

# SOFTWARE DESIGN DOCUMENT (SDD)

Website Sistem Booking Zara  
Eyelash

...

<b>BAB I Introduction .....</b>	<b>3</b>
<b>1.1 Purpose.....</b>	<b>3</b>
<b>1.2 Scope .....</b>	<b>3</b>
<b>1.3 Overview .....</b>	<b>4</b>
<b>Bab II System Overview .....</b>	<b>0</b>
<b>Bab III Application Design .....</b>	<b>3</b>
<b>3.2 Data Flow Diagram.....</b>	<b>3</b>
<b>3.3 Use Case Scenario .....</b>	<b>4</b>
<b>3.3 Class Diagram .....</b>	<b>11</b>
<b>3.3 Sequence Diagram.....</b>	<b>15</b>
Sequence Diagram : Melihat Daftar Layanan.....	15
Sequence Diagram : Melakukan Booking Layanan .....	18
Sequence Diagram : Membatalkan Booking .....	21
Sequence Diagram : Melihat Galeri .....	24
Sequence Diagram : Memberi Ulasan.....	27
Sequence Diagram : Melihat Riwayat Booking.....	30
Sequence Diagram : Mengelola Layanan .....	33
Sequence Diagram : Mengelola Transaksi .....	36
Sequence Diagram : Login Admin .....	39
<b>3.3 Activity Diagram .....</b>	<b>41</b>
Activity Diagram : Melakukan Booking Layanan .....	41
Activity Diagram : Membatalkan Booking.....	43
Activity Diagram : Memberi Ulasan .....	45
Activity Diagram : Mengelola Layanan .....	47
Activity Diagram : Mengelola Transaksi .....	49
Activity Diagram : Login Admin .....	51
<b>3.3 State Diagram.....</b>	<b>53</b>
<b>I.1.A. State Diagram : Booking.....</b>	<b>53</b>
<b>I.1.B. State Diagram : Transaksi.....</b>	<b>55</b>
<b>I.1.C. State Diagram : Layanan.....</b>	<b>57</b>
<b>3.3 Deployment Diagram .....</b>	<b>59</b>
<b>Bab IV Data Design .....</b>	<b>61</b>

**4.1 Logical Design ..... 61**  
**4.2 Physical Design..... 61**

# **BAB I Introduction**

## **1.1 Purpose**

Dokumen Software Design Document (SDD) ini disusun sebagai pedoman resmi dalam proses pengembangan *Website Sistem Booking Zara Eyelash*. Tujuan utama penyusunan dokumen ini adalah memberikan gambaran lengkap mengenai desain teknis, struktur arsitektur, alur proses, hingga kebutuhan sistem yang diperlukan agar aplikasi dapat dikembangkan secara konsisten, terstruktur, dan sesuai kebutuhan bisnis.

Selain berfungsi sebagai dokumentasi teknis bagi developer, SDD ini juga menjadi acuan bagi tim analis, desainer UI/UX, tester, serta pihak pemilik usaha dalam memahami cara kerja sistem. Dengan adanya dokumentasi ini, setiap pihak yang terlibat dalam proyek pengembangan dapat memiliki pemahaman yang sama, sehingga meminimalisir kesalahan interpretasi selama proses implementasi.

Dokumen ini juga memungkinkan proses pemeliharaan sistem berjalan lebih mudah di masa mendatang. Ketika terjadi penambahan fitur atau revisi, tim pengembang dapat mengacu pada dokumen ini untuk memastikan kesesuaian desain dan arsitektur yang telah direncanakan sebelumnya.

## **1.2 Scope**

Lingkup pengembangan sistem booking Zara Eyelash berfokus pada penyediaan platform pemesanan layanan kecantikan (eyelash) secara online, yang dapat diakses oleh pelanggan dan admin. Sistem ini dirancang untuk menangani berbagai proses bisnis yang terjadi dalam kegiatan operasional Zara Eyelash.

Secara garis besar, ruang lingkup sistem meliputi:

### **Fitur untuk Customer**

- Melihat daftar layanan eyelash yang tersedia
- Melakukan pemesanan (booking) secara online
- Menentukan jadwal layanan (tanggal & jam)
- Membatalkan booking
- Memberikan ulasan terhadap layanan yang telah diterima
- Melihat galeri hasil pengerjaan
- Mengakses riwayat booking lengkap

### Fitur untuk Admin

- Login sebagai administrator
- Mengelola layanan (tambah, edit, hapus)
- Mengelola galeri foto
- Memproses booking (konfirmasi, batalkan)
- Mengelola transaksi / status pembayaran
- Melihat statistik operasional melalui dashboard

Lingkup sistem **tidak mencakup fitur pembayaran online**, namun dapat diperluas pada pengembangan versi selanjutnya. Fokus utama sistem versi ini adalah memastikan proses booking berjalan dengan lancar, akurat, dan memudahkan kedua belah pihak.

## 1.3 Overview

Dokumen SDD ini memuat penjelasan lengkap mengenai desain sistem Zara Eyelash, yang disusun dalam beberapa bagian utama. Bagian-bagian tersebut menjelaskan bagaimana sistem bekerja dari berbagai perspektif, mulai dari alur penggunaan hingga struktur teknis internal.

Isi utama dokumen mencakup:

- **Gambaran umum sistem**, termasuk peran pengguna dan fungsi yang tersedia.
- **Desain arsitektur aplikasi**, yang menjelaskan hubungan antara komponen frontend, backend, dan database.
- **Diagram DFD**, Class Diagram, Sequence Diagram, Activity Diagram, State Diagram, dan Deployment Diagram. Diagram ini berguna untuk memvisualisasikan proses, alur, dan hubungan antar objek dalam sistem.
- **Perancangan data**, meliputi ERD (Entity Relationship Diagram), desain logis, dan desain fisik database.
- **Desain antarmuka pengguna (UI/UX)**, menjelaskan tampilan dan struktur halaman baik untuk customer maupun admin.
- **Kebutuhan teknis**, meliputi hardware, software, dan komunikasi yang diperlukan untuk menjalankan sistem secara optimal.

Dengan dokumentasi ini, seluruh proses pengembangan dapat dilakukan secara terarah, terukur, serta mudah dipahami.

## **Bab II System Overview**

Sistem Booking Zara Eyelash merupakan aplikasi berbasis web yang dirancang untuk menyediakan proses pemesanan layanan kecantikan secara cepat, mudah, dan efisien. Sistem ini dikembangkan untuk menggantikan proses manual yang selama ini mungkin dilakukan melalui chat atau komunikasi langsung, yang rentan terhadap kesalahan pencatatan dan antrian yang tidak teratur.

Dengan sistem ini, pelanggan dapat melihat semua layanan yang tersedia, melakukan booking berdasarkan jadwal yang disediakan, memantau riwayat booking, hingga memberikan ulasan terhadap layanan yang telah diterimanya. Sistem ini tidak hanya membantu pelanggan, tetapi juga memudahkan admin dalam mengelola operasional harian.

### **Fitur utama sistem:**

#### **1. Booking Layanan Eyelash**

Fitur ini memungkinkan customer untuk memesan layanan eyelash secara online tanpa harus datang langsung ke tempat atau menghubungi admin secara manual. Pada halaman booking, customer dapat memilih jenis layanan yang diinginkan, melihat detail layanan seperti harga dan durasi pengerjaan, serta mengisi data diri. Setelah proses booking dilakukan, sistem akan menyimpan informasi tersebut dan memberikan status awal “Pending” untuk menunggu verifikasi admin.

Fitur ini dirancang untuk memberikan kemudahan sekaligus mengurangi antrean dan kesalahan pencatatan.

#### **2. Penjadwalan Booking**

Sistem dilengkapi dengan mekanisme pemilihan jadwal berupa tanggal dan jam yang dapat dipilih oleh customer. Saat customer memilih waktu, sistem akan memeriksa apakah slot tersebut masih tersedia atau sudah terisi oleh pelanggan lain. Hal ini mencegah terjadinya bentrok jadwal. Dengan adanya penjadwalan yang terstruktur, proses layanan dapat berlangsung lebih efisien dan rapi, baik bagi customer maupun pihak Zara Eyelash.

#### **3. Melihat Galeri**

Galeri merupakan fitur yang menampilkan foto-foto hasil pengerjaan eyelash sebelumnya. Fitur ini bertujuan memberikan gambaran kepada calon customer mengenai kualitas layanan dan gaya pengerjaan yang tersedia.

Galeri biasanya berisi foto before-after, model eyelash yang tersedia, atau dokumentasi hasil pekerjaan. Fitur ini juga dapat membantu meningkatkan kepercayaan customer terhadap layanan Zara Eyelash.

#### **4. Memberi Ulasan Setelah Layanan**

Setelah layanan selesai dilakukan, customer dapat memberikan ulasan terhadap layanan yang mereka terima. Ulasan ini dapat berupa rating, komentar, atau kritik dan saran. Fitur ini tidak hanya membantu admin dalam mengevaluasi kualitas layanan, tetapi juga menjadi referensi bagi pelanggan baru yang ingin melihat pengalaman pengguna lain sebelum melakukan booking.

#### **5. CRUD Layanan oleh Admin**

Admin memiliki akses penuh untuk mengelola layanan yang tersedia pada sistem. Fitur CRUD (Create, Read, Update, Delete) memungkinkan admin:

- a. **Menambah layanan baru**, termasuk harga, deskripsi, dan durasi.
- b. **Mengedit layanan**, jika ada perubahan harga atau informasi tambahan.
- c. **Menghapus layanan**, jika layanan sudah tidak tersedia.
- d. **Melihat daftar layanan**, untuk memastikan informasi selalu up-to-date.

Dengan adanya fitur ini, data layanan tetap konsisten dan relevan bagi customer.

#### **6. Pengelolaan Transaksi dan Status Pembayaran**

Admin dapat mengelola transaksi terkait booking layanan yang dilakukan customer. Status transaksi dapat berupa:

- a. Pending
- b. Lunas / Paid
- c. Dibatalkan
- d. Menunggu Konfirmasi

Admin dapat memperbarui status transaksi setelah customer melakukan pembayaran secara langsung. Pencatatan transaksi ini membantu dalam proses pelaporan, rekap harian, dan pemantauan aktivitas bisnis.

## **7. Dashboard Statistik**

Dashboard merupakan fitur yang menyajikan ringkasan data operasional dalam bentuk grafik atau angka. Informasi yang ditampilkan dapat meliputi:

- a. Jumlah booking hari ini
- b. Jumlah booking dalam satu bulan
- c. Layanan paling banyak dipesan
- d. Jumlah transaksi masuk
- e. Rating rata-rata dari ulasan customer

Dashboard ini membantu admin atau pemilik Zara Eyelash untuk memantau kinerja bisnis secara cepat dan efisien.

## **8. 8. Notifikasi Sederhana melalui Tampilan Status**

Sistem menyediakan notifikasi berbasis tampilan status pada halaman customer maupun admin. Misalnya:

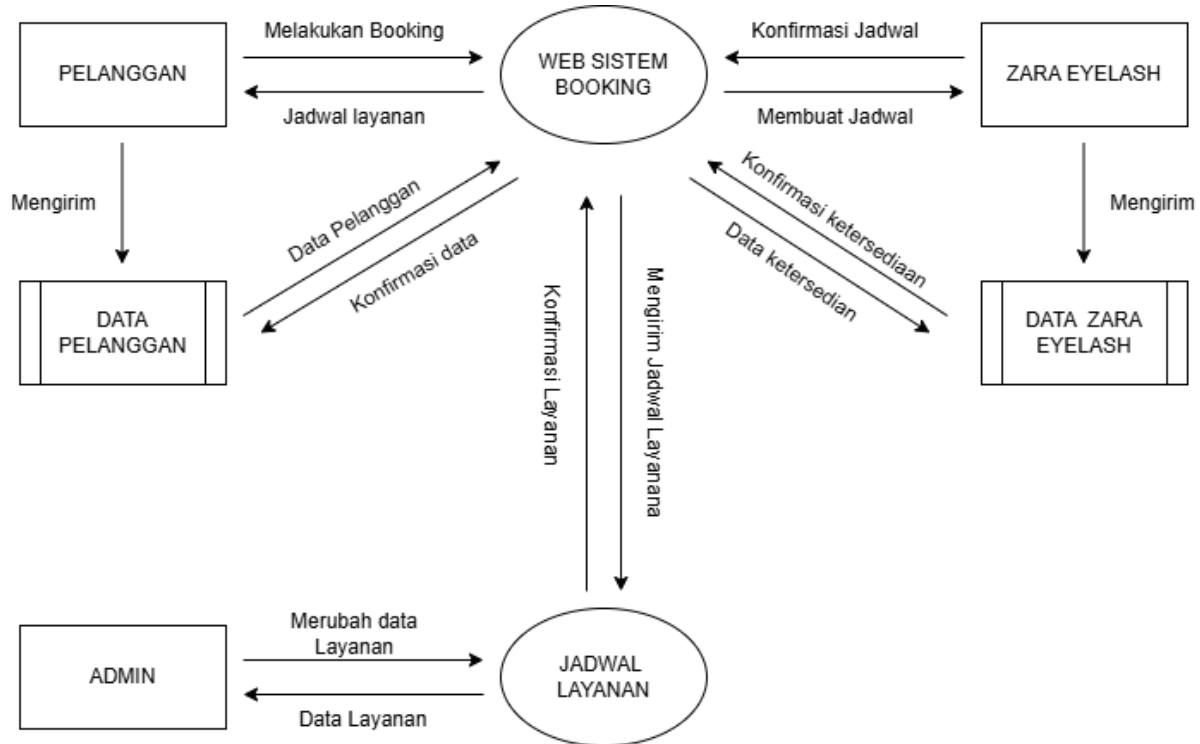
- a. “Booking berhasil dibuat”
- b. “Menunggu konfirmasi admin”
- c. “Booking Anda telah dikonfirmasi”
- d. “Layanan selesai, silakan beri ulasan”
- e. “Pembatalan berhasil dilakukan”

Meskipun tidak menggunakan notifikasi real-time seperti push notification, tampilan status ini sudah cukup memberikan kejelasan kepada customer terkait perkembangan booking mereka.



## Bab III Application Design

### 3.2 Data Flow Diagram



Gambar 1 Data Flow diagram

#### Penjelasan

Sistem ini dirancang untuk mengintegrasikan proses pemesanan layanan antara pelanggan, pihak manajemen (Admin), dan operasional toko (Zara Eyelash). Berikut adalah rincian fungsionalitas berdasarkan aliran data:

##### 1. Aliran Data Entitas Pelanggan

Pelanggan berinteraksi langsung dengan proses **Web Sistem Booking**. Proses dimulai saat pelanggan mengirimkan data untuk **Melakukan Booking**. Data tersebut kemudian diteruskan oleh sistem ke dalam *data store* **Data Pelanggan**. Sebaliknya, pelanggan akan menerima informasi berupa **Jadwal Layanan** sebagai konfirmasi bahwa pesanan telah diterima dan dijadwalkan oleh sistem.

##### 2. Aliran Data Entitas Zara Eyelash (Operasional)

Entitas Zara Eyelash bertanggung jawab atas ketersediaan layanan di lapangan. Pihak operasional melakukan input **Membuat Jadwal** ke dalam **Web Sistem Booking**, yang datanya disimpan dan divalidasi melalui *data store* **Data Zara Eyelash**. Setelah pelanggan melakukan pemesanan, entitas ini menerima informasi untuk kemudian memberikan **Konfirmasi Jadwal** kembali ke sistem agar status booking menjadi valid.

### 3. Aliran Data Entitas Admin (Manajemen)

Admin bertindak sebagai pengelola konten dan parameter layanan dalam sistem. Admin berinteraksi dengan proses **Jadwal Layanan** untuk melakukan pembaruan atau **Merubah Data Layanan** (seperti jenis treatment, durasi, atau harga). Admin juga dapat memantau **Data Layanan** yang aktif untuk memastikan seluruh informasi yang ditampilkan kepada pelanggan tetap akurat dan relevan.

### 4. Hubungan Antar Proses (Sinkronisasi Data)

Terdapat koordinasi internal antara proses **Web Sistem Booking** dan **Jadwal Layanan**:

- Proses **Web Sistem Booking** mengirimkan permintaan **Konfirmasi Layanan** ke proses jadwal untuk memastikan slot yang dipilih pelanggan tersedia.
- Proses **Jadwal Layanan** merespons dengan **Mengirim Jadwal Layanan** yang telah terverifikasi untuk ditampilkan kembali di sisi *interface* pengguna (web).

### Analisis Data Store (Penyimpanan Data)

Berdasarkan diagram, terdapat dua penyimpanan data utama:

- **Data Pelanggan:** Menyimpan profil dan rekam medis/transaksi pelanggan.
- **Data Zara Eyelash:** Menyimpan informasi ketersediaan slot, stok material, atau kapasitas terapis yang tersedia pada jam-jam tertentu.

## 3.3 Use Case Scenario

- Usecase Scenario : Melihat Daftar Layanan

Use Case	Melihat Daftar Layanan
Scenario	Customer ingin mengetahui layanan yang tersedia.

<b>Triggering Event</b>	Customer membuka halaman daftar layanan.	
<b>Brief Description</b>	Customer dapat melihat berbagai jenis layanan yang tersedia di sistem.	
<b>Actors</b>	Customer	
<b>Related Use</b>	Melakukan booking layanan	
<b>Stakeholders</b>	Customer	
<b>Preconditions</b>	Customer mengakses sistem.	
<b>Postconditions</b>	Daftar layanan ditampilkan ke customer.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Memilih menu “Daftar Layanan”	1.1 Menampilkan semua layanan yang tersedia
<b>Exception conditions</b>	Sistem gagal menampilkan daftar karena gangguan koneksi atau data kosong	

*Table 1 Usecase Scenario : Melihat Daftar Layanan*

- Usecase Scenario : Melakukan Booking Layanan

<b>Use Case</b>	Melakukan Booking Layanan	
<b>Scenario</b>	Customer ingin memesan jadwal layanan eyelash atau nail art secara online	
<b>Triggering Event</b>	Pelanggan membuka halaman booking dan memilih layanan	
<b>Brief Description</b>	Customer melakukan pemesanan layanan dengan memilih jenis layanan, tanggal, dan jam, lalu mengkonfirmasi booking	
<b>Actors</b>	Customer	
<b>Related Use</b>	Melihat daftar layanan, Membatalkan booking	
<b>Stakeholders</b>	Customer, Admin	
<b>Preconditions</b>	Customer sudah melihat daftar layanan.	
<b>Postconditions</b>	Data booking tersimpan dan menunggu konfirmasi.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Memilih layanan 2. Mengisi form booking	1.1 Menampilkan detail layanan 1.2 Menerima dan memvalidasi data

	3. Menekan tombol “Simpan”	1.3 Menyimpan data booking ke database
<b>Exception conditions</b>	Data tidak valid, form tidak lengkap, atau gagal simpan.	

