

Nama : Syifa Amilatun Nadiyah

NIM : 12030123120015

Mata Kuliah : Analisis dan Desain Sistem

Kelas : C

Dosen : Dr. Totok Dewayanto, S.E.,M.Si.,Akt.

UJIAN TENGAH SEMESTER ANALISIS DAN DESAIN SISTEM

Kasus : Sistem Pemesanan Tiket Bioskop

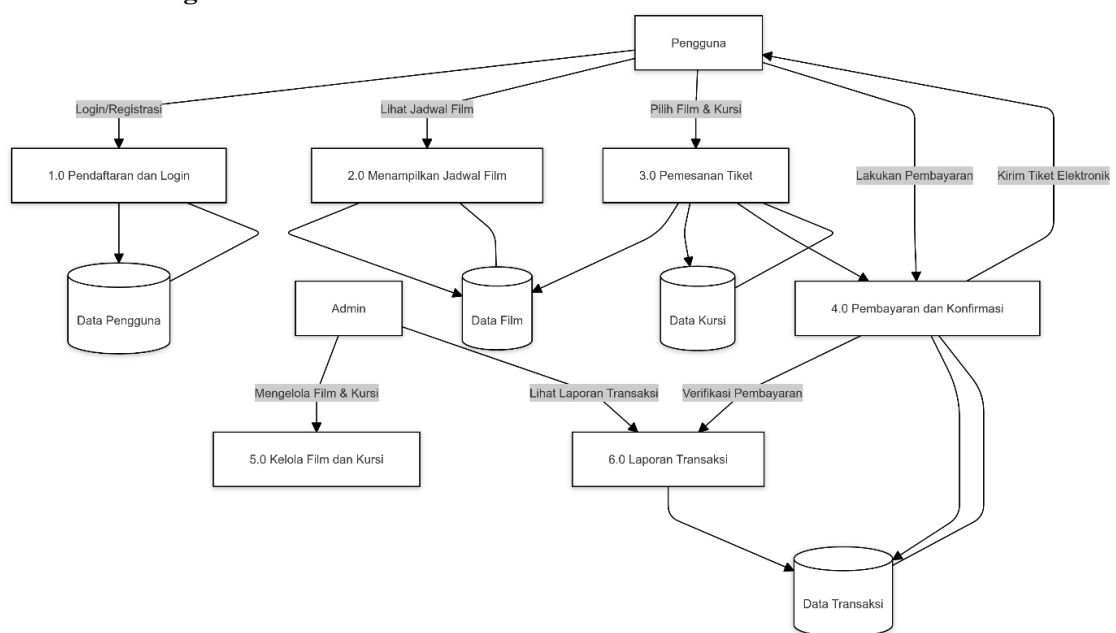
Deskripsi Kasus : Sebuah bioskop ingin mengembangkan sistem pemesanan tiket secara online.

Pengguna dapat melihat jadwal film, memilih film yang ingin ditonton, memilih kursi yang tersedia, melakukan pembayaran, dan mendapatkan tiket elektronik. Sistem juga mengelola data film, data kursi, data transaksi, dan laporan penjualan.

Kebutuhan Fungsional:

1. Pengguna dapat mendaftar dan login ke dalam sistem.
2. Pengguna dapat melihat jadwal film yang tersedia.
3. Pengguna dapat memilih film dan kursi sesuai jadwal.
4. Pengguna dapat melakukan pembayaran secara online.
5. Sistem mengirimkan tiket elektronik kepada pengguna.
6. Admin dapat mengelola data film, kursi, dan laporan transaksi.

1. Data Flow Diagram Kelas 1



Kode Mermaid :

graph TD

%% External Entities

A[Pengguna] --> |Login/Registrasi| P1[1.0 Pendaftaran dan Login]

A --> |Lihat Jadwal Film| P2[2.0 Menampilkan Jadwal Film]

A --> |Pilih Film & Kursi| P3[3.0 Pemesanan Tiket]

A --> |Lakukan Pembayaran| P4[4.0 Pembayaran dan Konfirmasi]

B[Admin] --> |Mengelola Film & Kursi| P5[5.0 Kelola Film dan Kursi]

%% Processes

P1 --> D1[(Data Pengguna)]

P2 --> D2[(Data Film)]

P3 --> D2

P3 --> D3[(Data Kursi)]

P3 --> P4

P4 --> |Verifikasi Pembayaran| P6[Payment Gateway]

P4 --> D4[(Data Transaksi)]

P4 --> |Kirim Tiket Elektronik| A

%% Admin Process

B --> |Lihat Laporan Transaksi| P6[6.0 Laporan Transaksi]

