

BAB 3

PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Pelaksanaan program magang di PT Summarecon Agung Tbk menempati posisi sebagai *Web Developer* (Full Stack) di bawah Divisi IT pada unit *Engineering*. Pembimbing lapangan merupakan *Head of Network Operation Infrastructure*, yang berperan sebagai *supervisor* magang dan memberikan tugas serta arahan selama proses magang berlangsung. Sebagai bagian dari koordinasi kerja, kegiatan rapat dilaksanakan sesuai jadwal yang telah ditentukan dan disesuaikan dengan kebutuhan mingguan. Rapat dilakukan secara tatap muka (*offline*) di kantor, dengan tujuan menyampaikan perkembangan tugas (*progress project*), menerima *feedback*, serta mendiskusikan arahan lanjutan terkait proyek yang sedang berjalan.

Pemantauan terhadap aktivitas kerja harian dilakukan melalui sistem evaluasi tugas harian (*daily task*) secara berkala. Evaluasi tersebut tercatat melalui situs merdeka.umn.ac.id sebagai bagian dari pelaporan kegiatan kepada *supervisor* magang dan pihak akademik. Dengan sistem tersebut, seluruh aktivitas kerja terdokumentasi secara sistematis dan dapat dievaluasi secara transparan.

3.2 Tugas yang Dilakukan

Tugas yang dilakukan selama pelaksanaan magang di PT Summarecon Agung Tbk adalah merancang dan membangun sistem manajemen informasi infrastruktur jaringan berbasis *website dashboard* admin dan *user*. *Website* ini mencakup pengelolaan data perjanjian kerja sama dengan berbagai penyedia layanan ISP (seperti *Telkom*, *Balifiber*, *XL Home*), data *homepass* data *design drawing*, serta data performance mengenai keuangan (*revenue*, *cost of sales*, *EBIT*, *NPAT*, *NPBI*, *rencana* dan *realisasi capex*). Tujuan *website* ini yaitu mempermudah pengelompokan informasi jaringan dari banyaknya data *project infrastructure* yang dikembangkan, serta membantu pihak lain seperti *finance* mempermudah informasi data yang dimiliki oleh jejaring ultra prima. Hal-hal yang dilakukan pada pelaksanaan kerja magang sebagai berikut:

1. Mendesain tampilan UI/UX halaman *dashboard* admin dan *user*. Tampilan

ini disesuaikan dengan keinginan yang diminta oleh pihak *supervisor* dan atasan.

2. Merancang dan membuat *schema structure database* dan *flowchart* dari alur pemakaian *website*. *Database* yang digunakan untuk menyimpan seluruh data, nantinya akan dimasukkan atau tersimpan menggunakan *MySQL Server*.
3. Mengembangkan pembuatan operasi CRUD (*Create, Read, Update, Delete*) pada *dashboard* admin, serta melakukan *debugging* selama proses uji memastikan program berjalan sesuai dengan fungsinya.
4. Merekap beberapa data dan dokumen periode waktu perjanjian kerja sama dalam *project infrastructure* terkait.
5. Mengikuti rapat untuk mendapatkan *requirement* yang diperlukan dan mempresentasikan tampilan pembuatan *website* yang dikembangkan.

3.3 Uraian Pelaksanaan Magang

3.3.1 Proses Pelaksanaan Kerja Magang

Rincian kegiatan magang yang dilaksanakan di PT Summarecon Agung Tbk disajikan pada Tabel 3.1.

Tabel 3.1. Aktivitas Mingguan Selama Magang

Minggu ke-	Aktifitas yang dilakukan
1	Mengikuti <i>meeting</i> dan <i>briefing</i> awal terkait kebutuhan sistem <i>website</i> yang akan dikembangkan.
2	Mendesain tampilan awal halaman admin dan pengguna, serta menyusun struktur awal <i>database</i> .
3	Mengimplementasikan <i>logic</i> untuk frontend dan sistem backend CRUD (<i>Create, Read, Update, Delete</i>) sebagai bagian dari pengelolaan informasi infrastruktur jaringan.
4	Membuat visualisasi tampilan dan menambahkan komponen <i>DataTables</i> untuk halaman unit, kawasan, dan proyek.
5	Memperbaiki kesalahan sistem CRUD dan menyempurnakan alur kerja pada halaman <i>dashboard</i> admin.

Lanjut pada halaman berikutnya

Lanjutan Tabel 3.1 Aktivitas Mingguan Selama Magang

Minggu Ke-	Aktifitas yang dilakukan
6	Melanjutkan pengembangan halaman <i>dashboard</i> admin dengan penambahan elemen visual lanjutan.
7	Menyusun ulang struktur <i>database</i> , menambahkan tampilan <i>dashboard</i> pengguna, serta mengembangkan logika pengiriman pengingat PKS melalui email.
8	Mempelajari penggunaan PHP <i>Mailer</i> untuk pengiriman pesan otomatis melalui sistem.
9	Mengembangkan halaman detail proyek yang memuat data <i>homepass</i> , ISP, PKS, dan <i>design drawing</i> .
10	Mengikuti rapat evaluasi terkait perkembangan proyek dan penambahan fitur yang diminta.
11	Melakukan revisi fitur dan penambahan data yang diminta, termasuk pengelolaan data performance dashboard seperti revenue, cost of sales, EBIT, NPAT, NPBI, serta rencana dan realisasi <i>capex</i> .
12	Merapikan tampilan halaman <i>dashboard</i> untuk admin dan pengguna, serta memperbaiki alur <i>flowchart</i> .
13	Mengembangkan fitur pengingat melalui email, fungsi reset password, dan sistem edit akun pengguna.
14	Menambahkan visualisasi data dalam bentuk grafik <i>chart</i> pada <i>dashboard</i> admin dan pengguna.
15	Menyusun buku panduan (<i>guide book</i>) dan menyiapkan materi presentasi dalam format PowerPoint.
16	Melakukan pengecekan akhir dan perbaikan terhadap kesalahan pada sistem dan tampilan <i>website</i> .
17	Melaksanakan presentasi dan demonstrasi penggunaan sistem kepada pihak perusahaan.

3.3.2 Perancangan Sistem

Pada tahap awal dalam proses perancangan sistem, dibuatlah konsep visual berupa desain awal *UI/UX* untuk mempresentasikan antarmuka serta pengalaman pengguna yang diharapkan. Setelah itu, struktur data dan relasi antar entitas dalam sistem dirancang melalui pembuatan *Entity-Relationship Diagram* (ERD) sebagai

dasar pemodelan *database*.

Selanjutnya, untuk menggambarkan peran dan interaksi antara pengguna dengan sistem, digunakan *Use Case Diagram* yang memetakan fitur-fitur inti yang tersedia untuk masing-masing aktor, seperti admin dan *user*. Lalu, alur proses dari fitur-fitur tersebut digambarkan dalam bentuk *flowchart* untuk memberikan visualisasi yang sistematis terhadap logika dan jalannya proses dalam sistem.

A Requirements

Sistem ini dirancang untuk mengelola informasi proyek infrastruktur jaringan secara efisien dan terstruktur. Sistem menyediakan akses dan fitur berbeda bagi pengguna berdasarkan peran atau hak akses masing-masing.

- **Admin**

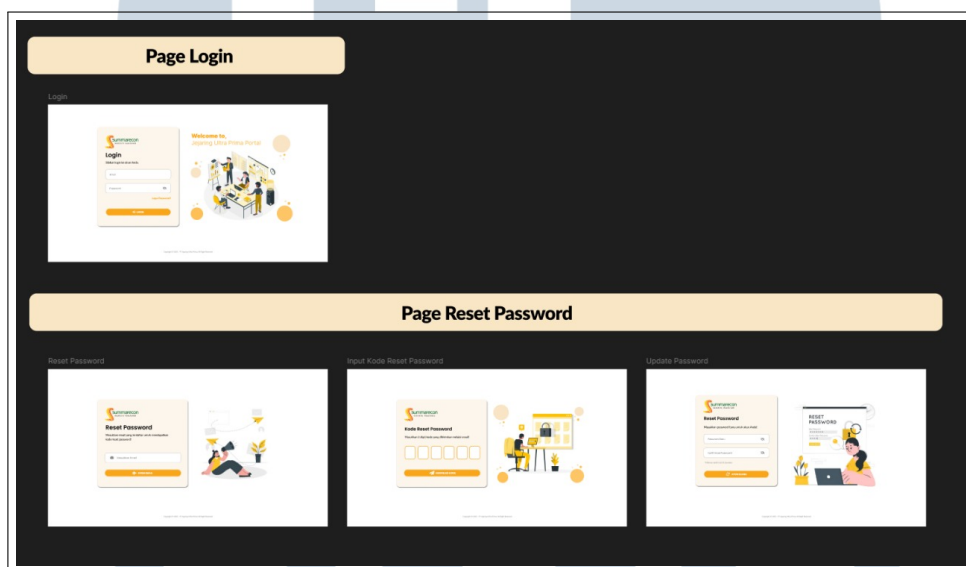
1. Dapat melakukan operasi CRUD (*Create, Read, Update, Delete*) terhadap *Project Data*, yang mencakup: data unit, kawasan, proyek infrastruktur, jumlah *homepass*, *Internet Service Provider* (ISP), data Perjanjian Kerja Sama (PKS), serta data *design drawing*.
2. Dapat mengelola *Performance Data* melalui operasi CRUD, termasuk data *revenue*, *cost of sales*, *EBIT*, *NPAT*, *NPBI*, serta data rencana dan realisasi *capex* per tahun.
3. Dapat mengelola akun pengguna (*manage users*) dengan melakukan operasi CRUD terhadap data user.
4. Akan menerima notifikasi email otomatis sebagai reminder ketika data PKS mendekati tanggal berakhir.

- **User**

1. Hanya memiliki akses untuk melihat (*read-only*) seluruh *project data* dan *performance data* tanpa akses untuk mengubah, menambah, atau menghapus data.

B Perancangan Desain Website

Pada tahap perancangan awal, berdasarkan permintaan dari pihak perusahaan, dikembangkan sebuah *website dashboard* yang berfungsi sebagai sistem manajemen informasi infrastruktur jaringan dari berbagai *project*. *Website* ini dirancang tidak hanya untuk administrator atau pengelola sistem, tetapi juga untuk mendukung kebutuhan berbagai pihak terkait, seperti tim *finance*, yang dalam sistem ini berperan sebagai pengguna dengan hak akses terbatas, yaitu hanya dapat melihat data yang tersedia tanpa dapat mengubah atau menghapusnya.

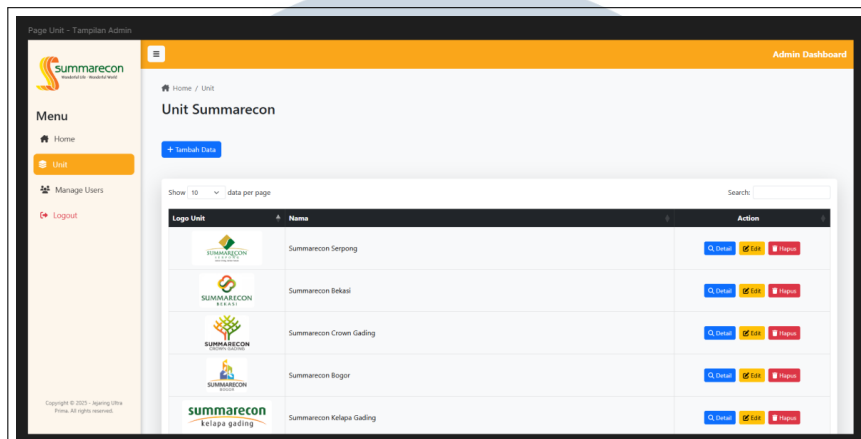


Gambar 3.1. Desain Tampilan Login dan Reset Password

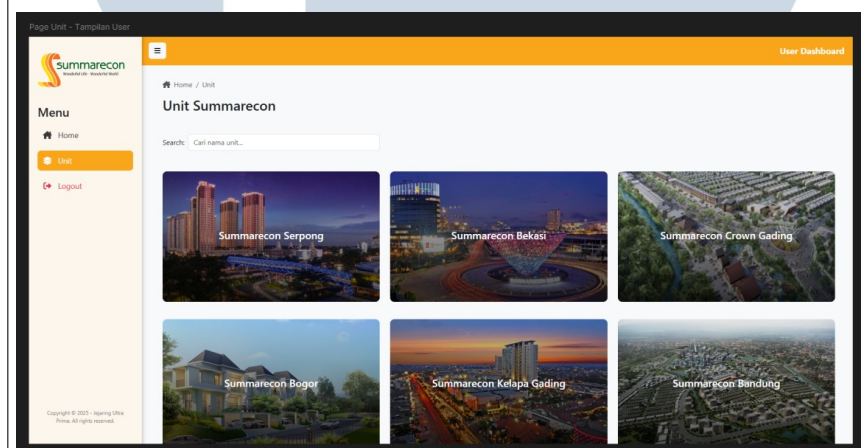
Sebelum memasuki tahap implementasi dan pengembangan kode secara menyeluruh, langkah awal yang dilakukan adalah merancang antarmuka pengguna (*UI*) dan pengalaman pengguna (*UX*) untuk seluruh halaman yang akan dikembangkan. Perancangan tampilan ini bertujuan untuk menciptakan tampilan yang tidak hanya menarik secara visual, tetapi juga memberikan kemudahan dan kenyamanan dalam penggunaan, sehingga pengguna dapat menjalankan fungsionalitas sistem secara efektif dan efisien.

Gambar 3.1 menunjukkan hasil perancangan awal tampilan halaman *login* dan halaman *reset password*. Tampilan ini mencakup beberapa tahapan penting dalam proses autentikasi pengguna, yaitu formulir untuk memasukkan *email* yang telah terdaftar dalam sistem, kolom untuk memasukkan kode *reset password*, serta halaman untuk mengatur ulang *password* pengguna. Seluruh elemen dalam tampilan ini dirancang secara sistematis dengan prinsip desain yaitu *8 Golden Rules*

agar mendukung proses masuk ke dalam sistem dengan cara yang aman, cepat, dan mudah dipahami oleh pengguna dari berbagai latar belakang.



(a) Tampilan Admin

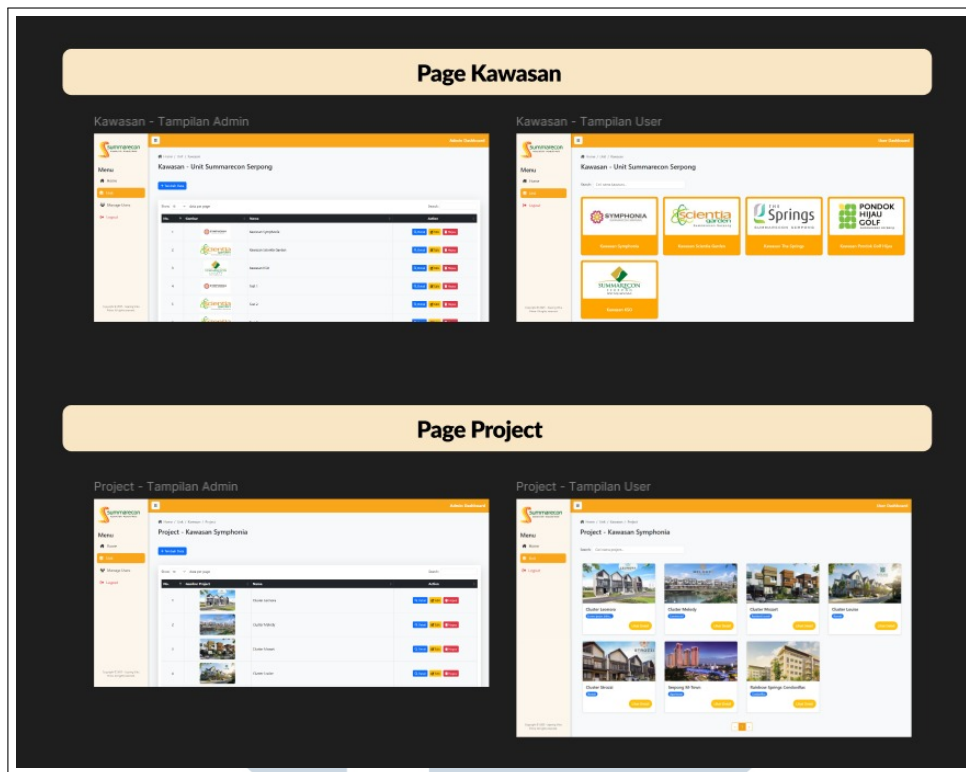


(b) Tampilan User

Gambar 3.2. Tampilan Halaman Data Unit (Admin) dan (User)

Tampilan halaman data unit untuk admin, sebagaimana ditunjukkan pada Gambar 3.2a, akan menampilkan tabel yang memuat seluruh data unit yang tersedia. Pada halaman ini, admin diberikan akses penuh terhadap sistem untuk melakukan operasi CRUD (*Create*, *Read*, *Update*, dan *Delete*) untuk mengelola data secara menyeluruh.

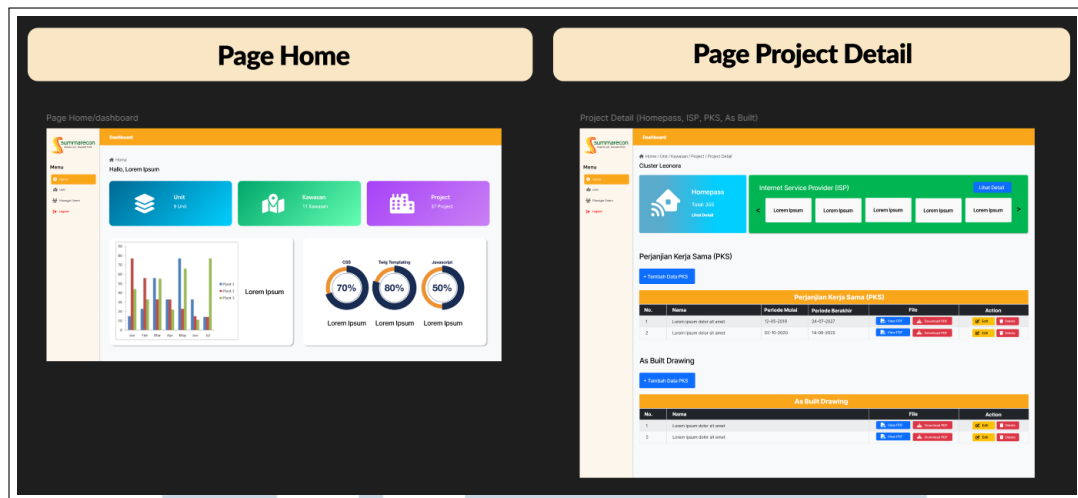
Sementara itu, Gambar 3.2b memperlihatkan tampilan halaman untuk pengguna (*user*), di mana hak akses dibatasi hanya pada fitur *Read* saja. Dalam tampilan ini, data unit disajikan dalam bentuk *card* bergambar yang informatif, sehingga memudahkan pengguna untuk melihat informasi unit secara visual tanpa dapat melakukan perubahan terhadap data.



Gambar 3.3. Perancangan Tampilan Website

Gambar 3.3 menampilkan rancangan antarmuka untuk halaman Kawasan dan Project. Dalam perancangannya, tampilan untuk *user* dan admin dibuat berbeda sesuai dengan hak akses masing-masing. Admin memiliki akses penuh terhadap sistem, sehingga dapat melakukan seluruh operasi CRUD pada halaman tersebut. Oleh karena itu, tampilan untuk admin dirancang menggunakan format *datatable* yang dilengkapi dengan fitur pengelolaan data secara lengkap.

Sebaliknya, pengguna (*user*) hanya diberikan hak akses untuk melihat informasi (*read only*). Untuk mempermudah pemahaman dan memberikan pengalaman visual yang lebih baik, tampilan halaman Kawasan dan Project bagi pengguna disajikan dalam bentuk card bergambar yang informatif dan mudah dipahami.



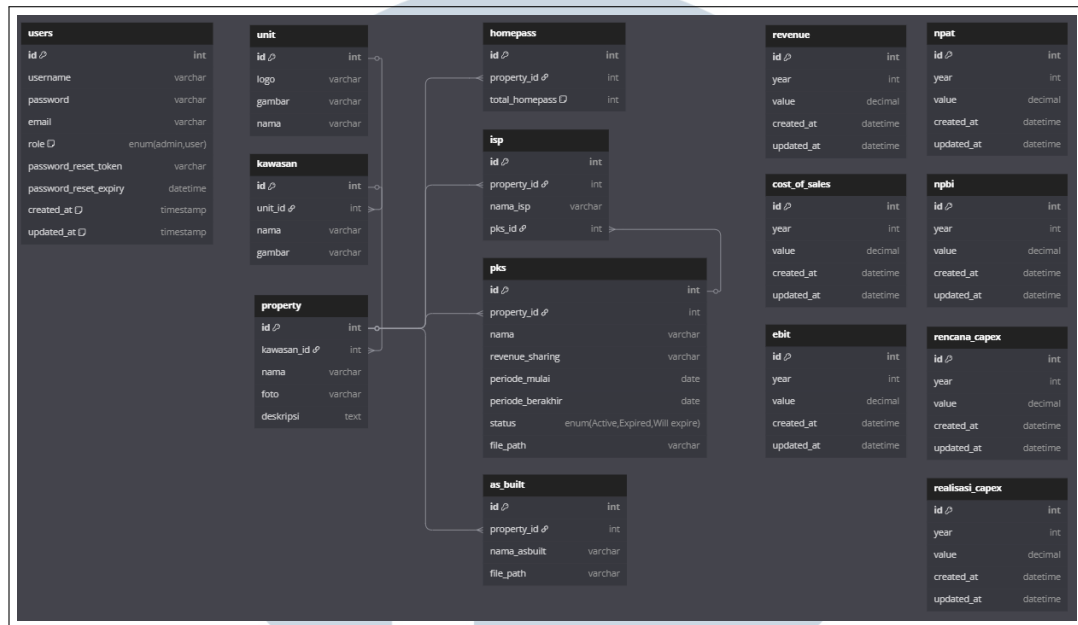
Gambar 3.4. Perancangan Tampilan Website

Gambar 3.4 menampilkan rancangan antarmuka untuk halaman utama dan halaman *Project Detail*. Pada bagian kiri halaman utama ditampilkan *dashboard project* yang memuat *summary* jumlah total unit, kawasan, dan *project* secara keseluruhan. Menampilkan visualisasi berupa grafik batang (*bar chart*) dan diagram donat (*donut chart*) untuk memudahkan ringkasan data.

Sementara itu, bagian kanan merupakan halaman *Project Detail* yang berfungsi untuk menampilkan informasi lebih lanjut terkait *project* tertentu. Beberapa data utama yang ditampilkan pada halaman ini meliputi: **homepass**, **Internet Service Provider (ISP)**, **Perjanjian Kerja Sama (PKS)**, dan **design drawing**. Data ISP ditampilkan dalam bentuk *card carousel* yang interaktif untuk memudahkan pengguna dalam melihat data. Sedangkan data PKS dan *design drawing* ditampilkan dalam format *datatable* yang terstruktur, lengkap dengan kolom *file* yang menyediakan fitur untuk melihat (*view*) dan mengunduh (*download*) dokumen dalam format *.pdf.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C Entity Relationship Diagram (ERD)



Gambar 3.5. Entity Relationship Diagram (ERD)

Gambar 3.5 di atas merupakan *Entity-Relationship Diagram (ERD)* dari sistem manajemen informasi infrastruktur jaringan. Terdapat enam entitas utama dalam sistem ini, yaitu users, unit, kawasan, property, pks, dan isp, serta beberapa entitas pendukung lainnya seperti homepass, as_built, dan tabel-tabel keuangan (revenue, cost_of_sales, ebit, npat, npbi, rencana_capex, dan realisasi_capex).

