

BAB III

PELAKSANAAN KERJA PROFESI

3.1 Bidang Kerja

Pada saat pelaksanaan Kerja Profesi (KP) di MP Project, praktikan menempati posisi sebagai seorang *Backend Developer*. Bidang kerja ini memfokuskan pada merancang dan membangun modul backend dari aplikasi *content management system* (CMS) sistem pemasaran perumahan Alpen Hills. Sebagai seorang *Backend Developer*, praktikan bertanggung jawab untuk mengembangkan berbagai fitur, mengatur manajemen data, mengintegrasikan sistem, serta memastikan kinerja yang optimal dan keamanan website.

3.2 Pelaksanaan Kerja

Sebagai seorang *backend developer* pada perusahaan MP Project, praktikan mengikuti serangkaian langkah dan tugas yang berfokus pada pengembangan fitur CMS website Perumahan Alpen Hills. Dalam melaksanakan tugas-tugas yang telah diberikan, praktikan menerapkan metode *System Development Life Cycle* (SDLC) untuk memastikan bahwa pengembangan berjalan dengan terstruktur dan sesuai dengan rencana. *System Development Life Cycle* (SDLC) adalah pendekatan terstruktur yang digunakan dalam pengembangan sistem, dan ini mencakup serangkaian tahapan yang harus diikuti dari awal hingga akhir proyek (Valacich & George, 2017). Dalam banyak proyek, tahapan dan langkah-langkah Siklus *System Development Life Cycle* (SDLC) mengikuti urutan logis yang dimulai dari awal dan berakhir pada tahap akhir. Namun, dalam proyek tertentu, tim proyek dapat menjalankan tahapan-tahapan tersebut secara berurutan, dengan pendekatan bertahap, secara iteratif, atau dalam pola lain yang berbeda. Dalam metode SDLC terdapat proses atau langkah-langkah yang harus dilakukan yaitu *Planning* (Perencanaan), *Analysis* (Analisis), *Design* (Desain), dan *Implementation* (Implementasi) (Dennis, Wixom, & Tegarden, 2015).

Berikut merupakan penjelasan dari setiap proses SDLC menurut (Dennis 2015) :

1. *Planning* (Perencanaan)

Planning adalah tahap pertama dalam memahami proyek sistem informasi yang akan dikembangkan serta bagaimana tim proyek akan menjalankannya. Terdapat dua tahap kunci:

1. Saat proyek dimulai, dilakukan identifikasi terhadap nilai bisnis sistem bagi organisasi. Langkah ini mencakup pemahaman tentang bagaimana sistem tersebut dapat mengurangi biaya atau meningkatkan pendapatan. Departemen TI bekerja sama dengan *stackholder* yang mengusulkan sistem untuk melakukan analisis kelayakan.
2. Setelah mendapatkan persetujuan, proyek memasuki tahap manajemen di mana manajer proyek merancang rencana kerja, menyusun tim proyek, dan menggunakan berbagai teknik manajemen proyek untuk mengawasi serta mengarahkan semua tahapan dalam Siklus Hidup Pengembangan Sistem (SDLC).

2. *Analysis* (Analisis)

Dalam fase analisis, tim proyek menjawab pertanyaan terkait pengguna sistem, fungsi sistem, serta lokasi dan jadwal penggunaan sistem. Selama tahap ini, tim proyek melakukan investigasi terhadap sistem yang sudah ada, mengidentifikasi potensi perbaikan, dan merumuskan konsep untuk sistem baru.

Fase ini terdiri dari tiga langkah:

1. Fase pertama dimulai dengan pengembangan strategi analisis yang menjadi pedoman bagi tim proyek. Strategi ini umumnya mencakup evaluasi sistem yang sedang berjalan, mengidentifikasi masalah yang ada, dan merancang pendekatan untuk sistem baru.
2. Langkah berikutnya melibatkan pengumpulan persyaratan, seringkali melalui wawancara atau kuesioner. Informasi yang dianalisis dari sumber-sumber ini, bersama dengan

masukan dari sponsor proyek dan pihak terkait lainnya, membantu dalam pengembangan konsep untuk sistem baru. Konsep ini menjadi dasar bagi pembuatan beberapa model analisis bisnis, yang mengilustrasikan bagaimana operasional bisnis akan terpengaruh jika sistem baru diterapkan.

3. Langkah selanjutnya adalah menggabungkan hasil analisis, konsep sistem, dan model-model dalam sebuah dokumen yang dikenal sebagai proposal sistem. Proposal ini kemudian diserahkan kepada sponsor proyek dan pemangku kepentingan utama lainnya, seperti anggota komite persetujuan, untuk menentukan kelanjutan proyek.

3. *Design* (Desain)

Fase desain memiliki tujuan untuk memberikan detail yang cermat mengenai fungsi sistem, termasuk aspek-aspek seperti perangkat keras, perangkat lunak, struktur jaringan, antarmuka pengguna, formulir, laporan, serta program dan basis data yang diperlukan. Meskipun keputusan-keputusan penting tentang sistem telah diambil selama tahap analisis, fase desain ini secara spesifik menjelaskan bagaimana sistem akan beroperasi. Terdapat empat tahapan dalam fase desain, yaitu pembentukan strategi desain, pembuatan kerangka dasar sistem, penentuan spesifikasi basis data dan file, serta perancangan program. Hasil dari tahapan ini dikenal sebagai spesifikasi sistem, yang akan diimplementasikan oleh tim pengembangan perangkat lunak. Sebelum melanjutkan, analisis kelayakan proyek dan rencana proyek diperbarui, dan keputusan kembali diambil mengenai kelanjutan proyek.

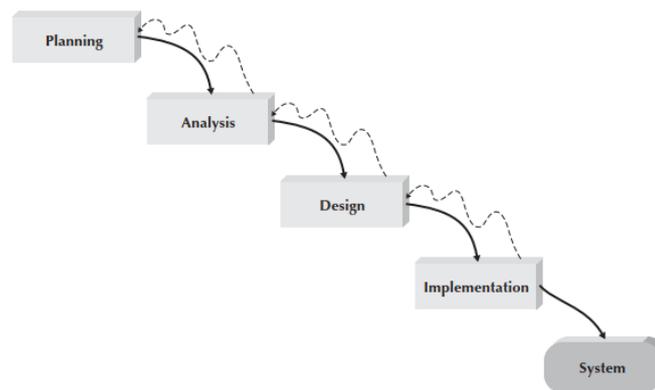
4. *Implementation* (Implementasi)

Fase akhir dalam *System Development Life Cycle* adalah fase implementasi. Tahap ini umumnya menjadi fokus utama, karena dalam sebagian besar kasus sistem, tahap ini merupakan bagian yang memakan waktu dan biaya paling besar dalam proses pengembangan. Fase ini terdiri dari tiga langkah:

1. Tahap pertama adalah pembangunan sistem di mana sistem dibuat dan diuji untuk memverifikasi kesesuaiannya dengan desain. Mengingat biaya dari kesalahan bisa sangat besar, pengujian merupakan tahapan yang sangat penting dalam proses implementasi. Sebagian besar organisasi mengalokasikan lebih banyak waktu dan perhatian pada pengujian daripada pada penulisan kode program.
2. Penginstalan sistem dilakukan dengan cara menonaktifkan sistem yang sudah ada dan mengaktifkan sistem baru. Salah satu aspek yang sangat penting dalam proses konversi ini adalah menyusun rencana pelatihan untuk mengedukasi pengguna tentang cara menggunakan sistem baru dan membantu mereka menyesuaikan diri dengan perubahan yang dibawa oleh sistem baru.
3. Tim analis membuat suatu rencana dukungan untuk sistem. Rencana ini biasanya mencakup evaluasi setelah implementasi, baik dalam bentuk formal maupun informal, serta metode sistematis untuk mengenali perubahan penting maupun yang kecil yang dibutuhkan oleh sistem.

Untuk menerapkan tahapan yang ada di SDLC tersebut praktikan menggunakan metodologi pengembangan *Waterfall*. Dua keuntungan utama dari pendekatan model *Waterfall* adalah metode ini terdapat proses untuk mengidentifikasi kebutuhan sistem sebelum implementasi dimulai. Hal tersebut dilakukan untuk meminimalisir perubahan kebutuhan proyek pada saat pengerjaan proyek berlangsung. Tetapi terdapat 2 (dua) kerugian pada metode ini yaitu dimana proses desain harus dijelaskan secara menyeluruh sebelum implementasi kode program dimulai dan hal tersebut memakan banyak waktu untuk penyelesaian proposal sistem. Metode *Waterfall* merupakan pendekatan untuk mengembangkan sistem atau perangkat lunak yang berfokus pada urutan tahapan yang linear dan terstruktur. Dalam metode ini, proyek perangkat lunak bergerak maju dari

satu fase ke fase berikutnya dengan cara yang serupa dengan aliran air terjun. Setiap fase memiliki dokumen hasil pekerjaan yang panjang dan komprehensif yang diajukan kepada sponsor proyek untuk persetujuan sebelum beralih ke fase berikutnya (Dennis, Wixom, & Tegarden, 2015). Pada gambar 3.1 merupakan visualisasi bagaimana metode *Waterfall* bekerja.



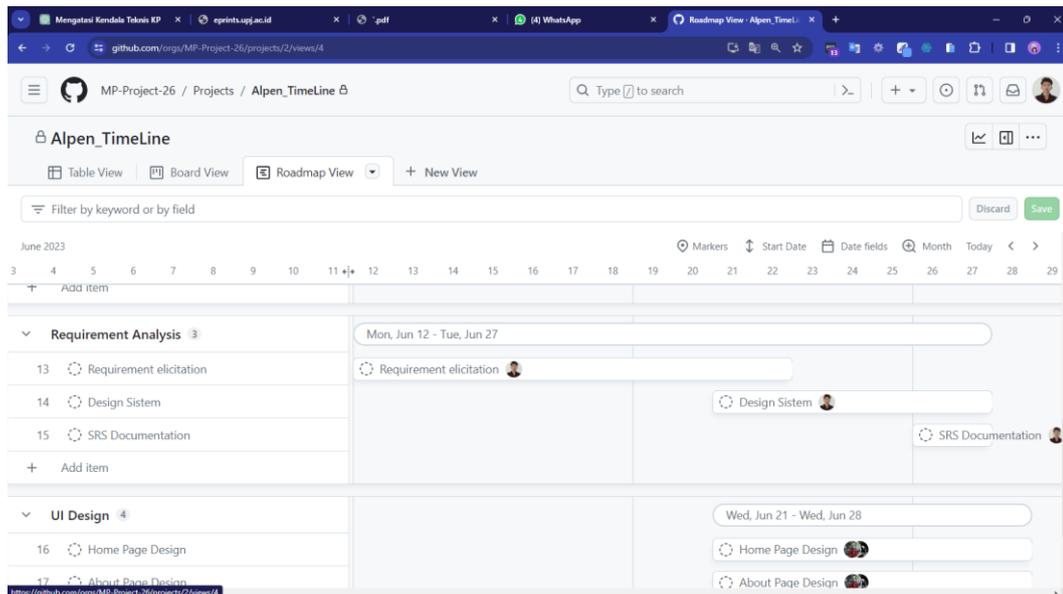
Gambar 3.1 Metodologi Waterfall

Sumber : (Dennis, Wixom, & Tegarden, 2015)

Pada bab berikutnya akan dijelaskan bagaimana praktikan dalam menerapkan tahapan-tahapan yang terdapat pada metodologi *Waterfall*.

3.2.1 Perencanaan (Planning)

Dalam tahap perencanaan (Planning) pelaksanaan Kerja Profesi (KP) di MP Project, praktikan diberikan tugas pada penyusunan timeline proyek dan rencana pengembangan. Praktikan merancang timeline proyek yang mencakup sejumlah tahapan dan tenggat waktu yang harus dipatuhi selama pengembangan backend website Perumahan Alpen Hills. Timeline proyek ini membantu memastikan bahwa setiap tugas dan tahapan pengembangan memiliki waktu yang telah ditentukan, sehingga proyek dapat berjalan sesuai rencana. Gambar 3.2 adalah visualisasi dari timeline proyek yang telah dirancang praktikan untuk menggambarkan urutan tahapan dan waktu yang diperlukan dalam pengembangan backend website Perumahan Alpen Hills.



Gambar 3.2 Time Line GitHub Projects

Dalam tahap perencanaan (Planning) pelaksanaan Kerja Profesi (KP) di MP Project, praktikan memanfaatkan fitur di GitHub Projects untuk membuat project timeline. Fitur ini memungkinkan praktikan untuk merencanakan dan melacak proyek dengan lebih efisien. Dengan GitHub Projects, praktikan dapat:

1. Membuat Proyek

Praktikan membuat proyek baru sesuai dengan proyek yang sedang dikerjakan. Dalam proyek ini, praktikan dapat menambahkan daftar tugas, fitur, dan komponen yang perlu dikembangkan.

2. Menjadwalkan Tugas

Praktikan menggunakan fitur GitHub Projects untuk menjadwalkan tugas-tugas yang harus diselesaikan dalam proyek. Setiap tugas memiliki deskripsi, prioritas, dan tenggat waktu yang ditetapkan.

3. Mengorganisir Tugas

Praktikan dapat mengorganisir tugas-tugas dalam kolom atau kategori yang sesuai, seperti "To Do," "In Progress," "Done," atau sesuai dengan kebutuhan proyek.

4. Melacak Kemajuan

Fitur ini memungkinkan praktikan untuk melacak kemajuan setiap tugas dan proyek secara keseluruhan. Praktikan dapat melihat tugas-tugas

yang telah selesai, yang sedang dikerjakan, dan yang masih harus dikerjakan.

