**SKPL**-01

SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Aplikasi Tiket Bioskop

untuk:

Bioskop

Dipersiapkan oleh:

Tyo Revandi (1301180314)

Fikri Al Fath Asyari (1301184090)

Mohammad Rifki Nurdiansyach (1301184244)

Seanford Revydo Suyana (1301184444)

Program Studi S1 Teknik Informatika – Fakultas Informatika

Universitas Telkom

Jalan Telekomunikasi Terusan Buah Batu, Bandung

Indonesia

|  |  |  |  |
| --- | --- | --- | --- |
| Program Studi S1 Teknik Informatika  -  Fakultas Informatika | Nomor Dokumen | | Halaman |
| *SKPL-01* | | 1/15 |
| Revisi | *<nomor revisi>* | *Tgl: <isi tanggal>* |

# Daftar Perubahan

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

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

# Daftar Halaman Perubahan

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

# Daftar Isi

[Daftar Perubahan 1](#_Toc702193)

[Daftar Halaman Perubahan 2](#_Toc702194)

[Daftar Isi 3](#_Toc702195)

[1. Pendahuluan 4](#_Toc702196)

[1.1 Tujuan Penulisan Dokumen 4](#_Toc702197)

[1.2 Ruang Lingkup / Cakupan Dokumen 4](#_Toc702198)

[1.3 Definisi, Singkatan, dan Akronim 4](#_Toc702199)

[1.4 Referensi 4](#_Toc702200)

[2. Deskripsi Global Perangkat Lunak 5](#_Toc702201)

[2.1 Statement of Objective Perangkat Lunak 5](#_Toc702202)

[2.2 Perspektif dan Fungsi Perangkat Lunak 5](#_Toc702203)

[2.3 Profil dan Karakteristik Pengguna 5](#_Toc702204)

[2.4 Lingkungan Operasi 5](#_Toc702205)

[2.5 Batasan Perangkat Lunak / Sistem 5](#_Toc702206)

[2.6 Asumsi dan Dependensi 6](#_Toc702207)

[3. Deskrpsi Rinci Perangkat Lunak 7](#_Toc702208)

[3.1 Deskripsi Kebutuhan 7](#_Toc702209)

[3.1.1 Kebutuhan Fungsional 7](#_Toc702210)

[3.1.2 Kebutuhan Non-Fungsional 7](#_Toc702211)

[3.2 Pemodelan Analisis 7](#_Toc702212)

[3.2.1 Usecase Diagram 7](#_Toc702213)

[3.2.2 Class Diagram: 8](#_Toc702214)

[4. Kebutuhan Antarmuka Eksternal 9](#_Toc702215)

[4.1 Antarmuka Pengguna 9](#_Toc702216)

[4.2 Antarmuka Perangkat Keras 9](#_Toc702217)

[4.3 Antarmuka Perangkat Lunak 9](#_Toc702218)

[4.4 Antarmuka Komunikasi 9](#_Toc702219)

[5. Requirements Lain 10](#_Toc702220)

# Pendahuluan

## Tujuan Penulisan Dokumen

Dokumen ini berisi penjelasan Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dengan menjelaskan sebuah perangkat lunak untuk pemesanan tiket bioskop yang akan dibuat dengan baik berupa gambaran umum dan penjelasan tentang kebutuhan yang akan digunakan pada perangkat lunak ini.

## Ruang Lingkup / Cakupan Dokumen

Perangkat lunak yang akan dikembangkan adalah perangkat lunak Pemesanan Tiket Bioskop yaitu merupakan perangkat lunak yang digunakan untuk mempermudah proses pemesanan tiket bioskop. Aplikasi ini membuat User menjadi lebih efisien waktu ketika melakukan pemesanan tiket bioskop dengan memesan secara online dan tidak perlu mengantri di lokasi bioskop.

## Definisi, Singkatan, dan Akronim

Definisi dari istilah yang akan digunakan pada dokumen ini adalah:

1. User adalah orang yang hanya bisa menggunakan aplikasi untuk melakukan pembelian tiket bioskop secara online.
2. Admin adalah orang yang mengelola penuh sistem seperti mengupdate informasi dan melayani user dalam fitur customer service.