- **Users**

Menyimpan data pengguna sistem, termasuk informasi *login* yaitu email, password dan peran (admin atau user). Atribut lainnya seperti username, password_reset_token dan password_reset_expiry digunakan untuk proses pemulihan akun, penjelasan lebih lanjut pada tabel 3.2.

Tabel 3.2. Struktur Tabel Users

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	username	VARCHAR	255	Username pengguna
3	password	VARCHAR	255	Password yang dimiliki user
4	email	VARCHAR	255	Unique
5	role	ENUM	-	'admin' atau 'user' (default 'user')
6	password_reset_token	VARCHAR	255	Token reset password
7	password_reset_expiry	DATETIME	-	Waktu kedaluwarsa token
8	created_at	TIMESTAMP	-	Default CURRENT_TIMESTAMP
9	updated_at	TIMESTAMP	-	Auto update saat perubahan data

- **Unit**

Merupakan entitas utama yang merepresentasikan satuan induk atau entitas pemilik kawasan. Setiap unit dapat memiliki banyak kawasan. Relasi antara unit dan kawasan adalah *one-to-many*, karena satu unit dapat memiliki banyak kawasan, namun satu kawasan hanya dimiliki oleh satu unit. Penjelasan lebih lanjut pada tabel 3.3.

Tabel 3.3. Struktur Tabel Unit

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	logo	VARCHAR	255	Gambar logo unit (path file)
3	gambar	VARCHAR	255	Gambar unit lainnya (path file)
4	nama	VARCHAR	100	Nama unit

- **Kawasan**

Berisi informasi nama dan gambar dari kawasan yang berada di bawah satu unit tertentu. Setiap kawasan dapat memiliki banyak *project* atau *property*. Relasi antara kawasan dan *property* juga merupakan *one-to-many*, yaitu satu kawasan dapat memiliki banyak properti, namun setiap properti hanya terkait ke satu kawasan. Penjelasan lebih lanjut pada tabel 3.4.

Tabel 3.4. Struktur Tabel Kawasan

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	unit_id	INT	-	Foreign Key ke unit (id), ON DELETE RESTRICT
3	nama	VARCHAR	255	Nama kawasan, memiliki index
4	gambar	VARCHAR	255	Gambar kawasan (path file)

- **Property**

Menyimpan data detail properti atau *project*, seperti nama, foto, dan deskripsi. *Project* berada dalam satu kawasan tertentu dan menjadi pusat relasi terhadap beberapa entitas lain seperti *homepass*, *isp*, *pks*, dan *as_built*. Penjelasan lebih lanjut pada tabel 3.5.

Tabel 3.5. Struktur Tabel Property

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	kawasan_id	INT	-	Foreign Key ke kawasan (id), ON DELETE RESTRICT
3	nama	VARCHAR	255	Nama properti, memiliki index
4	foto	VARCHAR	255	Path file foto project
5	deskripsi	TEXT	-	Deskripsi project

- **Homepass**

Mencatat jumlah *homepass* (unit sambungan layanan internet) untuk tiap *project*. Relasinya antara *property* bersifat *one-to-one*, karena satu properti hanya memiliki satu nilai *homepass*, dan setiap *homepass* hanya terkait ke satu properti. Penjelasan lebih lanjut pada tabel 3.6.

Tabel 3.6. Struktur Tabel Homepass

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	property_id	INT	-	Foreign Key ke property(id), ON DELETE CASCADE
3	total_homepass	INT	-	Jumlah homepass, default 0

- **ISP (Internet Service Provider)**

Menyimpan informasi tentang penyedia layanan internet yang bekerja sama dalam suatu *project*. Terhubung ke entitas *pks* untuk menunjukkan perjanjian kerja sama yang terkait. Relasi antara *pks* dan *isp* bersifat *one-to-one*, karena satu perjanjian kerja sama hanya digunakan oleh satu ISP, dan setiap ISP memiliki satu PKS terkait. Penjelasan lebih lanjut pada tabel 3.7.

Tabel 3.7. Struktur Tabel ISP

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	property_id	INT	-	Foreign Key ke property(id), ON DELETE CASCADE
3	nama_isp	VARCHAR	255	Nama ISP
4	pks_id	INT	-	Foreign Key ke pks(id), ON DELETE SET NULL

- **PKS (Perjanjian Kerja Sama)**

Berisi data Perjanjian Kerja Sama antara pengelola *project* dengan pihak ISP. Data yang disimpan termasuk nama perjanjian, periode waktu, status (Active, Expired, atau Will expire), serta file PKS berbentuk *.pdf. Penjelasan lebih lanjut terkait atribut kolom terdapat pada tabel 3.8.

Tabel 3.8. Struktur Tabel PKS

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	property_id	INT	-	Foreign Key ke property(id), ON DELETE CASCADE
3	nama	VARCHAR	255	Nama PKS
4	revenue_sharing	VARCHAR	255	Persentase bagi hasil
5	periode_mulai	DATE	-	Tanggal mulai periode
6	periode_berakhir	DATE	-	Tanggal akhir periode
7	status	ENUM	-	'Active', 'Expired', 'Will expire'
8	file_path	VARCHAR	255	Path file PDF perjanjian

- **As_Built**

Menyimpan dokumen hasil akhir pembangunan (*as-built drawing*) atau *design drawing* yang diunggah dan dikaitkan dengan masing-masing *project*. Penjelasan lebih lanjut pada tabel 3.9.

Tabel 3.9. Struktur Tabel as_built

No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	property_id	INT	-	Foreign Key ke property(id), ON DELETE CASCADE
3	nama_asbuilt	VARCHAR	255	Nama design drawing
4	file_path	VARCHAR	255	Path file PDF As Built

- **Tabel-tabel Keuangan**

Terdiri dari tabel *revenue*, *cost_of_sales*, *ebit*, *npat*, *npbi*, *rencana_capex*, dan *realisasi_capex*. Semua tabel ini memiliki struktur yang sama, menyimpan nilai keuangan pada kolom 'value' dan data tahun pada kolom 'year', serta atribut waktu pencatatan. Data keuangan ini mendukung analisis performa secara berkala. Penjelasan lebih lanjut terkait atribut-atribut *database* terdapat pada tabel 3.10.

Tabel 3.10. Struktur Tabel Keuangan

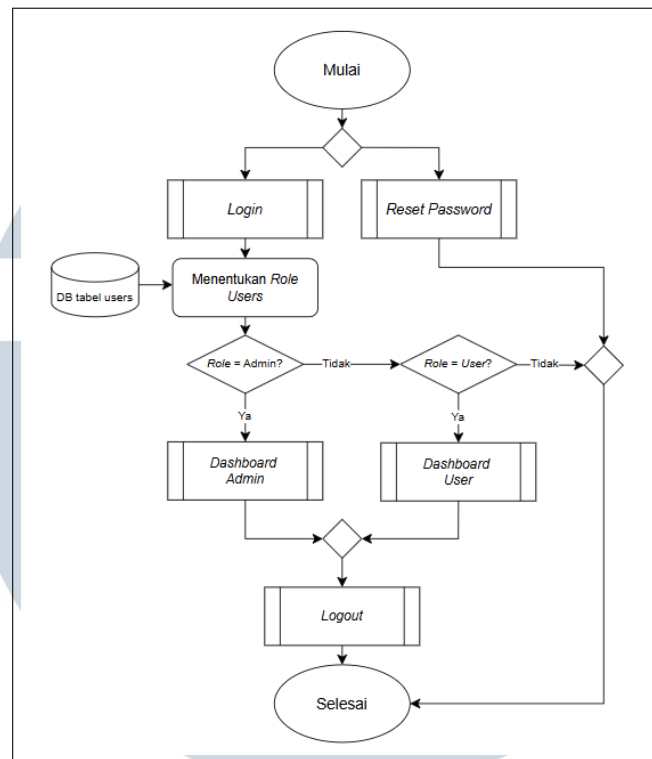
No	Nama Kolom	Tipe Data	Panjang	Keterangan
1	id	INT	-	Primary Key, Auto Increment
2	year	INT	-	Nilai tahun
3	value	DECIMAL	15,2	Nilai (desimal)
4	created_at	DATETIME	-	Tanggal data dibuat (default CURRENT_TIMESTAMP)
5	updated_at	DATETIME	-	Tanggal data terakhir diubah (auto update)

Relasi antar entitas disusun secara hierarki, dimulai dari unit ke kawasan, lalu ke property. Selanjutnya, property menjadi titik pusat bagi entitas lain seperti *homepass*, *isp*, *as_built* atau *design drawing*, dan *pks*. Dengan struktur ini, sistem dapat memantau seluruh data aset *project*, *ISP*, perjanjian kerja sama, beberapa dokumen, dan performa keuangan secara menyeluruh dan terintegrasi.

D Flowchart Diagram

Flowchart digunakan untuk menggambarkan alur proses yang terjadi pada *Website Dashboard* Sistem Manajemen Informasi Infrastruktur Jaringan. Setiap proses dari fitur-fitur utama divisualisasikan secara sistematis dalam bentuk *flowchart* untuk memberikan pemahaman yang jelas mengenai logika dan urutan tahapan yang berlangsung di dalam sistem.

- **Flowchart Main**

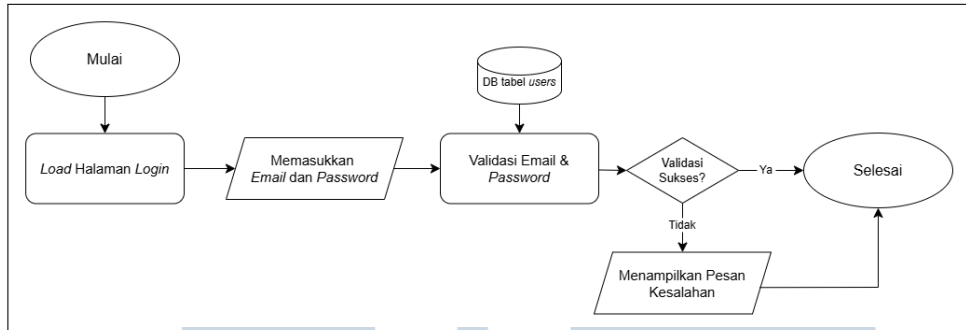


Gambar 3.6. Flowchart Website Sistem Manajemen Informasi Infrastruktur Jaringan

Gambar 3.6 menunjukkan *main flowchart* atau alur utama dari *Website Dashboard* Sistem Manajemen Informasi Infrastruktur Jaringan secara keseluruhan. Alur sistem dimulai ketika pengguna mengakses *website*, di mana secara otomatis akan ditampilkan halaman *login*. Pengguna yang belum memiliki akun tidak dapat melakukan *login*, karena proses pembuatan akun hanya dapat dilakukan oleh admin.

Setelah berhasil *login*, sistem akan memeriksa *role* (peran) pengguna berdasarkan data yang tersimpan di dalam *database* 'users' untuk menentukan hak aksesnya. Jika pengguna memiliki peran sebagai admin, maka akan diarahkan ke *dashboard* admin. Sebaliknya, jika berperan sebagai *user* biasa, maka akan diarahkan ke *dashboard* user. Selain itu, sistem juga menyediakan fitur *reset password* bagi pengguna yang lupa kata sandi, sehingga mereka dapat melakukan pengaturan ulang kata sandi secara mandiri.

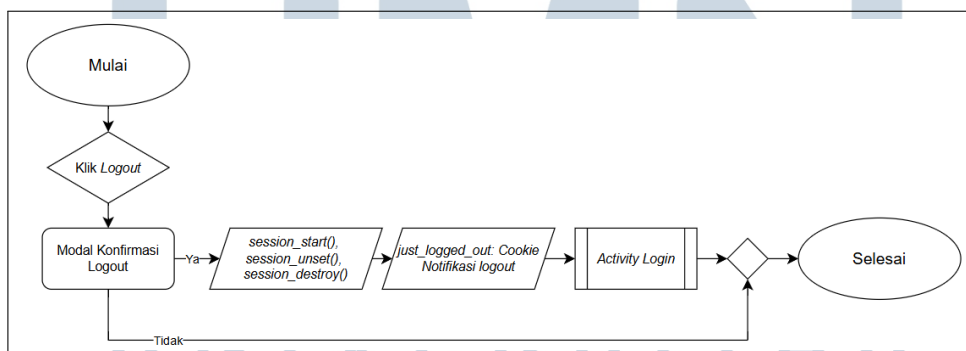
• Flowchart Login



Gambar 3.7. Flowchart Login

Gambar 3.7 menggambarkan alur proses pada halaman *login*. Pada tahap awal, pengguna akan ditampilkan *form login* untuk menginput *email* dan *password*. Setelah kedua data tersebut diisi, sistem akan melakukan proses validasi dengan membandingkan input yang dimasukkan terhadap data yang tersimpan di dalam *database 'users'*, khususnya pada kolom *email* dan *password*. Apabila validasi tidak berhasil, sistem akan menampilkan pesan kesalahan kepada pengguna sebagai tanggapan dari ketidaksesuaian data yang dimasukkan.

• Flowchart Logout



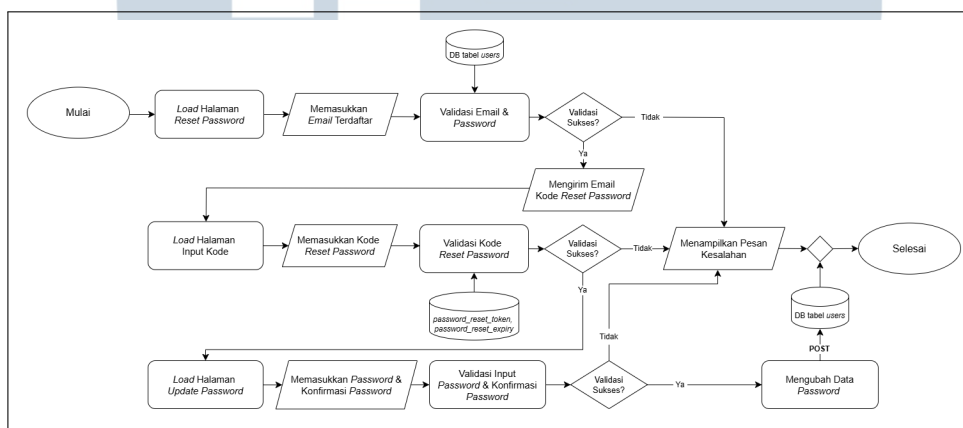
Gambar 3.8. Flowchart Logout

Gambar 3.8 menunjukkan alur proses pada halaman *logout*. Proses diawali jika dengan menampilkan modal konfirmasi *logout* setelah mengonfirmasi akan melakukan *logout*, yaitu memulai sesi menggunakan *session_start()*, kemudian semua variabel sesi yang tersimpan dihapus menggunakan *session_unset()*. Selanjutnya, sesi dihancurkan dengan

`session_destroy()` untuk memastikan semua data pengguna terhapus dari sisi server.

Setelah sesi dihapus, sistem akan membuat cookie sementara bernama `just_logged_out` yang berlaku selama 5 detik. cookie ini digunakan sebagai indikator bahwa pengguna baru saja melakukan *logout* dengan menampilkan notifikasi berhasil *logout*. Terakhir, pengguna akan diarahkan kembali ke halaman aktifitas login, dengan menampilkan **alert logout berhasil**.

- **Flowchart Reset Password**



Gambar 3.9. Flowchart Reset Password

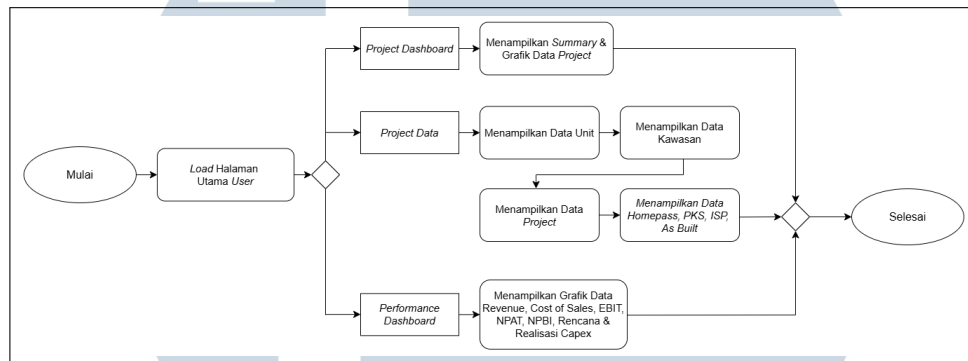
Gambar 3.9 menunjukkan *flowchart* dari proses *reset password*. Pada halaman pertama, sistem akan menampilkan *form* untuk *reset password*, di mana pengguna diminta untuk memasukkan alamat *email* yang terdaftar. Setelah itu, sistem akan melakukan validasi untuk memastikan apakah *email* tersebut terdaftar dalam tabel 'users'. Jika validasi berhasil, sistem akan mengirimkan kode *reset password* ke alamat *email* tersebut.

Selanjutnya, pengguna akan diarahkan ke halaman untuk memasukkan kode *reset password* yang telah dikirim. Sistem kemudian akan memverifikasi kode tersebut dengan mencocokkannya terhadap data di tabel `password_reset_token` dan memastikan bahwa kode tersebut masih berlaku berdasarkan kolom `password_reset_expiry`. Jika kode valid dan belum *expired*, pengguna akan diarahkan ke halaman *update password*.

Pada halaman ini, pengguna diminta untuk memasukkan *password* baru beserta konfirmasi *password*. Sistem akan memvalidasi apakah kedua

password tersebut cocok. Jika validasi berhasil, maka *password* pengguna akan diperbarui di dalam *database* tabel '*users*'. Namun, jika *password* dan konfirmasi *password* tidak cocok, sistem akan menampilkan pesan kesalahan.

- **Flowchart Dashboard User**



Gambar 3.10. Flowchart Sistem User

Gambar 3.10 menampilkan *flowchart* dari **dashboard user**. Setelah berhasil *login* dengan role sebagai *user* biasa, pengguna akan diarahkan ke halaman utama user yang menampilkan tiga menu utama, yaitu: **Project Dashboard**, **Project Data**, dan **Performance Dashboard**.

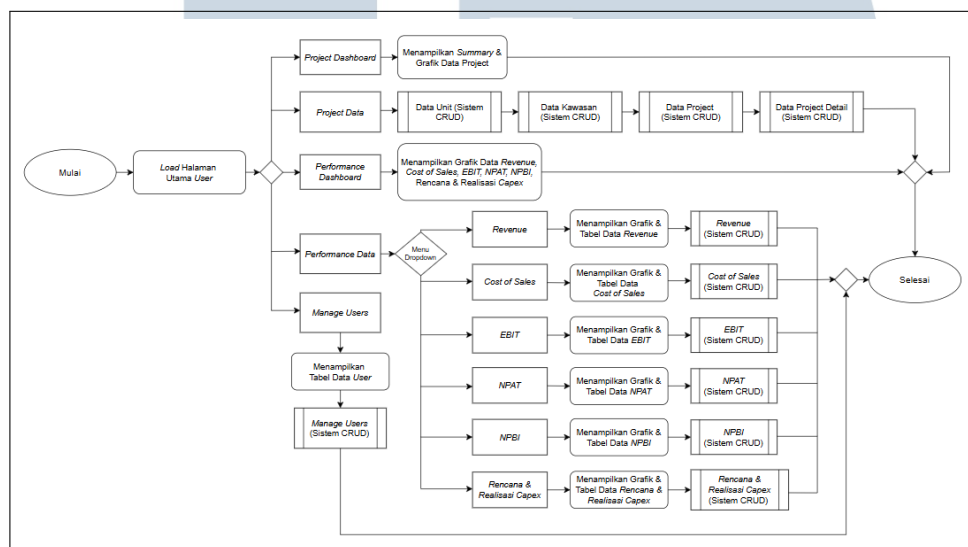
Pada menu **Project Dashboard**, sistem menampilkan ringkasan (*summary*) dan visualisasi grafik dari data *project* yang tersedia. Tujuannya adalah untuk memberikan gambaran umum secara cepat mengenai status dan perkembangan *project* kepada pengguna.

Selanjutnya, pada menu **Project Data**, halaman awal akan menampilkan daftar data unit. Ketika pengguna memilih salah satu unit, sistem akan menampilkan halaman Kawasan yang berisi daftar kawasan yang terhubung dengan unit tersebut, berdasarkan *id_unit*. Kemudian, saat pengguna memilih salah satu kawasan, sistem akan menampilkan daftar *Project Infrastructure* yang terkait berdasarkan *id_kawasan*. Jika pengguna memilih salah satu data *Project Infrastructure*, maka akan diarahkan ke halaman *Project Detail* dengan mengambil *id_property*.

Di halaman **Project Detail**, pengguna dapat melihat informasi secara lengkap, seperti jumlah *homepass*, data ISP, dokumen Perjanjian Kerja Sama (PKS), serta data *design drawing* dari proyek yang dipilih.

Terakhir, menu **Performance Dashboard** menampilkan visualisasi grafik dari berbagai data *performance*, seperti **revenue**, **cost of sales**, **EBIT**, **NPAT**, **NPBI**, serta **rencana dan realisasi capex**. Visualisasi ini digunakan untuk membandingkan performa *project* dari tahun ke tahun sebagai bagian dari proses evaluasi dan pemantauan kinerja.

- **Flowchart Dashboard Admin**



Gambar 3.11. Flowchart Sistem Admin

Gambar 3.11 menampilkan *flowchart* dari dashboard **admin** pada sistem. Setelah berhasil *login* dengan *role* sebagai admin, akan diarahkan ke halaman utama yang menyediakan beberapa menu utama, yaitu: **Project Dashboard**, **Project Data**, **Performance Dashboard**, **Performance Data**, dan **Manage Users**.

- **Project Dashboard**

Menu ini menampilkan ringkasan (*summary*) dan visualisasi grafik dari data *project* yang ada, sehingga admin dapat melihat gambaran umum dari status dan perkembangan *project* secara cepat.

- **Project Data**

Saat mengakses menu ini, sistem pertama kali menampilkan halaman **Unit** dalam bentuk tabel data. Admin dapat memilih salah satu unit untuk melihat informasi lebih lanjut:

- * Memilih unit akan mengarahkan ke halaman **Kawasan**, berdasarkan `id_unit`. Halaman ini menampilkan daftar kawasan yang terhubung dengan unit tersebut.
- * Memilih salah satu kawasan akan menampilkan daftar **Project Infrastructure** yang terkait, berdasarkan `id_kawasan`.
- * Memilih salah satu data *Project Infrastructure* akan membawa admin ke halaman **Project Detail**, berdasarkan `id_property`.
- * Di halaman **Project Detail**, ditampilkan berbagai informasi penting seperti jumlah *homepass*, data ISP, file dokumen *Perjanjian Kerja Sama (PKS)*, dan dokumen *design drawing*.

Penjelasan lebih lanjut mengenai masing-masing sistem CRUD pada data (Unit, Kawasan, *Project Infrastructure*, dan data-data seperti *homepass*, ISP, PKS, as built drawing) akan dijabarkan dalam bagian **flowchart sub-proses** tersendiri.

– Performance Dashboard

Menyediakan visualisasi grafik data performa yang meliputi:

- * *Revenue*
- * *Cost of Sales*
- * *EBIT (Earnings Before Interest and Taxes)*
- * *NPAT (Net Profit After Tax)*
- * *NPBI (Net Profit Before Interest)*
- * Rencana dan Realisasi *Capex*

Visualisasi ini digunakan untuk membandingkan performa dari tahun ke tahun sebagai bagian dari proses evaluasi.