5. Kolaborasi Tim

GitHub Projects juga memungkinkan kolaborasi antara anggota tim. Praktikan dapat mengundang rekan kerja atau anggota tim lainnya untuk bergabung dalam proyek dan berkontribusi dalam pengembangan backend.

Dengan menggunakan fitur GitHub Projects, praktikan dapat memiliki visibilitas yang lebih baik terhadap seluruh proyek, mengatur waktu dan prioritas tugas, serta mengikuti jadwal yang telah ditetapkan. Ini membantu memastikan bahwa proyek berjalan sesuai rencana dan memenuhi tenggat waktu yang telah ditetapkan.

3.2.2 Analisis (Analysis)

Dalam tahap Analisis (Analysis) pelaksanaan Kerja Profesi (KP) di MP Project, praktikan bertanggung jawab untuk melakukan analisis kebutuhan fungsional dan non-fungsional dalam pengembangan aplikasi. Untuk mengumpulkan kebutuhan pengguna praktikan berdiskusi langsung dengan staf dan manajer pemasaran MP Project. Praktikan menggunakan metode elicitation (elisitasi) untuk mengumpulkan kebutuhan dari stackholder. Hasil dari tahapan elisitasi ini terdokumentasikan dapat dilihat pada tabel 3.1 yang mencantumkan daftar kebutuhan fungsional dan non-fungsional yang telah diidentifikasi selama proses analisis.

Tabel 3.1 Final Elisitasi *User Requirements*

Fungsional	
Analisis Kebutuhan Pengguna	
Pengguna ingin Sistem Dapat	
1.	Pengguna bisa masuk ke akun mereka dengan menggunakan email dan kata sandi.
2.	Pengguna dapat melakukan pengeditan akun mereka seperti ganti email dan kata sandi
3.	Hak akses pengguna harus sesuai dengan peran mereka
4.	Staf dapat mengelola konten, seperti mengedit entri blog atau mengelola konten

	galeri.
5.	Staf harus dapat mengelola data pemesanan, memperbarui status pesanan, atau memberikan informasi tambahan tentang pesanan kepada pengguna.
6.	Manajer Marketing harus memiliki akses yang lebih tinggi daripada staf, dan mereka dapat mengelola tugas-tugas pemasaran.
7.	Manajer Marketing dapat mengelola konten yang berkaitan dengan strategi pemasaran, termasuk pengeditan blog dan gambar dalam galeri yang relevan dengan kampanye pemasaran.
8.	Manajer Marketing dapat membuat, mengubah role, dan menghapus akun staf
Non Fungional.	
Pengguna Ingin Sistem Dapat	
1.	Informasi akun pengguna harus terlindungi dengan baik dan enkripsi kata sandi.
2.	Sistem harus mengelola dengan cermat hak akses berdasarkan peran pengguna. Manajer marketing, sebagai peran yang lebih tinggi, dan dapat mengakses sistem secara keseluruhan
3.	Informasi sensitif seperti kata sandi dan data pemesanan harus dienkripsi dan dilindungi dengan baik.
4.	Sistem harus merespons dengan cepat terhadap permintaan pengguna, terutama dalam manajemen konten, data pemesanan, dan otentikasi.
5.	Antarmuka pengguna harus merespons dengan baik terhadap berbagai perangkat, termasuk perangkat mobile dan desktop.

3.2.3 Desain (Design)

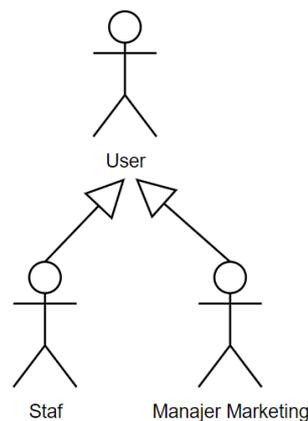
Dalam tahap Desain (Design) pelaksanaan Kerja Profesi (KP) di MP Project, praktikan membuat beberapa elemen desain sistem, termasuk Diagram Kasus Pengguna (Use Case Diagram), Diagram Aktifitas (Activity Diagram), dan rancangan desain database. Berikut merupakan hasil rancangan yang telah dibuat :

1. Diagram Kasus Pengguna (Use Case Diagram)

Diagram Kasus Pengguna atau *Use Case Diagram* adalah sebuah permodelan yang digunakan untuk menggambarkan bagaimana sistem bisnis berinteraksi dengan lingkungannya. Secara esensial, Use Case Diagram memberikan gambaran pada proses bisnis sistem informasi bisnis (Dennis, Wixom, & Tegarden, 2015).

Use Case Diagram yang telah dibuat praktikan mencakup berbagai aksi atau kasus penggunaan yang terlibat dalam sistem. Hal ini mencakup berbagai fungsi dan fitur yang akan diimplementasikan

dalam sistem, seperti pengelolaan konten, manajemen data pemesanan, manajemen akun staf, dan autentikasi. Dalam rancangan *use case diagram* yang dibuat oleh praktikan terdapat dua aktor yang merupakan dari aktor *user* dan dapat melakukan interaksi dengan sistem yaitu staf dan manajer marketing. Pada gambar 3.1 merupakan visualisasi dari aktor yang terdapat di *use case diagram*.



Gambar 3.3 *Generalisasi Aktor*

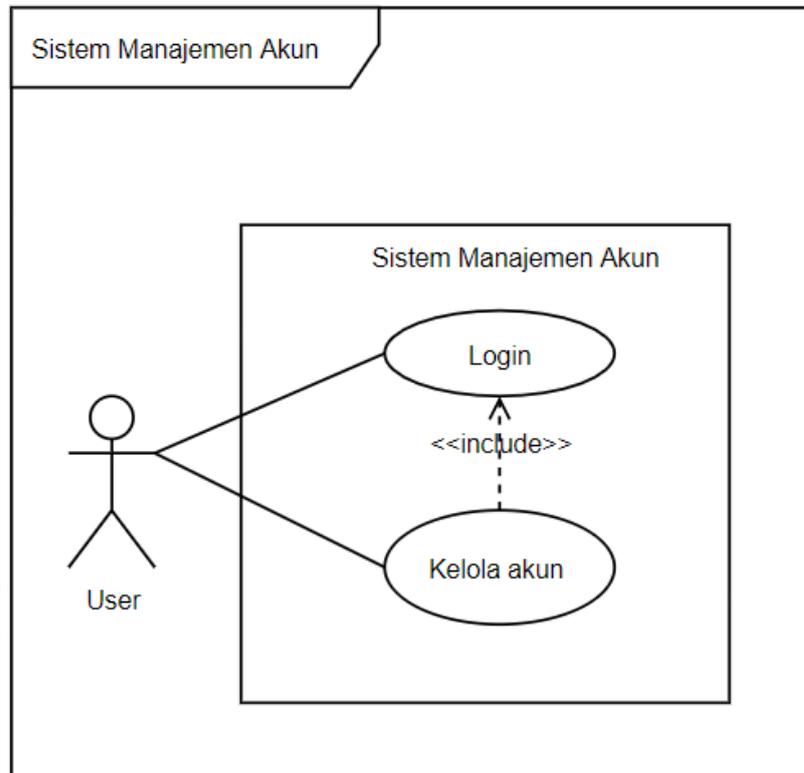
a. Staf

Staf adalah individu yang bekerja di dalam perusahaan MP Project dan memiliki akses ke sistem. Peran staf dalam sistem ini melibatkan beberapa aktivitas penting, termasuk pengelolaan konten, manajemen data pemesanan.

b. Manajer Marketing

Peran Manajer Marketing dalam sistem melibatkan aktivitas seperti mengelola kampanye pemasaran, mengidentifikasi peluang pasar, dan berinteraksi dengan staf dan calon konsumen.

Berikut adalah hasil dari *Use Case Diagram* sisi *Back-End* website perumahan Alpen Hills yang dapat dilihat pada gambar 3.4 dan gambar 3.5.



Gambar 3.4 Use Case Diagram Manajemen Akun

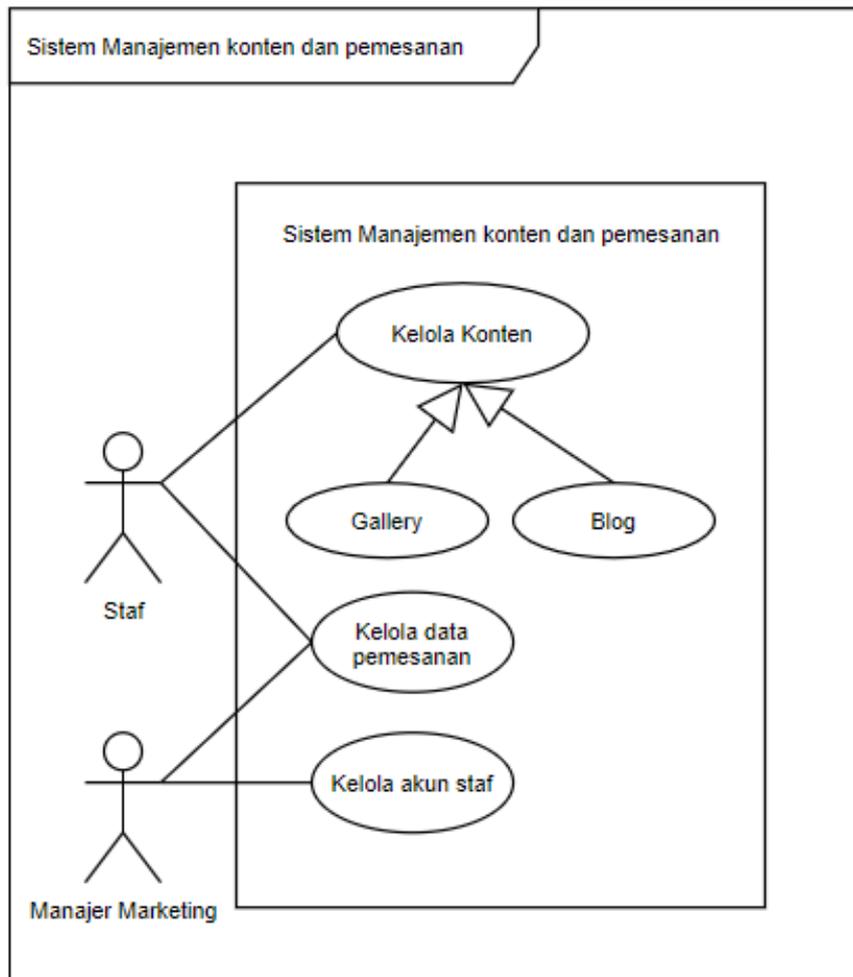
Berdasarkan gambar 3.4 diatas, terdapat beberapa fungsionalitas sistem yang dapat dilakukan oleh aktor user, diantaranya adalah :

a. Login

Use case login adalah langkah autentikasi yang memungkinkan *user* untuk mengakses akun mereka. Setelah login, *user* akan memiliki akses ke fitur-fitur yang sesuai dengan peran mereka dalam sistem.

b. Kelola Akun

User memiliki kemampuan untuk mengelola akun mereka dalam sistem. Hal ini mencakup tindakan seperti mengubah informasi profil, mengatur preferensi akun, dan mengganti kata sandi. *User* dapat mengelola akun mereka sesuai dengan kebutuhan mereka dalam konteks penggunaan sistem.



Gambar 3.5 Use Case Diagram Manajemen Konten dan Pemesanan

Berdasarkan gambar 3.5 diatas, terdapat 3 (tiga) use case yang terdapat pada sistem manajemen konten dan pemesanan, diantaranya adalah :

1. Kelola Konten

Kelola Konten merupakan *use case* yang lebih generik yang mencakup dua use case yang lebih spesifik, yaitu "Gallery" dan "Blog". *Use case* "Kelola Konten" berfungsi sebagai konsep tingkat tinggi yang mencakup pengelolaan konten secara keseluruhan di dalam sistem. Use case yang lebih spesifik, seperti "Gallery" dan "Blog," adalah sub-kasus

penggunaan yang menguraikan detail pengelolaan konten masing-masing.

2. Kelola data pemesanan

Use case ini mencakup kemampuan baik staf maupun manajer marketing untuk mengelola data pemesanan dalam sistem. Use case ini termasuk dalam pengelolaan rincian pemesanan, mengonfirmasi pemesanan yang masuk, dan memperbarui status pemesanan. Dalam prakteknya, use case ini memungkinkan aktor untuk melihat informasi pemesanan, menindaklanjuti dengan pelanggan, dan memastikan bahwa proses pemesanan berjalan dengan lancar.

3. Kelola akun staf

Use case ini adalah representasi dari kemampuan manajer marketing untuk mengelola akun staf yang memiliki akses ke sistem. Dalam kasus ini, "mengelola akun staf" mencakup tindakan seperti menambahkan anggota baru ke tim, menghapus anggota yang tidak aktif, atau memperbarui data akun staf yang ada. Manajer marketing bertanggung jawab untuk memastikan bahwa timnya memiliki akses yang sesuai ke sistem dan dapat berkontribusi dengan efektif dalam operasi perusahaan.

2. *Specification Use Case Diagram*

Specification Use Case Diagram digunakan untuk menjelaskan hasil dari *Use Case Diagram* agar dapat dipahami alur kinerja atau stepstep dari setiap use case yang ada. Berikut adalah hasil dari *Specification Use Case Diagram* sistem *backend* website perumahan Alpen Hills yang terdapat di tabel 3.2 sampai tabel 3.6 di bawah ini :

Tabel 3.2 Spesifikasi Kasus Pengguna Login

Use Case ID	UC01
Nama Use Case	Login
Aktor	User
Deskripsi	Use case ini memungkinkan pengguna untuk masuk ke dalam sistem dengan cara mengotentikasi diri menggunakan kredensial yang valid.