P6 --> D4

%% Data Stores

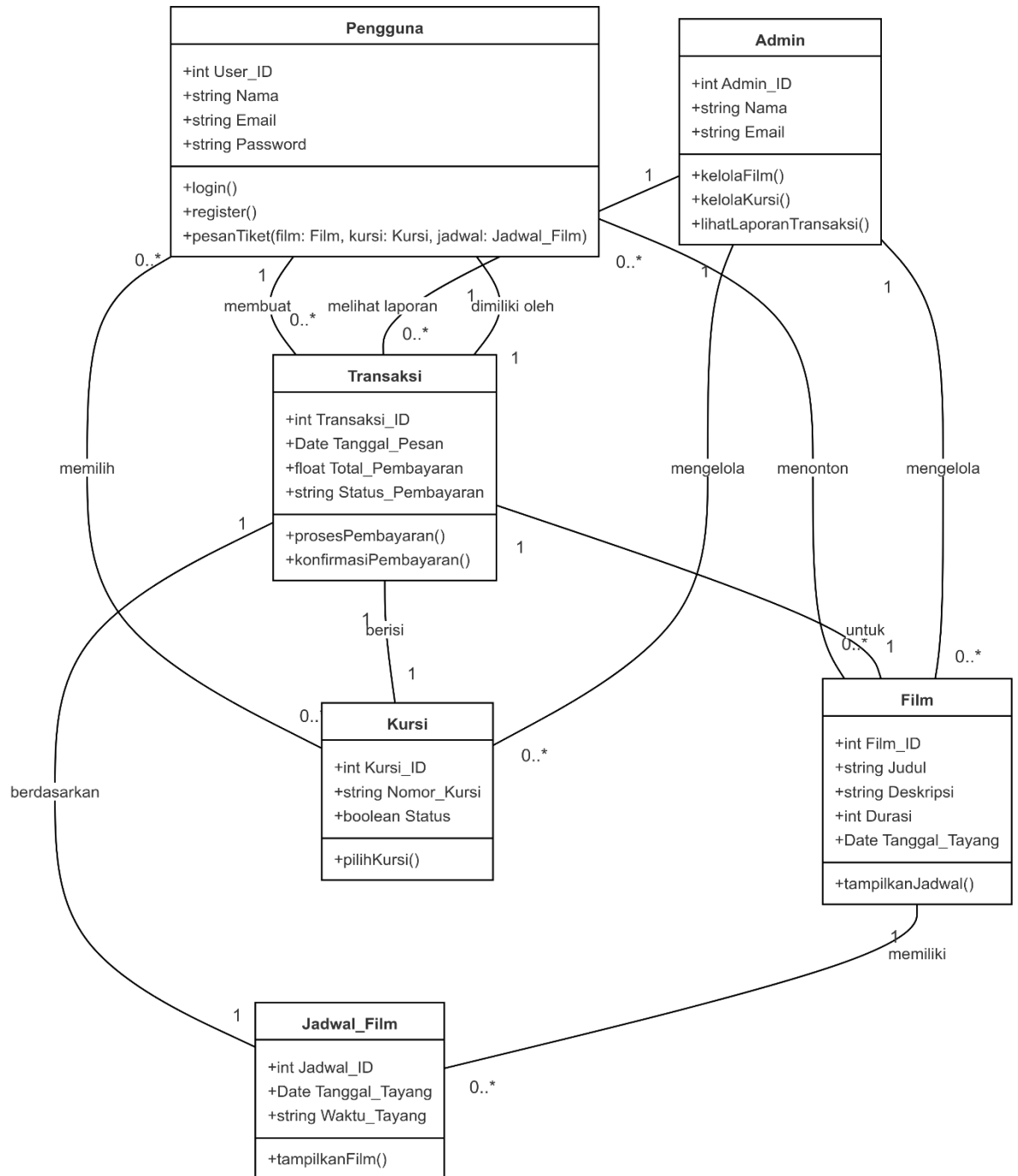
D1 --- P1

D2 --- P2

D3 --- P3

D4 --- P4

2. Class Diagram



Kode Mermaid :

```

classDiagram
class Pengguna {
  +int User_ID
  +string Nama
  +string Email
  +string Password
  +login()
  +register()
  +pesanTiket(film: Film, kursi: Kursi, jadwal: Jadwal_Film)
}

```

```
    +pesanTiket(film: Film, kursi: Kursi, jadwal: Jadwal_Film)
}
```

```
class Film {
    +int Film_ID
    +string Judul
    +string Deskripsi
    +int Durasi
    +Date Tanggal_Tayang
    +tampilkanJadwal()
}
```

```
class Kursi {
    +int Kursi_ID
    +string Nomor_Kursi
    +boolean Status
    +pilihKursi()
}
```

```
class Transaksi {
    +int Transaksi_ID
    +Date Tanggal_Pesan
    +float Total_Pembayaran
    +string Status_Pembayaran
    +prosesPembayaran()
    +konfirmasiPembayaran()
}
```

