

## BAB III

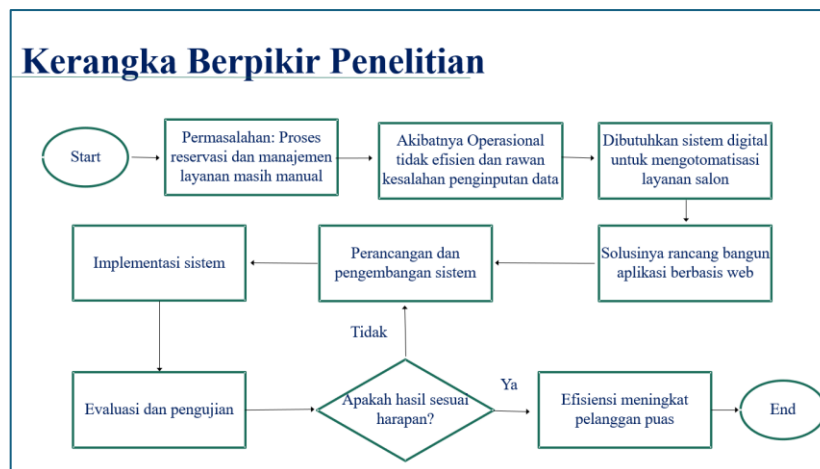
### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Kerangka Berpikir

Kerangka berpikir penelitian ini divisualisasikan dalam flowchart yang menggambarkan alur logis dari identifikasi masalah hingga evaluasi hasil. Masalah utama di Salon Purigeulis by REDDG Majalengka adalah proses reservasi dan manajemen layanan yang masih manual, menyebabkan ketidakefisienan waktu, kesalahan pencatatan data, serta ketidaknyamanan bagi pelanggan dan karyawan.

Untuk mengatasi hal tersebut, dibutuhkan sistem digital yang dapat mengotomatisasi proses agar lebih efisien dan terstruktur. Solusi yang diajukan adalah aplikasi berbasis web dengan fitur reservasi online, pembatalan mandiri, kelola data pelanggan dan karyawan, serta testimoni layanan.

Setelah perancangan dan pengembangan, sistem diimplementasikan pada operasional salon, lalu dievaluasi untuk menilai pencapaian efisiensi, pengurangan kesalahan, dan peningkatan kepuasan pelanggan. Jika hasilnya belum optimal, dilakukan evaluasi menyeluruh untuk mengidentifikasi masalah, kemudian dilanjutkan dengan perancangan ulang dan implementasi perbaikan. Pendekatan ini memastikan sistem tidak hanya berhasil di awal, tetapi juga mendukung perbaikan berkelanjutan.



**Gambar 3. 1** Kerangka Berpikir

## **3.2. Metode Penelitian**

Metode penelitian digunakan untuk memastikan bahwa proses perancangan dan pengembangan sistem dilakukan secara sistematis serta sesuai dengan kebutuhan pengguna. Dalam penelitian ini, metode yang digunakan mencakup teknik pengumpulan data dan metode pengembangan sistem.

### **3.2.1. Tempat dan Waktu Penelitian**

Penelitian ini dilakukan di lokasi dan pada waktu berikut:

1. Tempat : Salon Kecantikan Purigeulis by REDDG Majalengka.
2. Alamat : Jl. Raya KH Abdul Halim, Tonjong, Kec. Majalengka, Kabupaten Majalengka, Jawa Barat 45414.
3. Waktu penelitian : Januari 2025 – Juni 2025.

### **3.2.2. Metode Pengumpulan Data**

Untuk mendapatkan data yang relevan dalam penelitian ini, digunakan beberapa teknik pengumpulan data sebagai berikut:

#### **a. Observasi**

Observasi dilakukan dengan mengamati langsung sistem reservasi yang berjalan di Salon Kecantikan Purigeulis by REDDG Majalengka. Fokus utama observasi ini adalah bagaimana pelanggan melakukan reservasi, bagaimana *admin* mengelola jadwal pelayanan, serta kendala yang muncul akibat sistem manual yang digunakan hingga saat ini.

#### **b. Wawancara**

Wawancara dilakukan dengan pemilik dan pegawai salon untuk memperoleh informasi lebih dalam mengenai kebutuhan sistem, hambatan dalam pengelolaan layanan, serta harapan mereka terhadap sistem berbasis web yang akan dikembangkan.

### c. Kuesioner

Kuesioner digunakan untuk mengumpulkan data dari pelanggan salon terkait pengalaman mereka dalam melakukan reservasi serta ekspektasi mereka terhadap sistem yang akan dibuat. Kuesioner ini berisi pertanyaan tentang tingkat kepuasan pelanggan terhadap sistem reservasi yang ada, kemudahan dalam mengakses layanan, serta fitur yang mereka harapkan dari aplikasi berbasis web.

### 3.2.3. Metode Pengembangan Sistem

Penelitian ini menggunakan metode *Waterfall* sebagai pendekatan dalam pengembangan sistem. Model ini terdiri dari beberapa tahapan yang dilakukan secara berurutan, yaitu:

1. **Analisis Kebutuhan** : Mengidentifikasi dan menganalisis kebutuhan sistem berdasarkan hasil observasi dan wawancara.
2. **Perancangan Sistem** : Membuat rancangan sistem yang akan diimplementasikan.
3. **Implementasi** : Mengembangkan aplikasi berbasis web berdasarkan desain yang telah dibuat.
4. **Pengujian** : Melakukan uji coba terhadap sistem untuk memastikan bahwa *fitur* yang dikembangkan berfungsi sesuai dengan kebutuhan pengguna.
5. **Pemeliharaan** : Mengevaluasi serta memperbaiki sistem berdasarkan hasil pengujian dan masukan dari pengguna.

### 3.3 Prosedur Penelitian

Penelitian ini menggunakan metode kualitatif untuk memperoleh pemahaman mendalam terkait permasalahan dan kebutuhan sistem di Salon Purigeulis by REDDG Majalengka. Prosedur yang dilakukan meliputi beberapa tahap sebagai berikut:

1. **Studi Literatur**: Melakukan kajian terhadap teori, konsep, dan penelitian terdahulu yang relevan, seperti sistem reservasi berbasis web, manajemen layanan digital, dan metode pengembangan sistem *Waterfall*.

2. **Observasi Lapangan:** Mengamati langsung proses operasional di Salon Purigeulis, khususnya terkait alur reservasi dan manajemen layanan yang masih dilakukan secara manual, untuk mengidentifikasi permasalahan yang ada.
3. **Wawancara:** Mengumpulkan data dari pihak salon (pemilik, karyawan, dan pelanggan) melalui wawancara untuk mengetahui kebutuhan, harapan, dan kendala dalam sistem reservasi serta layanan yang diinginkan.
4. **Perancangan Sistem:** Berdasarkan hasil observasi dan wawancara, dilakukan perancangan sistem dengan metode *Waterfall* yang meliputi analisis kebutuhan, desain sistem, implementasi, pengujian, hingga evaluasi.
5. **Implementasi dan Uji Coba:** Mengimplementasikan aplikasi berbasis web ke dalam lingkungan operasional salon dan melakukan uji coba dengan melibatkan pengguna untuk mendapatkan *feedback*.
6. **Evaluasi:** Mengevaluasi hasil implementasi melalui analisis kualitatif berdasarkan *feedback* pengguna untuk menilai efektivitas sistem dalam meningkatkan efisiensi operasional dan kepuasan pelanggan.

### 3.4. Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk mengidentifikasi fitur dan kapabilitas yang harus dimiliki sistem agar dapat berfungsi secara optimal. Kebutuhan ini menjadi dasar dalam proses perancangan dan pengembangan aplikasi.

#### 3.4.1. Kebutuhan Fitur Dalam Sistem

Kebutuhan fitur dalam sistem merupakan tahap penting dalam proses perancangan aplikasi, yang berfungsi untuk mendefinisikan secara rinci fungsi-fungsi utama yang harus tersedia agar sistem dapat berjalan sesuai dengan tujuan yang telah ditentukan.

Untuk memperoleh daftar kebutuhan fitur ini, dilakukan beberapa langkah analisis, yaitu:

1. **Analisis masalah dan kebutuhan pengguna** melalui observasi langsung ke Salon Purigeulis by RDDG serta wawancara dengan pemilik, karyawan, dan pelanggan untuk mengetahui kendala pada sistem reservasi manual yang berjalan saat ini.
2. **Analisis dokumen dan proses bisnis** dengan mempelajari alur kerja salon, seperti pencatatan reservasi, layanan, dan data pelanggan untuk diidentifikasi bagian-bagian yang dapat dioptimalkan.
3. **Studi literatur dan aplikasi pembandingan** untuk melihat fitur-fitur yang umum ada pada sistem reservasi berbasis web lain, kemudian disesuaikan dengan kebutuhan lokal Purigeulis.
4. **Validasi kebutuhan** dengan pihak stakeholder untuk memastikan bahwa fitur-fitur yang dirancang sudah sesuai harapan dan mendukung operasional salon.

Hasil dari analisis tersebut menghasilkan daftar kebutuhan fitur berikut yang dirancang untuk mendukung kinerja sistem secara optimal:

**Tabel 3. 1** Kebutuhan Fitur Dalam Sistem

Fitur	Deskripsi
Fitur Reservasi	Memungkinkan pelanggan melakukan reservasi layanan secara online, termasuk memilih jenis layanan, jadwal, dan karyawan yang diinginkan.
Fitur Menu Layanan	Menampilkan daftar layanan yang ditawarkan oleh salon, lengkap dengan deskripsi, harga, dan durasi masing-masing layanan.
Fitur Artikel	Menyediakan informasi seputar tips kecantikan, tren, atau promosi salon yang dapat dibaca oleh pelanggan.
Fitur Testimoni	Pelanggan dapat memberikan ulasan dan pengalaman positif mereka terhadap layanan salon, yang akan ditampilkan di halaman khusus.
Fitur Kritik Saran	Mengumpulkan masukan atau keluhan dari pelanggan mengenai layanan salon, sebagai bahan evaluasi untuk meningkatkan kualitas pelayanan.
Fitur Kelola Profil Usaha	Admin dapat memperbarui informasi salon seperti alamat, kontak, deskripsi, dan logo, yang akan ditampilkan di halaman utama aplikasi.
Fitur Kelola Karyawan	Mengelola data karyawan salon, termasuk informasi pribadi, jadwal kerja, dan status keaktifan sebagai tenaga kerja di aplikasi.
Fitur Kelola Admin	Mengatur akses dan hak admin dalam mengelola fitur aplikasi, termasuk menambah, mengubah, atau menghapus data admin lainnya.
Fitur Kelola Akun Pelanggan	Mengelola data akun pelanggan yang terdaftar, yang didalamnya terdapat informasi nama, kontak, alamat, <i>username</i> . dan email pelanggan.

**b. Kebutuhan Data Dalam Sistem**

Sistem ini membutuhkan beberapa komponen data utama yang akan digunakan untuk mendukung proses reservasi dan manajemen layanan salon. Data tersebut disimpan dalam bentuk tabel pada basis data dan saling terhubung satu sama lain. Berikut ini adalah nama-nama tabel beserta fungsinya:

**Tabel 3. 2** *Kebutuhan Data Sistem*

No	Nama Tabel	Deskripsi Singkat
1	pelanggan	Menyimpan data akun pelanggan salon
2	booking	Menyimpan informasi reservasi layanan oleh pelanggan
3	karyawan	Menyimpan data karyawan salon
4	menu	Menyimpan jenis layanan dan detailnya
5	testimoni	Menyimpan testimoni dari pelanggan
6	artikel	Menyimpan artikel atau konten yang ditampilkan di web
7	gambar	Menyimpan gambar yang digunakan dalam konten
8	admin	Menyimpan data login admin
9	reset_password	Menyimpan token reset password untuk akun
10	profil_usaha	Menyimpan informasi profil dan lokasi salon
11	kritik_saran	Menyimpan kritik dan saran dari pelanggan

#### **3.4.2. Kebutuhan Perangkat Dalam Rancang Bangun Sistem**

Kebutuhan perangkat dalam rancang bangun sistem adalah spesifikasi perangkat keras dan perangkat lunak yang diperlukan untuk mendukung proses pembuatan, pengujian, dan pengembangan sistem agar berjalan lancar dan efektif. Hal ini penting agar seluruh tahapan pembangunan sistem dapat dilakukan tanpa kendala teknis, sehingga hasil yang diperoleh sesuai dengan tujuan penelitian.

**a. Kebutuhan Perangkat Keras**

**Tabel 3. 3 Kebutuhan Perangkat Keras**

Perangkat Keras	Spesifikasi	Keterangan
Laptop	Minimal RAM 4 GB	Digunakan untuk pengkodean dan testing
Penyimpanan	SSD 256 GB	Untuk menyimpan data dan program
Koneksi Internet	WiFi	Untuk akses internet selama pengembangan

**b. Kebutuhan Perangkat Lunak**

**Tabel 3. 4 Kebutuhan Perangkat Lunak**

Perangkat Lunak	Fungsi
Windows 10	Sistem operasi untuk menjalankan semua aplikasi pengembangan
XAMPP	Server lokal untuk menjalankan aplikasi web berbasis PHP dan MySQL
Visual Studio Code (VS Code)	Text editor untuk coding PHP, JavaScript, dan bahasa pemrograman lainnya
draw.io	Membuat diagram seperti UML dan flowchart sistem
Balsamiq	Membuat wireframe desain antarmuka
PHP & JavaScript	Bahasa pemrograman utama untuk backend dan frontend aplikasi
Bootstrap 5	Framework CSS untuk mempercepat desain antarmuka responsif

**3.5. Perancangan Sistem**

Perancangan sistem adalah tahap krusial dalam pengembangan perangkat lunak yang fokus pada desain arsitektur dan komponen utama dari sistem yang akan dibangun. Dalam konteks penelitian ini, rancangan sistem berfungsi untuk

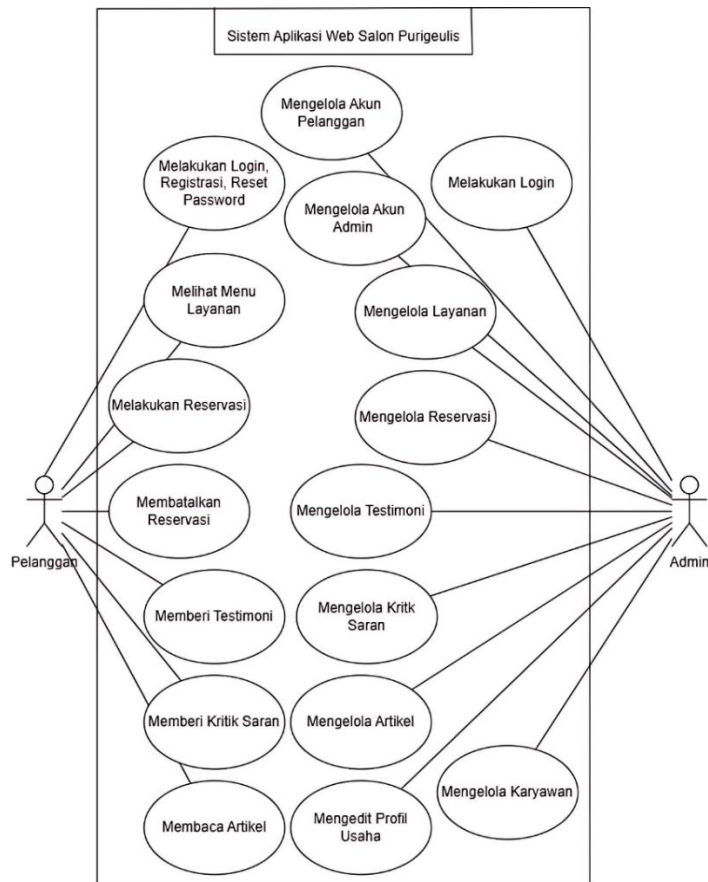
menjembatani analisis kebutuhan dengan implementasi teknis, memastikan solusi yang dikembangkan sesuai dengan tujuan dan kebutuhan yang telah ditentukan.

### **3.5.1. Perancangan Menggunakan UML**

Pada bagian ini dijelaskan pemodelan sistem menggunakan *Unified Modeling Language* (UML) yang berfungsi untuk menggambarkan struktur dan perilaku sistem secara visual. Diagram UML yang digunakan dalam perancangan sistem ini meliputi *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Deployment Diagram*, yang masing-masing merepresentasikan aspek yang berbeda dari sistem yang dibangun.

#### **a. *Use Case Diagram***

*Use Case Diagram* merupakan salah satu jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk memodelkan interaksi antara aktor (pengguna atau sistem eksternal) dengan sistem yang sedang dikembangkan. Diagram ini menggambarkan fungsionalitas sistem secara garis besar melalui representasi use case (kasus penggunaan) yang saling terhubung dengan aktor terkait.



**Gambar 3. 2** Use Case Diagram

Berikut ini merupakan rincian aktor, use case, dan relasi yang terdapat dalam sistem aplikasi web Salon Purigeulis. Informasi ini disusun berdasarkan diagram use case yang telah dirancang sebelumnya untuk menggambarkan interaksi antara pengguna dengan sistem.

**Tabel 3. 5** Aktor Dalam Sistem

Aktor	Definisi Aktor
Pelanggan	Pengguna yang melakukan registrasi, login, membuat dan membatalkan reservasi, memberikan testimoni, kritik/saran, serta membaca artikel.
Admin	Pengelola sistem yang bisa login dan mengelola data akun, layanan, reservasi, testimoni, kritik/saran, artikel, karyawan, dan profil usaha.

**Tabel 3. 6** *Relasi Aktor - Use Case*

Aktor	Use Case
Pelanggan	Melakukan Login, Registrasi, Reset Password
	Melihat Menu Layanan
	Melakukan Reservasi
	Membatalkan Reservasi
	Memberi Testimoni
	Memberi Kritik Saran
	Membaca Artikel
Admin	Melakukan Login
	Mengelola Akun Pelanggan
	Mengelola Akun Admin
	Mengelola Layanan
	Mengelola Reservasi
	Mengelola Testimoni
	Mengelola Kritik Saran
	Mengelola Artikel
	Mengelola Karyawan
	Mengedit Profil Usaha

Aktor yang terlibat dalam sistem ini terdiri dari *pelanggan* dan *admin*. *Pelanggan* memiliki beberapa *use case*, yaitu melakukan *login*, registrasi, dan *reset password* untuk mengakses sistem. Selain itu, *pelanggan* dapat melihat *menu layanan*, melakukan reservasi, serta membatalkan reservasi jika diperlukan. *Pelanggan* juga dapat memberikan *testimoni* dan mengisi *form kritik saran* sebagai masukan bagi pengelola. Selain itu, mereka dapat membaca *artikel* yang tersedia di aplikasi sebagai informasi tambahan.

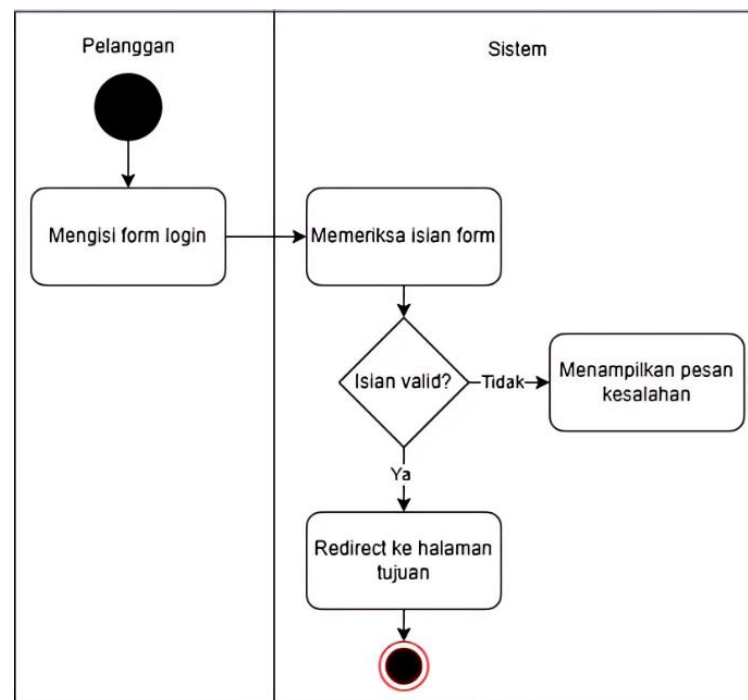
Sementara itu, *admin* memiliki hak akses lebih luas pada sistem. *Admin* dapat melakukan *login* untuk masuk ke halaman *dashboard* dan mengelola berbagai data. Beberapa *use case* yang dimiliki *admin* meliputi mengelola *akun pelanggan*, *akun admin*, *layanan*, *reservasi*, *testimoni*, *kritik saran*, *artikel*, serta data

*karyawan*. Selain itu, *admin* juga dapat melakukan pengeditan terhadap *profil usaha* agar informasi yang ditampilkan selalu terbaru.

