

# SOFTWARE DESIGN DOCUMENT (SDD)

FOR APLIKASI BOOKING  
LAPANGAN OLAHRAGA

KELOMPOK

ARYA WIJAYA  
2350081010

AJI KARTIKO H  
23500810

AMMAR BAGAS F  
2350081008

RICKY GUNALDO  
2350081028

# Table Of Contents

Table Of Contents .....	1
Bab I Introduction .....	2
1.1 Purpose .....	2
1.2 Scope .....	2
1.3 Overview .....	2
1.4 Reference .....	2
1.5 Definitions and Acronyms .....	3
Bab II System Overview .....	4
Bab III Application Design .....	5
3.2 Use Case Diagram .....	5
3.3 Use Case Scenario .....	5
Booking Lapangan .....	5
Kelola Lapangan .....	6
3.3 Class Diagram .....	8
3.3 Sequence Diagram .....	9
3.3 Activity Diagram .....	22
3.3 State Diagram .....	28
3.3 Deployment Diagram .....	33
Bab IV Data Design .....	34
4.1 Logical Design .....	34
4.2 Physical Design .....	35
Bab V User Interface Design .....	37
Bab VI Interface Requirements .....	43
6.1 User Interface .....	43
6.2 Hardware Interface .....	43
6.3 Software Interface .....	43
6.4 Communication Interface .....	45

# Bab I Introduction

## 1.1 Purpose

Dokumen ini menyajikan rancangan arsitektural dan teknis bagi pengembangan Aplikasi Booking Lapangan Olahraga. Penyusunan SDD ini bertujuan untuk merepresentasikan secara komprehensif hasil proses desain sistem, yang disusun berdasarkan analisis kebutuhan fungsional dan non-fungsional. Dokumen ini berfungsi sebagai referensi otoritatif bagi tim Software Engineering dalam mengimplementasikan solusi teknis yang telah ditetapkan.

## 1.2 Scope

Dokumen Perancangan Perangkat Lunak (SDD) ini menyajikan spesifikasi teknis komprehensif untuk sistem aplikasi Booking Lapangan Olahraga. Aplikasi ini dirancang untuk memudahkan pengguna mencari dan memesan lapangan olahraga secara online.

Cakupan rancangan meliputi

1. Fungsionalitas Inti: Pencarian, pemesanan, pembayaran online, notifikasi jadwal, dan fitur review serta rating lapangan.
2. Pemodelan Sistem: Mendefinisikan struktur data melalui Entity-Relationship Diagram (E-R Diagram) dan mendokumentasikan alur sistem dengan diagram UML (Use Case, Activity, Class, Sequence, State, Deployment)
3. Komponen Persyaratan Non-Fungsional: Memastikan keamanan transaksi dan data pengguna, serta performa tinggi untuk menangani banyak pemesanan tanpa kendala.

## 1.3 Overview

Dokumen ini menguraikan arsitektur, fungsionalitas, dan spesifikasi teknis sistem. Ruang lingkup pengguna yang diakomodasi mencakup:

Struktur Dokumen

1. Pengguna (Member/Guest): Melakukan pencarian, booking, pembayaran, dan memberikan review/rating.
2. Karyawan: Mengelola data lapangan, jadwal, pembayaran, dan data pengguna.
  - Admin/Owner: Mengelola pengguna, booking, dan membuat laporan.

## 1.4 Reference

Dokumen SRS APLIKASI BOOKING LAPANGAN OLAHRAGA

## 1.5 Definitions and Acronyms

1. SDD (Software Design Document): Dokumen perancangan perangkat lunak.
2. UML (Unified Modeling Language): Standar Diagram UML yang memetakan interaksi antara aktor dan fungsionalitas inti aplikasi.
3. Use Case Diagram: Diagram UML yang memetakan interaksi antara aktor eksternal (pengguna/sistem) dan fungsionalitas inti aplikasi.
4. Activity Diagram: Representasi grafis alur kerja sistem, mencakup aktivitas, keputusan bersyarat (*branching*), iterasi, dan proses konkuren.
5. Class Diagram: Diagram struktural UML yang mendefinisikan kelas objek, atribut, metode, serta relasi antar-entitas dalam sistem.
6. Sequence Diagram: Diagram perilaku UML yang memvisualisasikan pertukaran pesan antar-objek dalam kronologi temporal.
7. State Diagram: Diagram transisi keadaan (*finite-state machine*) untuk mendeskripsikan respons sistem terhadap *event* berdasarkan kondisi internal.
8. Deployment Diagram: Diagram infrastruktur UML yang memodelkan distribusi artefak perangkat lunak (*executable, library*) ke node fisik/virtual.
9. E-R Diagram (Entity-Relationship Diagram): Model konseptual untuk mendesain skema basis data melalui entitas, atribut, dan relasi antar-entitas.

## Bab II System Overview

Aplikasi Pemesanan Lapangan Olahraga merupakan sistem terpadu yang dikembangkan untuk kebutuhan pengelolaan fasilitas olahraga, dengan tujuan merapikan seluruh proses mulai dari pemesanan, penjadwalan, hingga konfirmasi penggunaan lapangan.

### 2.1. Fungsi Utama

Aplikasi ini menyediakan fungsi inti untuk mengotomatisasi proses booking lapangan olahraga:

1. Pencarian Lapangan: Memungkinkan pengguna mencari lapangan berdasarkan kriteria seperti lokasi dan harga.
2. Pemesanan dan Pembayaran: Menyediakan sistem pemesanan dan pembayaran secara online.
3. Notifikasi Jadwal: Memberikan notifikasi untuk mengingatkan jadwal bermain yang telah dipesan.
4. Review dan Rating: Menyediakan fitur bagi pengguna untuk memberikan review dan rating terhadap lapangan.

### 2.2. Fitur Pendukung

Untuk mengoptimalkan pengalaman pengguna, aplikasi dilengkapi fitur-fitur berikut:

1. Keamanan: Keamanan transaksi dan data pengguna dijamin, dengan data pengguna dilindungi menggunakan enkripsi selama proses transaksi.
2. Performa: Sistem dirancang untuk memiliki performa tinggi dan mampu menangani banyak pengguna secara bersamaan tanpa kendala.
3. Platform: Aplikasi ditujukan untuk platform mobile, dengan persyaratan minimal Smartphone Android versi 4.0 atau lebih tinggi.
4. Pembayaran: Mendukung sistem pembayaran online yang aman dan mudah digunakan.

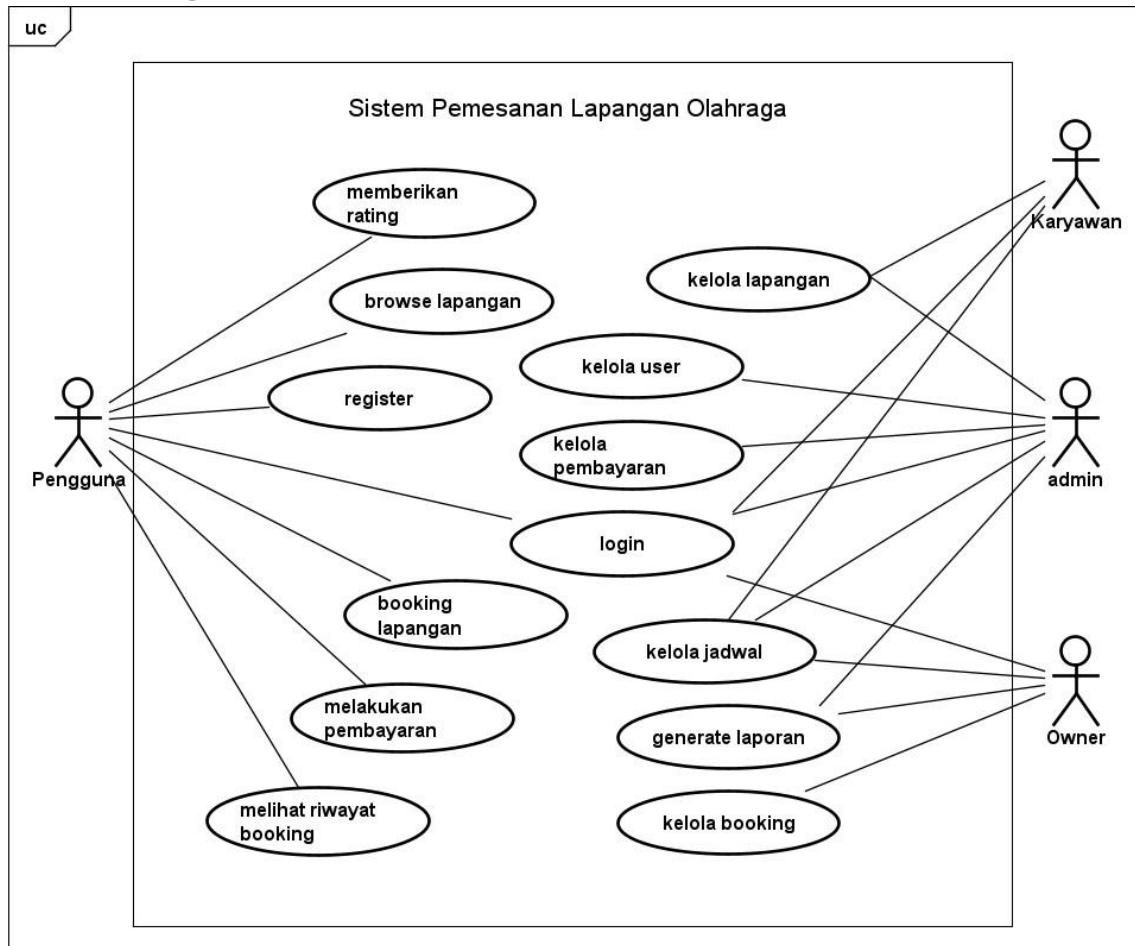
### 2.3. Proses Bisnis

Alur Pengguna:

1. Membuka dan Mencari: Pengguna membuka aplikasi dan dapat mengatur filter untuk mencari lapangan.
2. Melihat Detail: Sistem menampilkan daftar lapangan, dan pengguna memilih untuk melihat detail, rating, dan review.
3. Booking dan Pembayaran: Jika pengguna memutuskan booking, mereka akan melanjutkan ke proses pemilihan jadwal dan melakukan pembayaran.
4. Konfirmasi: Setelah pembayaran berhasil, sistem membuat konfirmasi booking dan mengirim notifikas.

## Bab III Application Design

### 3.2 Use Case Diagram



### 3.3 Use Case Scenario

#### Booking Lapangan

Aktor Utama	Pengguna/Member
Tujuan	Memesan lapangan pada tanggal dan waktu tertentu.

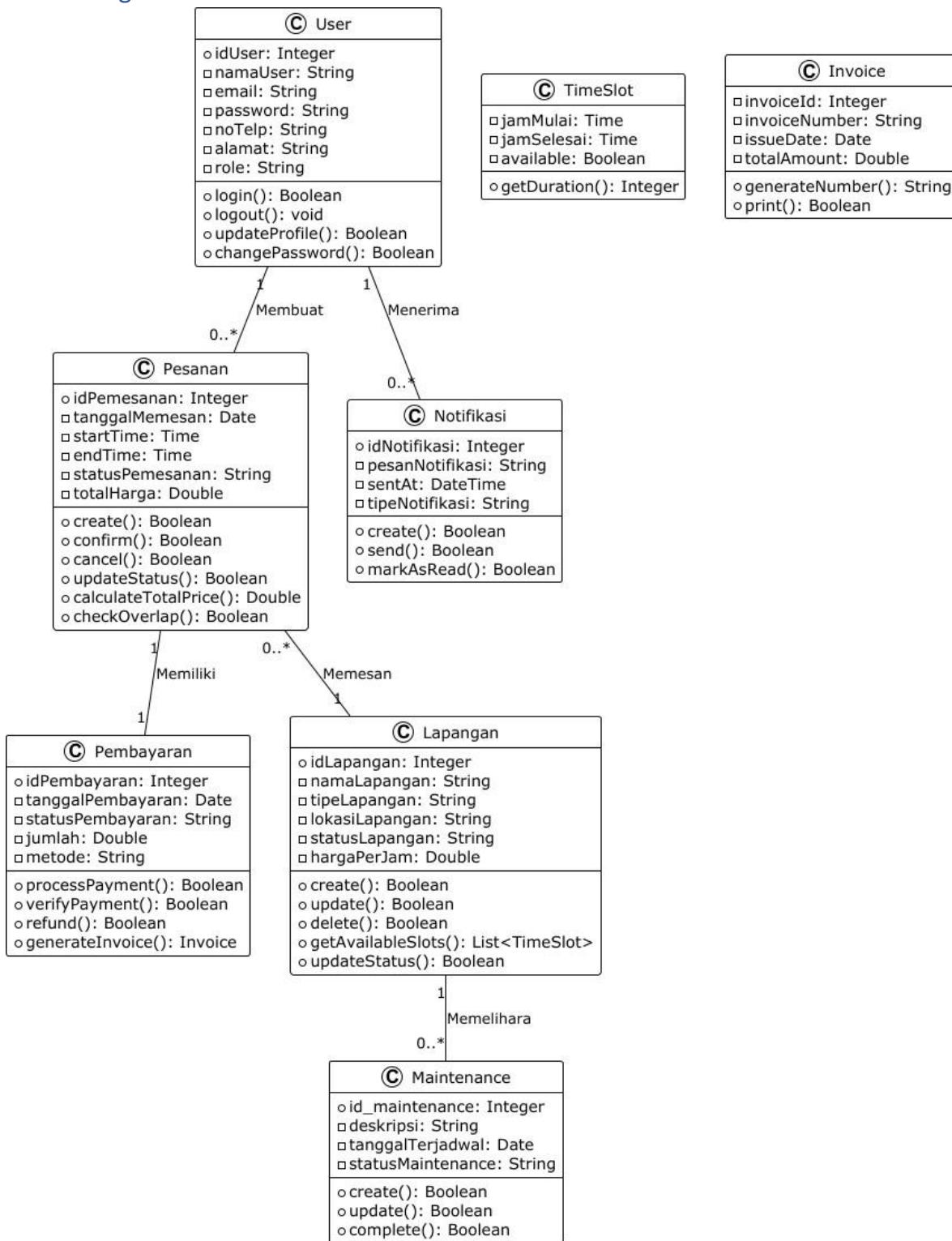
Aktor Utama	Pengguna/Member
Kondisi Sebelum	Pengguna sudah <i>login</i> dan Lapangan telah dipilih.
Kondisi Sesudah	Pemesanan terdaftar, pembayaran berhasil, dan status <i>booking</i> terkonfirmasi.
Alur Normal	1. Member <i>login</i> ke sistem. 2. Member memilih lapangan, tanggal, dan waktu. 3. Sistem membuat <i>booking</i> sementara & mengarahkan ke pembayaran. 4. Member melakukan pembayaran. 5. Sistem mengonfirmasi pembayaran dan meng-update status <i>booking</i> . 6. Sistem menampilkan konfirmasi <i>booking</i> .
Alur Alternatif	5.a. Jika pembayaran gagal: Sistem membatalkan <i>booking</i> sementara & menampilkan pesan gagal bayar.