– Performance Data

Menu ini memungkinkan admin untuk mengakses data performa dalam bentuk menu turunan atau *dropdown*, seperti:

- * *Revenue*
- * *Cost of Sales*
- * *EBIT (Earnings Before Interest and Taxes)*
- * *NPAT (Net Profit After Tax)*
- * *NPBI (Net Profit Before Interest)*
- * Rencana dan Realisasi *Capex*

Penjelasan lebih lanjut mengenai masing-masing submenu akan dijabarkan dalam bagian sub-proses tersendiri.

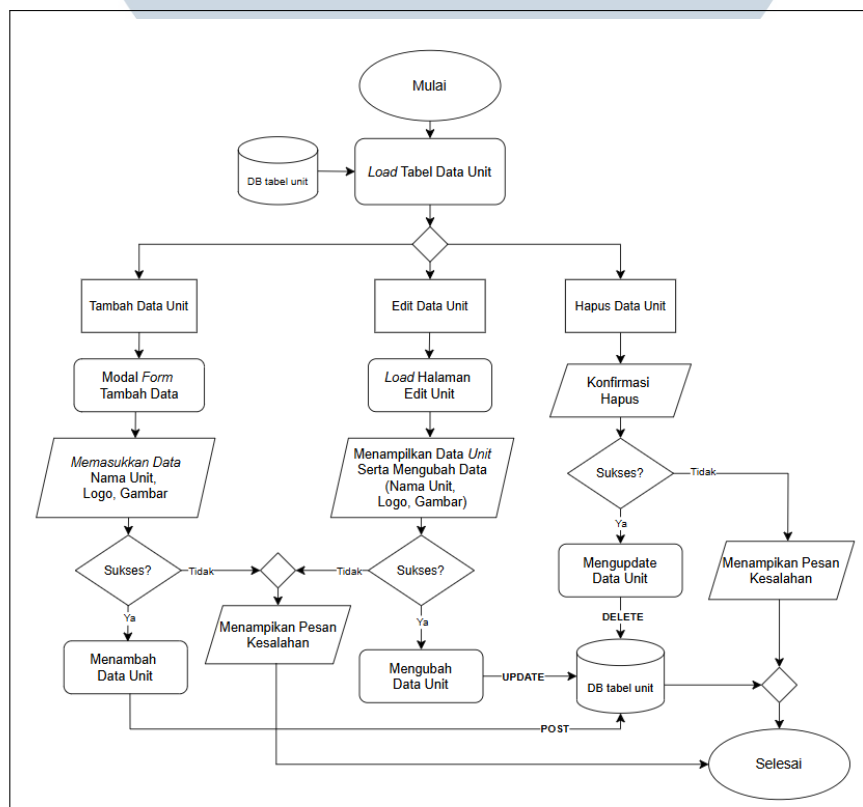
– Manage Users

Melalui halaman ini, admin dapat melakukan pengelolaan akun pengguna, termasuk:

- * Menambahkan akun baru
- * Memperbarui informasi akun pengguna
- * Menghapus akun pengguna

Fitur ini hanya dapat diakses oleh admin untuk menjaga keamanan dan pengelolaan sistem yang optimal.

• Flowchart Data Unit



Gambar 3.12. Flowchart Sub-proses Data Unit

Pada Gambar 3.12 ditampilkan sub-proses dari pengelolaan data unit pada Gambar 3.11. Halaman ini menampilkan tabel berisi daftar seluruh data unit

yang tersedia. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu *Create*, *Read*, *Update*, dan *Delete* terhadap data unit tersebut.

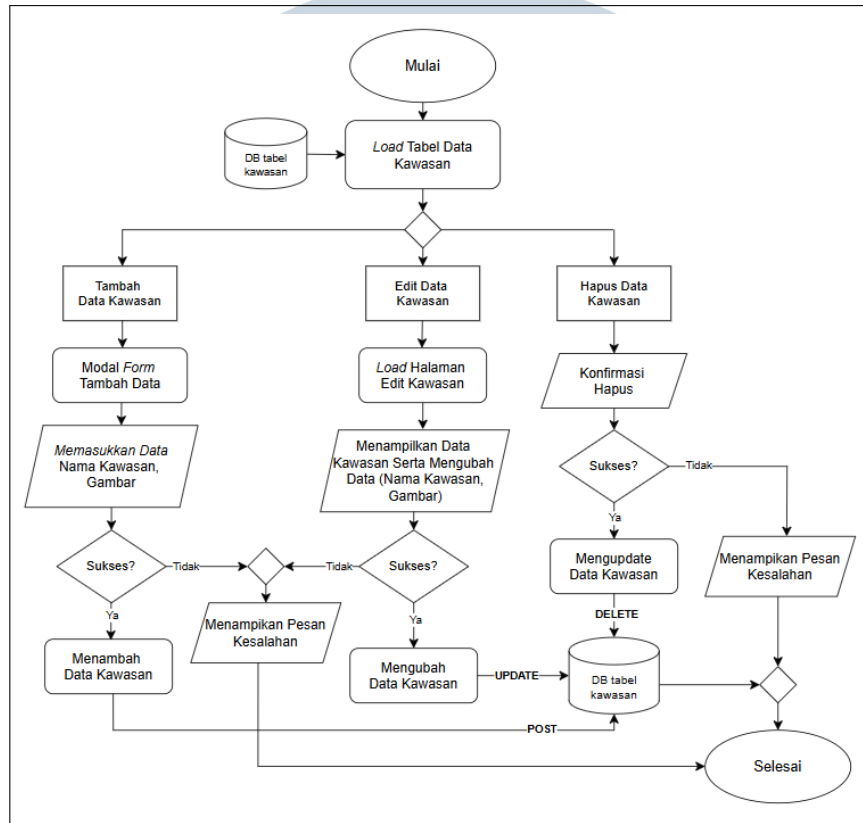
Untuk proses *create* atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput informasi unit, seperti nama unit, logo, dan gambar. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data unit akan disimpan ke dalam *database* pada tabel '**unit**' menggunakan metode **POST**.

Untuk proses *update* atau pengeditan data, sistem akan mengarahkan admin ke halaman edit unit. Pada halaman ini, admin dapat memperbarui informasi unit yang sudah ada, seperti nama, logo, maupun gambar. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**unit**' menggunakan metode **UPDATE**. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses *delete* atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data unit akan dihapus dari tabel '**unit**' dengan metode **DELETE**.



- **Flowchart Data Kawasan**



Gambar 3.13. Flowchart Sub-proses Data Kawasan

Pada Gambar 3.13 ditampilkan sub-proses dari pengelolaan data kawasan pada Gambar 3.11. Halaman ini menampilkan tabel berisi daftar seluruh data kawasan yang tersedia. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu **Create**, **Read**, **Update**, dan **Delete** terhadap data kawasan tersebut.

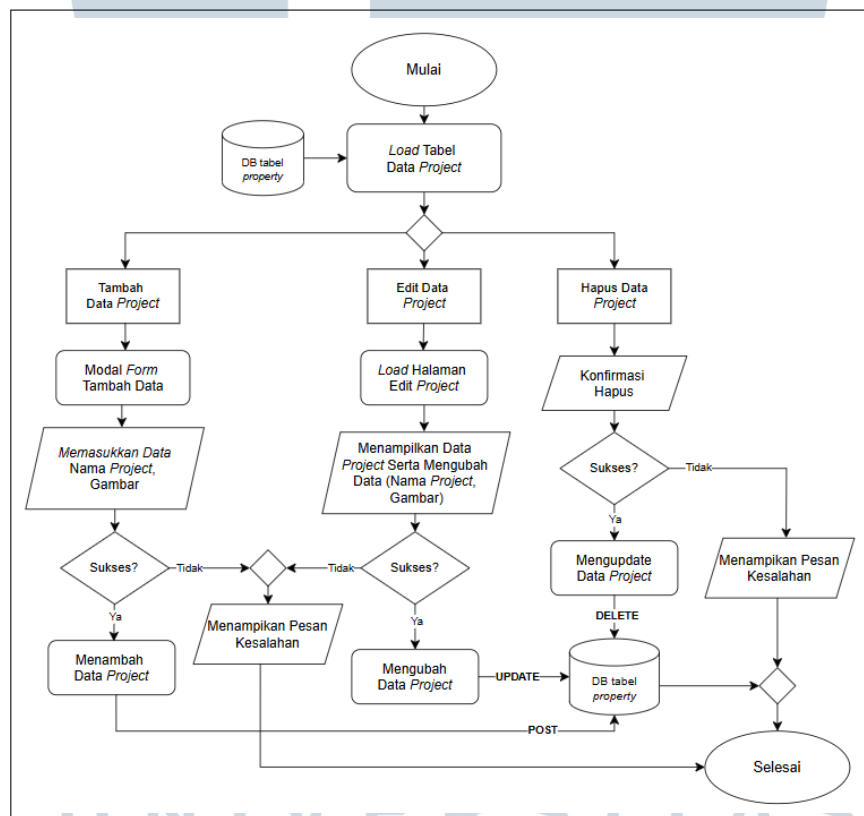
Untuk proses **create** atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput informasi kawasan, seperti nama kawasan dan gambar kawasan. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data kawasan akan disimpan ke dalam *database* pada tabel '**kawasan**' menggunakan metode **POST**.

Untuk proses **update** atau pengeditan data, sistem akan mengarahkan admin ke halaman edit kawasan. Pada halaman ini, admin dapat memperbarui

informasi kawasan yang sudah ada, seperti nama maupun gambar. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**kawasan**' menggunakan metode **UPDATE**. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses *delete* atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data kawasan akan dihapus dari tabel '**kawasan**' dengan metode **DELETE**.

- **Flowchart Data Project/Property**



Gambar 3.14. Flowchart Sub-proses Data Project

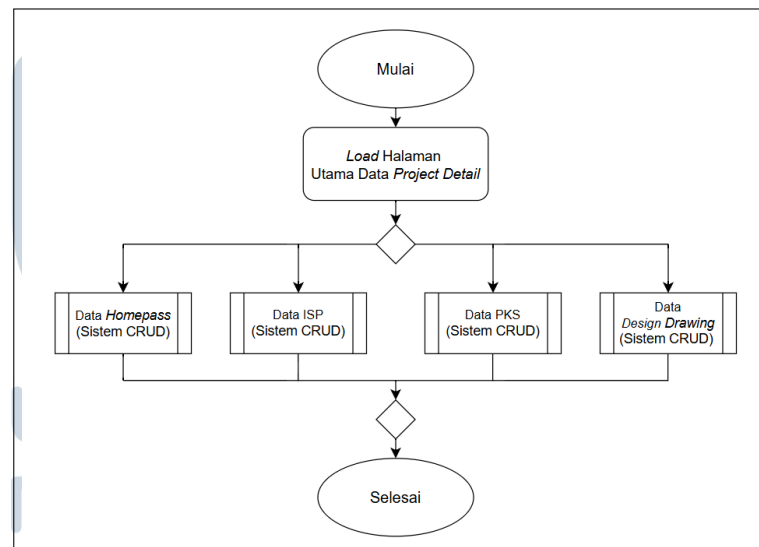
Pada Gambar 3.14 ditampilkan sub-proses dari pengelolaan data *project* pada Gambar 3.11. Halaman ini menampilkan tabel berisi daftar seluruh data *project* yang tersedia. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu **Create**, **Read**, **Update**, dan **Delete** terhadap data *project* tersebut.

Untuk proses *create* atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput informasi *project*, seperti nama *project* dan gambar. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data *project* akan disimpan ke dalam *database* pada tabel '**property**' menggunakan metode **POST**.

Untuk proses *update* atau pengeditan data, sistem akan mengarahkan admin ke halaman edit *project*. Pada halaman ini, admin dapat memperbarui informasi *project* yang sudah ada, seperti nama maupun gambar. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**property**' menggunakan metode **UPDATE**. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses *delete* atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data *project* akan dihapus dari tabel '**property**' dengan metode **DELETE**.

- **Flowchart Data Project Detail**



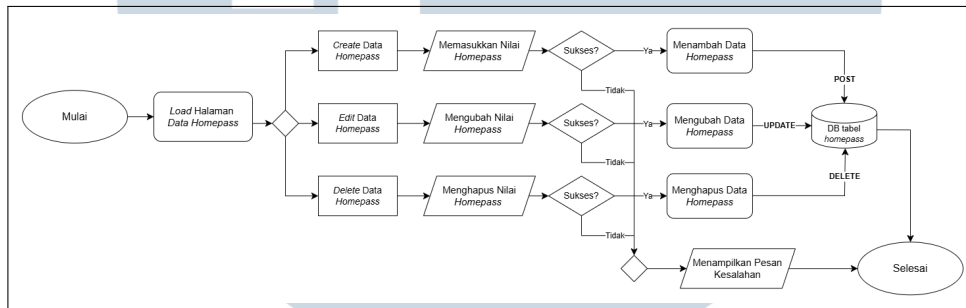
Gambar 3.15. Flowchart Sub-proses Data Project Detail

Pada Gambar 3.15 ditampilkan sub-proses pengelolaan data *project* yang merupakan bagian dari alur sistem pada Gambar 3.11. Pada halaman *Project*

Detail, sistem menampilkan berbagai informasi penting terkait data *project*, seperti data **Homepass**, **Internet Service Provider (ISP)**, **Perjanjian Kerja Sama (PKS)**, serta **Design Drawing (As Built)**.

Penjelasan lebih lanjut mengenai proses pengelolaan masing-masing data, termasuk operasi **CRUD (Create, Read, Update, dan Delete)**, akan dijabarkan secara rinci melalui diagram *flowchart* pada bagian sub-proses terkait.

- **Flowchart Data Homepass**



Gambar 3.16. Flowchart Sub-proses Data Homepass

Gambar 3.16 merupakan sub-proses dari pengelolaan data *homepass* yang juga termasuk dalam alur utama sistem pada Gambar 3.15. Pada halaman ini, admin memiliki akses penuh untuk melakukan operasi **CRUD (Create, Read, Update, dan Delete)** terhadap data *homepass*.

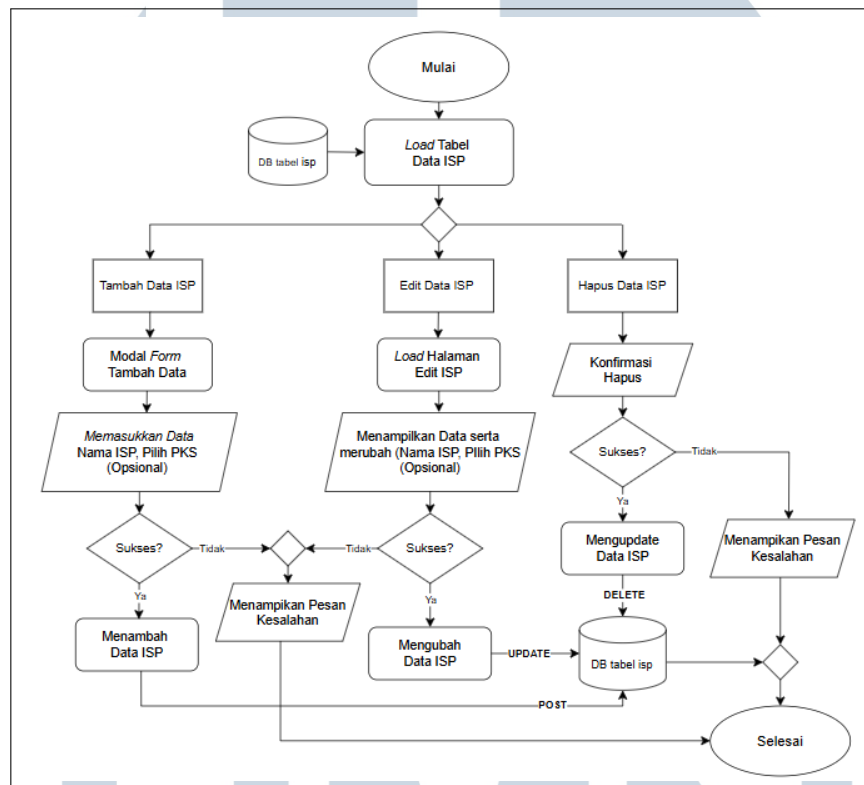
Setiap data yang dimasukkan akan melalui proses validasi. Jika validasi gagal, sistem akan menampilkan pesan kesalahan. Sebaliknya, jika validasi berhasil, sistem akan menampilkan notifikasi sukses dan melanjutkan proses penyimpanan ke dalam *database*.

Pada operasi tambah data, admin perlu menginput total *homepass*. Setelah melalui validasi, jika data valid, maka informasi tersebut akan disimpan ke dalam tabel '**homepass**' di *database* menggunakan metode **POST**.

Untuk proses edit data, admin dapat memperbarui nilai total *homepass*. Setelah perubahan berhasil divalidasi, sistem akan menyimpan perubahan tersebut ke dalam *database* tabel '**homepass**' menggunakan metode **UPDATE**. Sedangkan untuk proses hapus data, admin dapat menghapus data *homepass*. Jika proses penghapusan berhasil, sistem akan memperbarui

database dan menghapus data dari tabel '**homepass**' menggunakan metode **DELETE**.

- **Flowchart Data ISP**



Gambar 3.17. Flowchart Sub-proses Data ISP

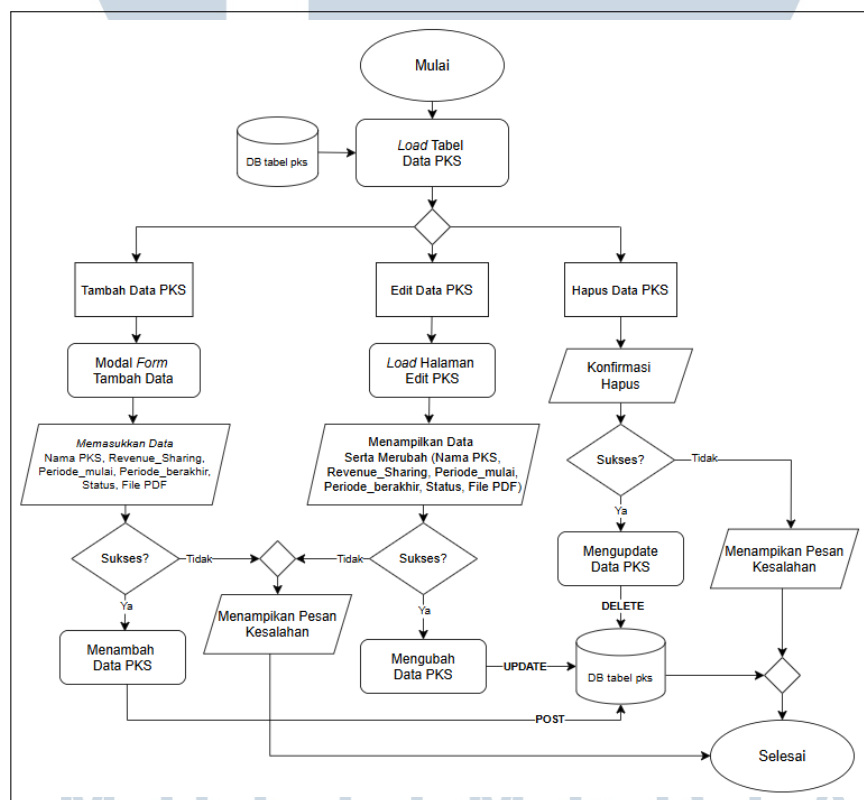
Pada Gambar 3.17 merupakan sub-proses dari pengelolaan data ISP yang juga termasuk dalam alur utama sistem pada Gambar 3.15. Halaman ini menampilkan tabel berisi daftar seluruh data ISP yang tersedia. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, terhadap data ISP tersebut.

Untuk proses **create** atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput nama ISP dan memilih data PKS (opsional) dengan mengambil `id_pks`. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data ISP akan disimpan ke dalam *database* pada tabel '**isp**' menggunakan metode **POST**.

Untuk proses **update** atau pengeditan data, sistem akan mengarahkan admin ke halaman edit ISP. Pada halaman ini, admin dapat memperbarui informasi ISP yang sudah ada, seperti nama, dan data PKS sesuai dengan `id_pks`. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**isp**' menggunakan metode **UPDATE**. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses **delete** atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data ISP akan dihapus dari tabel '**isp**' dengan metode **DELETE**.

- **Flowchart Data PKS**



Gambar 3.18. Flowchart Sub-proses Data PKS

Pada Gambar 3.18 merupakan sub-proses dari pengelolaan data PKS yang juga termasuk dalam alur utama sistem pada Gambar 3.15. Halaman ini menampilkan tabel berisi daftar seluruh data PKS yang tersedia. Admin

memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu **Create**, **Read**, **Update**, dan **Delete** terhadap data PKS tersebut.

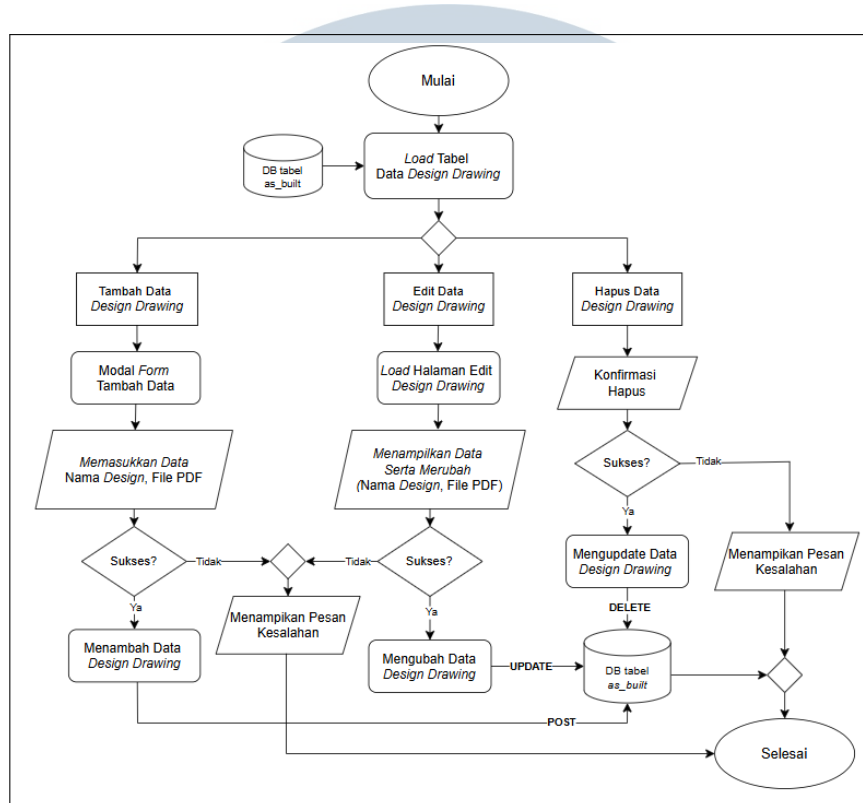
Untuk proses **create** atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput informasi PKS, seperti nama PKS, *revenue sharing*, tanggal periode mulai & periode berakhir, status ('Active', 'Expired', 'Will Expire'), dan *file* dokumen dalam format *.pdf. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data PKS akan disimpan ke dalam *database* pada tabel '**pks**' menggunakan metode **POST**.

Untuk proses **update** atau pengeditan data, sistem akan mengarahkan admin ke halaman edit PKS. Pada halaman ini, admin dapat memperbarui informasi PKS yang sudah ada, seperti nama PKS, *revenue sharing*, tanggal periode mulai & periode berakhir, status ('Active', 'Expired', 'Will Expire'), dan *file* dokumen dalam format *.pdf. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**pks**' menggunakan metode **UPDATE**. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses **delete** atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data PKS akan dihapus dari tabel '**pks**' dengan metode **DELETE**.