*Table 2 Usecase Scenario : Melakukan Booking Layanan*

- Usecase Scenario : Membatalkan Booking

<b>Use Case</b>	Membatalkan Booking	
<b>Scenario</b>	Customer membatalkan booking yang telah dibuat sebelumnya.	
<b>Triggering Event</b>	Customer memilih menu riwayat booking dan klik “Batal”.	
<b>Brief Description</b>	Booking yang dipilih akan dibatalkan oleh sistem.	
<b>Actors</b>	Customer	
<b>Related Use</b>	Melakukan booking layanan	
<b>Stakeholders</b>	Customer, Admin	
<b>Preconditions</b>	Booking masih dalam status aktif/belum diproses.	
<b>Postconditions</b>	Status booking berubah menjadi “dibatalkan”.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Membuka riwayat booking 2. Memilih booking dan klik "Batal"	1.1 Menampilkan daftar booking 1.2 Mengubah status booking menjadi dibatalkan
<b>Exception conditions</b>	Booking sudah tidak bisa dibatalkan (misal: layanan sudah berjalan).	

*Table 3 Usecase Scenario : Membatalkan Booking*

- Usecase Scenario : Melihat Galeri

<b>Use Case</b>	Melihat Galeri
<b>Scenario</b>	Customer ingin melihat dokumentasi atau gambar layanan.
<b>Triggering Event</b>	Customer mengklik menu “Galeri”.
<b>Brief Description</b>	Sistem menampilkan foto-foto layanan atau hasil layanan.
<b>Actors</b>	Customer

<b>Related Use</b>	Tidak ada	
<b>Stakeholders</b>	Customer	
<b>Preconditions</b>	Customer sudah login dan berada di halaman utama.	
<b>Postconditions</b>	Galeri tampil di layar.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	3. Klik menu Galeri	1.1 Mengambil dan menampilkan foto layanan
<b>Exception conditions</b>	Galeri kosong atau gagal dimuat karena error server.	

*Table 4 Usecase Scenario : Melihat Galeri*

- Usecase Scenario : Memberi Ulasan

<b>Use Case</b>	Memberi Ulasan	
<b>Scenario</b>	Customer memberikan ulasan setelah menggunakan layanan.	
<b>Triggering Event</b>	Customer membuka riwayat booking dan memilih “Beri ulasan”.	
<b>Brief Description</b>	Customer menulis ulasan dan sistem menyimpannya.	
<b>Actors</b>	Customer	
<b>Related Use</b>	Melihat riwayat booking	
<b>Stakeholders</b>	Customer, Admin, Pemilik layanan	
<b>Preconditions</b>	Layanan sudah selesai digunakan.	
<b>Postconditions</b>	Ulasan tersimpan dan bisa dilihat oleh admin.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Membuka riwayat booking 2. Klik “Beri Ulasan” 3. Menulis ulasan dan submit	1.1 Menampilkan daftar booking 1.2 Menampilkan form ulasan 1.3 Menyimpan ulasan ke database
<b>Exception conditions</b>	Ulasan gagal disimpan, atau layanan belum selesai.	

*Table 5 Usecase Scenario : Memberi Ulasan*

- Usecase Scenario : Melihat Riwayat Booking

<b>Use Case</b>	Melihat Riwayat Booking	
<b>Scenario</b>	Customer melihat daftar booking yang sudah dilakukan.	
<b>Triggering Event</b>	Customer mengakses menu riwayat booking.	
<b>Brief Description</b>	Sistem menampilkan semua transaksi booking milik customer.	
<b>Actors</b>	Customer	
<b>Related Use</b>	Memberi ulasan, Membatalkan booking	
<b>Stakeholders</b>	Customer	
<b>Preconditions</b>	Customer telah melakukan minimal satu booking.	
<b>Postconditions</b>	Daftar booking tampil di layar.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Klik menu “Riwayat Booking”	1.1 Menampilkan daftar booking milik customer
<b>Exception conditions</b>	Tidak ada riwayat booking atau sistem gagal mengambil data.	

*Table 6 Usecase Scenario : Melihat Riwayat Booking*

- Usecase Scenario : Mengelola Layanan

<b>Use Case</b>	Mengelola Layanan	
<b>Scenario</b>	Admin menambah, mengedit, atau menghapus layanan.	
<b>Triggering Event</b>	Admin membuka menu layanan.	
<b>Brief Description</b>	Sistem memungkinkan admin untuk memodifikasi data layanan.	
<b>Actors</b>	Admin	
<b>Related Use</b>	Tidak ada	
<b>Stakeholders</b>	Admin, Customer	
<b>Preconditions</b>	Admin sudah login ke sistem.	
<b>Postconditions</b>	Layanan berhasil ditambah, diubah, atau dihapus.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Membuka menu layanan 2. Pilih tambah/edit/hapus	1.1 Menampilkan daftar layanan 1.2 Menampilkan form atau aksi

	3. Menyimpan perubahan	1.3 Memproses dan menyimpan data
<b>Exception conditions</b>	Gagal menyimpan data atau data layanan tidak valid.	

Table 7 Usecase Scenario : Mengelola Layanan

- Usecase Scenario : Mengelola Transaksi

<b>Use Case</b>	Mengelola Transaksi	
<b>Scenario</b>	Admin memverifikasi dan mengelola data transaksi pembayaran booking.	
<b>Triggering Event</b>	Admin membuka menu transaksi.	
<b>Brief Description</b>	Sistem menampilkan dan memungkinkan pengelolaan transaksi.	
<b>Actors</b>	Admin	
<b>Related Use</b>	Melakukan booking layanan	
<b>Stakeholders</b>	Admin, Customer	
<b>Preconditions</b>	Transaksi booking sudah dilakukan customer.	
<b>Postconditions</b>	Status transaksi diperbarui (misal: terverifikasi).	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Membuka menu transaksi 2. Memilih transaksi 3. Mengubah status (verifikasi/dibatalkan)	1.1 Menampilkan daftar transaksi 1.2 Menampilkan detail transaksi 1.3 Menyimpan perubahan status
<b>Exception conditions</b>	Data transaksi tidak ditemukan atau terjadi kesalahan penyimpanan.	

Table 8 Usecase Scenario : Mengelola Transaksi

- Usecase Scenario : Login Admin

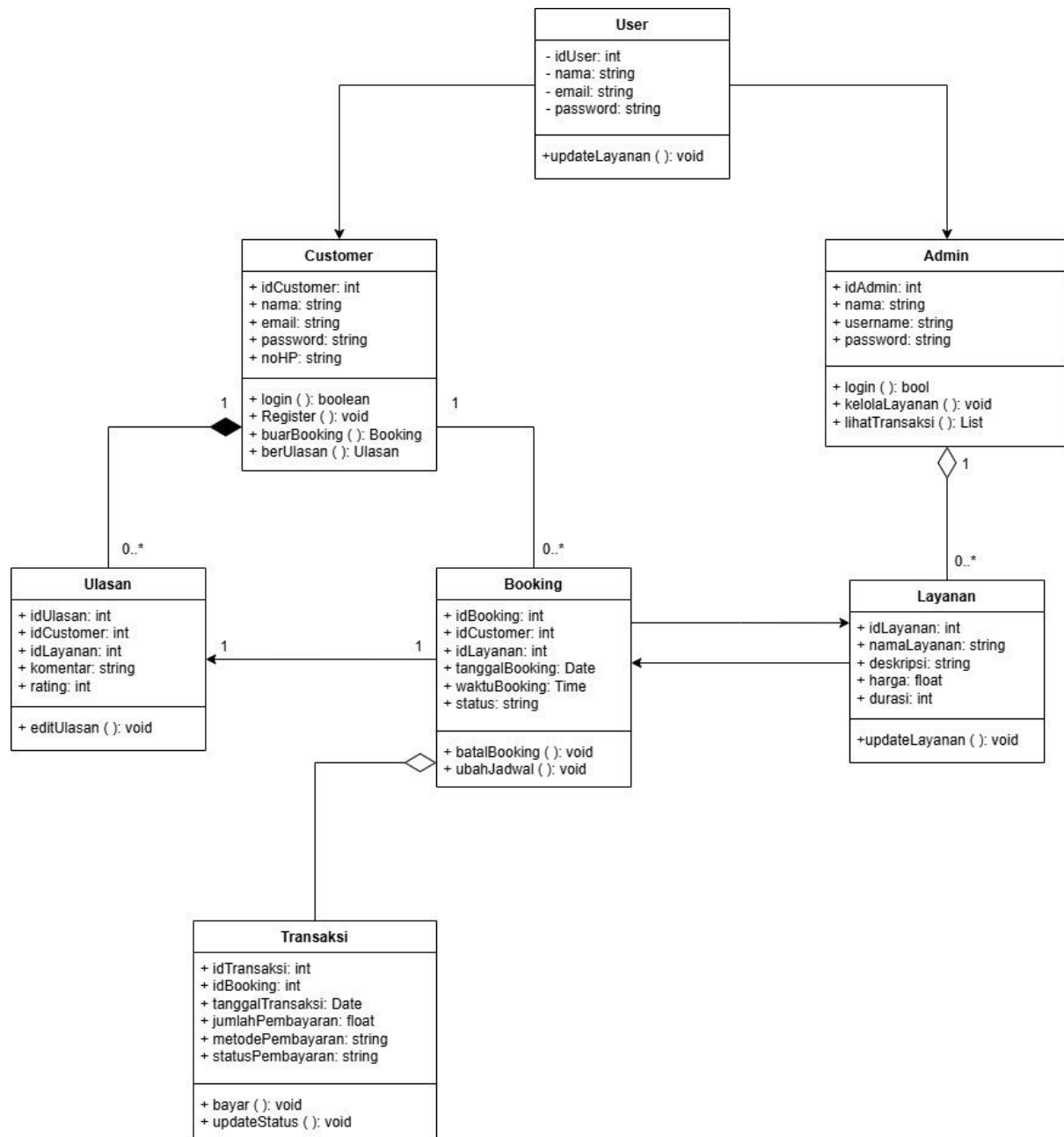
<b>Use Case</b>	Login Admin
<b>Scenario</b>	Admin masuk ke sistem untuk mengakses fitur manajemen.
<b>Triggering Event</b>	Admin membuka halaman login dan memasukkan data.
<b>Brief Description</b>	Sistem melakukan autentikasi berdasarkan username dan password admin.

<b>Actors</b>	Admin	
<b>Related Use</b>	Melakukan booking layanan, Mengelola Transaksi	
<b>Stakeholders</b>	Admin	
<b>Preconditions</b>	Admin belum Login.	
<b>Postconditions</b>	Admin berhasil login dan diarahkan ke dashboard.	
<b>Flow of activities</b>	<b>Actor</b>	<b>System</b>
	1. Membuka halaman login 2. Mengisi username dan password 3. Menekan tombol “Login”	1.1 Menampilkan form login 1.2 Memeriksa username dan password 1.3 Mengarahkan ke halaman utama jika berhasil
<b>Exception conditions</b>	Username/password salah, akun tidak terdaftar.	

*Table 9 Usecase Scenario : Login Admin*



### 3.3 Class Diagram



Gambar 2 Class Diagram

#### Penjelasan

##### 1. User:

- Kelas ini merepresentasikan pengguna secara umum.
- Atribut:
  - `idUser`: ID unik untuk pengguna.

- nama: Nama pengguna.
- email: Alamat email pengguna.
- password: Kata sandi untuk autentikasi.
- Metode:
  - updateLayanan(): Untuk memperbarui informasi layanan.

## 2. **Customer:**

- Subkelas dari User yang khusus untuk pelanggan.
- Atribut tambahan:
  - idCustomer: ID unik untuk pelanggan.
  - noHP: Nomor telepon pelanggan.
- Metode:
  - login(): Autentikasi pelanggan.
  - register(): Registrasi akun baru.
  - buatBooking(): Membuat booking baru.
  - beriUlasan(): Menambahkan ulasan untuk layanan.

## 3. **Admin:**

- Subkelas lain dari User yang memiliki hak akses lebih.
- Atribut tambahan:
  - idAdmin: ID unik untuk admin.
  - username: Nama pengguna admin.
- Metode:
  - login(): Autentikasi admin.
  - kelolaLayanan(): Mengelola layanan yang tersedia.
  - lihatTransaksi(): Melihat daftar transaksi.

## 4. **Ulasan:**

- Kelas ini digunakan untuk menyimpan ulasan yang diberikan pelanggan.
- Atribut:
  - idUlasan: ID unik ulasan.
  - idCustomer: ID pelanggan yang memberikan ulasan.
  - idLayanan: ID layanan yang diulas.
  - komentar: Isi ulasan.
  - rating: Penilaian dalam bentuk angka.
- Metode:
  - editUlasan(): Mengubah isi ulasan yang sudah dibuat.

## 5. **Booking:**

- Digunakan untuk mencatat pemesanan layanan.
- Atribut:
  - idBooking: ID unik untuk setiap booking.
  - idCustomer: ID pelanggan yang membuat booking.
  - idLayanan: ID layanan yang dipesan.
  - tanggalBooking: Tanggal pemesanan.
  - waktuBooking: Waktu pemesanan.
  - status: Status booking (misalnya "aktif" atau "dibatalkan").
- Metode:
  - batalBooking(): Membatalkan pemesanan.
  - ubahJadwal(): Mengubah jadwal pemesanan.

## 6. **Layanan:**

- Mencatat layanan yang tersedia.
- Atribut:
  - idLayanan: ID unik untuk layanan.
  - namaLayanan: Nama layanan.
  - deskripsi: Penjelasan tentang layanan.
  - harga: Biaya untuk layanan.
  - durasi: Lama waktu layanan (dalam menit).
- Metode:
  - updateLayanan(): Memperbarui informasi layanan.

## 7. **Transaksi:**

- Berisi informasi pembayaran untuk booking.
- Atribut:
  - idTransaksi: ID unik untuk transaksi.
  - idBooking: ID booking terkait.
  - tanggalTransaksi: Tanggal pembayaran dilakukan.
  - jumlahPembayaran: Total biaya yang dibayar.
  - metodePembayaran: Cara pembayaran (misalnya "transfer bank").
  - statusPembayaran: Status pembayaran (misalnya "berhasil" atau "pending").
- Metode:

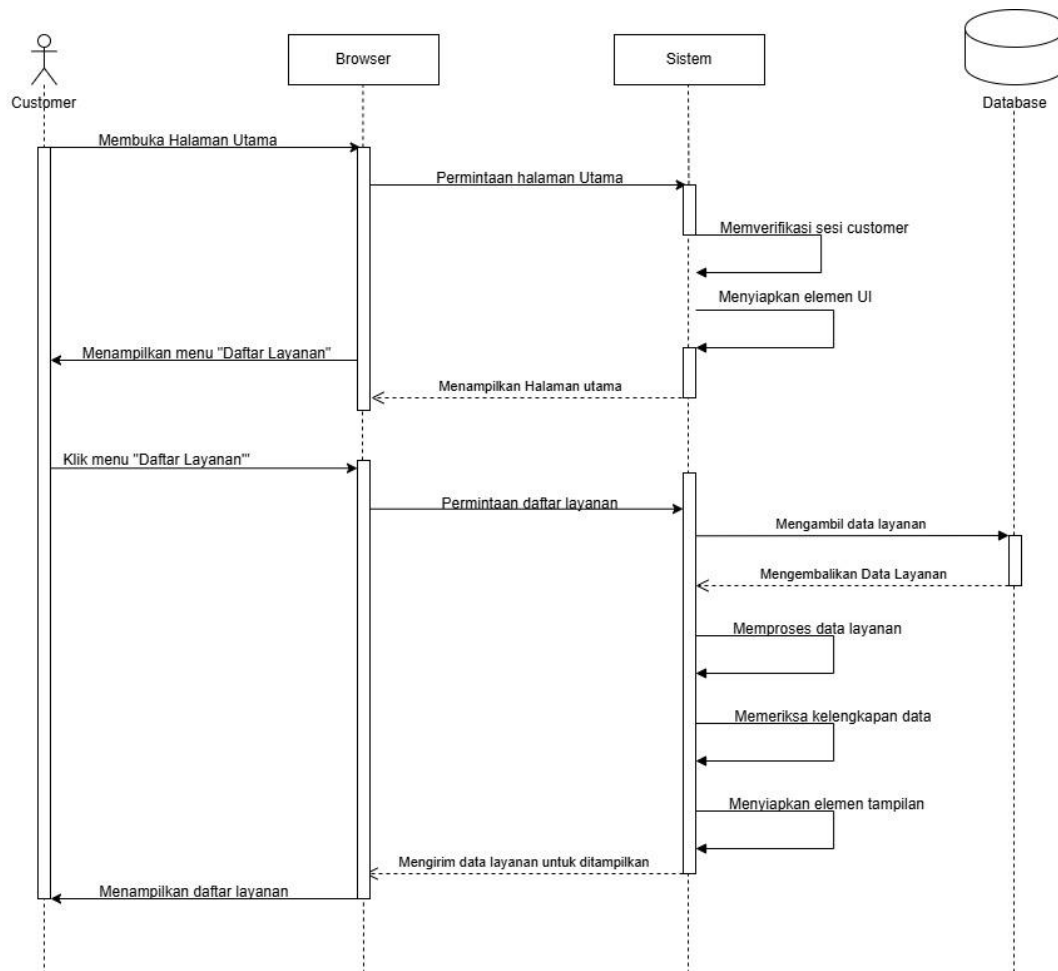
- `bayar()`: Melakukan pembayaran.
- `updateStatus()`: Mengubah status pembayaran.

#### **Relasi Antar Kelas:**

- **Customer** memiliki relasi ke **Ulasan** (banyak) karena satu pelanggan bisa memberikan banyak ulasan.
- **Customer** juga terhubung ke **Booking**, menunjukkan bahwa satu pelanggan bisa membuat beberapa pemesanan.
- **Booking** terhubung ke **Layanan**, menandakan setiap pemesanan terkait dengan satu layanan.
- **Booking** juga terkait ke **Transaksi**, menggambarkan bahwa setiap pemesanan memiliki satu transaksi pembayaran.
- **Admin** memiliki tanggung jawab untuk mengelola **Layanan** dan melihat **Transaksi**.

### 3.3 Sequence Diagram

#### Sequence Diagram : Melihat Daftar Layanan



Gambar 3 Sequence Diagram : Melihat Daftar Layanan

#### Penjelasan

##### 1. Aktor dan Komponen Sistem

Diagram ini menggambarkan proses interaksi antara empat komponen utama dalam sistem layanan berbasis web, yaitu:

- **Customer** (pengguna): Aktor utama yang berinteraksi dengan sistem untuk melihat layanan.
- **Browser**: Perangkat lunak antarmuka yang digunakan oleh customer untuk mengakses sistem.
- **Sistem**: Backend server atau aplikasi yang menangani proses logika dan pengolahan data.
- **Database**: Komponen penyimpanan data yang berisi informasi layanan yang tersedia.

##### 2. Urutan Proses Interaksi

#### **a. Customer Membuka Halaman Utama**

Langkah awal dimulai ketika customer membuka aplikasi melalui browser. Aksi ini memicu pengiriman permintaan ke server (sistem) untuk membuka halaman utama.

#### **b. Browser Mengirim Permintaan Halaman Utama ke Sistem**

Browser mengirim request ke sistem berupa permintaan untuk menampilkan halaman utama.

#### **c. Sistem Memverifikasi Sesi Customer**

Setelah menerima permintaan, sistem melakukan verifikasi sesi pengguna. Sistem memeriksa apakah customer sudah login atau memiliki sesi yang valid agar dapat mengakses konten.

#### **d. Sistem Menyiapkan Elemen UI**

Jika verifikasi berhasil, sistem menyiapkan elemen-elemen User Interface (UI) yang diperlukan untuk halaman utama, seperti menu, banner, dan lainnya.