## Kelola Lapangan

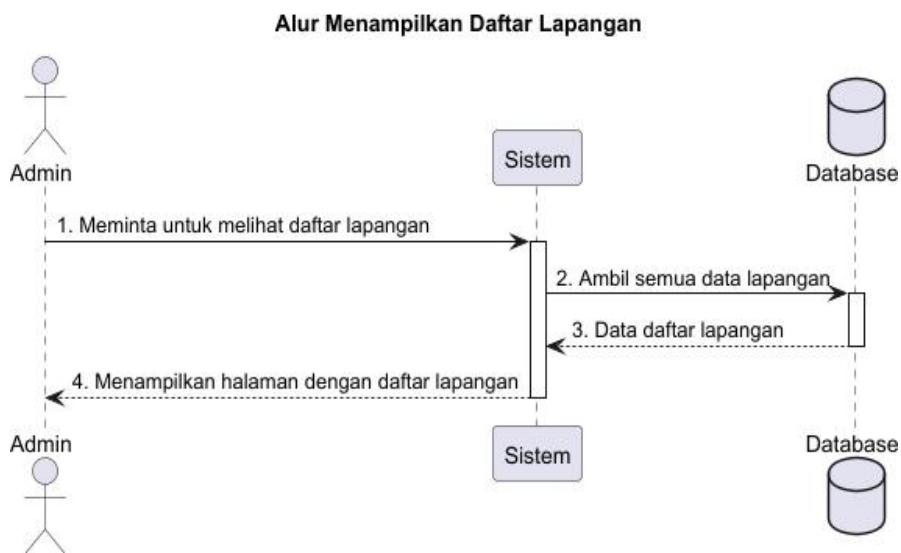
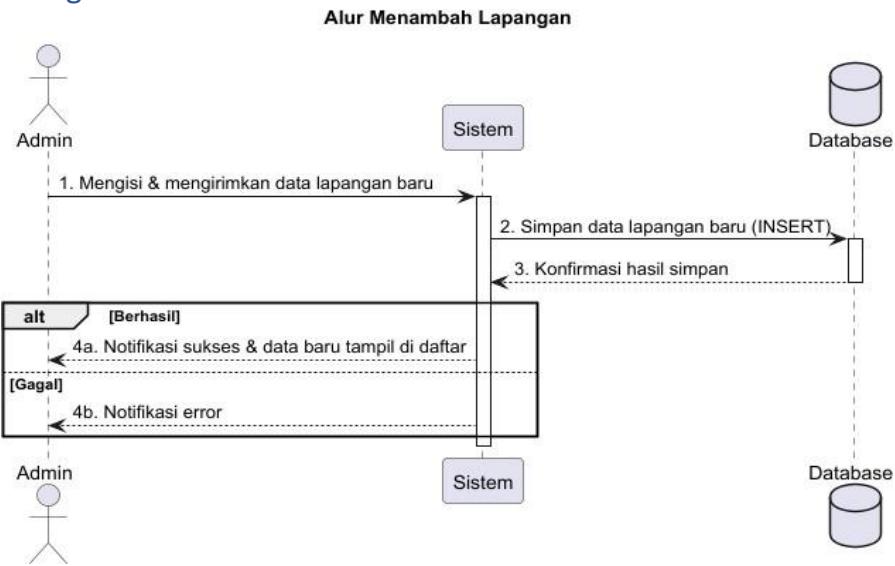
Aktor Utama	Admin/Employee
Tujuan	Menghapus data lapangan dari sistem.
Kondisi Sebelum	Admin/Employee sudah <i>login</i> .
Kondisi Sesudah	Lapangan terhapus dari sistem.
Alur Normal	1. Admin/Employee memilih lapangan untuk dihapus. 2. Sistem memeriksa apakah lapangan memiliki <i>booking</i> aktif. 3. Jika tidak ada <i>booking</i> aktif: Admin/Employee mengonfirmasi hapus. 4. Sistem menghapus lapangan dan data terkait. 5. Sistem menampilkan pesan sukses.

<b>Aktor Utama</b>	<b>Admin/Employee</b>
<b>Alur Alternatif</b>	3.a. Jika ada <i>booking</i> aktif: Sistem menampilkan pesan <i>error</i> . 3.b. Jika Admin/Employee membatalkan: Sistem membatalkan penghapusan.