## b. Activity Diagram

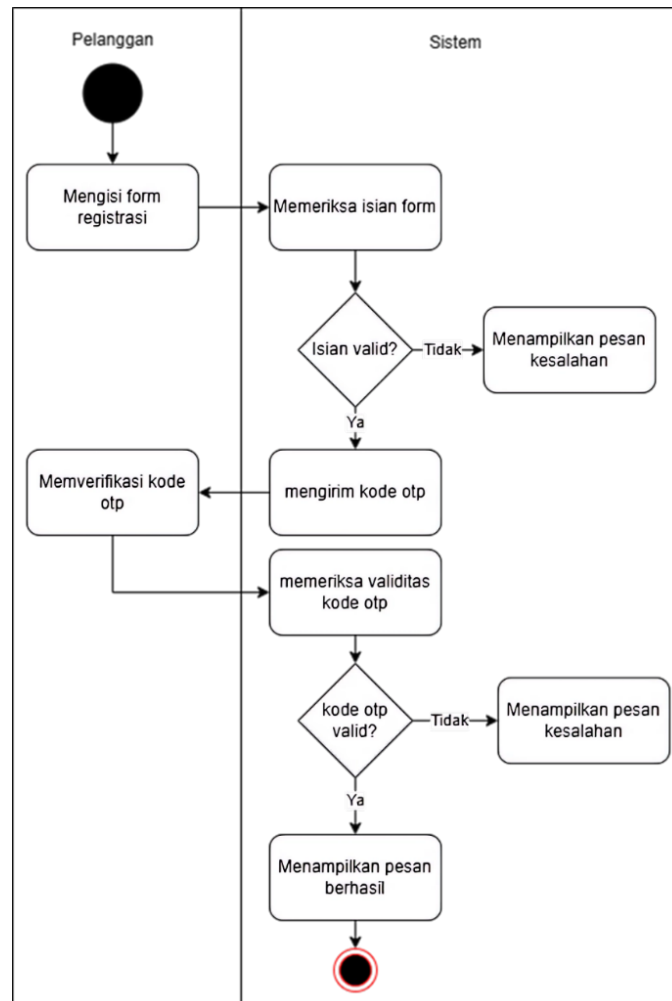
*Activity diagram* digunakan untuk menggambarkan alur kerja atau aktivitas dalam sistem, mulai dari kondisi awal hingga akhir suatu proses. Diagram ini menampilkan urutan aktivitas, keputusan, serta alur logika yang terjadi.

### 1. Activity Diagram Login, Registrasi, Reset Password Akun Pelanggan



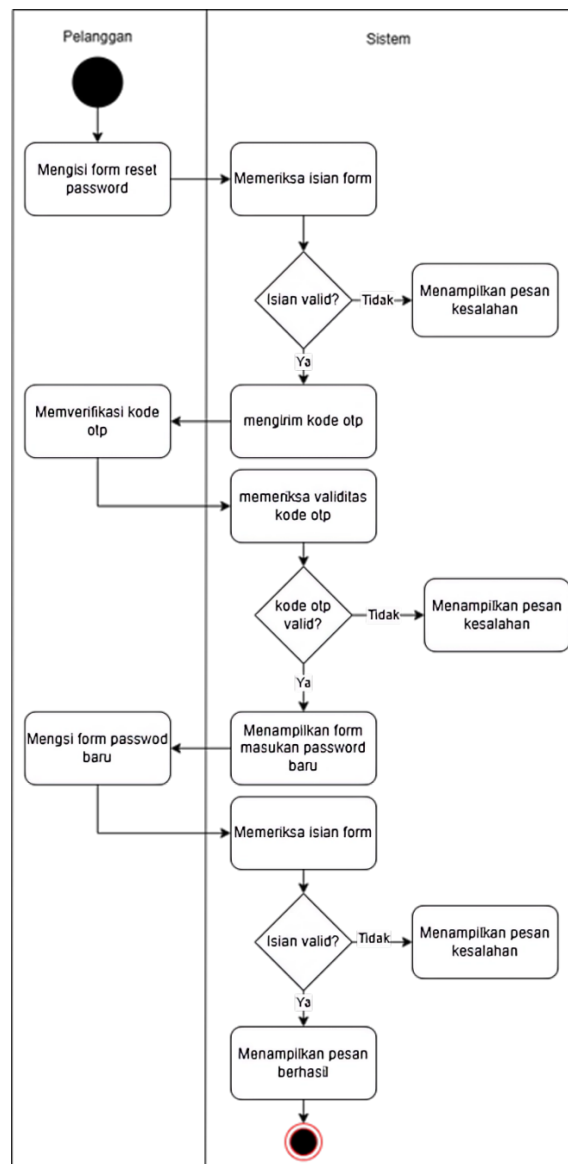
**Gambar 3. 3** Activity Diagram Login

Aktivitas *login* dimulai ketika pengguna mengisi form *login* dengan memasukkan *username* dan *password* pada halaman yang tersedia. Setelah *form* diisi, sistem akan memeriksa validitas data yang dimasukkan. Jika data valid, sistem akan mengarahkan (*redirect*) pengguna ke halaman tujuan sesuai dengan hak aksesnya. Namun, jika data tidak valid, sistem akan menampilkan pesan kesalahan kepada pengguna untuk memperbaiki isian *form*.



**Gambar 3. 4** Activity Diagram Registrasi

Proses *reset password* dimulai ketika pengguna mengisi *form* *reset password* dengan memasukkan email atau *username* yang terdaftar. Setelah itu, sistem memeriksa validitas data yang dimasukkan. Jika data valid, sistem mengirimkan kode OTP ke email atau nomor telepon pengguna. Selanjutnya, pengguna memasukkan kode OTP untuk diverifikasi oleh sistem. Jika kode OTP valid, sistem akan menampilkan *form* untuk memasukkan password baru. Setelah pengguna mengisi *form password* baru, sistem kembali memeriksa validitas data. Jika data sudah sesuai, sistem menyimpan password baru tersebut dan menampilkan pesan bahwa proses *reset password* berhasil. Sebaliknya, jika terjadi kesalahan pada tahap manapun, sistem akan menampilkan pesan kesalahan kepada pengguna

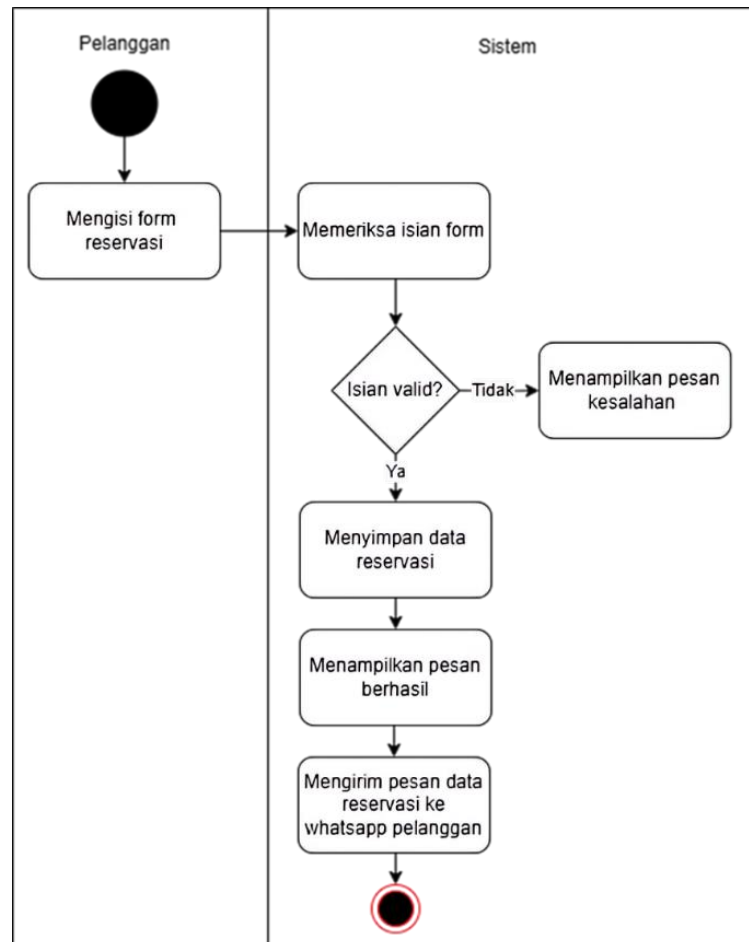


**Gambar 3.5** Activity Diagram Reset Password

Proses dimulai ketika pengguna mengisi *form* reset *password* dengan memasukkan email atau *username*. Sistem kemudian memeriksa validitas isian form tersebut. Jika isian valid, sistem mengirimkan kode OTP ke email atau nomor pengguna untuk verifikasi. Pengguna memasukkan kode OTP yang diterima, lalu sistem memeriksa apakah kode tersebut valid. Jika kode OTP valid, sistem menampilkan form untuk memasukkan password baru. Setelah pengguna mengisi form password baru, sistem kembali memvalidasi isian. Jika isian valid, sistem menyimpan password baru tersebut dan menampilkan pesan bahwa reset *password*

berhasil. Apabila pada salah satu tahap terjadi kesalahan, sistem akan menampilkan pesan error yang sesuai.

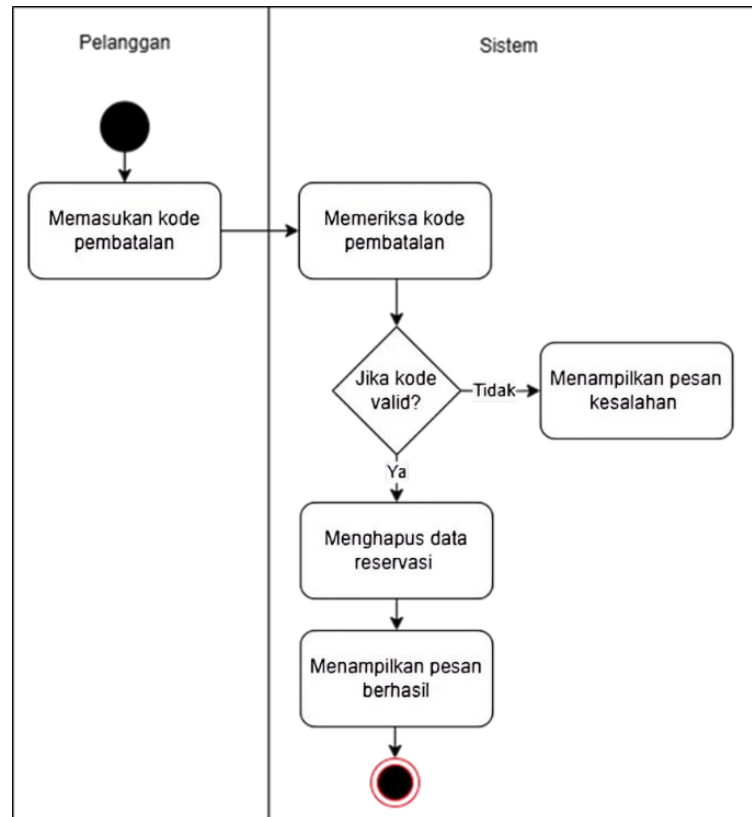
## 2. *Activity Diagram* Melakukan Reservasi



**Gambar 3. 6** *Activity Diagram* Proses Reservasi

Proses dimulai ketika pengguna mengisi *form reservasi* dengan data yang diperlukan. Selanjutnya, sistem memeriksa validitas isian *form*. Jika isian tidak valid, sistem akan menampilkan pesan kesalahan. Namun, jika isian valid, sistem menyimpan data reservasi ke *database*. Setelah itu, sistem menampilkan pesan bahwa reservasi berhasil dan mengirimkan detail data reservasi ke *WhatsApp* pelanggan sebagai notifikasi.

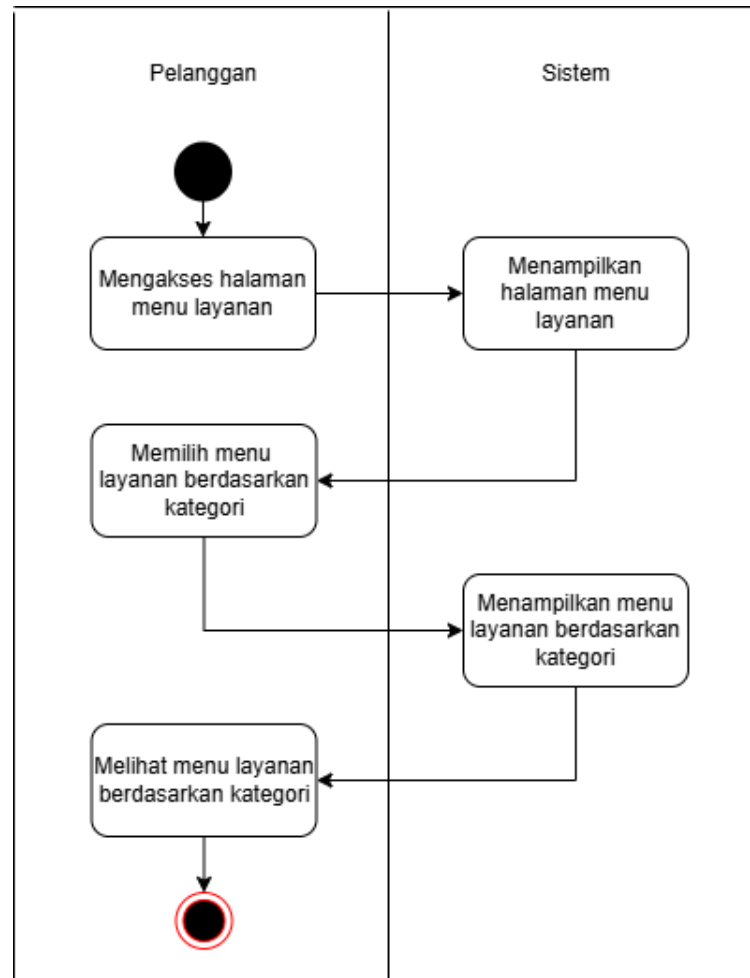
### 3. *Activity Diagram membatalkan Reservasi*



**Gambar 3. 7** *Activity Diagram Pembatalan Reservasi*

Proses dimulai ketika pengguna memasukkan *kode pembatalan* pada halaman yang tersedia. Sistem kemudian memeriksa validitas *kode pembatalan* tersebut. Jika *kode* tidak valid, sistem akan menampilkan pesan kesalahan. Namun, jika *kode* valid, sistem akan menghapus data reservasi dari *database* dan menampilkan pesan bahwa pembatalan berhasil dilakukan

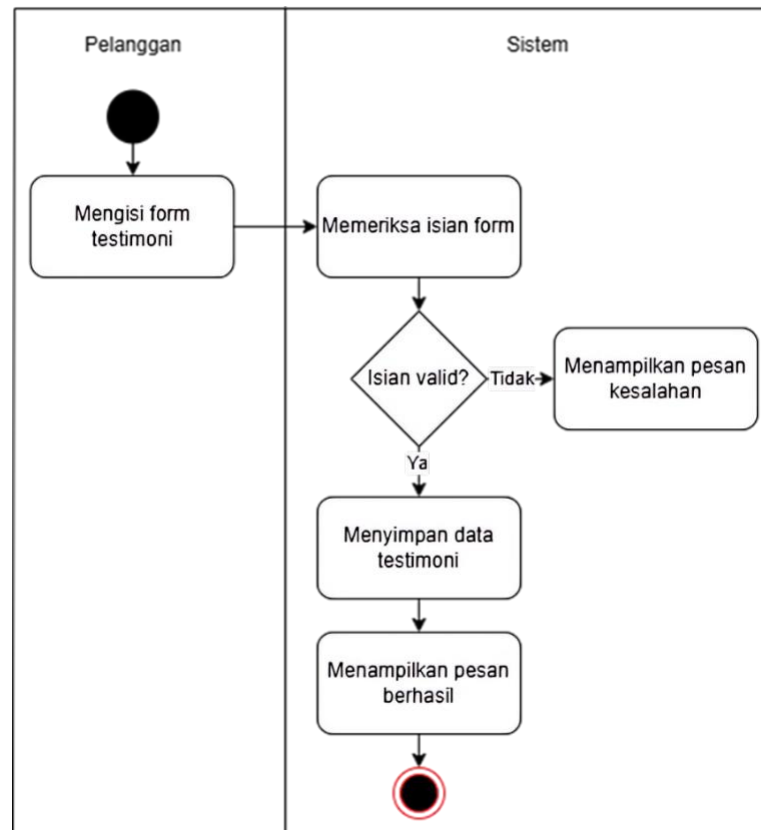
#### 4. Activity Diagram Melihat Menu Layanan



**Gambar 3. 8** Activity Diagram Melihat Menu Layanan

Proses diawali ketika *pelanggan* mengakses halaman *menu layanan*. Sistem kemudian menampilkan halaman *menu layanan* kepada pengguna. Selanjutnya, *pelanggan* memilih *menu layanan* berdasarkan kategori yang diinginkan, dan sistem akan menampilkan daftar *menu layanan* sesuai dengan kategori tersebut. Akhirnya, *pelanggan* dapat melihat daftar *menu layanan* berdasarkan kategori yang dipilih.

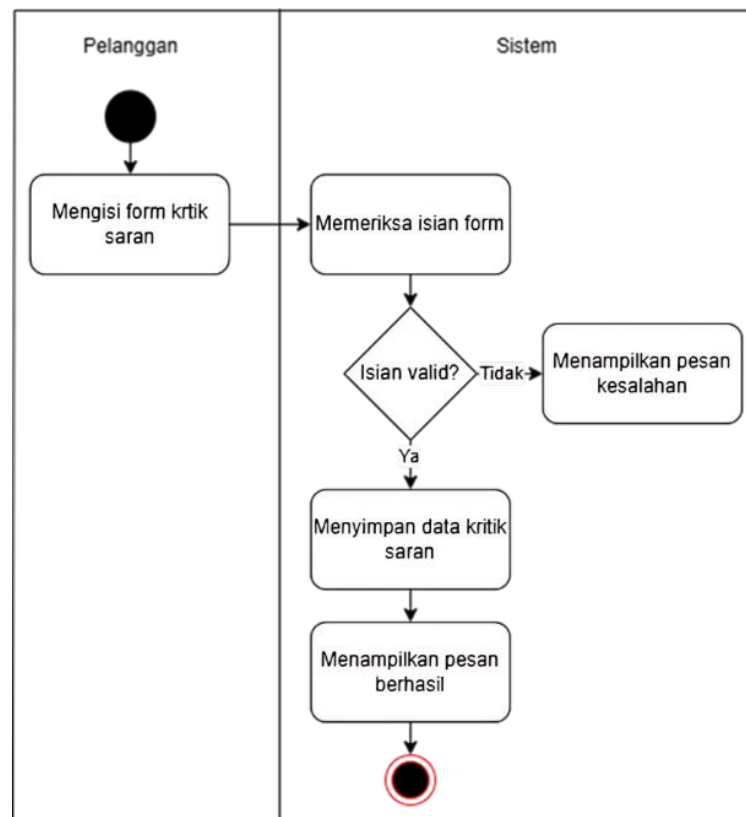
## 5. Activity Diagram Memberi Testimoni



**Gambar 3.9** Activity Diagram Memberi Testimoni

Proses dimulai ketika *pelanggan* mengisi *form testimoni* pada aplikasi. Setelah itu, sistem memeriksa validitas isian *form*. Jika isian tidak valid, sistem akan menampilkan pesan kesalahan. Namun, jika isian valid, sistem menyimpan data *testimoni* ke dalam *database* dan menampilkan pesan bahwa pengisian *testimoni* berhasil

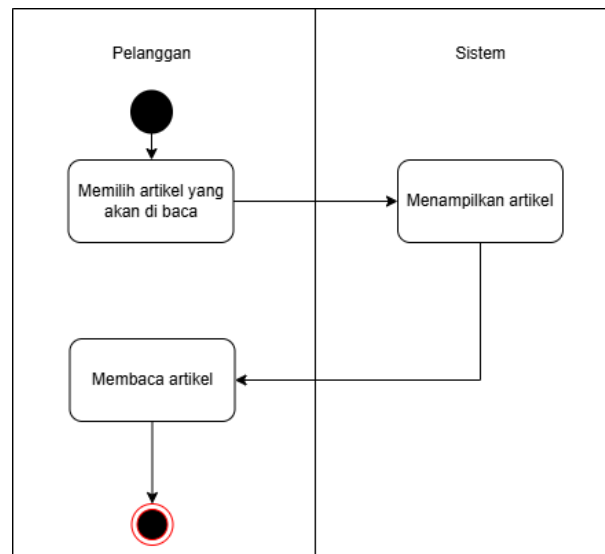
## 6. Activity Diagram Memberi Kritik & Saran



**Gambar 3. 10** Activity Diagram Memberi Kritik dan Saran

Proses dimulai ketika *pelanggan* mengisi *form kritik saran* pada aplikasi. Sistem kemudian memeriksa validitas isian *form*. Jika isian tidak valid, sistem akan menampilkan pesan kesalahan. Namun, jika isian valid, sistem akan menyimpan data *kritik saran* ke dalam *database* dan menampilkan pesan bahwa pengisian berhasil.