Prasyarat	Pengguna telah terdaftar dalam sistem dan memiliki kredensial login yang benar.	
Alur Utama	1. User membuka halaman login.	2. Menampilkan form login
	3. User memasukkan nama email dan kata sandi (password).	4. Sistem memverifikasi email dan password user beserta role user.
	5. User diarahkan ke halaman dashboard sesuai role.	
Alur Alternatif	<p>3a. Jika username atau password tidak valid, sistem memberikan respon: "email atau password tidak valid." Dan mengizinkan User untuk mencoba lagi.</p> <p>5a. Jika role user merupakan staf maka user akan diarahkan ke halaman dashboard staf</p> <p>5b. Jika role user merupakan Manajer Marketing user akan diarahkan ke halaman dashboard Manajer Marketing</p>	
Post-kondisi	User berhasil masuk ke dalam sistem dan memiliki akses ke fitur-fitur yang memerlukan otentikasi.	
Eksepsi	2a. Jika User tidak mengisi salah satu atau kedua bidang (email atau kata sandi), sistem memberikan respon: "Harap isi semua bidang" dan mengizinkan pengguna untuk mengisi kembali.	

Tabel 3.3 Spesifikasi Kasus Pengguna Kelola Akun

Use Case ID	UC02	
Nama Use Case	Kelola Akun	
Aktor	User	
Deskripsi	Use case ini memungkinkan User untuk mengelola akun pengguna mereka sendiri pada sistem.	
Prasyarat	Pengguna telah Login sistem	
Alur Utama	1. User memilih opsi "Kelola Akun" pada profil mereka.	2. Sistem menampilkan opsi-opsi yang tersedia: Mengubah Kata Sandi, Memperbarui Informasi Profil, Menonaktifkan Akun.
	3. User memilih salah satu opsi tindakan.	4. Sistem memproses tindakan yang dipilih.
Alur Alternatif	4a. Jika User membatalkan tindakan, sistem tidak mengubah atau menghapus akun mereka.	
Post-kondisi	Akun telah diperbarui sesuai dengan tindakan pengguna	
Eksepsi	-	

Tabel 3.4 Spesifikasi Kasus Pengguna Kelola Konten

Use Case ID	UC03	
Nama Use Case	Kelola Akun	
Aktor	Staf	
Deskripsi	Use case ini memungkinkan aktor untuk Mengelola konten galeri dan blog	
Prasyarat	Staf telah login ke dalam sistem	
Alur Utama	1. Staf memilih opsi menu "Kelola Galeri" atau "Kelola Blog	2. Sistem menampilkan daftar konten sesuai pilihan staf
	3. Staf dapat: a. Menambahkan konten baru. b. Memperbarui konten yang ada. c. Menghapus konten yang tidak diperlukan.	4. Sistem memproses tindakan staf
Alur Alternatif	4a. Jika staf membatalkan tindakan, sistem tidak membuat perubahan pada konten.	
Post-kondisi	Konten galeri atau blog telah ditambahkan, diperbarui, atau dihapus sesuai dengan tindakan staf.	
Eksepsi	-	

Tabel 3.5 Specification use case Kelola Data Pemesanan

Use Case ID	UC04	
Nama Use Case	Kelola Data Pemesanan	
Aktor	Staf, Manajer Marketing	
Deskripsi	Use case ini memungkinkan aktor untuk Mengelola data pemesanan	
Prasyarat	Aktor telah login ke dalam sistem	
Alur Utama	1. Aktor memilih opsi "Kelola Data Pemesanan.	2. Sistem menampilkan daftar pemesanan yang ada
	3. Aktor dapat: a. Menambahkan data pemesanan baru. b. Memperbarui data pemesanan yang ada. c. Menghapus data pemesanan yang tidak diperlukan.	4. Sistem memproses tindakan yang dilakukan oleh aktor
Alur Alternatif	4a. Jika aktor membatalkan tindakan, sistem tidak membuat perubahan pada data pemesanan.	
Post-kondisi	Data pemesanan telah ditambahkan, diperbarui, atau dihapus sesuai dengan tindakan aktor.	
Eksepsi	-	

Tabel 3.6 Spesifikasi Kasus Pengguna Kelola Akun Staf

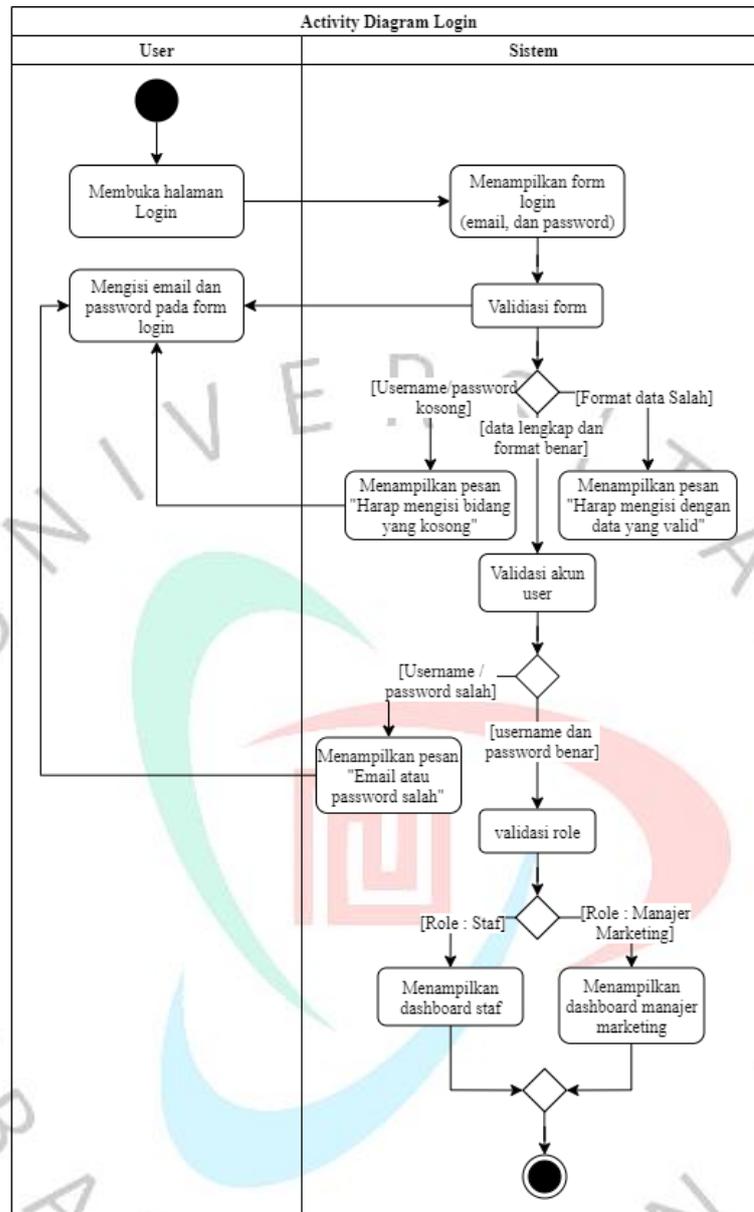
Use Case ID	UC05	
Nama Use Case	Kelola Akun Staf	
Aktor	Staf, Manajer Marketing	
Deskripsi	Use Case ini memungkinkan Manajer Marketing untuk mengelola akun staf yang bekerja di perusahaan MP Project. Manajer Marketing dapat menambahkan, memperbarui, atau menghapus akun staf sesuai dengan kebutuhan.	
Prasyarat	Aktor telah login ke dalam sistem	
Alur Utama	1. Manajer Marketing memilih opsi "Kelola Akun Staf."	2. Sistem menampilkan daftar akun staf yang ada.
	3. Manajer Marketing dapat melakukan tindakan berikut: a. Menambahkan akun staf baru. b. Memperbarui informasi akun staf yang ada. c. Menghapus akun staf yang tidak diperlukan.	4. Sistem memproses tindakan yang dilakukan oleh Manajer Marketing.
Alur Alternatif	4a. Jika aktor membatalkan tindakan, sistem tidak membuat perubahan pada data akun staf.	
Post-kondisi	Data akun staf telah ditambahkan, diperbarui, atau dihapus sesuai dengan tindakan aktor.	
Eksepsi	-	

3. Activity Diagram

Setelah kebutuhan pengguna telah dimodelkan ke dalam diagram *use case* praktikan kemudian merancang *activity diagram* untuk menggambarkan alur setiap proses pada sistem. Berikut merupakan hasil permodelan *activity diagram* berdasarkan *use case diagram* yang telah dirancang sesuai dengan persyaratan sistem:

a. Activity Diagram Login

Gambar 3.6 mengilustrasikan alur pada proses login, dimula saat seorang pengguna mencoba untuk masuk ke dalam sistem.

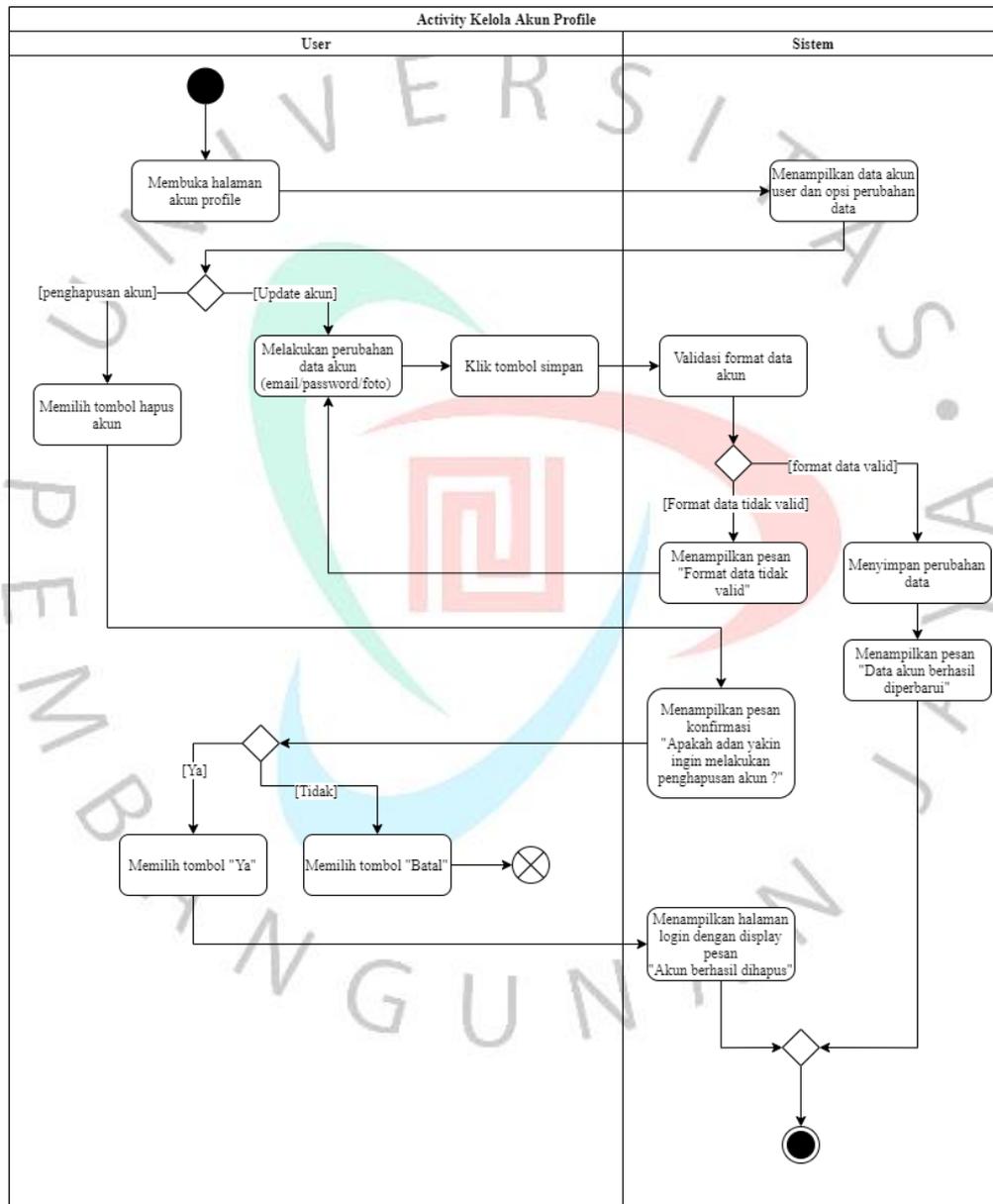


Gambar 3.6 Activity Login

Proses ini dimulai ketika pengguna memasukkan kredensial mereka, seperti email dan kata sandi, pada antarmuka login. Setelah itu, sistem melakukan validasi terhadap kredensial yang dimasukkan oleh pengguna. Selain itu, sistem juga melakukan verifikasi berdasarkan peran (role) pengguna untuk memastikan bahwa akses yang diberikan sesuai dengan peran yang dimiliki oleh pengguna. Fitur ini membantu memastikan bahwa setiap pengguna memiliki akses yang sesuai dan hak

akses yang telah ditetapkan dalam sistem sesuai dengan perannya. Dengan kata lain, Activity Diagram ini menggambarkan tahapan yang harus dilewati pengguna dan sistem untuk melakukan proses login dengan aman dan efisien.