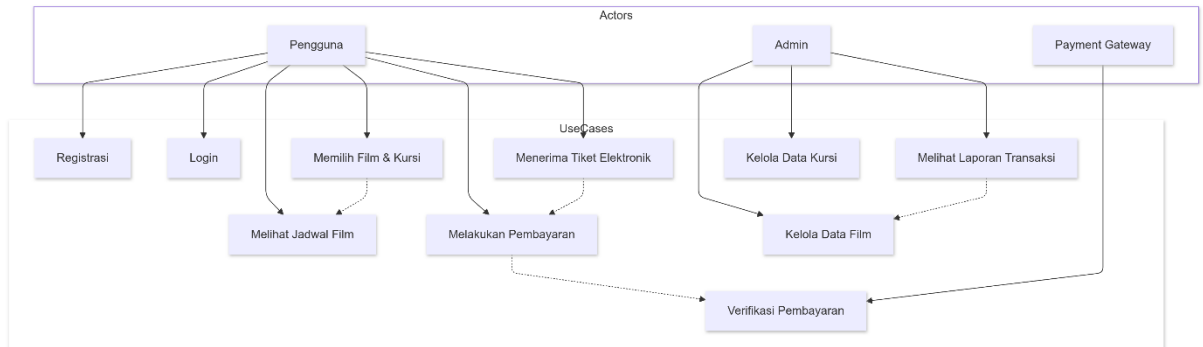
```
class Jadwal_Film {
    +int Jadwal_ID
    +Date Tanggal_Tayang
    +string Waktu_Tayang
    +tampilkanFilm()
}
```

```
class Admin {
    +int Admin_ID
    +string Nama
    +string Email
    +kelolaFilm()
    +kelolaKursi()
    +lihatLaporanTransaksi()
}
```

```
%% Relasi antar kelas
Pegguna "1" -- "0..*" Transaksi : membuat
Pegguna "0..*" -- "0..*" Kursi : memilih
Pegguna "0..*" -- "0..*" Film : menonton
Transaksi "1" -- "1" Pegguna : dimiliki oleh
Transaksi "1" -- "1" Film : untuk
Transaksi "1" -- "1" Kursi : berisi
```

Transaksi "1" -- "1" Jadwal_Film : berdasarkan
 Film "1" -- "0..*" Jadwal_Film : memiliki
 Admin "1" -- "0..*" Film : mengelola
 Admin "1" -- "0..*" Kursi : mengelola
 Admin "1" -- "0..*" Transaksi : melihat laporan

3. Use Case Diagram



Kode Mermaid :

%% Use Case Diagram for Sistem Pemesanan Tiket Bioskop

%% Definisi aktor dan use case yang sempurna

actor Pengguna as "Pengguna"

actor Admin as "Admin"

actor PaymentGateway as "Payment Gateway"

%% Use Case untuk Pengguna

Pengguna --> (Registrasi)

Pengguna --> (Login)

Pengguna --> (Melihat Jadwal Film)

Pengguna --> (Memilih Film & Kursi)

Pengguna --> (Melakukan Pembayaran)

Pengguna --> (Menerima Tiket Elektronik)

%% Use Case untuk Admin

Admin --> (Kelola Data Film)

Admin --> (Kelola Data Kursi)

Admin --> (Melihat Laporan Transaksi)

%% Use Case untuk Payment Gateway

PaymentGateway --> (Verifikasi Pembayaran)

%% Relasi antar Use Case dengan include dan extend

(Memilih Film & Kursi) --|> (Melihat Jadwal Film) : <<include>>

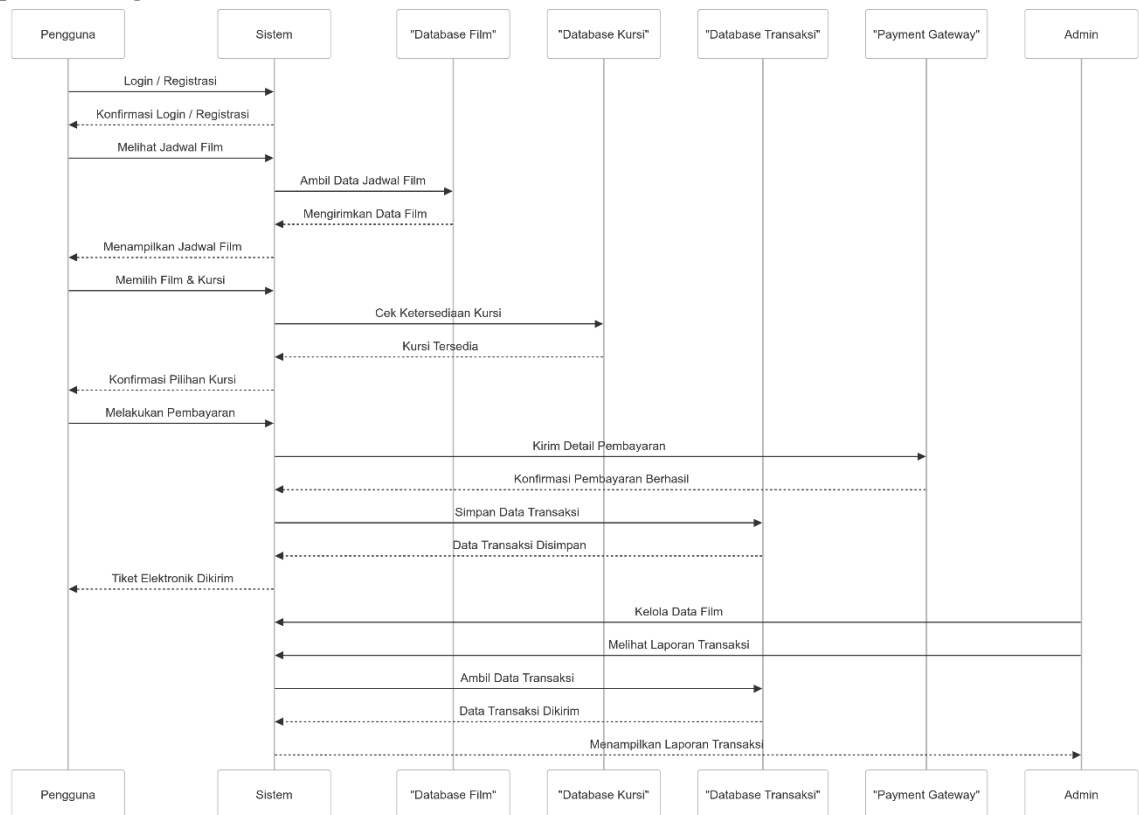
(Melakukan Pembayaran) --|> (Verifikasi Pembayaran) : <<include>>