# Deskripsi Global Perangkat Lunak

## Statement of Objective Perangkat Lunak

Sebuah project pembuatan aplikasi berbasis desktop yang dapat melihat daftar film yang ditayangkan di bioskop pada tanggal yang ditentukan dan memesan tiketnya secara online yang dapat mempermudah User dalam pembelian tiket.

## Perspektif dan Fungsi Perangkat Lunak

Kegunaan dari perangkat lunak ini adalah mempermudah User dalam pembelian tiket agar User tidak perlu mengantri dan mempermudah User memperoleh informasi tentang film yag sedang diputar atau yang akan diputar. Keadaan perangkat lunak ini membuat kegiatan operasional dalam pemesanan suatu tiket menjadi lebih efisien waktu serta dapat mengurangi kesalahan dalam pendataan.

## Profil dan Karakteristik Pengguna

Pengguna dari perangkat lunak ini adalah seluruh kalangan masyarakat yang ingin membeli tiket bioskop tanpa perlu mengantri ke lokasi bioskop.

|  |  |  |
| --- | --- | --- |
| Kategori Pengguna | Tugas | Hak akses aplikasi |
| Admin | Mengelola sistem perangkat lunak. | Seluruh sistem dan aktivitas. |
| User | Mengisi identitas diri,  Melakukan pemesanan tiket dan menyelesaikan pembayaran. | Mengakses login, pendaftaran akun, pemesanan tiket dan pembayaran. |

## Lingkungan Operasi

Lingkungan operasi dari aplikasi Tiket Bioskop ini adalah :

* 1. Server : Admin
  2. Client : User
  3. Operating system : Windows

## Batasan Perangkat Lunak / Sistem

Batasan – batasan pada aplikasi Tiket Bioskop ini ialah :

1. User dapat menyelesaikan pendaftaran akun dengan mengisi identitas diri.
2. User dapat melakukan login ketika sudah menyelesaian pendaftaran akun.
3. User dapat melakukan pemesanan tiket ketika sudah melakukan login.
4. User dapat melakukan pembayaran ketika sudah melakukan pemesanan tiket.
5. User akan mendapatkan e-tiket ketika sudah menyelesaikan pembayaran.

## Asumsi dan Dependensi

Asumsi dan Dependensi yang digunakan dalam pengembangan aplikasi Tiket Bioskop adalah :

* User hanya berperan sebagai pelanggan.
* Semua user dapat melakukan perubahan data diri.
* Admin dapat mengelola sistem, data user, dan customer service.

# Deskripsi Rinci Perangkat Lunak

Bab ini berisi akan informasi untuk kebutuhan sistem dari segi kebutuhan fungsional dan kebutuhan non-fungsional dan di bab ini juga melampirkan informasi berupa Use Case Diagram, Use case Scenario, dan Class Diagram.

## Deskripsi Kebutuhan

### Kebutuhan Fungsional

User:

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Kebutuhan | Fungsi | Deskripsi |
| 1. | FR-01 | Pendaftaran akun | User dapat melakukan pendaftaran akun di halaman menu awal. |
| 2. | FR-02 | Melihat daftar film | Fungsi ini dapat digunakan oleh user untuk melihat daftar film yang tersedia. |
| 3. | FR-03 | Memilih film | Fungsi ini dapat digunakan oleh user untuk memilih film yang akan ditonton. |
| 4. | FR-04 | Memilih lokasi bioskop | Fungsi ini dapat digunakan oleh user untuk memilih lokasi bioskop yang akan dipakai. |
| 5. | FR-05 | Memilih tipe studio | Fungsi ini dapat digunakan oleh user untuk memilih tipe studio apa yang akan dipakai. (contoh : regular, velvet, dsb.) |
| 6. | FR-06 | Memilih jam tayang | Fungsi ini dapat digunakan oleh user untuk memilih jam tayang dari film. |
| 7. | FR-07 | Memilih kursi | Fungsi ini dapat digunakan oleh user untuk memilih kursi yang akan digunakan untuk menonton film. |
| 8. | FR-08 | Melakukan pembayaran | Fungsi ini dapat digunakan oleh user untuk melakukan pembayaran secara online setelah memilih film, lokasi bioskop, tipe studio, jam tayang dan kursi. |
| 9. | FR-09 | Melihat kuitansi | Fungsi ini dapat digunakan oleh user untuk melihat kuitansi pembayaran. |
| 10. | FR-10 | Melihat e-tiket | Fungsi ini dapat digunakan oleh user untuk melihat e-tiket yang akan digunakan untuk ditukar menjadi tiket fisik. |