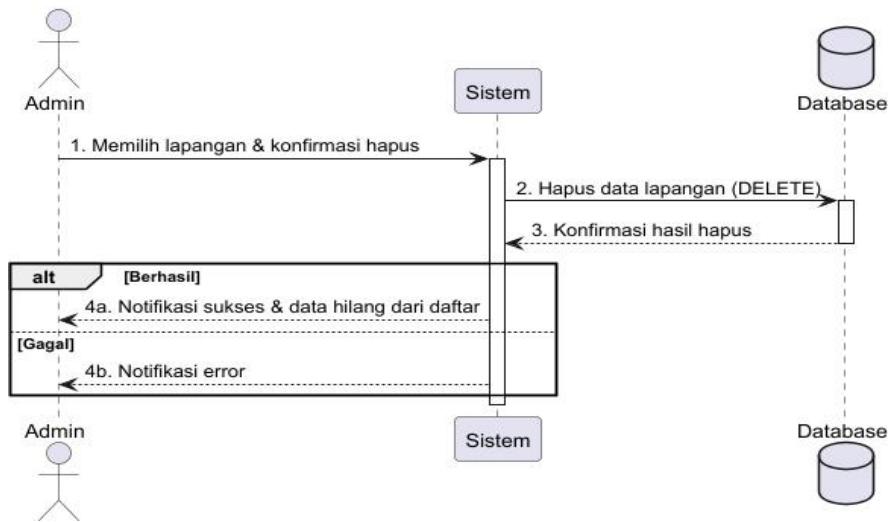
### 3.3 Class Diagram



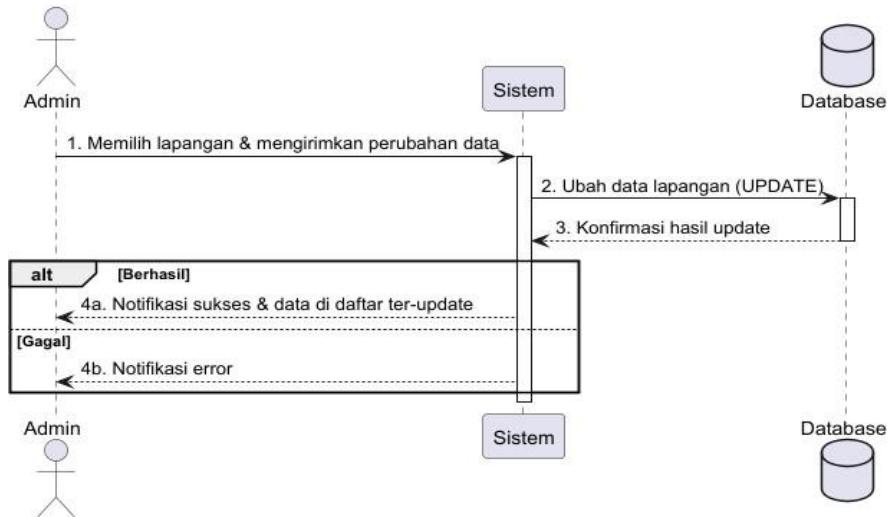
### 3.3 Sequence Diagram



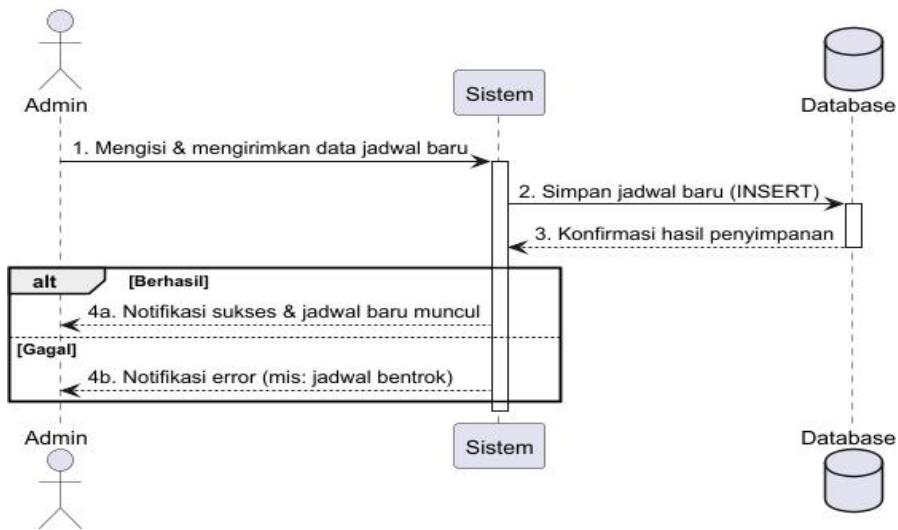
### Alur Menghapus Lapangan



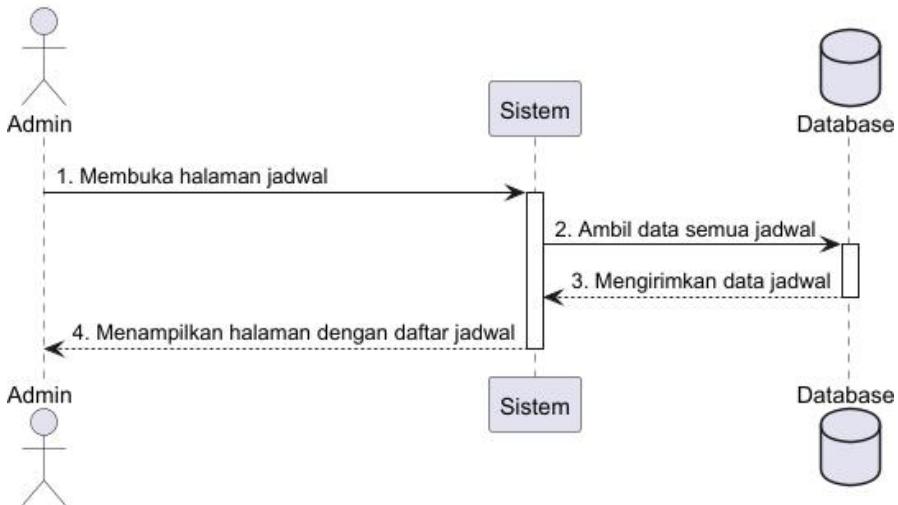
### Alur Mengubah Lapangan



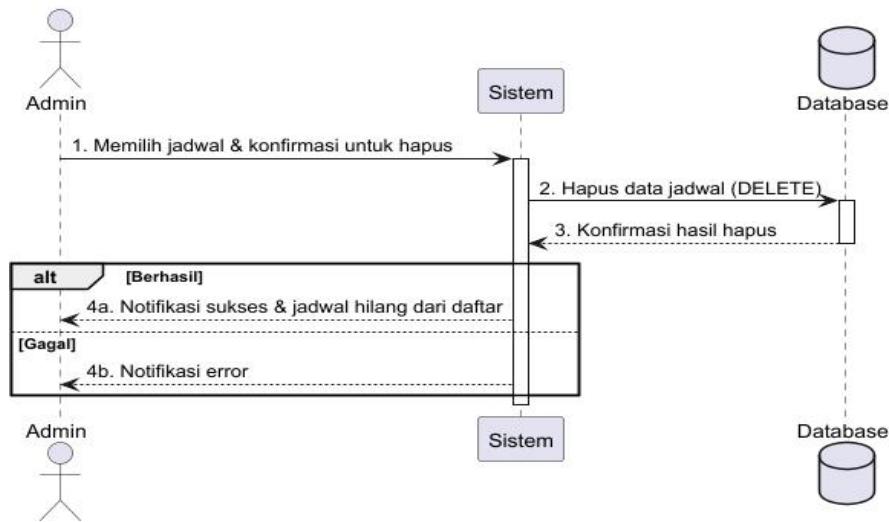
### Alur Membuat Jadwal Baru



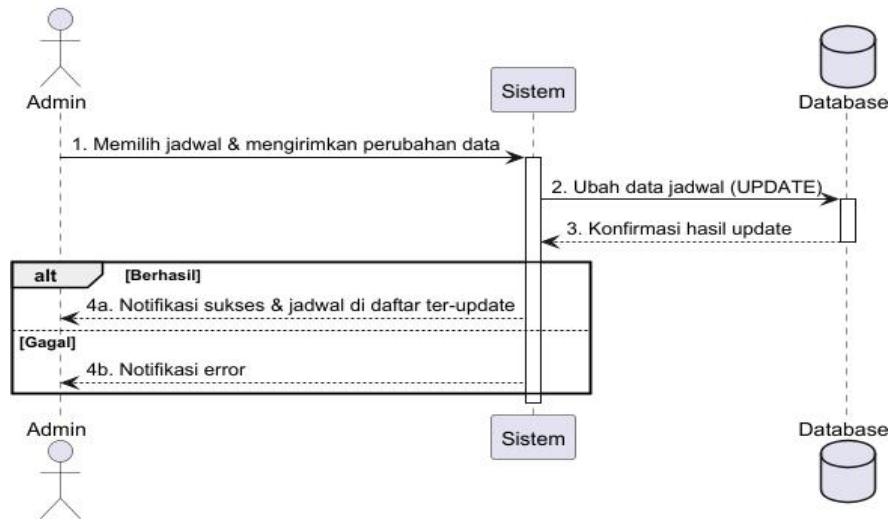
### Alur Menampilkan Daftar Jadwal



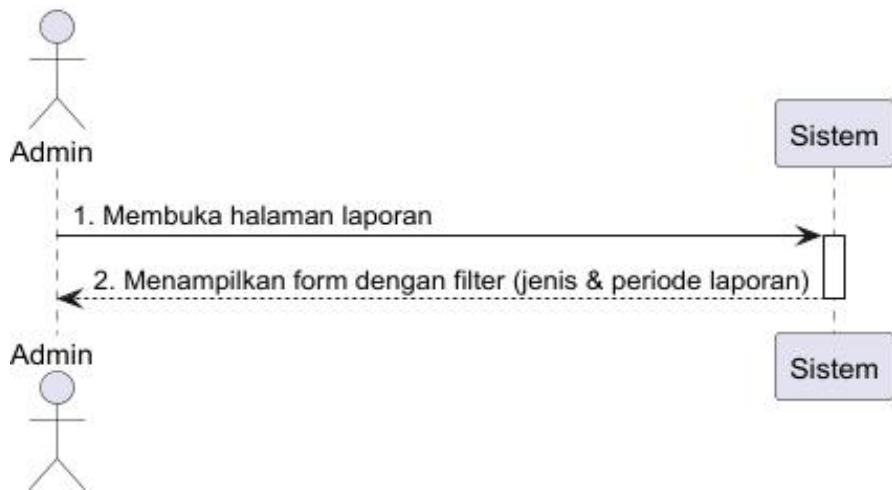
### Alur Menghapus Jadwal



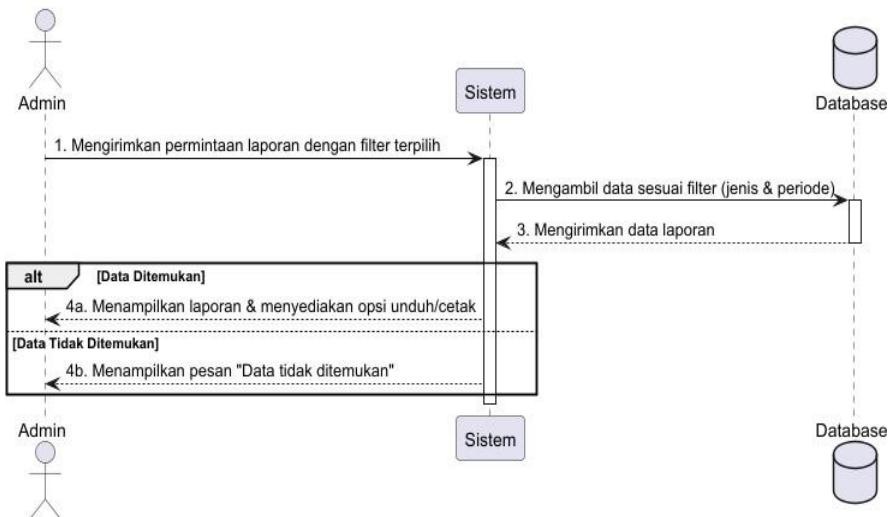
### Alur Mengubah Jadwal



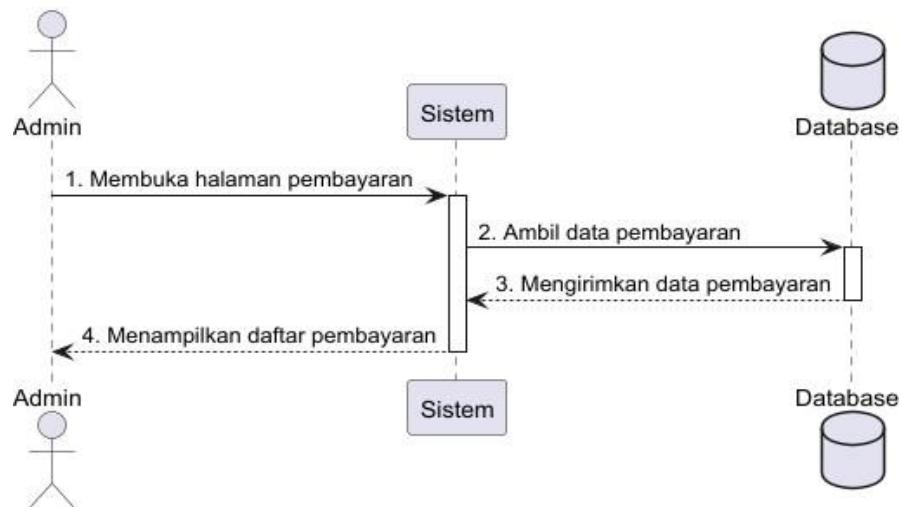
### Alur Memilih Kriteria Laporan



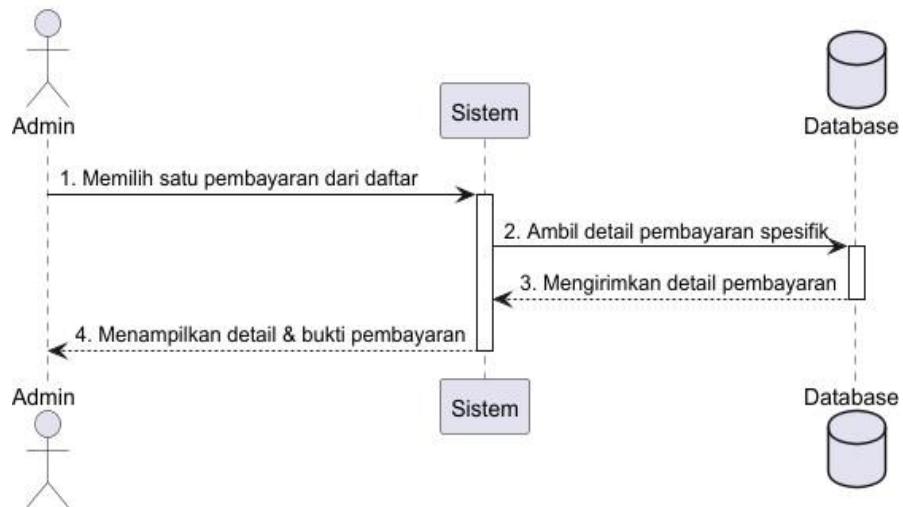
### Alur Menghasilkan Laporan



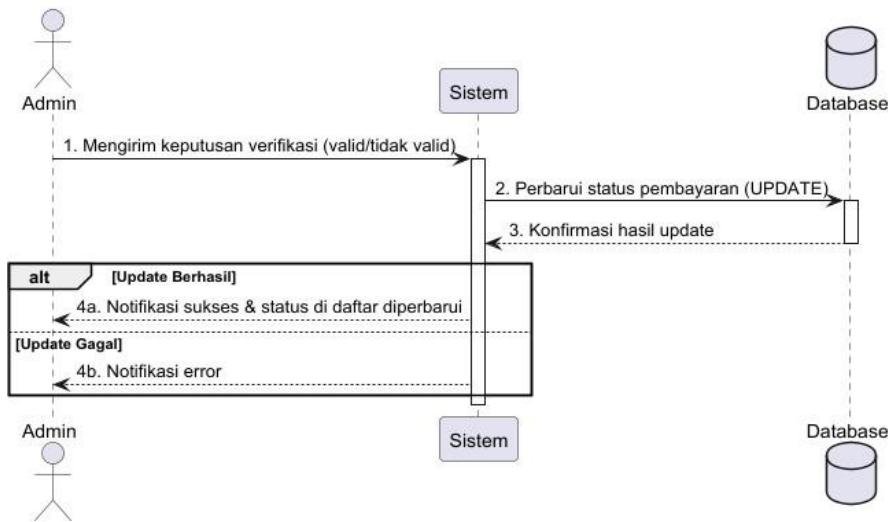
### Alur Melihat Daftar Pembayaran



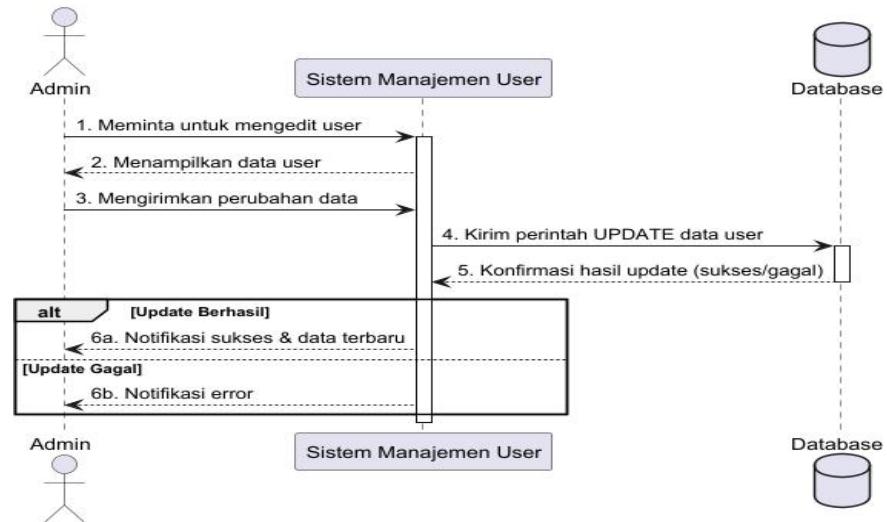
### Alur Melihat Detail Pembayaran



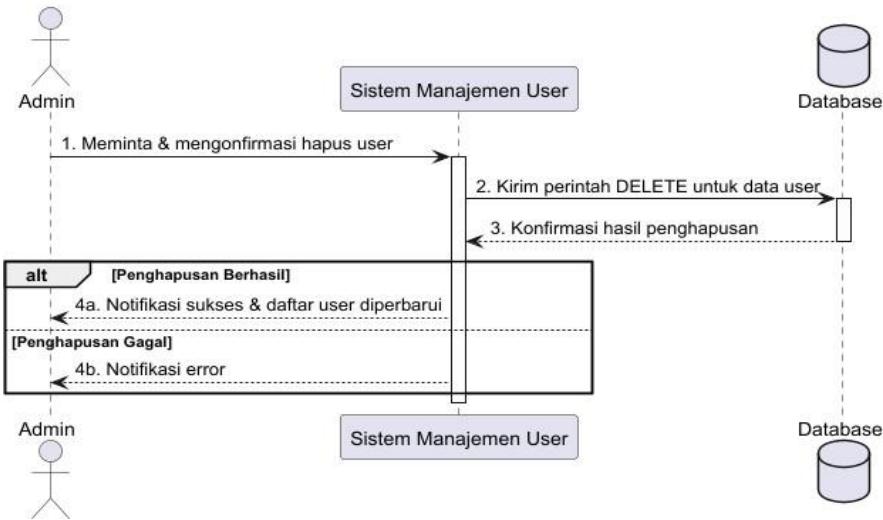
### Alur Memverifikasi Pembayaran



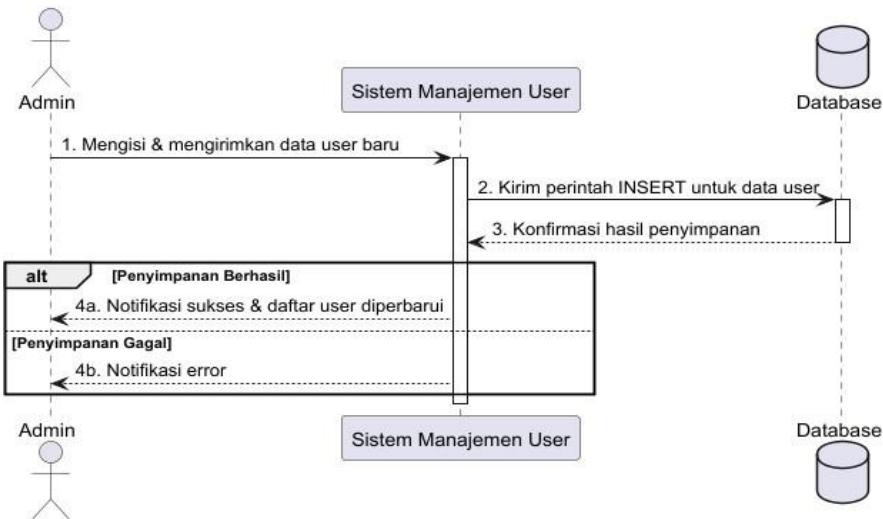