b. *Activity Diagram* Kelola Akun Profile

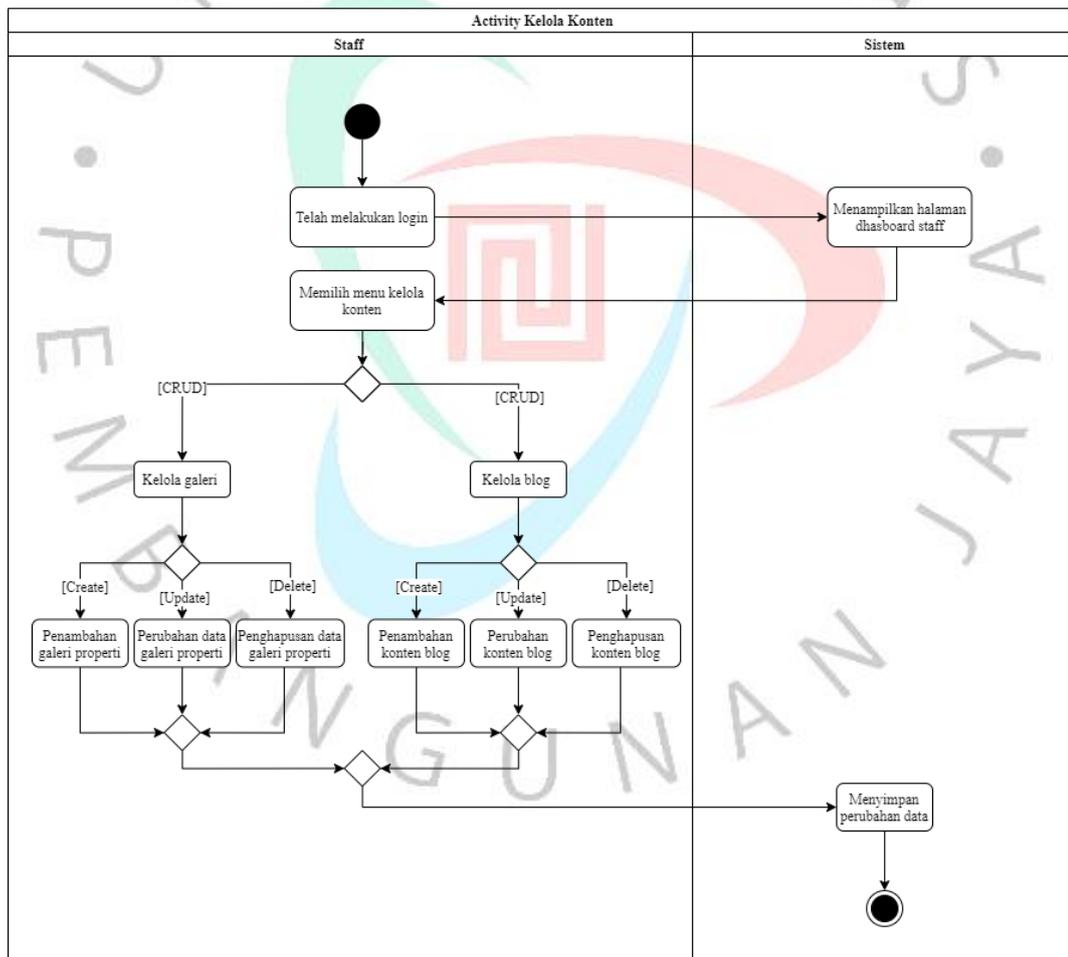


Gambar 3.7 Activity Diagram Kelola Akun Profile

Gambar 3.7 menunjukkan proses bagaimana pengguna melakukan proses "Kelola Akun Profile". *Activity Diagram* tersebut

menggambarkan serangkaian tindakan yang dapat dilakukan oleh pengguna pada akun mereka. Pengguna dapat memilih "Kelola Akun," melihat dan mengedit informasi akun, serta menyimpan perubahan yang diperlukan. Selain itu, mereka memiliki opsi untuk menghapus akun, yang memerlukan konfirmasi tambahan sebelum penghapusan. Diagram ini memberikan pandangan komprehensif tentang alur kerja interaktif antara pengguna dan sistem dalam mengelola akun pengguna, serta opsi untuk menghapus akun jika dibutuhkan.

c. *Activity Diagram Kelola Konten*

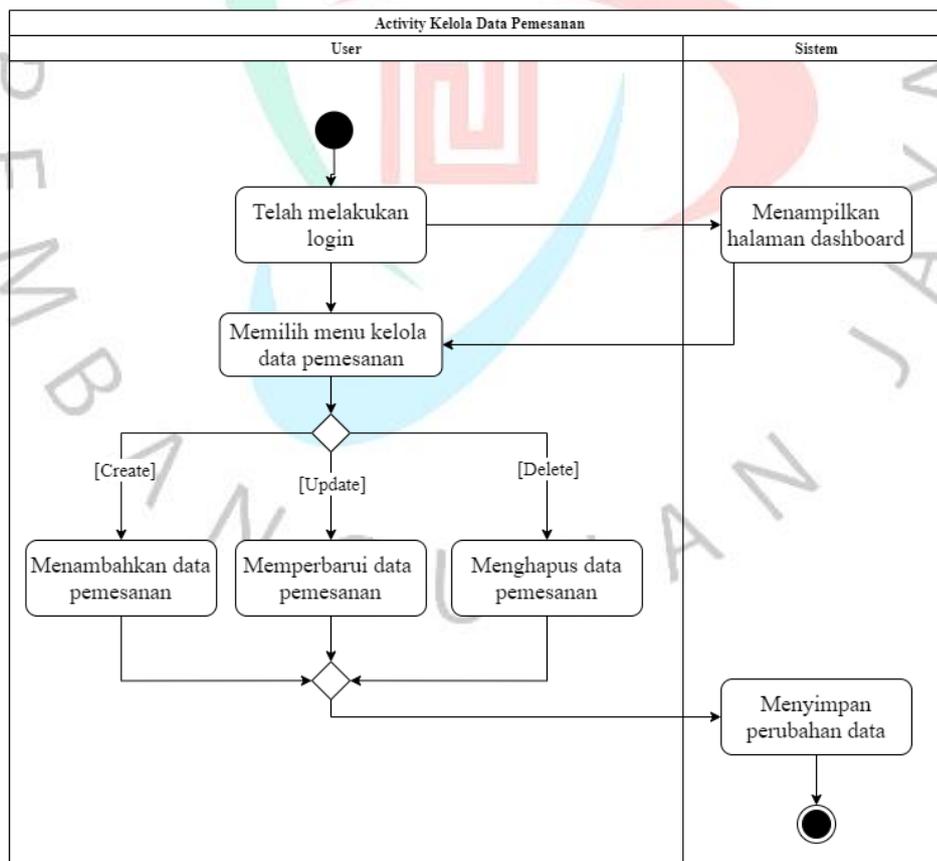


Gambar 3.8 *Activity Diagram Kelola Konten*

Activity Diagram proses Kelola Konten, yang terlihat pada Gambar 3.8, memvisualisasikan bagaimana pengelolaan konten di

situs web Perumahan Alpen Hills berlangsung. Kelola Konten adalah generalisasi dari dua fitur utama, yaitu "Galeri" dan "Blog." Dalam proses ini, Kelola Konten menggambarkan alur kerja umum yang berlaku baik untuk mengelola galeri gambar properti maupun untuk mengelola postingan blog. Alur dimulai ketika pengguna memilih opsi "Kelola Konten" di antarmuka pengguna. Setelah itu, sistem akan menampilkan daftar konten yang tersedia, baik itu galeri maupun blog. Pengguna memiliki opsi untuk menambah, mengedit, atau menghapus konten sesuai dengan jenisnya. Selanjutnya, sistem akan memungkinkan pengguna untuk melakukan tindakan yang sesuai, seperti mengunggah gambar untuk galeri atau menulis dan mempublikasikan postingan blog.

d. *Activity Diagram Kelola Data Pemesanan*



Gambar 3.9 *Activity Diagram Kelola Data Pemesanan*

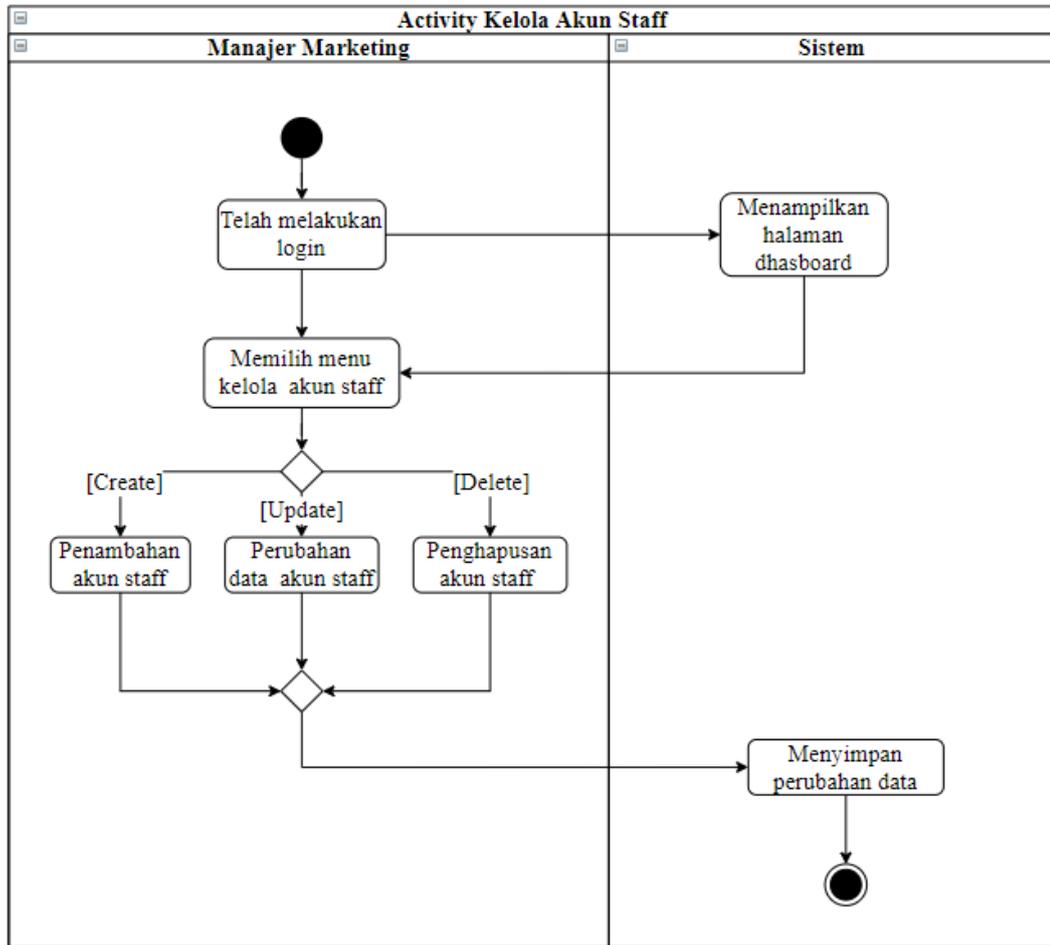
Activity Diagram proses Kelola Data Pemesanan, seperti yang ditunjukkan dalam Gambar 3.9, menggambarkan alur kerja yang terkait dengan pengelolaan data pemesanan dalam konteks situs web Perumahan Alpen Hills. Proses ini dimulai ketika pengguna memilih opsi "Kelola Data Pemesanan" di antarmuka pengguna. Setelah itu, sistem akan menampilkan daftar data pemesanan yang tersedia, seperti data pemesanan properti atau layanan yang terkait.

Pengguna memiliki beberapa opsi, termasuk menambahkan data pemesanan baru, mengedit data yang ada, atau menghapus data pemesanan. Selain itu, sistem memvalidasi data yang dimasukkan oleh pengguna untuk memastikan bahwa data yang dimasukkan sesuai dengan format yang benar dan memenuhi persyaratan. Setelah pengguna selesai mengelola data pemesanan, proses selesai, dan data pemesanan diperbarui sesuai dengan perubahan yang telah dilakukan.

Proses ini memungkinkan staf dan manajer marketing untuk mengelola data pemesanan properti dengan efisien, memastikan bahwa informasi yang terkait dengan pemesanan selalu akurat dan terkini. Hal ini juga membantu dalam mengoptimalkan layanan pelanggan dan manajemen properti Perumahan Alpen Hills secara keseluruhan.

