

**DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

**<Nama Perangkat Lunak>**

untuk:

Memenuhi tugas mata kuliah Analisis Perancangan Perangkat Lunak

Dipersiapkan oleh:

Christian Prasetyo - 1301180398

Hauzan Jiyad D. K - 1301180212

M. Thaariq Razaq - 1301184073

Pieter Edward R. - 1301184479

Program Studi Informatika

Fakultas Informatika

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Prodi S1- Informatika**  **Universitas Telkom** | **Nomor Dokumen** | | **Halaman** |
| ***DPPL-xx <xx:no grp>*** | | *<#>/<jml #* |
| **Revisi** | *<nomor revisi>* | *Tgl: <isi tanggal>* |

**DAFTAR PERUBAHAN**

|  |  |
| --- | --- |
| **Revisi** | **Deskripsi** |
| **A** |  |
| **B** |  |
| **C** |  |
| **D** |  |
| **E** |  |
| **F** |  |
| **G** |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| INDEX  TGL | - | A | B | C | D | E | F | G |
| Ditulis oleh |  |  |  |  |  |  |  |  |
| Diperiksa oleh |  |  |  |  |  |  |  |  |
| Disetujui oleh |  |  |  |  |  |  |  |  |

**Daftar Halaman Perubahan**

|  |  |  |  |
| --- | --- | --- | --- |
| **Halaman** | **Revisi** | **Halaman** | **Revisi** |
|  |  |  |  |

**Daftar Isi**

[**1. Pendahuluan**](#_heading=h.gjdgxs) **5**

[Tujuan Penulisan Dokumen](#_heading=h.30j0zll) [6](#_heading=h.30j0zll)

[Lingkup Masalah](#_heading=h.1fob9te) [6](#_heading=h.1fob9te)

[Definisi dan Istilah](#_heading=h.3znysh7) [6](#_heading=h.3znysh7)

[Referensi](#_heading=h.2et92p0) [6](#_heading=h.2et92p0)

[Sistematika Pembahasan](#_heading=h.tyjcwt) [6](#_heading=h.tyjcwt)

[**Deskripsi Perancangan Global**](#_heading=h.3dy6vkm) **6**

[Rancangan Lingkungan Implementasi](#_heading=h.1t3h5sf) [7](#_heading=h.1t3h5sf)

[Deskripsi Arsitektural](#_heading=h.4d34og8) [7](#_heading=h.4d34og8)

[Deskripsi Komponen](#_heading=h.2s8eyo1) [7](#_heading=h.2s8eyo1)

[**Perancangan Rinci**](#_heading=h.3rdcrjn) **8**

[Realisasi Use Case](#_heading=h.26in1rg) [8](#_heading=h.26in1rg)

[Use Case <nama use case 1>](#_heading=h.lnxbz9) [8](#_heading=h.lnxbz9)

[Identifikasi Kelas](#_heading=h.35nkun2) [8](#_heading=h.35nkun2)

[Sequence Diagram](#_heading=h.1ksv4uv) [8](#_heading=h.1ksv4uv)

[Diagram Kelas](#_heading=h.z337ya) [8](#_heading=h.z337ya)

[Perancangan Detil Kelas](#_heading=h.3j2qqm3) [8](#_heading=h.3j2qqm3)

[Kelas <nama kelas>](#_heading=h.1y810tw) [8](#_heading=h.1y810tw)

[Kelas <nama kelas>](#_heading=h.4i7ojhp) [9](#_heading=h.4i7ojhp)

[Diagram Kelas Keseluruhan](#_heading=h.2xcytpi) [9](#_heading=h.2xcytpi)

[Algoritma/Query](#_heading=h.1ci93xb) [9](#_heading=h.1ci93xb)

[Diagram Statechart](#_heading=h.1pxezwc) [9](#_heading=h.1pxezwc)

[Perancangan Antarmuka](#_heading=h.3whwml4) [9](#_heading=h.3whwml4)

[Perancangan Representasi Persistensi Kelas](#_heading=h.2bn6wsx) [10](#_heading=h.2bn6wsx)

[**Matriks Kerunutan**](#_heading=h.qsh70q) **10**

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

**1. Pendahuluan**

* 1. ***Tujuan Penulisan Dokumen***

Dokumen ini dibuat untuk menjelaskan informasi, fungsi-fungsi kepada pengembang dan pengguna. Bagi pengembang berfungsi untuk menjadi acuan dalam pengembangan, sedangkan bagi pengguna, dokumen ini berguna untuk mengetahui informasi dan fungsi-fungsi yang terdapat dalam sistem aplikasi dan spesifikasi yang dibutuhkan.

* 1. ***Lingkup Masalah***

Dokumen ini diprioritaskan untuk client dan programmer sehingga ditulis dengan spesifikasi yang mudah dimengerti dan mudah diakses.

* 1. ***Definisi dan Istilah***

WEB-TIX dirancang untuk membantu mempercepat serta mempermudah transaksi pembelian tiket bioskop, karena aplikasi dirancang dengan sasaran tidak hanya khalayak muda sehingga tampilan dirancang agar tampak seminimalis mungkin untuk membantu semua pengguna baik yang sudah terbiasa dalam melakukan transaksi secara online atau yang masih awam dengan teknologi tersebut.

Server merupakan suatu sistem komputer yang menyediakan jenis layanan tertentu pada sebuah jaringan komputer

FR: Fungsional

NFR: Non Fungsional

MOD: Modul

APPL: Analisis dan Perancangan Perangkal Lunak

* 1. ***Referensi***

Kami mendapatkan refresensi dari data atau file file yang sudah dikerjakan dalam Spesifikasi Kebutuhan Perangkat Lunak(SKPL) sebelumnya.