### Alur Edit User



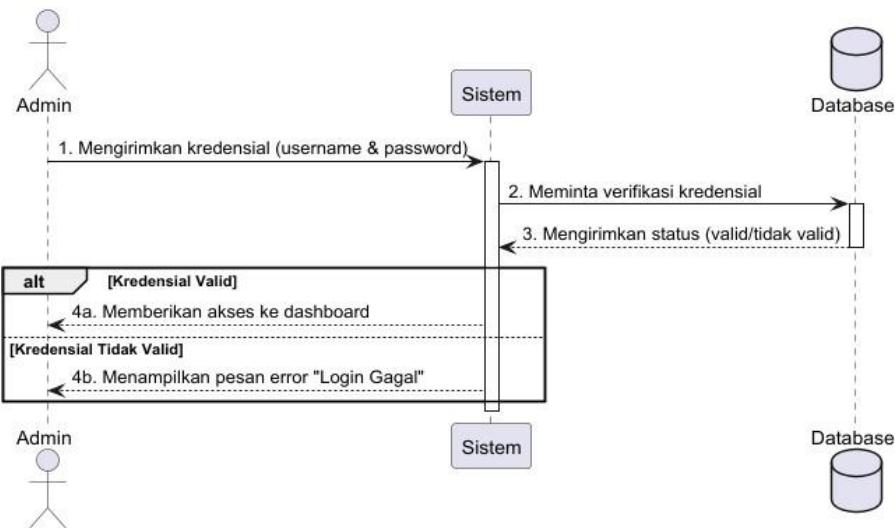
### Alur Hapus User



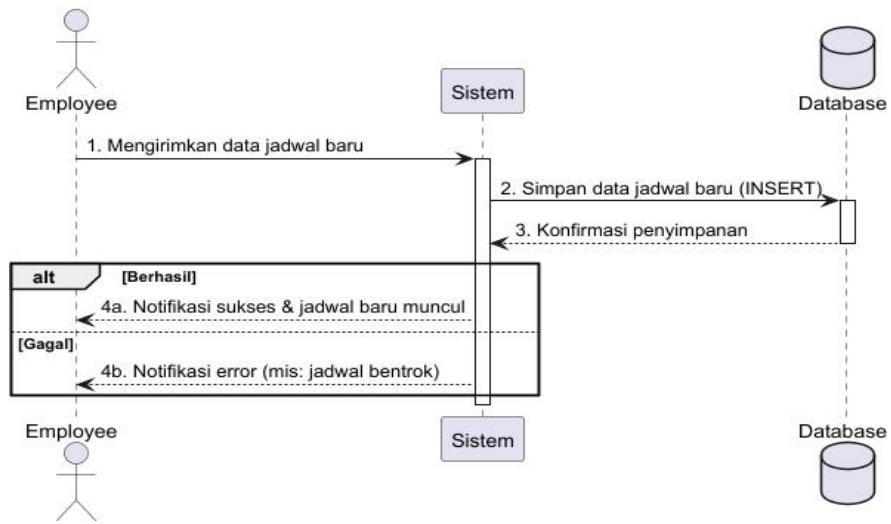
### Alur Tambah User



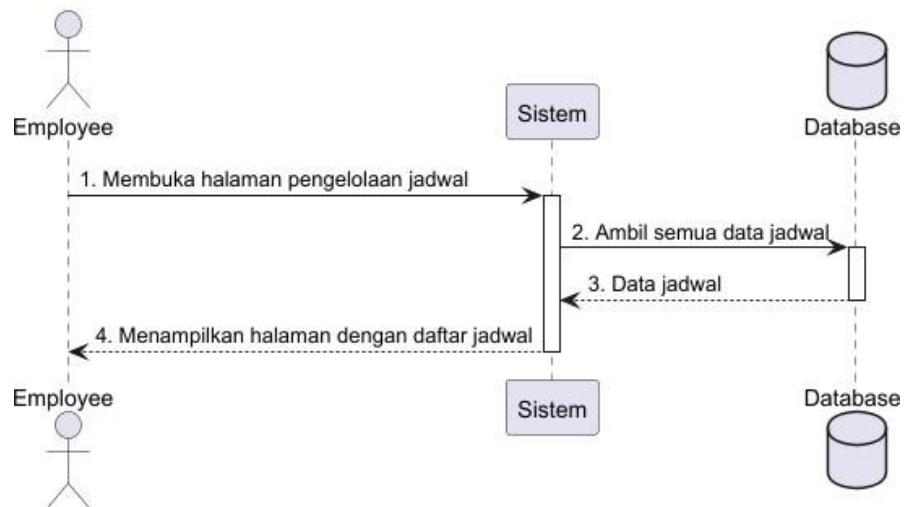
### Alur Login Admin



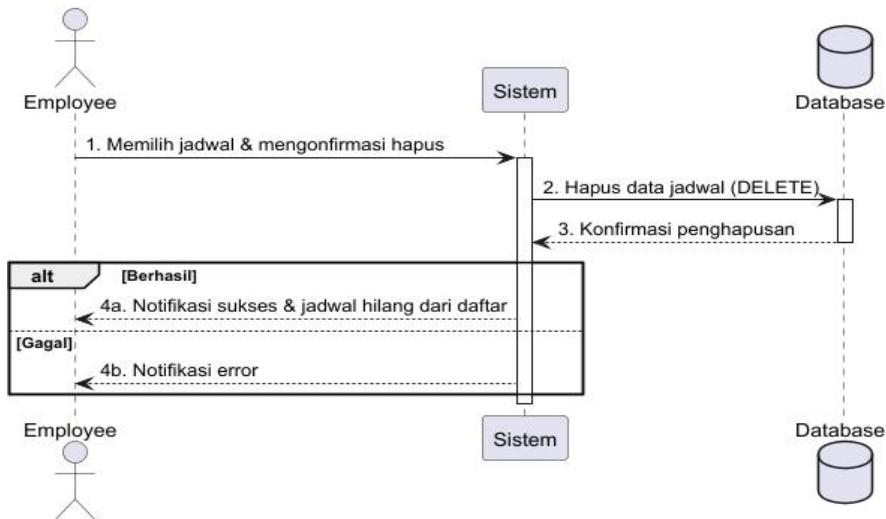
### Alur Menambah Jadwal Baru



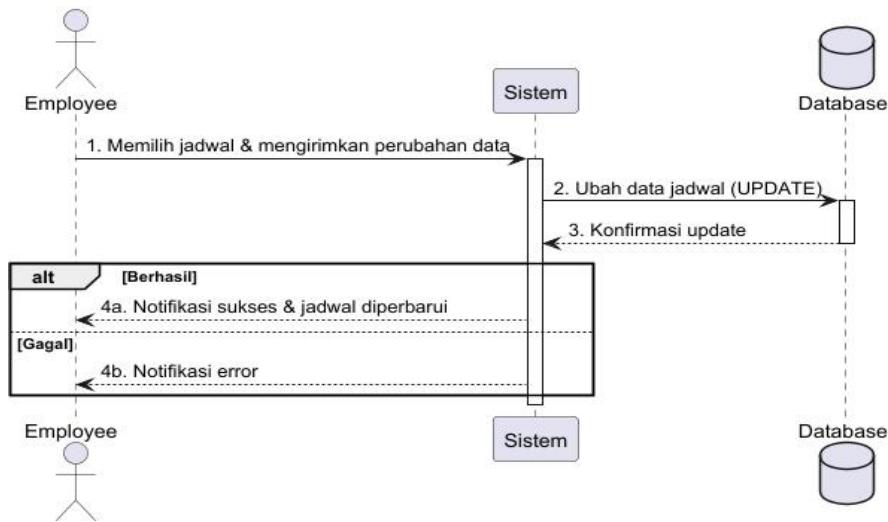
### Alur Menampilkan Daftar Jadwal



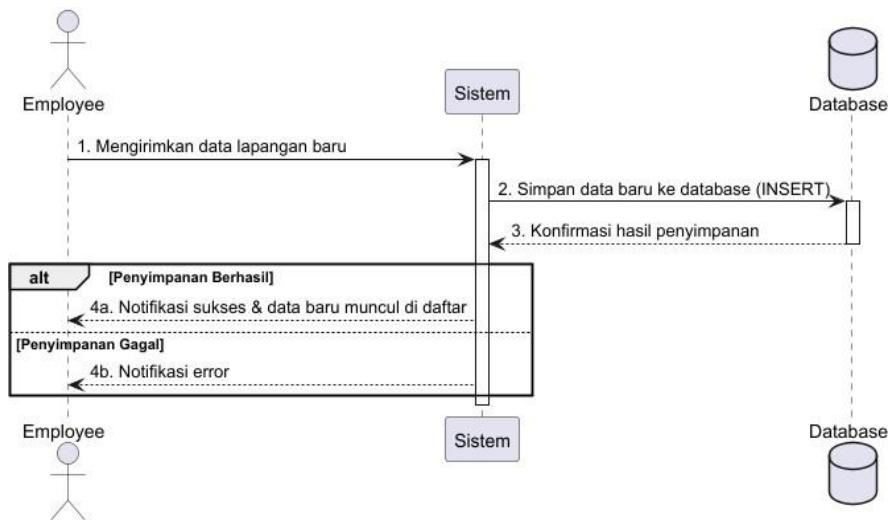
### Alur Menghapus Jadwal



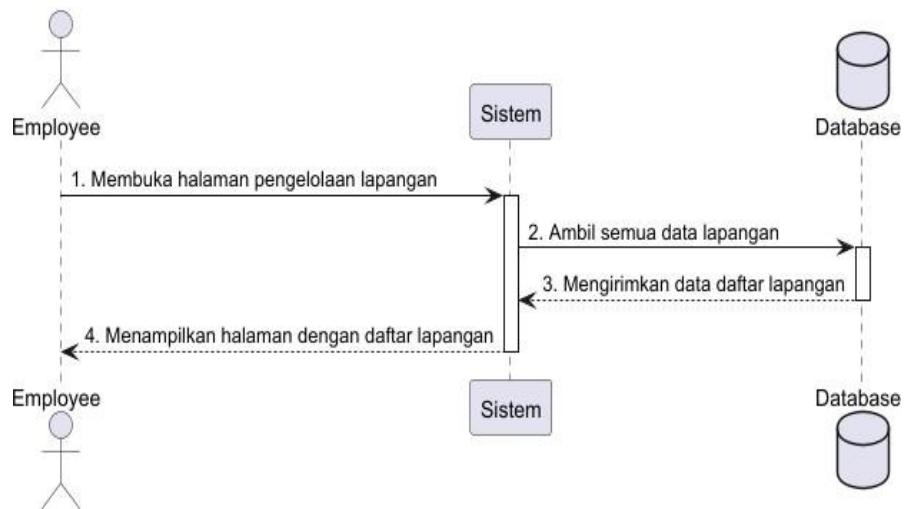
### Alur Mengubah Jadwal



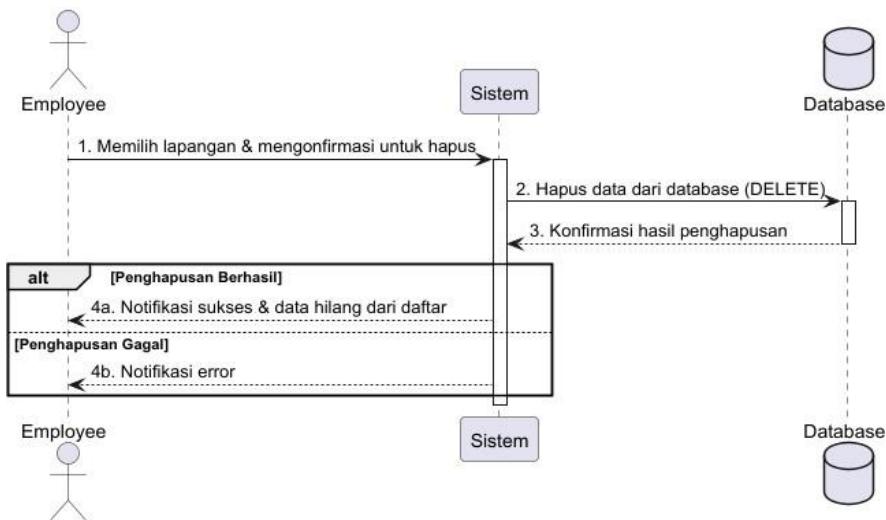
### Alur Menambah Data Lapangan (Create)



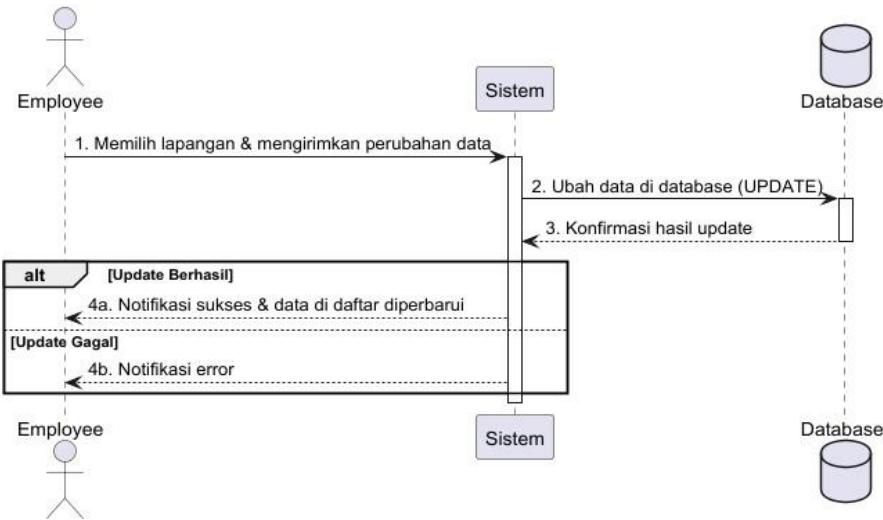
### Alur Menampilkan Daftar Lapangan



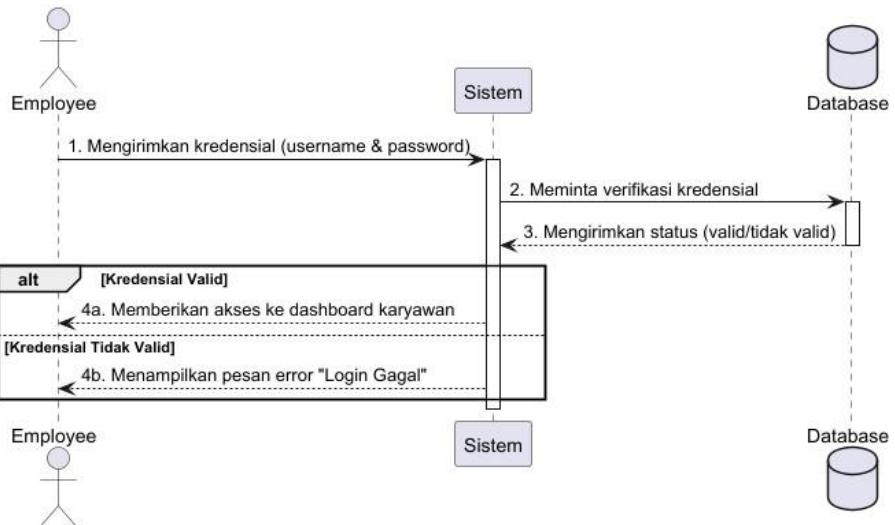
### Alur Menghapus Data Lapangan (Delete)



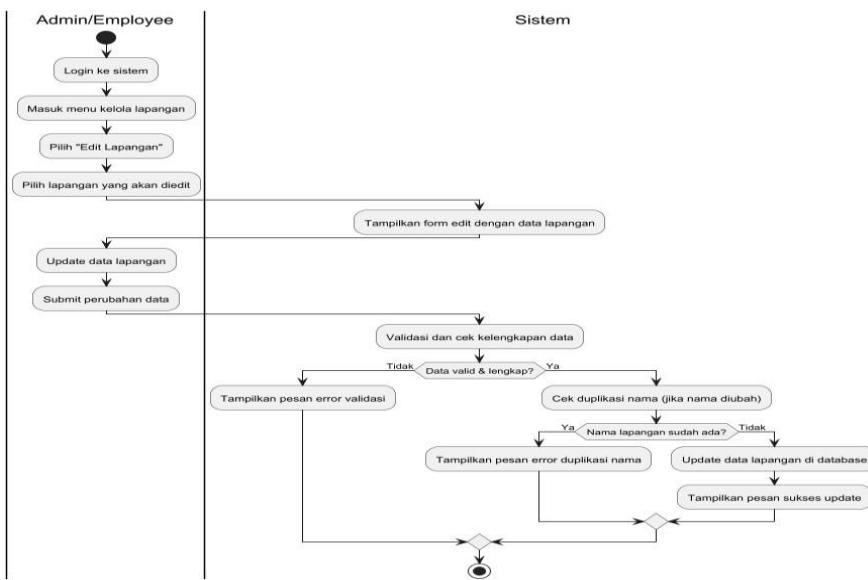
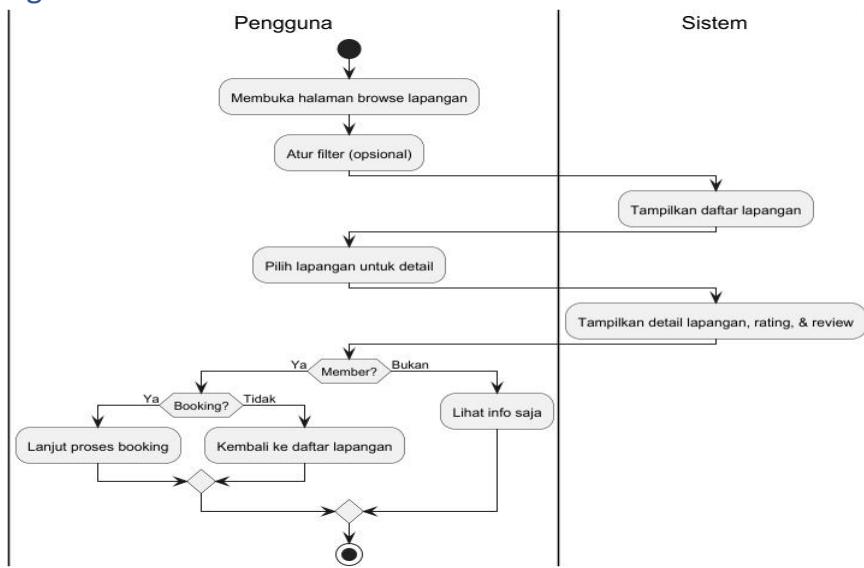
### Alur Mengubah Data Lapangan (Update)

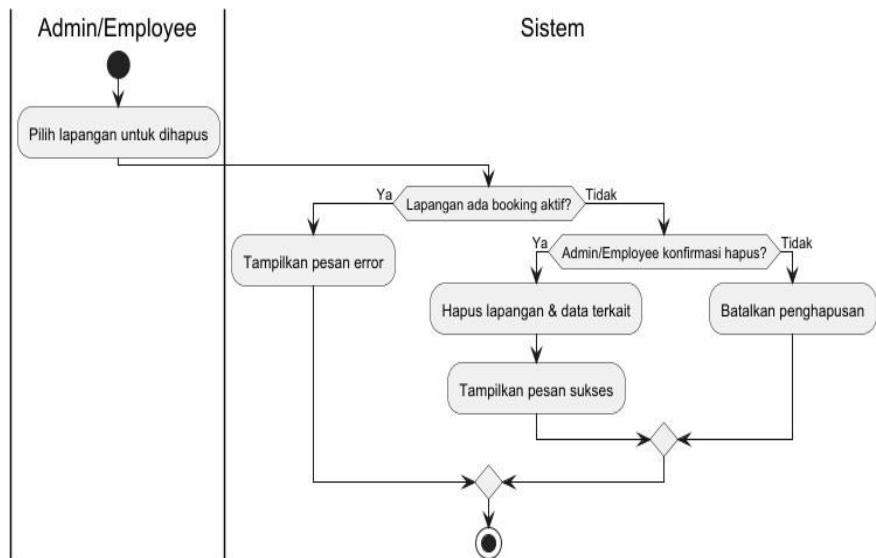
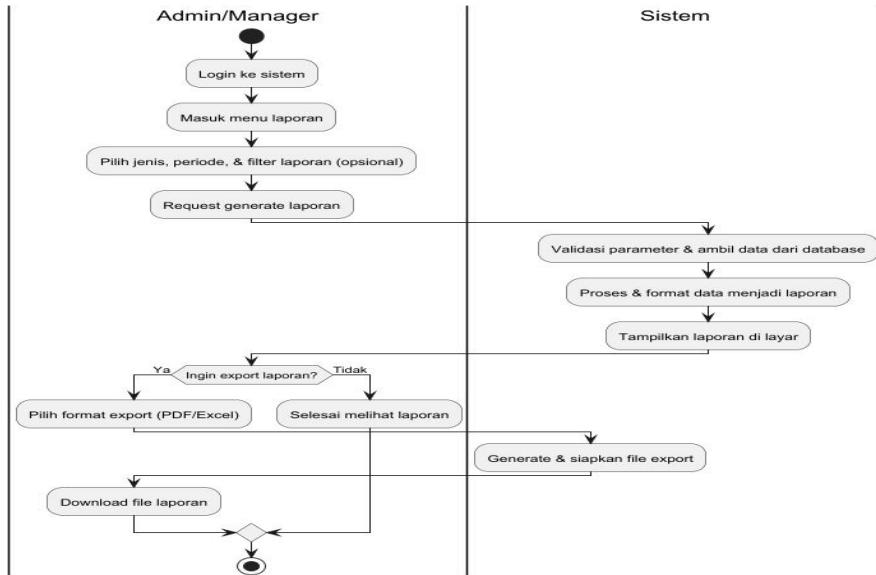