- **Flowchart Data Design Drawing**



Gambar 3.19. Flowchart Sub-proses Data Design Drawing

Pada Gambar 3.19 merupakan sub-proses dari pengelolaan data *design drawing* yang juga termasuk dalam alur utama sistem pada Gambar 3.15. Halaman ini menampilkan tabel berisi daftar seluruh data *design drawing* yang tersedia. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu **Create**, **Read**, **Update**, dan **Delete** terhadap data *design drawing* tersebut.

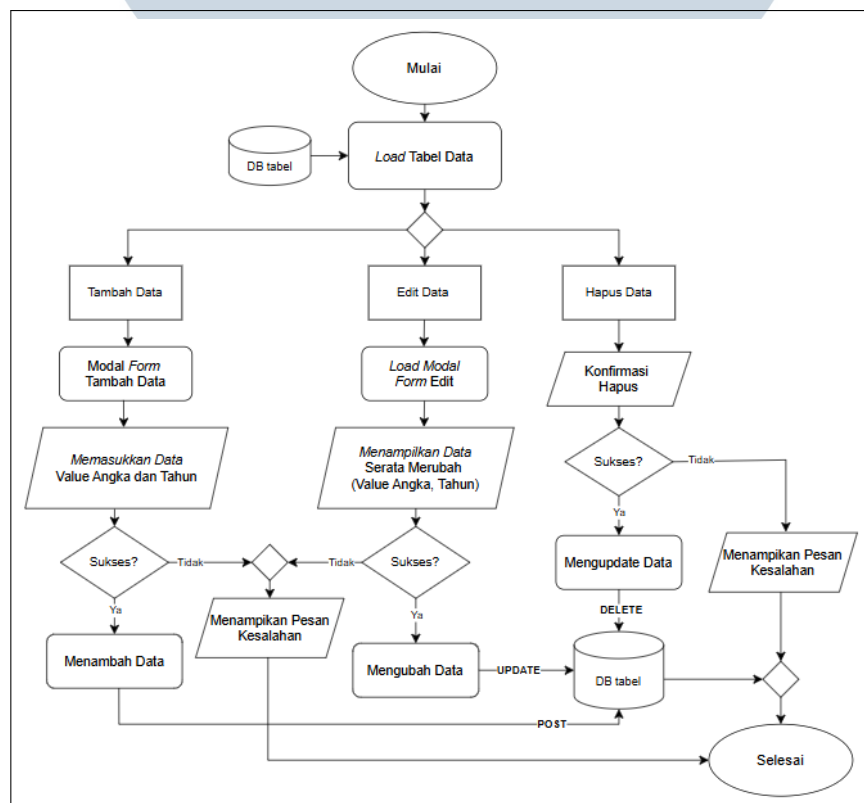
Untuk proses **create** atau penambahan data, sistem akan menampilkan *modal form* yang memungkinkan admin menginput informasi *design drawing*, seperti nama dan file dokumen dalam format *.pdf. Setelah formulir diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan. Namun, jika berhasil, data *design drawing* akan disimpan ke dalam *database* pada tabel **as_built** menggunakan metode **POST**.

Untuk proses **update** atau pengeditan data, sistem akan mengarahkan

admin ke halaman edit *design drawing*. Pada halaman ini, admin dapat memperbarui informasi yang sudah ada, seperti nama *design drawing* dan file dokumen dalam format *.pdf. Setelah proses pengeditan selesai dan divalidasi, perubahan akan disimpan ke dalam *database* pada tabel **as_built** menggunakan metode **UPDATE**. Jika proses perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses *delete* atau penghapusan data, sistem akan menampilkan *modal* konfirmasi kepada admin. Jika penghapusan gagal, maka sistem akan menampilkan pesan kesalahan. Namun, jika berhasil, data *design drawing* akan dihapus dari tabel **as_built** menggunakan metode **DELETE**.

- **Flowchart Data Performa Keuangan**



Gambar 3.20. Flowchart Sub-proses Data Performa Keuangan

Pada Gambar 3.20 ditampilkan proses sub-menu pengelolaan data performa keuangan, yang merupakan bagian dari alur utama sistem seperti terlihat pada

Gambar 3.11. Flowchart ini mencakup seluruh sub-menu dari *Performance Data*, yaitu: ***Revenue, Cost of Sales, EBIT, NPAT, NPBI, serta Rencana & Realisasi Capex.***

Halaman ini menampilkan tabel seluruh data yang tersedia, di mana admin memiliki hak akses penuh untuk melakukan operasi CRUD (*Create, Read, Update, dan Delete*).

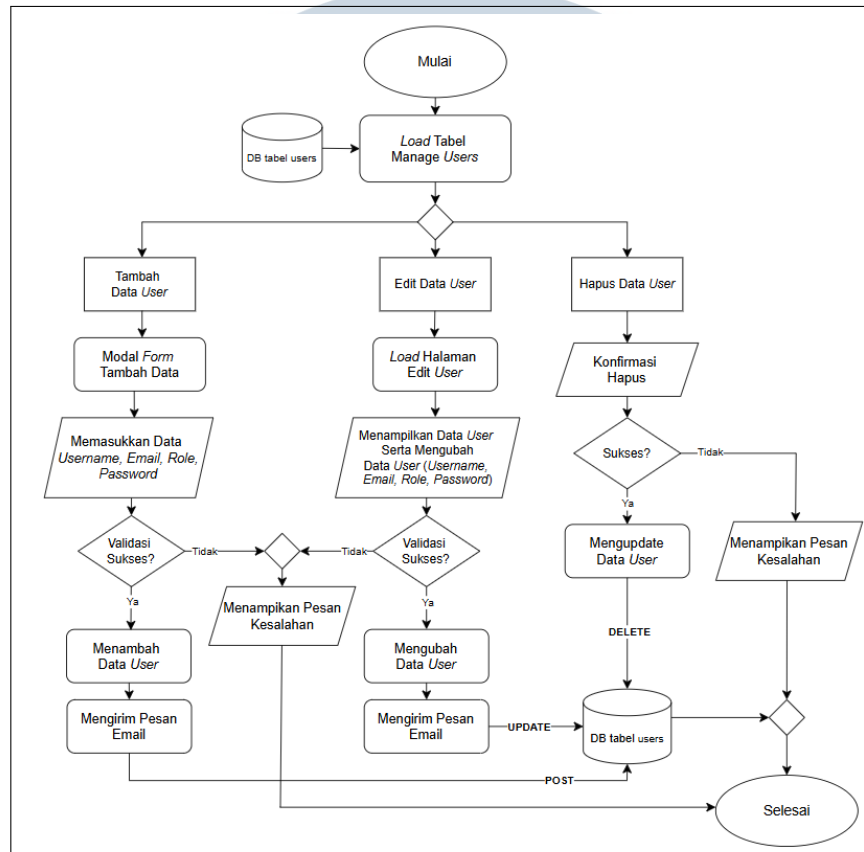
Untuk proses ***Create*** (penambahan data), sistem akan menampilkan modal *form* untuk admin menginput informasi seperti nilai atau angka dari Revenue/Cost of Sales/EBIT/NPAT/NPBI/Rencana & Realisasi Capex, serta data tahun. Setelah formulir diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data akan disimpan ke dalam tabel *database* terkait menggunakan metode **POST**.

Pada proses ***Update*** (pengeditan data), sistem akan menampilkan modal edit untuk admin memperbarui data yang sudah ada. Setelah proses pengeditan selesai dan divalidasi, perubahan akan disimpan ke dalam tabel *database* masing-masing menggunakan metode **UPDATE**. Apabila terjadi kesalahan, sistem akan menampilkan notifikasi kegagalan.

Untuk proses ***Delete*** (penghapusan data), sistem akan menampilkan modal konfirmasi kepada admin. Jika penghapusan berhasil, data akan dihapus dari tabel *database* menggunakan metode **DELETE**. Jika gagal, sistem akan menampilkan pesan kesalahan.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

- **Flowchart Manage Users**



Gambar 3.21. Flowchart Sub-proses Data Manage Users

Pada Gambar 3.21 ditampilkan sub-proses dari *manage users* pada Gambar 3.11. Halaman ini menampilkan tabel berisi daftar seluruh data pengguna. Admin memiliki hak akses penuh untuk melakukan operasi **CRUD**, yaitu **Create**, **Read**, **Update**, dan **Delete** terhadap data pengguna tersebut.

Untuk proses **create** atau penambahan data, sistem akan menampilkan *modal form* untuk admin menginput informasi username, email, role dan password. Setelah form diisi dan dikirim, sistem akan melakukan validasi. Jika validasi gagal, akan muncul pesan kesalahan, namun jika berhasil data pengguna akan disimpan ke dalam *database* pada tabel '**users**' menggunakan metode **POST** serta mengirimkan pesan email pembuatan akun pada email yang sudah didaftarkan.

Untuk proses **update** atau pengeditan data, sistem akan mengarahkan admin ke halaman edit users. Pada halaman ini, admin dapat memperbarui informasi

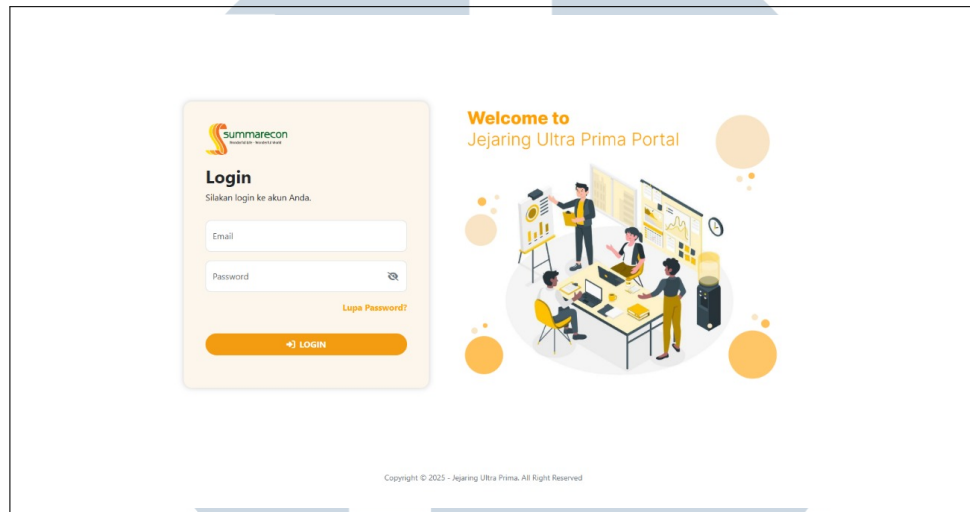
pengguna yang sudah ada, seperti username, email, role, dan password. Setelah proses edit selesai dan divalidasi, perubahan akan disimpan kembali ke dalam tabel *database* '**users**' menggunakan metode **UPDATE** dan akan mengirimkan pesan email perubahan informasi akun. Jika perubahan gagal, sistem akan menampilkan pesan kesalahan.

Sementara itu, untuk proses *delete* atau penghapusan data, sistem akan menampilkan modal konfirmasi hapus kepada admin. Jika penghapusan gagal, maka akan muncul pesan kesalahan. Namun jika berhasil, data pengguna akan dihapus dari tabel '**users**' dengan metode **DELETE**.



3.3.3 Tampilan Hasil Website Dashboard

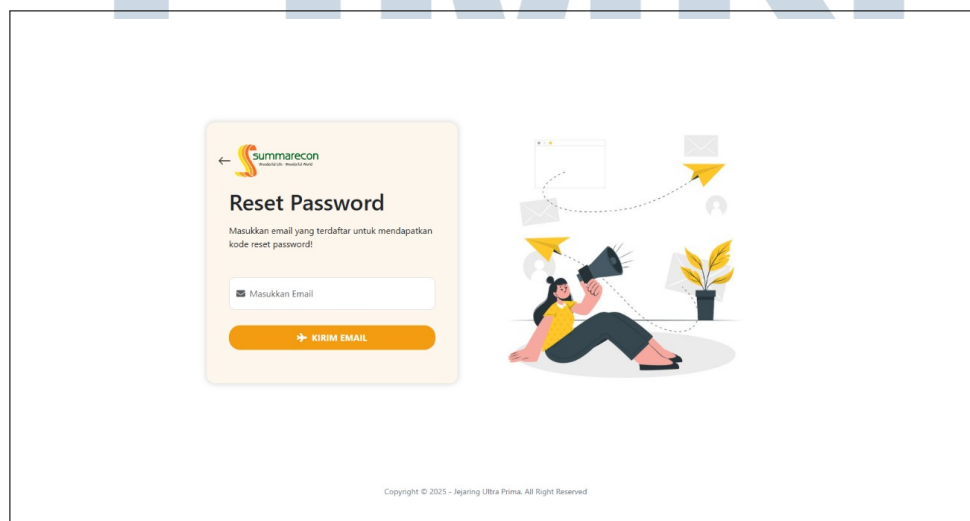
A Tampilan Login



Gambar 3.22. Tampilan Utama Login

Pada Gambar 3.22 ditampilkan antarmuka halaman *login*. Pada halaman ini, pengguna perlu memasukkan *email* dan *password* untuk dapat mengakses halaman *dashboard*.

B Tampilan Reset Password



Gambar 3.23. Tampilan Reset Password

Gambar 3.23 menunjukkan tampilan awal ketika tombol 'Lupa Password' diklik pada halaman *login*. Pengguna diminta untuk memasukkan email yang terdaftar, lalu menekan tombol 'Kirim Email'. Berikut Kode 3.1 merupakan proses pengiriman email *reset password*:

```

1 // Cek email ada di database
2 $stmt = $con->prepare("SELECT * FROM users WHERE email = ?");
3 $stmt->bind_param("s", $email);
4 $stmt->execute();
5 $result = $stmt->get_result();
6
7 if ($result && $result->num_rows === 1) {
8     $user = $result->fetch_assoc();
9
10    // Generate kode (6 digit angka)
11    $token = rand(100000, 999999);
12    $expiry = date("Y-m-d H:i:s", strtotime("+1 hour"));
13
14    // Update kode dan expiry
15    $stmt = $con->prepare("UPDATE users SET password_reset_token =
16    ?, password_reset_expiry = ? WHERE email = ?");
17    $stmt->bind_param("sss", $token, $expiry, $email);
18    $stmt->execute();
19
20    // Kirim email
21    $mail = new PHPMailer(true);
22    .....
23 }

```

Kode 3.1: Proses Validasi Email dan Pengiriman Kode Reset Password

Kode 3.1 menjelaskan proses pengiriman email *reset password* kepada pengguna. Penjelasan alur dari kode tersebut adalah sebagai berikut:

- **Pemeriksaan Email Pengguna:**

Sistem melakukan pengecekan apakah alamat email yang dimasukkan pengguna terdaftar di dalam *database*. Hal ini dilakukan menggunakan *prepared statement* untuk mencegah serangan SQL Injection.

```

1 $stmt = $con->prepare("SELECT * FROM users WHERE email = ?");
2 $stmt->bind_param("s", $email);
3 $stmt->execute();
4 $result = $stmt->get_result();

```

5

- **Validasi Email:**

Jika hasil pencarian menunjukkan bahwa email ditemukan (jumlah baris hasil pencarian adalah 1), maka sistem akan mengambil data pengguna terkait.

```
1 if ($result && $result->num_rows === 1) {  
2     $user = $result->fetch_assoc();  
3 }
```

- **Pembuatan Kode Reset Password:**

Sistem menghasilkan token/kode berupa angka acak 6 digit serta menetapkan waktu kedaluwarsa kode selama 1 jam dari waktu dibuat.

```
1 $token = rand(100000, 999999);  
2 $expiry = date("Y-m-d H:i:s", strtotime("+1 hour"));  
3
```

- **Penyimpanan Kode ke Database:**

Token/kode yang telah dibuat beserta waktu kedaluwarsanya akan disimpan ke dalam *database* pada kolom *password_reset_token* dan *password_reset_expiry*.

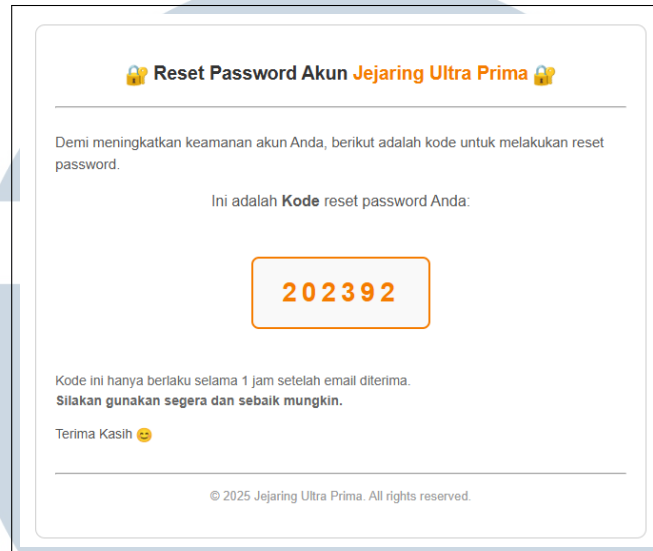
```
1 $stmt = $con->prepare("UPDATE users SET password_reset_token  
    = ?, password_reset_expiry = ? WHERE email = ?");  
2 $stmt->bind_param("sss", $token, $expiry, $email);  
3 $stmt->execute();  
4
```

- **Pengiriman Email ke Pengguna:**

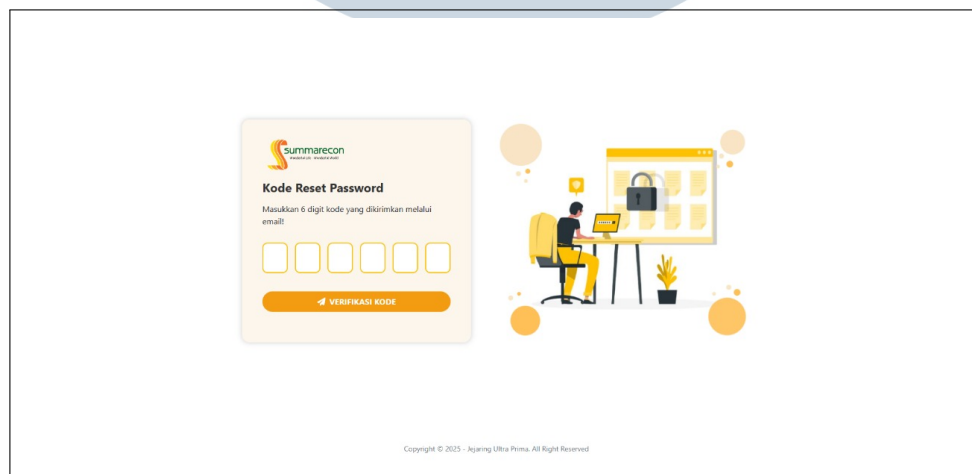
Setelah kode disimpan, sistem akan menggunakan library PHPMailer untuk mengirimkan email yang berisi kode *reset password* ke alamat email yang dimasukkan pengguna.

```
1 $mail = new PHPMailer(true);  
2
```

Gambar 3.24 di bawah merupakan tampilan pesan email kode *reset password* yang berhasil dikirimkan kepada email sudah divalidasi sebelumnya.

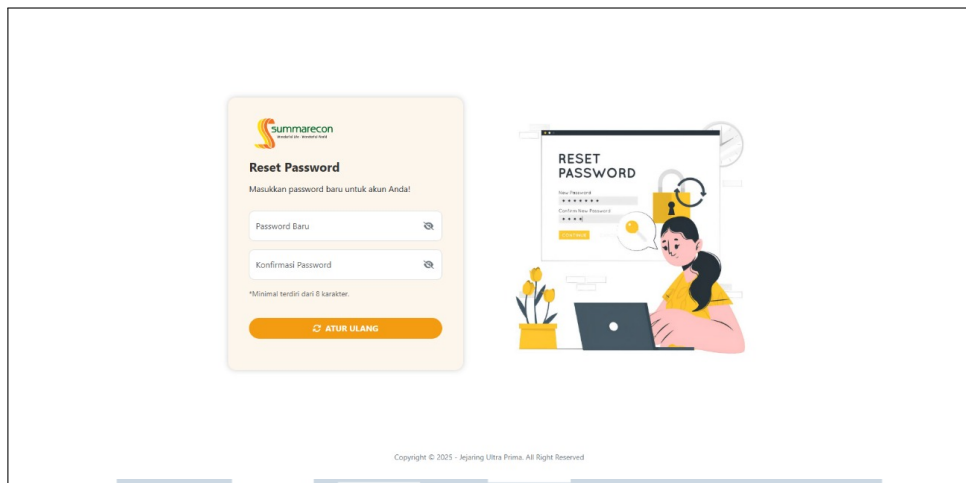


Gambar 3.24. Tampilan Pesan Email Reset Password



Gambar 3.25. Tampilan Input Kode

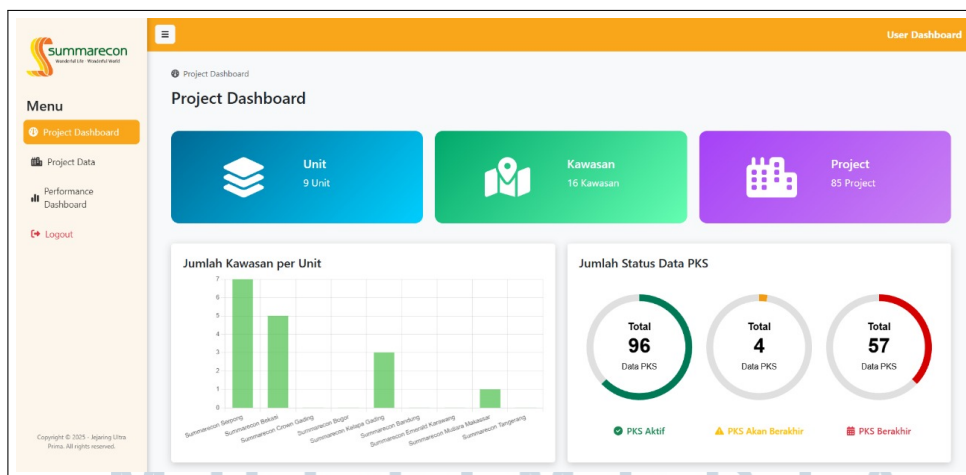
Gambar 3.25 di atas menampilkan halaman saat pengguna telah berhasil menerima email berisi kode *reset password* seperti pada Gambar 3.24 di atas. Pada halaman ini, pengguna diminta untuk memasukkan 6 digit kode dari pesan email tersebut.



Gambar 3.26. Tampilan Update Password

Setelah kode *reset password* yang dimasukkan berhasil divalidasi, pengguna akan diarahkan ke halaman seperti yang ditampilkan pada Gambar 3.26 di atas. Pada halaman ini, pengguna diminta untuk mengatur ulang kata sandi dengan memasukkan *password* baru serta konfirmasi *password* tersebut.