(Menerima Tiket Elektronik) --|> (Melakukan Pembayaran) : <<include>>

%% Admin melihat laporan transaksi yang tergantung pada pengelolaan transaksi

(Melihat Laporan Transaksi) --|> (Kelola Data Film) : <<include>>

4. Sequence Diagram



Kode Mermaid :

sequenceDiagram

participant Pengguna

participant Sistem

participant FilmDatabase as "Database Film"

participant KursiDatabase as "Database Kursi"

participant TransaksiDatabase as "Database Transaksi"

participant PaymentGateway as "Payment Gateway"

%% Alur dimulai ketika pengguna login atau registrasi

Pengguna ->> Sistem: Login / Registrasi

Sistem -->> Pengguna: Konfirmasi Login / Registrasi

%% Pengguna melihat jadwal film

Pengguna ->> Sistem: Melihat Jadwal Film

Sistem ->> FilmDatabase: Ambil Data Jadwal Film

FilmDatabase -->> Sistem: Mengirimkan Data Film

Sistem -->> Pengguna: Menampilkan Jadwal Film

%% Pengguna memilih film dan kursi

Pengguna ->> Sistem: Memilih Film & Kursi

Sistem ->> KursiDatabase: Cek Ketersediaan Kursi

KursiDatabase -->> Sistem: Kursi Tersedia

Sistem -->> Pengguna: Konfirmasi Pilihan Kursi

%% Pengguna melakukan pembayaran

Pengguna ->> Sistem: Melakukan Pembayaran

Sistem ->> PaymentGateway: Kirim Detail Pembayaran

PaymentGateway -->> Sistem: Konfirmasi Pembayaran Berhasil

Sistem ->> TransaksiDatabase: Simpan Data Transaksi