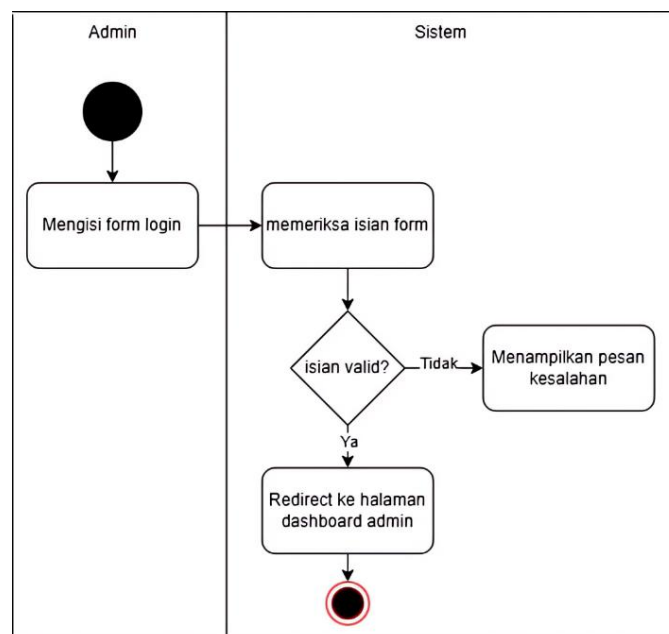
### 7. Activity Diagram Membaca Artikel



**Gambar 3. 11** Activity Diagram Membaca Artikel

Proses diawali ketika *pelanggan* memilih *artikel* yang ingin dibaca pada aplikasi. Sistem kemudian menampilkan isi *artikel* yang dipilih. Selanjutnya, *pelanggan* dapat membaca *artikel* tersebut hingga selesai.

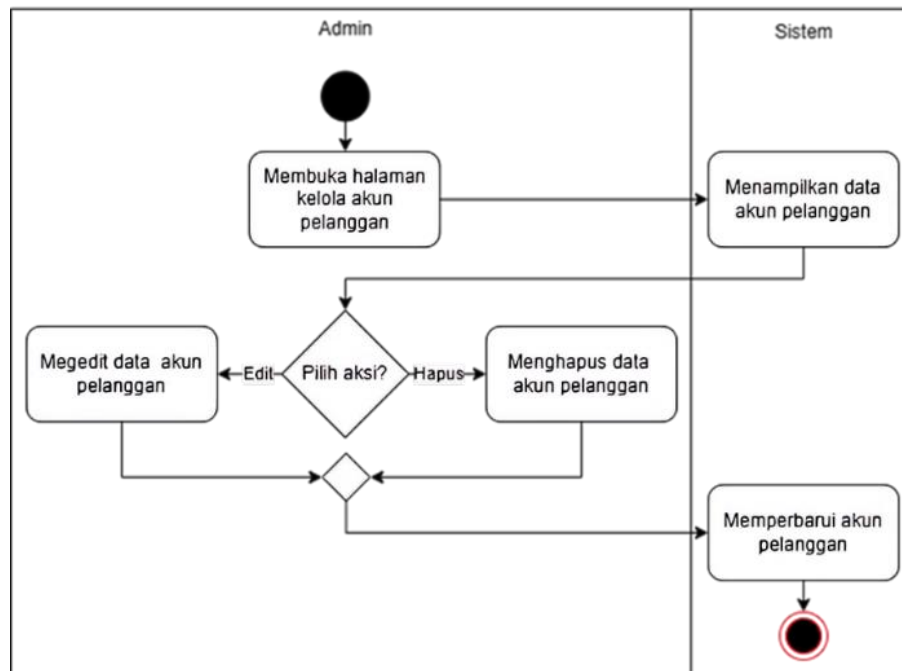
### 8. Activity Diagram Melakukan Login Admin



**Gambar 3. 12** Activity Diagram Melakukan login Admin

Proses dimulai ketika *admin* mengisi *form login* dengan *username* dan *password*. Sistem kemudian memeriksa validitas isian *form*. Jika isian tidak valid, sistem akan menampilkan pesan kesalahan. Namun, jika isian valid, sistem akan melakukan *redirect* ke halaman *dashboard admin*.

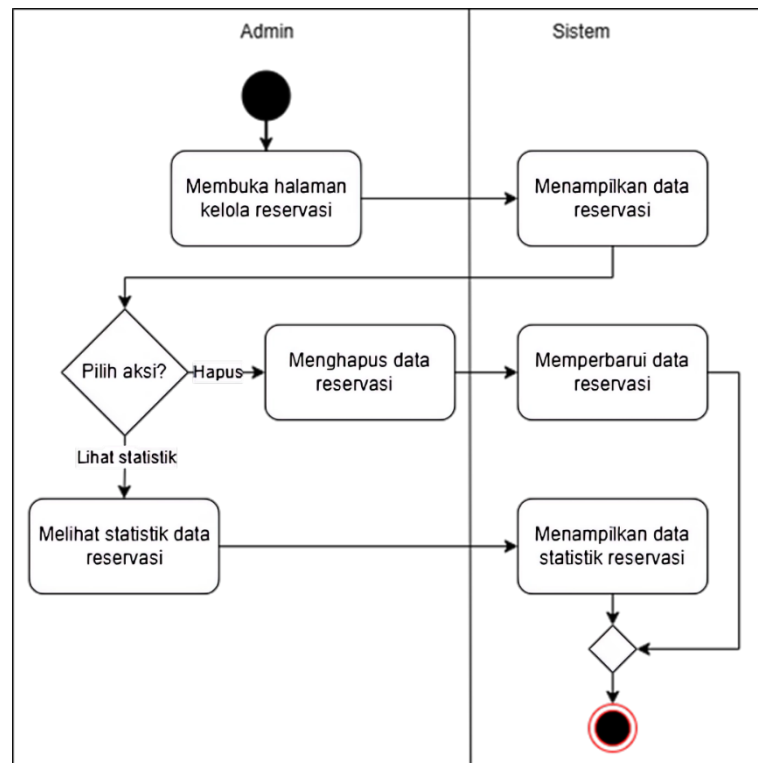
### 9. Activity Diagram Mengelola Akun Pelanggan



**Gambar 3. 13** Activity Diagram Mengelola Akun Pelanggan

Proses diawali ketika *admin* membuka halaman *kelola akun pelanggan*. Sistem kemudian menampilkan data *akun pelanggan* yang ada. *Admin* dapat memilih aksi, yaitu *edit* untuk mengubah data atau *hapus* untuk menghapus data. Jika memilih *edit*, sistem akan memperbarui *akun pelanggan* sesuai perubahan yang dilakukan. Jika memilih *hapus*, sistem akan menghapus data *akun pelanggan* dari *database*.

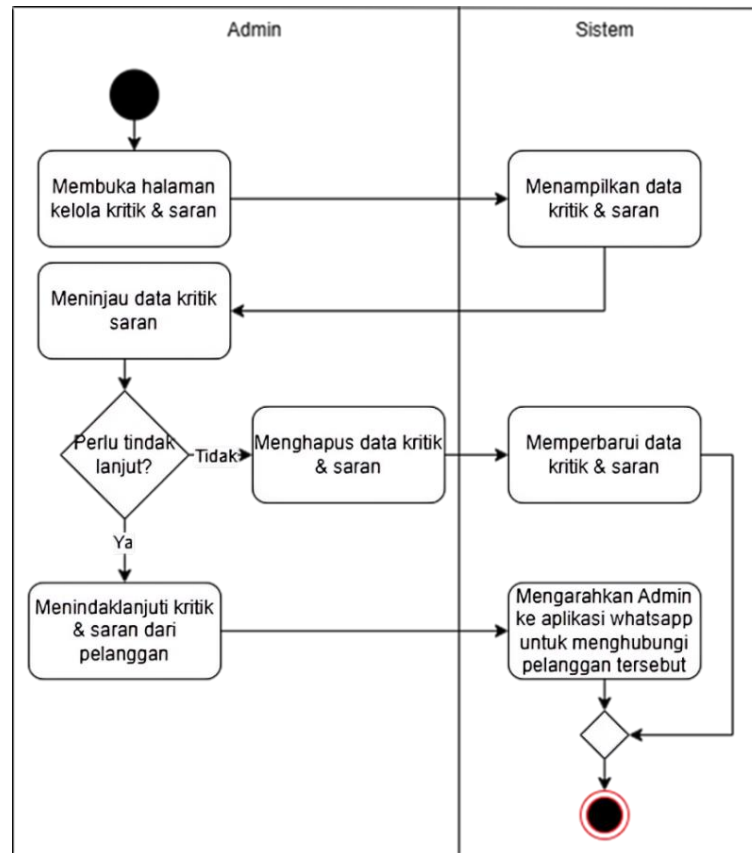
## 10. Activity Diagram Mengelola Reservasi



**Gambar 3. 14** Activity Diagram Mengelola Reservasi

Proses diawali dengan pengguna membuka halaman kelola reservasi, lalu sistem menampilkan data reservasi. Pengguna kemudian dapat memilih untuk menghapus atau memperbarui data, atau melihat dan menampilkan statistik reservasi, sebelum proses berakhir.

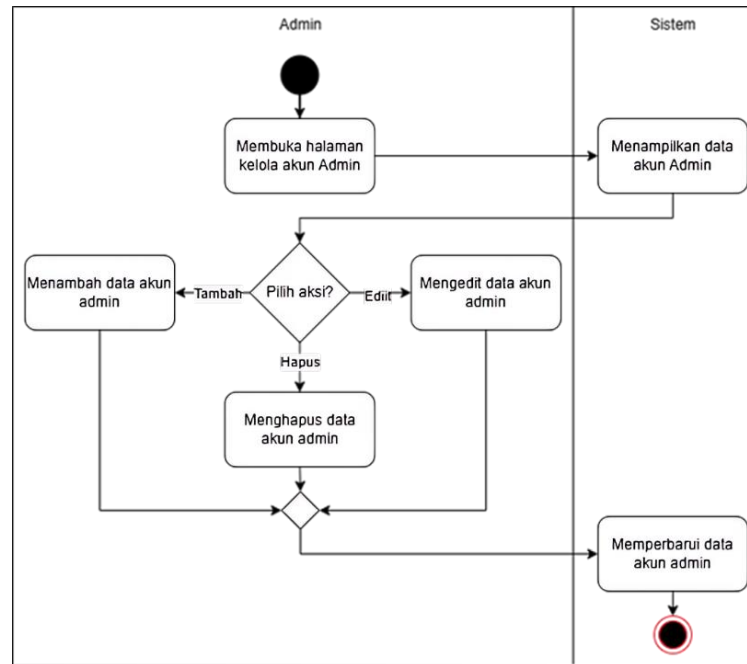
### 11. Activity Diagram Mengelola Kritik & Saran



**Gambar 3. 15** Activity Diagram Mengelola Kritik & Saran

Proses pengelolaan kritik dan saran dimulai dengan membuka halaman khusus dan menampilkan data kritik serta saran yang telah masuk. Selanjutnya, data tersebut akan ditinjau untuk menentukan apakah memerlukan tindak lanjut. Apabila tidak ada tindak lanjut yang diperlukan, data kritik dan saran tersebut dapat dihapus. Namun, jika memang perlu ditindaklanjuti, kritik dan saran dari pelanggan akan diproses, yang dapat mencakup pengarahannya ke aplikasi *WhatsApp* untuk menghubungi pelanggan terkait. Di samping itu, sistem juga menyediakan fungsionalitas untuk memperbarui data kritik dan saran yang ada. Seluruh proses ini akan berakhir setelah tindakan yang dipilih selesai dilaksanakan.

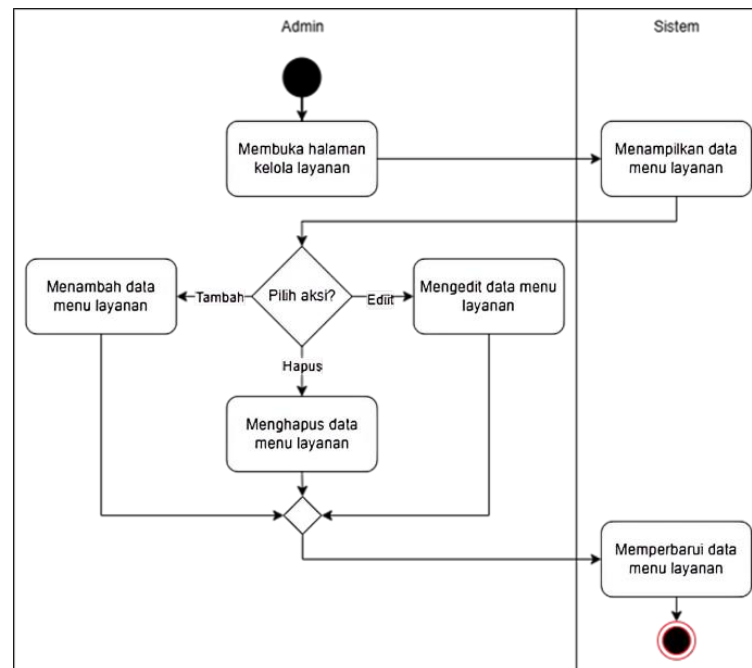
## 12. Activity Diagram Mengelola Akun Admin



**Gambar 3. 16** Activity Diagram Mengelola Akun

Proses pengelolaan akun *admin* dimulai dengan membuka halaman *kelola* dan menampilkan data akun *admin*. Pengguna kemudian dapat memilih untuk menambah, mengedit, atau menghapus data akun *admin*. Selain itu, tersedia juga opsi untuk memperbarui data akun *admin*. Proses ini berakhir setelah aksi yang dipilih selesai.

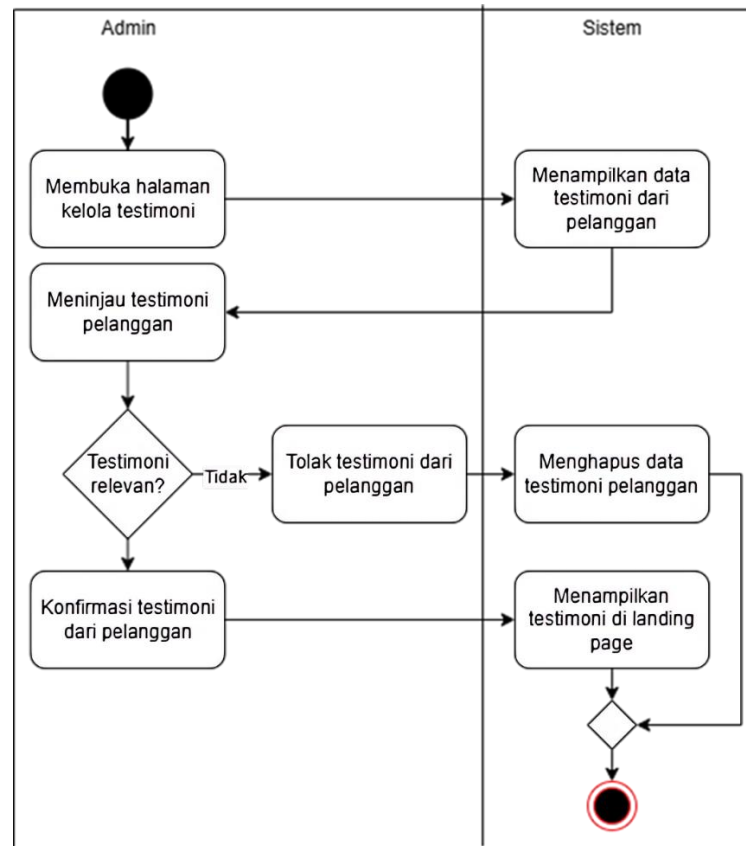
### 13. Activity Diagram Mengelola Layanan



**Gambar 3. 17** Activity Diagram Mengelola Layanan

Proses pengelolaan menu layanan dimulai dengan membuka halaman *kelola* layanan dan menampilkan data menu layanan. Pengguna kemudian dapat memilih aksi untuk menambah, mengedit, atau menghapus data menu layanan. Selain itu, terdapat juga opsi untuk memperbarui data menu layanan. Proses ini berakhir setelah aksi yang dipilih selesai.

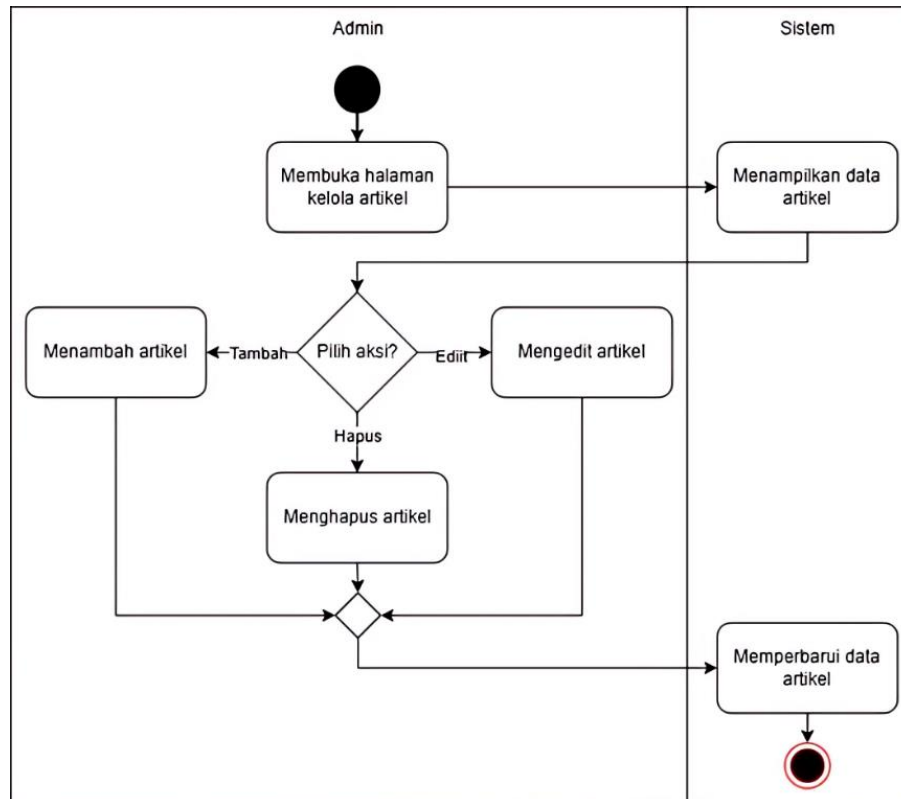
#### 14. Activity Diagram Mengelola Testimoni



**Gambar 3. 18** Activity Diagram Mengelola Testimoni

Proses pengelolaan testimoni dimulai dengan membuka halaman *kelola* dan menampilkan data testimoni dari pelanggan. Testimoni kemudian ditinjau untuk menentukan apakah relevan. Jika tidak relevan, testimoni akan ditolak dari pelanggan dan dapat dihapus. Jika relevan, testimoni akan dikonfirmasi dari pelanggan dan ditampilkan di *landing page*. Proses ini berakhir setelah

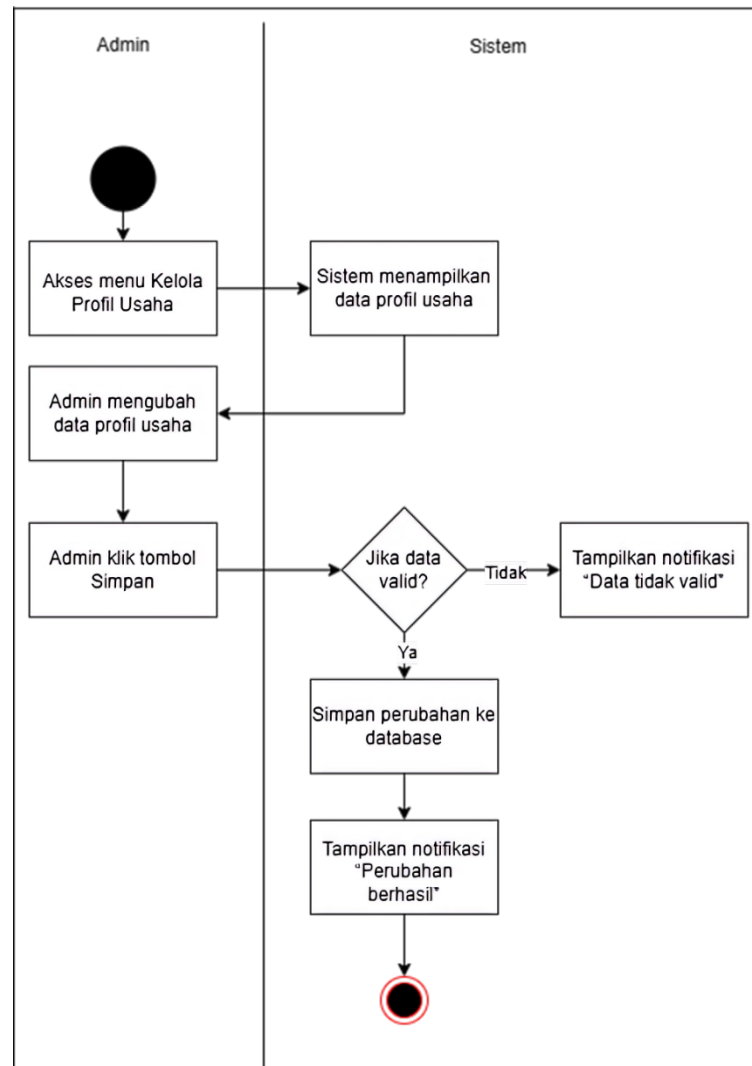
### 15. Activity Diagram Mengelola Artikel



**Gambar 3. 19** Activity Diagram Mengelola Artikel

Proses pengelolaan artikel dimulai ketika *admin* membuka halaman *kelola* artikel, yang secara otomatis akan menampilkan data artikel. *Admin* kemudian dapat memilih aksi untuk menambah, mengedit, atau menghapus artikel. Setelah salah satu aksi tersebut, data artikel dapat diperbarui. Proses ini berakhir setelah pembaruan data artikel.