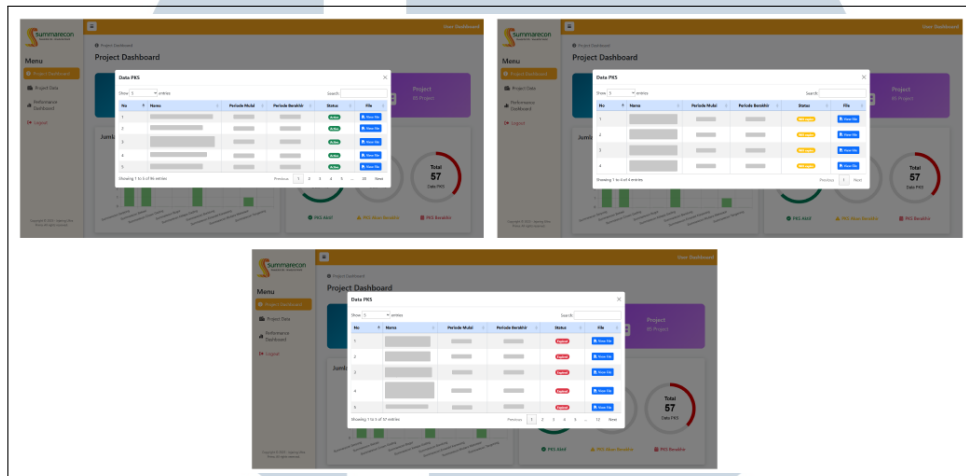
C Tampilan Dashboard User



Gambar 3.27. Tampilan Utama Dashboard User

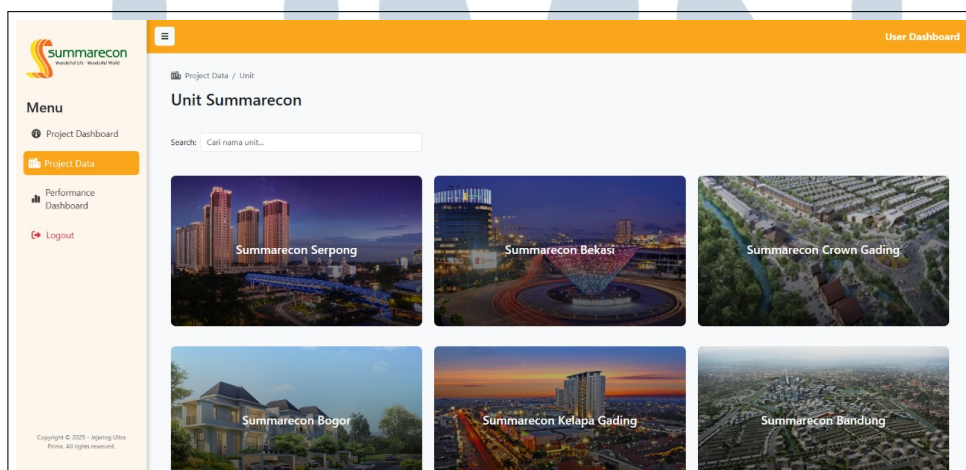
Halaman utama pada *dashboard user*, ditampilkan pada Gambar 3.27, masuk ke menu **project dashboard** menampilkan ringkasan (*summary*) dari data *project infrastructure*. Informasi yang ditampilkan meliputi jumlah unit, jumlah kawasan, serta total *project infrastructure*. Selain itu, pada halaman ini juga

menampilkan visualisasi data berupa grafik *bar chart* untuk jumlah kawasan per unit dan grafik *donut chart* yang menggambarkan status data PKS. Ketika pengguna memilih salah satu status pada grafik *donut chart* (**Active**, **Will Expire**, atau **Expired**), sistem akan menampilkan *data table* berisi informasi PKS sesuai dengan status yang dipilih. Tampilan tabel tersebut dapat dilihat pada Gambar 3.35 dibawah ini:



Gambar 3.28. Tampilan Tabel Data PKS (*Active*, *Will Expire*, *Expired*)

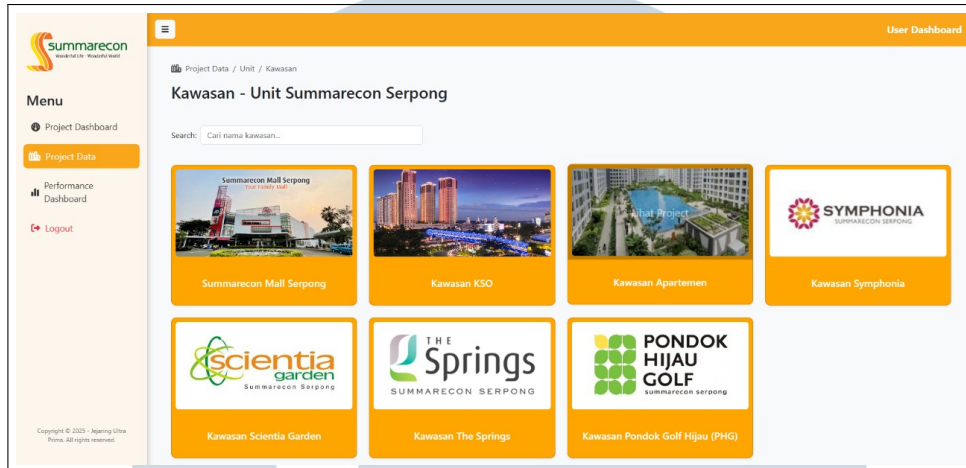
C.1 Tampilan Project Data Unit (User)



Gambar 3.29. Tampilan Data Unit (User)

Gambar 3.29 menampilkan tampilan awal ketika pengguna memilih menu **Project Data**. Data unit ditampilkan dalam bentuk *card* bergambar yang memuat informasi mengenai unit-unit Summarecon yang tersedia.

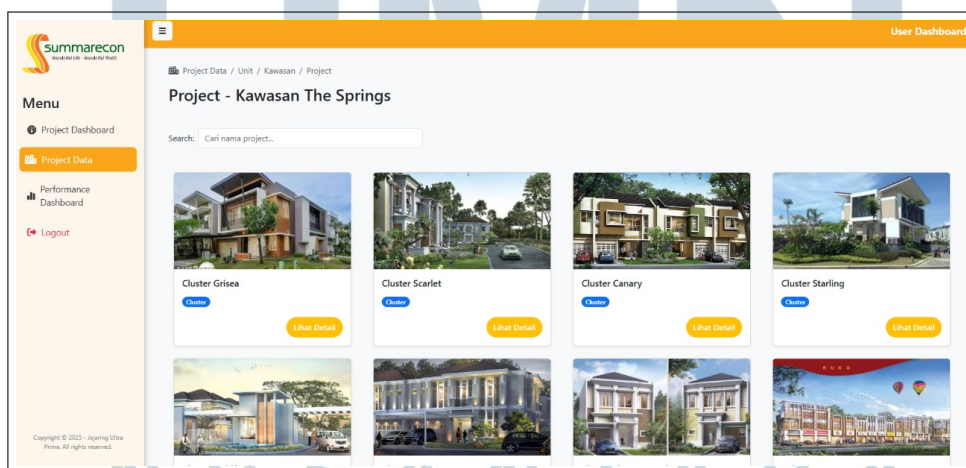
C.2 Tampilan Project Data Kawasan (User)



Gambar 3.30. Tampilan Data Kawasan (User)

Gambar 3.30 menampilkan antarmuka tampilan data kawasan berdasarkan unit yang telah dipilih sebelumnya. Data kawasan disajikan dalam bentuk *card* bergambar yang dilengkapi dengan nama kawasan, menampilkan informasi mengenai kawasan-kawasan Summarecon yang terkait dengan unit tersebut.

C.3 Tampilan Data Project Infrastructure (User)

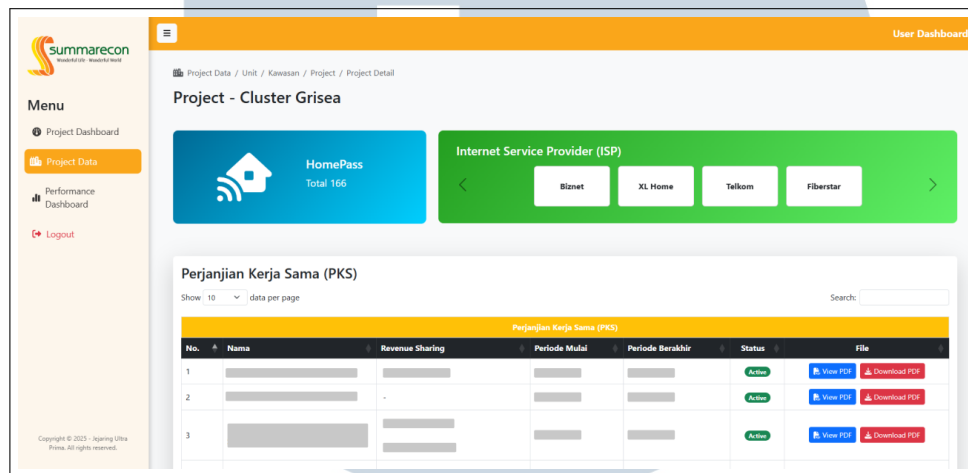


Gambar 3.31. Tampilan Data Project (User)

Gambar 3.31 menampilkan antarmuka tampilan data *project infrastructure* berdasarkan kawasan yang telah dipilih sebelumnya. Data *project infrastructure*

ditampilkan dalam bentuk *card* bergambar yang dilengkapi dengan nama project dan deskripsi *project* seperti (Ruko, Cluster, Mall maupun Apartemen), menampilkan informasi mengenai *project infrastructure* Summarecon yang terkait dengan kawasan tersebut.

C.4 Tampilan Project Detail (User)

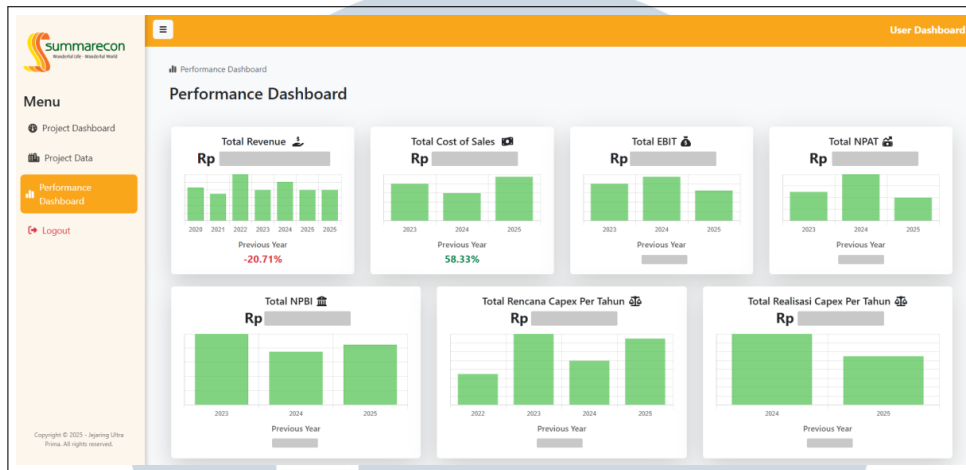


Gambar 3.32. Tampilan Data Project Detail (User)

Pada Gambar 3.32 akan menampilkan data-data seperti *homepass*, *ISP*, *PKS* dan *design drawing* dari data *project infrastructure* yang dipilih. Data *ISP* ditampilkan dalam bentuk *card carousel* yang interaktif untuk memudahkan pengguna dalam melihat data. Kemudian terdapat data *PKS* dan *design drawing* ditampilkan dalam *datatable*, dengan kolom *file* yang menyediakan fitur untuk melihat (*view*) dan mengunduh (*download*) dokumen dalam format *.pdf.

UNIVERSITAS
MULTIMEDIA
NUSANTARA

C.5 Tampilan Performance Dashboard (User)



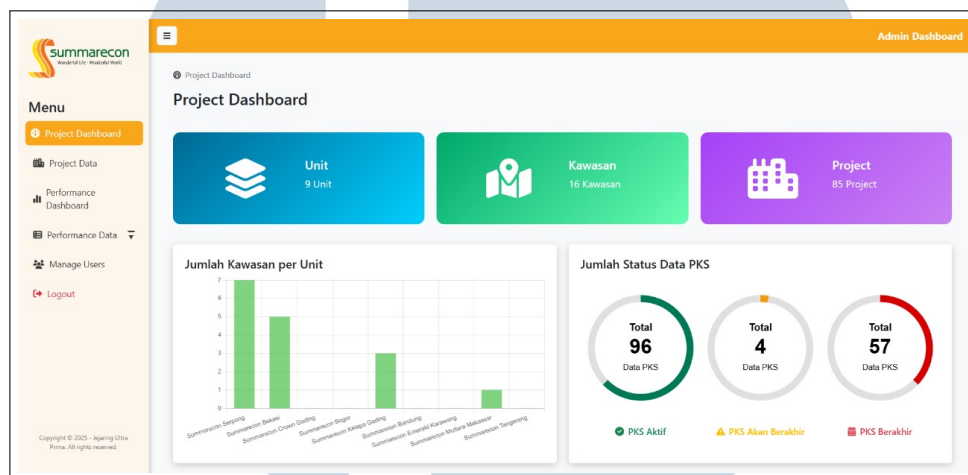
Gambar 3.33. Tampilan Performance Dashboard(User)

Halaman utama pada *Performance Dashboard*, seperti ditampilkan pada Gambar 3.33, menampilkan ringkasan (*summary*) data keuangan dalam bentuk visualisasi grafik *bar chart*. Informasi yang disajikan mencakup performa keuangan, antara lain:

- **Revenue**
- **Cost of Sales**
- **EBIT (*Earnings Before Interest and Taxes*)**
Laba perusahaan sebelum dikurangi beban bunga (*interest*) dan pajak (*tax*).
- **NPAT (*Net Profit After Tax*)**
Laba bersih yang diperoleh perusahaan setelah dikurangi semua biaya, termasuk bunga dan pajak.
- **NPBI (*Net Profit Before Interest*)**
Laba bersih perusahaan sebelum dikurangi beban bunga.
- **Rencana dan Realisasi Capex**
Capex (*Capital Expenditure*) adalah pengeluaran modal yang digunakan perusahaan untuk membeli, meningkatkan, atau memelihara aset tetap seperti properti.

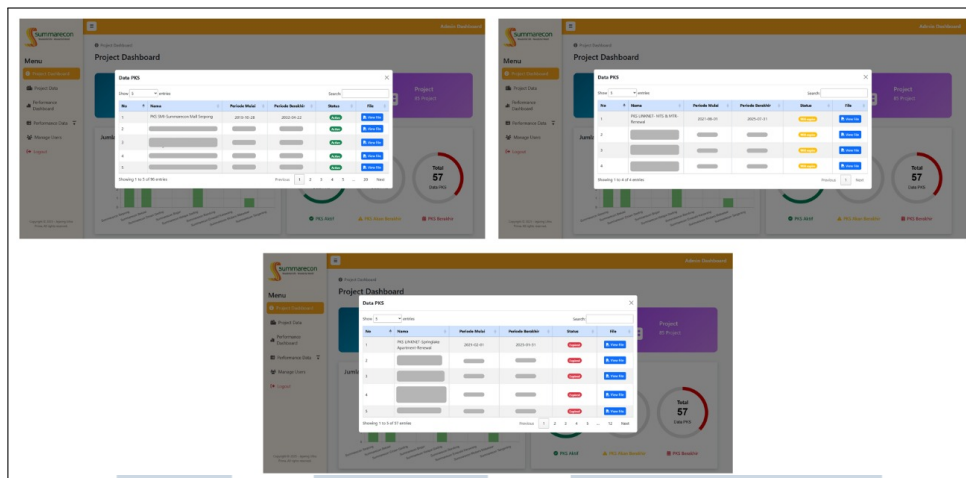
Visualisasi ini digunakan untuk membandingkan performa keuangan tahunan sebagai bagian dari evaluasi. Warna hijau menunjukkan peningkatan dibanding tahun sebelumnya, sedangkan warna merah menandakan penurunan.

D Tampilan Dashboard Admin



Gambar 3.34. Tampilan Utama Dashboard Admin

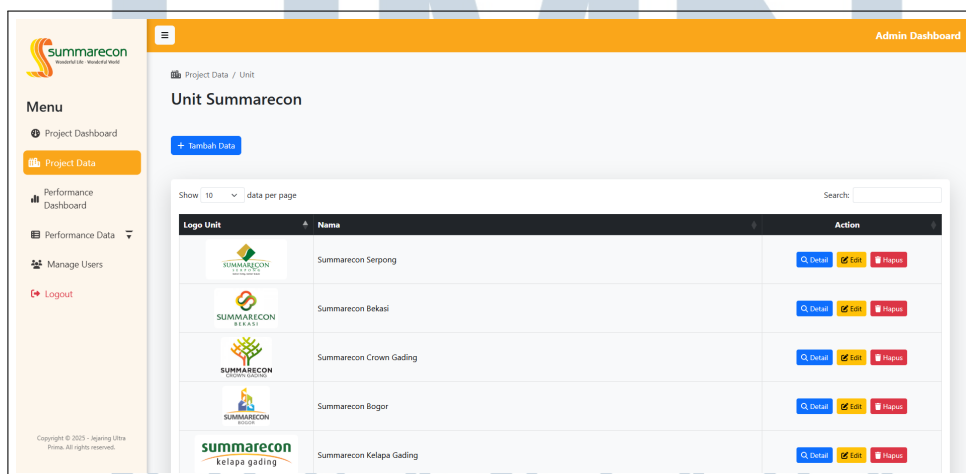
Halaman utama pada *dashboard admin*, ditampilkan pada Gambar 3.34, masuk ke menu **project dashboard** menampilkan ringkasan (*summary*) dari data *project infrastructure*. Informasi yang ditampilkan meliputi jumlah unit, jumlah kawasan, serta total *project infrastructure*. Selain itu, pada halaman ini juga menampilkan visualisasi data berupa grafik *bar chart* untuk jumlah kawasan per unit dan grafik *donut chart* yang menggambarkan status data PKS. Ketika pengguna memilih salah satu status pada grafik *donut chart* (**Active**, **Will Expire**, atau **Expired**), sistem akan menampilkan *data table* berisi informasi PKS sesuai dengan status yang dipilih. Tampilan tabel tersebut dapat dilihat pada Gambar 3.35 dibawah ini:



Gambar 3.35. Tampilan Tabel Data PKS (*Active*, *Will Expire*, *Expired*)

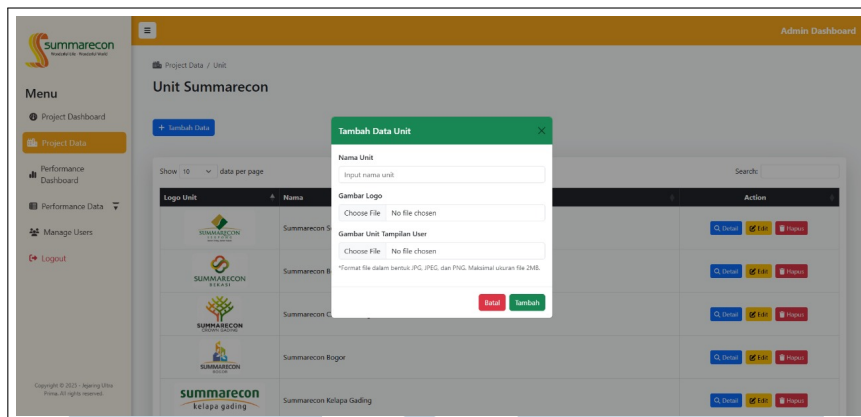
Gambar 3.35 merupakan tampilan *datatable* pada saat memilih *donut chart* ('*Active*', '*Will Expire*', '*Expired*'), tabel tersebut menampilkan nama PKS, waktu periode mulai, periode berakhir, dan status PKS tersebut. Terdapat juga kolom file untuk melihat dokumen yang terkait. Pada kolom status, warna status dibedakan warna hijau untuk PKS '*Active*', kuning untuk PKS yang akan berakhir '*Will Expire*', dan warna merah untuk PKS '*Expired*' atau sudah berakhir.

D.1 Tampilan Project Data Unit

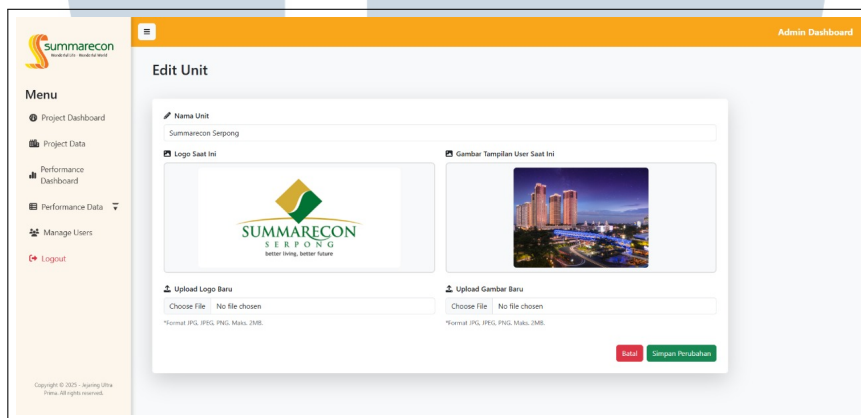


Gambar 3.36. Tampilan Data Unit

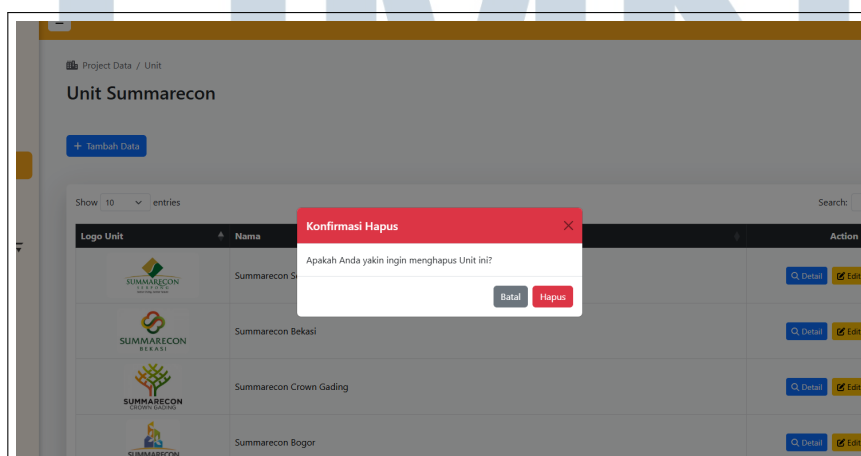
Pada Gambar 3.37 tampilan saat pertama kali memilih menu **project data**, dimana akan menampilkan tabel data unit yang berisikan unit-unit yang terdaftar.