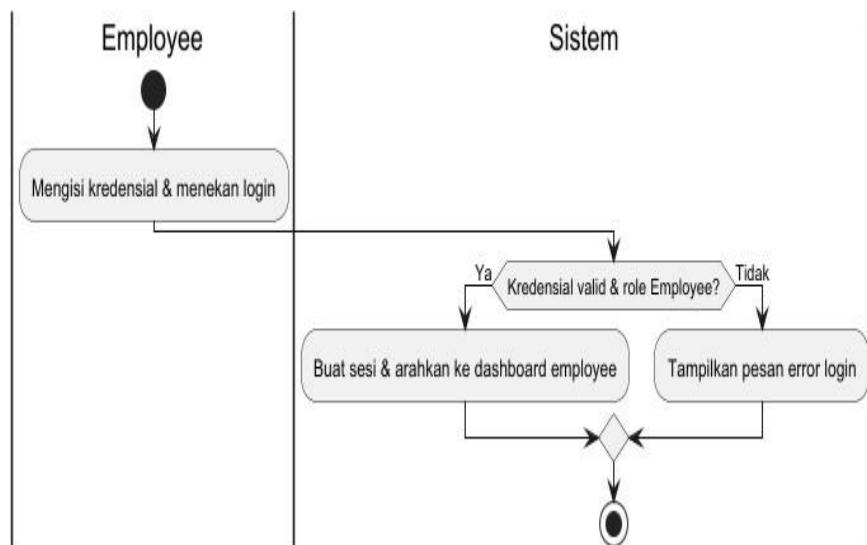
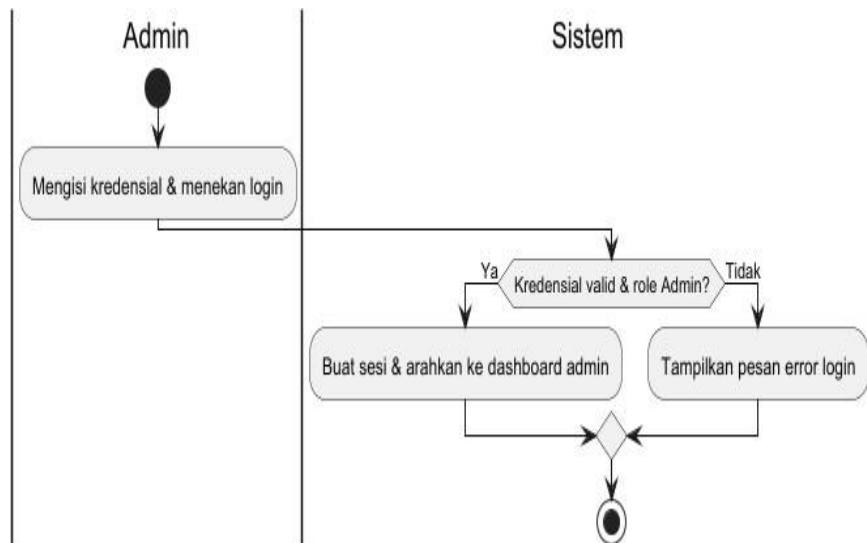
### Alur Login Karyawan

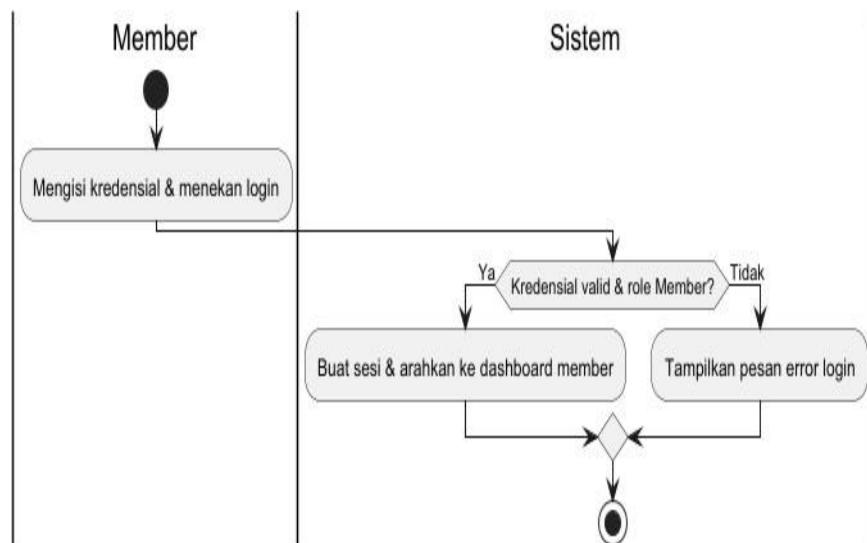
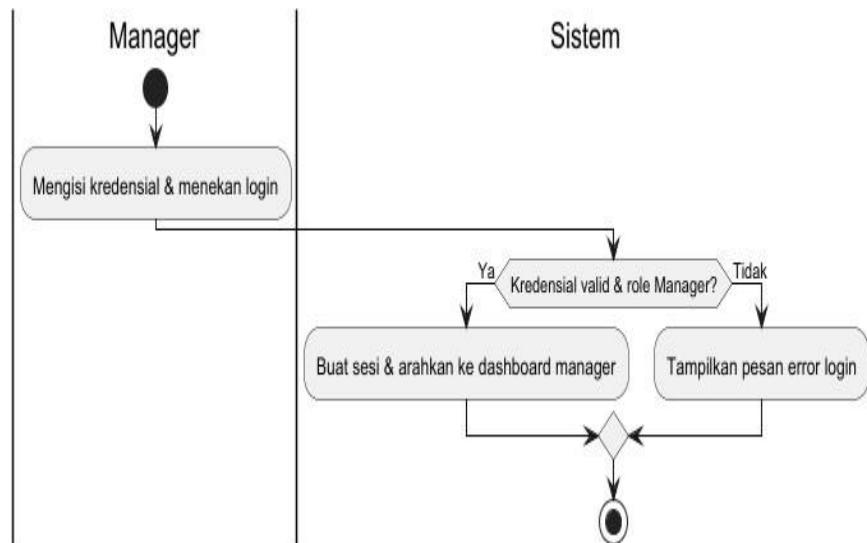


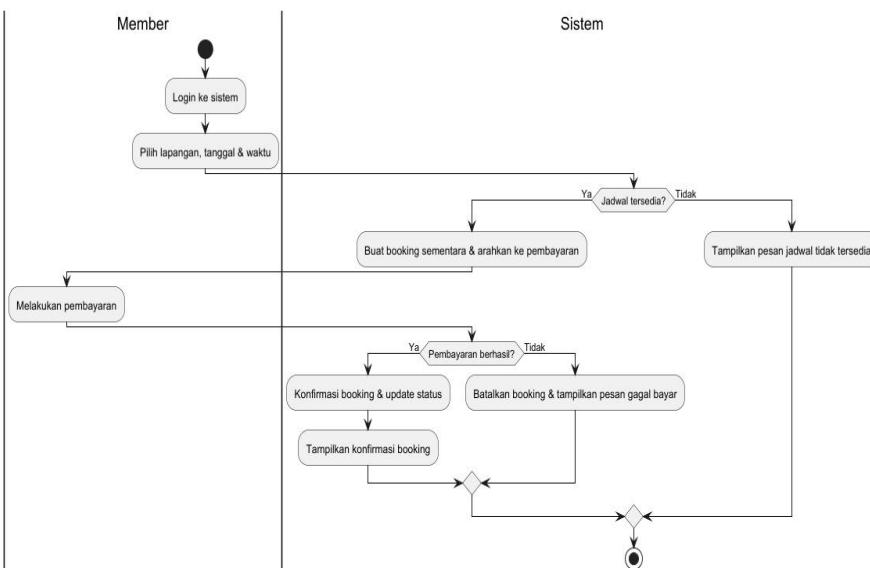
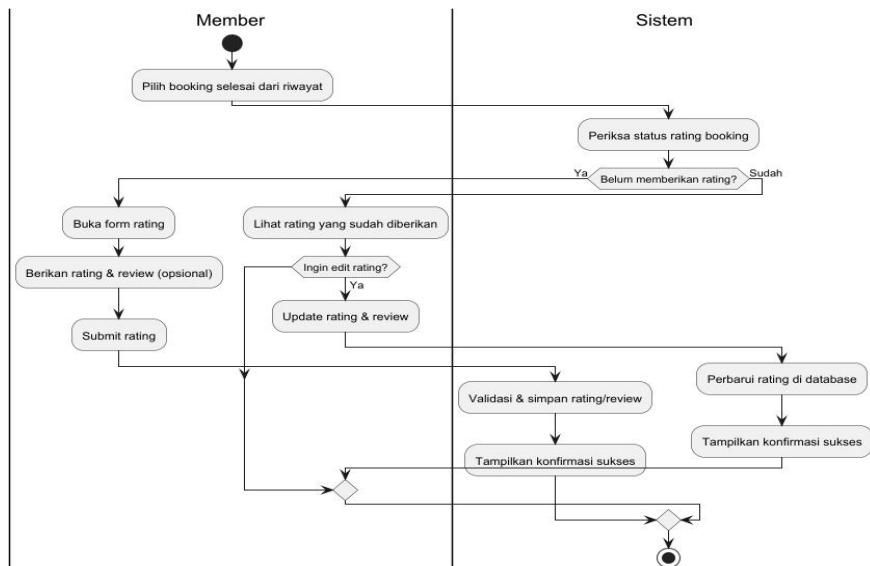
### 3.3 Activity Diagram

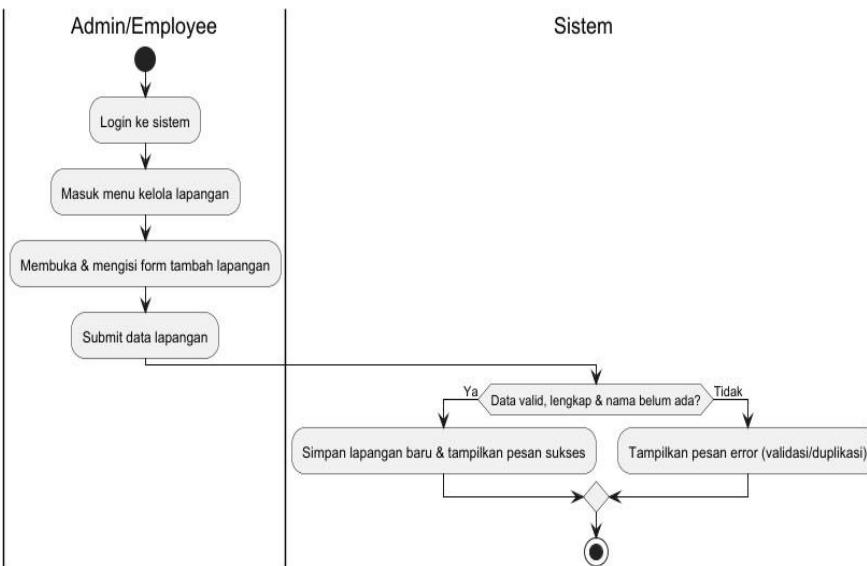
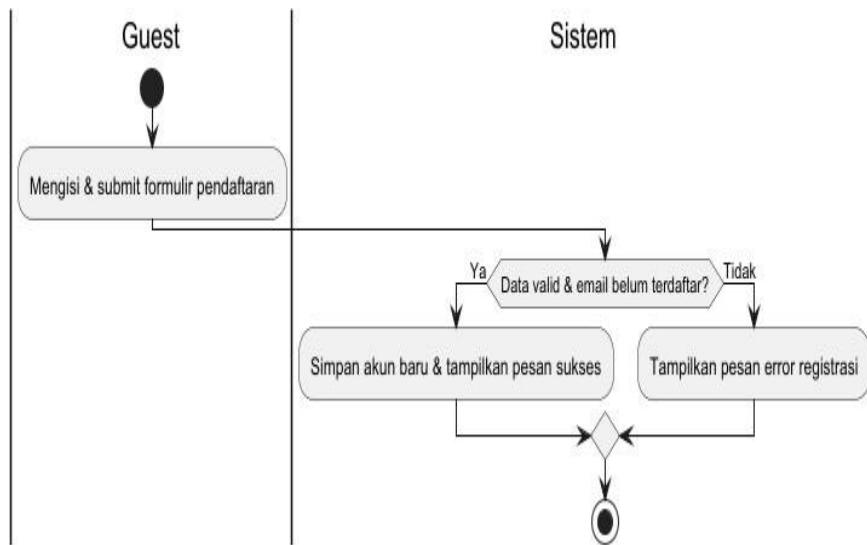