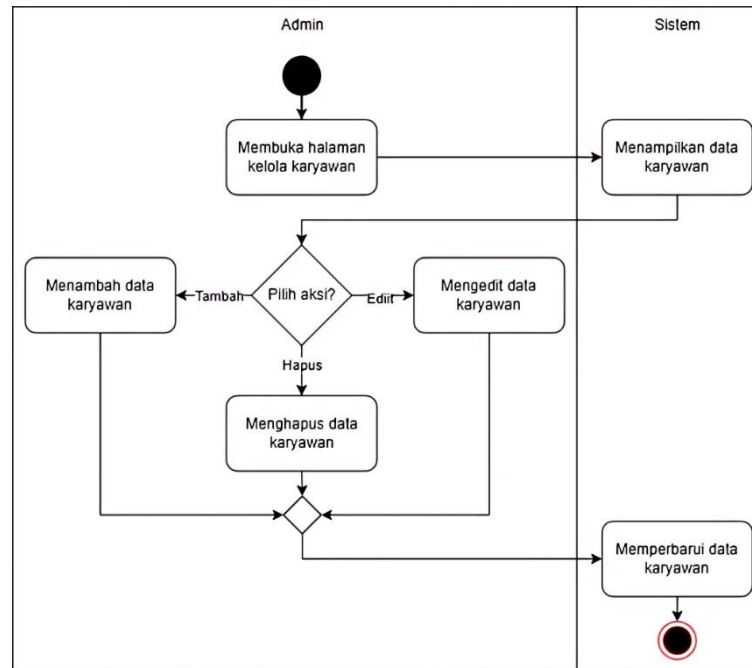
## 16. Activity Diagram Mengelola Profil Usaha



**Gambar 3. 20** Activity Diagram Mengelola Profil Usaha

Proses pengelolaan profil usaha dimulai ketika *admin* mengakses menu *kelola* profil usaha, dan sistem akan menampilkan data profil usaha. *Admin* kemudian dapat mengubah data tersebut dan mengklik tombol "Simpan". Sistem akan memvalidasi data; jika tidak *valid*, notifikasi "Data tidak *valid*" akan ditampilkan. Jika data *valid*, perubahan akan disimpan ke *database*, dan notifikasi "Perubahan berhasil" akan ditampilkan, mengakhiri proses.

### 17. Activity Diagram Mengelola Karyawan



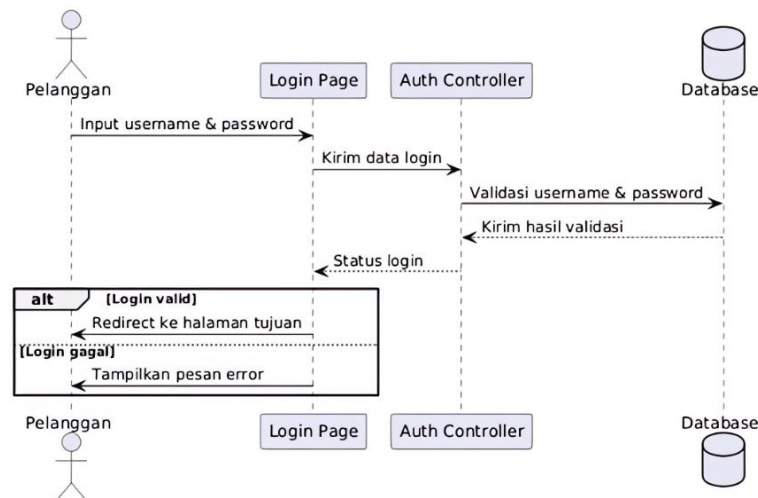
**Gambar 3. 21** Activity Diagram Mengelola Karyawan

Proses pengelolaan karyawan dimulai ketika *admin* membuka halaman *kelola* karyawan, yang kemudian sistem akan menampilkan data karyawan. *Admin* dapat memilih aksi untuk menambah, mengedit, atau menghapus data karyawan. Setelah salah satu aksi tersebut, data karyawan dapat diperbarui. Proses ini berakhir setelah pembaruan data karyawan.

### c. *Sequence Diagram*

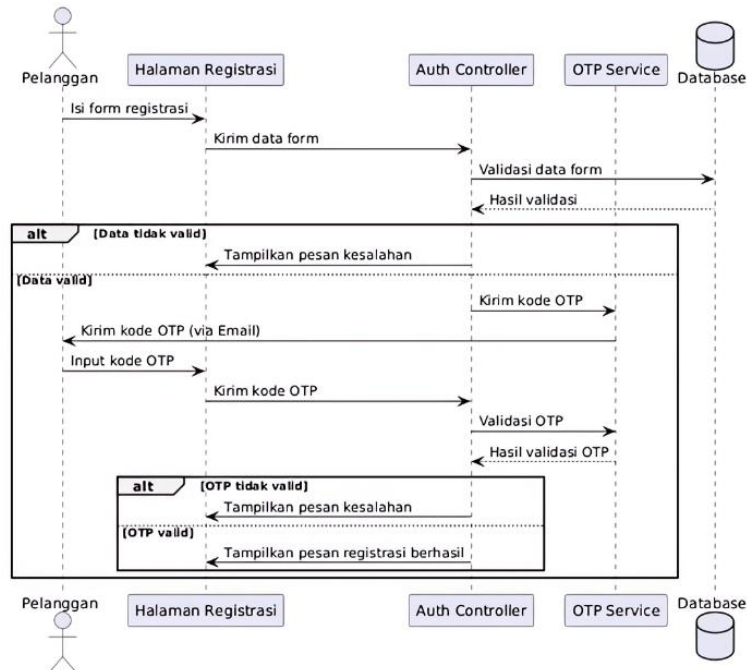
*Sequence diagram* digunakan untuk menggambarkan alur interaksi antar objek dalam sistem secara runtut berdasarkan waktu. Diagram ini menjelaskan bagaimana proses dimulai, pesan yang dikirim antar objek, serta urutan eksekusi dari setiap proses.

#### 1. *Sequence Diagram Login, Registrasi, Reset Password Akun Pelanggan*



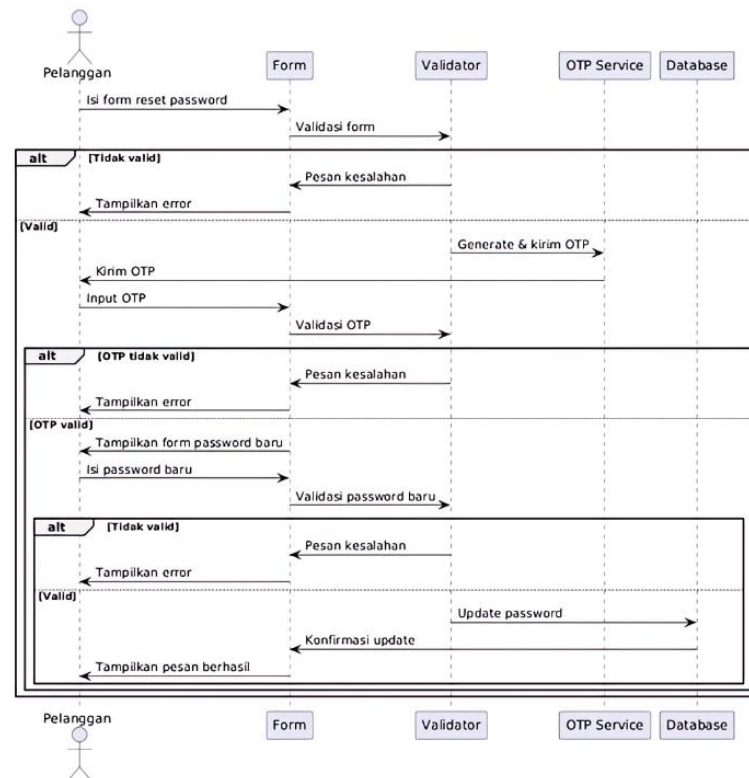
**Gambar 3. 22** Sequence Diagram Login Akun Pelanggan

Proses *login* pelanggan dimulai ketika pelanggan menginput *username* dan *password* pada *Login Page*. Data tersebut dikirim ke *Auth Controller* yang kemudian memvalidasi *username* dan *password* dengan *Database*. Hasil validasi dikirim kembali ke *Auth Controller*, lalu status *login* dikirim ke *Login Page*. Jika *login valid*, pelanggan akan di-*redirect* ke halaman tujuan. Namun, jika *login gagal*, *Login Page* akan menampilkan pesan *error* kepada pelanggan.



**Gambar 3. 23** Sequence Diagram Registrasi Akun Pelanggan

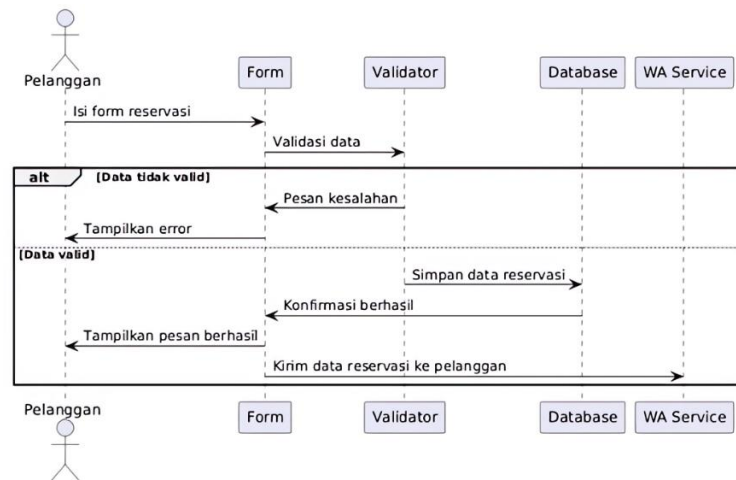
Proses registrasi pelanggan diawali dengan pelanggan mengisi formulir pada Halaman Registrasi dan mengirim data. *Auth Controller* memvalidasi data dengan *Database*. Jika data tidak *valid*, pesan kesalahan ditampilkan. Jika data *valid*, *Auth Controller* mengirim kode *OTP* melalui *OTP Service* ke email pelanggan. Pelanggan menginput kode *OTP*, yang kemudian divalidasi oleh *Auth Controller*. Jika *OTP* tidak *valid*, pesan kesalahan ditampilkan. Jika *OTP valid*, pesan registrasi berhasil ditampilkan kepada pelanggan.



**Gambar 3. 24** Sequence Diagram Reset Password Akun Pelanggan

Proses *reset password* dimulai ketika pelanggan mengisi formulir *reset password*. *Form* akan divalidasi; jika tidak *valid*, pesan *error* ditampilkan. Jika *valid*, kode *OTP* akan di-*generate* dan dikirim melalui *OTP Service* ke pelanggan. Pelanggan menginput kode *OTP* yang kemudian divalidasi. Jika *OTP* tidak *valid*, pesan *error* ditampilkan. Jika *OTP valid*, form *password* baru akan ditampilkan. Pelanggan mengisi *password* baru, yang divalidasi. Jika tidak *valid*, pesan *error* ditampilkan. Jika *valid*, *password* akan di-*update* di *database*, konfirmasi *update* dikirim, dan pesan berhasil ditampilkan, mengakhiri proses.

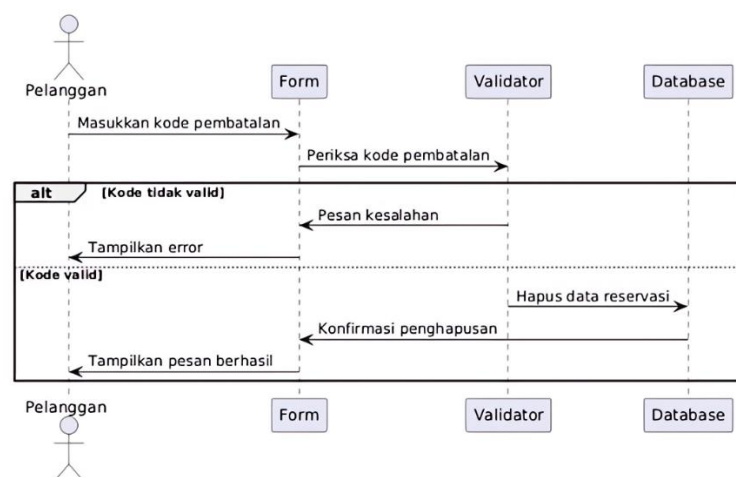
## 2. *Sequence Diagram Melakukan Reservasi*



**Gambar 3. 25** *Sequence Diagram Melakukan Reservasi*

Proses reservasi pelanggan dimulai ketika pelanggan mengisi formulir reservasi yang kemudian divalidasi. Jika data tidak *valid*, pesan kesalahan dan *error* akan ditampilkan. Jika data *valid*, data reservasi akan disimpan ke *database*, dan konfirmasi berhasil akan diterima. Setelah itu, pesan berhasil akan ditampilkan kepada pelanggan, dan data reservasi akan dikirimkan ke pelanggan melalui *WA Service*

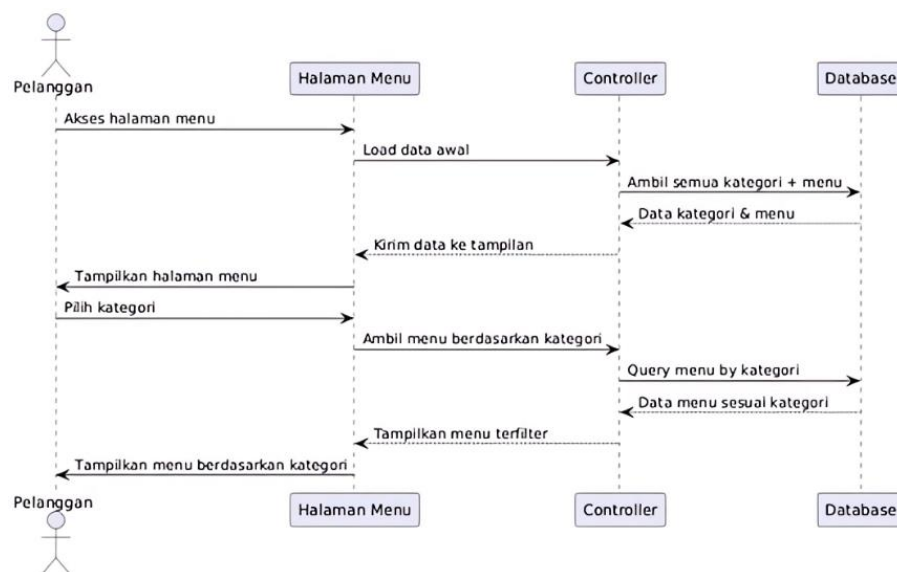
## 3. *Sequence Diagram Membatalkan Reservasi*



**Gambar 3. 26** *Sequence Diagram Membatalkan Reservasi*

Proses pembatalan reservasi pelanggan dimulai ketika pelanggan memasukkan kode pembatalan pada *Form*, yang kemudian akan diperiksa validasinya. Jika kode tidak *valid*, pesan kesalahan dan *error* akan ditampilkan. Jika kode *valid*, data reservasi akan dihapus dari *Database*, konfirmasi penghapusan akan diterima, dan pesan berhasil akan ditampilkan kepada pelanggan. Proses ini berakhir setelah pesan berhasil ditampilkan.

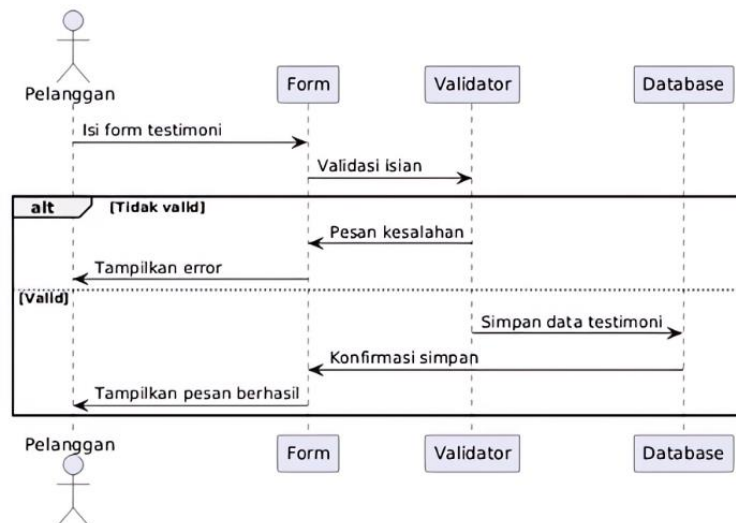
#### 4. *Sequence Diagram* Melihat Menu Layanan



**Gambar 3. 27** *Sequence Diagram* Melihat Menu Layanan

Proses akses halaman menu dimulai ketika pelanggan mengakses halaman menu. Sistem akan memuat data awal dengan mengambil semua kategori dan menu dari *Database* melalui *Controller*, kemudian mengirimkan data tersebut untuk ditampilkan di halaman menu. Pelanggan dapat memilih kategori, yang akan memicu pengambilan menu berdasarkan kategori dari *Database* melalui *Controller*. Selanjutnya, menu yang terfilter akan ditampilkan di halaman menu.

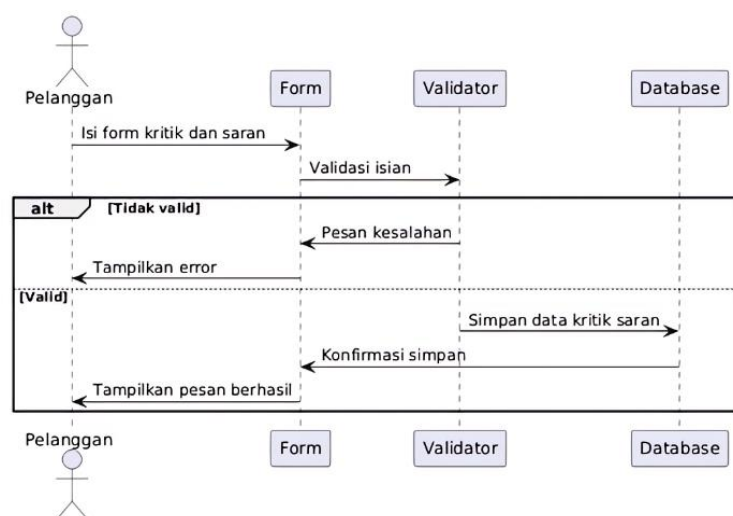
## 5. *Sequence Diagram Memberi Testimoni*



**Gambar 3. 28** *Sequence Diagram Memberi Testimoni*

Proses pengisian testimoni dimulai ketika pelanggan mengisi formulir testimoni, yang kemudian di-*validasi* isiannya. Jika isian tidak *valid*, pesan kesalahan dan *error* akan ditampilkan. Jika isian *valid*, data testimoni akan disimpan ke *Database*, konfirmasi simpan akan diterima, dan pesan berhasil akan ditampilkan kepada pelanggan.

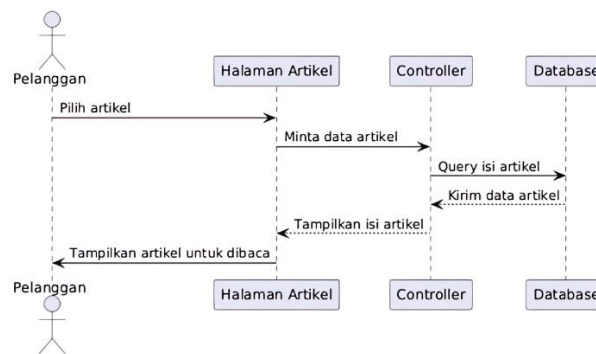
## 6. *Sequence Diagram Memberi Kritik & Saran*



**Gambar 3. 29** *Sequence Diagram Memberi Kritik & Saran*

Proses pengisian kritik dan saran dimulai ketika pelanggan mengisi formulir, yang kemudian di-*validasi* isiannya. Jika isian tidak *valid*, pesan kesalahan dan *error* akan ditampilkan. Jika isian *valid*, data kritik dan saran akan disimpan ke *Database*, konfirmasi simpan akan diterima, dan pesan berhasil akan ditampilkan kepada pelanggan.