Admin:

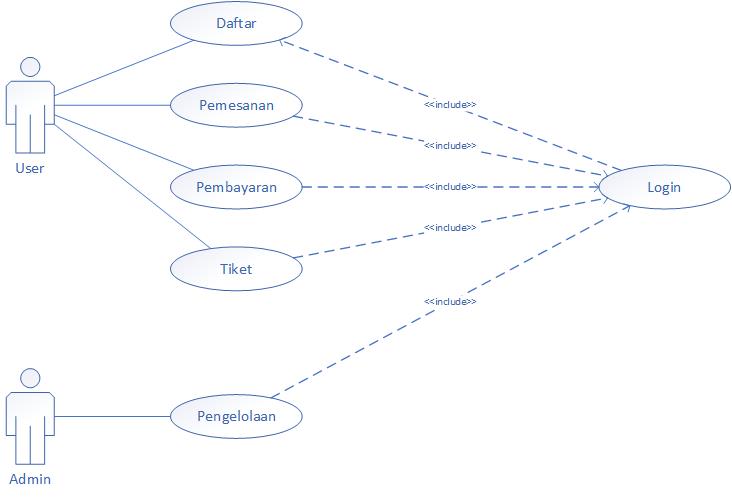
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Kode Kebutuhan | Fungsi | Deskripsi |
| 1. | FR-11 | RUD data user | Fungsi ini dapat digunakan oleh Admin untuk melihat, mengupdate dan menghapus data user. |
| 2. | FR-12 | R data pemesanan | Fungsi ini dapat digunakan oleh Admin untuk melihat data pemesanan. |
| 3. | FR-13 | R data pembayaran | Fungsi ini dapat digunakan oleh Admin untuk melihat data pembayaran. |
| 4. | FR-14 | CRUD data bioskop | Fungsi ini dapat digunakan oleh Admin untuk melihat, mengupdate dan menghapus data bioskop. |
| 5. | FR-15 | CRUD data film | Fungsi ini dapat digunakan oleh Admin untuk melihat, mengupdate dan menghapus data film. |

### Kebutuhan Non-Fungsional

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Quality | Kode Kebutuhan | Deskripsi |
| 1. | System Work | NFR-01 | Memungkinkan untuk sistem bekerja dalam 1x24 jam. |
| 2. | View Quality | NFR-02 | Memungkinkan user dapat mudah memahami tampilan aplikasi. |
| 3. | Display resolution | NFR-03 | Membuat aplikasi tersebut dapat ditampilkan di semua macam resolusi layar handphone. |
| 4. | Wide-scale Server | NFR-04 | Memungkinkan banyak user yang mengakses aplikasi ini tanpa adanya *server down.* |

## Pemodelan Analisis

### Usecase Diagram



#### Usecase Scenario **#1**

Nama use case : **DAFTAR**

Aktor : **USER**

Pre-condition : **Aktor ingin menggunakan program dan belum mempunyai akun**

Post-condition : **Aktor telah melakukan pendaftaran dan berinterakasi dengan**

**program**

Deskripsi : **Aktor melakukan pendaftaran untuk dapat melakukan login**

|  |  |
| --- | --- |
| USER | SYSTEM |
| 1. Menginputkan identitas User   1.1 Jika mempunyai KTP:   * NIK * Nomor HP * Email * Password   1.2 Jika tidak mempunyai KTP:   * Nama Lengkap * Tempat dan Tanggal Lahir * Alamat * NIK Orang Tua/Wali * Nomor HP (User) * Nomor HP (Orang Tua/Wali) * Email * Password |  |
| 1. Mengklik **Daftar** |  |
|  | 1. Mencari data 1.1 dan 1.2 di dalam database |
|  | 1. Jika gagal (terdapat data yang sama), notifikasi akan muncul dan kembali ke no 1 |
|  | 1. Jika berhasil, notifikasi akan muncul dan lanjut ke Nomor 6 |
|  | 1. Menampilkan preview konfirmasi data (Data 1.1 dan 1.2 ditampilkan) |
| 1. Mengklik **Konfirmasi** atau **Batal** |  |
|  | 1. Jika **Batal**, kembali ke Nomor 1 |
|  | 1. Jika **Konfirmasi**, lanjut ke   Nomor 10 |
|  | 1. Menampilkan halaman **Login** |