* 1. ***Sistematika Pembahasan***

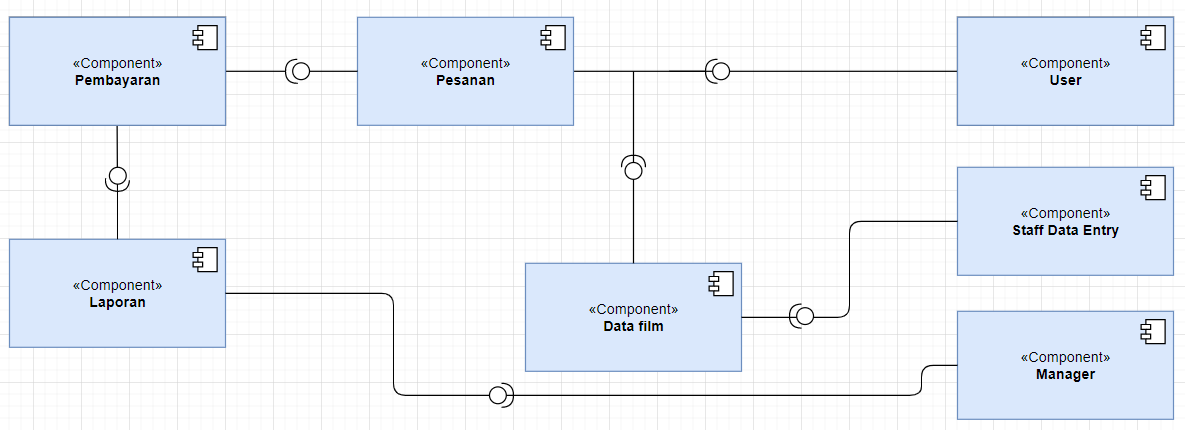
Dokumen perancangan perangkat lunak penjualan tiket online ini berisikan rincian dari deskripsi pengembangan perangkat lunak. Sistematika pembahasan yang akan dilakukan dikelompokkan adalah sebagai berikut.

1. Bab 1 pendahuluan, berisi tujuan penulisan dokumen, lingkup masalah yang ditangani dalam perangkat lunak yang akan dibangun serta deskripsi umum dokumen.
2. Bab 2 deskripsi perancangan global, berisikan uraian singkat mengenai perangkat lunak yang akan dikembangkan dan bagaimana rancangan aplikasi itu dibuat.
3. Bab 3 rincian perancangan, berisikan detail komponen dan fitur yang akan diimplementasikan pada perangkat lunak. Bagian ini terdiri dari
4. Bab 4 matriks kerunutan

1. **Deskripsi Perancangan Global**
   1. ***Rancangan Lingkungan Implementasi***

|  |  |
| --- | --- |
| Sistem Operasi | Window 7, 8, dan 10 |
| DBMS | MySql |
| Bahasa Pemrograman | JavaScript, PHP |
| Browser | Google Chrome 80.0.3987 keatas atau Mozilla Firefox 74.0 keatas |

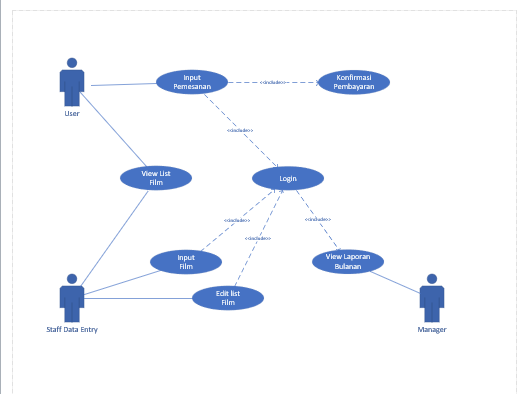
* 1. ***Deskripsi Arsitektural***



* 1. ***Deskripsi Komponen***

|  |  |  |
| --- | --- | --- |
| **No** | **Nama Komponen** | **Keterangan** |
| 1 | User | Halaman user |
| 2 | Staff Data Entry | Halaman Staff Data Entry |
| 3 | Data Film | Database yang menyimpan data film |
| 4 | Pesanan | Database yang menyimpan pesanan |
| 5 | Pembayaran | Halaman Pembayaran |
| 6 | Laporan | Halaman Akumulasi Pendapatan |
| 7 | Manager | Halaman Manager |

1. **Perancangan Rinci**
   1. ***Realisasi Use Case***
      1. **Use Case <nama use case 1>**

****

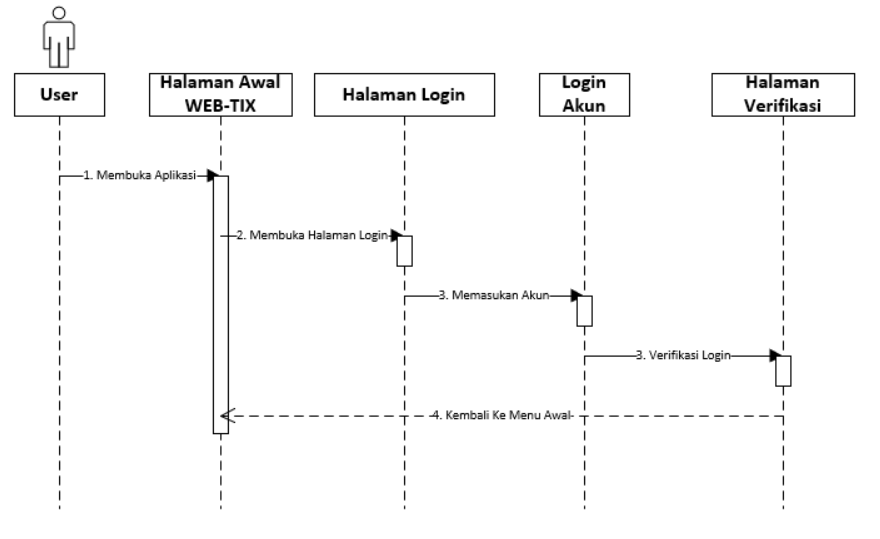
Jika use case ini akan direalisasikan dalam bentuk aplikasi berbasis web, maka subbab yang terkait dengan perancangan elemen aplikasi berbasis web harus diisi.