Gambar 3.37. Tampilan Tambah Data Unit



Gambar 3.38. Tampilan Edit Data Unit

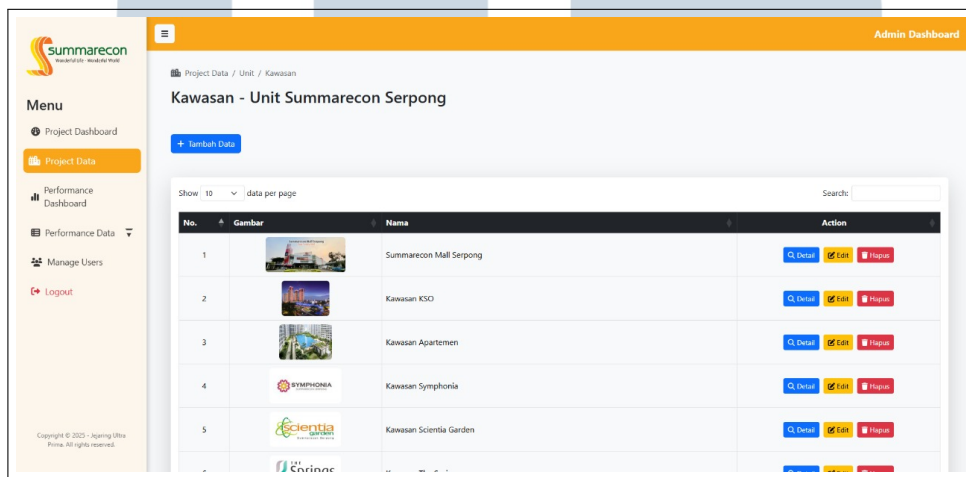


Gambar 3.39. Tampilan Hapus Data Unit

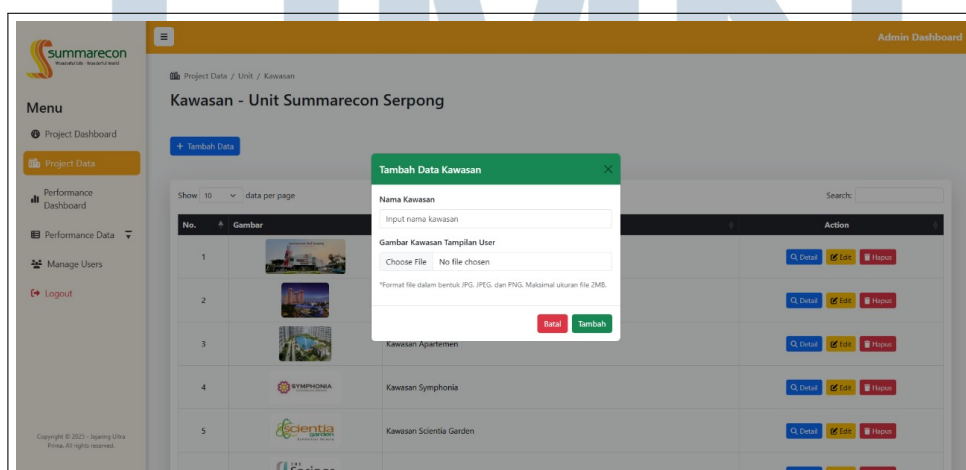
Gambar 3.37, 3.38, dan 3.39 menampilkan antarmuka sistem untuk pengelolaan data unit. Gambar 3.36 menunjukkan tampilan saat admin melakukan

penambahan data unit, di mana akan muncul *modal form* untuk mengisi nama unit, logo, dan gambar. Gambar 3.38 memperlihatkan tampilan ketika admin melakukan pengubahan data unit yang sudah ada. Pada proses ini, admin dapat mengunggah *file* logo dan gambar dengan ekstensi *.jpeg, .jpg, atau .png, dengan ukuran maksimum 2MB. Terakhir, Gambar 3.39 menampilkan *modal konfirmasi* yang muncul saat admin akan menghapus data unit.

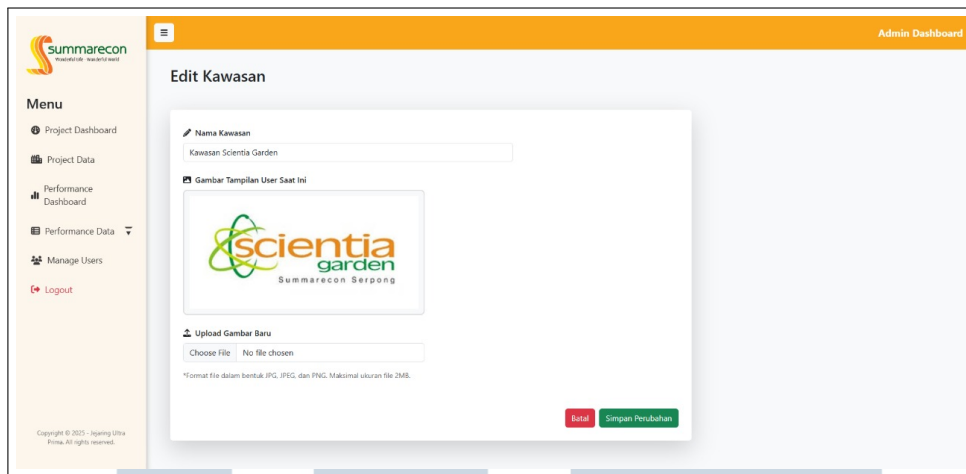
D.2 Tampilan Project Data Kawasan



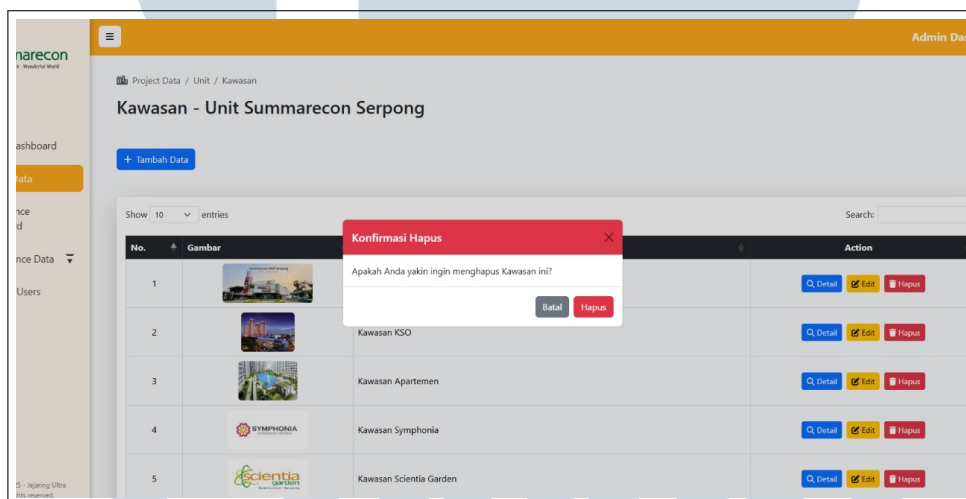
Gambar 3.40. Tampilan Data Kawasan



Gambar 3.41. Tampilan Tambah Data Kawasan



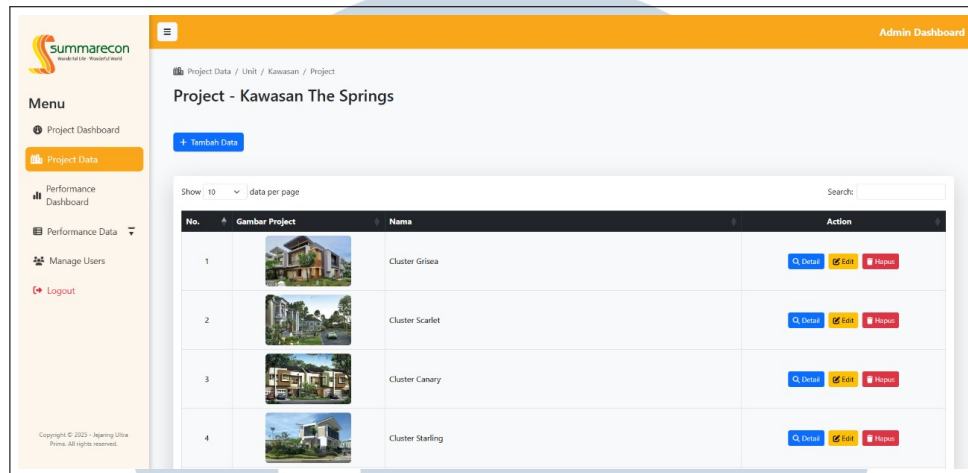
Gambar 3.42. Tampilan Edit Data Kawasan



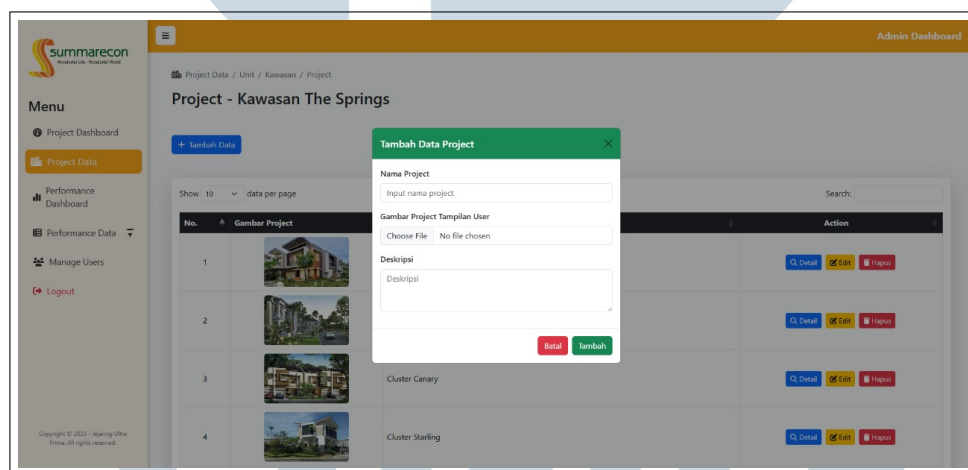
Gambar 3.43. Tampilan Hapus Data Kawasan

Gambar 3.40, 3.41, dan 3.42 menampilkan antarmuka sistem untuk pengelolaan data kawasan. Gambar 3.41 menunjukkan tampilan saat admin melakukan penambahan data kawasan, di mana akan muncul *modal form* untuk mengisi nama kawasan dan gambar. Gambar 3.42 memperlihatkan tampilan ketika admin melakukan pengubahan data kawasan yang sudah ada. Pada proses ini, admin dapat mengunggah *file* gambar dengan ekstensi *.jpeg, .jpg, atau .png, dengan ukuran maksimum 2MB. Terakhir, Gambar 3.43 menampilkan *modal konfirmasi* yang muncul saat admin akan menghapus data kawasan.

D.3 Tampilan Data Project Infrastructure

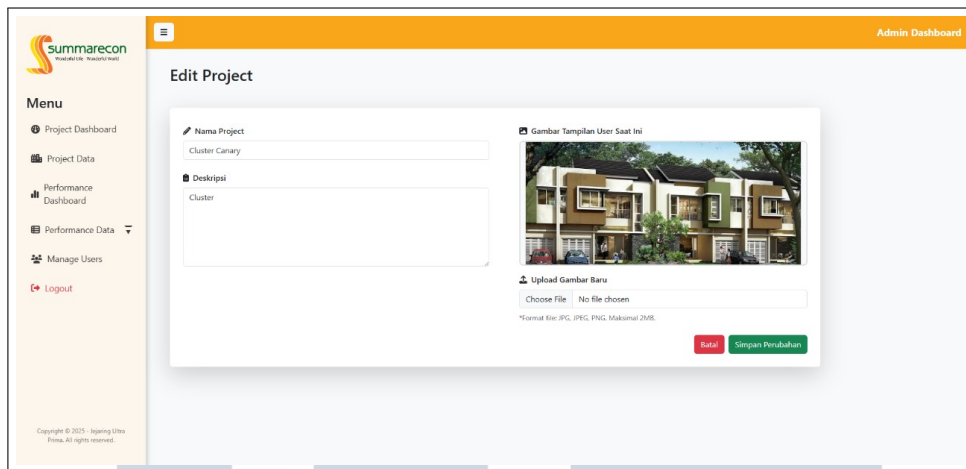


Gambar 3.44. Tampilan Data Project Infrastructure

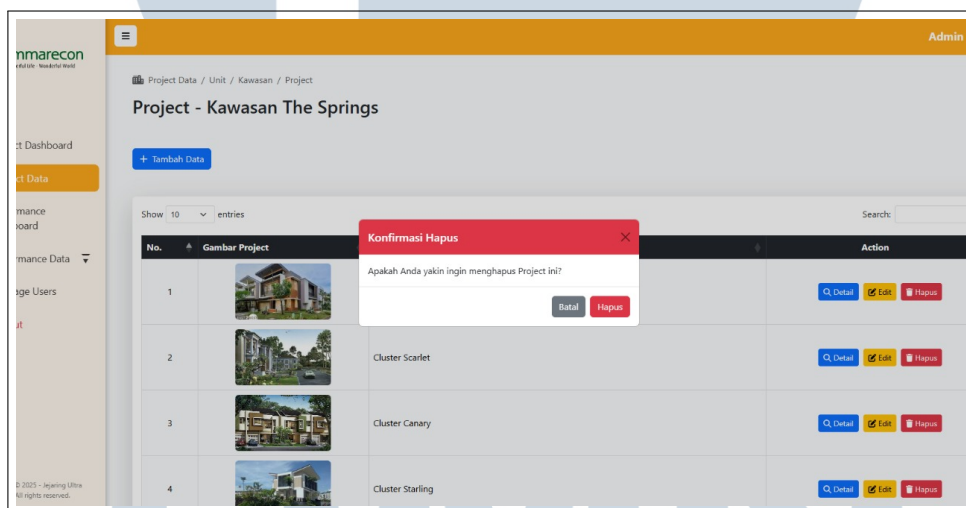


Gambar 3.45. Tampilan Tambah Data Project Infrastruktur

UNIVERSITAS
MULTIMEDIA
NUSANTARA



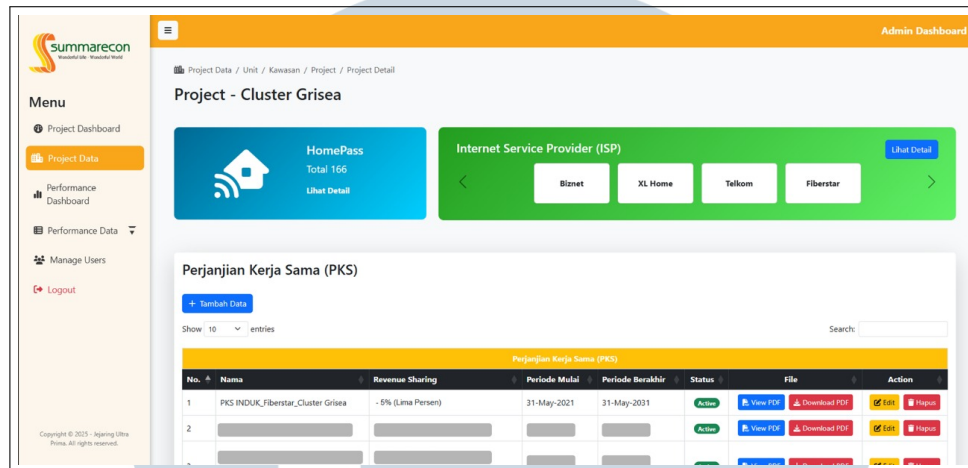
Gambar 3.46. Tampilan Edit Data Project Infrastruktur



Gambar 3.47. Tampilan Hapus Data Project Infrastruktur

Gambar 3.44, 3.45, dan 3.46 menampilkan antarmuka sistem untuk pengelolaan data *project* infrastruktur. Gambar 3.45 menunjukkan tampilan saat admin melakukan penambahan data *project*, di mana akan muncul *modal form* untuk mengisi nama *project* dan gambar. Gambar 3.46 memperlihatkan tampilan ketika admin melakukan pengubahan data *project* yang sudah ada. Pada proses ini, admin dapat mengunggah *file* gambar dengan ekstensi *.jpeg, .jpg, atau .png, dengan ukuran maksimum 2MB. Terakhir, Gambar 3.47 menampilkan *modal konfirmasi* yang muncul saat admin akan menghapus data *project* infrastruktur.

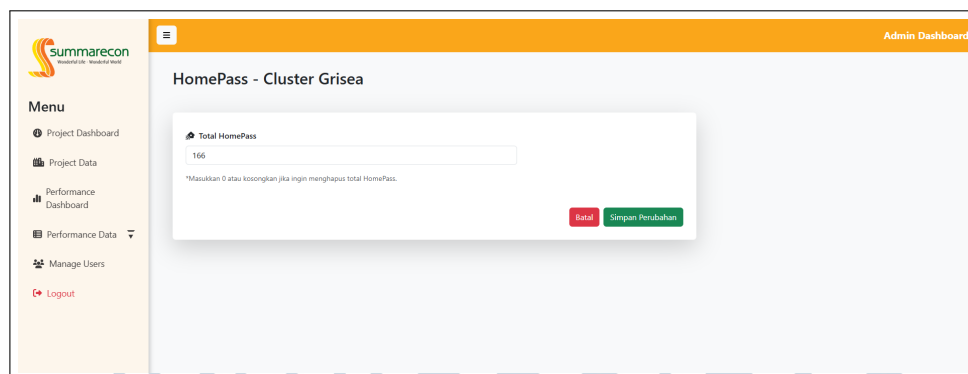
D.4 Tampilan Project Detail



Gambar 3.48. Tampilan Project Detail

Pada Gambar 3.48 akan menampilkan data-data seperti *homepass*, ISP, PKS dan *design drawing* dari data *project infrastructure* yang dipilih. Berikut ini akan ditampilkan tampilan lebih rinci dari pengelolaan data-data tersebut:

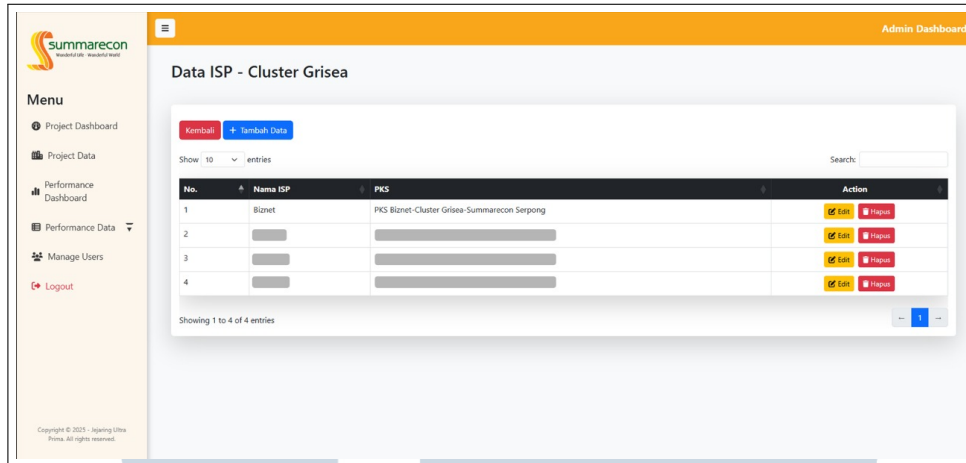
- Tampilan Data Homepass



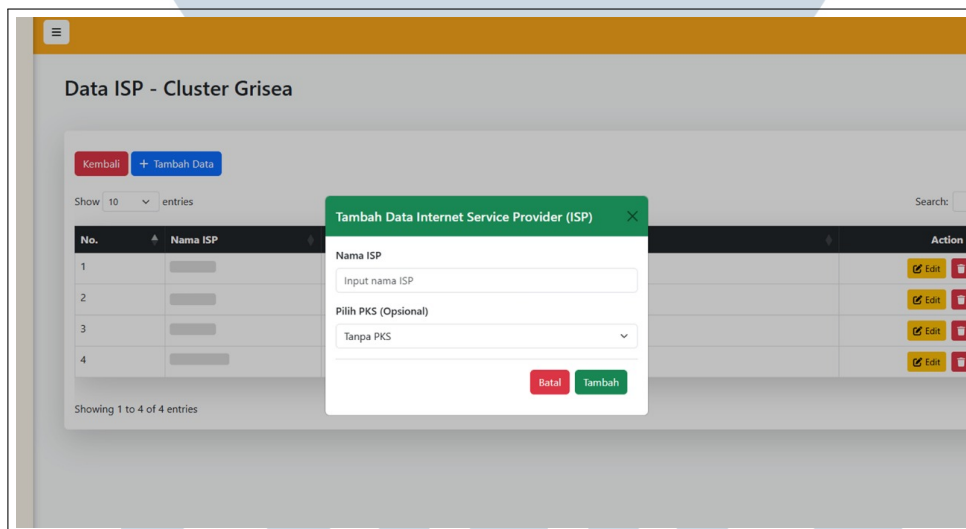
Gambar 3.49. Tampilan Data Homepass

Gambar 3.49 menunjukkan tampilan halaman pengelolaan data *homepass*. Pada halaman ini, admin dapat melakukan operasi penambahan (*create*), pengeditan (*edit*), dan penghapusan (*delete*) terhadap nilai data *homepass*.

- **Tampilan Data ISP**

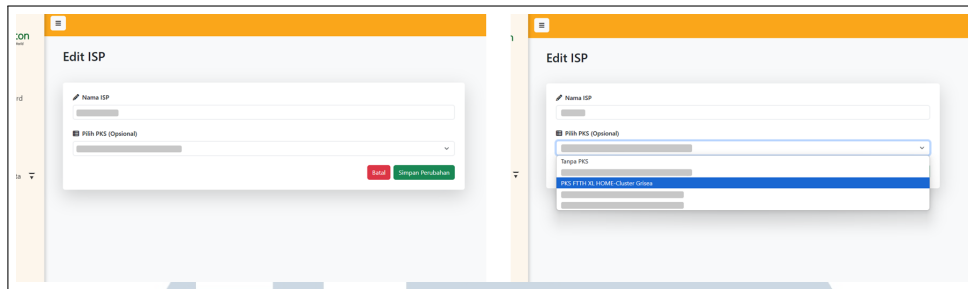


Gambar 3.50. Tampilan Data ISP



Gambar 3.51. Tampilan Tambah Data ISP

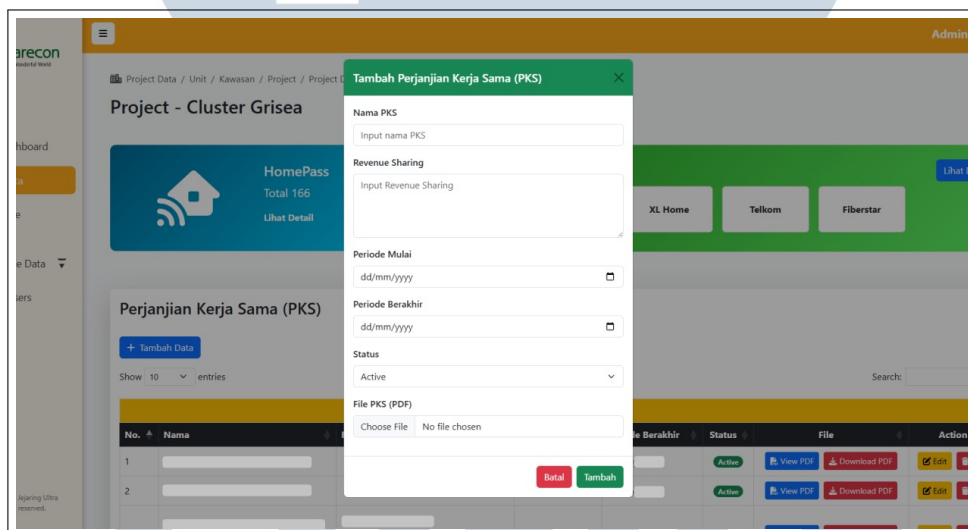
Gambar 3.50 menampilkan data **Internet Service Provider (ISP)** yang terkait dengan *project* tertentu. Sementara itu, Gambar 3.51 menunjukkan tampilan saat admin menambahkan data ISP melalui *modal form*. Pada form tersebut, admin mengisi nama ISP dan memilih data PKS (opsional) dari daftar yang tersedia, di mana pilihan PKS diambil berdasarkan `id_pks` yang telah ada sebelumnya.