TransaksiDatabase -->> Sistem: Data Transaksi Disimpan

Sistem -->> Pengguna: Tiket Elektronik Dikirim

%% Admin mengelola data film dan melihat laporan transaksi

participant Admin

Admin ->> Sistem: Kelola Data Film

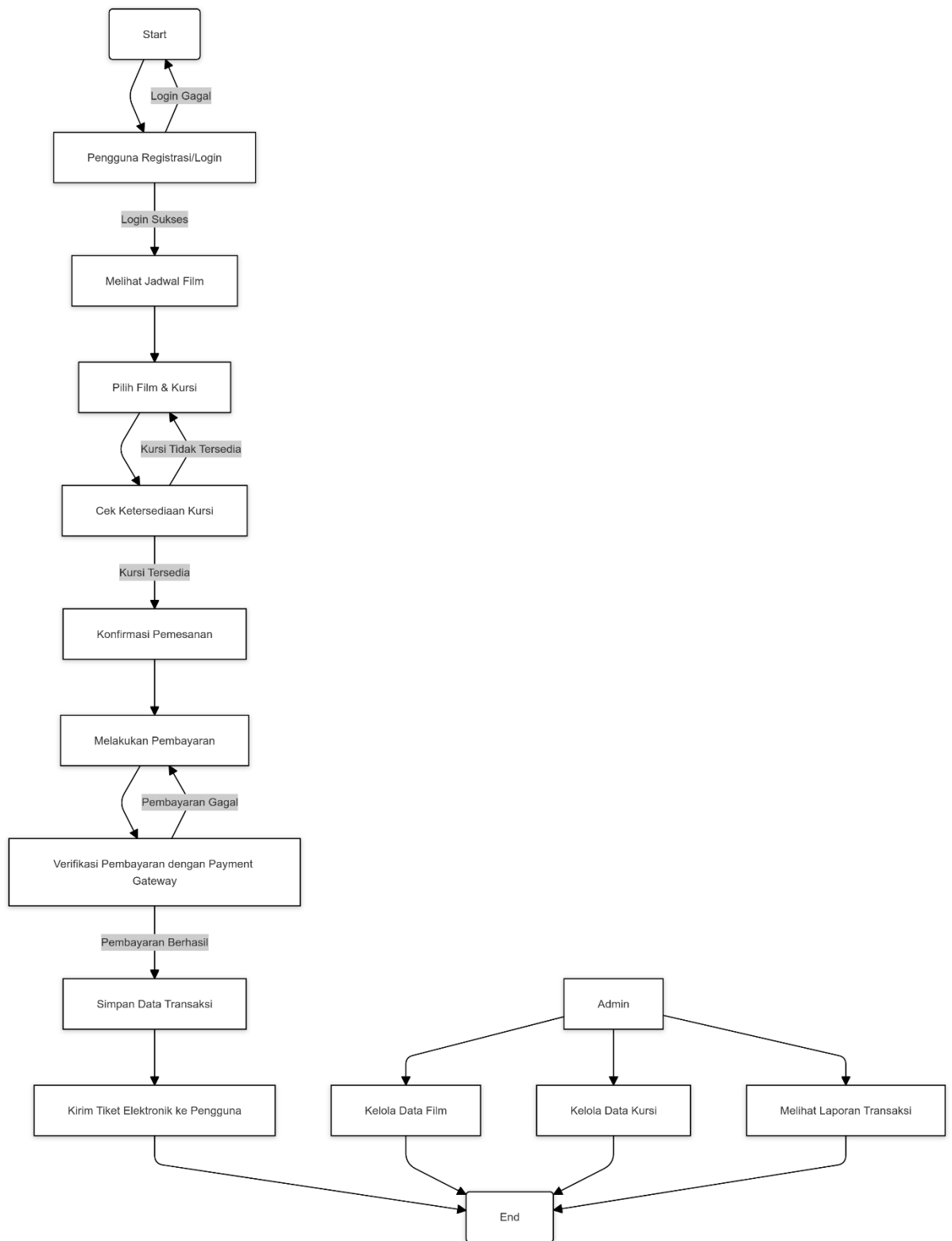
Admin ->> Sistem: Melihat Laporan Transaksi

Sistem ->> TransaksiDatabase: Ambil Data Transaksi

TransaksiDatabase -->> Sistem: Data Transaksi Dikirim

Sistem -->> Admin: Menampilkan Laporan Transaksi

5. Activity Diagram



Kode Mermaid :

%% Activity Diagram for Sistem Pemesanan Tiket Bioskop

%% Definisi aktivitas dan alur

flowchart TD

%% Mulai dan terminasi

A(Start) --> B[Pengguna Registrasi/Login]

%% Login sukses atau gagal

B --> |Login Sukses| C[Melihat Jadwal Film]

B --> |Login Gagal| A

%% Proses melihat jadwal dan memilih film & kursi

C --> D[Pilih Film & Kursi]

D --> E[Cek Ketersediaan Kursi]

E --> |Kursi Tersedia| F[Konfirmasi Pemesanan]

E --> |Kursi Tidak Tersedia| D

%% Pembayaran setelah kursi tersedia

F --> G[Melakukan Pembayaran]

G --> H[Verifikasi Pembayaran dengan Payment Gateway]

H --> |Pembayaran Berhasil| I[Simpan Data Transaksi]

H --> |Pembayaran Gagal| G

%% Setelah pembayaran berhasil

I --> J[Kirim Tiket Elektronik ke Pengguna]

J --> K(End)

%% Admin mengelola film dan kursi

Admin --> L[Kelola Data Film]

Admin --> M[Kelola Data Kursi]