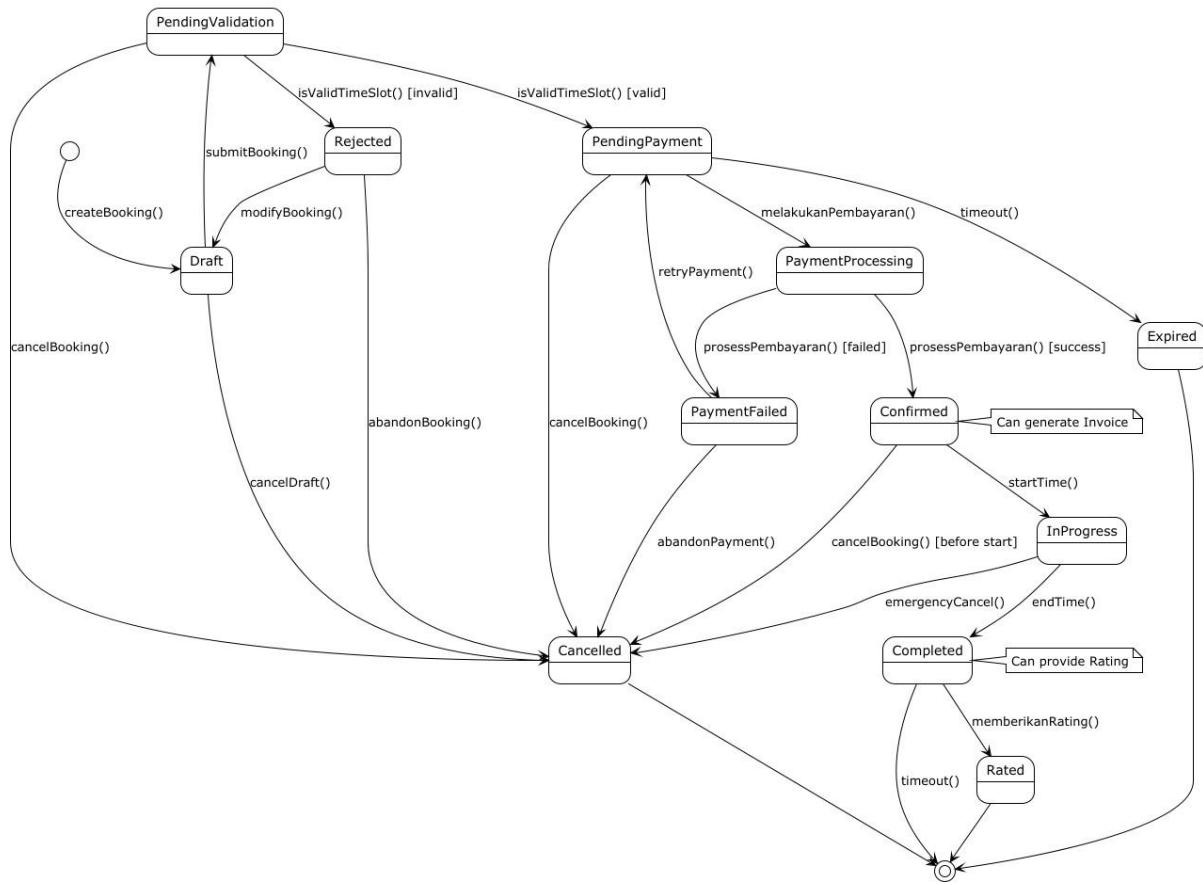




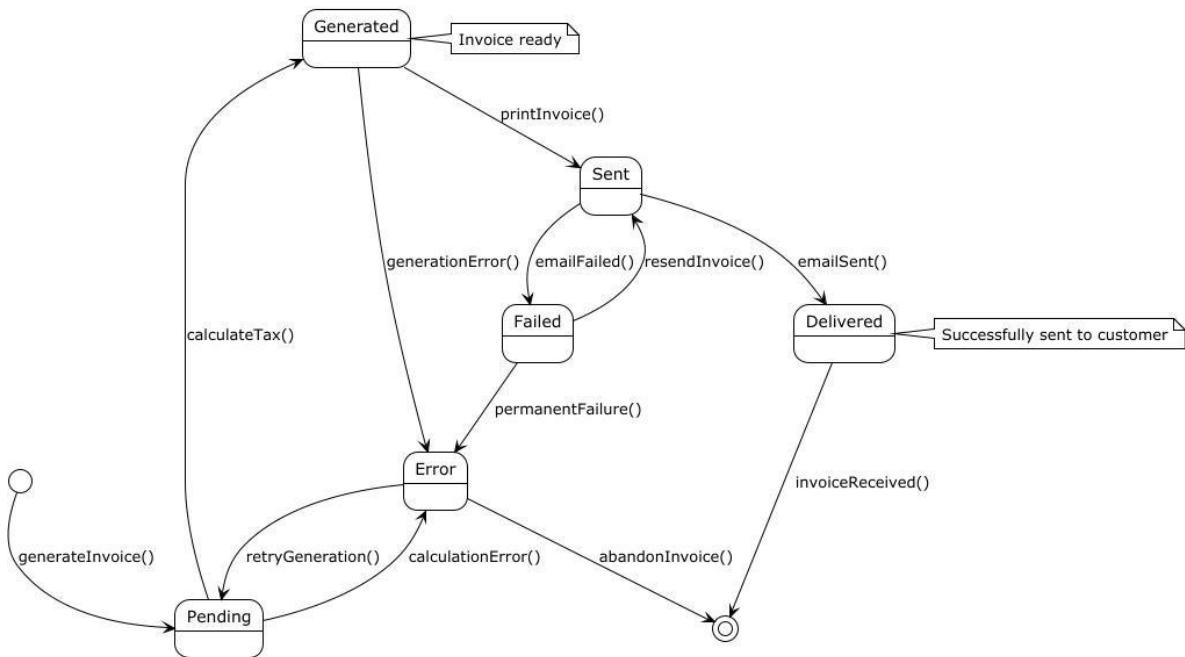


### 3.3 State Diagram

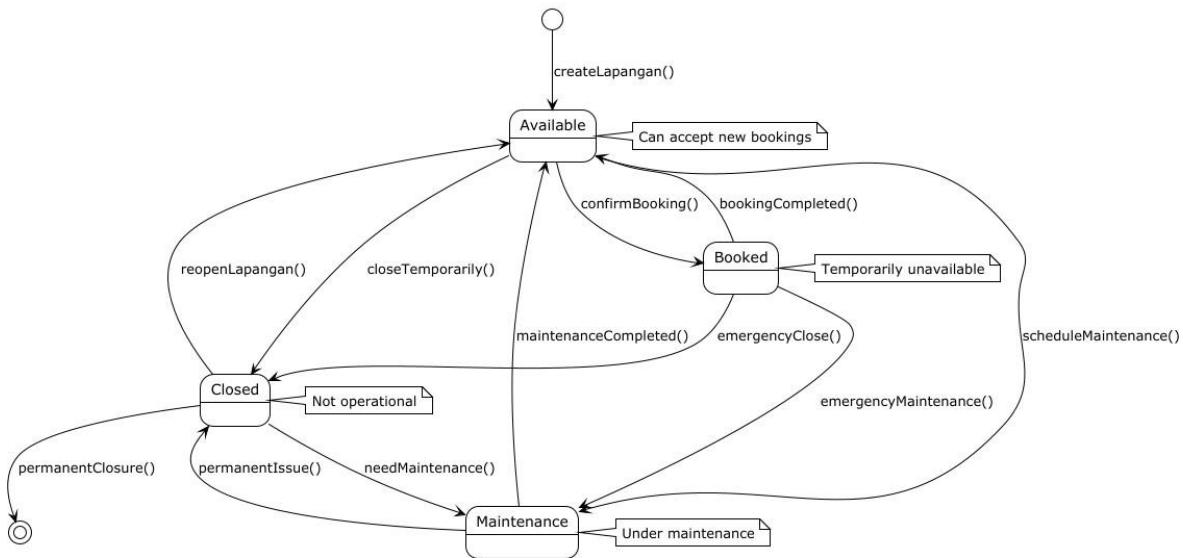
## **State Diagram - Booking Lifecycle**



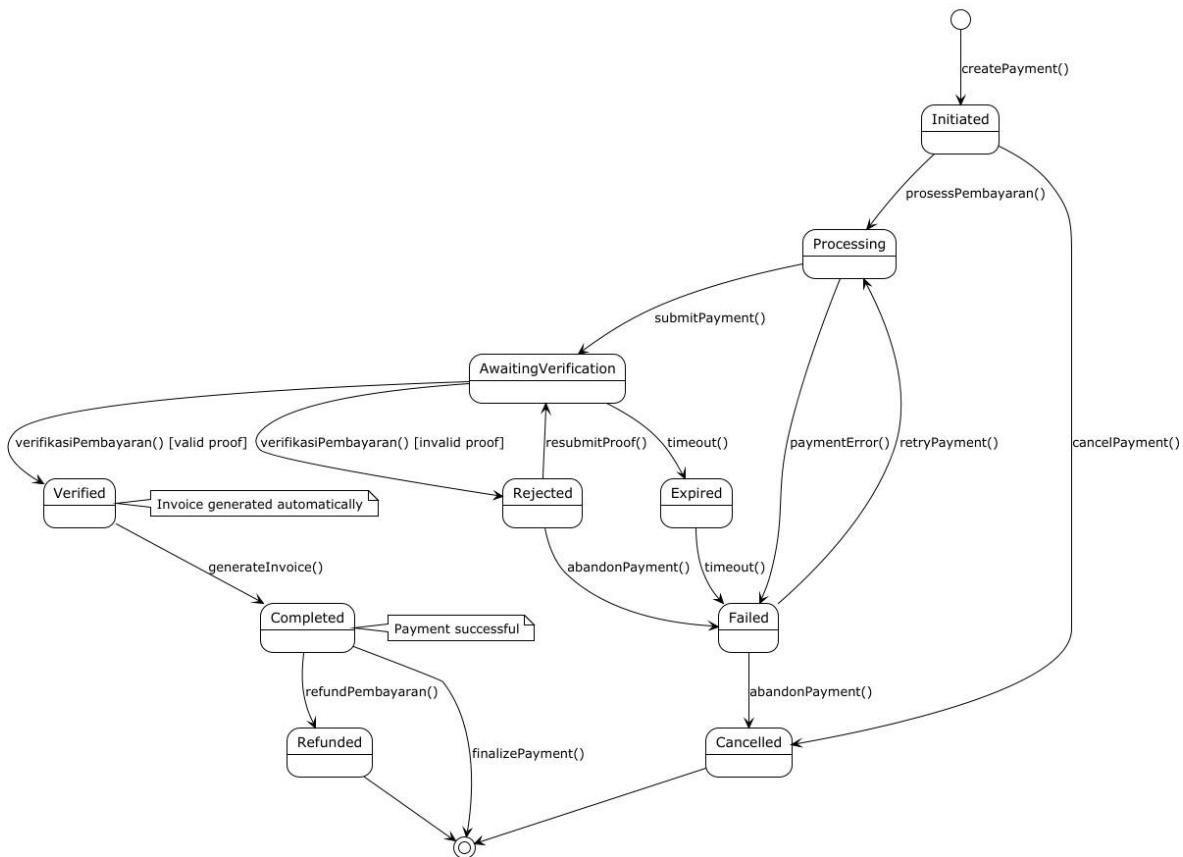
## **State Diagram - Invoice Generation**