#### **e. Sistem Menampilkan Halaman Utama ke Browser**

Sistem mengirim kembali hasil persiapan elemen UI dalam bentuk halaman utama ke browser.

#### **f. Browser Menampilkan Menu “Daftar Layanan”**

Setelah halaman utama dimuat, salah satu elemen UI yang ditampilkan adalah menu “Daftar Layanan”. Menu ini kini dapat dilihat dan diklik oleh customer.

#### **g. Customer Klik Menu “Daftar Layanan”**

Customer kemudian melakukan klik pada menu “Daftar Layanan”, yang akan memicu permintaan baru dari browser ke sistem.

#### **h. Browser Mengirim Permintaan Daftar Layanan ke Sistem**

Browser mengirim request ke sistem untuk meminta daftar layanan yang tersedia pada sistem.

#### **i. Sistem Mengambil Data Layanan dari Database**

Sistem kemudian mengambil data layanan dari database dengan melakukan query atau pengambilan data dari tabel layanan.

#### **j. Database Mengembalikan Data Layanan ke Sistem**

Database merespons permintaan sistem dengan mengirimkan data layanan yang telah diminta.

#### **k. Sistem Memproses Data Layanan**

Setelah data layanan diterima, sistem mulai memproses data tersebut. Proses ini bisa meliputi filtering, formatting, atau sorting data agar sesuai kebutuhan tampilan.

#### **l. Sistem Memeriksa Kelengkapan Data**

Sistem juga melakukan pemeriksaan terhadap kelengkapan dan validitas data. Hal ini penting untuk memastikan bahwa data yang akan ditampilkan tidak ada yang kosong atau corrupt.

#### **m. Sistem Menyiapkan Elemen Tampilan**

Setelah data dipastikan lengkap dan valid, sistem menyiapkan tampilan daftar layanan dalam format yang sesuai (HTML atau JSON) agar bisa ditampilkan di browser.

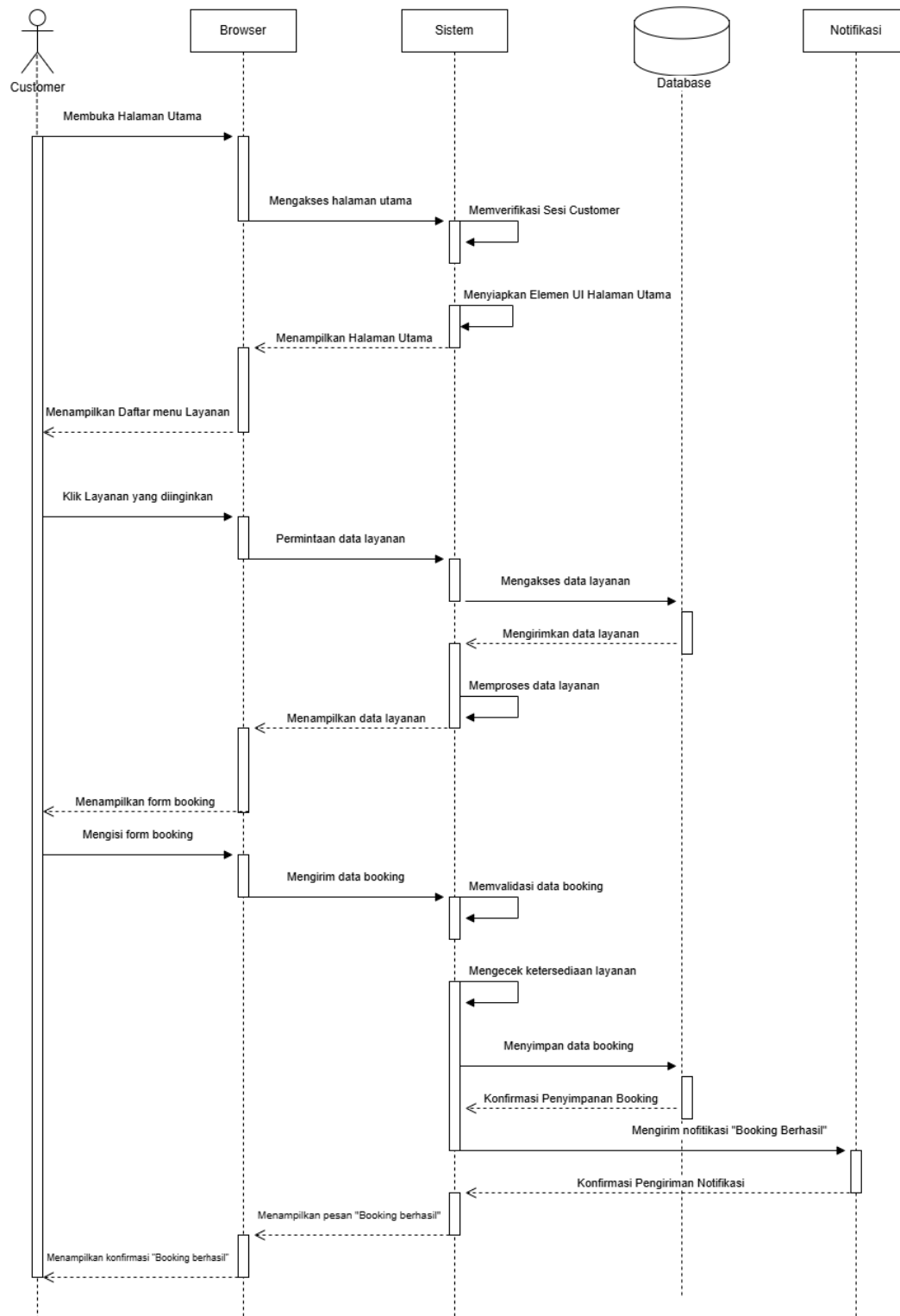
#### **n. Sistem Mengirim Data Layanan ke Browser**

Sistem mengirimkan data yang sudah diformat ke browser sebagai hasil akhir proses backend.

#### **o. Browser Menampilkan Daftar Layanan**

Browser menerima data tersebut dan menampilkan daftar layanan kepada customer di halaman browser.

## Sequence Diagram : Melakukan Booking Layanan



Gambar 4 Sequence Diagram : Melakukan Booking Layanan



## Penjelasan

### 1. Customer membuka halaman utama

- Customer memulai dengan membuka aplikasi melalui browser.

### 2. Browser mengakses halaman utama

- Browser mengirim permintaan (request) ke sistem agar dapat menampilkan halaman utama.

### 3. Sistem memverifikasi sesi customer

- Sistem melakukan verifikasi terhadap sesi pengguna (misalnya login atau autentikasi token).

### 4. Sistem menyiapkan elemen UI halaman utama

- Sistem menyiapkan berbagai elemen antarmuka pengguna (UI) seperti tombol, menu, dan konten halaman utama.

### 5. Menampilkan halaman utama ke browser

- UI dikirim kembali ke browser, kemudian browser menampilkan halaman utama tersebut ke customer.

### 6. Customer melihat daftar menu layanan

- Pada halaman utama, customer melihat daftar menu layanan yang tersedia.

### 7. Customer mengklik layanan yang diinginkan

- Ketika customer mengklik salah satu layanan, browser mengirim permintaan data detail layanan ke sistem.

### 8. Permintaan data layanan ke sistem

- Sistem menerima permintaan data dan mengakses data layanan yang sesuai dari database.

### 9. Database mengirimkan data layanan ke sistem

- Sistem memproses data yang diterima dan menyiapkan tampilan informasi layanan.

### 10. Sistem menampilkan data layanan

- Browser menampilkan informasi detail layanan berdasarkan respons dari sistem.

### 11. Customer mengisi form booking

- Setelah melihat detail layanan, customer mengisi form booking (misalnya memilih tanggal, jam, dan informasi lainnya).

### 12. Browser mengirim data booking ke sistem

- Data booking yang telah diisi customer dikirim ke sistem untuk diproses.

13. Sistem memvalidasi data booking

- Sistem memverifikasi apakah data booking valid, seperti format tanggal, isian yang tidak kosong, dsb.

14. Sistem mengecek ketersediaan layanan

- Sistem memeriksa di database apakah slot layanan tersedia sesuai permintaan customer.

15. Sistem menyimpan data booking

- Jika tersedia, sistem menyimpan data booking ke dalam database.

16. Sistem mengirim konfirmasi penyimpanan booking

- Setelah data tersimpan, sistem memberi konfirmasi bahwa booking berhasil.

17. Sistem mengirim notifikasi "Booking Berhasil"

- Sistem juga mengirim notifikasi (via email, push notification, dsb) bahwa booking telah berhasil dilakukan.

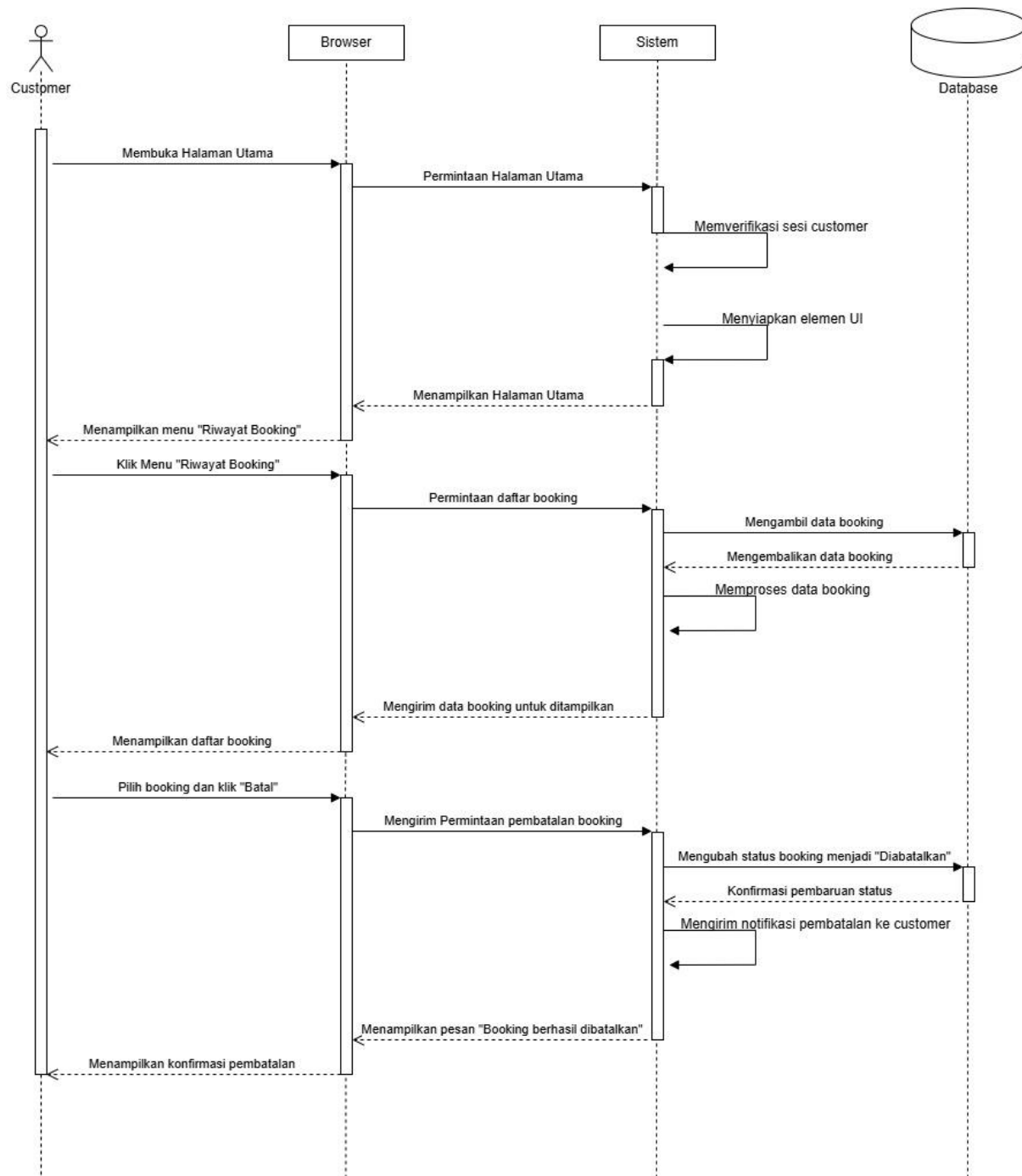
18. Sistem notifikasi mengirim konfirmasi pengiriman

- Sistem notifikasi mengonfirmasi bahwa pesan notifikasi berhasil dikirim ke customer.

19. Browser menampilkan pesan "Booking berhasil"

- Setelah seluruh proses selesai, browser menampilkan pesan bahwa booking berhasil dilakukan kepada customer.

## Sequence Diagram : Membatalkan Booking



Gambar 5 Sequence Diagram : Membatalkan Booking

## **Penjelasan**

### **Aktor & Komponen:**

1. **Customer** – Pengguna sistem yang ingin membatalkan booking.
2. **Browser** – Media antarmuka tempat customer berinteraksi.
3. **Sistem** – Server backend yang memproses permintaan.
4. **Database** – Menyimpan data riwayat booking layanan.

### **Penjelasan Langkah demi Langkah**

#### **1. Customer membuka halaman utama**

- Customer memulai proses dengan membuka aplikasi melalui browser.

#### **2. Browser mengirim permintaan halaman utama ke sistem**

- Browser meminta halaman utama agar dapat ditampilkan.

#### **3. Sistem memverifikasi sesi customer**

- Sistem mengecek apakah customer telah login atau memiliki sesi yang valid.

#### **4. Sistem menyiapkan elemen UI**

- Sistem menyiapkan antarmuka halaman utama seperti menu navigasi termasuk menu "Riwayat Booking".

#### **5. Sistem mengirim tampilan halaman utama ke browser**

- Browser menampilkan halaman utama, termasuk menu "Riwayat Booking".

#### **6. Customer klik menu “Riwayat Booking”**

- Customer mengakses daftar riwayat booking yang pernah dilakukan.

#### **7. Browser mengirim permintaan daftar booking ke sistem**

- Sistem menerima permintaan dan memproses data yang diminta dari database.

#### **8. Sistem mengambil data booking dari database**

- Sistem men-query database untuk mendapatkan seluruh riwayat booking customer tersebut.

#### **9. Database mengembalikan data booking ke sistem**

- Data dikembalikan ke sistem.

#### **10. Sistem memproses data booking**

- Data diolah agar bisa ditampilkan dengan baik ke browser.

#### **11. Sistem mengirim data booking ke browser**

- Browser menampilkan daftar riwayat booking customer.

**12. Customer memilih booking yang ingin dibatalkan dan klik tombol “Batal”**

- Customer memutuskan untuk membatalkan salah satu booking dan mengklik tombol yang sesuai.

**13. Browser mengirim permintaan pembatalan booking ke sistem**

- Permintaan dikirim ke backend untuk memproses pembatalan.

**14. Sistem mengubah status booking menjadi “Dibatalkan”**

- Sistem mengupdate data booking pada database menjadi status dibatalkan.

**15. Sistem mengirim konfirmasi pembaruan status**

- Sistem mengonfirmasi bahwa proses update telah berhasil dilakukan.

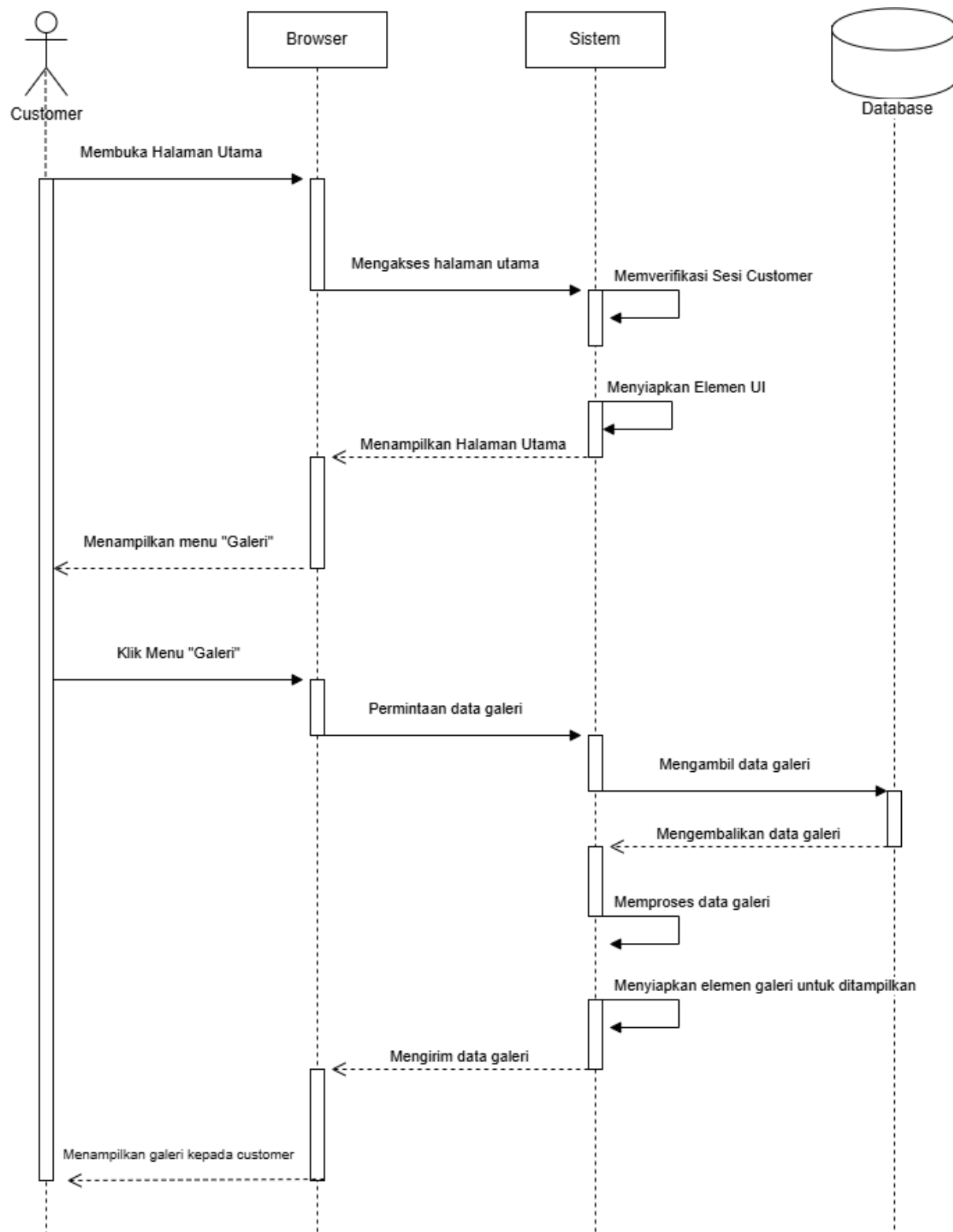
**16. Sistem mengirim notifikasi pembatalan ke customer**

- Sistem dapat mengirimkan notifikasi melalui email atau dalam aplikasi bahwa booking telah berhasil dibatalkan.

**17. Browser menampilkan pesan “Booking berhasil dibatalkan”**

- Sebagai umpan balik, sistem mengirimkan pesan ke browser, dan browser menampilkannya kepada customer.