Gambar 3.52. Tampilan Edit Data ISP

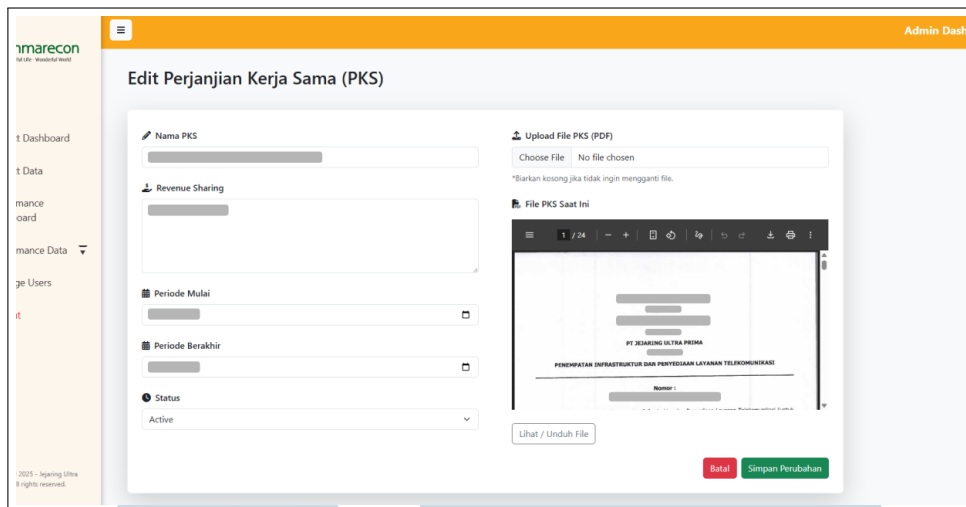
Gambar 3.52 memperlihatkan tampilan ketika admin melakukan perubahan data ISP yang sudah ada. Pada proses ini, admin dapat mengunggah nama ISP dan pilih PKS yang terkait.

- **Tampilan Data PKS**



Gambar 3.53. Tampilan Tambah Data PKS

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.54. Tampilan Edit Data PKS

Gambar 3.53 menunjukkan tampilan saat admin menambahkan data PKS melalui *modal form*. Pada form tersebut, admin mengisi nama PKS, revenue sharing, waktu periode (mulai & berakhir), status PKS ('Active', 'Will Expire', 'Expired') dan file dokumen dalam format *.pdf.

Pada Gambar 3.54 memperlihatkan tampilan ketika admin melakukan pengubahan data PKS yang sudah ada. Pada proses ini, admin dapat memasukkan nama PKS, revenue sharing, waktu periode (mulai & berakhir), status PKS ('Active', 'Will Expire', 'Expired') dan file dokumen dalam format *.pdf.

```

1 $updateStatusWillExpireQuery = "
2     UPDATE pks
3     SET status = 'Will expire'
4     WHERE property_id = $property_id
5     AND periode_berakhir >= CURDATE()
6     AND periode_berakhir <= DATE_ADD(CURDATE(), INTERVAL 6
7     MONTH)
8     AND status != 'Will expire'
9     AND status != 'Expired';
10 $resultUpdateWillExpire = mysqli_query($con,
11     $updateStatusWillExpireQuery);
12 if (!$resultUpdateWillExpire) {
13     echo "Error mengupdate status PKS (Will expire): " .
14     mysqli_error($con);

```


13 }

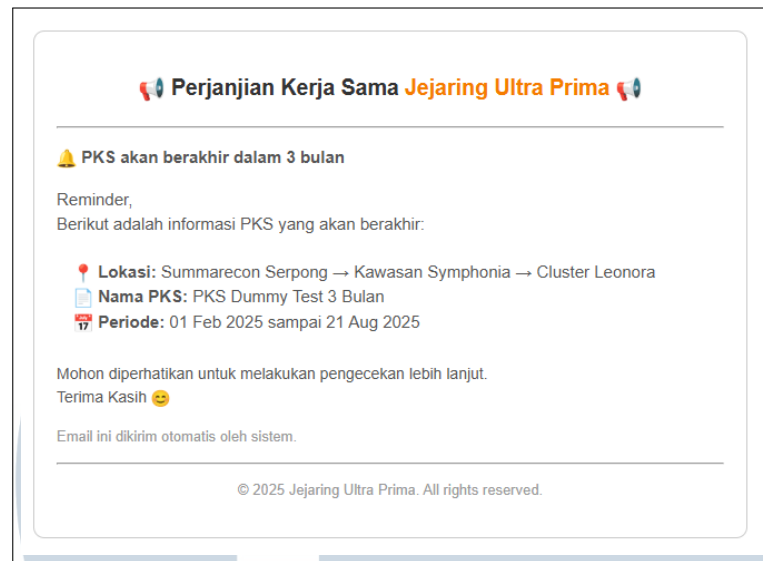
Kode 3.2: Query update status menjadi 'Will expire'

Kode 3.2 digunakan untuk memperbarui status data PKS menjadi Will expire secara otomatis apabila masa berlakunya masih aktif namun akan berakhir dalam rentang waktu 6 bulan ke depan. *Query* ini bekerja dengan mengecek data pada tabel pks berdasarkan property_id tertentu. Baris akan diperbarui jika nilai pada kolom periode_berakhir lebih besar atau sama dengan tanggal saat ini (CURDATE()) dan kurang dari atau sama dengan 6 bulan ke depan (DATE_ADD(CURDATE(), INTERVAL 6 MONTH)), serta status saat ini belum bernilai Will expire maupun Expired. Jika proses pembaruan gagal dijalankan, maka akan ditampilkan pesan kesalahan melalui fungsi mysqli_error().

```
1 $updateStatusExpiredQuery = "  
2     UPDATE pks  
3     SET status = 'Expired'  
4     WHERE property_id = $property_id  
5     AND periode_berakhir < CURDATE()  
6     AND status != 'Expired';  
7  
8 $resultUpdateExpired = mysqli_query($con,  
9     $updateStatusExpiredQuery);  
10 if (!$resultUpdateExpired) {  
11     echo "Error mengupdate status PKS (Expired): " .  
12     mysqli_error($con);  
13 }
```

Kode 3.3: Query update status menjadi 'Expired'

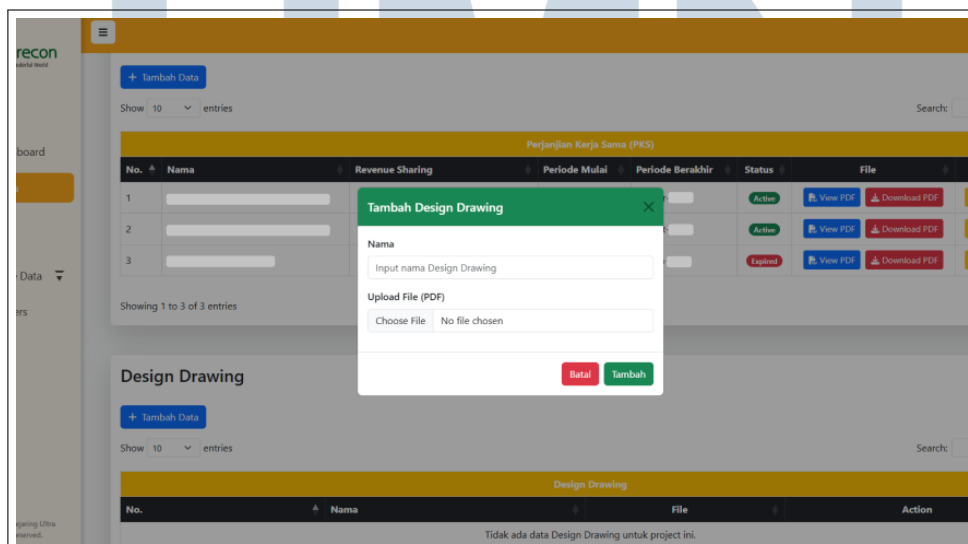
Kode 3.3 atas merupakan *query* untuk memperbarui status data PKS menjadi Expired secara otomatis apabila masa berlaku PKS tersebut telah berakhir, yaitu ketika tanggal pada kolom periode_berakhir lebih kecil dari tanggal saat ini (CURDATE()) dan status sebelumnya belum berstatus Expired. *Query* akan memperbarui data pada tabel pks berdasarkan property_id tertentu. Jika terjadi kesalahan saat menjalankan *query*, maka akan ditampilkan pesan error dengan menggunakan fungsi mysqli_error().



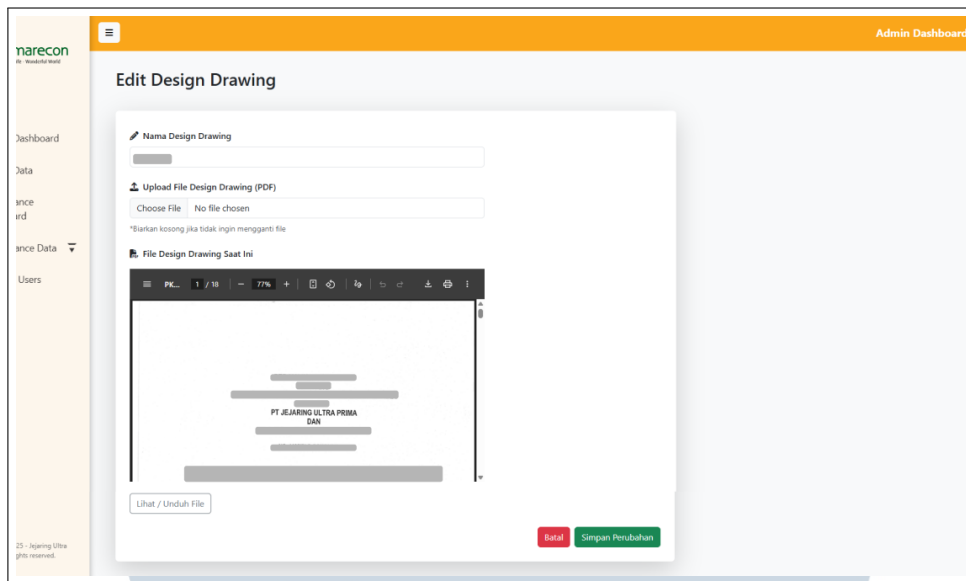
Gambar 3.55. Tampilan Pesan Email Reminder PKS

Gambar 3.55 menunjukkan tampilan pesan email pengingat (*reminder*) PKS yang akan dikirimkan secara otomatis ketika masa berlaku suatu PKS mendekati tanggal *periode_akhir*, yaitu pada rentang waktu 6, 3, dan 2 bulan sebelum berakhir. Email *reminder* ini hanya dikirimkan kepada pengguna yang memiliki *role* sebagai admin.

- **Tampilan Data Design Drawing**



Gambar 3.56. Tampilan Tambah Data Design Drawing

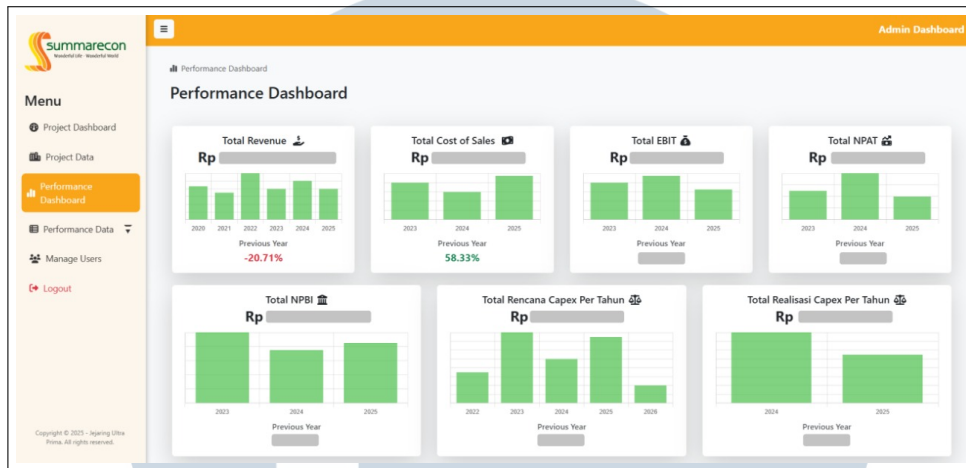


Gambar 3.57. Tampilan Edit Data Design Drawing

Gambar 3.56 menunjukkan tampilan saat admin menambahkan data *Design Drawing* melalui *modal form*. Pada form tersebut, admin mengisi nama *Design Drawing* dan file dokumen dalam format *.pdf.

Pada Gambar 3.57 memperlihatkan tampilan ketika admin melakukan pengubahan data *Design Drawing* yang sudah ada. Pada proses ini, admin dapat mengunggah nama *Design Drawing* dan file dokumen dalam format *.pdf.

D.5 Tampilan Performance Dashboard

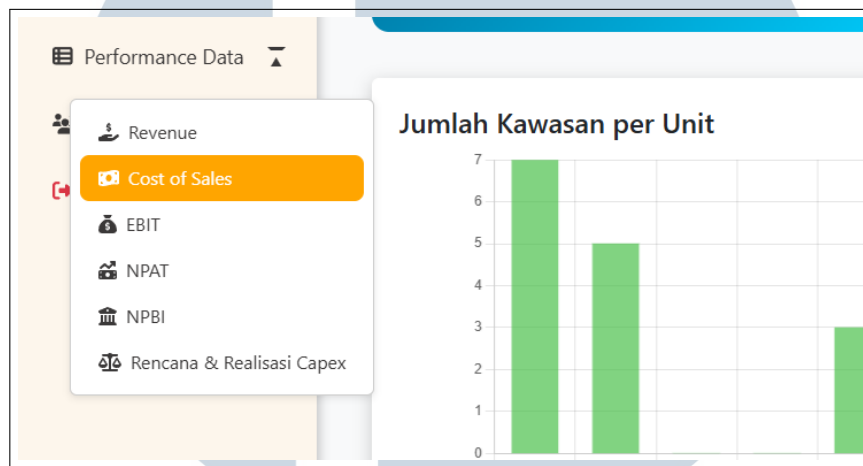


Gambar 3.58. Tampilan Menu Performance Dashboard

Halaman utama pada **Performance Dashboard**, seperti ditampilkan pada Gambar 3.58, menampilkan ringkasan (*summary*) data keuangan dalam bentuk visualisasi grafik *bar chart*. Informasi yang disajikan mencakup performa keuangan, antara lain:

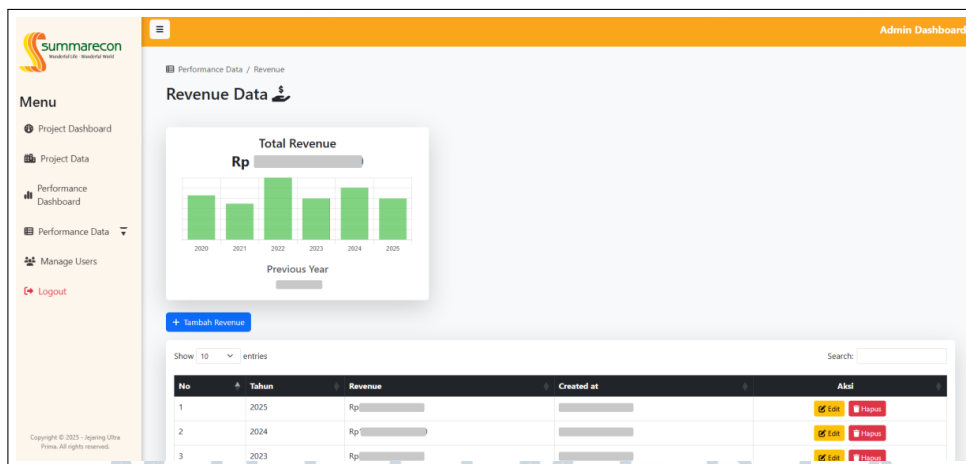
- **Revenue**
- **Cost of Sales**
- **EBIT (*Earnings Before Interest and Taxes*)**
Laba perusahaan sebelum dikurangi beban bunga (*interest*) dan pajak (*tax*).
- **NPAT (*Net Profit After Tax*)**
Laba bersih yang diperoleh perusahaan setelah dikurangi semua biaya, termasuk bunga dan pajak.
- **NPBI (*Net Profit Before Interest*)**
Laba bersih perusahaan sebelum dikurangi beban bunga.
- **Rencana dan Realisasi Capex**
Capex (*Capital Expenditure*) adalah pengeluaran modal yang digunakan perusahaan untuk membeli, meningkatkan, atau memelihara aset tetap seperti properti.

Visualisasi ini digunakan untuk membandingkan performa keuangan tahunan sebagai bagian dari evaluasi. Warna hijau menunjukkan peningkatan dibanding tahun sebelumnya, sedangkan warna merah menandakan penurunan. Untuk mengatur data seperti revenue dan data performa lainnya, pengguna dapat mengakses menu **Performance Data** sebagaimana ditunjukkan pada Gambar 3.59 dibawah:

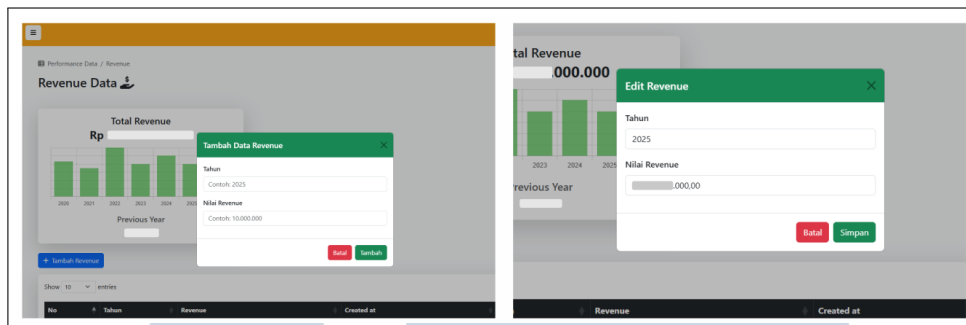


Gambar 3.59. Menu Dropdown Performance Data

- **Tampilan Data Revenue**



Gambar 3.60. Tampilan Performance Data Revenue

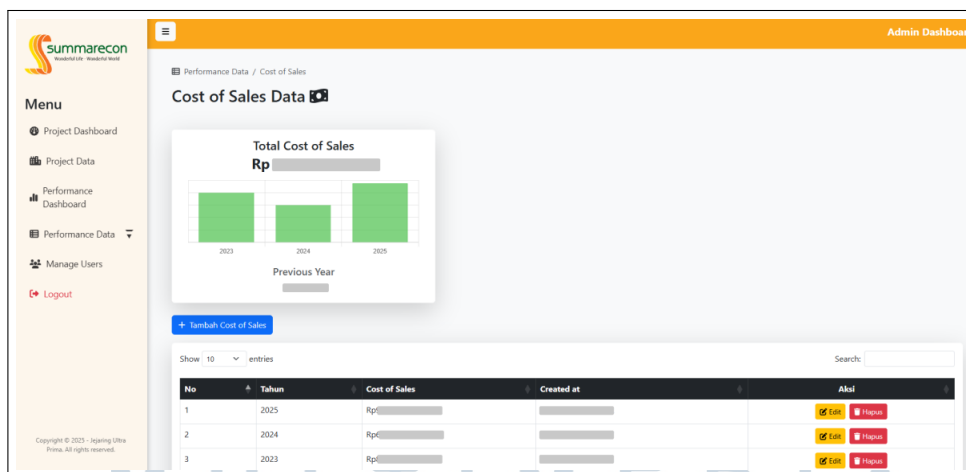


Gambar 3.61. Tampilan Tambah & Edit Revenue

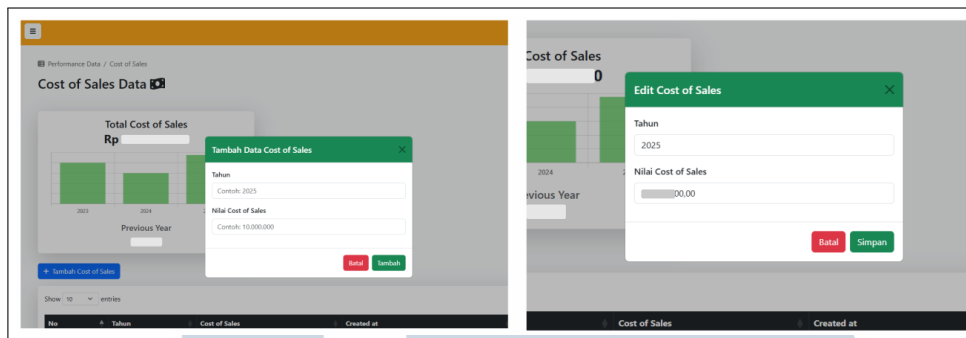
Gambar 3.60 menampilkan halaman pengelolaan data *revenue* yang mencakup fitur **CRUD** (*Create, Read, Update, Delete*). Pada halaman ini tersedia visualisasi dalam bentuk grafik *bar chart* serta *data table* yang menampilkan data *revenue*.

Pada Gambar 3.61 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data *revenue* melalui *modal form*. Admin dapat menginputkan tahun dan jumlah *revenue*, yang kemudian secara otomatis akan diperbarui pada tampilan visualisasi grafik *bar chart*.

- **Tampilan Data Cost of Sales**



Gambar 3.62. Tampilan Performance Data Revenue

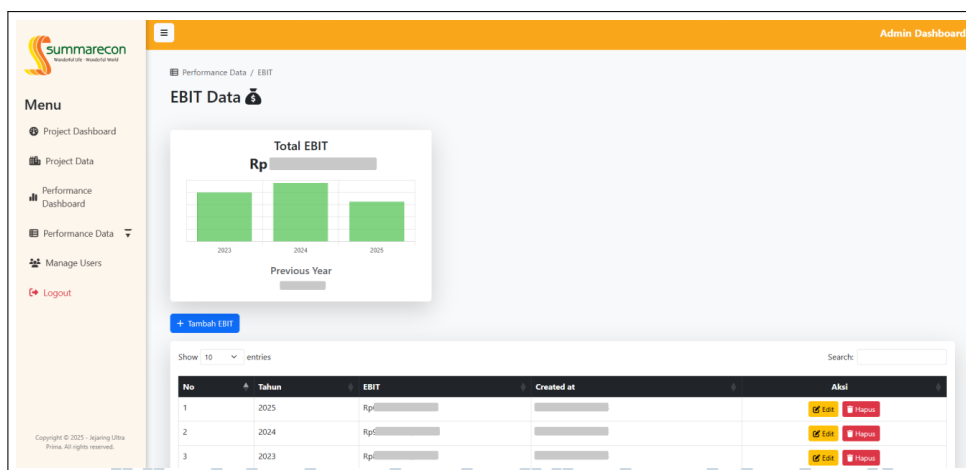


Gambar 3.63. Tampilan Tambah & Edit Cost of Sales

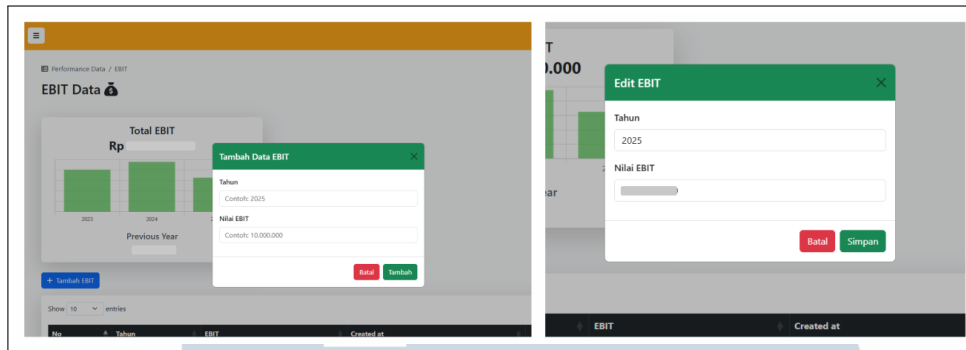
Gambar 3.62 menampilkan halaman pengelolaan data *cost of sales* yang mencakup fitur **CRUD** (*Create, Read, Update, Delete*). Pada halaman ini tersedia visualisasi dalam bentuk grafik *bar chart* serta *data table* yang menampilkan data *cost of sales*.

Pada Gambar 3.63 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data *cost of sales* melalui *modal form*. Admin dapat menginputkan tahun dan jumlah *revenue*, yang kemudian secara otomatis akan diperbarui pada tampilan visualisasi grafik *bar chart*.