* + - 1. **Identifikasi Kelas**

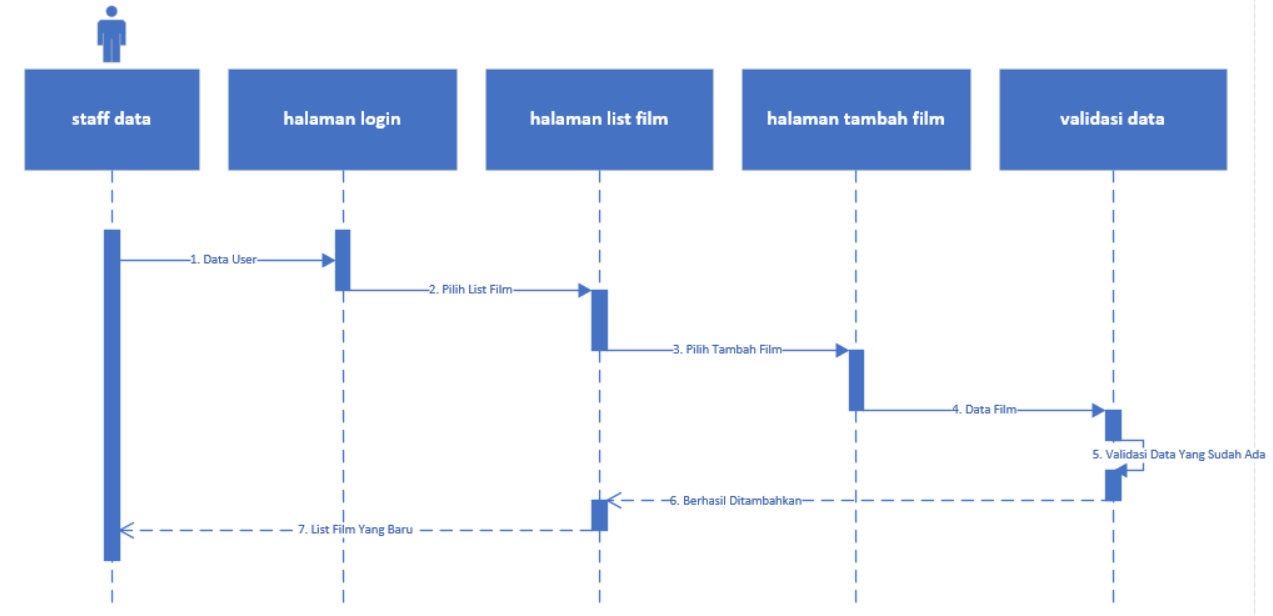
|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Tipe Kelas* |
| 1 | Akun | Controller |
| 2 | User | Controller |
| 3 | Staff Data Entry | Controller |
| 4 | Manager | Controller |
| 5 | Pembayaran | Controller |
| 6 | Pemesanan | Controller |
| 7 | Bank | Controller |
| 8 | Tiket | Controller |