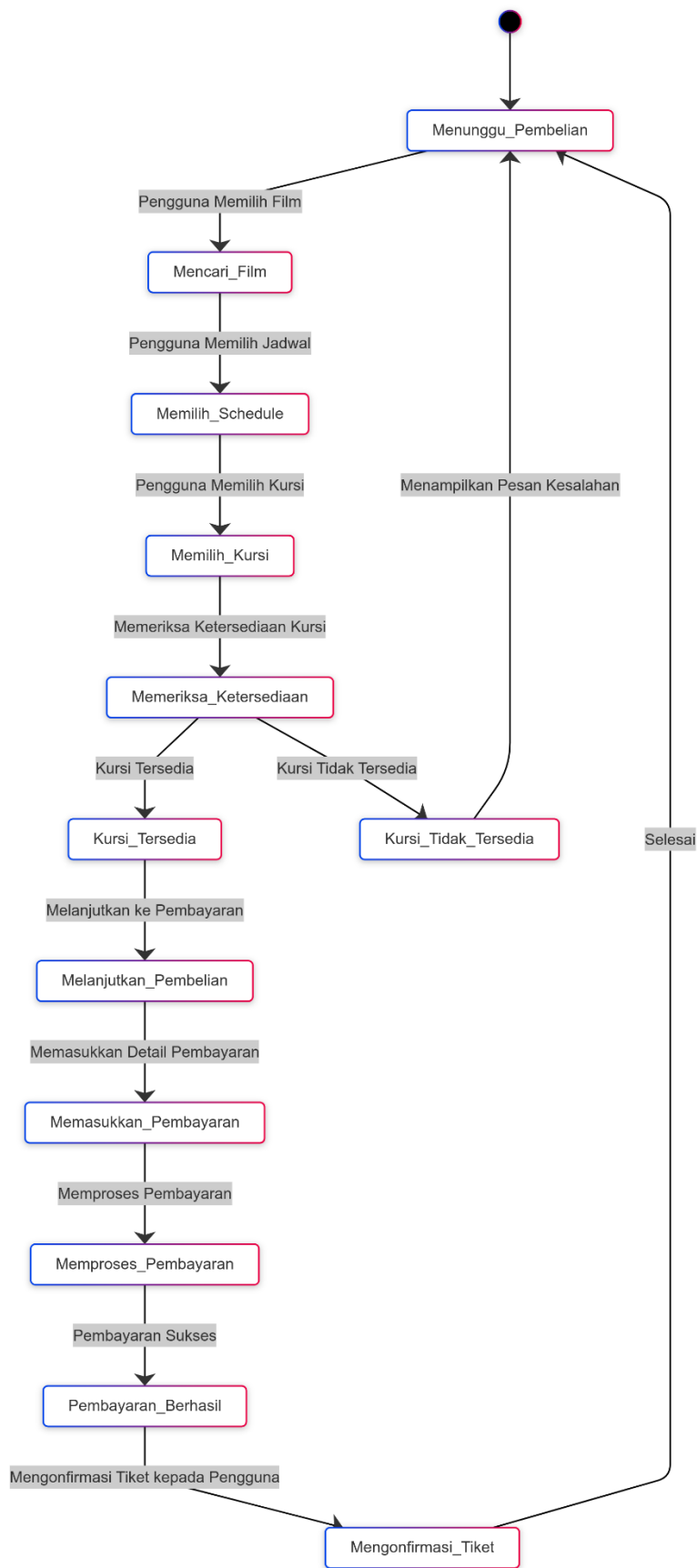
Admin --> N[Melihat Laporan Transaksi]

L --> K

M --> K

N --> K

6. State Diagram



Kode Mermaid :

%% State Diagram for Cinema Ticket Purchase System

%% Showing the state transitions for purchasing tickets.

stateDiagram-v2

[*] --> Menunggu_Pembelian

Menunggu_Pembelian --> Mencari_Film: Pengguna Memilih Film

Mencari_Film --> Memilih_Schedule: Pengguna Memilih Jadwal

Memilih_Schedule --> Memilih_Kursi: Pengguna Memilih Kursi

Memilih_Kursi --> Memeriksa_Ketersediaan: Memeriksa Ketersediaan Kursi

Memeriksa_Ketersediaan --> Kursi_Tersedia: Kursi Tersedia

Kursi_Tersedia --> Melanjutkan_Pembelian: Melanjutkan ke Pembayaran

Memeriksa_Ketersediaan --> Kursi_Tidak_Tersedia: Kursi Tidak Tersedia

Kursi_Tidak_Tersedia --> Menunggu_Pembelian: Menampilkan Pesan Kesalahan

Melanjutkan_Pembelian --> Memasukkan_Pembayaran: Memasukkan Detail Pembayaran

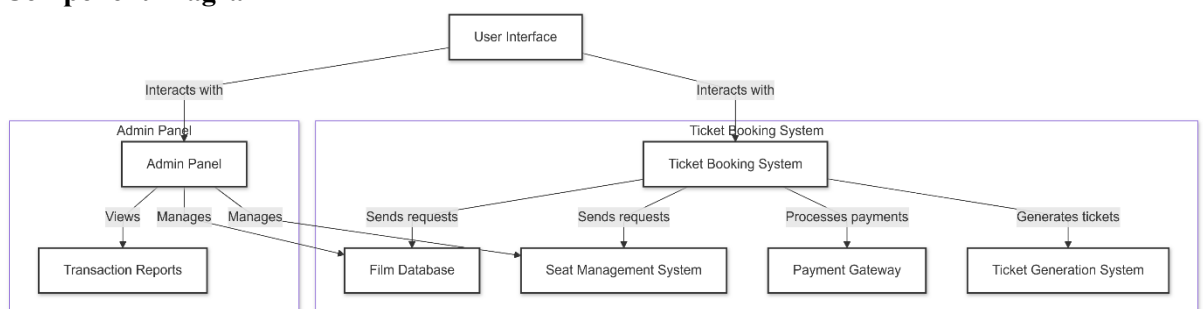
Memasukkan_Pembayaran --> Memproses_Pembayaran: Memproses Pembayaran