- **Tampilan Data EBIT**



Gambar 3.64. Tampilan Performance Data EBIT

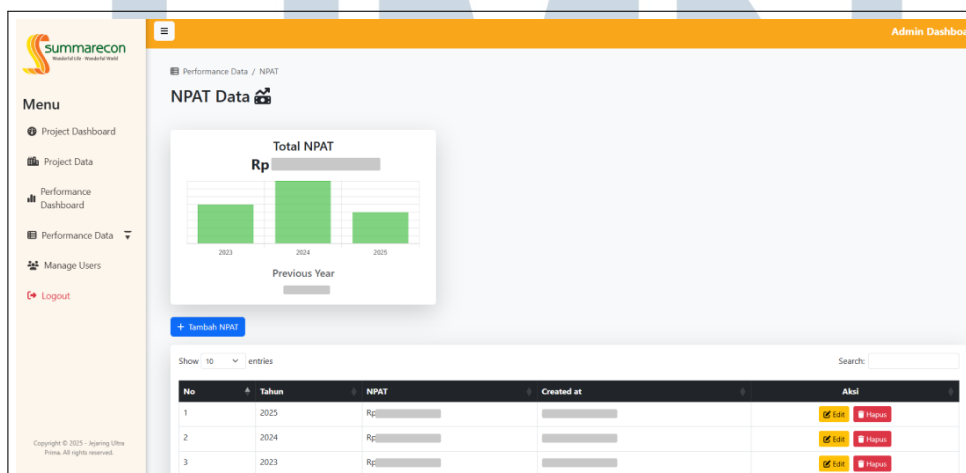


Gambar 3.65. Tampilan Tambah & Edit EBIT

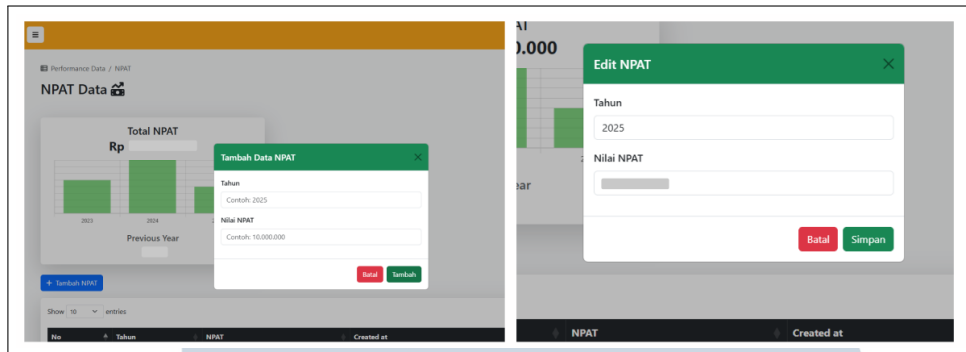
Gambar 3.64 menampilkan halaman pengelolaan data EBIT yang mencakup fitur **CRUD (Create, Read, Update, Delete)**. Pada halaman ini tersedia visualisasi dalam bentuk grafik *bar chart* serta *data table* yang menampilkan data EBIT.

Pada Gambar 3.65 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data EBIT melalui *modal form*. Admin dapat menginputkan tahun dan jumlah EBIT, yang kemudian secara otomatis akan diperbarui pada tampilan visualisasi grafik *bar chart*. akan diperbarui pada tampilan visualisasi grafik *bar chart*.

- **Tampilan Data NPAT**



Gambar 3.66. Tampilan Performance Data NPAT



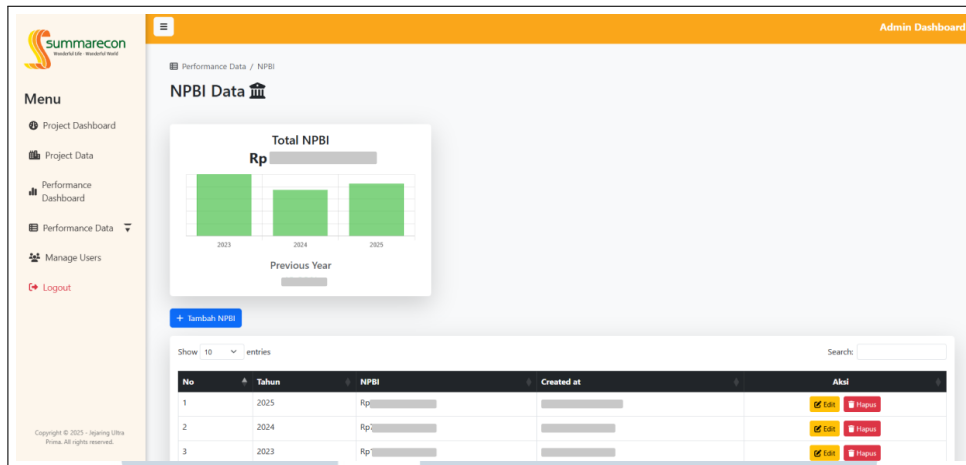
Gambar 3.67. Tampilan Tambah & Edit NPAT

Gambar 3.66 menampilkan halaman pengelolaan data NPAT yang mencakup fitur **CRUD (Create, Read, Update, Delete)**. Pada halaman ini tersedia visualisasi dalam bentuk grafik *bar chart* serta *data table* yang menampilkan data NPAT.

Pada Gambar 3.67 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data NPAT melalui *modal form*. Admin dapat menginputkan tahun dan jumlah NPAT, yang kemudian secara otomatis akan diperbarui pada tampilan visualisasi grafik *bar chart*. akan diperbarui pada tampilan visualisasi grafik *bar chart*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

- **Tampilan Data NPBI**



Gambar 3.68. Tampilan Performance Data NPBI

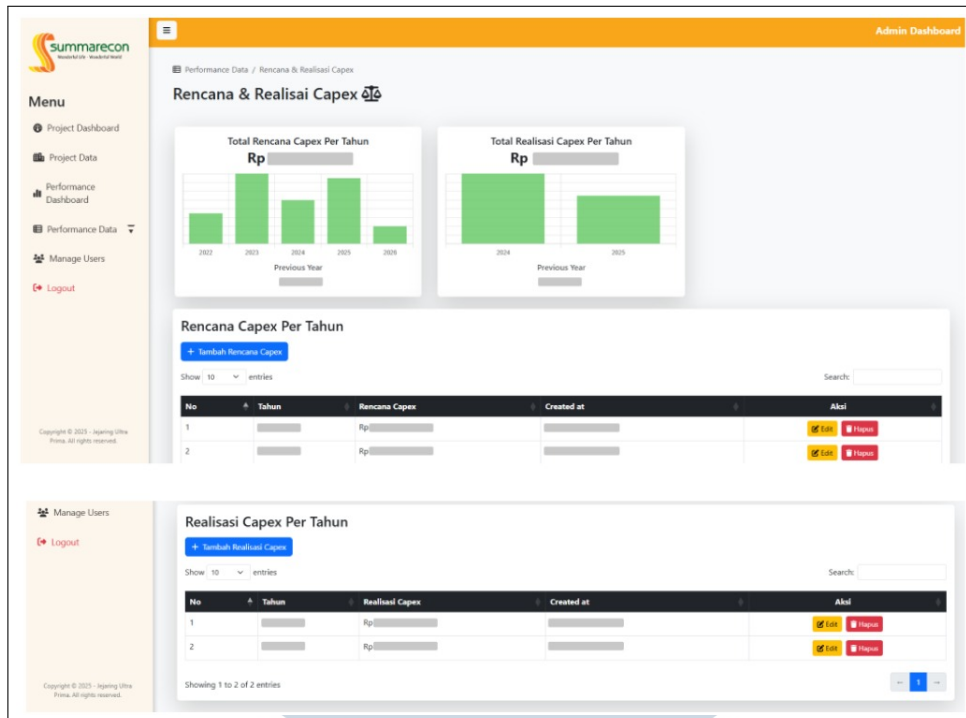
The screenshot shows two modal forms for managing NPBI data. The 'Tambah NPBI' modal form has fields for 'Tahun' (Year) and 'Nilai NPBI' (NPBI Value). The 'Edit NPBI' modal form has fields for 'Tahun' (Year) and 'Nilai NPBI' (NPBI Value). Both forms have 'Batal' (Cancel) and 'Simpan' (Save) buttons.

Gambar 3.69. Tampilan Tambah & Edit NPBI

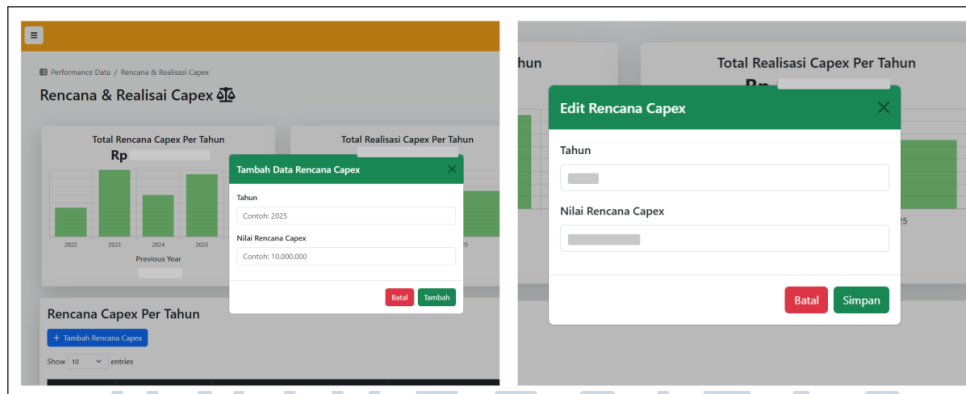
Gambar 3.68 menampilkan halaman pengelolaan data NPBI yang mencakup fitur **CRUD (Create, Read, Update, Delete)**. Pada halaman ini tersedia visualisasi dalam bentuk grafik *bar chart* serta *data table* yang menampilkan data NPBI.

Pada Gambar 3.69 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data NPBI melalui *modal form*. Admin dapat menginputkan tahun dan jumlah NPBI, yang kemudian secara otomatis akan diperbarui pada tampilan visualisasi grafik *bar chart*.

- Tampilan Data Rencana & Realisasi Capex



Gambar 3.70. Tampilan Performance Data Rencana dan Realisasi Capex

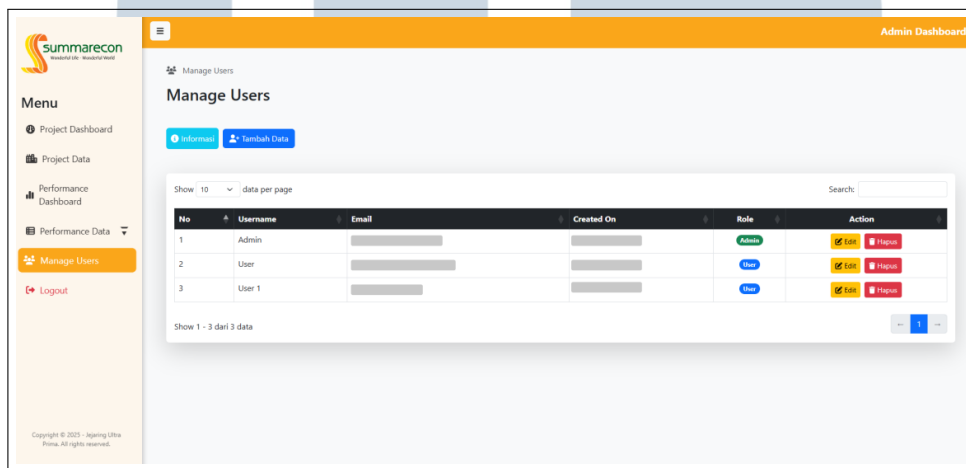


Gambar 3.71. Tampilan Tambah, Edit Rencana dan Realisasi Capex

Gambar 3.70 menampilkan halaman pengelolaan data Rencana dan Realisasi Capex yang dilengkapi dengan fitur **CRUD** (*Create, Read, Update, Delete*). Pada halaman ini, data divisualisasikan dalam bentuk *bar chart* dan *data table* yang memisahkan informasi rencana dan realisasi capex ke dalam 2 tabel berbeda.

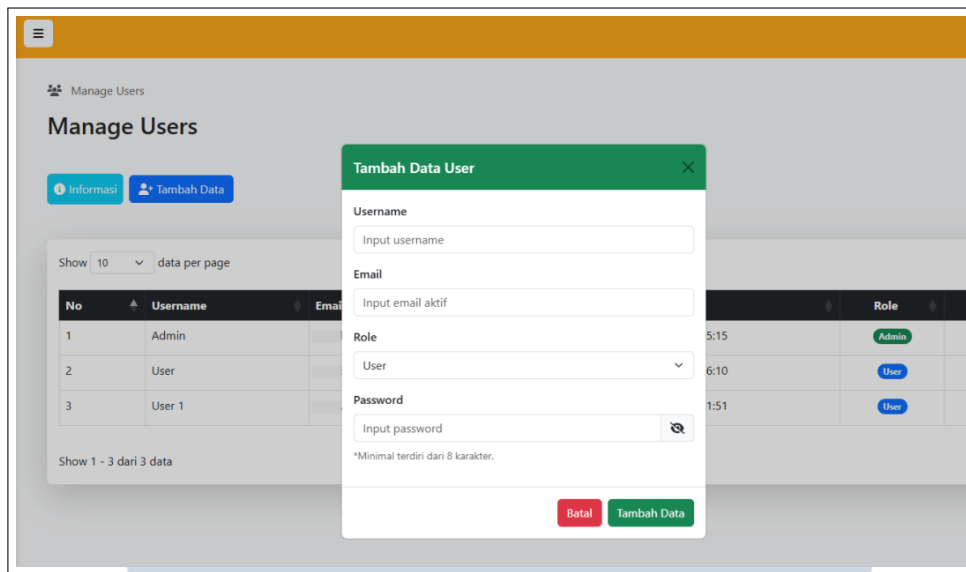
Selanjutnya, Gambar 3.71 memperlihatkan tampilan saat admin melakukan penambahan atau pengeditan data rencana dan realisasi capex melalui *modal form*. Form untuk rencana dan realisasi *capex* memiliki tampilan yang sama. Admin dapat memasukkan tahun dan jumlah capex, yang kemudian akan secara otomatis diperbarui pada tampilan *bar chart* sebagai tampilan visual dari data tersebut.

D.6 Tampilan Manage Users



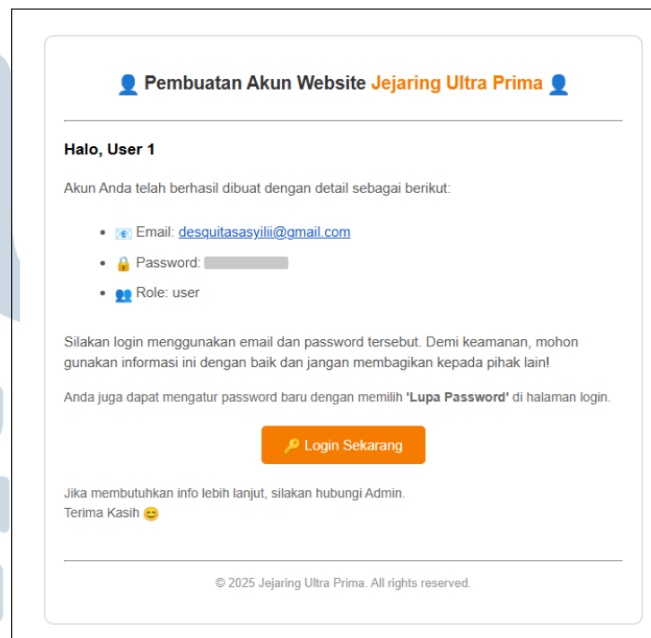
Gambar 3.72. Tampilan Manage Users

Gambar 3.72 menunjukkan tampilan halaman awal *Manage Users*, yang digunakan untuk mengelola data pengguna. Halaman ini dilengkapi dengan datatable yang menampilkan informasi seperti *username*, *email*, tanggal pembuatan akun (*created on*), dan peran pengguna (*role*), serta fitur operasi CRUD. Pada halaman ini, admin dapat membuat akun baru untuk pengguna yang akan mengakses sistem. Setelah akun berhasil dibuat, informasi akun akan secara otomatis dikirimkan ke email yang didaftarkan.



Gambar 3.73. Tampilan Tambah Users

Gambar 3.73 memperlihatkan tampilan saat admin akan menambahkan akun pengguna baru. Akan muncul *modal form* yang meminta input berupa *username*, *email*, peran pengguna (Admin atau User), dan password. Sistem akan melakukan validasi untuk memastikan bahwa *email* yang didaftarkan belum pernah digunakan sebelumnya.



Gambar 3.74. Tampilan Pesan Email Pembuatan Akun

Gambar 3.74 menampilkan contoh pesan *email* yang dikirimkan setelah

akun berhasil dibuat. Pesan ini berisi informasi *email* pengguna, *password* awal, dan peran pengguna. Setelah menerima email ini, pengguna dapat langsung masuk ke dalam sistem menggunakan akun tersebut, dan melakukan pengaturan ulang password melalui halaman *login* menggunakan fitur ”Reset Password”.

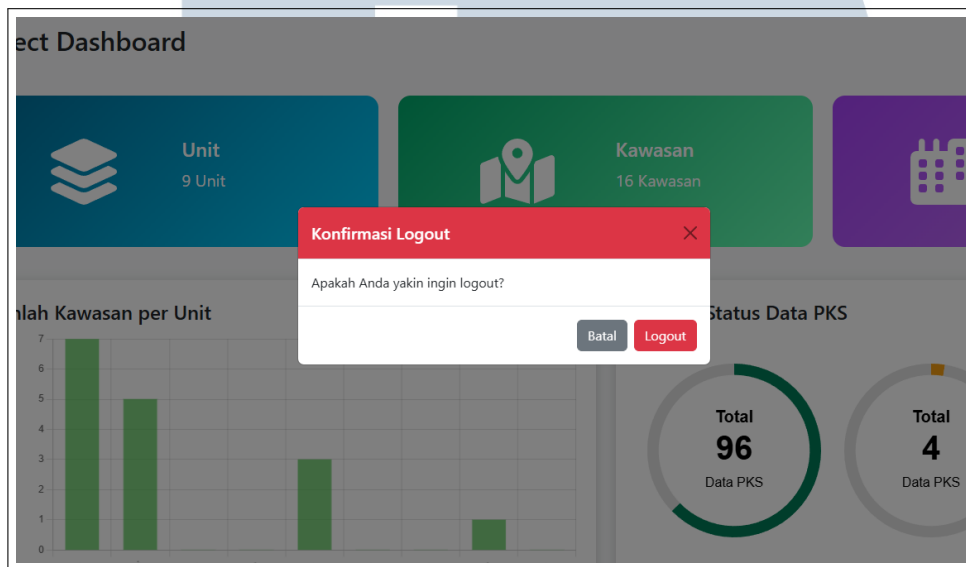
Gambar 3.75. Tampilan Edit Users

Gambar 3.75 menunjukkan tampilan saat admin melakukan perubahan informasi pada akun pengguna. Admin dapat mengubah data seperti *username*, *email*, peran, dan *password*. Semua data yang dimasukkan akan divalidasi terlebih dahulu. Misalnya, sistem akan memeriksa apakah *email* sudah terdaftar, serta memastikan bahwa *password* baru memiliki panjang minimal 8 karakter. Jika perubahan berhasil dilakukan, sistem secara otomatis mengirimkan notifikasi perubahan melalui email kepada pengguna terkait, seperti ditunjukkan pada Gambar 3.76.

Gambar 3.76. Tampilan Pesan Email Perubahan Akun

Gambar 3.76 menampilkan isi *email* notifikasi apabila terjadi perubahan data akun yang dilakukan oleh admin. *Email* ini mencantumkan informasi perubahan yang terjadi secara rinci, agar pengguna dapat mengetahui dan memastikan keamanannya.

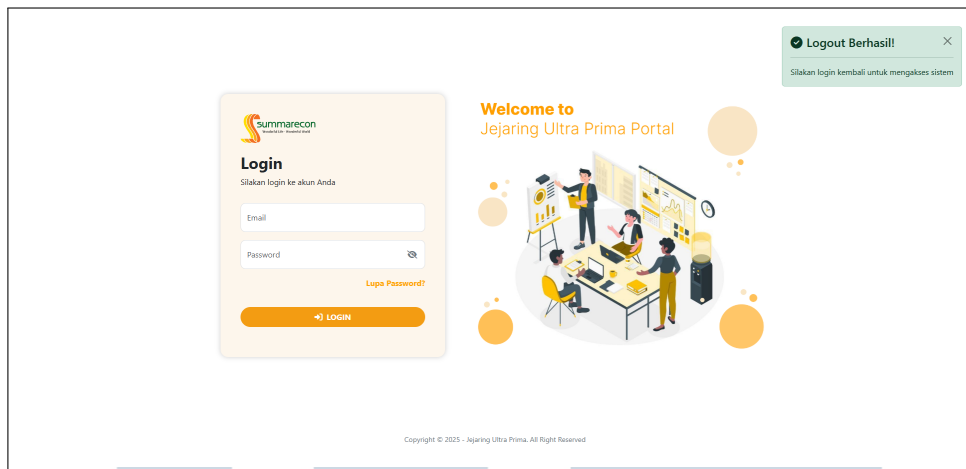
E Tampilan Logout



Gambar 3.77. Tampilan Konfirmasi Logout

Gambar 3.77 menampilkan tampilan saat pengguna akan melakukan proses *logout* dari sistem. Pada tahap ini, pengguna ditampilkan sebuah *modal* konfirmasi yang menanyakan apakah pengguna yakin ingin keluar dari sistem. Setelah pengguna mengonfirmasi pilihan tersebut, sistem akan secara otomatis melakukan *logout* dan mengarahkan pengguna kembali ke halaman *login*, ditunjukkan pada Gambar 3.78 berikut:

UNIVERSITAS
MULTIMEDIA
NUSANTARA



Gambar 3.78. Tampilan Setelah Logout

Gambar 3.78 memperlihatkan tampilan setelah proses *logout* berhasil dilakukan. Pada bagian pojok kanan atas layar, sistem menampilkan notifikasi *alert* berwarna hijau dengan pesan **”Logout Berhasil! Silakan login kembali untuk mengakses sistem.”** Notifikasi ini berfungsi sebagai *feed back* kepada pengguna bahwa proses *logout* telah sukses.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Selama menjalani kerja magang, terdapat beberapa kendala yang muncul dalam proses perancangan dan pengembangan *website*, antara lain:

1. Ketidakjelasan *requirement*, yaitu kebutuhan sistem yang diberikan terkadang kurang terperinci, seperti data apa saja yang perlu ditampilkan serta alur sistem (*flow*) yang sering mengalami perubahan selama pengembangan berlangsung.
2. Kompleksitas struktur sistem jaringan yang digunakan pada perusahaan, sehingga proses integrasi sistem manajemen informasi dengan data infrastruktur jaringan membutuhkan waktu dan penyesuaian lebih lanjut, terutama dalam hal penyesuaian skema data dan relasi.

3.4.2 Solusi

Untuk mengatasi kendala-kendala tersebut, solusi yang diterapkan selama pelaksanaan proyek magang antara lain:

1. Melakukan diskusi dan rapat secara berkala bersama pihak terkait di perusahaan untuk memperoleh kejelasan *requirement*, sekaligus menyesuaikan pengembangan sistem berdasarkan masukan yang diterima agar sesuai dengan kebutuhan.
2. Membuat rancangan awal sistem dan struktur *database* yang fleksibel, serta mengadaptasi pengembangan secara bertahap mengikuti kebutuhan infrastruktur jaringan yang kompleks.