* + - 1. **Sequence Diagram**

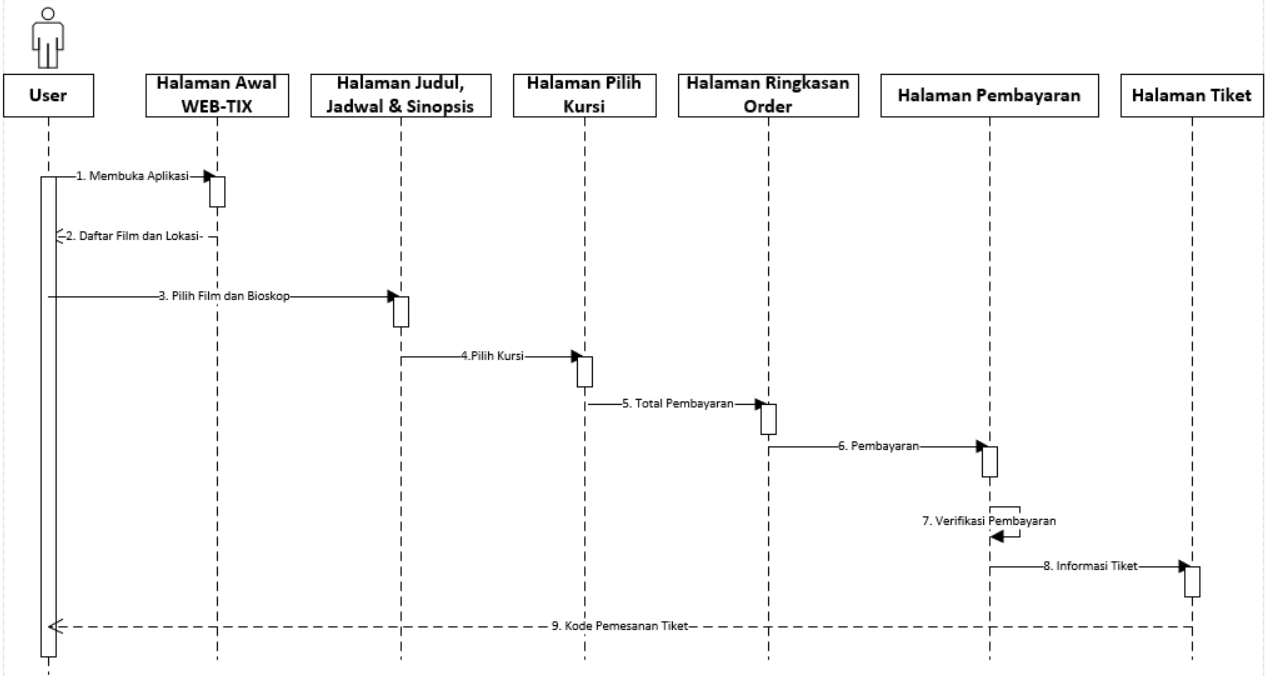
1. Login



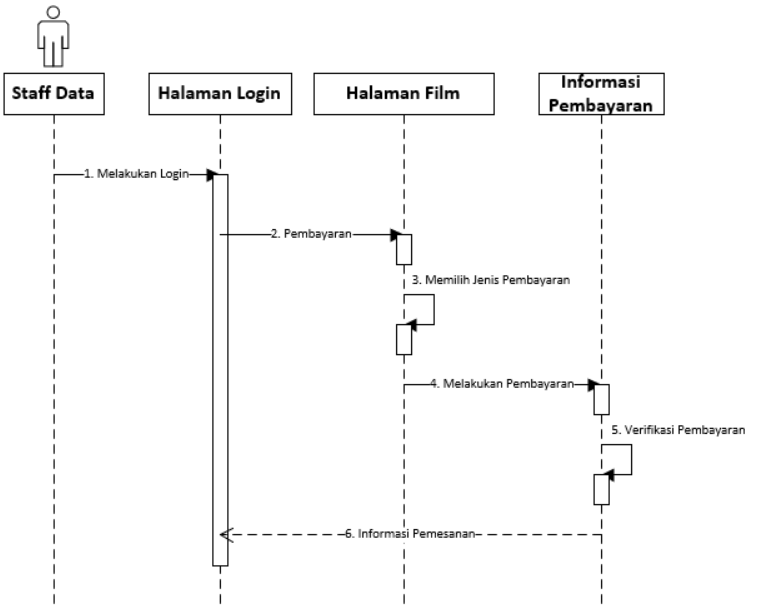
1. Edit data film



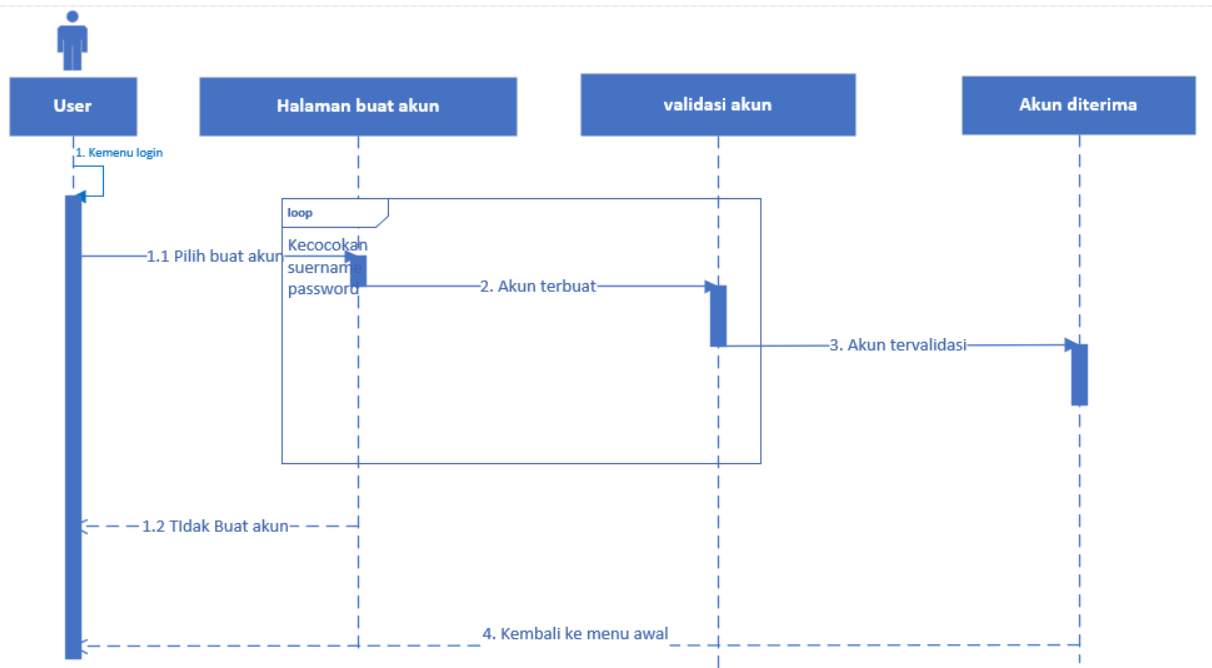
1. Input Pemesanan



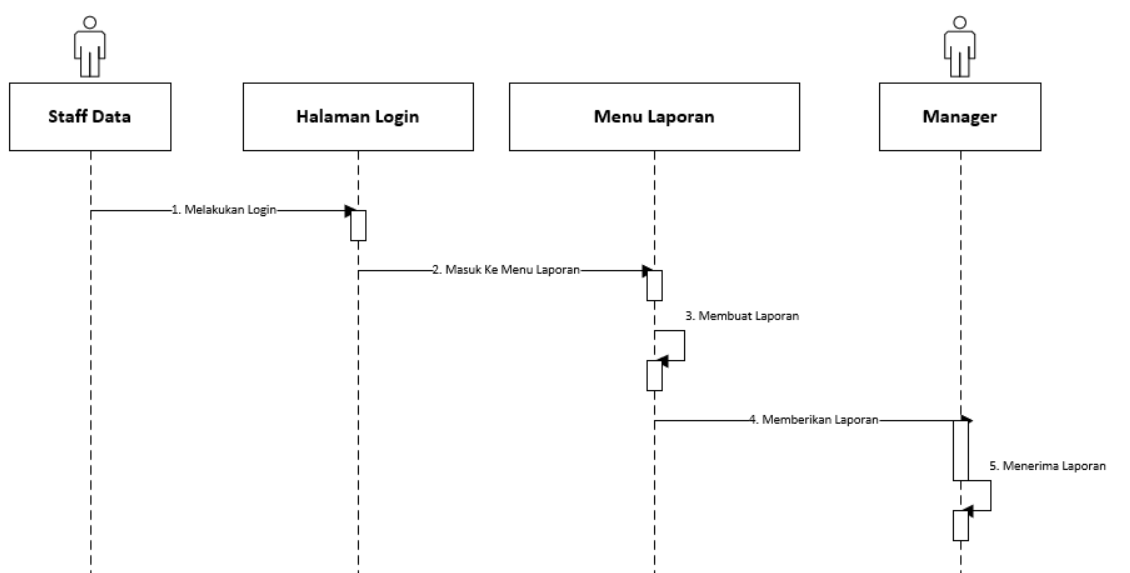
1. Pembayaran