Memproses_Pembayaran --> Pembayaran_Berhasil: Pembayaran Sukses

Pembayaran_Berhasil --> Mengonfirmasi_Tiket: Mengonfirmasi Tiket kepada Pengguna

Mengonfirmasi_Tiket --> Menunggu_Pembelian: Selesai

7. Component Diagram



Kode Mermaid :

%%{ init : { "theme" : "default" } }%%

graph TD

A[User Interface] -->|Interacts with| B[Ticket Booking System]

A -->|Interacts with| C[Admin Panel]

subgraph Ticket Booking System

B -->|Sends requests| D[Film Database]

B -->|Sends requests| E[Seat Management System]

B -->|Processes payments| F[Payment Gateway]

B -->|Generates tickets| G[Ticket Generation System]

end

subgraph Admin Panel

C -->|Manages| D

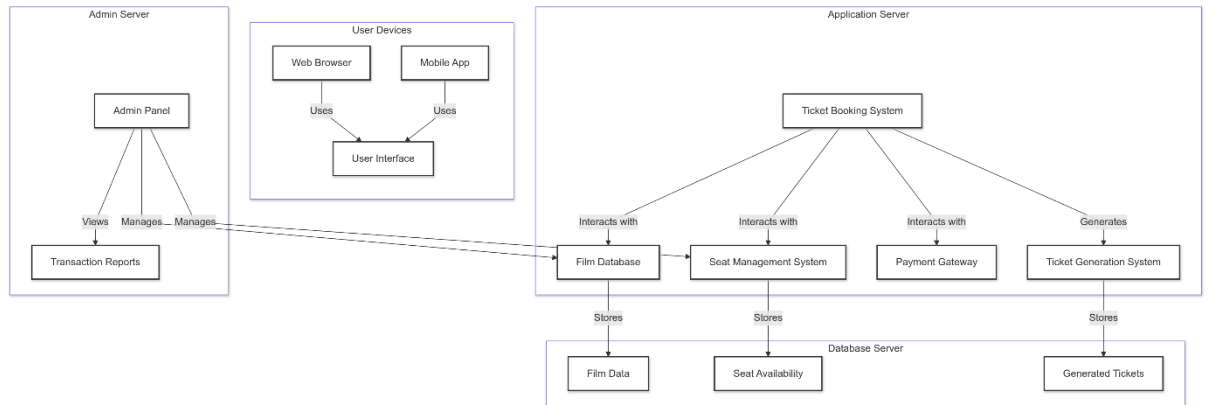
C -->|Manages| E

C -->|Views| H[Transaction Reports]

end

style A fill:#f9f,stroke:#333,stroke-width:2px;
 style B fill:#ccf,stroke:#333,stroke-width:2px;
 style C fill:#ccf,stroke:#333,stroke-width:2px;
 style D fill:#cff,stroke:#333,stroke-width:2px;
 style E fill:#cff,stroke:#333,stroke-width:2px;
 style F fill:#cff,stroke:#333,stroke-width:2px;
 style G fill:#cff,stroke:#333,stroke-width:2px;
 style H fill:#cff,stroke:#333,stroke-width:2px;

8. Deployment Diagram



Code Mermaid :

```
%%{ init : { "theme" : "default" } }%%
```

```
graph TD
```

```
  subgraph User Devices
```

```
    direction TB
```

```
    A[Web Browser] -->|Uses| B[User Interface]
```

```
    C[Mobile App] -->|Uses| B
```

```
  end
```

```
  subgraph Application Server
```

```
    direction TB
```

```
    D[Ticket Booking System] -->|Interacts with| E[Film Database]
```

```
    D -->|Interacts with| F[Seat Management System]
```

```
    D -->|Interacts with| G[Payment Gateway]
```

```
    D -->|Generates| H[Ticket Generation System]
```

```
  end
```

```
  subgraph Admin Server
```

```
    direction TB
```

```
    I[Admin Panel] -->|Manages| E
```

```
    I -->|Manages| F
```

```
    I -->|Views| J[Transaction Reports]
```

```
  end
```

```
  subgraph Database Server
```

```
    direction TB
```