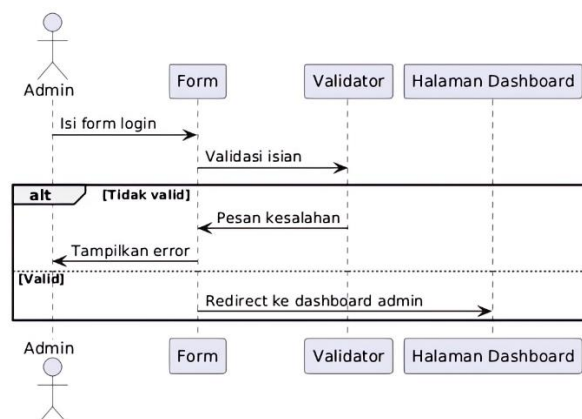
### 7. *Sequence Diagram Membaca Artikel*



**Gambar 3. 30** *Sequence Diagram Membaca Artikel*

Proses akses artikel dimulai ketika pelanggan memilih artikel dari Halaman Artikel. Halaman Artikel kemudian meminta data artikel dari *Controller*, yang akan melakukan *query* isi artikel dari *Database*. Setelah data artikel dikirim kembali ke *Controller*, isi artikel akan ditampilkan di Halaman Artikel. Akhirnya, artikel akan ditampilkan kepada pelanggan untuk dibaca.

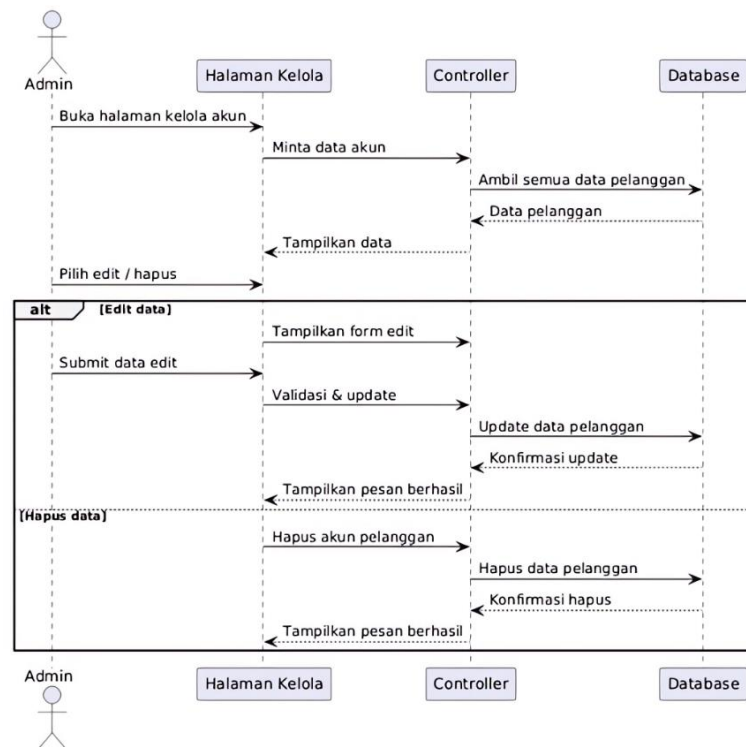
### 8. *Sequence Diagram Melakukan Login Admin*



**Gambar 3. 31** *Sequence Diagram Melakukan Login Admin*

Proses *login admin* dimulai ketika *admin* mengisi formulir *login*, yang kemudian di-*validasi* isiannya. Jika isian tidak *valid*, pesan kesalahan dan *error* akan ditampilkan. Jika isian *valid*, *admin* akan di-*redirect* langsung ke halaman *dashboard admin*.

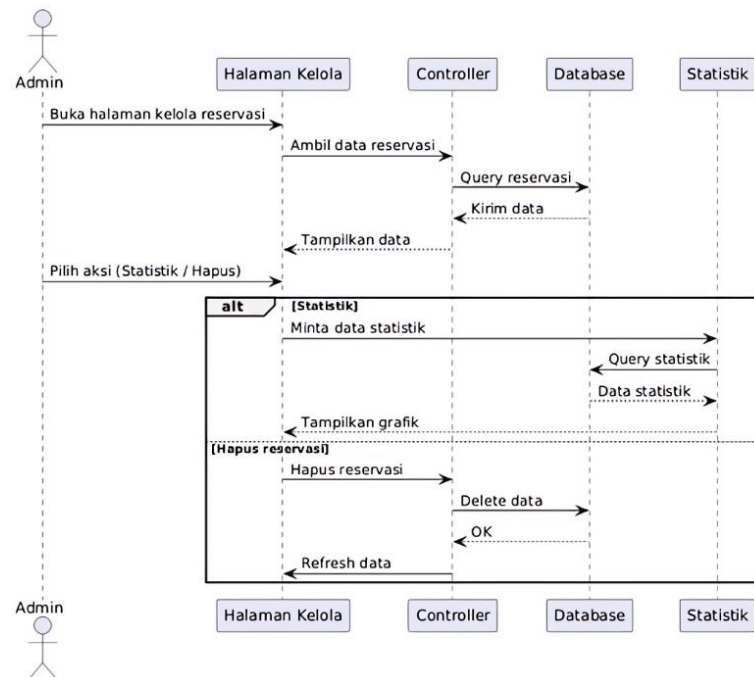
## 9. Sequence Diagram Mengelola Akun Pelanggan



**Gambar 3. 32** Sequence Diagram Mengelola Akun Pelanggan

Proses pengelolaan akun pelanggan dimulai ketika *admin* membuka halaman *kelola* akun, yang kemudian meminta dan menampilkan semua data pelanggan dari *Database*. *Admin* dapat memilih untuk mengedit data (menampilkan *form* edit, *submit*, memvalidasi, memperbarui di *Database*, dan menampilkan pesan berhasil) atau menghapus data (menghapus akun dan data pelanggan dari *Database*, lalu menampilkan pesan berhasil).

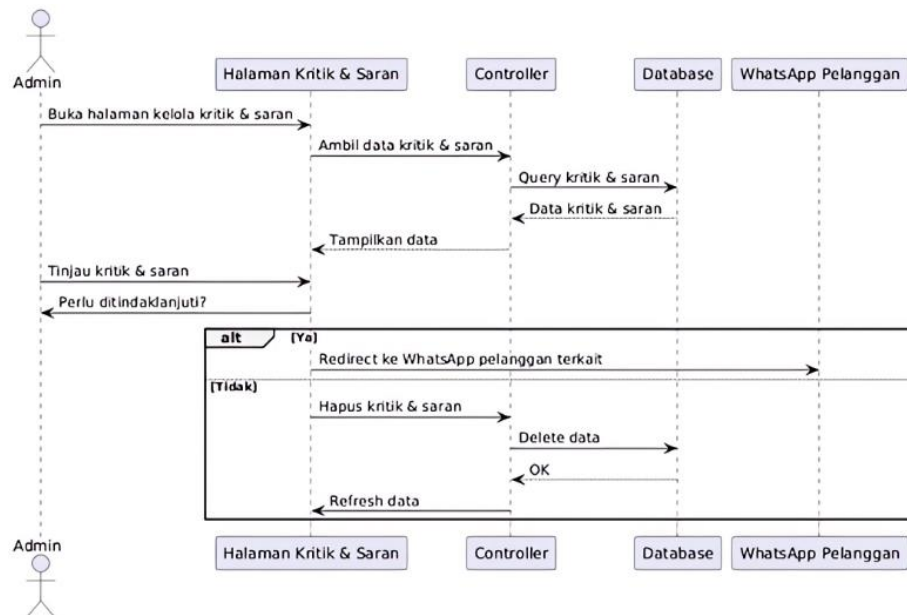
## 10. Sequence Diagram Mengelola Reservasi



**Gambar 3.33** Sequence Diagram Mengelola Reservasi

Proses pengelolaan reservasi oleh *admin* dimulai dengan *admin* membuka halaman *kelola* reservasi. Sistem akan mengambil data reservasi dari *Database* melalui *Controller* dan menampilkannya. *Admin* dapat memilih aksi: jika "Statistik", sistem akan meminta data statistik, melakukan *query* ke *Database*, dan menampilkan *grafik*; jika "Hapus Reservasi", sistem akan menghapus reservasi dari *Database* dan me-*refresh* data, kemudian menampilkan pesan berhasil.

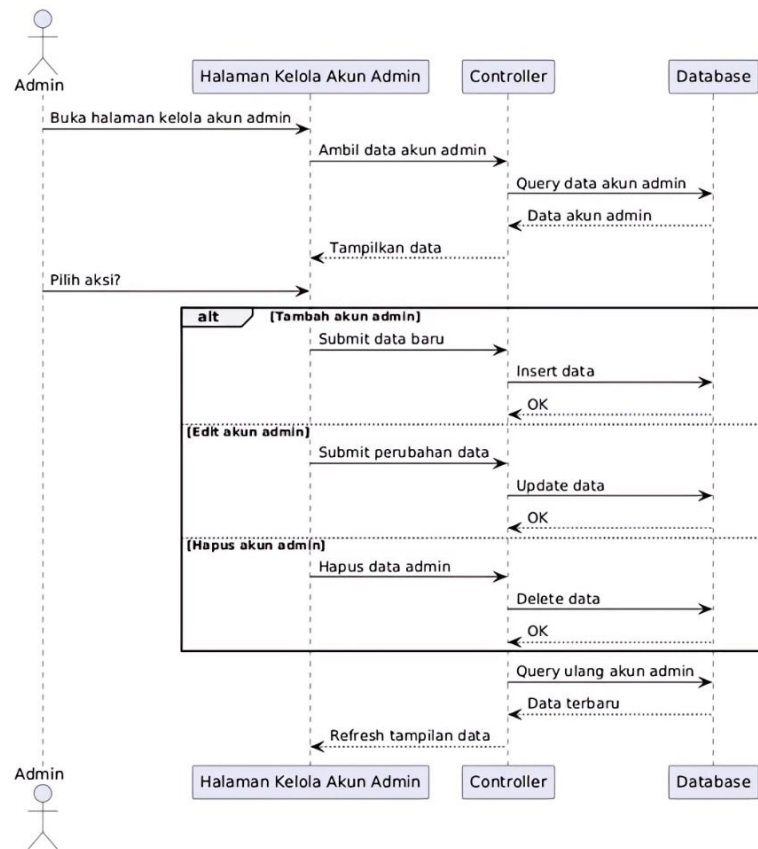
## 11. Sequence Diagram Mengelola Kritik & Saran



**Gambar 3. 34** Sequence Diagram Mengelola Kritik & Saran

Proses pengelolaan kritik dan saran oleh *admin* dimulai dengan *admin* membuka halaman *kelola* kritik & saran, yang akan mengambil dan menampilkan data kritik & saran dari *Database* melalui *Controller*. *Admin* kemudian meninjau kritik & saran tersebut untuk menentukan apakah perlu ditindaklanjuti. Jika "Ya", *admin* akan di-*redirect* ke WhatsApp pelanggan terkait. Jika "Tidak", kritik & saran akan dihapus dari *Database* dan data akan di-*refresh*.

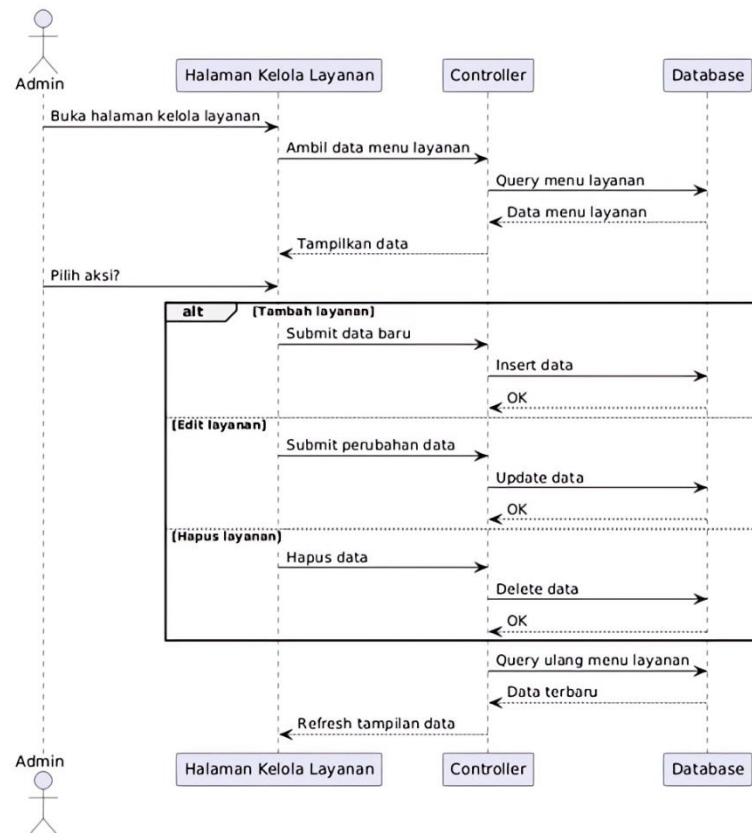
## 12. Sequence Diagram Mengelola Akun Admin



**Gambar 3. 35** Sequence Diagram Mengelola Akun Admin

Proses pengelolaan akun *admin* dimulai ketika *admin* membuka halaman *kelola* akun *admin*, yang akan mengambil dan menampilkan data akun *admin* dari *Database* melalui *Controller*. *Admin* dapat memilih aksi: jika "Tambah akun *admin*", data baru akan di-*submit* dan di-*insert* ke *Database*; jika "Edit akun *admin*", perubahan data akan di-*submit*, divalidasi, dan di-*update* di *Database*; jika "Hapus akun *admin*", data akan dihapus dari *Database*. Setelah setiap aksi ini, sistem akan mengkonfirmasi (OK), me-*query* ulang data, dan me-*refresh* tampilan data.

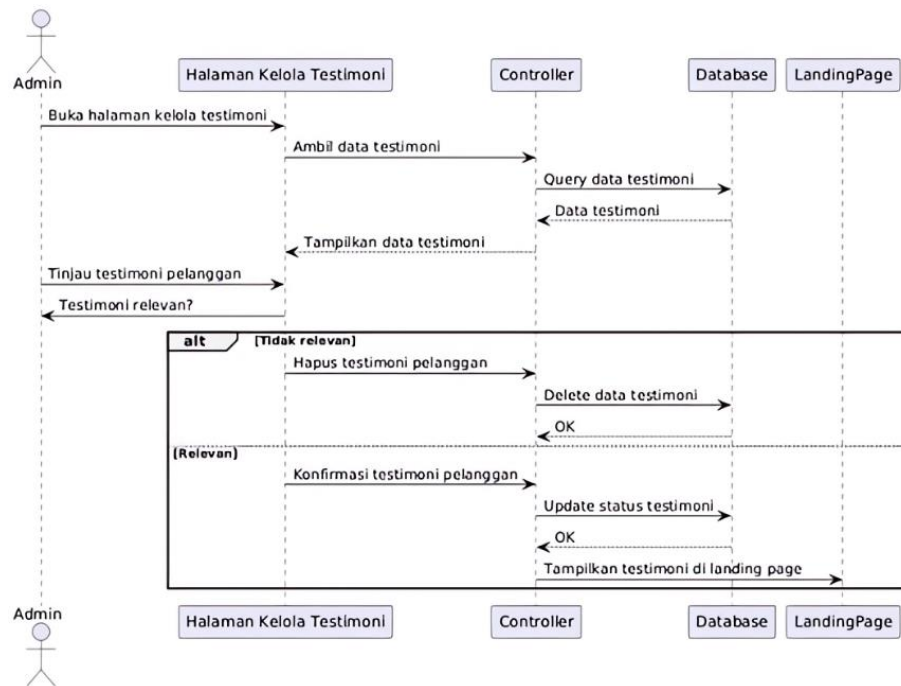
### 13. Sequence Diagram Mengelola Layanan



**Gambar 3. 36** Sequence Diagram Mengelola Layanan

Proses pengelolaan menu layanan oleh *admin* dimulai dengan *admin* membuka halaman *kelola* layanan, yang akan mengambil dan menampilkan data menu layanan dari *Database* melalui *Controller*. *Admin* dapat memilih aksi: jika "Tambah layanan", data baru akan di-*submit* dan di-*insert* ke *Database*; jika "Edit layanan", perubahan data akan di-*submit*, divalidasi, dan di-*update* di *Database*; jika "Hapus layanan", data akan dihapus dari *Database*. Setelah setiap aksi ini, sistem akan mengkonfirmasi (OK), me-*query* ulang data, dan me-*refresh* tampilan data.

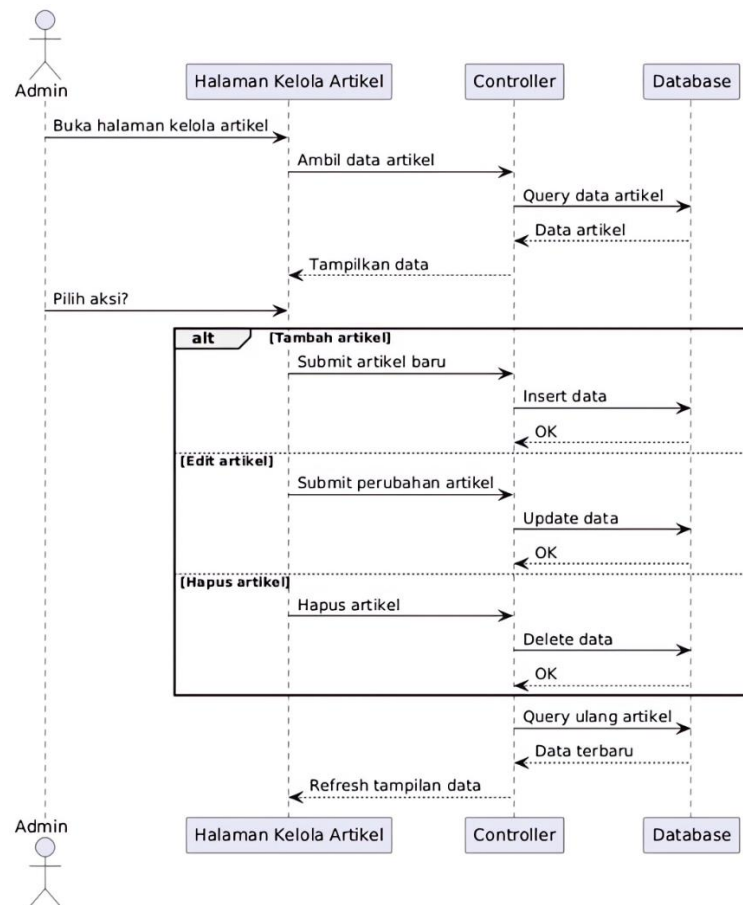
#### 14. Sequence Diagram Mengelola Testimoni



**Gambar 3. 37** Sequence Diagram Mengelola Testimoni

Proses pengelolaan testimoni dimulai ketika *admin* membuka halaman *kelola* testimoni. Sistem akan mengambil data testimoni dari *Database* melalui *Controller* dan menampilkannya. *Admin* kemudian meninjau testimoni pelanggan untuk menentukan apakah relevan. Jika "Tidak relevan", testimoni akan dihapus dari *Database*. Jika "Relevan", testimoni akan dikonfirmasi, statusnya diperbarui di *Database*, dan kemudian ditampilkan di *Landing Page*.

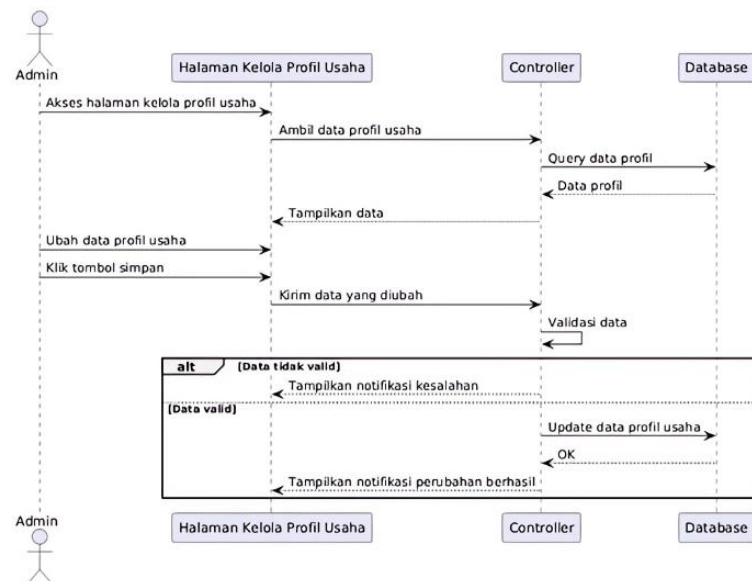
## 15. Sequence Diagram Mengelola Artikel



**Gambar 3. 38** Sequence Diagram Mengelola Artikel

Proses pengelolaan artikel dimulai ketika *admin* membuka halaman *kelola* artikel. Sistem akan mengambil dan menampilkan data artikel dari *Database* melalui *Controller*. *Admin* dapat memilih aksi: jika "Tambah artikel", artikel baru akan di-*submit* dan di-*insert* ke *Database*; jika "Edit artikel", perubahan artikel akan di-*submit*, divalidasi, dan di-*update* di *Database*; jika "Hapus artikel", artikel akan dihapus dari *Database*. Setelah setiap aksi ini, sistem akan mengkonfirmasi (OK), me-*query* ulang data, dan me-*refresh* tampilan data.