1. Registrasi Akun

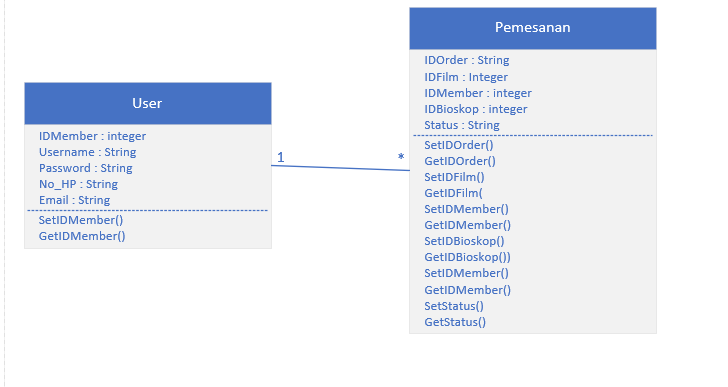
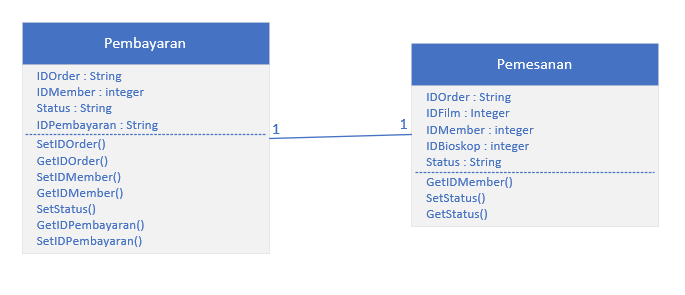
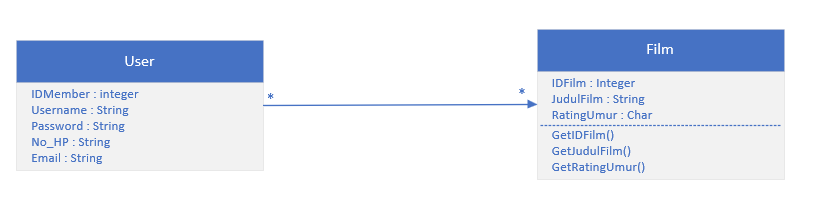
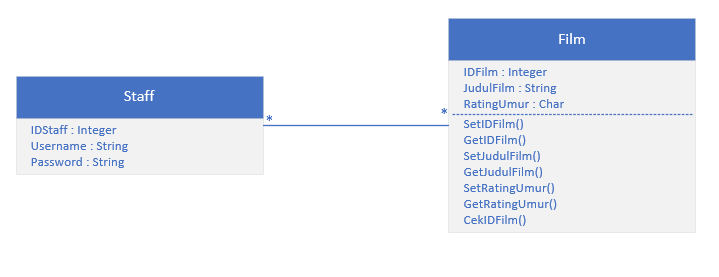
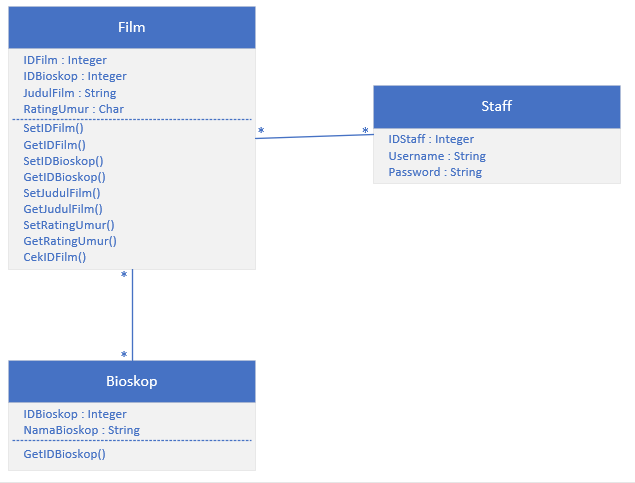
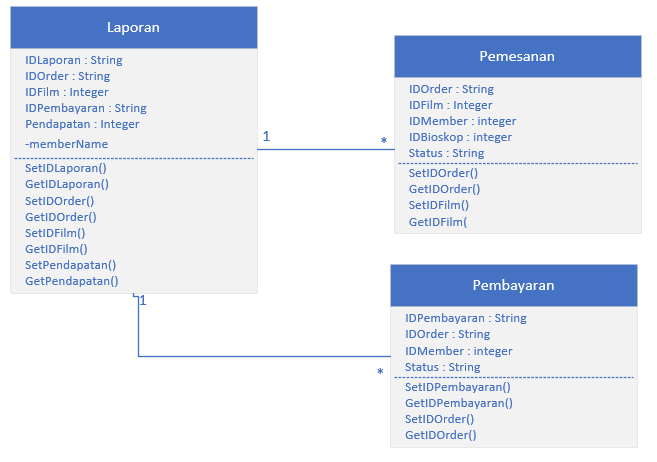


