agar navigasi berjalan lancar dan data tetap sinkron.
3. **Implementasi Fitur Dinamis pada Dashboard**
Menampilkan data statistik yang akurat dan grafik tren yang informatif pada Halaman Dashboard memerlukan beberapa langkah. Kendala yang dihadapi termasuk perancangan query SQL yang efisien untuk mengagregasi data dari database, pemrosesan data tersebut untuk format grafik, serta integrasi pustaka matplotlib dengan antarmuka Tkinter agar grafik dapat ditampilkan secara dinamis dan responsif.
4. **Fleksibilitas Impor Data dari Berkas Eksternal**
Kendala utama dalam implementasi fitur ini adalah potensi ketidaksamaan dalam penulisan header kolom pada berkas sumber (misalnya, perbedaan huruf besar/kecil, penggunaan spasi atau simbol) dan struktur header yang mungkin tidak sama persis dengan urutan atau nama kolom pada tabel database.
5. **Keterbatasan Pengetahuan dan Pengalaman Praktis**
Meskipun telah dibekali dengan dasar-dasar teoritis selama perkuliahan, beberapa aspek teknis spesifik dalam pengembangan aplikasi merupakan hal yang relatif baru bagi praktikan. Keterbatasan dalam pengalaman praktis pada teknologi ini serta pada penanganan kasus-kasus pengembangan yang kompleks memerlukan waktu adaptasi dan pembelajaran mandiri yang lebih intensif.

3.4. Cara Mengatasi Kendala

Untuk menyelesaikan berbagai kendala yang dihadapi selama pengembangan Aplikasi Buku Induk Digital Sekolah, praktikan melakukan langkah-langkah perbaikan dan adaptasi sebagai berikut:

1. Optimalisasi Pengelolaan Data dan Operasi CRUD

- Menerapkan struktur kelas Python yang modular untuk setiap halaman, sehingga mempermudah pengelolaan logika spesifik untuk operasi *Create, Read, Update, Delete* (CRUD) pada masing-masing modul.
- Merancang dan mengimplementasikan *query* SQL yang lebih spesifik dan efisien untuk setiap operasi database, termasuk penggunaan klausa *WHERE* yang tepat dan penanganan error integritas data (NIPD unik) pada database SQLite.
- Memastikan penggunaan blok *try-except-finally* dalam setiap interaksi database untuk menangani potensi error dan menjamin koneksi database ditutup dengan benar setelah digunakan.

2. Peningkatan Integrasi Antar Halaman dan Alur Kerja Pengguna

- Mengimplementasikan mekanisme *callback function* yang jelas untuk memfasilitasi komunikasi dan perpindahan data antar halaman aplikasi. Contohnya, NIPD siswa dari *DaftarSiswaFrame* diteruskan ke *DetailSiswaFrame* melalui *callback* yang diatur oleh *main.py*.
- Menyediakan referensi objek aplikasi utama (*app_ref*) ke setiap *frame* anak, memungkinkan akses terpusat ke metode bersama seperti *get_db_connection()*.

3. Optimalisasi Fitur Dashboard dan Visualisasi Data

- Merancang *query* SQL yang efisien untuk melakukan agregasi data siswa (misalnya, *COUNT(*)*) yang diperlukan untuk statistik pada Halaman Dashboard.
- Menggunakan pustaka *pandas* untuk mempermudah pra-pemrosesan dan agregasi data sebelum divisualisasikan.

- Mengintegrasikan pustaka matplotlib dengan antarmuka Tkinter (FigureCanvasTkAgg) untuk menampilkan grafik tren siswa secara dinamis dan informatif.
4. Peningkatan Fleksibilitas dan Ketangguhan Fitur Impor Data
- Mengembangkan algoritma normalisasi *header* di `import_export.py` (`_normalize_header_text`) untuk mengubah *header* dari file sumber (Excel/CSV) menjadi format standar (lowercase, underscore), sehingga mengurangi sensitivitas terhadap variasi penulisan.
 - Mengimplementasikan mekanisme pemetaan alias (*header aliasing*) melalui `header_aliases_map` yang memungkinkan sistem mengenali berbagai variasi nama kolom di file sumber dan memetakannya ke kolom database yang benar. Praktikan memanfaatkan bantuan AI sebagai alat untuk mendapatkan ide awal dan kerangka algoritma untuk solusi ini.
 - Melakukan validasi data per baris selama proses impor untuk mengidentifikasi dan melewati data yang tidak valid (misalnya NIPD kosong) atau NIPD duplikat, sambil tetap melanjutkan impor data yang valid.
5. Peningkatan Pengetahuan dan Keterampilan Teknis
- Melakukan studi mandiri secara intensif terhadap dokumentasi resmi Python, Tkinter, ttkbootstrap, pandas, matplotlib, dan SQLite untuk memahami fungsionalitas dan praktik terbaik.
 - Memanfaatkan sumber daya daring seperti tutorial dan forum komunitas untuk mencari solusi atas permasalahan teknis spesifik.
 - Melakukan diskusi dan konsultasi rutin dengan dosen pembimbing untuk mendapatkan arahan, validasi pendekatan, dan solusi atas kendala yang lebih kompleks.
 - Menggunakan alat bantu AI secara bertanggung jawab untuk menyelesaikan kesalahan dalam pemrograman, mendapatkan contoh kode, dan membantu dalam proses eksplorasi pengetahuan.