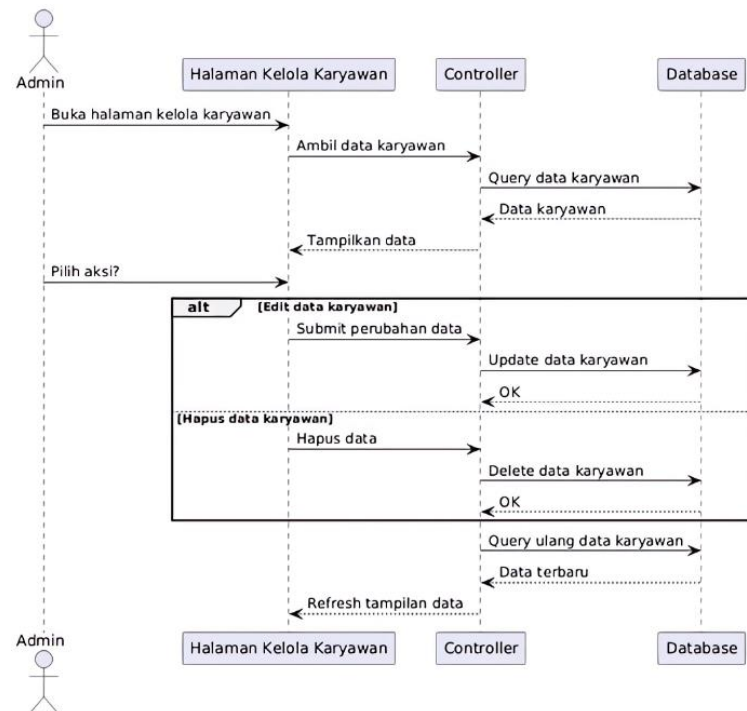
## 16. Sequence Diagram Mengelola Profil Usaha



**Gambar 3. 39** Sequence Diagram Mengelola Profil Usaha

Proses pengelolaan profil usaha dimulai ketika *admin* mengakses menu *kelola* profil usaha, dan sistem akan menampilkan data profil usaha. *Admin* kemudian dapat mengubah data tersebut dan mengklik tombol "Simpan". Sistem akan memvalidasi data; jika tidak *valid*, notifikasi kesalahan akan ditampilkan. Jika data *valid*, perubahan akan diperbarui di *database*, dan notifikasi "Perubahan berhasil" akan ditampilkan, mengakhiri proses.

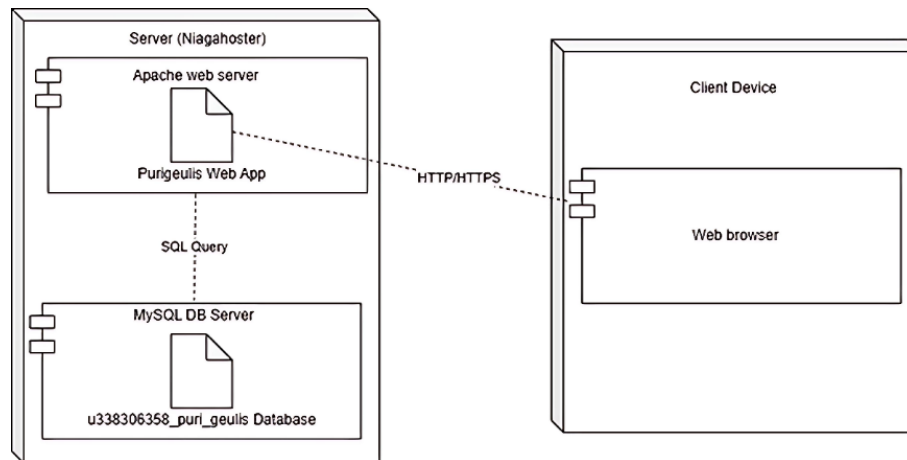
## 17. Sequence Diagram Mengelola Karyawan



**Gambar 3. 40** Sequence Diagram Mengelola Karyawan

Proses pengelolaan karyawan oleh *admin* dimulai dengan *admin* membuka halaman *kelola* karyawan, yang akan mengambil dan menampilkan data karyawan dari *Database* melalui *Controller*. *Admin* dapat memilih aksi: jika "Edit data karyawan", perubahan data akan di-*submit*, divalidasi, dan di-*update* di *Database*; jika "Hapus data karyawan", data akan dihapus dari *Database*. Setelah setiap aksi ini, sistem akan mengkonfirmasi (OK), me-*query* ulang data, dan me-*refresh* tampilan data.

d. **Deployment Diagram**



**Gambar 3. 41** *Deployment Diagram*

Deployment diagram pada **Gambar 3. 41** menggambarkan arsitektur fisik dari sistem Puriguelis Web App yang diimplementasikan menggunakan layanan hosting dari Niagahoster. Diagram ini menunjukkan bagaimana komponen-komponen perangkat lunak *dideploy* ke dalam *node* fisik dan bagaimana interaksi antar komponen tersebut terjadi.

*Node* pertama adalah *Server (Niagahoster)*, yang terdiri dari dua komponen utama, yaitu:

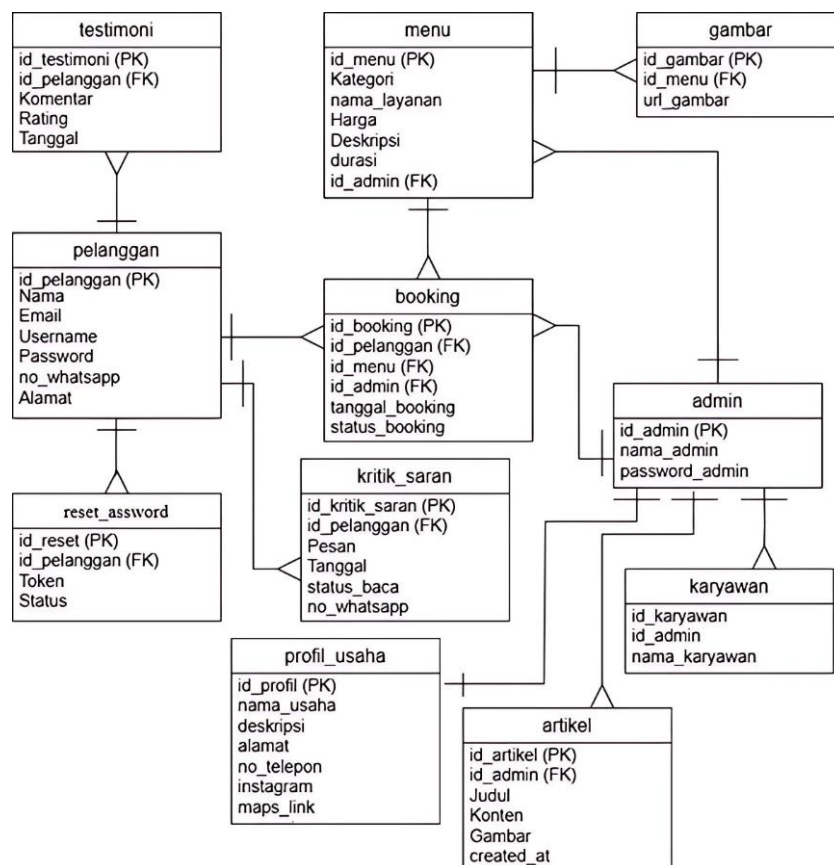
1. **Apache Web Server:** Bertanggung jawab untuk menjalankan aplikasi web *Puriguelis Web App*. Komponen ini mengelola permintaan dari pengguna dan mengatur alur proses layanan seperti reservasi, manajemen pelanggan, serta pengelolaan data lainnya.
2. **MySQL Database Server:** Berfungsi sebagai penyimpanan utama sistem. Seluruh data operasional disimpan dalam sebuah database bernama `u338306358_puri_geulis`, yang dapat diakses oleh aplikasi melalui komunikasi berbasis SQL.

Kedua komponen ini saling terhubung melalui komunikasi internal menggunakan *SQL Query*. Sementara itu, komunikasi antara server dan pengguna dilakukan melalui protokol HTTP/HTTPS.

*Node* kedua adalah **Client Device**, yang merepresentasikan perangkat pengguna seperti komputer, laptop, maupun smartphone. Perangkat ini menggunakan web *browser* untuk mengakses aplikasi yang berjalan di *server*. Karena aplikasi bersifat *web-based*, maka tidak memerlukan instalasi tambahan di sisi *client*.

### 3.5.2. ERD (Entity Relationship Diagram)

*Entity Relationship Diagram* (ERD) merupakan diagram yang digunakan untuk memodelkan hubungan antar entitas dalam sistem basis data. ERD membantu dalam memahami struktur data dan keterkaitan antar entitas yang digunakan dalam pengembangan aplikasi. Pada penelitian ini, ERD dibuat untuk menggambarkan hubungan antar entitas dalam sistem reservasi dan manajemen layanan salon kecantikan Purigeulis by REDDG Majalengka.



**Gambar 3. 42** ERD (Entity Relationship Diagram)

### a. Entitas Pelanggan

Entitas Pelanggan merepresentasikan pengguna yang melakukan reservasi layanan salon. Setiap pelanggan memiliki *id\_pelanggan* sebagai *Primary Key* yang unik, serta atribut lainnya seperti *nama*, *email*, *username*, *password*, *no\_whatsapp*, dan *alamat* untuk autentikasi dan komunikasi.

**Tabel 3. 7** Entitas Pelanggan

Nama Atribut	Tipe Data	Keterangan
id_pelanggan	INT (PK)	Primary Key, ID unik pelanggan
Nama	VARCHAR	Nama pelanggan
Email	VARCHAR	Email pelanggan
Username	VARCHAR	Username pelanggan
Password	VARCHAR	Password pelanggan
no_whatsapp	VARCHAR	Nomor WhatsApp pelanggan
Alamat	TEXT	Alamat pelanggan

Entitas Pelanggan memiliki hubungan *One-to-Many* dengan entitas lain, yaitu *Booking*, *Testimoni*, *Kritik & Saran*, dan *Reset Password*. Ini menunjukkan bahwa satu pelanggan dapat memiliki banyak catatan terkait reservasi, ulasan, masukan, dan permintaan *reset password*.

**Tabel 3. 8** Relasi Entitas Pelanggan Dengan Entitas Lainnya

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Booking	One-to-Many	id_pelanggan di Booking	Seorang pelanggan dapat melakukan lebih dari satu reservasi layanan.
Testimoni	One-to-Many	id_pelanggan di Testimoni	Pelanggan dapat memberikan lebih dari satu testimoni terkait layanan yang digunakan.
kritik_Saran	One-to-Many	id_pelanggan di Kritik & Saran	Pelanggan dapat mengirim lebih dari satu kritik & saran kepada salon.
reset Password	One-to-Many	id_pelanggan di Reset Password	Jika mengalami kendala login, pelanggan dapat melakukan lebih dari satu permintaan reset password.

**b. Entitas *Booking***

Entitas *Booking* menyimpan informasi mengenai reservasi layanan salon yang dilakukan pelanggan. Setiap *booking* memiliki *id\_booking* sebagai Primary Key, *tanggal\_booking* untuk mencatat waktu pemesanan, dan *status\_booking* yang menyimpan status reservasi (misalnya: *pending*, *confirmed*, *canceled*).

**Tabel 3. 9** *Entitas Booking*

Nama Atribut	Tipe Data	Keterangan
id_booking	INT (PK)	Primary Key, ID unik untuk setiap booking
id_pelanggan	INT (FK)	Foreign Key, menghubungkan ke entitas Pelanggan
id_menu	INT (FK)	Foreign Key, menghubungkan ke entitas Menu
id_admin	INT (FK)	Foreign Key, menghubungkan ke entitas Admin
tanggal_booking	DATE	Menyimpan tanggal reservasi dilakukan
status_booking	VARCHAR	Menyimpan status reservasi (misalnya: pending, confirmed, canceled)

Entitas *Booking* memiliki tiga *Foreign Key*: *id\_pelanggan* yang mengarah ke entitas *Pelanggan*, *id\_menu* yang mengarah ke entitas *Menu*, dan *id\_admin* yang merujuk ke entitas *Admin* yang menangani proses *booking*. Ini memungkinkan sistem untuk melacak siapa yang melakukan reservasi, layanan apa yang dipesan, dan admin yang bertugas dalam mengelola reservasi.

**Tabel 3. 10** *Relasi Entitas Booking Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Pelanggan	Many-to-One	id_pelanggan	Setiap booking dilakukan oleh satu pelanggan, tetapi satu pelanggan dapat memiliki banyak booking.
Menu	Many-to-One	id_menu	Setiap booking terkait dengan satu layanan dari Menu, tetapi satu layanan dapat dipesan pada banyak booking.
Admin	Many-to-One	id_admin	Setiap booking dapat dikelola atau dikonfirmasi oleh satu admin, tetapi satu admin dapat mengelola banyak booking.

### c. Entitas Menu

Entitas *Menu* mewakili layanan-layanan yang ditawarkan oleh salon kepada pelanggan. Setiap layanan memiliki *id\_menu* sebagai *Primary Key* dan atribut seperti *kategori*, *nama\_layanan*, *harga*, *deskripsi*, dan *durasi*, yang memberi gambaran lengkap mengenai layanan tersebut, termasuk harga, waktu pengerjaan, dan deskripsi rinci.

**Tabel 3. 11** *Entitas Menu*

Nama Atribut	Tipe Data	Keterangan
id_menu	INT (PK)	Primary Key, ID unik untuk setiap layanan
Kategori	VARCHAR	Kategori layanan (misalnya: Hair, Spa, Nail)
nama_layanan	VARCHAR	Nama layanan salon yang tersedia
Harga	DECIMAL	Harga layanan yang ditawarkan
Deskripsi	TEXT	Deskripsi layanan secara detail
Durasi	INT	Estimasi waktu layanan dalam menit
id_admin	INT (FK)	Foreign Key, menghubungkan ke entitas Admin yang menambahkan atau mengelola layanan

Entitas *Menu* memiliki beberapa relasi penting, di antaranya relasi dengan *Booking* bersifat *One-to-Many*, di mana satu layanan dapat dipesan dalam banyak transaksi *booking*. Relasi dengan *Gambar* juga bersifat *One-to-Many*, yang berarti satu layanan dapat memiliki banyak gambar pendukung. Relasi dengan *Admin* bersifat *Many-to-One*, di mana setiap layanan dikelola oleh satu admin, namun satu admin dapat mengelola banyak layanan.

**Tabel 3. 12** *Relasi Entitas Menu Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
booking	One-to-Many	id_menu di Booking	Satu layanan dalam menu dapat dipesan dalam banyak transaksi booking.
gambar	One-to-Many	id_menu di Gambar	Satu layanan dapat memiliki banyak gambar pendukung untuk ditampilkan kepada pelanggan.
admin	Many-to-One	id_admin di Menu	Setiap layanan ditambahkan atau dikelola oleh satu admin, namun satu admin dapat mengelola banyak layanan.

**d. Entitas Gambar**

Entitas *Gambar* digunakan untuk menyimpan informasi visual terkait layanan salon yang ditampilkan di sistem. Setiap gambar memiliki *id\_gambar* sebagai *Primary Key* dan *url\_gambar* yang menyimpan tautan atau lokasi penyimpanan *file* gambar, memungkinkan gambar ditampilkan dinamis di halaman layanan.

**Tabel 3. 13** *Entitas Gambar*

Nama Atribut	Tipe Data	Keterangan
id_gambar	INT (PK)	Primary Key, ID unik untuk setiap gambar
id_menu	INT (FK)	Foreign Key, menghubungkan ke entitas Menu
url_gambar	VARCHAR	Tautan atau lokasi penyimpanan gambar layanan

Entitas *Gambar* memiliki relasi *Many-to-One* dengan *Menu*, di mana satu layanan dapat memiliki banyak gambar pendukung.

**Tabel 3. 14** *Relasi Entitas Gambar Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Menu	Many-to-One	id_menu Gambar di	Satu layanan dalam menu dapat memiliki banyak gambar pendukung.

e. **Entitas Testimoni**

Entitas *Testimoni* menyimpan ulasan pelanggan setelah menggunakan layanan salon. Identitas unik tiap ulasan disimpan dalam *id\_testimoni* sebagai *Primary Key*, sementara atribut lain seperti *komentar*, *rating*, dan *tanggal* digunakan untuk mencatat isi ulasan, nilai penilaian, serta waktu testimoni dibuat.

**Tabel 3. 15** *Entitas Testimoni*

Nama Atribut	Tipe Data	Keterangan
id_testimoni	INT (PK)	Primary Key, ID unik untuk setiap testimoni
id_pelanggan	INT (FK)	Foreign Key, menghubungkan ke entitas Pelanggan
Komentar	TEXT	Ulasan pelanggan mengenai layanan yang diterima
Rating	INT	Penilaian pelanggan dalam bentuk angka (misalnya: 1-5)
Tanggal	DATE	Tanggal testimoni dibuat

Relasi dengan entitas *Pelanggan* bersifat *Many-to-One*, di mana satu pelanggan bisa memberikan banyak testimoni, namun satu testimoni hanya dimiliki oleh satu pelanggan.

**Tabel 3. 16** Relasi Entitas Testimoni Dengan Entitas Lainnya

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Pelanggan	Many-to-One	id_pelanggan di Testimoni	Satu pelanggan dapat memberikan lebih dari satu testimoni, tetapi setiap testimoni hanya bisa dibuat oleh satu pelanggan tertentu.

**f. Entitas Admin**

Entitas *Admin* mewakili pengguna dengan akses khusus yang bertugas mengelola berbagai fitur dalam sistem salon. Identitas unik tiap admin disimpan dalam *id\_admin* sebagai *Primary Key*, dengan atribut pendukung berupa *nama\_admin* untuk nama lengkap dan *password\_admin* untuk autentikasi saat *login*.

**Tabel 3. 17** Entitas Admin

Nama Atribut	Tipe Data	Keterangan
id_admin	INT (PK)	Primary Key, ID unik untuk setiap admin
nama_admin	VARCHAR	Nama lengkap administrator sistem
password_admin	VARCHAR	Kata sandi untuk autentikasi

Dalam hal relasi, satu admin dapat mengelola banyak data layanan (*Menu*), proses reservasi (*Booking*), data *Karyawan*, dan konten *Artikel*. Selain itu, terdapat relasi *One-to-One* dengan entitas *Profil Usaha*, yang memastikan bahwa hanya satu admin utama yang bertanggung jawab atas profil salon.

**Tabel 3. 18** Relasi Entitas Admin Dengan Entitas Lainnya

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Menu	One-to-Many	id_admin di tabel menu	Admin bertanggung jawab dalam menambah, mengedit, dan menghapus data layanan.
Booking	One-to-Many	id_admin di tabel booking	Admin mengelola dan mengawasi proses reservasi yang dilakukan oleh pelanggan.
Karyawan	One-to-Many	id_admin di tabel karyawan	Admin bertugas menambah dan menghapus data karyawan salon.
Artikel	One-to-Many	id_admin di tabel articles	Admin membuat dan mengelola artikel yang ditampilkan pada sistem.
profil_usaha	One-to-One	id_admin di tabel profil_usaha	Admin bertanggung jawab dalam memperbarui informasi profil usaha salon.

**g. Entitas Karyawan**

Entitas *Karyawan* menyimpan data tenaga kerja yang bekerja di salon. Identitas unik tiap karyawan ditentukan oleh *id\_karyawan* sebagai *Primary Key*, dengan atribut pendukung *nama\_karyawan* untuk mencatat nama lengkap. Hubungan antara karyawan dan sistem dikelola oleh admin yang dihubungkan melalui atribut *id\_admin* sebagai *Foreign Key*.

**Tabel 3. 19** Entitas Karyawan

Nama Atribut	Tipe Data	Keterangan
id_karyawan	INT (PK)	Primary Key, ID unik untuk setiap karyawan
id_admin	INT (FK)	Foreign Key, menghubungkan ke entitas Admin
nama_karyawan	VARCHAR	Nama lengkap karyawan yang bekerja di salon

Dalam relasinya, entitas ini memiliki hubungan *Many-to-One* dengan entitas *Admin*, menandakan bahwa satu admin dapat mengelola banyak data karyawan dalam sistem.

**Tabel 3. 20** *Relasi Entitas Karyawan Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Admin	Many-to-One	id_admin di Karyawan	Satu admin dapat menambah, menghapus, atau memperbarui informasi beberapa karyawan yang bekerja di salon.

**h. Entitas Artikel**

Entitas *Artikel* digunakan untuk menyimpan informasi berupa tulisan yang ditampilkan dalam aplikasi salon, seperti tips perawatan, tren kecantikan, atau promosi layanan. Setiap artikel dikenali secara unik melalui *id\_artikel* sebagai *Primary Key*. Atribut lain yang menyertai mencakup *judul*, *konten*, *gambar*, dan *created\_at* yang masing-masing berfungsi mencatat topik utama, isi artikel, ilustrasi, dan waktu publikasi.