1. Laporan



* + - 1. **Diagram Kelas**

*Buatlah diagram kelas untuk use case tersebut. buat class diagram* ***BUKAN KESELURUHAN, tapi PER USE CASE***

1. Input Pemesanan  
   
2. Konfirmasi Pembayaran  
   
3. View List\_Film  
   
4. Input List\_Film  
   
5. Edit List\_Film  
   
6. Laporan Bulanan  
   
   1. ***Perancangan Detail Kelas***

*Bagian ini diisi dengan daftar seluruh kelas dalam tabel berikut:*

|  |  |  |
| --- | --- | --- |
| *No* | *Nama Kelas Perancangan* | *Nama Kelas Analisis Terkait* |
|  |  |  |
|  |  |  |
|  |  |  |

*Untuk setiap kelas:*

* *identifikasi operasi (mengacu pada tanggung-jawab kelas), termasuk visibility-nya*
* *identifikasi atribut, termasuk visibility-nya*
  + 1. **Kelas <User>**

Bagian ini diisi dengan daftar operasi dan atribut Buat untuk setiap kelas.

*Nama Kelas : User*

|  |  |  |
| --- | --- | --- |
| ***Nama Operasi*** | ***Visibility***  ***(private, public)*** | ***Keterangan*** |
| *SetIDMember()* | *Public* | *Method untuk menyimpan IDMember* |
| *GetIDMember()* | *Public* | *Method untuk memanggil IDMember* |
| *SetUsername()* | *Public* | *Method untuk menyimpan Username* |
| *GetUsername()* | *Public* | *Method untuk memanggil Username* |
| *SetPassword()* | *Public* | *Method untuk menyimpan Password* |
| *GetPassword()* | *Public* | *Method untuk memanggil Password* |
| *SetNoHP()* | *Public* | *Method untuk menyimpan No\_HP* |
| *GetNoHP()* | *Public* | *Method untuk memanggil No\_HP* |
| *SetEmail()* | *Public* | *Method untuk menyimpan Email* |
| *GetEmail()* | *Public* | *Method untuk memanggil Email* |
| ***Nama Atribut*** | ***Visibility***  ***(private, public)*** | ***Tipe*** |
| *IDMember* | *Private* | *Integer* |
| *Username* | *Private* | *String* |
| *Password* | *Private* | *String* |
| *No\_HP* | *Private* | *String* |
| *Email* | *Private* | *String* |

* + 1. **Kelas <Pemesanan>**

**Nama Kelas : Pemesanan**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDMember() | Public | Method untuk menyimpan IDMember |
| GetIDMember() | Public | Method untuk memanggil IDMember |
| SetIDOrder() | Public | Method untuk menyimpan IDOrder |
| GetIDOrder() | Public | Method untuk memanggil IDOrder |
| SetIDFilm() | Public | Method untuk menyimpan IDFilmd |
| GetIDFilm() | Public | Method untuk memanggil IDFilm |
| SetStatus() | Public | Method untuk menyimpan Status |
| GetStatus() | Public | Method untuk memanggil Status |
| SetIDBioskop() | Public | Method untuk menyimpan IDBioskop |
| GetIDBioskopl() | Public | Method untuk memanggil IDBioskop |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDOrder | Private | String |
| IDFilm | Private | Integer |
| IDMember | Private | Integer |
| Status | Private | String |
| IDBioskop | Private | Integer |

**3.2.3** **Kelas <Laporan>**

**Nama Kelas : Laporan**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDLaporan() | Public | Method untuk menyimpan IDLaporan |
| GetIDLaporan() | Public | Method untuk memanggil IDLaporan |
| SetIDOrder() | Public | Method untuk menyimpan IDOrder |
| GetIDOrder() | Public | Method untuk memanggil IDOrder |
| SetIDFilm() | Public | Method untuk menyimpan IDFilm |
| GetIDFilm() | Public | Method untuk memanggil IDFilm |
| SetIDPembayaran() | Public | Method untuk menyimpan IDPembayaran |
| GetIDPembayaran() | Public | Method untuk memanggil IDPembayaran |
| SetPendapatan() | Public | Method untuk menyimpan Pendapatan |
| GetPendapatan() | Public | Method untuk memanggil Pendapatan |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDLaporan | Private | String |
| IDOrder | Private | String |
| IDFilm | Private | Integer |
| IDPembayaran | Private | String |
| Pendapatan | Private | Integer |

**3.2.4** **Kelas <Manager>**

**Nama Kelas : Manager**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDManager() | Public | Method untuk menyimpan IDManager |
| GetIDManager() | Public | Method untuk memanggil IDManager |
| SetUsername() | Public | Method untuk menyimpan Username |
| GetUsername() | Public | Method untuk memanggil Username |
| SetPassword() | Public | Method untuk menyimpan Password |
| GetPassword() | Public | Method untuk memanggil Password |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDManager | Private | Integer |
| Username | Private | String |
| Password | Private | String |

**3.2.5** **Kelas <Film>**

**Nama Kelas : Film**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDFilm() | Public | Method untuk menyimpan IDFilm |
| GetIDFilm() | Public | Method untuk memanggil IDFilm |
| SetJudulFilm() | Public | Method untuk menyimpan JudulFilm |
| GetJudulFilm() | Public | Method untuk memanggil JudulFilm |
| SetRatingUmur() | Public | Method untuk menyimpan RatingUmur |
| GetRatingUmur() | Public | Method untuk memanggil RatingUmur |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDFilm | Private | Integer |
| JudulFilm | Private | String |
| RatingUmur | Private | Char |