e. *Activity Diagram* Kelola Akun Staff

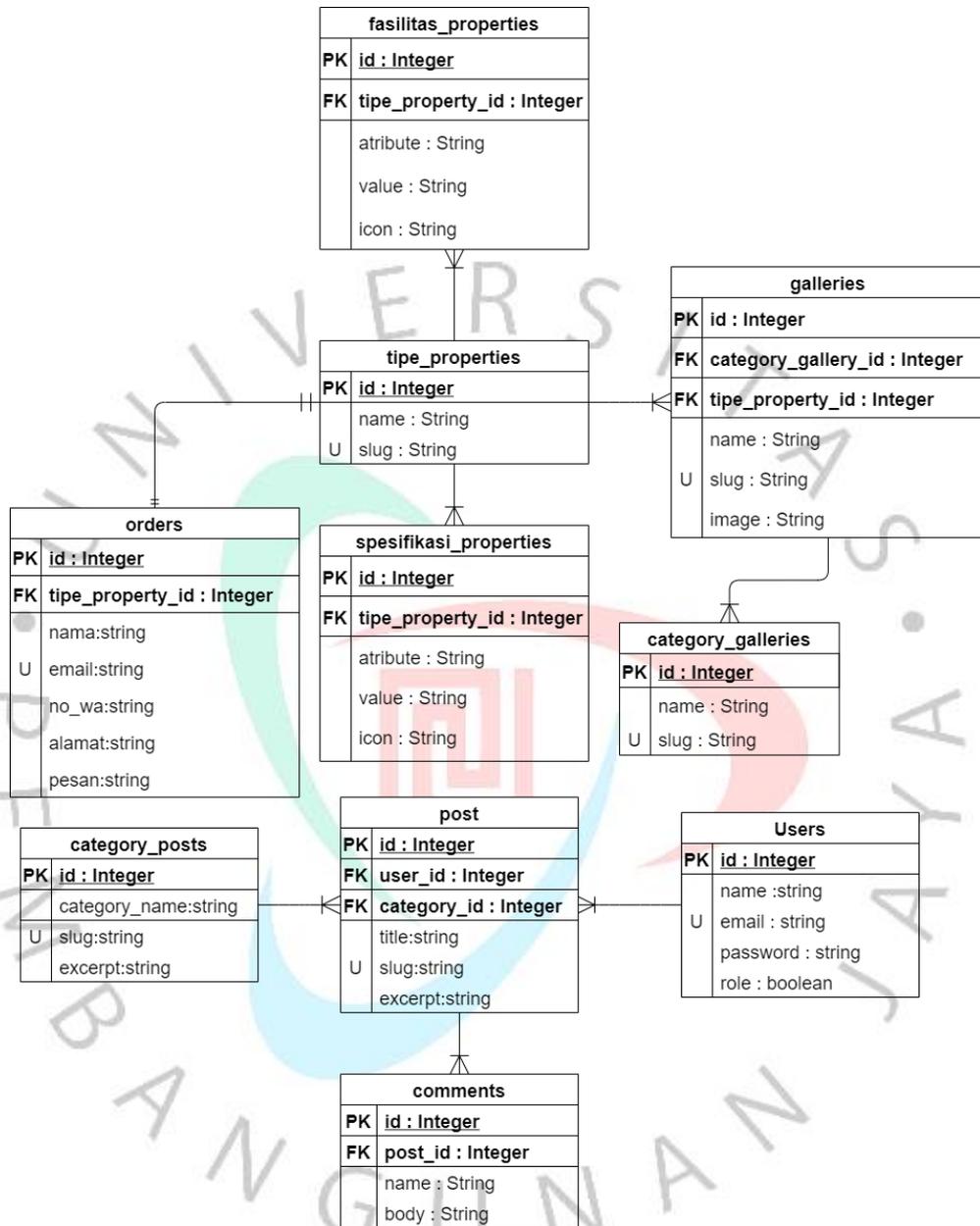
Activity Diagram proses Kelola Akun Staff, yang digambarkan dalam Gambar 3.10, mengilustrasikan alur kerja yang terkait dengan pengelolaan akun staf dalam situs web Perumahan Alpen Hills



Gambar 3.10 Ativity Diagram Kelola Akun Staff

Gambar 3.10 Ativity Diagram Kelola Akun Staff tersebut menggambarkan proses ketika seorang pengguna, yang memiliki peran sebagai administrator atau manajer, memilih opsi "Kelola Akun Staff" di antarmuka pengguna. Sistem akan menampilkan daftar akun staf yang tersedia, termasuk informasi seperti nama, alamat email, dan peran masing-masing staf. Pengguna dengan peran manajemen memiliki beberapa opsi, termasuk menambahkan akun staf baru, mengedit informasi akun staf yang ada, atau menghapus akun staf jika diperlukan.

4. Rancangan Desain Database

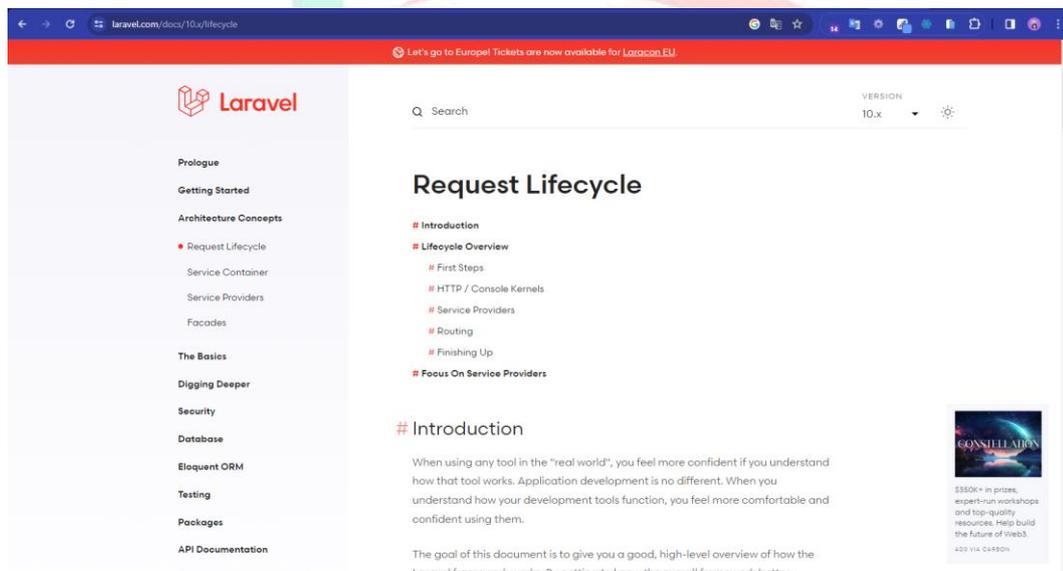


Gambar 3.11 Rancangan desain database

3.2.3 Pemahaman *tools*

Pemahaman tentang alat-alat yang akan digunakan dalam pengembangan sangat penting. Praktikan harus mempelajari Laravel sebagai *framework* PHP yang digunakan untuk pengembangan *backend*. Mereka juga harus memahami Inertia JS sebagai alat yang menghubungkan *frontend* dan *backend*, serta React JS yang digunakan dalam pengembangan *frontend*. Pemahaman alat-alat ini membantu praktikan dalam mengintegrasikan komponen-komponen sistem dengan baik. Berikut merupakan rangkaian pemahaman *tools* yang akan digunakan praktikan untuk mengembangkan *backend* webstie perumahan alpen hills

1. Pemahaman *Framework* Laravel



Gambar 3.12 Dokumentasi *Framework* Laravel Konsep Arsitektur

Gambar 3.12 merupakan dokumentasi *Framework* Laravel resmi versi 10, sebelum melakukan pengembengan area *backend*, praktikan diminta untuk memahami secara mendalam tentang *Framework* Laravel. *Framework* Laravel adalah salah satu kerangka kerja berbasis bahasa pemrograman PHP yang menggunakan konsep *Model View Controller* (MVC) (Yudho & Perasetyo, 2018). Praktikan harus mempelajari konsep

arsitektur, struktur direktori, dan fitur-fitur utama yang disediakan oleh Laravel. Hal ini dimaksudkan supaya praktikan memiliki fundamental yang kuat dalam merancang dan mengembangkan backend website Perumahan Alpen Hills.

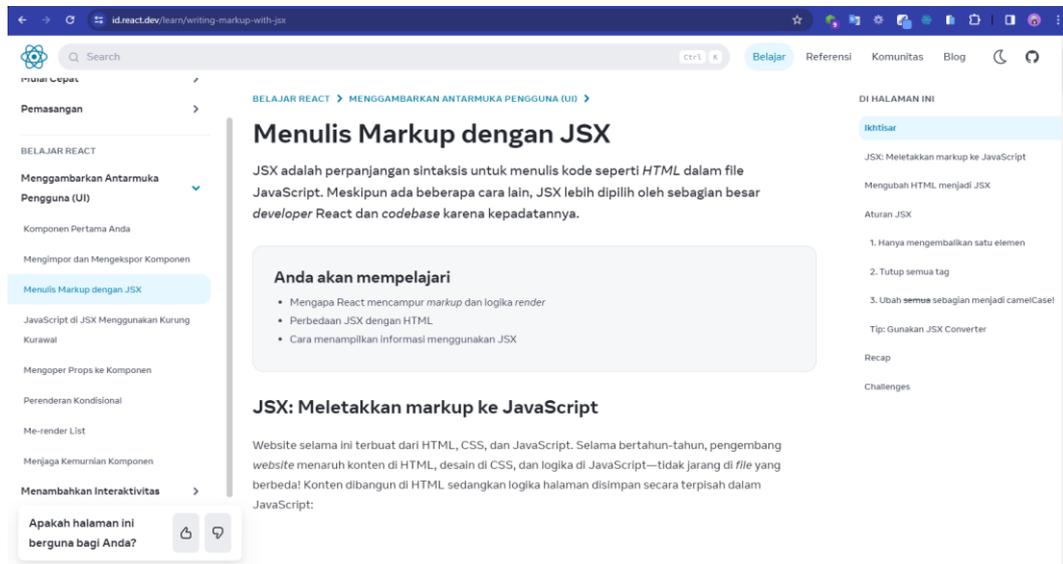
2. Pemahaman *Library* Inertia JS



Gambar 3.13 Dokumentasi *Library* Inertia JS

Selanjutnya, praktikan harus memahami Inertia JS. Inertia JS adalah alat yang memfasilitasi pengembangan aplikasi web dengan menghubungkan *frontend* dan *backend* dengan cara yang efisien dan sederhana. Gambar 3.13 menunjukkan dokumentasi *Library* Inertia JS yang akan menjadi panduan praktikan dalam memahami cara menggunakan alat ini. Inertia JS membantu dalam membangun aplikasi web yang dinamis dengan merespons perubahan-perubahan di sisi *frontend* tanpa perlu me-refresh seluruh halaman. Dokumentasi Inertia JS adalah sumber informasi utama yang akan membantu praktikan memahami konsep-konsep dasar dan praktik terbaik dalam penggunaan alat ini. Dengan pemahaman yang kuat tentang Framework Laravel, React JS, dan Inertia JS, praktikan akan siap untuk menjalankan tugas mereka dalam mengembangkan *backend* website Perumahan Alpen Hills dengan baik.

3. Pemahaman *Library* React JS



Gambar 3.14 Dokumentasi React JS Komunitas Indonesia

Setelah memahami Framework Laravel dan Library Inertia JS sebagai dasar pengembangan *backend*, praktikan kemudian memfokuskan pada pemahaman Library React JS. React JS adalah salah satu library JavaScript yang sangat populer digunakan dalam pengembangan aplikasi web modern, terutama dalam pengembangan *frontend*. React js dibangun barawal dari kebutuhan Facebook untuk menciptakan tampilan antar muka pengguna yang dinamis (Minnick, 2022). Gambar 3.14 menunjukkan dokumentasi React JS dari Komunitas Indonesia. Praktikan disarankan untuk memahami konsep-konsep dasar React JS, termasuk komponen, virtual DOM, dan siklus hidup komponen. Mereka juga akan mempelajari bagaimana menggunakan React JS untuk membangun antarmuka pengguna yang dinamis dan responsif. Dokumentasi dari Komunitas Indonesia menjadi sumber rujukan untuk memahami React JS, dan praktikan dapat mengaplikasikan pengetahuan ini dalam pengembangan *backend* website Perumahan Alpen Hills.

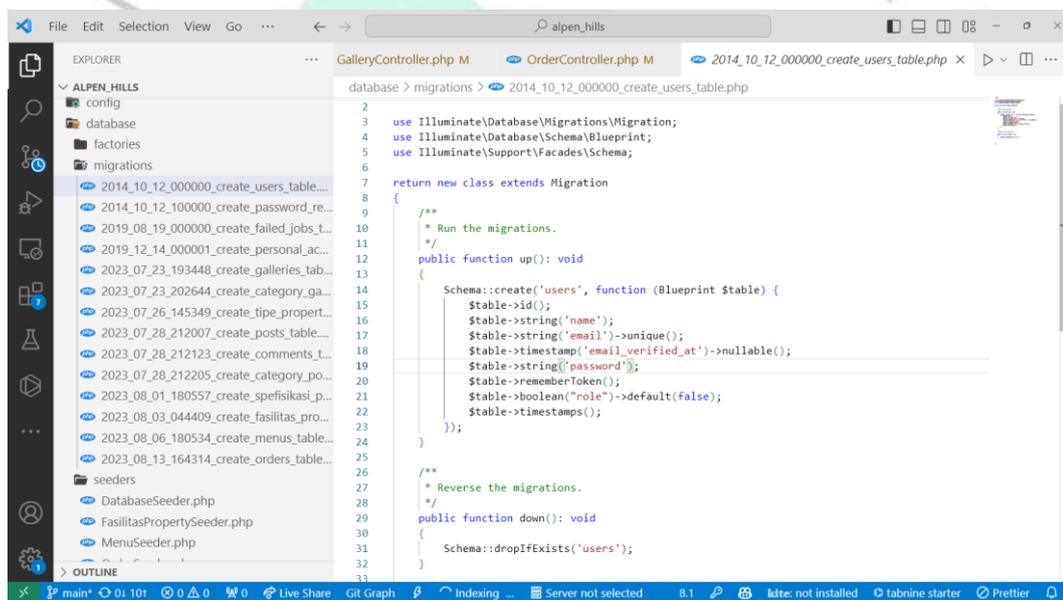
3.2.5 *Implemtation* (Implementasi)

Tugas utama praktikan adalah mengembangkan komponen *backend* website Perumahan Alpen Hills sesuai dengan kebutuhan dan desain sitem. Proses ini mencakup pembuatan berbagai fungsi, modul,

dan fitur yang mendukung operasional website, seperti manajemen konten, otentikasi pengguna, pengelolaan database, dan lainnya. Praktikan harus memastikan bahwa semua fitur berjalan dengan baik, memiliki kinerja yang optimal, dan terintegrasi dengan frontend dengan baik.