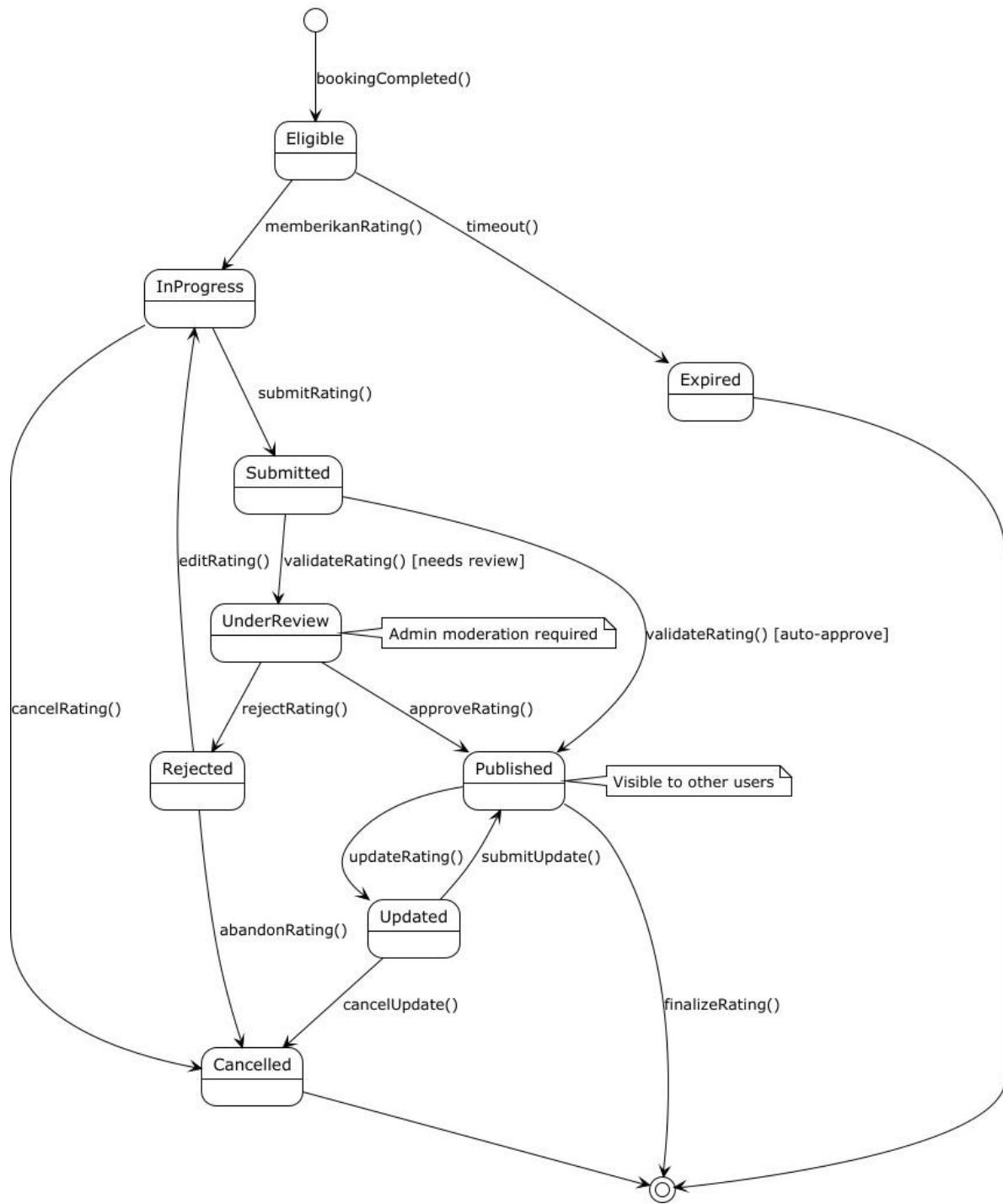
**State Diagram - Lapangan Status**



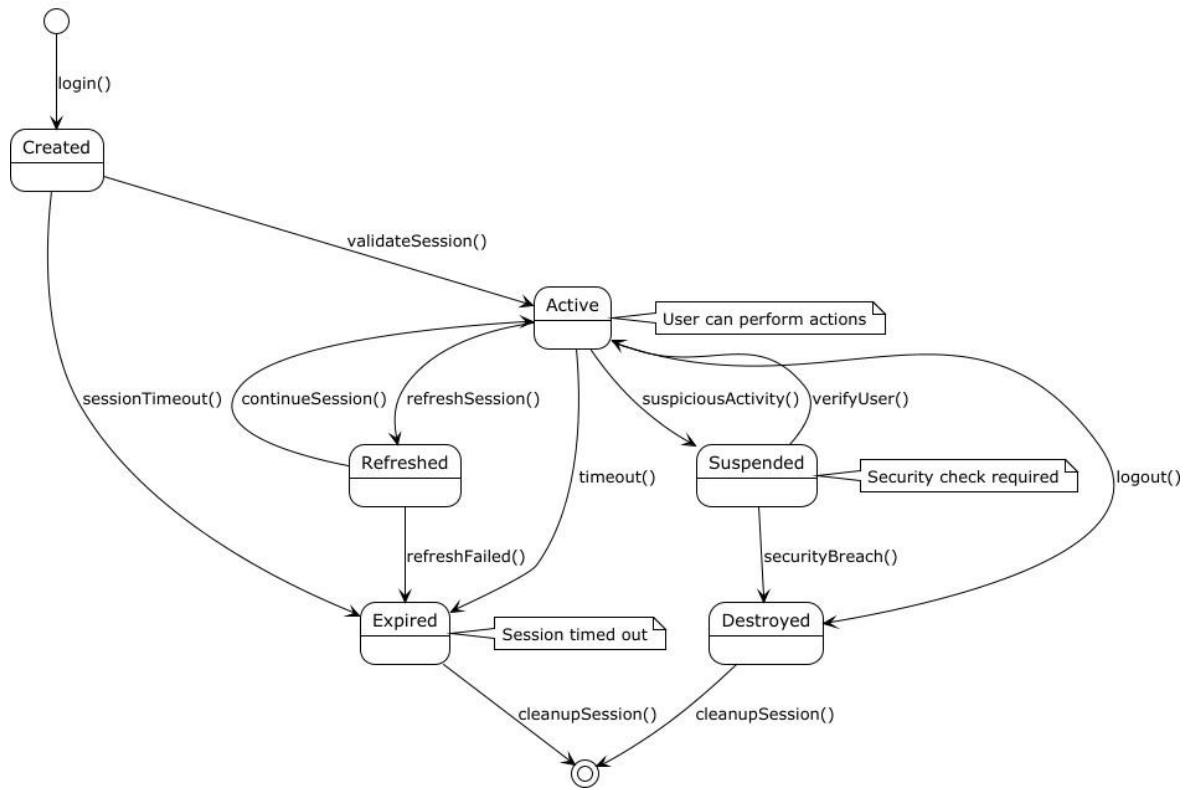
**State Diagram - Payment Processing**



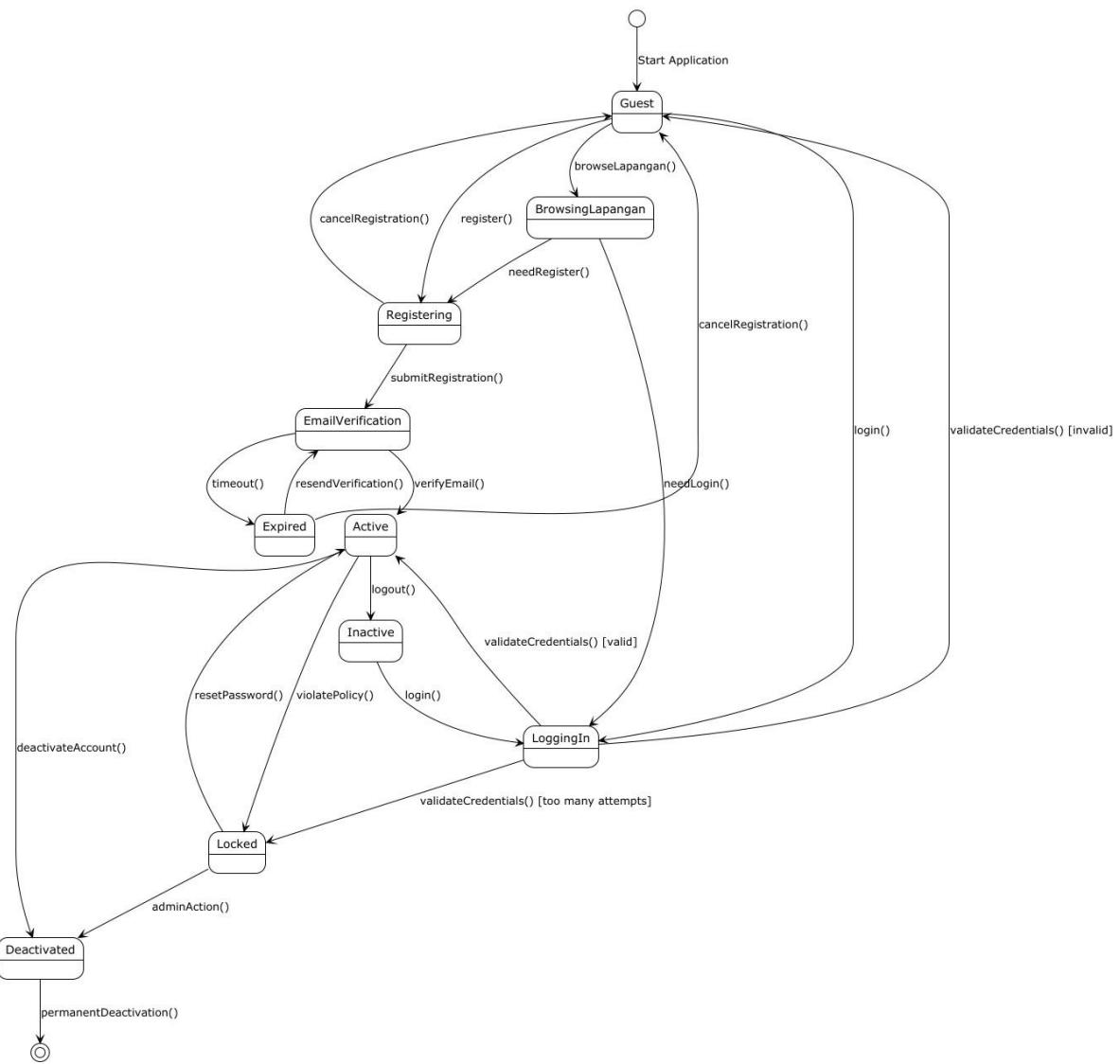
## State Diagram - Rating System



## State Diagram - Session Management

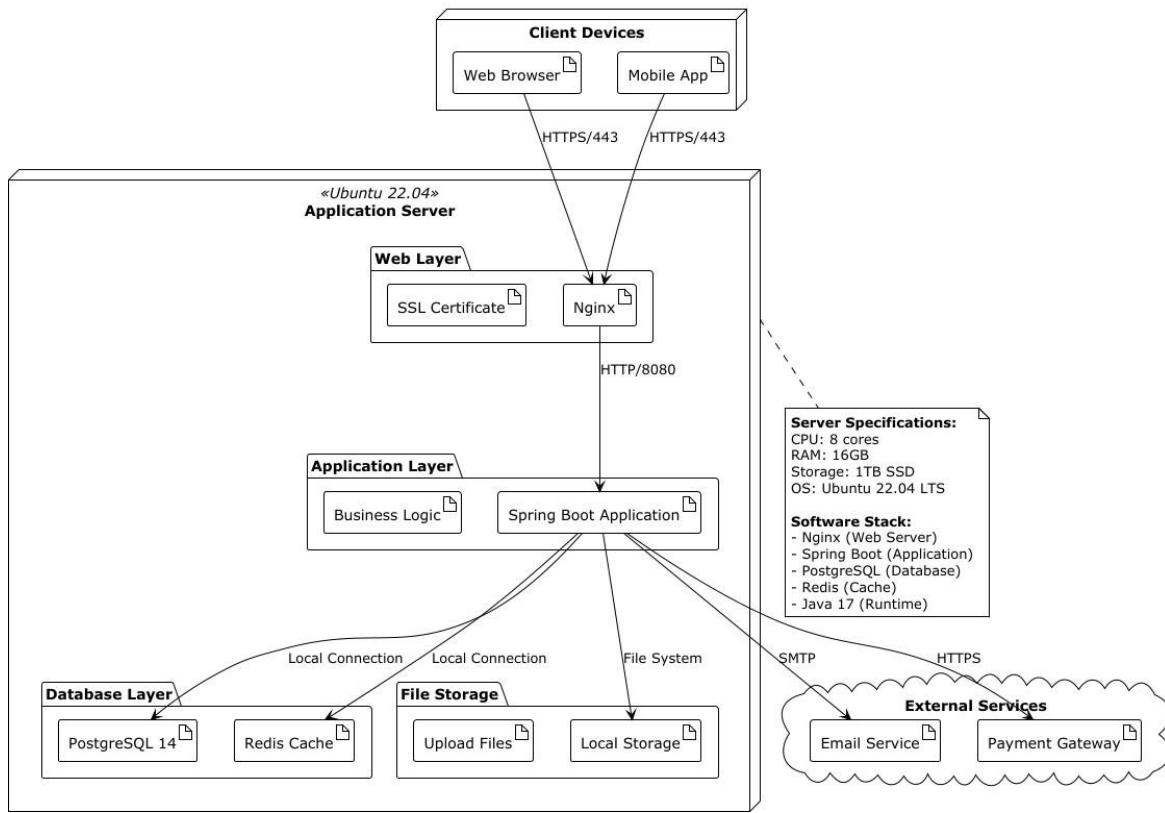


**State Diagram - User Authentication**



### 3.3 Deployment Diagram

**Deployment Diagram - Sistem Booking Lapangan Olahraga**

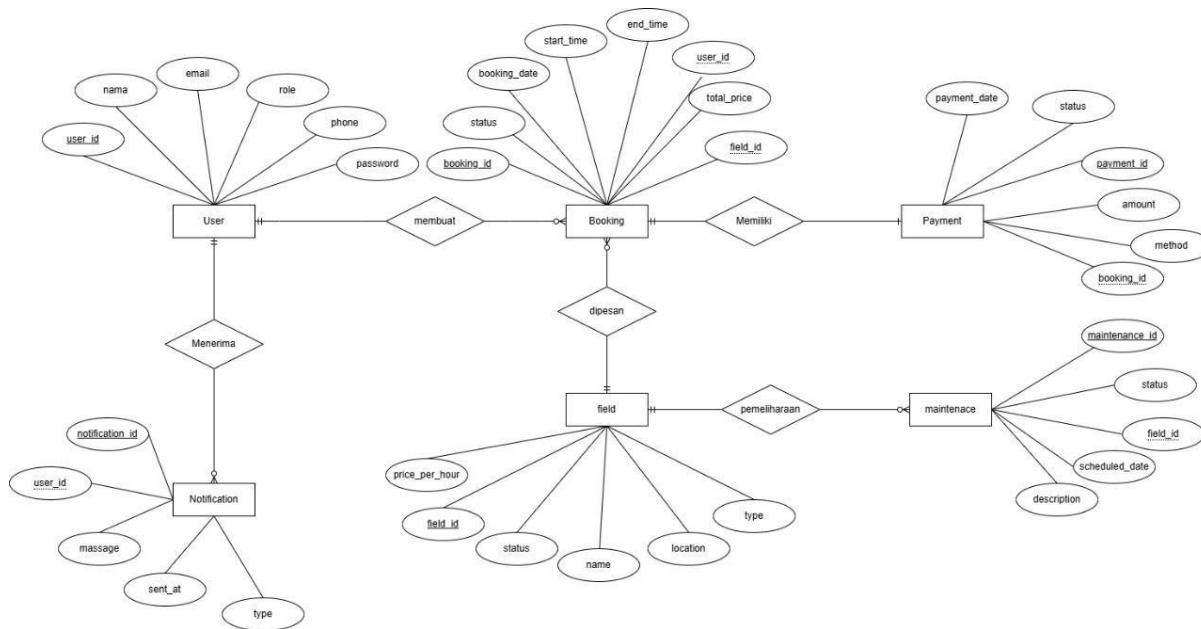


## Bab IV Data Design

Bagian ini menjelaskan arsitektur penyimpanan data pada sistem aplikasi Booking Lapangan Olahraga. Perancangan data berperan sebagai kerangka yang mengatur struktur, hubungan, dan integritas basis data agar seluruh proses operasional sistem dapat berjalan secara terpadu. Tujuan utamanya adalah menjamin bahwa seluruh entitas data, mulai dari informasi pengguna, jadwal pemesanan, ketersediaan lapangan, hingga data referensi pendukung, dapat dikelola secara sistematis, konsisten, dan sesuai dengan kebutuhan fungsional aplikasi.

Untuk mempermudah pemahaman secara konseptual, arsitektur data ini akan disajikan dalam bentuk Entity-Relationship Diagram (ERD). Diagram tersebut memvisualisasikan keterkaitan antarantara entitas, atribut kunci, serta kardinalitas hubungan secara grafis. Selanjutnya, spesifikasi rinci setiap entitas, termasuk tipe data, batasan (constraints), dan aturan bisnis yang berlaku, akan diuraikan dalam skema tabel relasional beserta diagram relasi antar objek.

### 4.1 Logical Design



## 4.2 Physical Design

Struktur basis data akan diimplementasikan berdasarkan ERD, dengan tabel-tabel utama berikut:

### Tabel User

Kolom:

ID\_User (Primary Key),

Nama,

Email,

Password,

No\_Telepon,

Peran,

Alamat;

### Tabel Lapangan

Kolom:

ID\_Lapangan (Primary Key),

Nama\_Lapangan,

Tipe\_Lapangan,

Lokasi,

Harga\_Per\_Jam,

Status\_Lapangan,

Deskripsi;

### Tabel Pemesanan

Kolom:

ID\_Pemesanan (Primary Key),

ID\_User (Foreign Key) mengacu pada tabel User,

ID\_Lapangan (Foreign Key) mengacu pada tabel Lapangan,

Tanggal\_Pemesanan,

Tanggal\_Main,

Jam\_Mulai,

Jam\_Selesai,

Status\_Pemesanan,

Catatan;

### **Tabel Pembayaran**

Kolom:

ID\_Pembayaran (Primary Key),  
ID\_Pemesanan (Foreign Key) mengacu pada tabel Pemesanan,  
Tanggal\_Pembayaran,  
Metode\_Pembayaran,  
Jumlah\_Pembayaran,  
Status\_Pembayaran,  
Bukti\_Pembayaran;

### **Tabel Maintenance**

Kolom:

ID\_Maintenance (Primary Key),  
ID\_Lapangan (Foreign Key) mengacu pada tabel Lapangan,  
Tanggal\_Maintenance,  
Jenis\_Maintenance,  
Deskripsi,  
Status\_Maintenance;

## Bab V User Interface Design

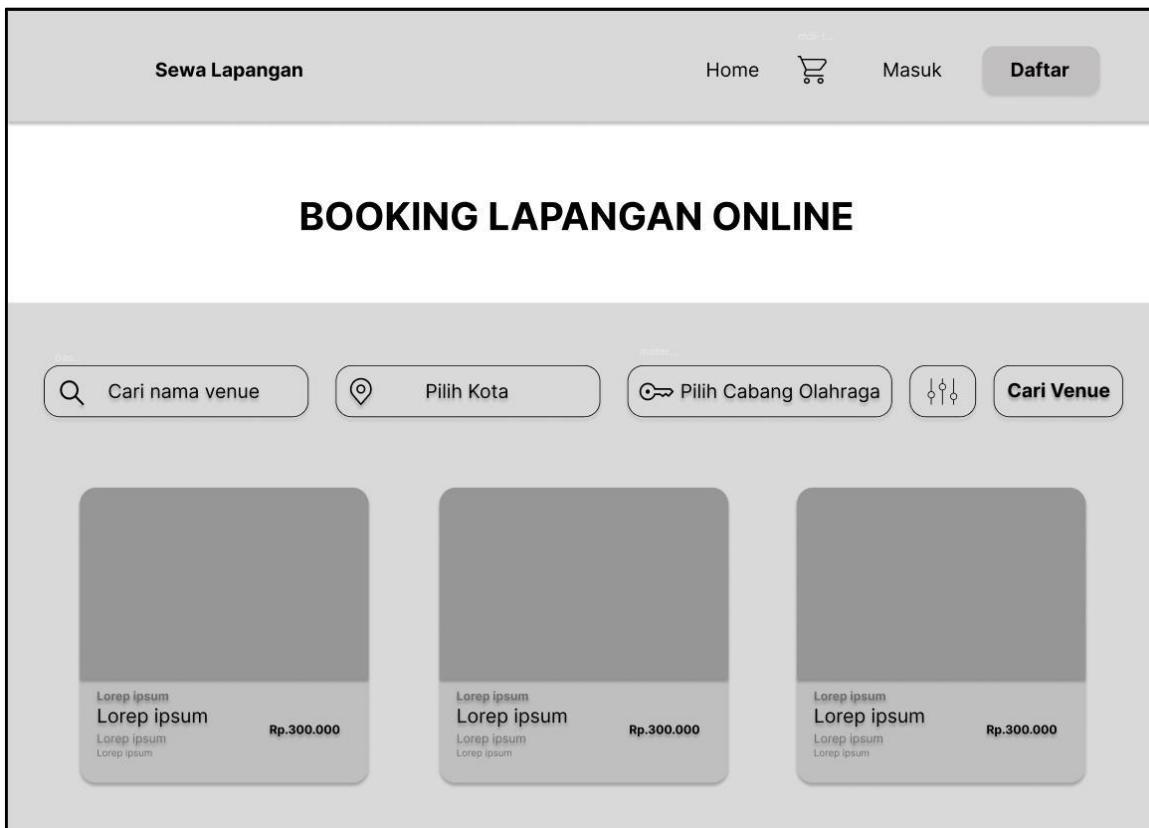
Antarmuka pengguna pada aplikasi perpustakaan ini dirancang dengan pendekatan minimalis untuk memastikan kemudahan operasional bagi seluruh pengguna. Proses pengembangannya mengadopsi prinsip-prinsip desain terstruktur, meliputi:

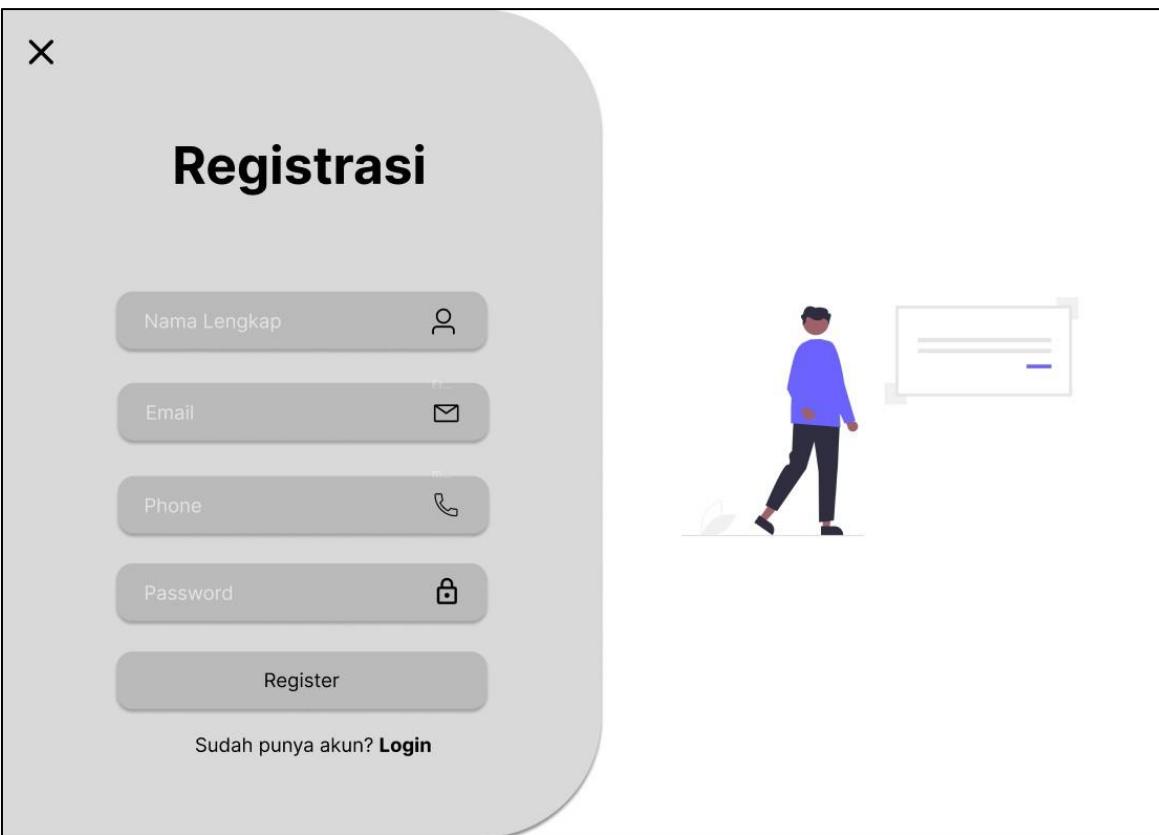
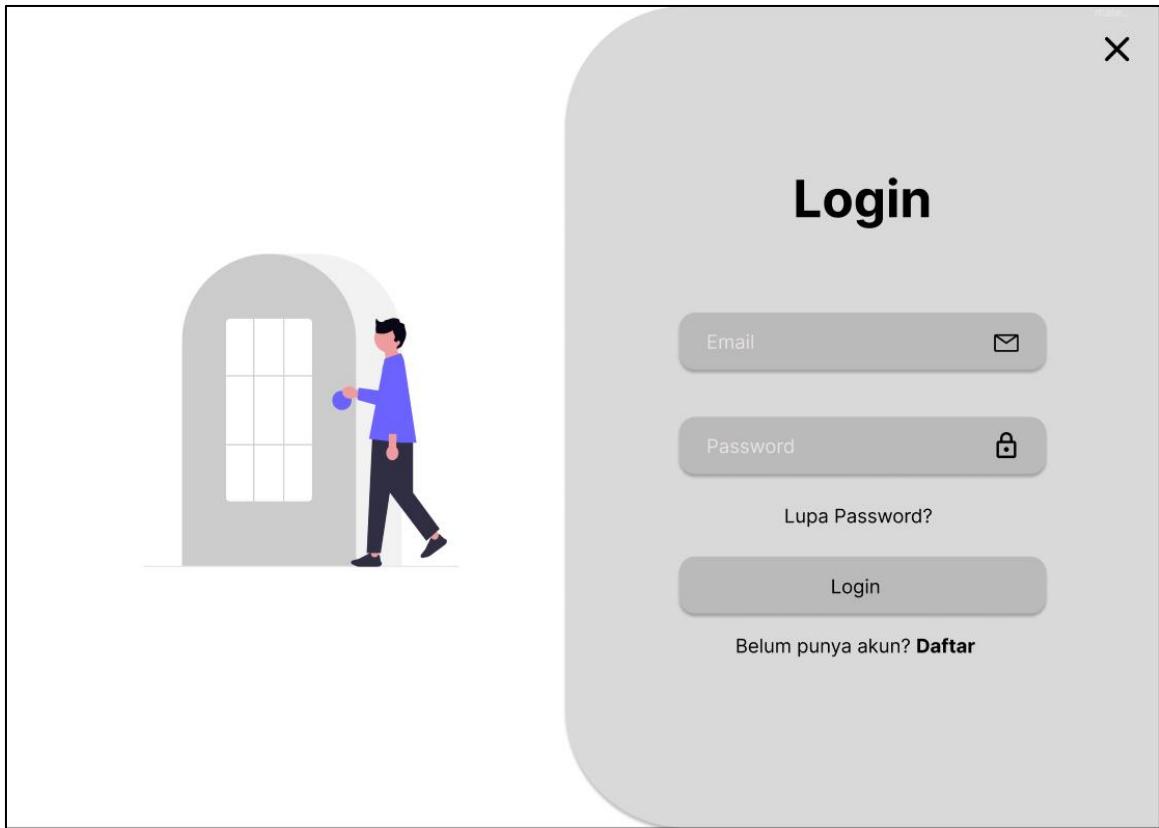
- Kesederhanaan (*Simplicity*): Menghilangkan elemen non-esensial untuk mengurangi beban kognitif,
- Efisiensi (*Efficiency*): Mengoptimalkan alur tugas dengan minimal interaksi,
- Konsistensi (*Consistency*): Menjaga pola navigasi dan tata letak antar modul,
- Fleksibilitas (*Flexibility*): Menyediakan jalur alternatif penyelesaian tugas sesuai preferensi pengguna.