**Tabel 3. 21** *Entitas Artikel*

Nama Atribut	Tipe Data	Keterangan
id_artikel	INT (PK)	Primary Key, ID unik untuk setiap artikel
id_admin	INT (FK)	Foreign Key, menghubungkan ke entitas Admin
Judul	VARCHAR	Judul artikel
Jonten	TEXT	Isi artikel
Gambar	VARCHAR	URL atau path gambar pendukung artikel
created_at	DATETIME	Waktu ketika artikel diterbitkan

Relasi entitas ini bersifat *Many-to-One* dengan entitas *Admin*, artinya satu admin dapat mengelola banyak artikel, termasuk membuat, mengedit, dan menghapus kontennya dalam sistem.

**Tabel 3. 22** Relasi Entitas Artikel Dengan Entitas Lainnya

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Admin	Many-to-One	id_admin Artikel	di Satu admin dapat menambah, mengedit, dan menghapus banyak artikel dalam sistem.

**i. Entitas Kritik & Saran**

Entitas *Kritik & Saran* menyimpan masukan dari pelanggan terkait layanan salon. Setiap data masukan memiliki *id\_kritik\_saran* sebagai *Primary Key* dan *id\_pelanggan* sebagai *Foreign Key* untuk mengaitkan masukan dengan pelanggan yang mengirimkannya. Atribut lainnya mencakup *pesan* (isi kritik atau saran), *tanggal* (waktu pengiriman), *status\_baca* (penanda apakah sudah dibaca), dan *no\_whatsapp* (kontak pelanggan).

**Tabel 3. 23** Entitas Kritik & Saran

Nama Atribut	Tipe Data	Keterangan
id_kritik_saran	INT (PK)	Primary Key, ID unik untuk setiap kritik & saran
id_pelanggan	INT (FK)	Foreign Key, menghubungkan ke entitas Pelanggan
Pesan	TEXT	Isi kritik atau saran yang diberikan pelanggan
Tanggal	DATETIME	Waktu pengiriman kritik atau saran
status_baca	BOOLEAN	Menandakan apakah kritik atau saran sudah dibaca
no_whatsapp	VARCHAR	Nomor WhatsApp pelanggan

Entitas ini memiliki relasi *Many-to-One* dengan entitas *Pelanggan*, artinya satu pelanggan dapat mengirim banyak kritik atau saran, namun setiap masukan hanya berasal dari satu pelanggan.

**Tabel 3. 24** *Relasi Entitas Kritik & Saran Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Pelanggan	Many-to-One	id_pelanggan di Kritik & Saran	Satu pelanggan dapat mengirimkan banyak kritik atau saran ke sistem.

**j. Entitas *Reset Password***

Entitas *Reset Password* digunakan untuk mencatat permintaan pengaturan ulang kata sandi dari pelanggan. Atribut utama *id\_reset* berfungsi sebagai *Primary Key*, sedangkan *id\_pelanggan* sebagai *Foreign Key* untuk menghubungkan permintaan dengan akun pelanggan. Atribut *token* menyimpan kode unik untuk proses *reset*, dan *status* menandai apakah token masih aktif atau sudah digunakan.

**Tabel 3. 25** *Entitas Reset Password*

Nama Atribut	Tipe Data	Keterangan
id_reset	INT (PK)	Primary Key, ID unik untuk setiap permintaan reset password
id_pelanggan	INT (FK)	Foreign Key, menghubungkan ke entitas Pelanggan
Token	VARCHAR	Kode unik yang dikirimkan ke pelanggan untuk reset password
Status	BOOLEAN	Menunjukkan apakah permintaan reset sudah digunakan atau masih aktif

Entitas ini memiliki relasi *Many-to-One* dengan *Pelanggan*, artinya satu pelanggan dapat melakukan beberapa permintaan reset, namun setiap permintaan hanya untuk satu pelanggan.

**Tabel 3. 26** *Relasi Entitas Reset Password Dengan Entitas Lainnya*

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Pelanggan	Many-to-One	id_pelanggan di Reset Password	Satu pelanggan dapat melakukan beberapa kali permintaan reset password jika diperlukan.

### k. Entitas Profil Usaha

Entitas *Profil Usaha* digunakan untuk menyimpan informasi lengkap mengenai salon, seperti nama usaha, deskripsi layanan, alamat, kontak, media sosial, dan tautan peta lokasi. Atribut utamanya adalah *id\_profil* sebagai Primary Key, sedangkan *created\_at* mencatat waktu pembuatan atau pembaruan data profil.

**Tabel 3. 27** Entitas Profil Usaha

Nama Atribut	Tipe Data	Keterangan
id_profil	INT (PK)	Primary Key, ID unik untuk profil usaha
nama_usaha	VARCHAR	Nama resmi salon
Deskripsi	TEXT	Gambaran singkat tentang layanan yang ditawarkan
Alamat	TEXT	Alamat lengkap salon
no_telepon	VARCHAR	Nomor telepon yang dapat dihubungi pelanggan
Instagram	VARCHAR	Tautan ke akun Instagram salon
maps_link	VARCHAR	Tautan ke lokasi salon pada peta digital
created_at	DATETIME	Waktu pertama kali profil usaha dibuat atau diperbarui

Entitas ini memiliki relasi *One-to-One* dengan *Admin*, yang berarti setiap profil usaha hanya dikelola oleh satu admin utama untuk menjaga keakuratan dan konsistensi informasi.

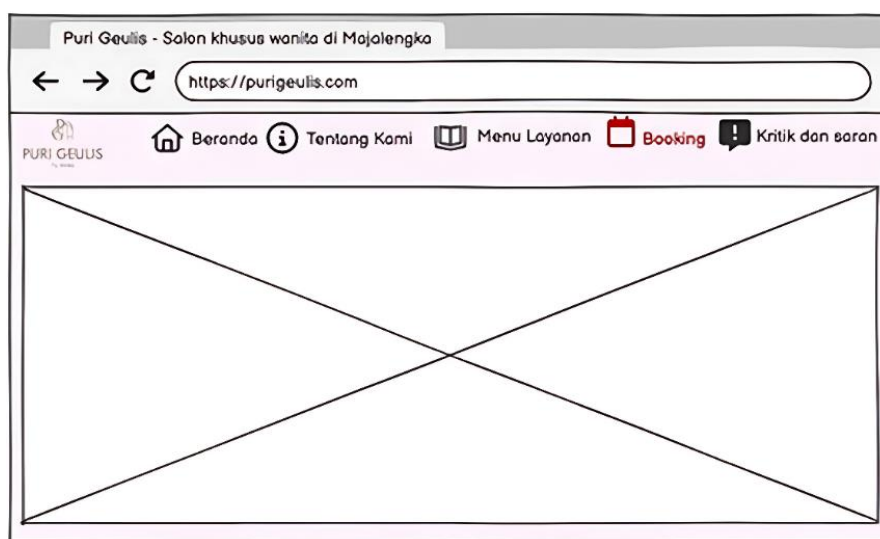
**Tabel 3. 28** Relasi Entitas Profil Usaha Dengan Entitas Lainnya

Entitas Terkait	Jenis Relasi	Foreign Key	Keterangan
Admin	One-to-One	id_admin Admin	di Setiap profil usaha hanya dikelola oleh satu admin utama.

### 3.5.3. Perancangan Antarmuka

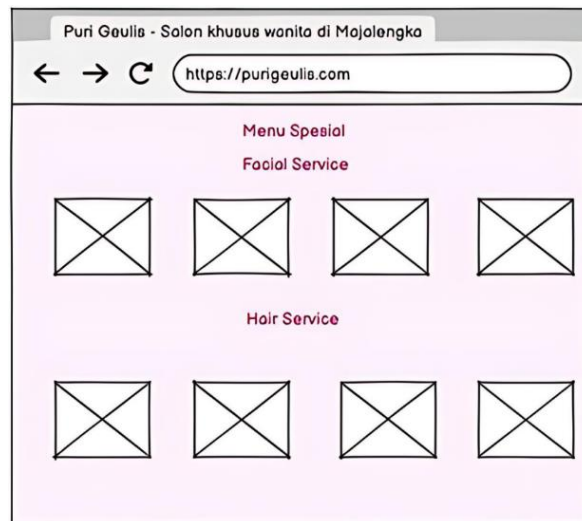
Perancangan antarmuka pengguna (*user interface*) disusun untuk memberikan tampilan yang sederhana, mudah dipahami, dan *user-friendly*, sesuai kebutuhan pengguna dalam menggunakan sistem reservasi dan manajemen layanan salon. Antarmuka dibuat berdasarkan hasil analisis kebutuhan, serta menyesuaikan fungsi dari masing-masing fitur yang tersedia dalam sistem.

#### a. Halaman Beranda *User*



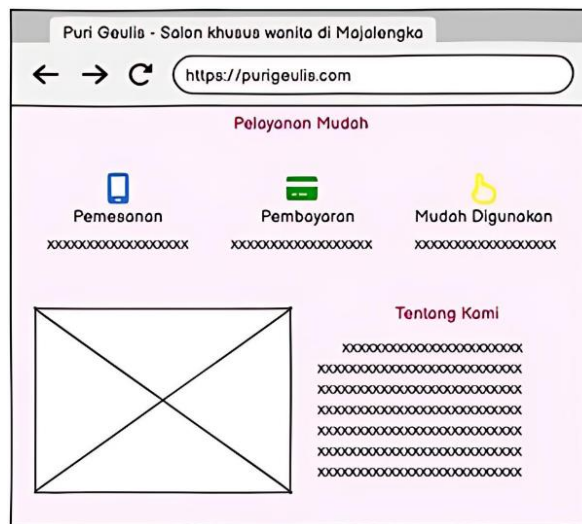
**Gambar 3. 43** *Desain Navigasi Dan Carousel Promosi*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman beranda untuk Puri Geulis, sebuah salon khusus wanita di Majalengka, dan memiliki menu navigasi dengan opsi seperti *Beranda*, *Tentang Kami*, *Menu Layanan*, *Booking*, dan *Kritik dan saran*.



**Gambar 3. 44** *Desain Menu Layanan Dalam Beranda*

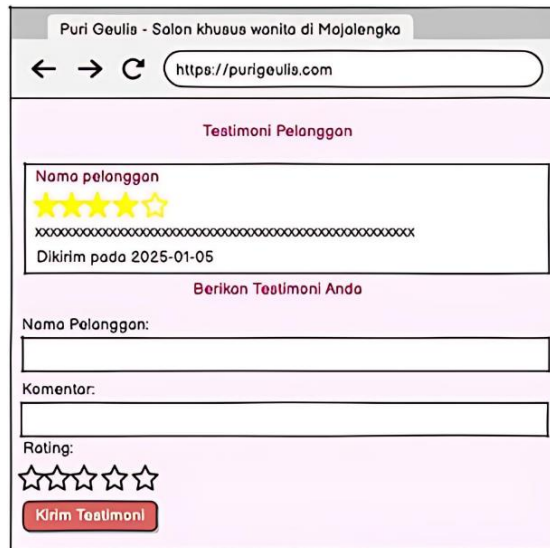
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman menu spesial untuk Puri Geulis, yang mencakup kategori *Facial Service* dan *Hair Service* beserta *placeholder* untuk setiap layanan.



**Gambar 3. 45** *Desain Deskripsi Salon*

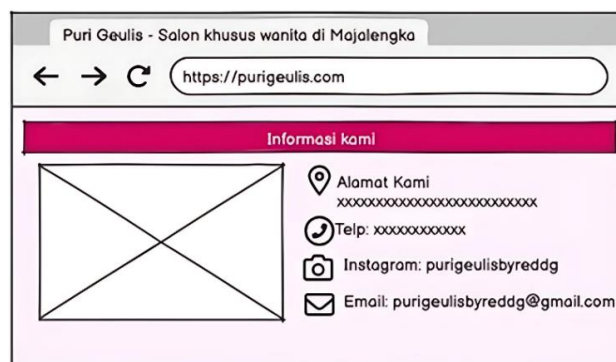
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan bagian "Pelayanan Mudah" yang mencakup ikon dan teks untuk Pemesanan, Pembayaran, dan Mudah

Digunakan, serta bagian "Tentang Kami" dengan *placeholder* teks dan sebuah kotak untuk kemungkinan gambar atau informasi lebih lanjut.



**Gambar 3. 46** *Desain Testimoni*

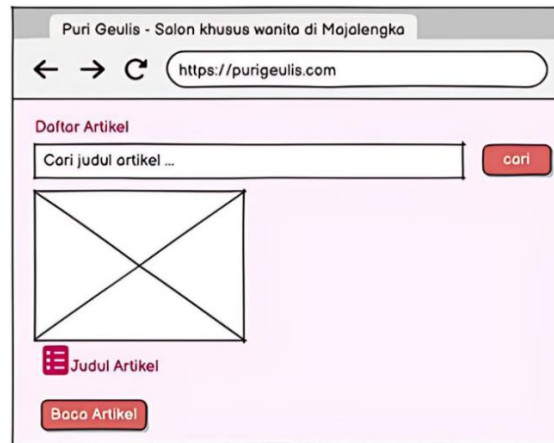
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman testimoni pelanggan yang berisi contoh testimoni dengan nama, *rating* bintang, teks, dan tanggal pengiriman, serta formulir untuk "Berikan Testimoni Anda" yang mencakup kolom nama pelanggan, komentar, *rating*, dan tombol *Kirim Testimoni*.



**Gambar 3. 47** *Desain Informasi Salon*

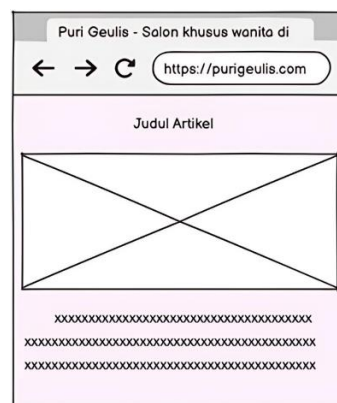
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman "Informasi kami" yang berisi detail kontak seperti alamat, nomor telepon, akun *Instagram*

@purigeulisbyreddg, dan alamat *Email* purigeulisbyreddg@gmail.com, disertai dengan *placeholder* gambar di sisi kiri.



**Gambar 3. 48** *Desain Artikel*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman "Daftar Artikel" dengan kolom pencarian "Cari judul artikel..." dan tombol "cari", serta menampilkan *placeholder* untuk pratinjau artikel, judul artikel, dan tombol "Baca Artikel".

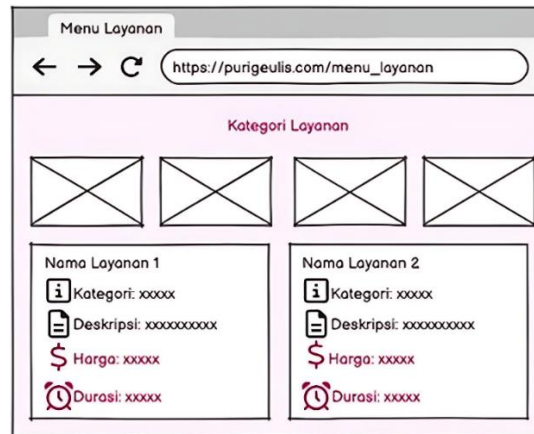


**Gambar 3. 49** *Desain Artikel Ketika Di Buka*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com>, menunjukkan halaman detail artikel dengan "Judul Artikel" di bagian atas, diikuti oleh *placeholder* besar yang kemungkinan

untuk gambar atau media, dan diakhiri dengan beberapa baris *placeholder* teks untuk isi artikel.

### b. Halaman Menu Layanan User



**Gambar 3. 50** Desain Halaman Menu Layanan User

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/menu\\_layanan](https://purigeulis.com/menu_layanan), menunjukkan halaman "Menu Layanan" dengan bagian "Kategori Layanan" yang memiliki empat *placeholder* di atas, serta menampilkan dua kartu layanan yang berisi detail seperti nama layanan, kategori, deskripsi, harga, dan durasi.

### c. Halaman Reservasi

**Gambar 3. 51** Desain Formulir Halaman Reservasi

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/reservasi>, menunjukkan "Formulir Reservasi" yang mencakup kolom input untuk nama lengkap dan nomor *Whatsapp* pelanggan, serta *dropdown* untuk memilih karyawan dengan pilihan Karyawan 1, Karyawan 2, dan Karyawan 3.



**Gambar 3. 52** *Desain Memilih Menu Layanan Pada Reservasi*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/reservasi>, menunjukkan bagian "Pilih layanan (Bisa memilih lebih dari satu)" dengan *dropdown* "Kategori Layanan", serta daftar lima opsi layanan berupa kotak centang yang masing-masing mencantumkan harga dan durasi.



**Gambar 3. 53** *Desain Submit Dan Penentuan Waktu Reservasi*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/reservasi>, menunjukkan bagian untuk "Pilih Tanggal" yang sudah terisi 05/01/2025 dan "Pilih Waktu", dilengkapi dengan tombol "Reservasi", tampilan "Total Biaya: Rp 0", dan tombol "Ajukan pembatalan reservasi".

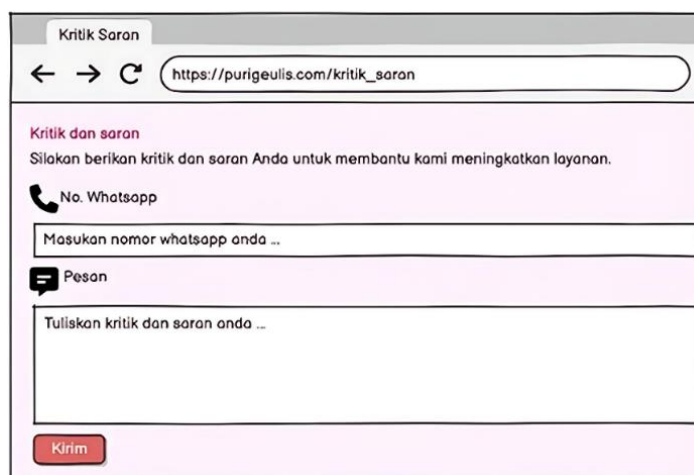


No	Nama Lengkap	Tanggal Booking	Waktu Booking	Waktu Selesai	Nama Karyawan
1	Pelanggan 1	05/01/2025	9.00	10.00	Karyawan 1
2	Pelanggan 2	05/01/2025	10.00	11.00	Karyawan 2
3	Pelanggan 3	05/01/2025	11.00	12.00	Karyawan 3

**Gambar 3. 54** *Desain Daftar Reservasi Yang Telah Dilakukan*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/reservasi>, menunjukkan "Daftar Reservasi" dalam bentuk tabel yang mencakup nomor urut, nama lengkap pelanggan, tanggal *booking*, waktu *booking*, waktu selesai, dan nama karyawan, dengan tiga contoh entri data.

#### d. Halaman Kritik Dan Saran



**Kritik Saran**

Kritik dan saran  
Silakan berikan kritik dan saran Anda untuk membantu kami meningkatkan layanan.

No. Whatsapp  
Masukan nomor whatsapp anda ...

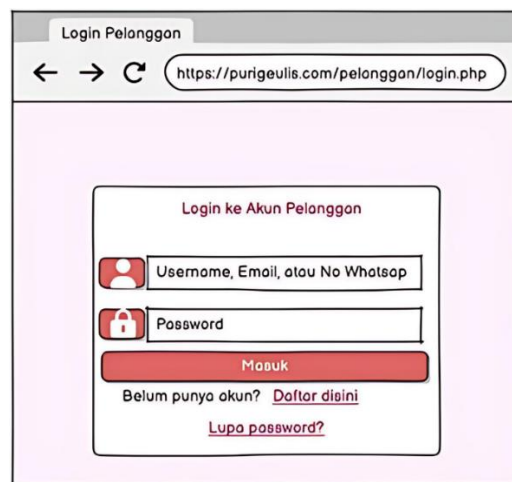
Pesan  
Tuliskan kritik dan saran anda ...

Kirim

**Gambar 3. 55** *Desain Halaman Kritik Dan Saran*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/kritik\\_saran](https://purigeulis.com/kritik_saran), menunjukkan halaman "Kritik dan saran" yang menyediakan kolom input untuk nomor *Whatsapp* dan pesan, disertai *placeholder* teks serta tombol "Kirim" untuk mengirimkan kritik dan saran pelanggan.

e. **Halaman *Login*, Registrasi Dan Lupa *Password***



**Gambar 3. 56** *Desain Halaman Login Akun Pelanggan*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/pelanggan/login.php>, menunjukkan halaman *Login* Pelanggan dengan formulir "Login ke Akun Pelanggan" yang berisi kolom untuk *Username*, *Email*, atau nomor *Whatsapp*, serta *Password*, dilengkapi tombol "Masuk", tautan "Daftar disini", dan tautan "Lupa *password*?".

**Gambar 3. 57** *Desain Login Akun Admin*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/login\\_admin.php](https://purigeulis.com/admin/login_admin.php), menunjukkan halaman *Login Admin* dengan formulir yang meminta *Username* dan *Password* melalui kolom input dengan *placeholder* teks, serta tombol *Login*.