## Sequence Diagram : Melihat Galeri



Gambar 6 Sequence Diagram : Melihat Galeri

## Penjelasan

### Aktor & Komponen Sistem:

Diagram ini menggambarkan interaksi antara:

1. **Customer** – Pengguna akhir yang ingin melihat galeri.
2. **Browser** – Antarmuka pengguna berbasis web.
3. **Sistem** – Backend server yang memproses permintaan dan menyiapkan data.
4. **Database** – Penyimpanan gambar atau informasi konten galeri.

### Penjelasan Langkah demi Langkah

#### 1. Customer membuka halaman utama

- Customer membuka aplikasi melalui browser untuk mulai menjelajah fitur sistem.

#### 2. Browser mengirim permintaan akses halaman utama

- Browser meminta sistem untuk menampilkan halaman utama aplikasi.

#### 3. Sistem memverifikasi sesi customer

- Sistem mengecek apakah customer sudah login atau memiliki sesi aktif.
- Jika tidak valid, biasanya akan diarahkan ke halaman login.

#### 4. Sistem menyiapkan elemen UI halaman utama

- Sistem membuat elemen-elemen tampilan seperti menu navigasi termasuk menu “Galeri”.

#### 5. Browser menampilkan halaman utama

- Setelah elemen disiapkan, browser menampilkannya ke customer, termasuk menu “Galeri”.

#### 6. Customer mengklik menu “Galeri”

- Customer memilih fitur galeri dengan mengklik menu yang telah disediakan.

#### 7. Browser mengirim permintaan data galeri

- Browser mengirim request ke sistem untuk menampilkan konten galeri layanan.

#### 8. Sistem mengambil data galeri dari database

- Sistem mengambil data yang berkaitan dengan galeri (gambar, judul, deskripsi) dari database.

#### 9. Database mengembalikan data galeri ke sistem

- Data galeri dikirim kembali ke sistem untuk diproses dan ditampilkan.

#### **10. Sistem memproses data galeri**

- Sistem memformat, menyusun, atau memfilter data galeri agar siap ditampilkan di browser.

#### **11. Sistem menyiapkan elemen tampilan galeri**

- Sistem menyusun elemen-elemen visual (gambar, layout) dalam bentuk HTML atau objek JSON.

#### **12. Sistem mengirim data galeri ke browser**

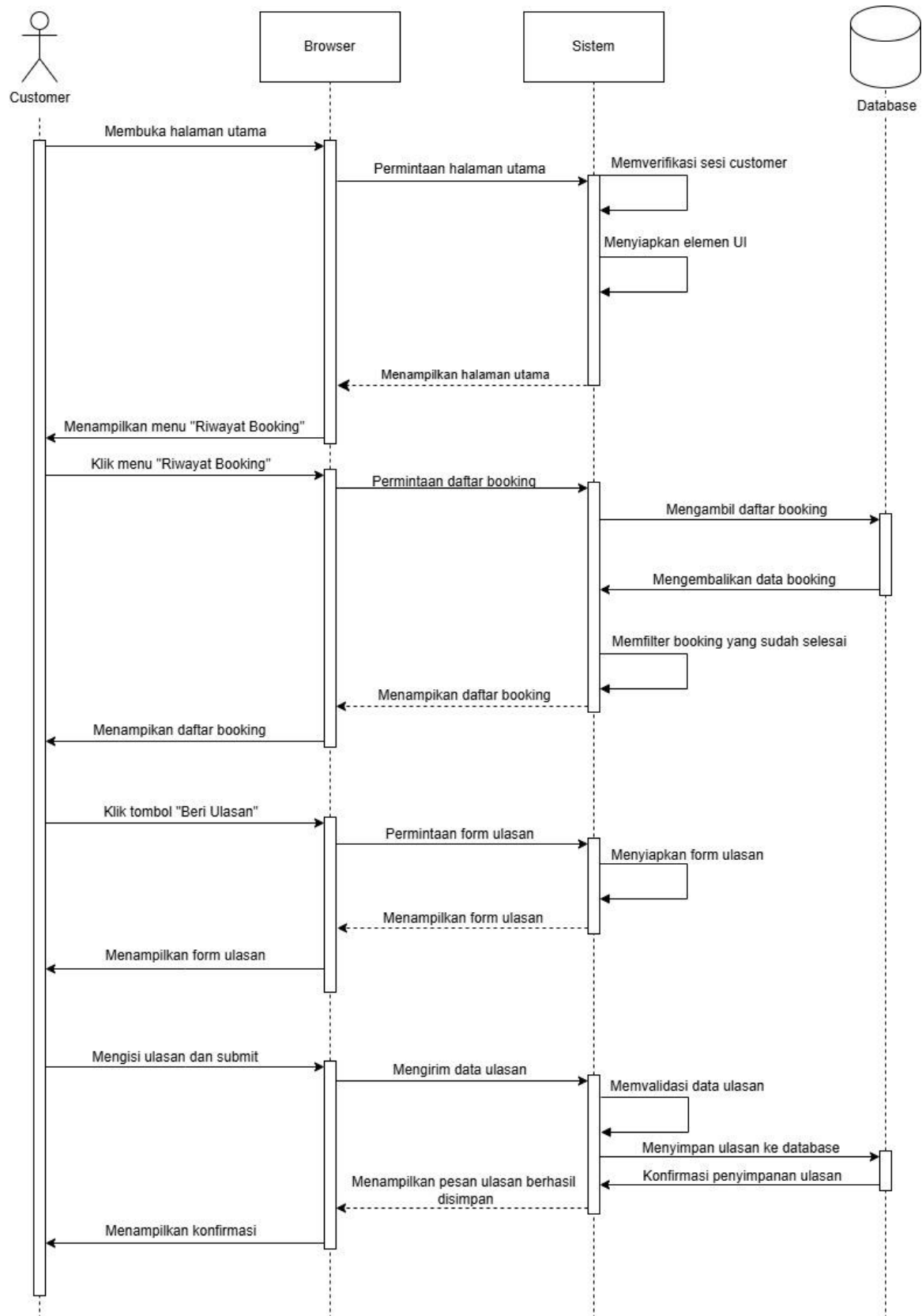
- Browser menerima data hasil pengolahan dari sistem.

#### **13. Browser menampilkan galeri kepada customer**

- Galeri ditampilkan ke layar customer dengan tampilan visual yang sesuai desain.



## Sequence Diagram : Memberi Ulasan



## **Penjelasan**

### **Aktor & Komponen**

- **Customer:** Pengguna yang akan memberikan ulasan atas layanan.
- **Browser:** Media antarmuka pengguna.
- **Sistem:** Server yang menangani logika dan pengolahan data.
- **Database:** Tempat penyimpanan data booking dan ulasan.

### **Penjelasan Langkah demi Langkah**

#### **1. Customer membuka halaman utama**

- Customer memulai dengan mengakses sistem melalui browser.

#### **2. Browser mengirim permintaan halaman utama**

- Permintaan dikirim ke sistem agar halaman utama bisa ditampilkan.

#### **3. Sistem memverifikasi sesi customer**

- Sistem melakukan autentikasi atau pengecekan sesi aktif untuk memastikan keamanan akses.

#### **4. Sistem menyiapkan elemen UI halaman utama**

- UI disiapkan oleh sistem (termasuk menu "Riwayat Booking").

#### **5. Halaman utama ditampilkan ke browser**

- Browser menerima dan menampilkan tampilan halaman utama ke customer.

#### **6. Customer klik menu "Riwayat Booking"**

- Customer mengakses riwayat booking untuk memilih layanan yang sudah digunakan.

#### **7. Browser mengirim permintaan daftar booking ke sistem**

- Sistem menerima permintaan tersebut dan mulai mengambil data dari database.

#### **8. Sistem mengambil daftar booking dari database**

- Sistem mengakses tabel booking untuk mengambil semua riwayat milik customer.

#### **9. Database mengembalikan data booking**

- Sistem menerima data riwayat booking dari database.

#### **10. Sistem memfilter booking yang sudah selesai**

- Hanya layanan yang sudah selesai akan ditampilkan untuk bisa diberikan ulasan.

#### **11. Sistem menampilkan daftar booking ke browser**

- Browser menampilkan daftar layanan yang sudah selesai.

#### **12. Customer klik tombol "Beri Ulasan"**

- Customer memilih layanan tertentu dan mengklik tombol untuk memberi ulasan.

#### **13. Browser mengirim permintaan form ulasan ke sistem**

- Permintaan form dikirim agar sistem menyediakan tempat untuk menulis ulasan.

#### **14. Sistem menyiapkan dan menampilkan form ulasan**

- Formulir disiapkan oleh sistem dan dikirim kembali ke browser.

#### **15. Customer mengisi ulasan dan submit**

- Customer menuliskan komentarnya dan menekan tombol kirim.

#### **16. Browser mengirim data ulasan ke sistem**

- Data yang sudah diisi dikirim ke sistem untuk diproses.

#### **17. Sistem memvalidasi data ulasan**

- Sistem mengecek apakah data ulasan sesuai format (tidak kosong, panjang karakter sesuai, dsb.).

#### **18. Sistem menyimpan ulasan ke database**

- Jika valid, sistem menyimpan ulasan tersebut ke tabel ulasan.

#### **19. Sistem memberikan konfirmasi penyimpanan**

- Database merespons bahwa data berhasil disimpan.

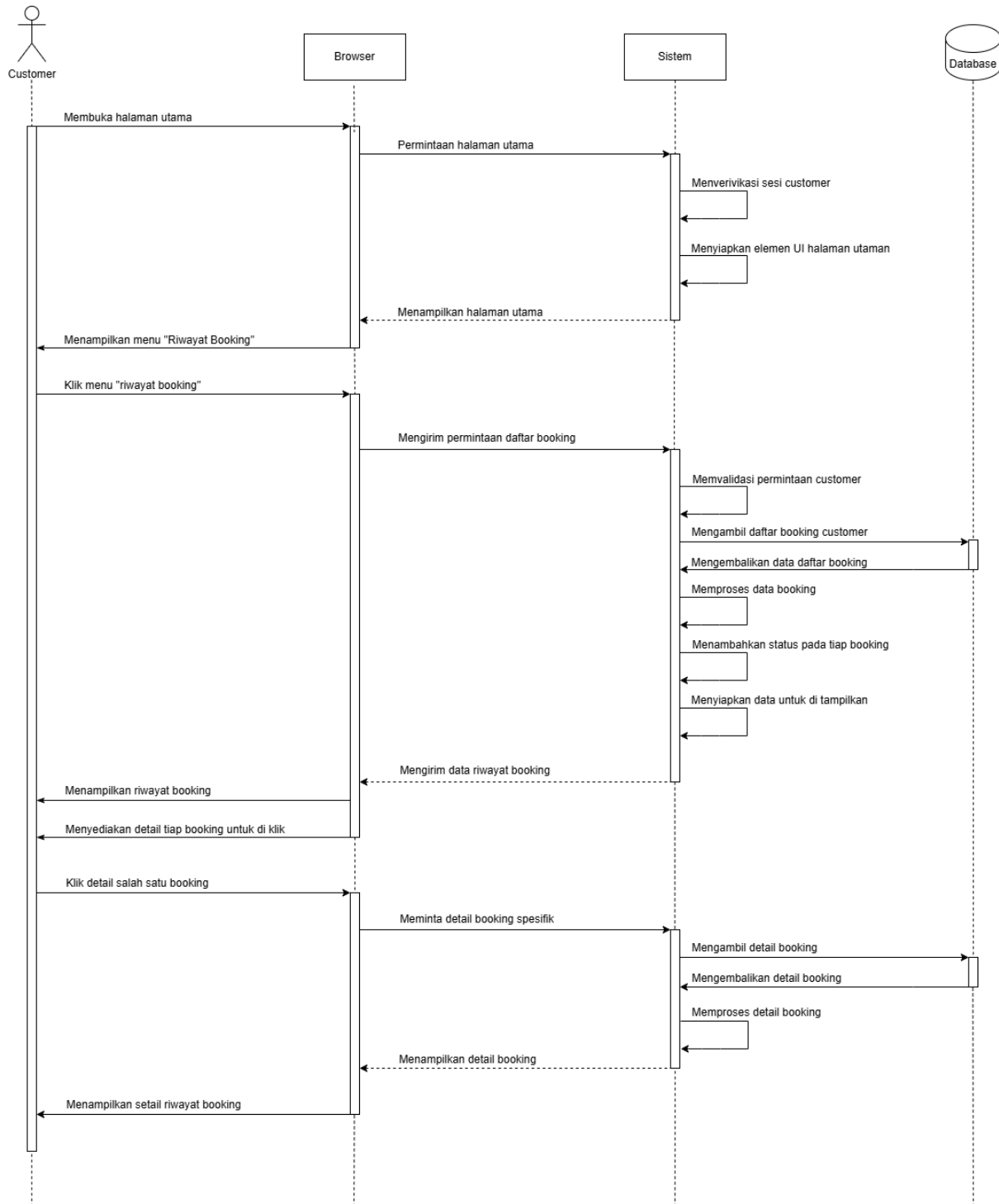
#### **20. Sistem menampilkan pesan ulasan berhasil disimpan**

- Customer mendapatkan umpan balik bahwa ulasan berhasil dikirim.

#### **21. Browser menampilkan pesan konfirmasi**

- Pesan konfirmasi terlihat oleh customer, menandakan bahwa proses selesai.

## Sequence Diagram : Melihat Riwayat Booking



Gambar 8 Sequence Diagram : Melihat Riwayat Booking

## **Penjelasan**

### **Aktor & Komponen Sistem**

- **Customer** – Pengguna yang ingin login ke dalam sistem.
- **Browser** – Antarmuka yang digunakan customer.
- **Sistem** – Backend server yang memproses login.
- **Database** – Tempat penyimpanan data akun customer.

### **Penjelasan Langkah demi Langkah**

#### **1. Customer membuka halaman login**

- Customer mengakses halaman login melalui browser.

#### **2. Browser mengirim permintaan halaman login ke sistem**

- Browser meminta halaman login untuk ditampilkan ke customer.

#### **3. Sistem menyiapkan elemen UI halaman login**

- Sistem menyiapkan form login (input username/email dan password).

#### **4. Browser menampilkan form login**

- Halaman login ditampilkan ke layar customer.

#### **5. Customer mengisi form login dan submit**

- Customer memasukkan data seperti email/username dan password.

#### **6. Browser mengirim data login ke sistem**

- Data login dikirimkan ke server untuk divalidasi.

#### **7. Sistem memvalidasi data login**

- Sistem mengecek apakah field telah diisi dan dalam format yang sesuai.

#### **8. Sistem mengakses database untuk mencocokkan akun**

- Sistem melakukan query ke database untuk mencocokkan email dan password yang dikirim.

#### **9. Database mengembalikan data akun**

- Data user yang sesuai (jika ada) dikembalikan ke sistem.

#### **10. Sistem memverifikasi kecocokan akun**

- Sistem membandingkan apakah email dan password cocok.

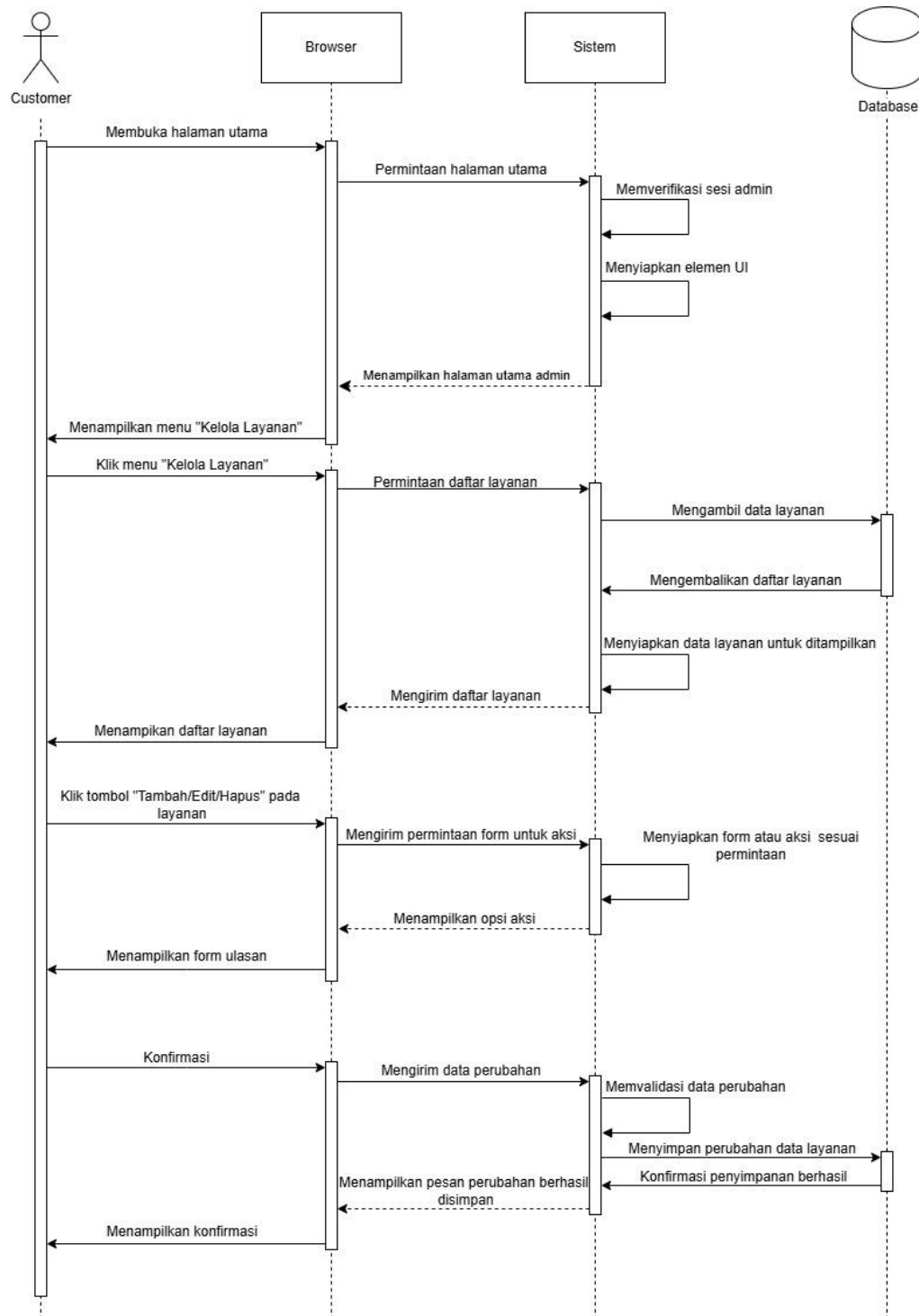
**11. Jika berhasil, sistem menyiapkan sesi login**

- Sistem akan menyimpan sesi user dan menyatakan login berhasil.

**12. Sistem mengarahkan customer ke halaman utama**

- Setelah login berhasil, browser akan diarahkan ke halaman utama.

## Sequence Diagram : Mengelola Layanan



Gambar 9 Sequence Diagram : Mengelola Layanan

## Penjelasan

### Aktor & Komponen Sistem

- **Admin (seharusnya bukan "Customer")** – Pengguna yang memiliki hak untuk mengelola data layanan.
- **Browser** – Antarmuka pengguna yang digunakan untuk interaksi.
- **Sistem** – Backend server yang memproses permintaan dari admin.
- **Database** – Tempat penyimpanan daftar layanan dan data perubahan.