**3.2.6** **Kelas <Staff>**

**Nama Kelas : Staff**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDStaff() | Public | Method untuk menyimpan IDStaff |
| GetIDStaff() | Public | Method untuk memanggil IDStaff |
| SetUsername() | Public | Method untuk menyimpan Username |
| GetUsername() | Public | Method untuk memanggil Username |
| SetPassword() | Public | Method untuk menyimpan Password |
| GetPassword() | Public | Method untuk memanggil Password |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDStaff | Private | Integer |
| Username | Private | String |
| Password | Private | String |

**3.2.7** **Kelas <Bioskop>**

**Nama Kelas : Bioskop**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetNamaBioskop() | Public | Method untuk menyimpan NamaBIoskop |
| GetNamaBioskop() | Public | Method untuk memanggil NamaBioskop |
| SetIDBioskop() | Public | Method untuk menyimpan IDBioskop |
| GetIDBioskop() | Public | Method untuk memanggil IDBioskop |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| NamaBioskop | Private | string |
| IDbioskop | Private | Integer |

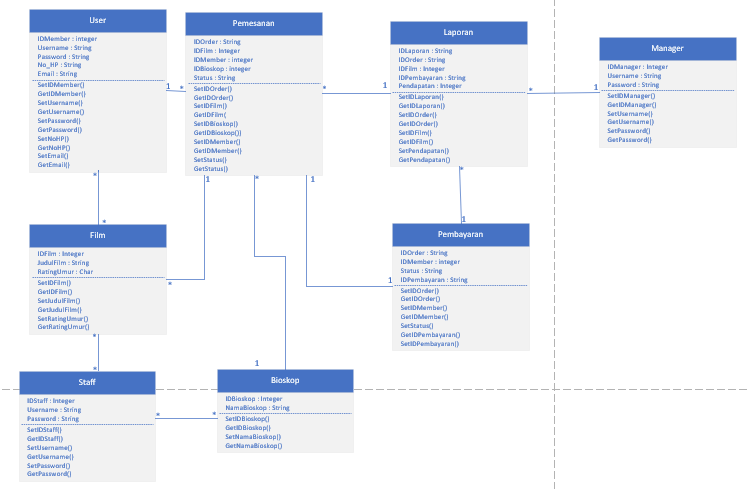
**3.2.8** **Kelas <Pembayaran>**

**Nama Kelas : Pembayaran**

|  |  |  |
| --- | --- | --- |
| *Nama Operasi* | *Visibility*  *(private, public)* | *Keterangan* |
| SetIDMember() | Public | Method untuk menyimpan IDMember |
| GetIDMember() | Public | Method untuk memanggil IDMember |
| SetIDOrder() | Public | Method untuk menyimpan IDOrder |
| GetIDOrder() | Public | Method untuk memanggil IDOrder |
| SetIDPembayran() | Public | Method untuk menyimpan IDPembayaran |
| GetIDPembayaran() | Public | Method untuk memanggil IDPembayaran |
| SetStatus() | Public | Method untuk menyimpan Status |
| GetStatus() | Public | Method untuk memanggil Status |
| *Nama Atribut* | *Visibility*  *(private, public)* | *Tipe* |
| IDOrder | Private | String |
| IDPembayaran | Private | String |
| IDMember | Private | Integer |
| Status | Private | String |

* 1. ***Diagram Kelas Keseluruhan***

*Bagian ini diisi dengan diagram kelas keseluruhan.*

**

* 1. ***Algoritma/Query***

*Bagian ini hanya diisi untuk kerangka algoritma untuk* **method-method****dari Class** *yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat subbab per kelas.*

Contoh:

Nama Kelas : Pemesanan

Nama Operasi : Memesan Tiket

Algoritma : (Algo-001)

1. Saat event dipilih, sistem langsung memproses pesanan yang sudah dipilih oleh user dengan mengirim ID user yang memesan

2. Pemesanan akan di insert ke dalam database dengan status= “Belum Terbayar”

3. Sistem akan memberikan tata cara pembayaran berserta rekening tujuan untuk melakukan pembayaran

*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| Q-001 | Insert into Pemesanan(IDOrder, IDFilm, IDMember, Status) Values($IDOrder, $IDFilm, $IDMember, “Belum Terbayar”); | Memasukan data pemesanan ke *database* dengan status “Belum Terbayar” |

Nama Kelas : Pemesanan

Nama Operasi : Verifikasi tiket

Algoritma : (Algo-002)

1. Sistem akan memilih data Tiket yang memiliki status “Belum Terbayar”.

2. Saat Sistem menganggap data bisa diverifikasi pembayarannya, Sistem Otomatis mengubah status pada database menjadi “Terbayar”