**Gambar 3. 58** *Desain Halaman Registrasi Pelanggan*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/pelanggan/registrasi>, menunjukkan formulir "Registrasi Akun Pelanggan" yang meminta input untuk nama lengkap, nomor

telepon (contoh), alamat lengkap, *Email*, *Username*, *Password* (minimal 6 karakter), dan konfirmasi *Password*, dilengkapi dengan tombol "Daftar".

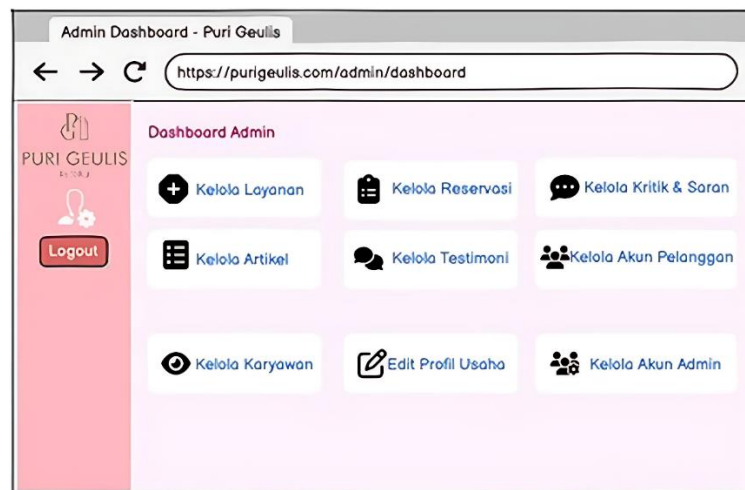
**Gambar 3. 59** *Desain Halaman Metode Reset Password*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/pelanggan/lupa\\_password](https://purigeulis.com/pelanggan/lupa_password), menunjukkan halaman "Lupa Password" atau "Reset Password" yang menawarkan opsi *reset* melalui nomor *Whatsapp* atau *Email*, diikuti kolom input untuk memasukkan nomor *Whatsapp* atau *Email*, dan tombol "Kirim link reset password".

**Gambar 3. 60** *Desain Halaman Perbarui Password*

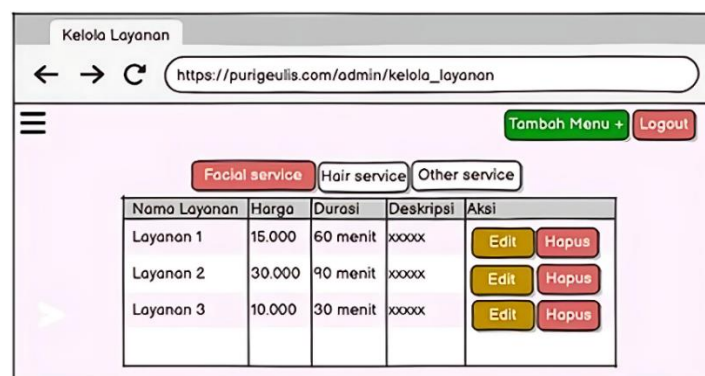
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/pelanggan/reset-password.php?token=xxxxx>, menunjukkan halaman "*Reset Password*" yang berisi kolom input untuk "Password baru" dan "Konfirmasi password", serta tombol "Perbarui password".

f. **Halaman *Dashboard Admin***



**Gambar 3. 61** *Desain Halaman Dashboard Admin*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat <https://purigeulis.com/admin/dashboard>, menunjukkan "*Admin Dashboard*" Puri Geulis yang memiliki bilah sisi dengan logo dan tombol "*Logout*", serta panel utama yang menampilkan opsi-opsi pengelolaan seperti Layanan, Reservasi, Kritik & Saran, Artikel, Testimoni, Akun Pelanggan, Karyawan, *Edit Profil Usaha*, dan Akun *Admin*.



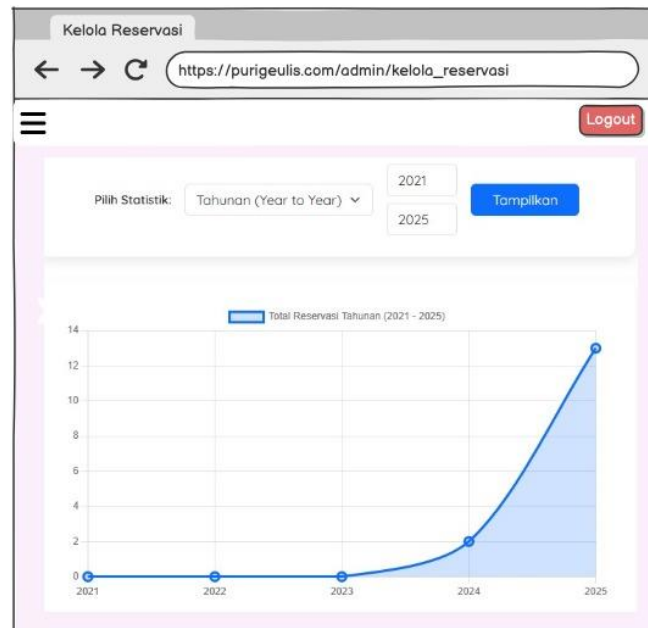
**Gambar 3. 62** *Desain Fitur kelola Layanan*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_layanan](https://purigeulis.com/admin/kelola_layanan), menunjukkan halaman "Kelola Layanan" yang memiliki ikon menu, tombol "Tambah Menu +" dan "Logout", tiga tombol filter kategori layanan (*Facial service*, *Hair service*, *Other service*), serta tabel berisi daftar layanan lengkap dengan harga, durasi, deskripsi, dan tombol "Edit" serta "Hapus".

ID	Nama Lengkap	No WhatsApp	Tanggal Booking	Waktu Booking	Waktu Selesai	Total Biaya	Layanan	Karyawan	Aksi
296	Nama pelanggan	085722792480	2025-04-29	12.00.00	00.00.00	Rp 750.000	Layanan 1	Karyawan 1	Hapus
297	Nama pelanggan	085722792480	2025-04-29	12.00.00	00.00.00	Rp 750.000	Layanan 2	Karyawan 2	Hapus
298	Nama pelanggan	085722792480	2025-04-29	12.00.00	00.00.00	Rp 750.000	Layanan 3	Karyawan 3	Hapus

**Gambar 3. 63** *Desain Fitur Kelola Resrvasi*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_reservasi](https://purigeulis.com/admin/kelola_reservasi), menunjukkan halaman "Kelola Reservasi" yang dilengkapi ikon menu, tombol "Logout", bilah pencarian "Cari No *Whatsapp*", tombol "Lihat Statistik Reservasi", dan tabel data reservasi dengan kolom-kolom detail serta tombol "Hapus" untuk setiap entri.



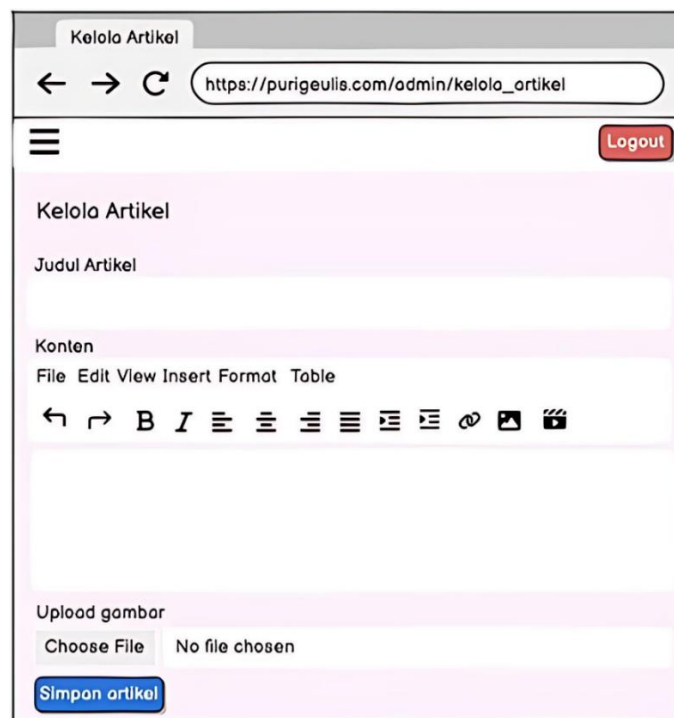
**Gambar 3. 64** Desain Tampilan Statistik Reservasi

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_reservasi](https://purigeulis.com/admin/kelola_reservasi), menunjukkan halaman "Kelola Reservasi" yang dilengkapi ikon menu, tombol "Logout", opsi "Pilih Statistik:" dengan *dropdown* "Tahunan (*Year to Year*)" beserta kolom tahun dan tombol "Tampilkan", serta grafik garis "Total Reservasi Tahunan (2021 - 2025)" yang menunjukkan peningkatan reservasi.

Pesan	No Whatsapp	Aksi
xxxxxxxxxxxx	086722692380	Belum dibaca Hapus
xxxxxxxxxxxx	086322192280	Sudah dibaca Hapus

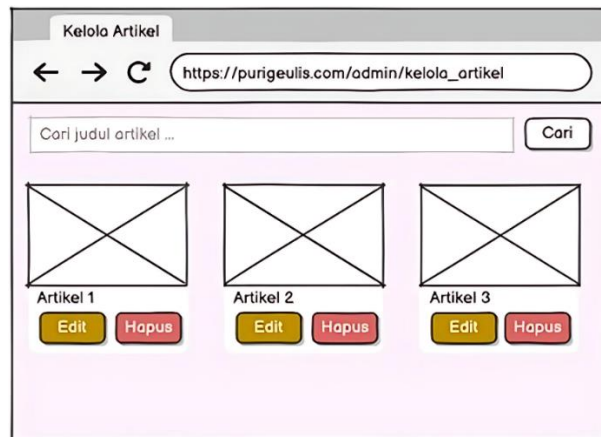
**Gambar 3. 65** Desain Fitur Kelola Kritik Dan Saran

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_kritik\\_saran](https://purigeulis.com/admin/kelola_kritik_saran), menunjukkan halaman "Kelola Kritik dan Saran" yang dilengkapi ikon menu dan tombol "Logout", serta tabel "Kritik & Saran Dari Pelanggan" yang berisi kolom pesan, nomor *Whatsapp*, dan kolom aksi dengan tombol "Belum dibaca" atau "Sudah dibaca" serta tombol "Hapus".



**Gambar 3. 66** *Desain Fitur Tulis Artikel*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_artikel](https://purigeulis.com/admin/kelola_artikel), menunjukkan halaman "Kelola Artikel" yang dilengkapi ikon menu, tombol "Logout", kolom input untuk "Judul Artikel", area "Konten" dengan bilah alat *editor* teks kaya, bagian "Upload gambar" dengan tombol "Choose File", dan tombol "Simpan artikel".



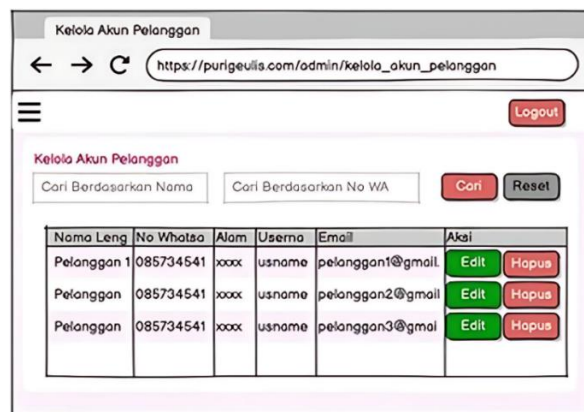
**Gambar 3. 67** Desain Fitur Kelola Artikel

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_artikel](https://purigeulis.com/admin/kelola_artikel), menunjukkan halaman "Kelola Artikel" dengan kolom pencarian "Cari judul artikel..." dan tombol "Cari", serta tiga *placeholder* artikel lengkap dengan tombol "Edit" dan "Hapus" untuk masing-masing artikel.



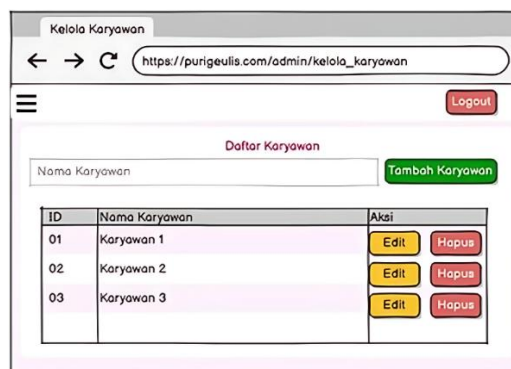
**Gambar 3. 68** Desain Fitur Kelola Testimoni

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_testimoni](https://purigeulis.com/admin/kelola_testimoni), menunjukkan halaman "Kelola Testimoni" yang memiliki ikon menu dan tombol "*Logout*", serta daftar "Testimoni Pelanggan" yang berisi detail seperti *rating* bintang, teks *placeholder*, tanggal pengiriman, dan tombol aksi "Konfirmasi", "Tolak", atau "Hapus" berdasarkan status testimoni.



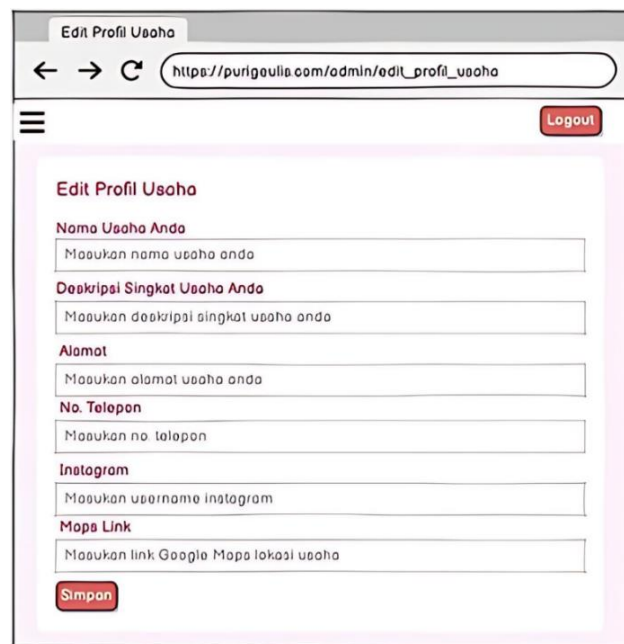
**Gambar 3. 69** Desain Fitur Kelola Akun Pelanggan

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_akun\\_pelanggan](https://purigeulis.com/admin/kelola_akun_pelanggan), menunjukkan halaman "Kelola Akun Pelanggan" yang dilengkapi ikon menu, tombol "*Logout*", dua kolom pencarian "Cari Berdasarkan Nama" dan "Cari Berdasarkan No *WA*", tombol "Cari" dan "*Reset*", serta tabel data akun pelanggan dengan detail nama, nomor *Whatsapp*, alamat, *Username*, *Email*, dan kolom aksi berisi tombol "*Edit*" serta "Hapus".



**Gambar 3. 70** Desain Fitur Tinjau Karyawan

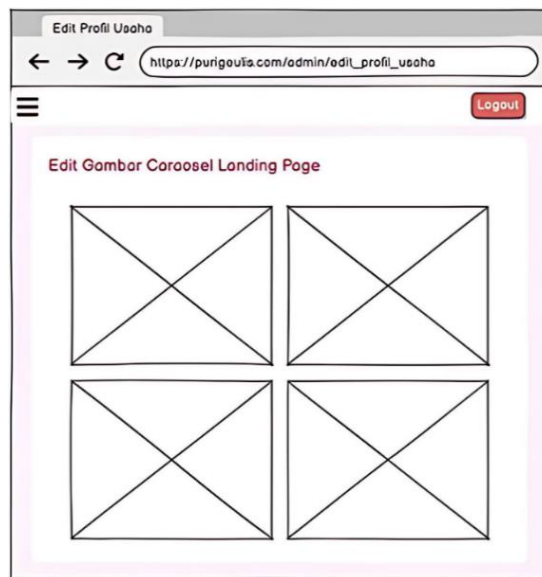
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/elola\\_karyawan](https://purigeulis.com/admin/elola_karyawan), menunjukkan halaman "Kelola Karyawan" yang dilengkapi ikon menu dan tombol "Logout", bagian "Daftar Karyawan" dengan kolom input "Nama Karyawan" dan tombol "Tambah Karyawan", serta tabel data karyawan dengan kolom ID, Nama Karyawan, dan kolom Aksi yang berisi tombol "Edit" serta "Hapus".



The screenshot shows a web browser window with the title "Edit Profil Usaha" and the URL "https://purigeulis.com/admin/edit\_profil\_usaha". The page features a navigation menu icon on the left and a "Logout" button on the right. The main content area is titled "Edit Profil Usaha" and contains several input fields with placeholder text: "Nama Usaha Anda" (placeholder: "Masukkan nama usaha anda"), "Deskripsi Singkat Usaha Anda" (placeholder: "Masukkan deskripsi singkat usaha anda"), "Alamat" (placeholder: "Masukkan alamat usaha anda"), "No. Telepon" (placeholder: "Masukkan no telepon"), "Instagram" (placeholder: "Masukkan username instagram"), and "Maps Link" (placeholder: "Masukkan link Google Maps lokasi usaha"). A "Simpan" button is located at the bottom of the form.

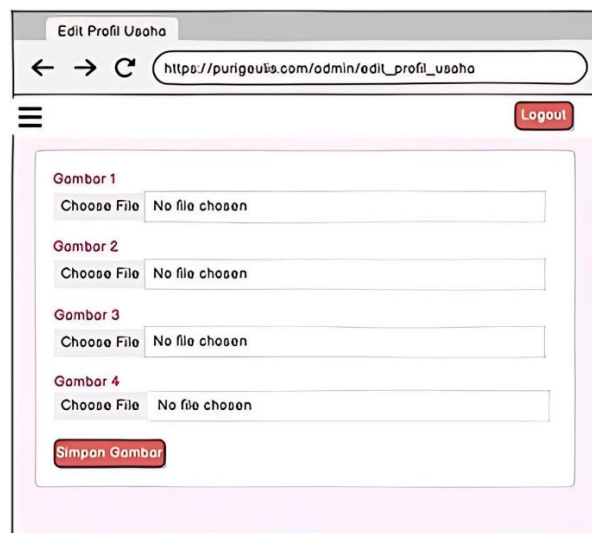
**Gambar 3. 71** Desain Fitur Edit Profil Usaha Salon

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/edit\\_profil\\_usaha](https://purigeulis.com/admin/edit_profil_usaha), menunjukkan halaman "Edit Profil Usaha" yang dilengkapi ikon menu, tombol "Logout", dan beberapa kolom input dengan *placeholder* untuk nama usaha, deskripsi singkat, alamat, nomor telepon, *Instagram*, dan *Maps Link*, serta tombol "Simpan".



**Gambar 3. 72** *Desain Tinjau Carousel Pada Halaman Beranda*

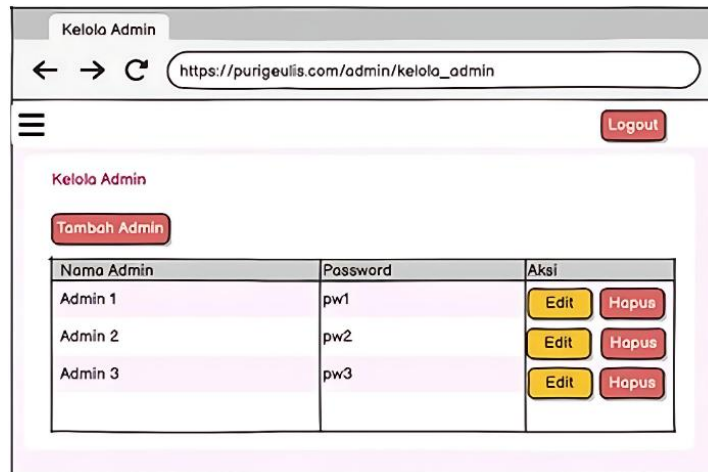
Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/edit\\_profil\\_usaha](https://purigeulis.com/admin/edit_profil_usaha), menunjukkan halaman "Edit Gambar *Carousel Landing Page*" yang dilengkapi ikon menu, tombol "Logout", dan empat kotak *placeholder* gambar yang disusun dalam grid.



**Gambar 3. 73** *Desain Edit Carousel Pada Halaman Beranda.*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/edit\\_profil\\_usaha](https://purigeulis.com/admin/edit_profil_usaha), menunjukkan halaman "Edit *Profil Usaha*" yang berisi ikon menu, tombol "Logout", dan empat

bagian "Gambar [nomor]" dengan tombol "*Choose File*" untuk mengunggah berkas, serta tombol "Simpan Gambar".



**Gambar 3. 74** *Desain Fitur Kelola Admin*

Gambar perancangan antarmuka ini menampilkan tampilan peramban *web* dengan alamat [https://purigeulis.com/admin/kelola\\_admin](https://purigeulis.com/admin/kelola_admin), menunjukkan halaman "Kelola Admin" yang dilengkapi ikon menu, tombol "Logout", tombol "Tambah Admin", serta tabel daftar Admin yang berisi nama, *password*, dan kolom aksi dengan tombol "Edit" serta "Hapus".