### Penjelasan Langkah demi Langkah

#### 1. Admin membuka halaman utama

- Admin mengakses halaman awal sistem melalui browser.

#### 2. Browser mengirim permintaan halaman utama ke sistem

- Sistem menerima permintaan untuk menampilkan halaman admin.

#### 3. Sistem memverifikasi sesi admin

- Sistem memverifikasi apakah pengguna memiliki hak sebagai admin (autentikasi + otorisasi).

#### 4. Sistem menyiapkan elemen UI untuk halaman utama

- Sistem menyiapkan menu dan fitur yang hanya bisa diakses oleh admin seperti “Kelola Layanan”.

#### 5. Browser menampilkan halaman utama admin

- Browser menampilkan UI halaman utama versi admin.

#### 6. Admin klik menu “Kelola Layanan”

- Admin memilih fitur manajemen layanan.

#### 7. Browser mengirim permintaan daftar layanan ke sistem

- Sistem menerima request untuk mengambil data layanan.

#### 8. Sistem mengambil data layanan dari database

- Sistem mengambil data seluruh layanan yang sudah ada.

#### 9. Database mengembalikan data layanan ke sistem

- Data layanan dikembalikan untuk diproses.

#### 10. Sistem menyiapkan data layanan untuk ditampilkan



- Data disusun agar mudah dibaca dalam UI.

#### **11. Browser menampilkan daftar layanan**

- Admin dapat melihat daftar layanan yang tersedia.

#### **12. Admin klik tombol “Tambah/Edit/Hapus” pada layanan**

- Admin memilih salah satu aksi pengelolaan layanan.

#### **13. Browser mengirim permintaan form untuk aksi tersebut**

- Permintaan dikirim untuk membuka form pengisian (jika tambah/edit) atau konfirmasi (jika hapus).

#### **14. Sistem menyiapkan form atau opsi aksi**

- Sistem memuat form atau dialog aksi sesuai permintaan.

#### **15. Browser menampilkan form aksi**

- Admin melihat dan mengisi atau mengonfirmasi aksi.

#### **16. Admin klik tombol konfirmasi**

- Setelah form terisi atau dikonfirmasi, admin mengirimkan perubahan.

#### **17. Browser mengirim data perubahan ke sistem**

- Perubahan data dikirim ke server.

#### **18. Sistem memvalidasi data perubahan**

- Sistem mengecek apakah data valid (format, kelengkapan, dll).

#### **19. Sistem menyimpan perubahan ke database**

- Jika valid, data diubah atau ditambahkan atau dihapus di database.

#### **20. Database mengonfirmasi penyimpanan berhasil**

- Database mengirim respons berhasil menyimpan data.

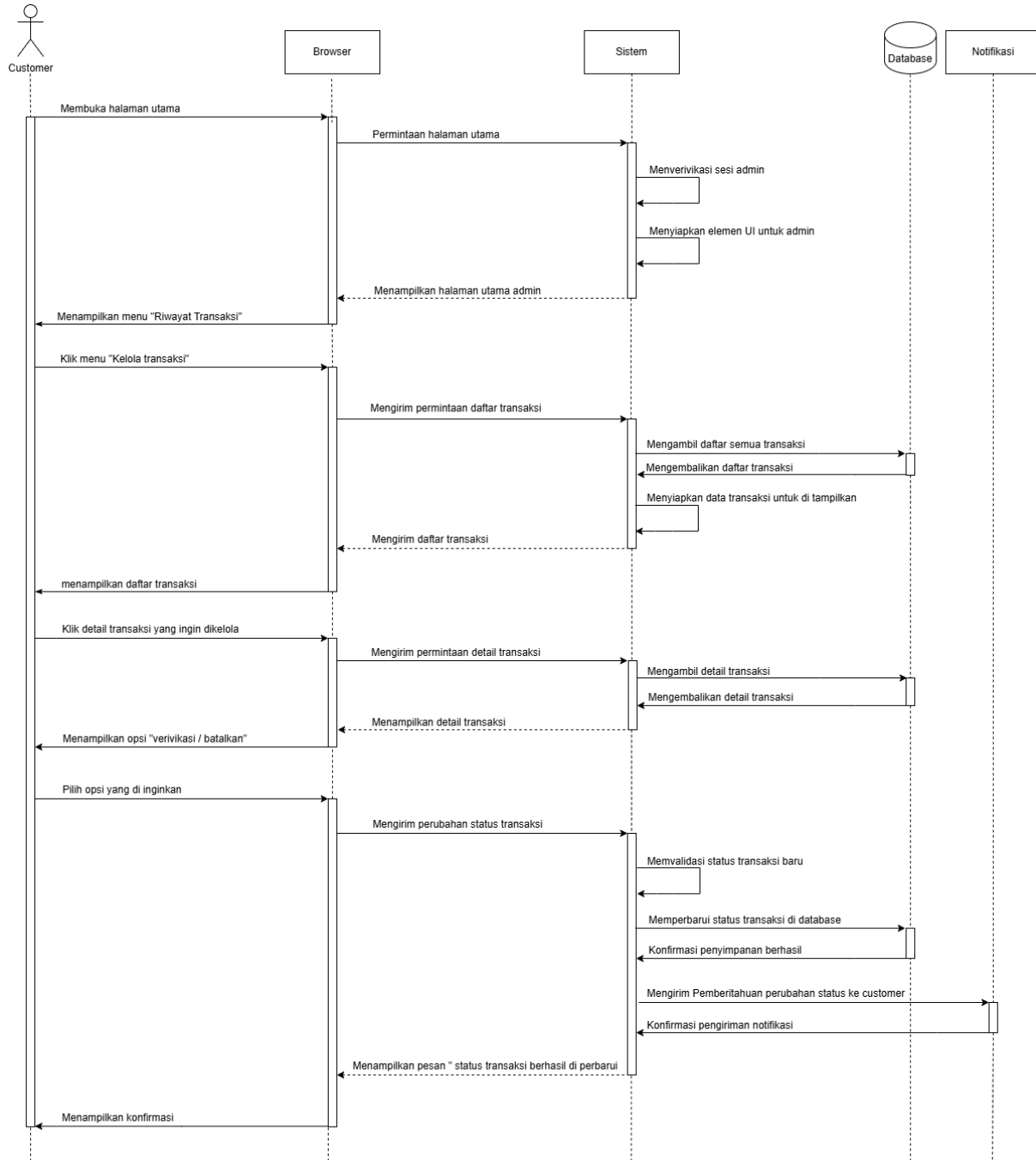
#### **21. Sistem menampilkan pesan perubahan berhasil disimpan**

- Sistem mengirimkan feedback ke browser.

#### **22. Browser menampilkan pesan konfirmasi**

- Admin melihat pesan bahwa perubahan berhasil dilakukan.

## Sequence Diagram : Mengelola Transaksi



Gambar 10 Sequence Diagram : Mengelola Transaksi

## Penjelasan

### Aktor & Komponen Sistem

- **Admin** – Pengguna dengan hak akses untuk membatalkan booking customer.
- **Browser** – Antarmuka pengguna (UI) yang digunakan oleh admin.
- **Sistem** – Backend yang memproses permintaan dari admin.
- **Database** – Penyimpan data booking.
- **Notifikasi** – Sistem pengiriman pesan pemberitahuan kepada customer.

### Penjelasan Langkah demi Langkah

#### 1. Admin membuka halaman utama

- Admin mengakses halaman utama melalui browser.

#### 2. Browser mengirim permintaan halaman utama

- Sistem menerima permintaan untuk menampilkan halaman utama.

#### 3. Sistem memverifikasi sesi admin

- Sistem memastikan bahwa user yang login memiliki hak sebagai admin.

#### 4. Sistem menyiapkan elemen UI

- Sistem memuat elemen tampilan yang hanya bisa diakses oleh admin.

#### 5. Browser menampilkan halaman utama admin

- Halaman admin ditampilkan, termasuk menu manajemen booking.

#### 6. Admin memilih menu “Kelola Booking”

- Admin klik menu tersebut untuk mengakses daftar booking yang perlu dikelola.

#### 7. Browser mengirim permintaan daftar booking

- Sistem menerima permintaan data booking untuk ditampilkan.

#### 8. Sistem mengambil data booking dari database

- Sistem mengakses database untuk mengambil seluruh data booking.

#### 9. Database mengembalikan data booking

- Data booking dikirim kembali ke sistem.

#### 10. Sistem menyiapkan data booking untuk ditampilkan

- Sistem menyusun data agar siap untuk divisualisasikan di UI.

#### **11. Browser menampilkan daftar booking**

- Admin dapat melihat semua booking yang tersedia.

#### **12. Admin klik tombol “Batalkan” pada salah satu booking**

- Admin memilih booking tertentu dan mengirim permintaan pembatalan.

#### **13. Browser mengirim data pembatalan ke sistem**

- Sistem menerima data booking yang akan dibatalkan.

#### **14. Sistem memvalidasi permintaan pembatalan**

- Sistem memastikan booking masih berlaku dan bisa dibatalkan.

#### **15. Sistem memperbarui status booking**

- Sistem mengubah status booking menjadi “Dibatalkan” di database.

#### **16. Database menyimpan pembaruan dan mengonfirmasi**

- Database mengonfirmasi bahwa status berhasil diperbarui.

#### **17. Sistem mengirim notifikasi pembatalan ke customer**

- Notifikasi dikirim ke sistem notifikasi untuk disampaikan ke customer.

#### **18. Sistem notifikasi mengirimkan pesan dan mengonfirmasi**

- Notifikasi diterima oleh sistem dan dikirim ke customer (misalnya via email atau sistem dalam aplikasi).

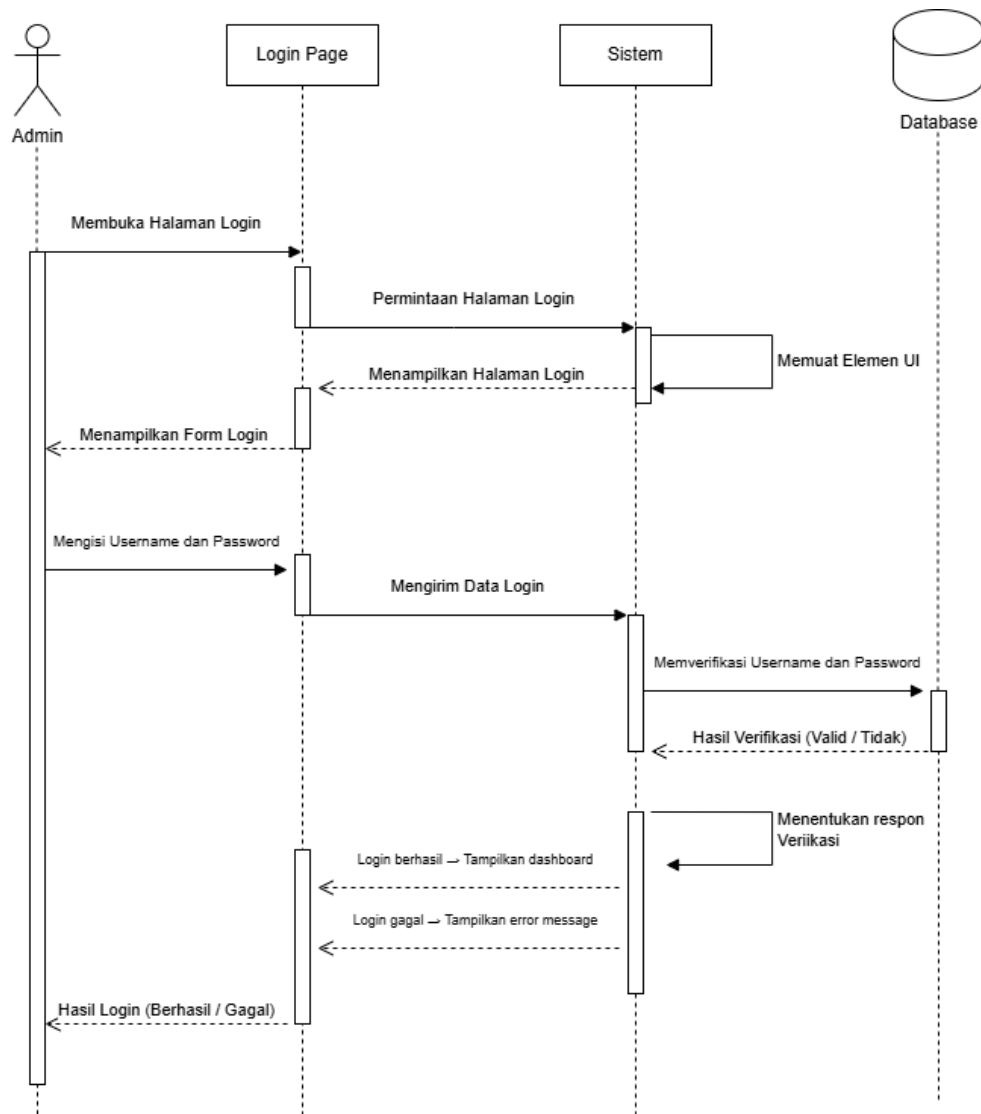
#### **19. Sistem mengirimkan konfirmasi ke browser**

- Setelah notifikasi berhasil dikirim, sistem memberi tahu UI bahwa pembatalan berhasil.

#### **20. Browser menampilkan pesan “Booking dibatalkan”**

- Admin melihat konfirmasi bahwa proses pembatalan telah selesai.

## Sequence Diagram : Login Admin



Gambar 11 Sequence Diagram : Login Admin

## Penjelasan

### Aktor & Komponen Sistem

- **Admin** – Pengguna sistem yang ingin mengakses halaman dashboard admin.
- **Login Page** – Halaman web tempat admin memasukkan username dan password.
- **Sistem** – Backend server yang menangani proses otentikasi.
- **Database** – Menyimpan data username dan password admin.

### Langkah-langkah Detail

### **1. Admin membuka halaman login**

- Admin membuka sistem melalui browser dan diarahkan ke halaman login.

### **2. Login Page mengirim permintaan halaman login ke sistem**

- Permintaan ini diteruskan ke sistem backend untuk memuat tampilan halaman login.

### **3. Sistem memuat elemen UI halaman login**

- Sistem menyusun tampilan form login yang terdiri dari kolom username dan password.

### **4. Login Page menampilkan form login kepada admin**

- Halaman login berhasil ditampilkan, dan admin bisa mulai mengisi data.

### **5. Admin mengisi username dan password**

- Admin mengetik kredensial login di form yang tersedia.

### **6. Login Page mengirim data login ke sistem**

- Data dikirim ke backend untuk diverifikasi.

### **7. Sistem memverifikasi username dan password**

- Sistem mengakses database untuk mencocokkan input dengan data admin yang tersimpan.

### **8. Database mengirim hasil verifikasi (valid / tidak)**

- Jika data cocok, maka hasil verifikasi valid; jika tidak cocok, maka tidak valid.

### **9. Sistem menentukan respons verifikasi**

- Berdasarkan hasil dari database, sistem:
  - Jika login **valid**, akan meneruskan ke dashboard admin.
  - Jika **tidak valid**, sistem akan memberikan pesan error ke browser.

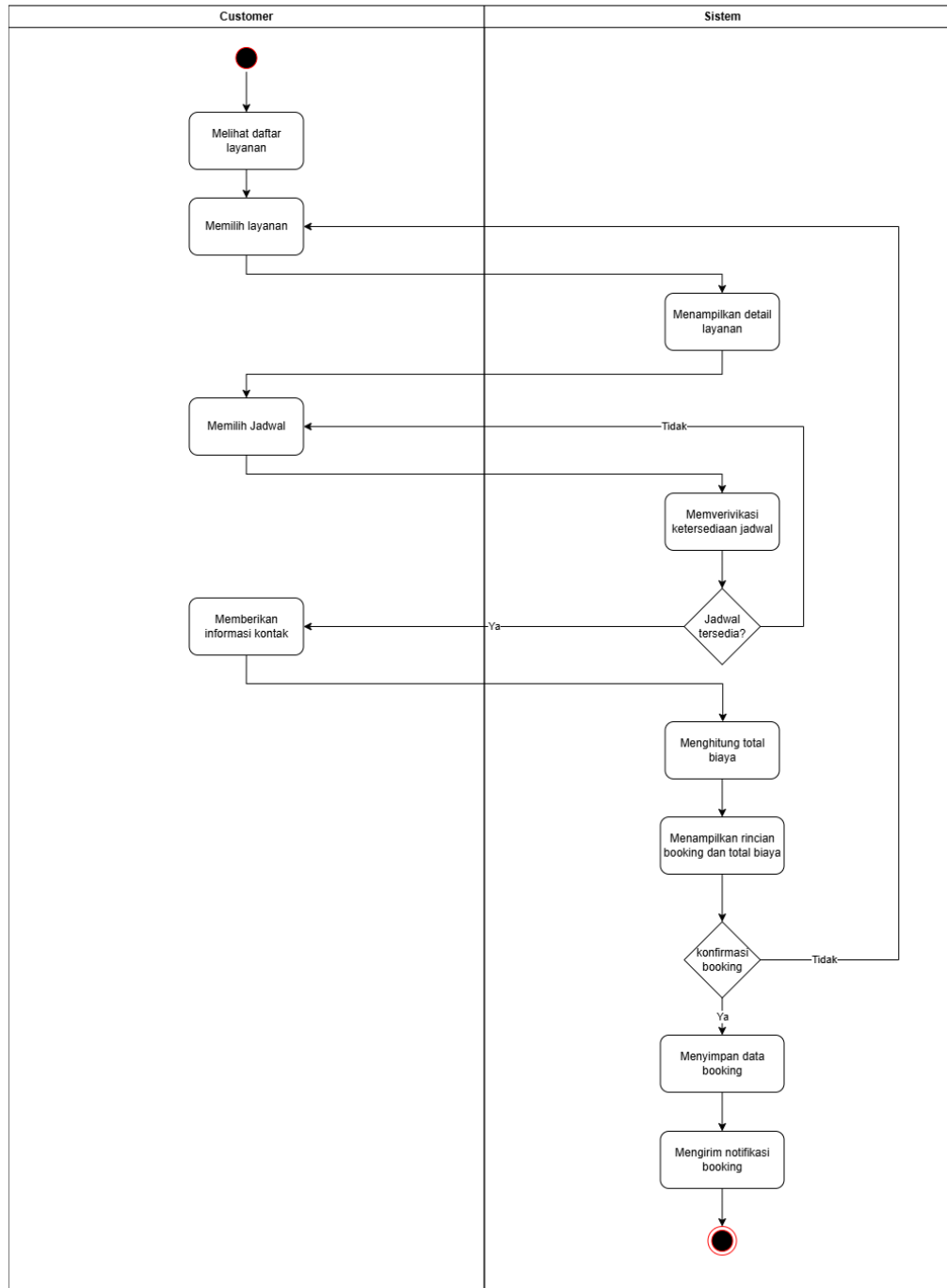
### **10. Login Page menampilkan hasil login**

- **Jika berhasil:** menampilkan halaman dashboard.
- **Jika gagal:** menampilkan pesan error seperti “Username atau Password salah”.