1. Implementasi skema database

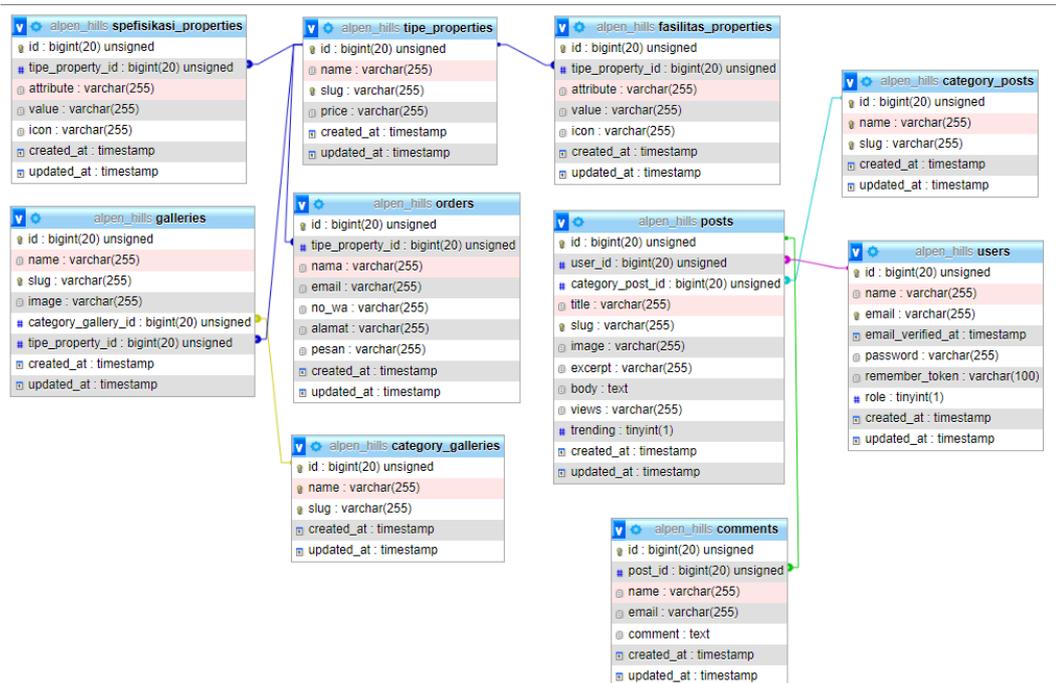
Dalam membangun fitur dan modul yang sudah dirancang praktikan memulai dengan melakukan implementasi desain database dengan menggunakan salah satu fitur yang ada pada *framework* yaitu Migrasi Database. Gambar 3.15 merupakan proses coding dalam pembuatan table migrasi database.



```
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      */
12     public function up(): void
13     {
14         Schema::create('users', function (Blueprint $table) {
15             $table->id();
16             $table->string('name');
17             $table->string('email')->unique();
18             $table->timestamp('email_verified_at')->nullable();
19             $table->string('password');
20             $table->rememberToken();
21             $table->boolean('role')->default(false);
22             $table->timestamps();
23         });
24     }
25
26     /**
27      * Reverse the migrations.
28      */
29     public function down(): void
30     {
31         Schema::dropIfExists('users');
32     }
33 }
```

Gambar 3.15 Implementasi skema database

Setelah membuat skema database sesuai desain yang telah dirancang praktikan melakukan migrasi table menggunakan fitur di Laravel yang sudah disediakan, pada Gambar 3.15 merupakan hasil dari migrasi table database



Gambar 3.16 Ouput migrasi database

2. Implementasi fitur Autentikasi

Selanjutnya praktikan bertanggung jawab untuk mengimplementasikan sistem autentikasi pengguna, pengelolaan akun, dan pengaturan peran (role) pada website Perumahan Alpen Hills. Fitur ini memungkinkan pengguna untuk login ke akun mereka dan mengelola akun serta peran mereka dalam sistem. Gambar 3.16 merupakan proses coding pembuatan fitur autentikasi.

```

app > Http > Controllers > AuthAdmin > AuthController.php
26     });
27   }
28
29   public function show(User $user): Response
30   {
31     return Inertia::render('Admin/Profile/Index', [
32       'title' => 'Profile',
33       'user' => $user,
34     ]);
35   }
36
37   /**
38    * Store a newly created resource in storage.
39    */
40   public function store(LoginRequest $request): RedirectResponse
41   {
42     $request->authenticate();
43     $request->session()->regenerate();
44     return redirect()->intended(RouteServiceProvider::HOME);
45   }
46
47   public function update(Request $request, User $user)
48   {
49     $request->validate([
50       'name' => 'required|string|max:255',
51       'email' => 'required|string|email|max:255|unique:users,email,' . $user->id,
52       'new_password' => 'nullable|required_with:confirm_password|min:8|max:255',
53       'confirm_password' => 'nullable|max:255|same:new_password|required_with:new_password',
54     ], [
55       'name.required' => 'Nama tidak boleh kosong',
56       'name.max' => 'Nama maksimal 255 karakter',
57     ]);
58   }
59 }

```

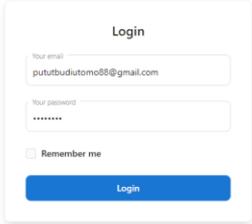
Gambar 3.17 implementasi fitur autentikasi

Dalam tahap implementasi ini, praktikan membuat berbagai fungsi dan modul yang memungkinkan pengguna untuk melakukan hal berikut:

- Login
Pengguna dapat memasukkan kredensial mereka (seperti nama pengguna dan kata sandi) untuk masuk ke akun mereka di website.
- Kelola Akun
Pengguna dapat mengakses dan mengedit informasi akun mereka, seperti alamat email, kata sandi, atau data profil lainnya.
- Pengaturan Role
Praktikan mengimplementasikan mekanisme yang memungkinkan sistem untuk mengenali peran (role) pengguna, sehingga akses dan izin yang sesuai dapat diberikan kepada pengguna sesuai dengan peran mereka. Ini penting untuk mengatur siapa yang memiliki akses ke fitur dan data tertentu.

Praktikan memastikan bahwa implementasi fitur autentikasi dan pengelolaan akun ini menyesuaikan dengan rancangan sistem yang

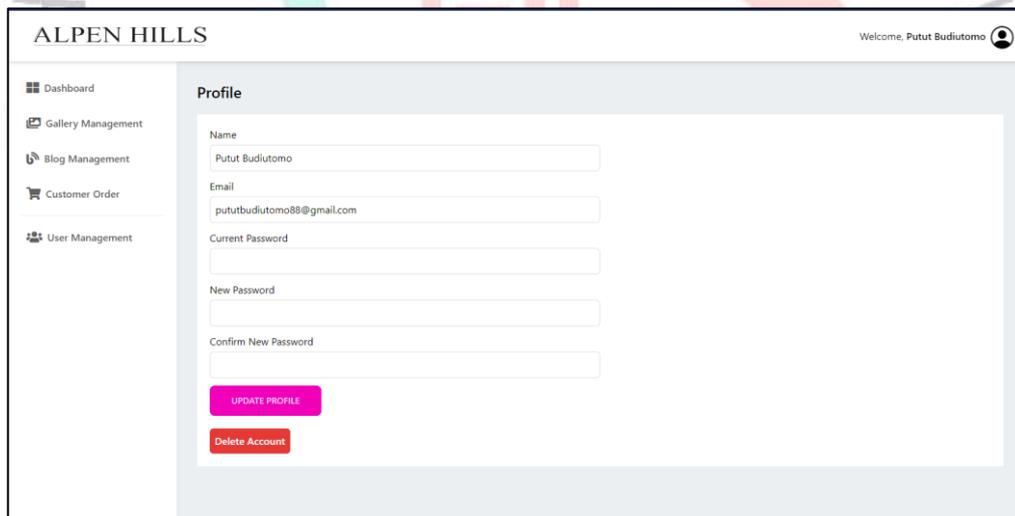
sudah dirancang sesuai dengan kebutuhan pengguna. Pada gambar 3.18 dan gambar 3.19 merupakan *output* dari proses implentasi.



The screenshot shows a login form with the following elements:

- Title: Login
- Input field for email: Your email (pututbudutomo88@gmail.com)
- Input field for password: Your password (masked with dots)
- Checkbox: Remember me
- Submit button: Login

Gambar 3.18 Output implementasi fitur login



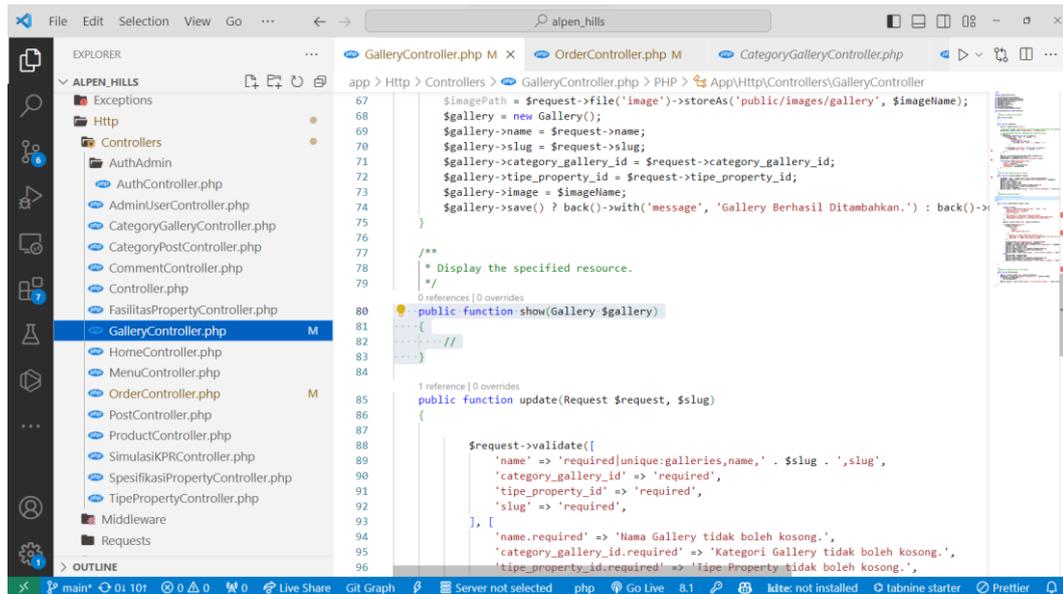
The screenshot shows a user profile management page with the following elements:

- Header: ALPEN HILLS (left) and Welcome, Putut Budiutomo (right)
- Left sidebar menu: Dashboard, Gallery Management, Blog Management, Customer Order, User Management
- Main content area: Profile
- Form fields: Name (Putut Budiutomo), Email (pututbudutomo88@gmail.com), Current Password, New Password, Confirm New Password
- Buttons: UPDATE PROFILE (pink), Delete Account (red)

Gambar 3.19 Output implementasi fitur kelola akun

3. Implementasi modul kelola konten

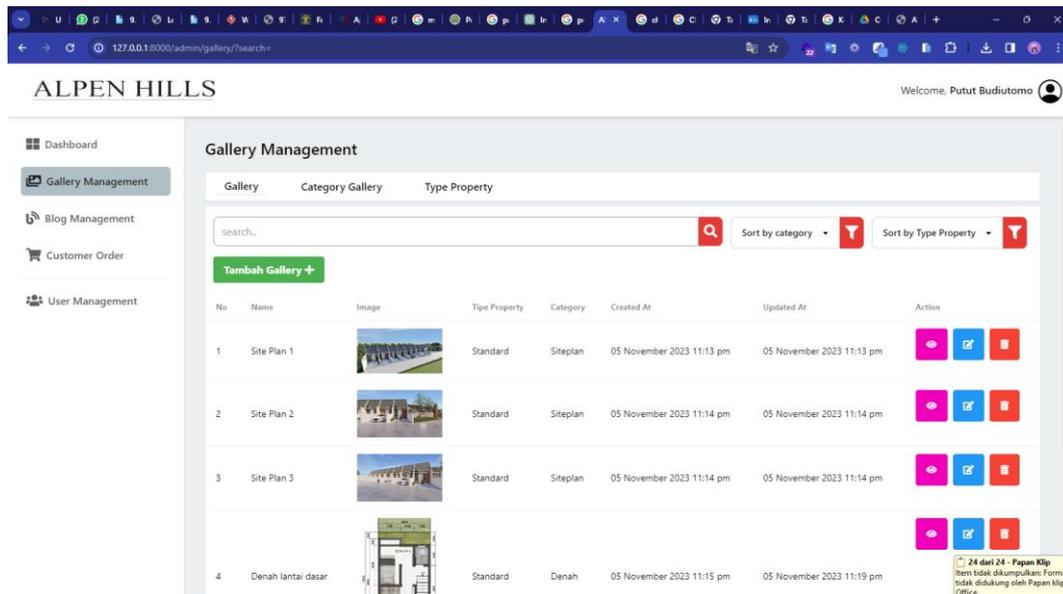
Dalam tahap implementasi modul kelola konten, praktikan mengembangkan dua bagian utama: fitur kelola konten galeri dan fitur kelola konten blog. Gambar 3.20 menggambarkan proses coding implementasi fitur kelola konten galeri.



```
67 $imagePath = $request->file('image')->storeAs('public/images/gallery', $imageName);
68 $gallery = new Gallery();
69 $gallery->name = $request->name;
70 $gallery->slug = $request->slug;
71 $gallery->category_gallery_id = $request->category_gallery_id;
72 $gallery->type_property_id = $request->type_property_id;
73 $gallery->image = $imageName;
74 $gallery->save() ? back()->with('message', 'Gallery Berhasil Ditambahkan.') : back()->with('message', 'Gagal Ditambahkan.').back();
75 }
76
77 /**
78  * Display the specified resource.
79  */
80 public function show(Gallery $gallery)
81 {
82     //
83 }
84
85 public function update(Request $request, $slug)
86 {
87     $request->validate([
88         'name' => 'required|unique:galleries,name,' . $slug . ',slug',
89         'category_gallery_id' => 'required',
90         'type_property_id' => 'required',
91         'slug' => 'required',
92     ]), [
93         'name.required' => 'Nama Gallery tidak boleh kosong.',
94         'category_gallery_id.required' => 'Kategori Gallery tidak boleh kosong.',
95         'type_property_id.required' => 'Tipe Property tidak boleh kosong.',
96     ];
```

Gambar 3.20 Proses Coding implementasi fitur kelola konten galeri

Dalam proses ini, praktikan mengembangkan fungsi dan modul yang memungkinkan pengguna untuk mengelola konten galeri, seperti mengunggah gambar, mengatur deskripsi, mengelompokkan gambar-gambar dalam kategori tertentu, dan melakukan operasi dasar seperti penambahan, pembaruan, dan penghapusan. Output dari proses implementasi ini, seperti yang ditunjukkan pada Gambar 3.18, adalah tampilan antarmuka pengguna yang memungkinkan pengguna untuk mengelola konten galeri dengan mudah.