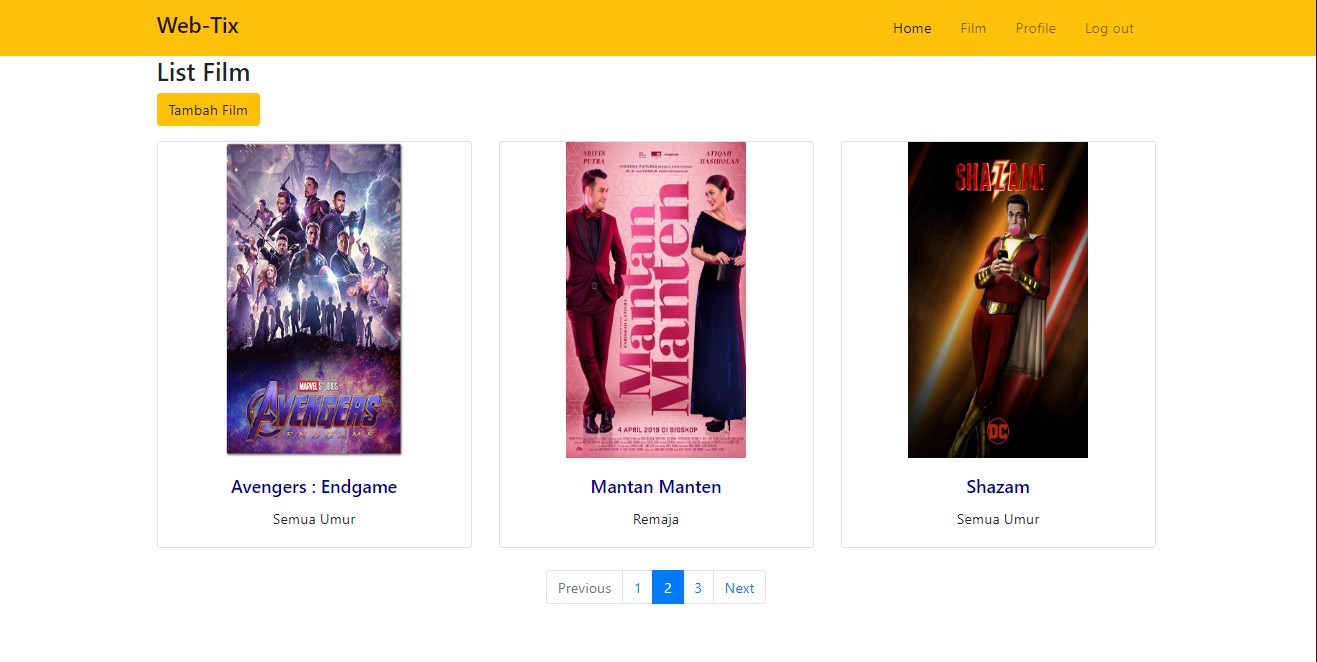
*{Jika mengacu query tertentu, lengkapi tabel query di bawah}*

*Query :*

|  |  |  |
| --- | --- | --- |
| *No Query* | *Query* | *Keterangan* |
| Q-002 | UPDATE Pemesanan  SET Status= ”Terbayar”  WHERE $IDOrder; | Mengubah Status pada Pemesanan menjasi “Terbayar” |

* 1. ***Perancangan Antarmuka***

*Bagian ini diisi dengan versi awal prototipe antarmuka.*

**

*Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya, misalnya seperti di bawah ini:*

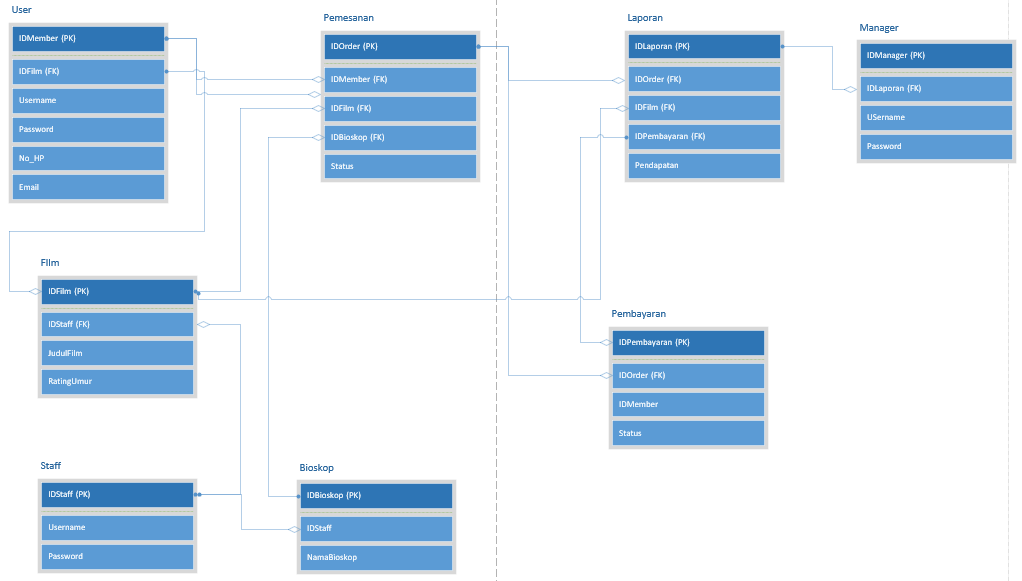
*Antarmuka : {diisi dengan no. layar atau no gambar rancangan antarmuka}*

|  |  |  |  |
| --- | --- | --- | --- |
| **Id\_Objek** | **Jenis** | **Nama** | **Keterangan** |
| HEAD1 | Link | Home | Link yang membawa ke halaman utama |
| HEAD2 | Link | Film | Link yang membawa ke halaman list Film |
| HEAD3 | Link | Profile | Link yang membawa ke halaman Profile |
| HEAD4 | Link | Login atau Logout | Link yang membawa ke halaman login maupun logout |

*Jika objek dikaitkan ke File lain (misalnya file gambar, file teks), berikan nama file terkait dan deskripsi ringkas dalam kolom keterangan*

* 1. ***Perancangan Representasi Persistensi Kelas***

*Bagian ini diisi dengan rancangan skema basisdata dan traceability-nya terhadap kelas entity. (PEMBUATAN SKEMA RELASI)*

**

1. **Matriks Kerunutan**

*Mapping use case dengan kelas-kelas terkait*

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Usecase Terkait** | **Kelas** |
| FR-01 | editfilm, hapusfilm, login, kelolaAkun | staff entry |
| FR-02 | Pembayaran, kelolaAkun, registrasi, login, viewfilm, | User |
| FR-03 | login, viewlaporan | Manager |
| FR-04 | editfilm, hapusfilm, viewfilm | Film |
| FR-05 | pembayaran, pemesanan | Ticket |
| FR-06 | pembayaran | Pembayaran |