### 3.3 Activity Diagram

#### Activity Diagram : Melakukan Booking Layanan

Booking



Gambar 12 Activity Diagram : Melakukan Booking Layanan

## Penjelasan

Diagram ini alur proses pemesanan layanan yang melibatkan interaksi antara pelanggan dan sistem. Langkah-langkahnya mencakup memilih layanan, menentukan jadwal, konfirmasi pemesanan, hingga sistem mengirimkan notifikasi. Proses ini dibuat untuk memastikan bahwa pelanggan mendapatkan layanan sesuai jadwal yang tersedia dan terkonfirmasi dengan baik.

### 1. Swimlane:

- Diagram ini menggunakan dua swimlane:
  - **Customer:** Merepresentasikan tindakan yang dilakukan oleh pelanggan.
  - **Sistem:** Merepresentasikan proses yang dilakukan oleh sistem.

### 2. Alur Aktivitas:

- **Melihat daftar layanan (Customer):** Pelanggan memulai proses dengan melihat daftar layanan yang tersedia.
- **Menampilkan detail layanan (Sistem):** Sistem menampilkan informasi detail mengenai layanan yang dipilih oleh pelanggan.
- **Memilih layanan (Customer):** Pelanggan memilih salah satu layanan yang tersedia.
- **Memilih jadwal (Customer):** Setelah memilih layanan, pelanggan menentukan jadwal yang diinginkan.
- **Memverifikasi ketersediaan jadwal (Sistem):** Sistem memeriksa apakah jadwal yang dipilih pelanggan tersedia.
  - **Jika tidak tersedia:** Proses kembali ke langkah memilih jadwal.
  - **Jika tersedia:** Proses dilanjutkan.
- **Menghitung total biaya (Sistem):** Sistem menghitung biaya yang perlu dibayar pelanggan berdasarkan layanan yang dipilih.
- **Menampilkan rincian booking dan total biaya (Sistem):** Sistem menampilkan informasi lengkap tentang pemesanan, termasuk biaya total.
- **Konfirmasi booking (Customer):** Pelanggan memutuskan untuk mengonfirmasi pemesanan.
  - **Jika tidak dikonfirmasi:** Proses dihentikan.
  - **Jika dikonfirmasi:** Proses dilanjutkan.
- **Menyimpan data booking (Sistem):** Sistem menyimpan data pemesanan ke database.
- **Mengirim notifikasi booking (Sistem):** Sistem mengirimkan notifikasi kepada pelanggan terkait pemesanan yang telah dilakukan.

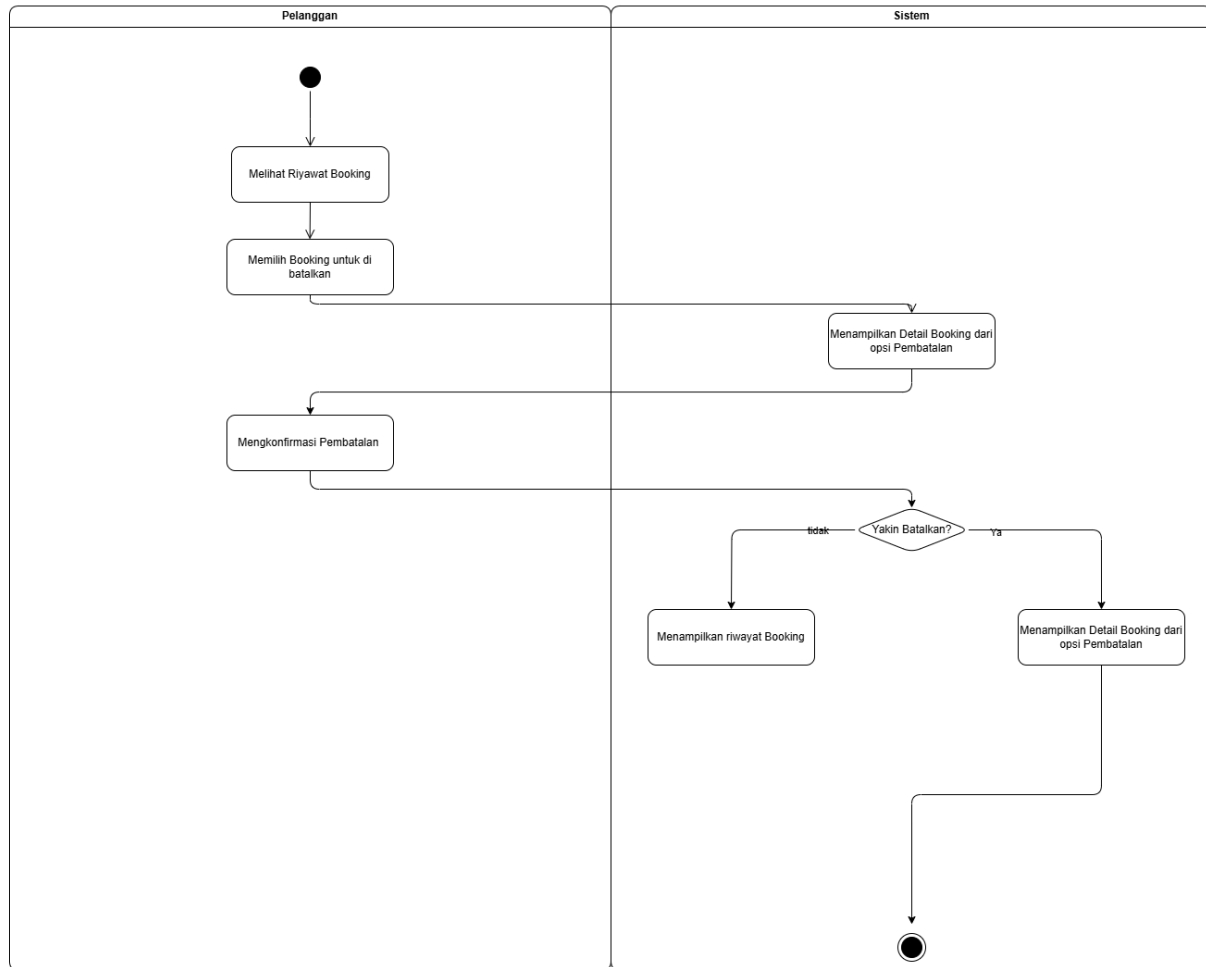
### 3. Awal dan Akhir Aktivitas:



- Diagram dimulai dari lingkaran hitam di swimlane Customer (awal proses).
- Berakhir dengan lingkaran hitam yang dilingkari di bagian bawah (akhir proses).

## Activity Diagram : Membatalkan Booking

Pembatalan Booking



Gambar 13 Activity Diagram : Membatalkan Booking

### Penjelasan

Diagram ini menjelaskan alur logis yang terjadi ketika pelanggan ingin membatalkan booking. Langkah-langkahnya dirancang untuk memastikan bahwa pelanggan membuat keputusan yang tepat sebelum melakukan pembatalan. Sistem memberikan informasi detail sebagai bahan pertimbangan, dan pelanggan diberikan kesempatan untuk membatalkan atau melanjutkan proses. Dengan alur ini, pembatalan booking dapat dilakukan secara terstruktur dan terverifikasi.

1. **Swimlane:** Diagram ini terbagi menjadi dua swimlane:

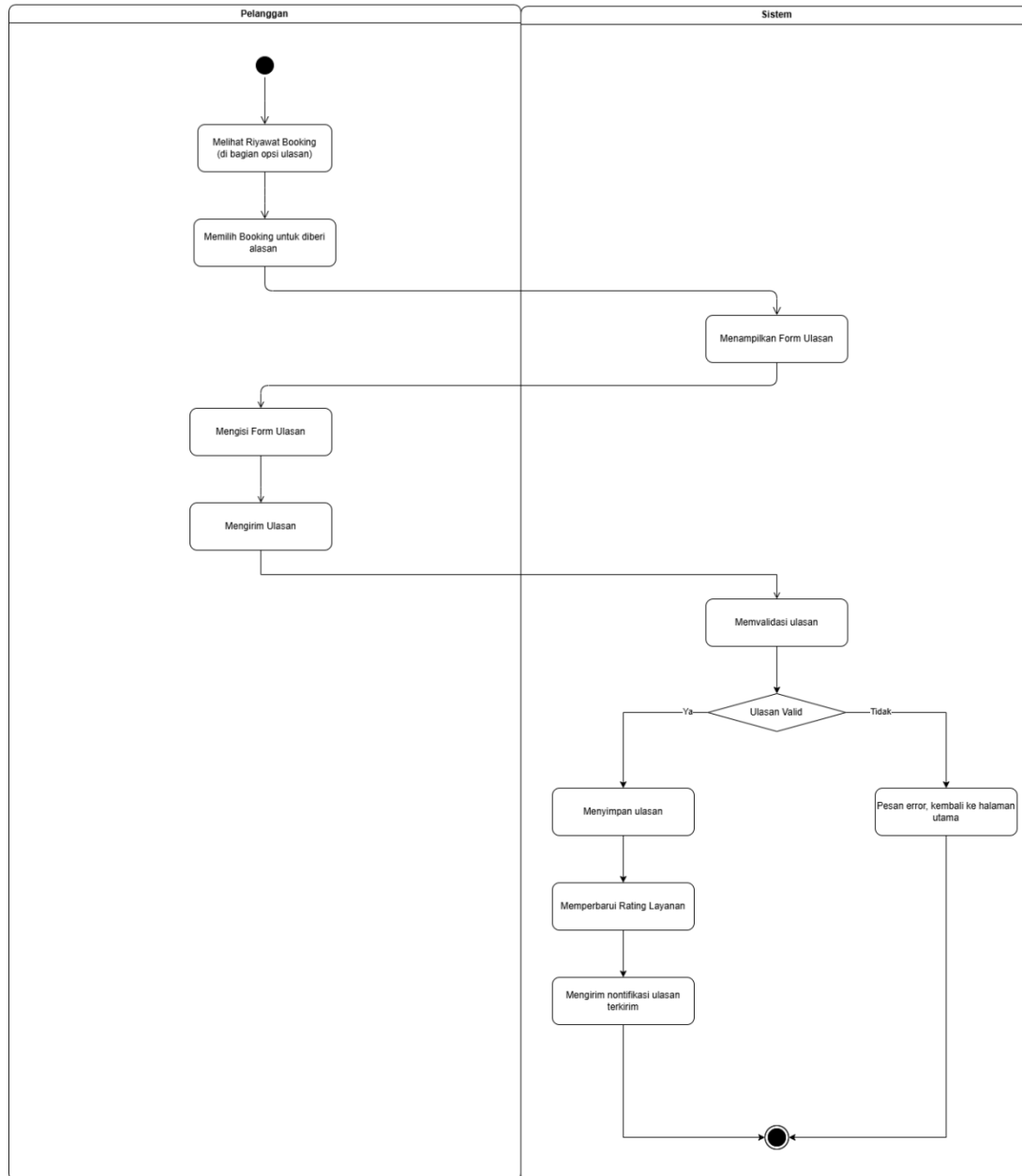
- **Pelanggan:** Berisi aktivitas yang dilakukan oleh pelanggan dalam proses pembatalan booking.
- **Sistem:** Berisi langkah-langkah yang dilakukan oleh sistem untuk mendukung proses tersebut.

2. **Alur Aktivitas:**

- **Melihat Riwayat Booking** (Pelanggan): Pelanggan memulai proses dengan melihat daftar riwayat booking yang dimiliki.
- **Memilih Booking untuk Dibatalkan** (Pelanggan): Pelanggan memilih salah satu booking yang ingin dibatalkan.
- **Menampilkan Detail Booking dari Opsi Pembatalan** (Sistem): Sistem menampilkan informasi detail dari booking yang dipilih, termasuk opsi untuk melanjutkan pembatalan.
- **Mengonfirmasi Pembatalan** (Pelanggan): Pelanggan diminta untuk mengonfirmasi apakah benar-benar ingin membatalkan booking.
  - **Keputusan "Yakin Batalkan?":**
    - Jika pelanggan memilih **tidak**, proses akan kembali ke langkah **Menampilkan Riwayat Booking**.
    - Jika pelanggan memilih **ya**, sistem akan melanjutkan pembatalan.
- **Menampilkan Detail Booking dari Opsi Pembatalan** (Sistem): Sistem memproses pembatalan dan menampilkan informasi terkait status pembatalan kepada pelanggan.
- **Akhir Proses:** Proses selesai setelah pembatalan berhasil diproses.

## Activity Diagram : Memberi Ulasan

Memberikan Ulasan



Gambar 14 Activity Diagram : Memberi Ulasan

### Penjelasan

Diagram ini untuk proses pemberian ulasan. Proses dimulai dari tindakan pelanggan untuk memilih booking yang akan diulas, dilanjutkan dengan sistem menampilkan form dan memvalidasi data ulasan. Jika ulasan valid, sistem menyimpan data, memperbarui rating layanan, dan memberi notifikasi kepada pelanggan. Jika

ulasan tidak valid, sistem memberikan pesan error sebagai umpan balik. Alur ini memastikan ulasan yang diterima sistem memenuhi standar tertentu dan memberikan kontribusi pada evaluasi kualitas layanan.

## 1. Swimlane:

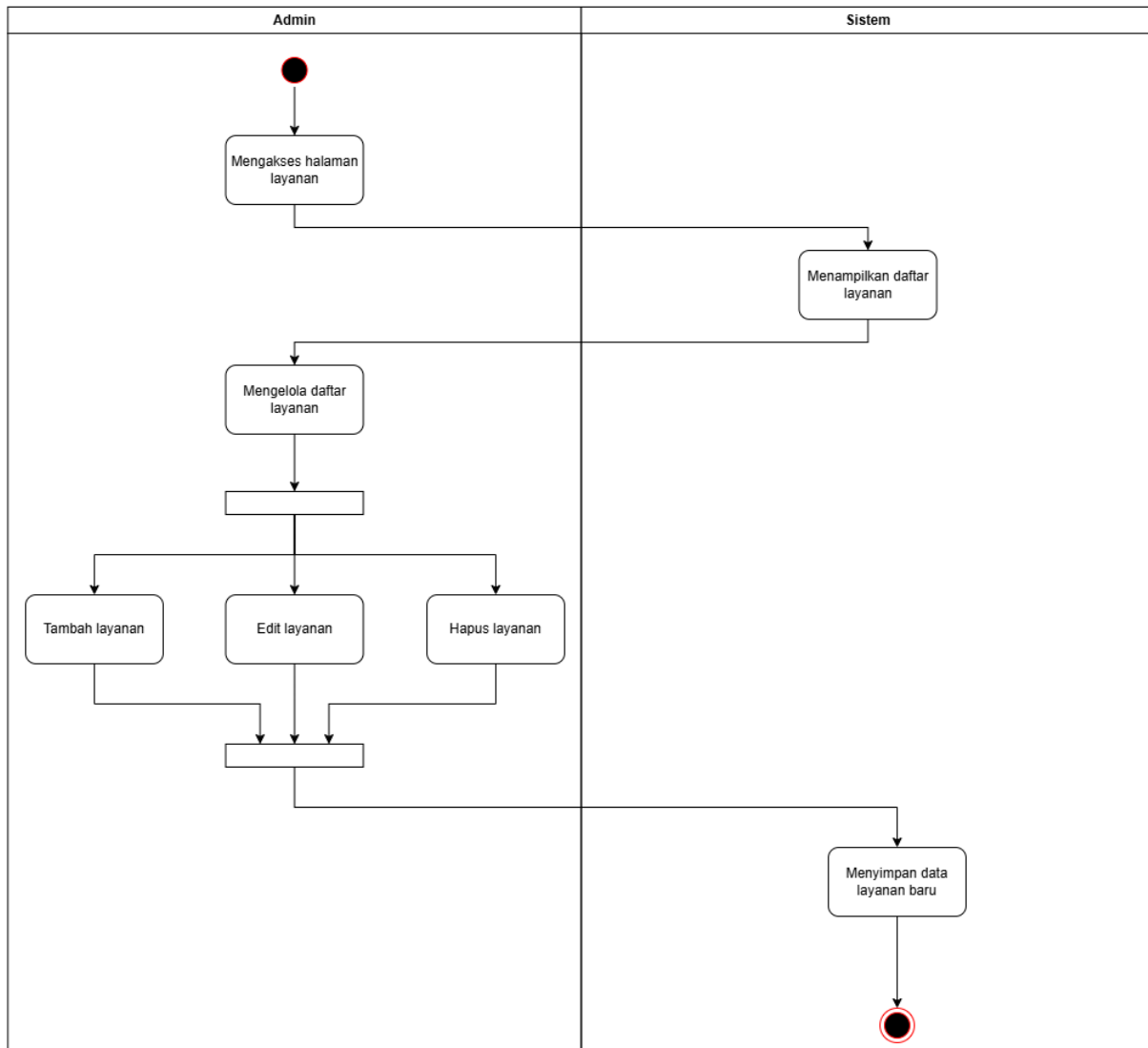
- **Pelanggan:** Berisi aktivitas yang dilakukan oleh pelanggan untuk memberikan ulasan.
- **Sistem:** Berisi langkah-langkah yang dilakukan sistem untuk memproses dan memvalidasi ulasan tersebut.

## 2. Alur Aktivitas:

- **Melihat Riwayat Booking (di bagian opsi ulasan) (Pelanggan):** Pelanggan memulai dengan membuka daftar riwayat booking yang telah dilakukan, khususnya bagian untuk memberikan ulasan.
- **Memilih Booking untuk Diberi Ulasan (Pelanggan):** Pelanggan memilih salah satu booking yang ingin diberikan ulasan.
- **Menampilkan Form Ulasan (Sistem):** Sistem menampilkan formulir untuk mengisi ulasan.
- **Mengisi Form Ulasan (Pelanggan):** Pelanggan mengisi formulir dengan komentar dan rating untuk layanan yang digunakan.
- **Mengirim Ulasan (Pelanggan):** Setelah mengisi, pelanggan mengirimkan ulasan tersebut.
- **Memvalidasi Ulasan (Sistem):** Sistem memeriksa kelengkapan dan keabsahan ulasan yang dikirim.
  - **Keputusan "Ulasan Valid?":**
    - Jika ulasan valid:
      - **Menyimpan Ulasan (Sistem):** Sistem menyimpan ulasan ke database.
      - **Memperbarui Rating Layanan (Sistem):** Sistem menghitung ulang rating layanan berdasarkan ulasan baru.
      - **Mengirim Notifikasi Ulasan Terkirim (Sistem):** Sistem mengirimkan notifikasi kepada pelanggan bahwa ulasan berhasil dikirim.
    - Jika ulasan tidak valid:
      - **Pesan Error:** Sistem memberikan pesan error dan mengembalikan pelanggan ke halaman utama.
- **Akhir Proses:** Proses selesai setelah ulasan berhasil disimpan atau pelanggan kembali ke halaman utama.

## Activity Diagram : Mengelola Layanan

Mengelola Layanan



Gambar 15 Activity Diagram : Mengelola Layanan

### Penjelasan

Activity ini menunjukkan bagaimana admin mengelola layanan melalui sistem, mulai dari menampilkan daftar layanan hingga mengambil tindakan seperti menambah, mengedit, atau menghapus layanan. Sistem bertanggung jawab untuk memproses dan menyimpan perubahan yang dilakukan admin. Alur ini memastikan bahwa layanan yang tersedia di sistem selalu diperbarui sesuai kebutuhan. Proses dirancang untuk memberikan fleksibilitas kepada admin dalam mengelola layanan secara efisien.