Gambar 3.21 Output implementasi fitur kelola konten galeri

Sementara itu, Gambar 3.22 dan Gambar 3.23 merupakan proses coding implementasi fitur kelola konten blog. Fitur ini memungkinkan pengguna untuk mengelola konten blog, termasuk pembuatan, penambahan, pengeditan, dan penghapusan postingan blog. Praktikan mengimplementasikan modul ini dengan memastikan pengguna dapat mengelola konten blog dengan lancar, termasuk penambahan gambar, pengaturan kategori, dan pemformatan teks. Gambar 3.20 menunjukkan tampilan antarmuka pengguna untuk mengelola konten blog setelah proses implementasi selesai.

```

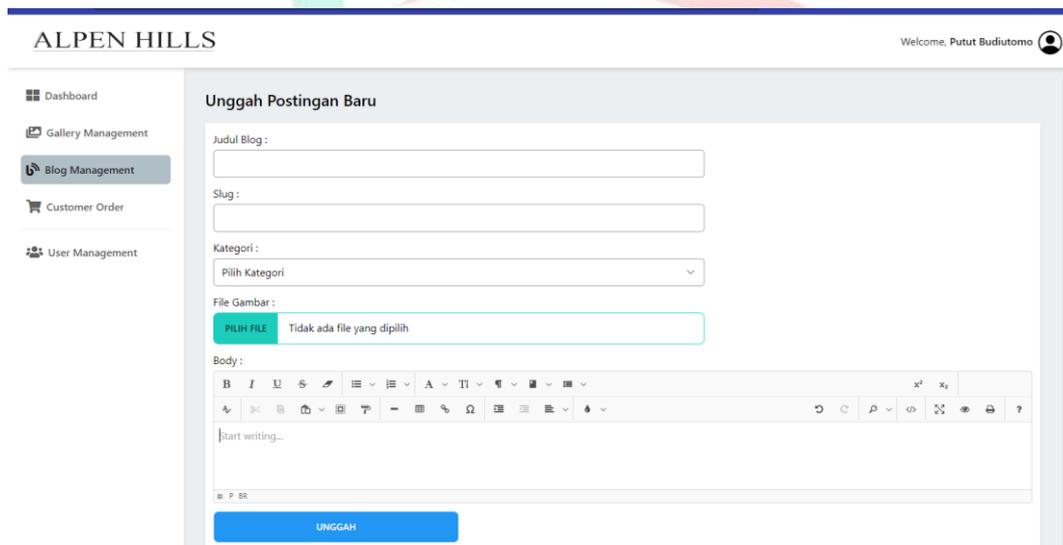
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
public function create()
{
    $categoryPost = CategoryPost::all();
    return Inertia::render('Admin/Posts/TambahPost', [
        'title' => 'Unggah Postingan Baru',
        'categoryPost' => $categoryPost,
    ]);
}

/**
 * Store a newly created resource in storage.
 */
public function store(StorePostRequest $request)
{
    $imageName = time() . '-' . $request->image->getClientOriginalName();
    $imagePath = $request->file('image')->storeAs('public/images/blog', $imageName);
    $post = new Post();
    $post->category_post_id = $request->category_post_id;
    $post->user_id = $request->user_id;
    $post->title = $request->title;
    $post->slug = $request->slug;
    $post->image = $imagePath;
    $post->body = $request->body;
    $post->excerpt = Str::limit(strip_tags($request->body), 100);
    $post->save() ? back()->with('message', 'Post Berhasil Ditambahkan.') : back()->with('
}

/**
 * Display the specified resource.
 */

```

Gambar 3.22 Proses coding implementasi fitur kelola konten blog

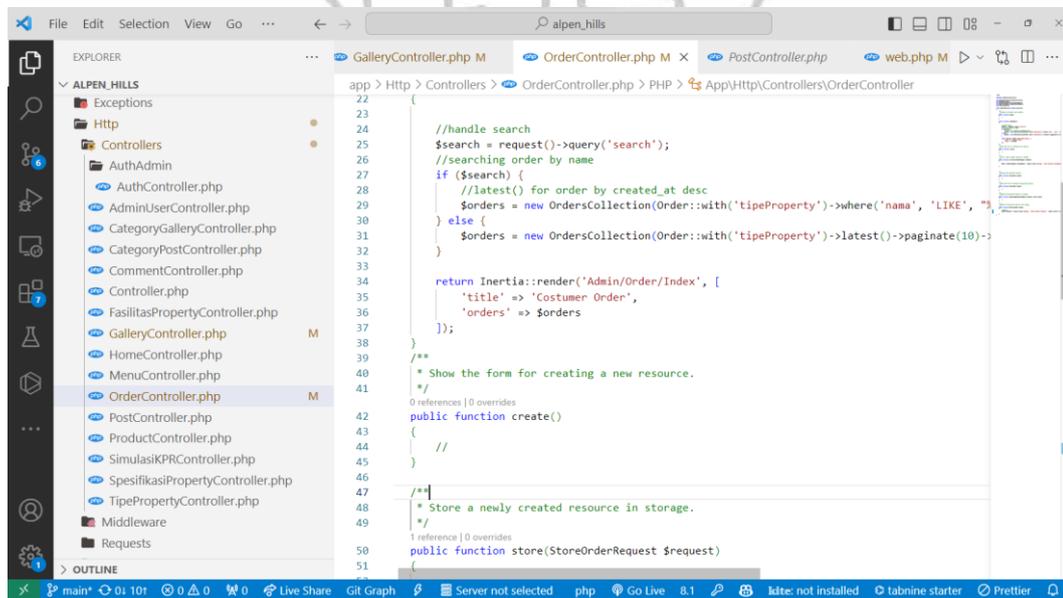


Gambar 3.23 Output implementasi fitur kelola konten blog

Kedua fitur ini, kelola konten galeri dan kelola konten blog, memungkinkan pemeliharaan dan pengelolaan konten website dengan efisien dan sesuai dengan kebutuhan pengguna serta pemangku kepentingan. Praktikan memastikan bahwa fitur-fitur ini terintegrasi dengan baik dalam backend sistem dan memiliki performa yang optimal untuk mendukung pengalaman pengguna yang baik.

4. Implementasi modul kelola data pemesanan

Dalam tahap implementasi modul kelola data pemesanan, praktikan fokus pada mengembangkan fitur yang memungkinkan pengguna untuk mengelola data pemesanan pada website. Gambar 3.24 menggambarkan proses coding implementasi fitur kelola konten data pemesanan.

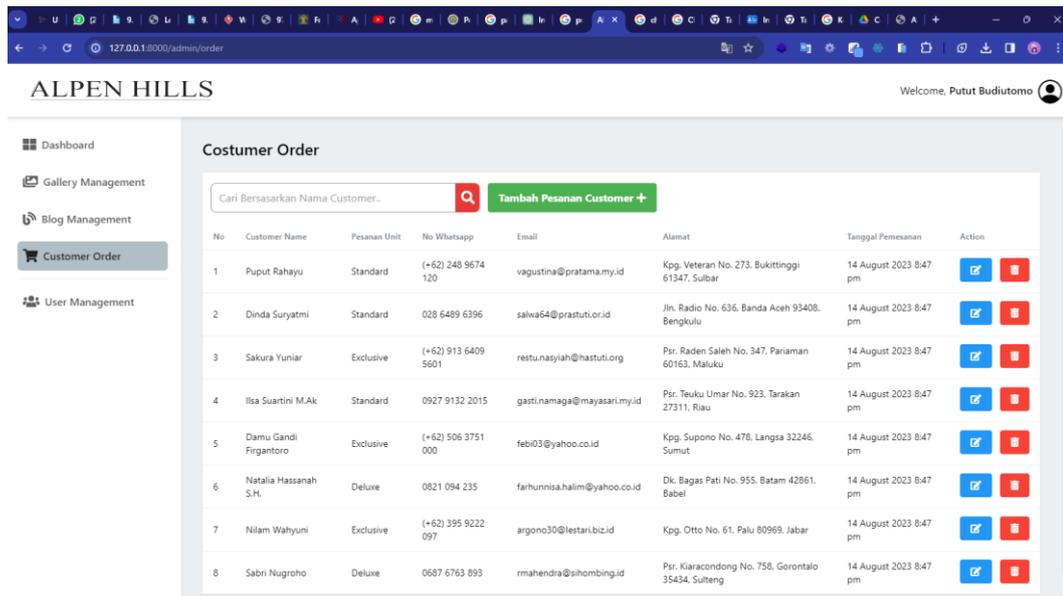


```
app > Http > Controllers > OrderController.php > PHP > App\Http\Controllers\OrderController
22 {
23
24 //handle search
25 $search = request()->query('search');
26 //searching order by name
27 if ($search) {
28 //latest() for order by created_at desc
29 $orders = new OrdersCollection(Order::with('tipeProperty')->where('nama', 'LIKE', "%$search%")->latest());
30 } else {
31 $orders = new OrdersCollection(Order::with('tipeProperty')->latest()->paginate(10)->toArray());
32 }
33
34 return Inertia::render('Admin/Order/Index', [
35 'title' => 'Customer Order',
36 'orders' => $orders
37 ]);
38 }
39
40 /**
41 * Show the form for creating a new resource.
42 */
43 0 references | 0 overrides
44 public function create()
45 {
46 //
47 }
48
49 /**
50 * Store a newly created resource in storage.
51 */
52 1 reference | 0 overrides
53 public function store(StoreOrderRequest $request)
54 {
55 }
```

Gambar 3.24 Proses coding implementasi fitur kelola konten data pemesanan

Proses coding ini mencakup pengembangan berbagai fungsi dan modul, termasuk kemampuan untuk menampilkan data pemesanan, mengelola informasi detail pesanan, mengkonfirmasi pesanan, dan melakukan operasi dasar seperti penambahan, pembaruan, dan penghapusan pemesanan.

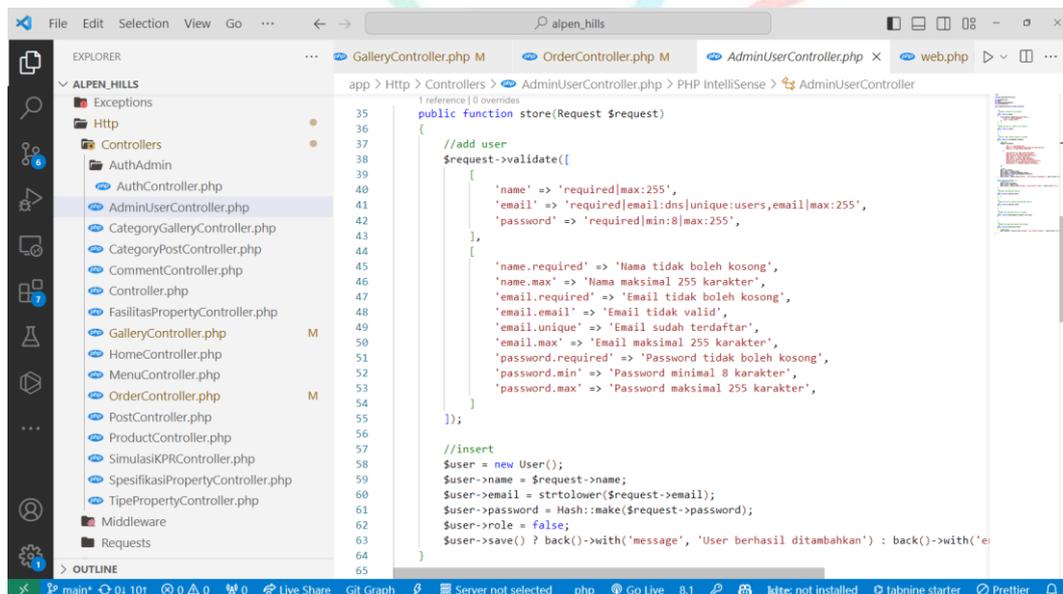
Setelah proses implementasi selesai, pengguna akan memiliki akses ke tampilan antarmuka pengguna yang memungkinkan mereka untuk melihat, memproses, dan mengelola data pemesanan dengan mudah. Gambar 3.25, menampilkan tampilan antarmuka pengguna setelah proses implementasi selesai.