Sebuah antarmuka yang diimplementasikan secara optimal berfungsi sebagai mekanisme pencegahan kesalahan (*error-proofing*), secara signifikan menurunkan risiko tindakan keliru pengguna melalui:

- *Constraint design* yang membatasi opsi invalid,
- *Feedback visual* real-time untuk konfirmasi aksi,
- Hierarki informasi terkuras yang memandu fokus pengguna.

Berikut diilustrasikan tampilan antarmuka Aplikasi Booking Lapangan Olahraga yang merefleksikan prinsip-prinsip tersebut:





## ← BOOKING LAPANGAN ONLINE

Nama Pemesan

No Telp

### Nama Lapangan

Rating Lapangan | Lokasi

Senin, 1 Juni 2025 - 08.00 - 09.00  
Rp.50.000



Senin, 1 Juni 2025 - 09.00 - 10.00  
Rp.50.000



Senin, 1 Juni 2025 - 10.00 - 11.00  
Rp.50.000



← Tambah Jadwal

### Rincian Biaya

Biaya Sewa Rp.150.000

Total Bayar Rp.150.000

Lanjutkan Pembayaran

Sewa Lapangan

Home

Masuk

Daftar

Nama Lapangan

Lokasi Lapangan | Lekatu

Lokasi Venue  
Jl. Lekatu Spesial

Buka Peta

Mulai dari  
**Rp 50,000** / Jam

Cek Ketersediaan

Pilih Jadwal

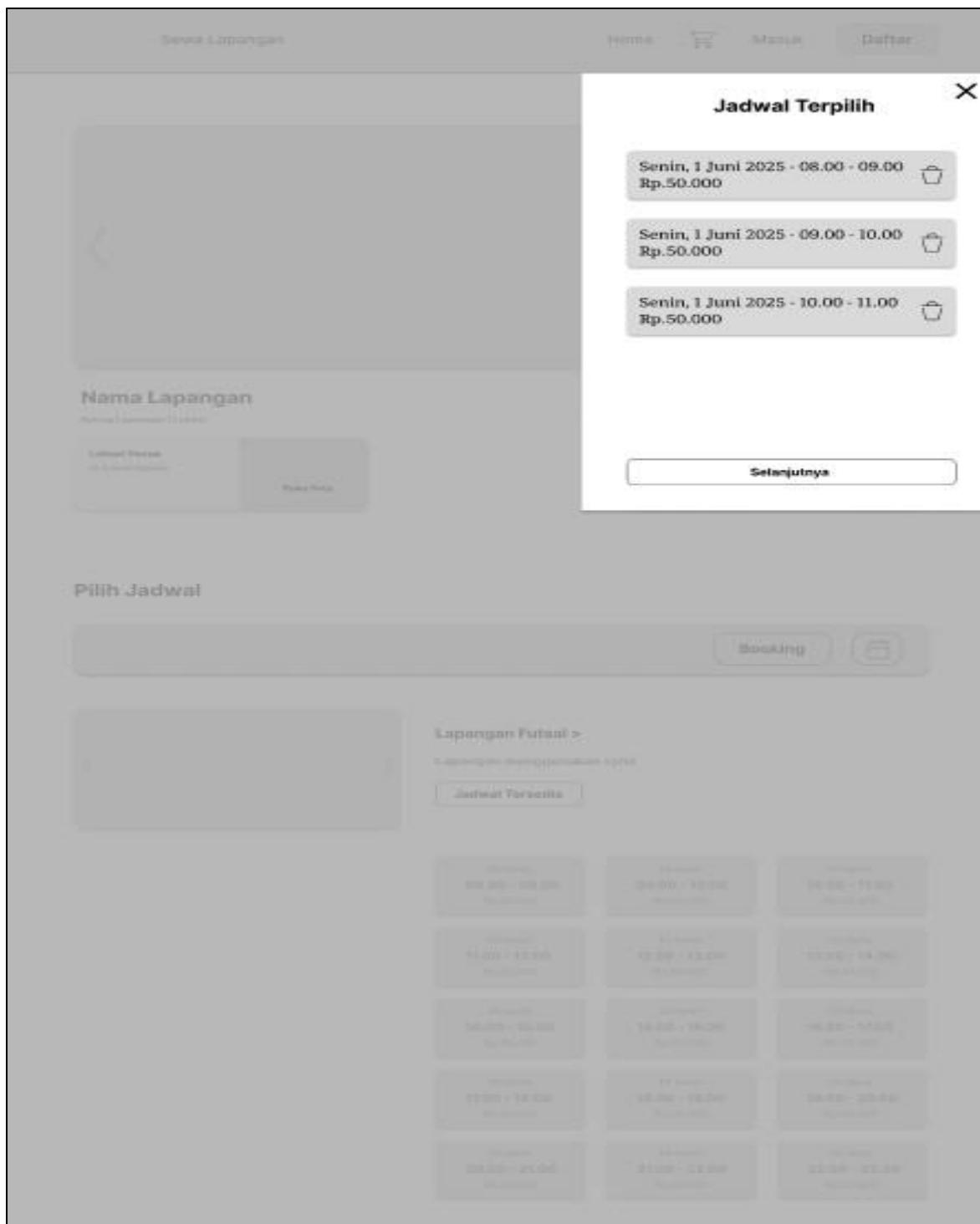
Booking

Lapangan Futsal >

Lapangan menggunakan vinyl

Jadwal Tersedia

00 Menit 08.00 - 09.00 Rp.50.000	00 Menit 09.00 - 10.00 Rp.50.000	00 Menit 10.00 - 11.00 Rp.50.000
00 Menit 11.00 - 12.00 Rp.50.000	00 Menit 12.00 - 13.00 Rp.50.000	00 Menit 13.00 - 14.00 Rp.50.000
00 Menit 14.00 - 15.00 Rp.50.000	00 Menit 15.00 - 16.00 Rp.50.000	00 Menit 16.00 - 17.00 Rp.50.000
00 Menit 17.00 - 18.00 Rp.50.000	00 Menit 18.00 - 19.00 Rp.50.000	00 Menit 19.00 - 20.00 Rp.50.000
00 Menit 20.00 - 21.00 Rp.50.000	00 Menit 21.00 - 22.00 Rp.50.000	00 Menit 22.00 - 23.00 Rp.50.000



BOOKING LAPANGAN ONLINE

☰ Home Contact 🔎 ⚙️

## Data Booking Lapangan

Tambah Data

Show 7 entries

No	Kode Booking	Nama Lapangan	Tanggal	Customer	No Telp	Durasi	Total	Status	Action
1	21313	Lapangan Basket	01-06-2025	Unjani FC	08999999	180	150.000	Reserved	☒ 🗑

## Bab VI Interface Requirements

### 6.1 User Interface

Antarmuka pengguna (UI) diimplementasikan sebagai Graphical User Interface (GUI) yang dioptimalkan untuk perangkat desktop. Desain UI pada aplikasi Booking Lapangan Olahraga memprioritaskan:

1. Pengalaman Pengguna (UX) Desktop: Antarmuka dirancang untuk interaksi menggunakan mouse dan keyboard, dengan tata letak yang responsif terhadap berbagai resolusi monitor. Navigasi utama memanfaatkan menu, tombol, dan tabel data yang memudahkan pengguna dalam melakukan pencarian jadwal, pemesanan, dan pengelolaan data.
2. Aksesibilitas Berbasis Peran: Tampilan dan fungsionalitas UI disesuaikan dengan peran pengguna (User/Penyewa, Petugas/Karyawan, Admin, dan Owner). Setiap peran hanya dapat mengakses menu dan fitur yang relevan, seperti pemesanan lapangan untuk User, pengelolaan jadwal untuk Petugas, serta pengelolaan master data dan laporan untuk Admin dan Owner.

### 6.2 Hardware Interface

Aplikasi Booking Lapangan Olahraga versi desktop dirancang agar dapat diakses dan berfungsi optimal pada komputer desktop maupun laptop.

1. Persyaratan Minimal Perangkat
  - o Komputer/laptop dengan arsitektur x64,
  - o Prosesor setara dual-core,
  - o RAM minimal 4 GB,
  - o Ruang penyimpanan yang memadai untuk pemasangan aplikasi dan penyimpanan data sementara.
2. Kebutuhan Perangkat Keras Pendukung
  - o Monitor dengan resolusi minimal 1366×768,
  - o Perangkat input seperti keyboard dan mouse,
  - o Akses ke jaringan lokal (LAN) atau koneksi internet untuk proses pemesanan, sinkronisasi data, dan pembayaran online (jika terintegrasi dengan payment gateway).

### 6.3 Software Interface

Bagian ini mendefinisikan antarmuka perangkat lunak yang diperlukan agar sistem aplikasi Booking Lapangan Olahraga versi desktop dapat beroperasi dan berinteraksi dengan komponen lain, baik internal maupun eksternal.

## 1. Sistem Operasi yang Kompatibel

Aplikasi memerlukan sistem operasi modern yang stabil dan didukung secara luas. Rekomendasi minimal:

- Microsoft Windows (minimal Windows 10),
- atau distribusi Linux (LTS seperti Ubuntu 22.04 atau setara) yang mendukung arsitektur x64.

## 2. Lingkungan Eksekusi dan Pendukung Aplikasi

Bergantung pada implementasi (misalnya aplikasi desktop berbasis web atau client–server), diperlukan:

- Web browser modern (seperti Chrome, Firefox, atau Edge versi stabil) jika aplikasi diakses melalui peramban,
- Web server / application server (misalnya Apache atau Nginx) jika aplikasi berbasis web client–server,
- Database Management System (DBMS) seperti MySQL/MariaDB atau PostgreSQL untuk pengelolaan data pemesanan, pengguna, dan lapangan,
- Library atau framework pendukung (misalnya framework backend dan frontend) yang digunakan sesuai rancangan arsitektur sistem.

## 3. Lingkungan Pengembangan (Development Environment)

Untuk proses pengembangan dan pemeliharaan:

- Integrated Development Environment (IDE) seperti Visual Studio Code, IntelliJ IDEA, atau sejenisnya dengan dukungan version control,
- Tool manajemen basis data untuk administrasi skema dan query,
- Sistem version control (misalnya Git) untuk pengelolaan kode sumber dan kolaborasi tim pengembang.

## 6.4 Communication Interface

Antarmuka komunikasi mendefinisikan protokol, standar, dan metodologi yang mengatur pertukaran data antar komponen sistem. Spesifikasi ini menjamin interoperabilitas sempurna antara klien (*user browser*), *server aplikasi*, dan *server basis data*, selaras dengan representasi dalam Diagram Deployment.

Komunikasi Klien-Server:

Interaksi antara perangkat pengguna dan *server aplikasi* mengadopsi protokol web standar dengan implementasi keamanan ketat:

- Protokol Transmisi: HTTP (*Hypertext Transfer Protocol*) digunakan sebagai fondasi, dengan enkripsi wajib melalui lapisan HTTPS (*HTTP Secure*).
- Mekanisme Keamanan: Implementasi TLS (*Transport Layer Security*) versi 1.3 atau lebih baru bersifat imperatif guna mengenkripsi seluruh *payload* data. Proteksi ini secara khusus melindungi kredensial otentikasi, data pribadi mahasiswa, dan catatan akademik dari risiko *eavesdropping* dan serangan *man-in-the-middle*.

Antarmuka Pemrograman Aplikasi (API) dan Format Data:

Komunikasi internal antara *front-end* (berjalan di lingkungan klien) dan *back-end* (*server aplikasi*) diatur melalui kontrak API terstandarisasi:

- Arsitektur API: Diterapkan gaya arsitektur RESTful (*Representational State Transfer*) dengan prinsip *stateless*. Operasi CRUD (*Create, Read, Update, Delete*) dieksekusi melalui metode HTTP native (GET/POST/PUT/DELETE) pada sumber daya yang diidentifikasi URI (contoh: /api/mahasiswa, /api/proposal/{id}).
- Format Pertukaran Data: JSON (*JavaScript Object Notation*) ditetapkan sebagai format baku *request/response* karena efisiensi *parsing*, *human-readable*, dan kompatibilitas optimal dengan ekosistem JavaScript.

Antarmuka Basis Data:

Konektivitas antara *server aplikasi* dan DBMS (MySQL) mengikuti standar industri:

- Protokol Koneksi: Digunakan *driver* spesifik bahasa pemrograman:
  - JDBC (*Java Database Connectivity*) untuk arsitektur Java
  - Konektor native (contoh: mysql-connector-python, PDO\_MySQL) untuk Python/PHP
- Mekanisme Keamanan:
  - Komunikasi melalui TCP/IP dalam jaringan privat terisolasi
  - Autentikasi berbasis *credentials* dengan *role-based access control*
  - Enkripsi *in-transit* menggunakan TLS untuk koneksi eksternal
  - *Firewall* berbasis whitelist yang membatasi akses hanya dari *server aplikasi*