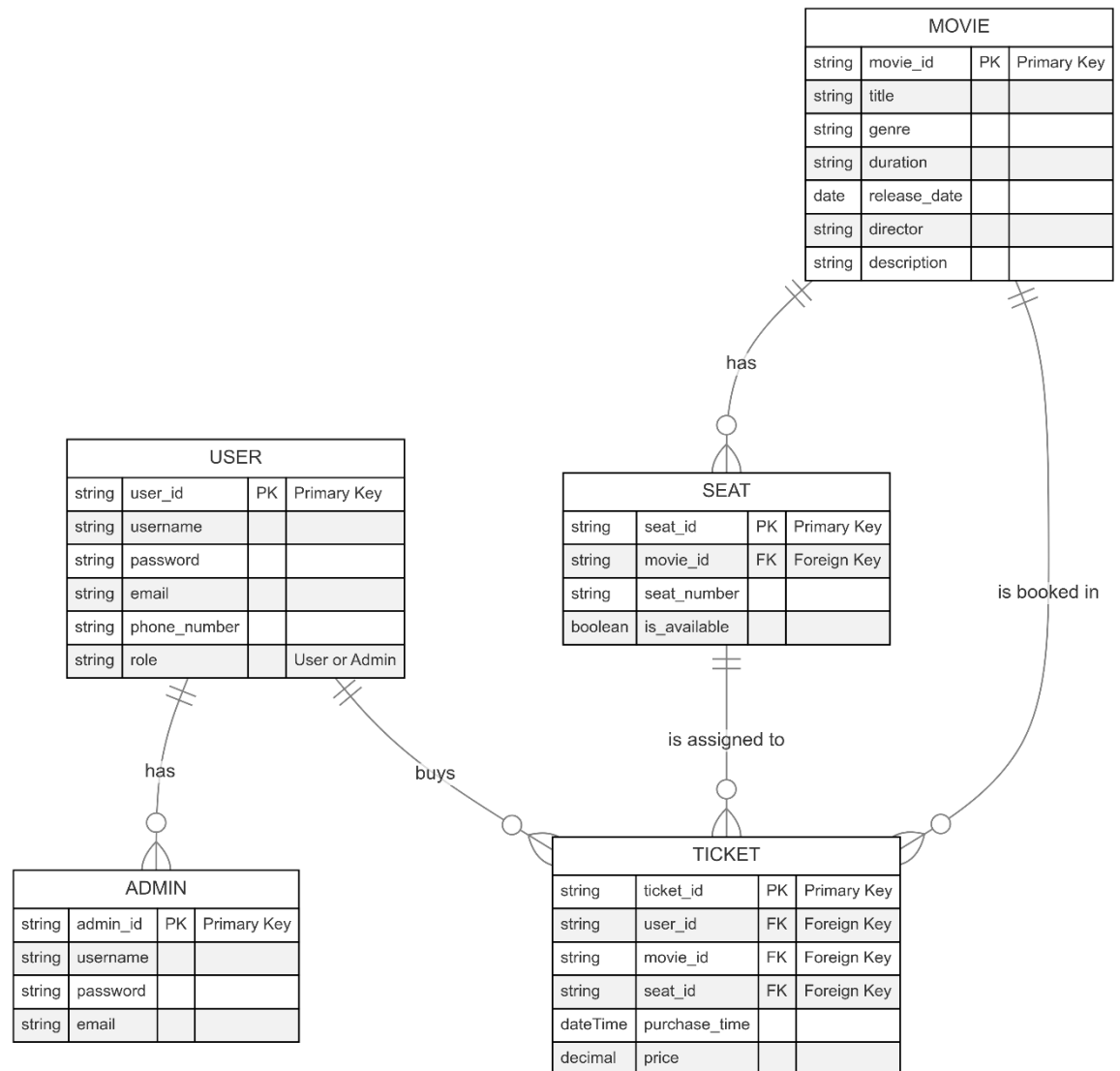
```
    E -->|Stores| K[Film Data]
```

```
    F -->|Stores| L[Seat Availability]
```

H -->|Stores| M[Generated Tickets]
end

style A fill:#f9f,stroke:#333,stroke-width:2px;
style B fill:#ccf,stroke:#333,stroke-width:2px;
style C fill:#f9f,stroke:#333,stroke-width:2px;
style D fill:#ccf,stroke:#333,stroke-width:2px;
style E fill:#cff,stroke:#333,stroke-width:2px;
style F fill:#cff,stroke:#333,stroke-width:2px;
style G fill:#cff,stroke:#333,stroke-width:2px;
style H fill:#cff,stroke:#333,stroke-width:2px;
style I fill:#ccf,stroke:#333,stroke-width:2px;
style J fill:#cff,stroke:#333,stroke-width:2px;
style K fill:#cff,stroke:#333,stroke-width:2px;
style L fill:#cff,stroke:#333,stroke-width:2px;
style M fill:#cff,stroke:#333,stroke-width:2px;

9. Entity Relationship Diagram Plus



Kode Mermaid :

erDiagram

```
USER {
    string user_id PK "Primary Key"
    string username
    string password
    string email
    string phone_number
    string role "User or Admin"
}
```

```
MOVIE {
    string movie_id PK "Primary Key"
    string title
    string genre
    string duration
    date release_date
    string director
    string description
}
```

```
SEAT {
    string seat_id PK "Primary Key"
    string movie_id FK "Foreign Key"
    string seat_number
    boolean is_available
}
```

```
TICKET {
    string ticket_id PK "Primary Key"
    string user_id FK "Foreign Key"
    string movie_id FK "Foreign Key"
    string seat_id FK "Foreign Key"
    dateTime purchase_time
    decimal price
}
```

```
ADMIN {
    string admin_id PK "Primary Key"
    string username
    string password
    string email
}
```

```
USER ||--o{ TICKET : "buys"
MOVIE ||--o{ TICKET : "is booked in"
SEAT ||--o{ TICKET : "is assigned to"
USER ||--o{ ADMIN : "has"
MOVIE ||--o{ SEAT : "has"
```