Gambar 3.25 Output implementasi fitur kelola konten data pemesanan

5. Implementasi modul kelola akun staf

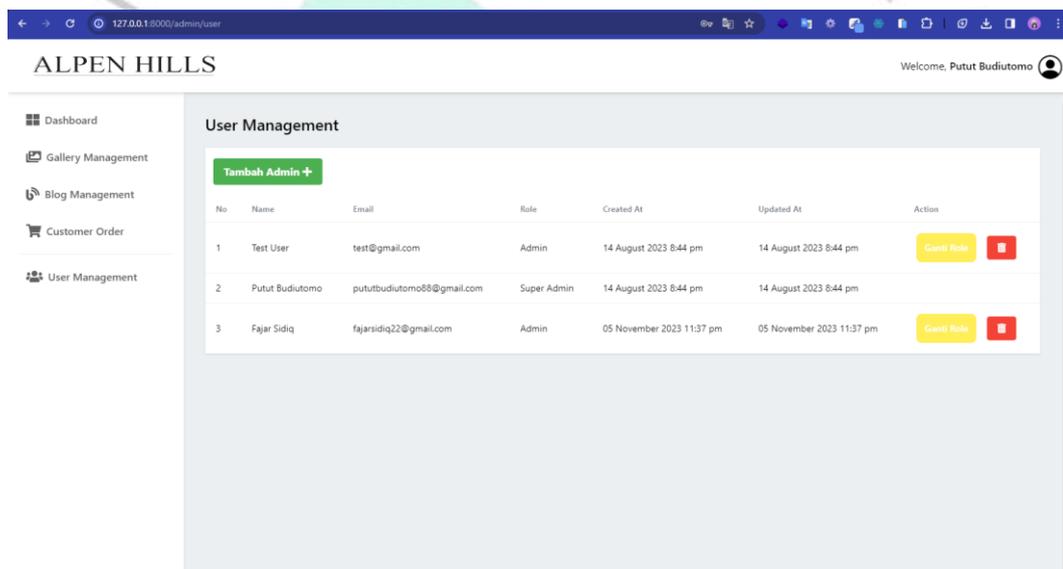
Pada tahap implementasi modul kelola akun staf, praktikan juga bertugas mengembangkan fitur yang memungkinkan pengelolaan akun staf di dalam sistem. Gambar 3.26 menggambarkan proses coding implementasi fitur kelola akun staf.



Gambar 3.26 Proses Coding implementasi fitur kelola akun staf

Proses coding ini melibatkan pengembangan berbagai fungsi dan modul yang memungkinkan pengguna, yang dalam hal ini adalah pengelola sistem, untuk mengelola akun staf. Fitur ini mencakup pembuatan, pembaruan, dan penghapusan akun staf, serta pengaturan izin dan peran staf sesuai kebutuhan.

Setelah proses implementasi selesai, pengelola sistem dapat menggunakan antarmuka pengguna yang sesuai, yang mencakup tampilan daftar staf, pengaturan izin akses, dan penggantian role staf. Gambar 3.27, yang tidak tersedia dalam teks sebelumnya, akan menampilkan tampilan antarmuka pengguna setelah proses implementasi selesai.



Gambar 3.27 Output implementasi fitur kelola akun staf

3.3 Kendala Yang Dihadapi

Sebagai seorang praktikan yang bertugas sebagai *Backend Developer* di MP Project, praktikan menghadapi sejumlah kendala yang memengaruhi pelaksanaan pekerjaan saya. Kendala-kendala ini mencakup berbagai aspek, terutama yang berhubungan dengan aspek teknis. Beberapa dari kendala ini termasuk:

1. Perbedaan pemahaman dan penguasaan teknologi

Salah satu kendala utama yang praktikan hadapi adalah perbedaan dalam pemahaman dan penguasaan teknologi di antara anggota tim pengembang. Selama melakukan kerja profesi di MP Project praktikan diberikan tanggung jawab untuk menentukan teknologi yang akan digunakan dalam melakukan pengembangan namun tetap harus mempertimbangan dari aspek kebutuhan proyek, kinerja, kemudahan penggunaan, dan ketersediaan sumber daya Tim. Tim pengembang terdiri dari individu dengan latar belakang yang berbeda dan pengetahuan teknologi yang beragam. Hal ini menyebabkan ketidakpastian dalam menentukan teknologi yang akan digunakan untuk mengembangkan website Perumahan Alpen Hills. Ada anggota tim lebih terbiasa dengan bahasa pemrograman Javascript, sementara anggota yang lain lebih memahami bahasa pemrograman PHP. Dalam situasi ini, praktikan harus memastikan bahwa pemilihan dilakukan berdasarkan pertimbangan obyektif yang memenuhi kebutuhan proyek.

2. Perubahan persyaratan proyek

Proyek pengembangan perangkat lunak sering kali mengalami perubahan persyaratan yang mempengaruhi jadwal dan rencana pengembangan. Ketika persyaratan berubah, praktikan dan tim pengembang perlu menyesuaikan kode yang sudah ada dan mengembangkan fitur tambahan, yang menambah kompleksitas dan waktu yang dibutuhkan. Pada awalnya, proyek pengembangan website Perumahan Alpen Hills telah mempertimbangkan hanya beberapa fitur utama, termasuk sistem pemesanan properti, manajemen konten, dan katalog properti. Tim pengembang, termasuk praktikan, telah merancang arsitektur backend yang sesuai dengan persyaratan awal dan telah memulai pengembangan. Namun, ketika proyek berada di tahap pengembangan, pemangku kepentingan proyek, seperti pihak manajemen Perumahan Alpen Hills, mengkomunikasikan bahwa mereka ingin menambahkan sejumlah fitur baru yang signifikan. Fitur-fitur ini mencakup penambahan fitur blog, layanan notifikasi real-time, dan sistem ulasan pelanggan. Perubahan ini mempengaruhi jadwal dan kebutuhan pengembangan.

3. Kurangnya Pemahaman Seragam tentang Prioritas

Selama proses pengembangan, terkadang anggota tim memiliki pandangan yang beragam tentang prioritas proyek. Terkadang beberapa anggota tim lebih fokus pada aspek antarmuka dan kenyamanan pengguna, sementara yang lain lebih menekankan aspek keamanan atau kegunaan. Kesimpang-siuran pemahaman tentang tujuan dan prioritas ini memunculkan konflik dalam pengambilan keputusan dan menghambat kemajuan proyek. Dalam tim pengembangan website Perumahan Alpen Hills, beberapa anggota tim, terutama pengembang *frontend*, sangat mengutamakan aspek kegunaan dan tampilan antarmuka pengguna yang responsif. Mereka merasa bahwa pengalaman pengguna harus menjadi yang utama dan bahwa performa harus sejalan dengan harapan pengguna. Di sisi lain, beberapa anggota tim, termasuk praktikan sebagai *Backend Developer*, lebih berfokus pada aspek kinerja dan keamanan. Mereka memandang bahwa website harus berjalan dengan cepat dan aman dari potensi ancaman keamanan.

Ketidaksesuaian dalam pandangan ini muncul ketika tim harus mengambil keputusan tentang bagaimana mengimplementasikan fitur-fitur baru dalam website. Sebagai contoh, saat mempertimbangkan pengembangan fitur pemesanan unit, beberapa anggota tim cenderung memilih solusi yang lebih mengutamakan keamanan, sementara yang lain ingin memastikan responsivitas dan kenyamanan pengguna.

3.4 Cara Mengatasi Kendala

Dalam menghadapi permasalahan atau kendala yang dialami, praktikan sebagai tim pengembang di MP Project memikirkan solusi dan melakukan berbagai upaya untuk mengatasi kendala tersebut untuk memastikan kelancaran pelaksanaan proyek pengembangan website Perumahan Alpen Hills, upaya dan solusi yang dilakukan antara lain :

1. Diskusi dan Penyelarasan

Dalam menentukan teknologi yang akan digunakan praktikan dan tim pengembang melakukan riset terlebih dahulu terkait teknologi-

teknologi yang sesuai dengan kebutuhan proyek dan kemampuan anggota tim. Riset yang dilakukan mempertimbangkan dari beberapa aspek yaitu antara lain dokumentasi teknologi yang lengkap, jumlah komunitas yang memadahi, integrasi fitur yang mudah, dan juga aspek keamanan dan kinerja dari teknologi yang akan dipakai. Praktikan dan anggota tim juga melakukan diskusi rutin sebelum dan sesudah mengimplementasikan sebuah fitur seperti menyelaraskan pemahaman tentang teknologi yang digunakan dan juga melakukan peninjauan ulang kode yang telah dibuat antar anggota tim. Kemudian dalam diskusi yang dilakukan anggota. Diskusi ini dapat membantu dalam memahami kelebihan dan kelemahan masing-masing teknologi serta bagaimana teknologi tersebut dapat memenuhi kebutuhan proyek.

2. Manajemen Perubahan Persyaratan

Untuk mengatasi perubahan persyaratan proyek, praktikan dan tim pengembang menerapkan pendekatan yang melibatkan praktikan dalam proses elisitasi persyaratan. Elisitasi adalah metode di mana praktikan dan tim berdiskusi langsung dengan pemangku kepentingan proyek untuk memahami kebutuhan dan harapan mereka secara rinci. Proses elisitasi ini membantu praktikan dan tim dengan baik dalam mendokumentasikan perubahan persyaratan secara terinci dan jelas.

Selain itu, setiap perubahan persyaratan yang diidentifikasi melewati proses evaluasi yang ketat dan persetujuan oleh semua pemangku kepentingan proyek, persetujuan ini mencakup pemahaman yang mendalam oleh semua pihak terkait tentang dampak perubahan yang akan diimplementasikan. Hal ini membantu meminimalkan risiko perubahan tak terduga atau perubahan yang tidak sepenuhnya dipahami oleh semua pemangku kepentingan.

Setelah mengevaluasi dampak, prioritas harus ditentukan praktikan dan juga tim pengembang melakukan tahap perencanaan ulang, di mana tim pengembang memprioritaskan perubahan berdasarkan urgensi dan kebutuhan. Hal tersebut dilakukan supaya

membantu dalam mengalokasikan sumber daya dengan lebih efisien dan memastikan proyek tetap berjalan sesuai rencana.

Kemudian semua perubahan persyaratan didokumentasikan secara rinci dan disetujui oleh semua pemangku kepentingan sebelum diimplementasikan. Dokumentasi ini mencakup perubahan apa yang dibutuhkan, alasan perubahan, dan dampaknya. Hal ini memastikan bahwa tidak ada ketidakjelasan dalam mengenai perubahan yang terjadi. Melalui pendekatan ini, praktikan dan tim pengembang dapat mengatasi kendala perubahan persyaratan proyek dengan lebih efektif, memastikan kelancaran pengembangan proyek, dan memenuhi harapan pemangku kepentingan.

3. Pembuatan Roadmap Proyek dan *Pair Programming*

Praktikan dan tim pengembang membuat roadmap proyek yang berisi daftar prioritas dan target pencapaian proyek dalam rentang waktu tertentu. Dengan roadmap yang jelas, seluruh tim akan memiliki pandangan yang lebih terstruktur tentang apa yang harus diprioritaskan dalam pengembangan. Roadmap ini menjadi panduan tim dalam menentukan prioritas pada setiap tahap pengembangan. Praktikan dan tim berusaha untuk mengembangkan solusi yang mengintegrasikan aspek-aspek yang berbeda dengan cara yang seimbang. Sebagai contoh, saat mengimplementasikan fitur-fitur baru dalam proyek, praktikan dan tim akan mencari solusi yang mempertimbangkan kegunaan, kinerja, dan keamanan secara bersamaan. Ini mungkin melibatkan penggunaan teknologi tertentu, pengoptimalan kode, atau strategi desain yang sesuai.

Kemudian praktikan harus mendorong kolaborasi antar anggota tim. Hal tersebut dilakukan dengan sesi brainstorming atau pair programming, di mana anggota tim bekerja bersama untuk menemukan solusi terbaik yang memadukan berbagai aspek prioritas. Dengan berkolaborasi, anggota tim dapat memahami sudut pandang satu sama lain dan mencapai kesepakatan.

3.5 Pembelajaran Yang Diperoleh dari Kerja Profesi

Selama menjalani kerja profesi di MP Project, praktikan telah mendapatkan banyak pembelajaran yang sangat berharga yang tidak hanya meningkatkan keterampilan dan pemahaman, tetapi juga membentuk landasan yang kuat untuk karier profesional di masa depan. Berikut adalah pembelajaran yang diperoleh dari pekerjaan profesi mereka:

1. Keterampilan Manajerial dan Kepemimpinan

Praktikan terlibat dalam manajemen tim dalam proyek mereka. Mereka belajar bagaimana mengorganisir tugas, mengatur jadwal, dan mengelola sumber daya manusia dan materi yang tersedia. Selain itu, pembelajaran ini memungkinkan praktikan untuk memahami kompleksitas berkolaborasi dengan anggota tim yang memiliki latar belakang dan keterampilan yang beragam. Praktikan memperoleh keterampilan manajemen yang kuat dan juga pemahaman mengenai prinsip-prinsip kepemimpinan yang efektif.

2. Pemahaman Proses Bisnis

Melalui pengamatan langsung dan keterlibatan dalam pekerjaan profesional, praktikan memiliki kesempatan untuk mendapatkan pemahaman yang lebih mendalam tentang proses bisnis. Belajar tentang bagaimana berbagai fungsi dan departemen dalam sebuah organisasi berinteraksi dan bekerja sama untuk mencapai tujuan bisnis. Pengalaman tersebut membantu praktikan untuk memiliki wawasan yang lebih baik tentang dinamika bisnis. Praktikan belajar tentang berbagai proses bisnis, seperti pengembangan bisnis perusahaan, pemasaran, dan dukungan pelanggan. Hal tersebut memberikan pemahaman kepada praktikan dalam melihat gambaran lengkap tentang dinamika bisnis yang belum dipahami selama studi akademis.

3. *Problem Solving*

Pada saat praktikan melakukan kerja profesi di MP Project sering kali terlibat dalam pemecahan masalah yang kompleks. Praktikan diajarkan dan dibimbing dalam menganalisis masalah, mengidentifikasi opsi solusi, dan membuat keputusan yang tepat.

Praktikan juga belajar untuk beradaptasi dengan perubahan situasi dan mengatasi hambatan yang mungkin muncul.

4. Pengembangan keahlian teknis

Selain aspek manajerial dan bisnis, praktikan memiliki kesempatan untuk mendapatkan pengalaman yang sesuai dengan kebutuhan industri saat ini melalui pengembangan dan penerapan keahlian teknis yang relevan dengan pekerjaan. Praktikan dapat mempraktikkan keterampilan teknis yang dipelajari selama studi dan menguasainya lebih lanjut. Hal tersebut membuat praktikan lebih kompeten dalam bidang yang sedang digeluti dan membuat lebih berguna bagi tim kerja atau organisasi.