### **1. Mengakses Halaman Layanan (Admin):**

Proses dimulai ketika admin mengakses halaman untuk mengelola layanan. Ini merupakan langkah awal sebelum melakukan pengelolaan layanan apa pun.

### **2. Menampilkan Daftar Layanan (Sistem):**

Sistem akan menampilkan daftar layanan yang tersedia. Langkah ini memberikan gambaran kepada admin tentang layanan yang sudah ada.

### **3. Mengelola Daftar Layanan (Admin):**

Admin memiliki opsi untuk mengelola layanan. Pada tahap ini, admin dapat memilih tiga tindakan utama:

- Tambah Layanan: Admin menambahkan layanan baru yang akan disimpan oleh sistem.
- Edit Layanan: Admin melakukan perubahan pada layanan yang sudah ada.
- Hapus Layanan: Admin menghapus layanan tertentu dari daftar.

### **4. Menyimpan Data Layanan Baru (Sistem):**

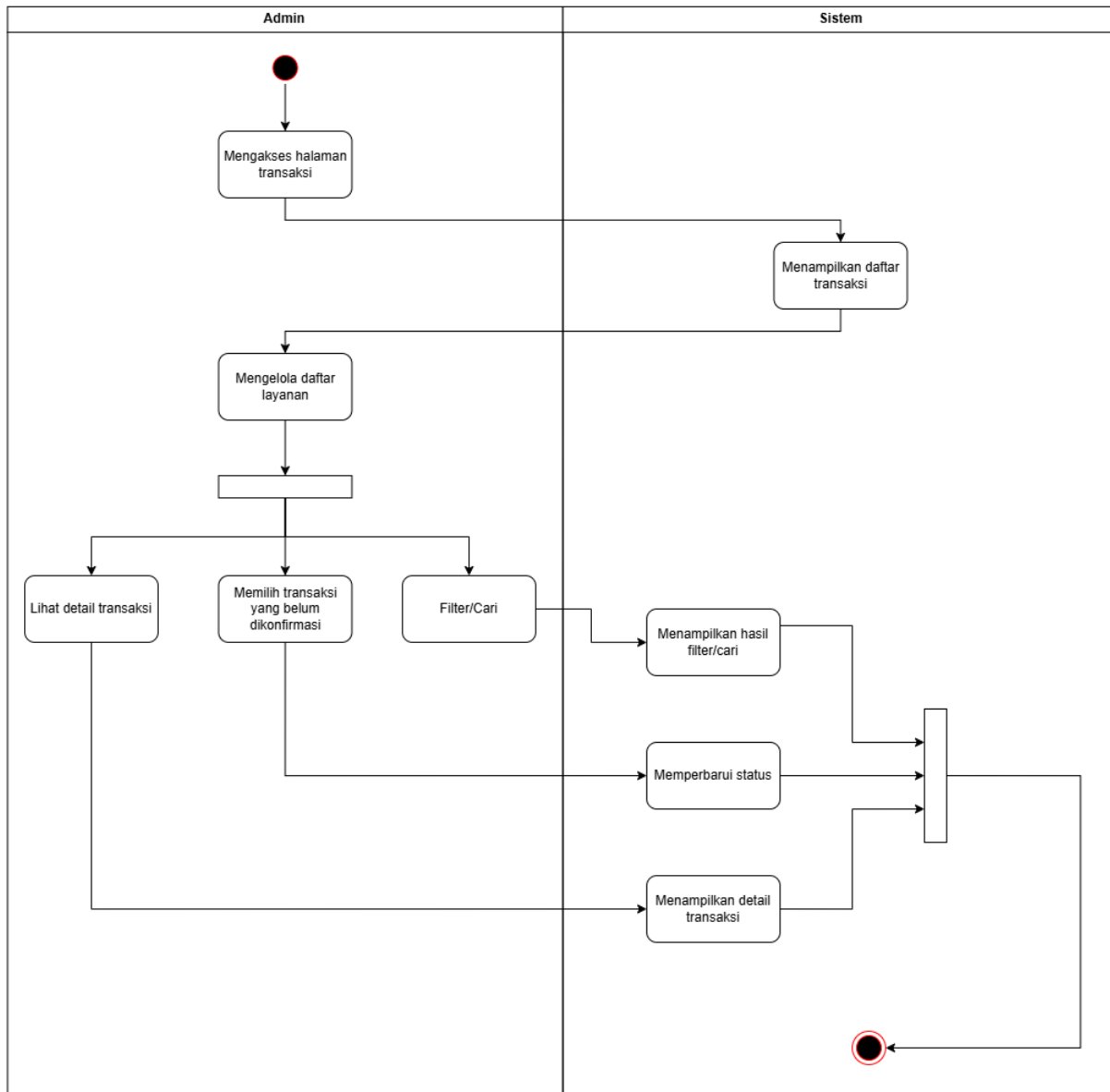
Setelah admin melakukan tindakan seperti menambah, mengedit, atau menghapus layanan, sistem akan menyimpan data perubahan tersebut.

### **5. Akhir Proses:**

Proses selesai setelah data layanan baru disimpan oleh sistem, baik itu layanan baru, layanan yang sudah diperbarui, maupun penghapusan layanan.

## Activity Diagram : Mengelola Transaksi

Mengelola Transaksi



Gambar 16 Activity Diagram : Mengelola Transaksi

### Penjelasan

Diagram ini menunjukkan bagaimana admin dapat mengelola transaksi dengan fleksibilitas, seperti melihat detail transaksi, memperbarui status, atau memfilter transaksi tertentu. Sistem memastikan bahwa semua informasi yang dibutuhkan admin ditampilkan dengan jelas dan memberikan opsi untuk melakukan pembaruan status jika diperlukan. Proses ini admin dapat memantau dan mengelola transaksi dengan lebih efisien, sehingga mendukung pengelolaan operasional secara keseluruhan.

### **1. Mengakses Halaman Transaksi (Admin):**

Proses dimulai ketika admin membuka halaman transaksi di sistem untuk mengelola transaksi yang ada.

### **2. Menampilkan Daftar Transaksi (Sistem):**

Sistem menampilkan semua transaksi yang tersimpan, termasuk informasi detail seperti status pembayaran, layanan terkait, dan pelanggan.

### **3. Mengelola Daftar Transaksi (Admin):**

Pada tahap ini, admin memiliki beberapa opsi untuk mengelola transaksi:

- **Lihat Detail Transaksi:** Admin memilih transaksi tertentu untuk melihat informasi detailnya, seperti data pelanggan, jumlah pembayaran, atau status transaksi.
- **Memilih Transaksi yang Belum Dikonfirmasi:** Admin dapat memilih transaksi dengan status "belum dikonfirmasi" untuk memperbarui statusnya.
- **Filter/Cari Transaksi:** Admin menggunakan fitur pencarian atau filter untuk menemukan transaksi tertentu berdasarkan kriteria seperti tanggal, status, atau nama pelanggan.

### **4. Menampilkan Hasil Filter/Cari (Sistem):**

Jika admin menggunakan fitur filter atau pencarian, sistem akan menampilkan hasil yang sesuai dengan kriteria yang dimasukkan.

### **5. Memperbarui Status Transaksi (Sistem):**

Jika admin memilih transaksi yang belum dikonfirmasi, sistem memungkinkan admin untuk mengubah status transaksi, misalnya dari "pending" menjadi "sukses".

### **6. Menampilkan Detail Transaksi (Sistem):**

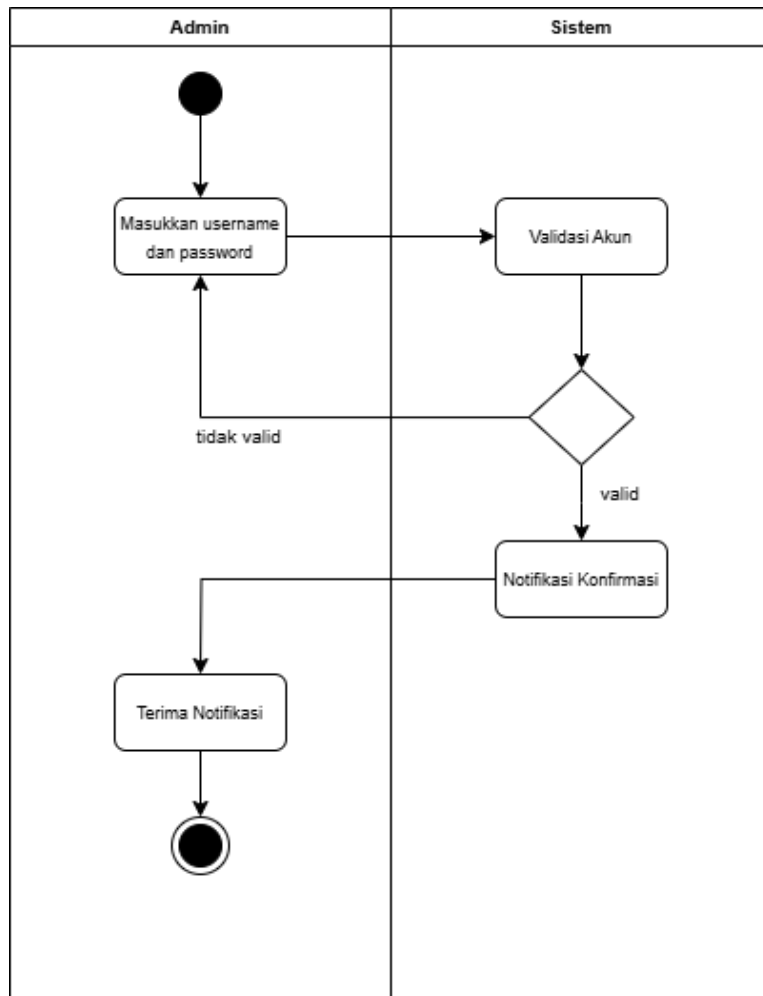
Sistem menampilkan informasi detail dari transaksi yang dipilih oleh admin, memberikan transparansi dan kemudahan pengelolaan.

### **7. Akhir Proses:**

Proses selesai setelah admin menyelesaikan pengelolaan transaksi, baik itu melihat detail transaksi, memperbarui status, atau menggunakan fitur filter/cari.



## Activity Diagram : Login Admin



Gambar 17 Activity Diagram : Login Admin

### Penjelasan

Activity diagram ini menggambarkan alur proses autentikasi admin saat masuk ke dalam sistem.

#### 1. Masukkan Username dan Password (Admin):

Proses dimulai ketika admin memasukkan kredensial (username dan password) untuk mengakses sistem.

#### 2. Validasi Akun (Sistem):

Sistem menerima input dari admin dan melakukan validasi terhadap username dan password yang dimasukkan. Validasi ini mencakup pemeriksaan kecocokan data dengan informasi yang tersimpan di database.

### **3. Keputusan Validasi:**

Setelah proses validasi, ada dua kemungkinan:

- Tidak Valid: Jika kredensial tidak sesuai, admin diarahkan kembali untuk mengulangi proses memasukkan username dan password.
- Valid: Jika kredensial sesuai, proses dilanjutkan.

### **4. Notifikasi Konfirmasi (Sistem):**

Jika kredensial valid, sistem mengirimkan notifikasi kepada admin bahwa proses autentikasi berhasil.

### **5. Terima Notifikasi (Admin):**

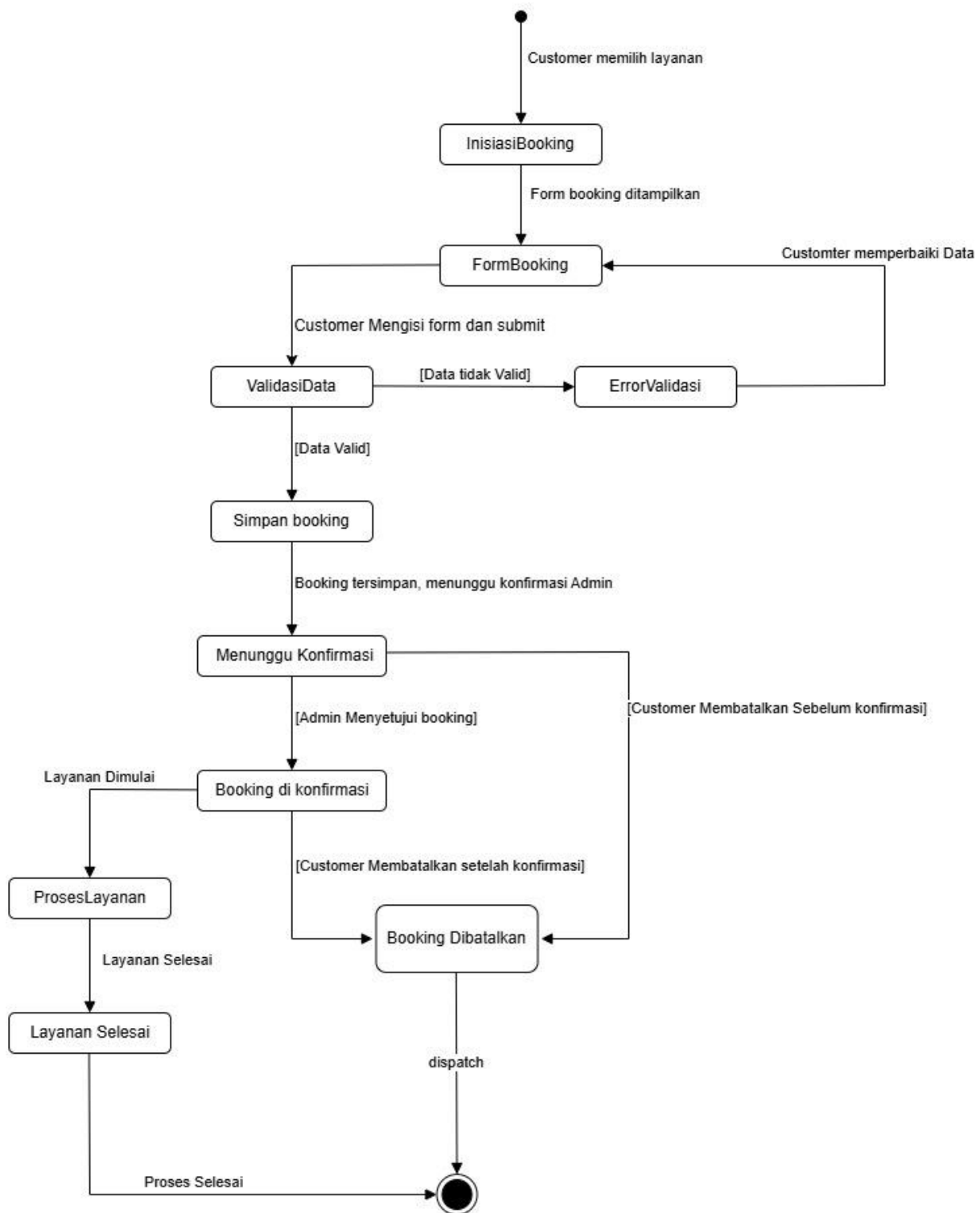
Admin menerima notifikasi bahwa ia berhasil masuk ke sistem.

### **6. Akhir Proses:**

Proses autentikasi selesai setelah admin menerima notifikasi keberhasilan.

### 3.3 State Diagram

#### I.1.A. State Diagram : Booking



Gambar 18 State Diagram : Booking

## Penjelasan

State diagram ini menggambarkan alur proses pemesanan layanan (booking) yang melibatkan beberapa tahapan mulai dari inisiasi hingga layanan selesai atau pembatalan.

### 1. Inisiasi Booking:

Proses dimulai saat customer memilih layanan yang diinginkan. Setelah itu, sistem akan menampilkan formulir booking.

### 2. Form Booking:

Customer mengisi formulir booking dan mengirimkannya (submit).

- Jika data tidak valid, sistem akan menampilkan Error Validasi dan meminta customer untuk memperbaiki data. Proses kembali ke tahap Form Booking.
- Jika data valid, proses dilanjutkan ke tahap berikutnya.

### 3. Simpan Booking:

Data booking yang valid akan disimpan oleh sistem, kemudian statusnya menunggu konfirmasi dari admin.

### 4. Menunggu Konfirmasi:

Pada tahap ini, sistem menunggu tindakan dari admin:

- Jika admin menyetujui booking, statusnya berubah menjadi Booking Dikonfirmasi.
- Jika customer membatalkan sebelum admin memberikan konfirmasi, status berubah menjadi Booking Dibatalkan.

### 5. Booking Dikonfirmasi:

Setelah admin mengonfirmasi, layanan akan dimulai. Namun, customer masih memiliki opsi untuk membatalkan meskipun sudah dikonfirmasi. Jika pembatalan dilakukan, status berubah menjadi Booking Dibatalkan.

### 6. Proses Layanan:

Ketika layanan dimulai, status booking masuk ke tahap Proses Layanan.

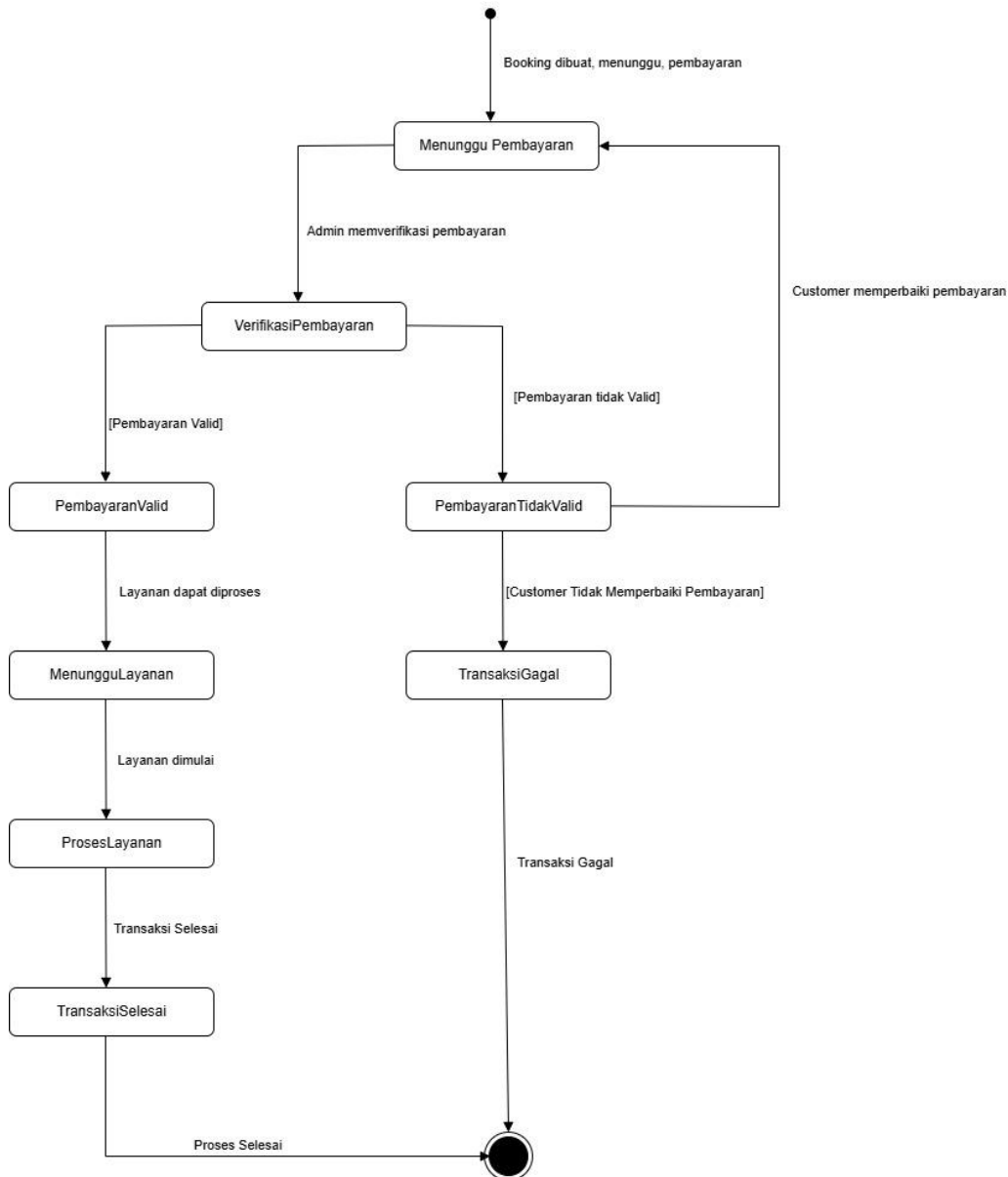
### 7. Layanan Selesai:

Setelah layanan selesai diberikan, status booking berubah menjadi Layanan Selesai, dan proses berakhir.

### 8. Booking Dibatalkan:

Jika customer membatalkan booking pada tahap mana pun (sebelum atau sesudah konfirmasi), status berubah menjadi Booking Dibatalkan, dan proses berakhir.

### I.1.B. State Diagram : Transaksi



Gambar 19 State Diagram : Transaksi

## Penjelasan

State diagram ini menunjukkan bagaimana sistem menangani alur proses dari pembayaran hingga penyelesaian layanan. Proses mencakup verifikasi pembayaran oleh admin, penanganan kesalahan pembayaran, dan penjadwalan layanan. Sistem memberikan fleksibilitas kepada customer untuk memperbaiki kesalahan pembayaran, tetapi jika tidak dilakukan, status berakhir dengan kegagalan transaksi. Diagram ini memastikan setiap tahap transaksi tercatat dan memiliki alur penyelesaian yang jelas.

### 1. Menunggu Pembayaran:

Proses dimulai setelah booking dibuat. Sistem menunggu customer untuk melakukan pembayaran.

- Jika pembayaran dilakukan tetapi terdapat kesalahan, customer diminta untuk memperbaiki pembayaran.

### 2. Verifikasi Pembayaran:

Admin memverifikasi status pembayaran yang masuk:

- Pembayaran Valid: Jika pembayaran sesuai, status pembayaran dianggap valid, dan proses dilanjutkan.
- Pembayaran Tidak Valid: Jika pembayaran tidak sesuai, customer diberi kesempatan untuk memperbaiki. Jika tidak diperbaiki dalam waktu tertentu, status berubah menjadi Transaksi Gagal.

### 3. Pembayaran Valid:

Ketika pembayaran valid, layanan dapat diproses. Status ini menandai bahwa pembayaran berhasil dan booking akan dilanjutkan.

### 4. Menunggu Layanan:

Setelah pembayaran valid, sistem menunggu waktu yang telah ditentukan untuk memulai layanan.

### 5. Proses Layanan:

Pada tahap ini, layanan dimulai sesuai dengan jadwal yang telah disepakati

### 6. Transaksi Selesai:

Setelah layanan selesai dilakukan, status transaksi berubah menjadi selesai, dan proses berakhir.

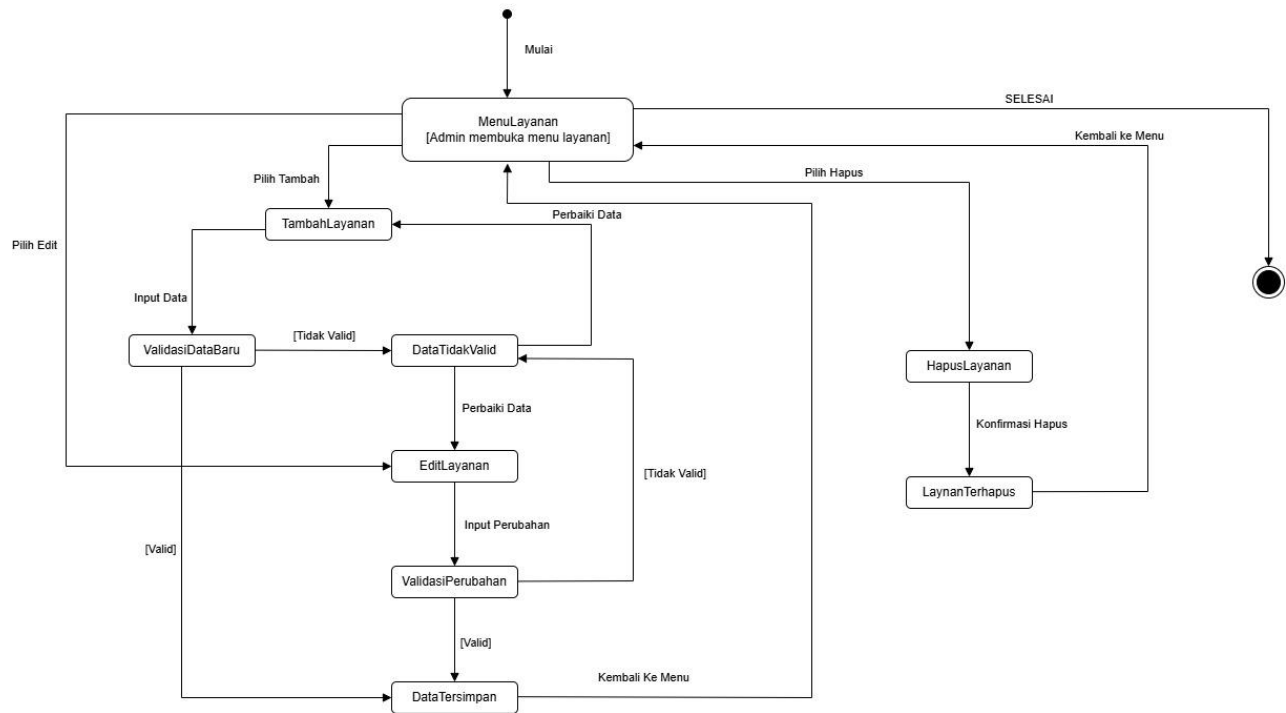
### 7. Pembayaran Tidak Valid / Transaksi Gagal:

- Jika pembayaran tidak valid dan customer tidak memperbaiki pembayaran dalam waktu tertentu, status berubah menjadi Transaksi Gagal.
- Jika proses layanan gagal karena alasan tertentu, status juga dapat menjadi Transaksi Gagal.

## 8. Proses Selesai:

Diagram berakhir setelah transaksi selesai (baik berhasil atau gagal).

### I.1.C. State Diagram : Layanan



Gambar 20 State Diagram : Layanan

## Penjelasan

Diagram ini menunjukkan bagaimana sistem memastikan integritas data melalui validasi pada setiap tahap pengelolaan layanan. Admin memiliki fleksibilitas untuk menambah, mengedit, atau menghapus layanan sesuai kebutuhan. Proses ini juga mencakup mekanisme pengulangan untuk memastikan bahwa data yang dimasukkan benar dan sesuai dengan standar sistem.

### 1. Mulai:

Proses dimulai ketika admin membuka menu layanan melalui antarmuka sistem.

### 2. Menu Layanan:

Pada tahap ini, admin memiliki tiga opsi:

- Pilih Tambah: Menambahkan layanan baru.
- Pilih Edit: Mengedit data layanan yang sudah ada.

- Pilih Hapus: Menghapus layanan yang ada.

### **3. Tambah Layanan:**

- Admin memasukkan data untuk layanan baru.
- Validasi Data Baru: Sistem memverifikasi data yang dimasukkan.
  - Data Tidak Valid: Jika data tidak valid, admin diminta untuk memperbaiki data, dan proses kembali ke tahap pengisian data.
  - Valid: Jika data valid, layanan baru akan tersimpan dalam sistem.

### **4. Edit Layanan:**

- Admin memilih layanan yang ingin diperbarui, lalu memasukkan perubahan data.
- Validasi Perubahan: Sistem memverifikasi perubahan data.
  - Data Tidak Valid: Jika perubahan tidak valid, admin diminta memperbaiki data, dan proses kembali ke tahap pengisian perubahan.
  - Valid: Jika perubahan valid, data layanan akan diperbarui dan tersimpan dalam sistem.

### **5. Hapus Layanan:**

- Admin memilih layanan yang ingin dihapus.
- Konfirmasi Hapus: Sistem meminta konfirmasi dari admin sebelum penghapusan dilakukan.
- Layanan Terhapus: Jika admin mengonfirmasi, layanan dihapus dari sistem.

### **6. Kembali ke Menu:**

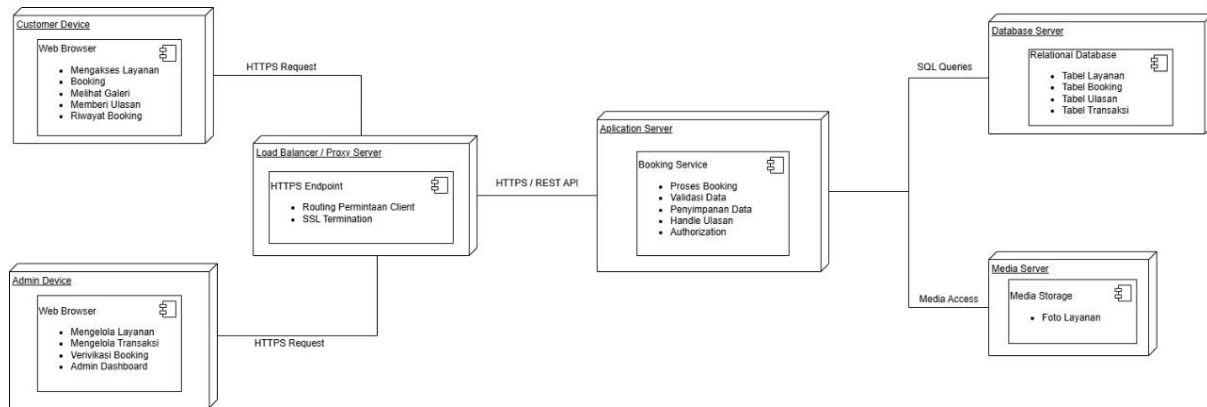
Setelah menyelesaikan salah satu tindakan (tambah, edit, atau hapus), admin diarahkan kembali ke menu layanan untuk melakukan tindakan lain atau menyelesaikan proses.

### **7. Selesai:**

Proses berakhir setelah admin menyelesaikan pengelolaan layanan.



### 3.3 Deployment Diagram



Gambar 21 Deployment Diagram

### Penjelasan

#### 1. Customer Device

Perangkat pengguna (misalnya laptop atau HP) yang digunakan oleh customer. Akses dilakukan melalui web browser. Fitur yang bisa digunakan oleh customer antara lain:

- Mengakses daftar layanan
- Melakukan booking
- Melihat galeri layanan
- Memberikan ulasan
- Melihat riwayat booking

Semua aktivitas ini dikirim ke server lewat **HTTPS Request**.

#### 2. Admin Device

Perangkat yang digunakan oleh admin, juga lewat browser. Admin memiliki akses untuk:

- Mengelola data layanan
- Mengelola transaksi
- Melihat dan memverifikasi booking
- Mengakses dashboard admin

Sama seperti customer, request dari admin dikirim lewat protokol **HTTPS**.

#### 3. Load Balancer / Proxy Server

Komponen ini bertugas mengatur lalu lintas request dari client (customer dan admin). Fungsi utamanya:

- Menyediakan HTTPS Endpoint (akses aman)
- Routing permintaan ke Application Server
- SSL Termination (mengamankan koneksi)

Load balancer memastikan request dibagi rata ke server aplikasi dan bisa diakses secara aman.

#### 4. Application Server

Ini adalah jantung dari sistem, tempat seluruh logika aplikasi berjalan. Di sini terdapat **Booking Service**, yang memiliki tanggung jawab untuk:

- Memproses booking dari user
- Melakukan validasi data
- Menyimpan data ke database
- Menangani ulasan dari pelanggan
- Autentikasi dan otorisasi pengguna

Komunikasi antara Load Balancer dan Application Server menggunakan **HTTPS / REST API**.

#### 5. Database Server

Tempat penyimpanan utama untuk semua data sistem. Di dalamnya ada **Relational Database** yang mencakup beberapa tabel penting:

- Tabel Layanan
- Tabel Booking
- Tabel Ulasan
- Tabel Transaksi

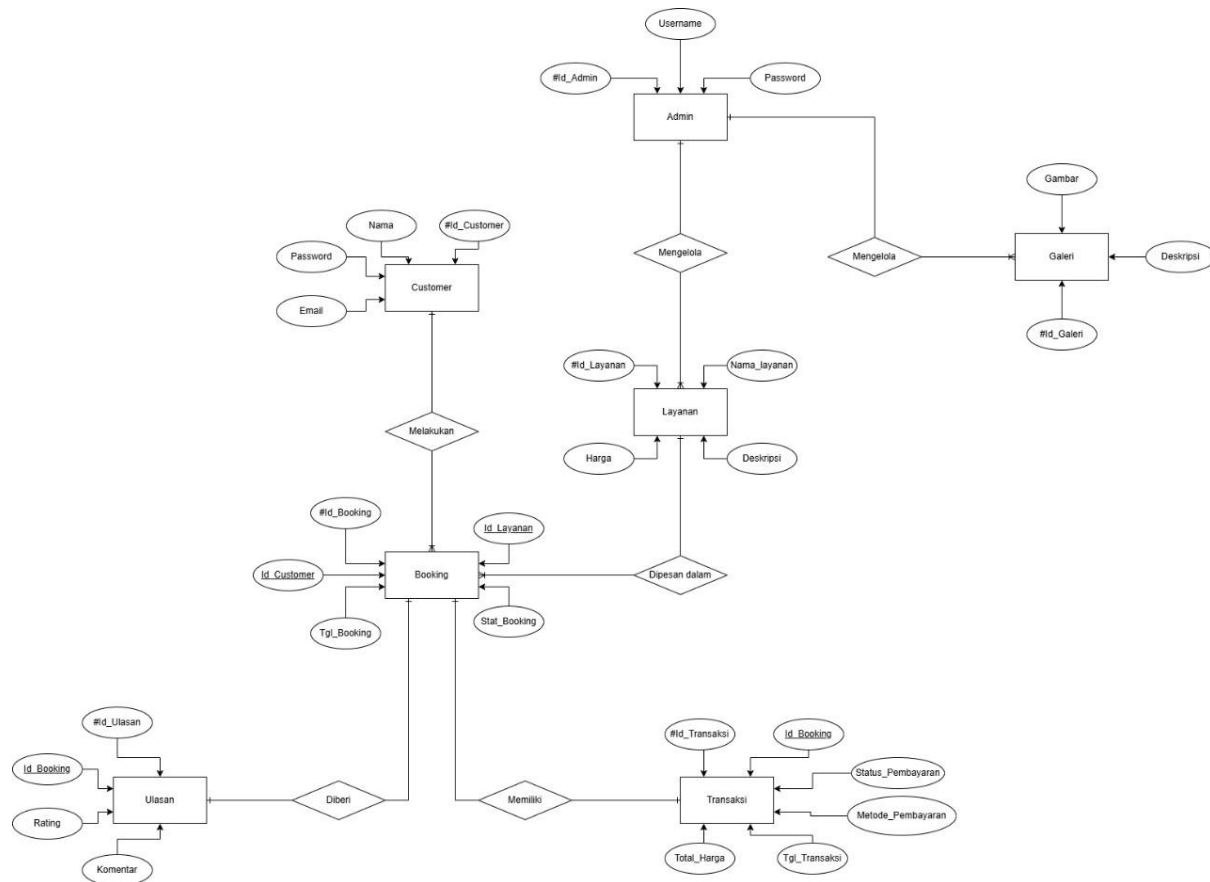
Application Server akan mengakses database ini menggunakan **SQL Queries** untuk membaca/menyimpan data.

#### 6. Media Server

Digunakan untuk menyimpan file media seperti foto-foto layanan. Saat user membuka galeri, data akan diambil dari Media Server melalui koneksi **Media Access** dari Application Server.

## Bab IV Data Design

### 4.1 Logical Design



Gambar 22 Entity Relation Diagram (ERD)

### 4.2 Physical Design

Skema relasi dan struktur tabel yang didapat dari ERD :

#### Penjelasan

Diagram ini menggambarkan struktur basis data dari sistem booking layanan, yang digunakan oleh sebuah usaha treatment kecantikan seperti eyelash atau nail art. Dalam ERD ini, ada beberapa entitas utama yang saling terhubung melalui relasi. Berikut adalah penjelasan dari masing-masing entitas dan relasinya:

#### 1. Customer

Entitas ini menyimpan data pelanggan yang melakukan booking layanan. Atributnya meliputi:

- Id\_Customer (primary key)
- Nama
- Email
- Password

Relasi:

- Satu customer bisa melakukan lebih dari satu **Booking**.

## 2. Booking

Entitas ini mencatat proses pemesanan layanan oleh customer. Atribut yang dimiliki:

- Id\_Booking (primary key)
- Id\_Customer (foreign key)
- Id\_Layanan (foreign key)
- Tgl\_Booking (tanggal booking dilakukan)
- Stat\_Booking (status booking)

Relasi:

- Booking dilakukan oleh satu customer dan memesan satu layanan.
- Booking dapat memiliki satu **Ulasan**.
- Booking juga akan memunculkan satu **Transaksi**.

## 3. Layanan

Mewakili jenis layanan yang tersedia, seperti eyelash extension atau nail art. Atributnya meliputi:

- Id\_Layanan (primary key)
- Nama\_layanan
- Deskripsi
- Harga

Relasi:

- Layanan dikelola oleh **Admin**.
- Layanan dapat dipesan dalam banyak **Booking**.

## 4. Admin

Entitas ini menyimpan data admin yang bertugas mengelola sistem. Atribut:

- Id\_Admin (primary key)
- Username
- Password

Relasi:

- Admin mengelola data **Layanan** dan **Galeri**.

## 5. Galeri

Berisi dokumentasi visual dari layanan yang sudah pernah dilakukan. Atribut:

- Id\_Galeri (primary key)
- Gambar
- Deskripsi

Relasi:

- Galeri dikelola oleh **Admin**.

## 6. Ulasan

Berisi feedback dari customer setelah melakukan booking. Atribut:

- Id\_Ulasan (primary key)
- Id\_Booking (foreign key)
- Rating
- Komentar

Relasi:

- Satu ulasan diberikan untuk satu **Booking**.

## 7. Transaksi

Mencatat pembayaran dari booking yang telah dilakukan. Atribut:

- Id\_Transaksi (primary key)
- Id\_Booking (foreign key)
- Total\_Harga
- Tgl\_Transaksi
- Metode\_Pembayaran
- Status\_Pembayaran

Relasi:

- Satu transaksi hanya terkait dengan satu **Booking**.