#### Usecase Scenario **#2**

Nama use case : **LOGIN**

Aktor : **USER**

Pre-condition : **Aktor ingin menggunakan program dan belum melakukan login**

Post-condition : **Aktor telah melakukan login dan berinterakasi dengan program**

Deskripsi : **Aktor melakukan login untuk dapat menggunakan program**

|  |  |
| --- | --- |
| USER | SYSTEM |
| 1. Menginputkan email dan password |  |
| 1. Mengklik **Login** |  |
|  | 1. Eksekusi validasi email dan password dengan yang tersimpan di dalam database |
|  | 1. Jika gagal, notifikasi akan muncul dan kembali ke Nomor 1 |
|  | 1. Jika sesuai, notifikasi akan muncul dan lanjut ke Nomor 6 |
|  | 1. Menampilkan **Halaman Utama** program |

#### Usecase Scenario **#3**

Nama use case : **PEMESANAN**

Aktor : **USER**

Pre-condition : **Aktor sudah mendaftar, login dan belum memesan tiket**

Post-condition : **Aktor telah berhasil memesan tiket**

Deskripsi : **Aktor memesan tiket**

|  |  |
| --- | --- |
| USER | SYSTEM |
|  | 1. Menampilkan film yang tersedia |
| 1. Memilih film yang ingin ditonton |  |
| 1. Mengklik **Pesan Tiket** |  |
|  | 1. Menampilkan halaman **Pemesanan**. |
|  | 1. Menampilkan jadwal tayang yang tersedia |
| 1. Memilih jadwal tayang yang tersedia |  |
|  | 1. Menampilkan daftar alamat bioskop yang menampilkan film sesuai dengan jadwal tayang yang tersedia |
| 1. Memilih alamat bioskop yang tersedia |  |
|  | 1. Menampilkan kursi yang tersedia |
| 1. Memilih kursi yang tersedia |  |
| 1. Mengklik **Konfirmasi Pemesanan** |  |
|  | 1. Menampilkan preview konfirmasi pemesanan (Data no 2, 6, 8, dan 10 ditampilkan) |
| 1. Mengklik **Konfirmasi** atau **Batal** |  |
|  | 1. Jika **Cancel**, kembali ke no 1 |
|  | 1. Jika **Confirm**, lanjut ke **Pembayaran** |

#### **Usecase Scenario #4**

Nama use case : **PEMBAYARAN**

Aktor : **USER**

Pre-condition : **Aktor sudah memesan tiket dan belum melakukan pembayaran**

Post-condition : **Aktor telah berhasil menyelesaikan pembayaran**

Deskripsi : **Aktor menyelesaikan pembayaran**

|  |  |
| --- | --- |
| USER | SYSTEM |
|  | 1. Menampilkan deskripsi pemesanan dan total harga yang harus dibayar |
| 1. Mengklik **Bayar** atau **Batal** |  |
|  | 1. Jika **Batal**, kembali ke halaman **Pemesanan** Nomor 12 |
|  | 1. Jika **Bayar**, lanjut ke Nomor 5 |
|  | 1. Menampilkan Instruksi Pembayaran melalui Mobile Banking   Contoh:  Pembayaran Melalui MB Mandiri 2.1. Pilih Bayar pada halaman utama  2.2. Buat Pembayaran Baru  2.3. Pilih Multipayment  2.4. Pilih Penyedia Jasa:  theMovieKhamphanks  2.5. Masukan Nomor HP User  2.6. Klik Lanjut  2.7. Pilih Data Pembayaran yang sesuai  2.8. Klik Bayar |
| 3. User melakukan pembayaran sesuai  dengan instruksi |  |
|  | 4. Mengkonfirmasi pembayaran |
|  | 5. Jika **gagal**, lanjut ke Nomor 7 |
|  | 6. Jika **berhasil**, lanjut ke Nomor 11 |
|  | 7. Menampilkan notifikasi, dan  menampilkan form upload manual  bukti pembayaran |
| 8. User mengupload bukti pembayaran  secara manual |  |
|  | 9. Menyimpan bukti pembayaran |
|  | 10. Mengkonfirmasi bukti pembayaran |
|  | 11. Lanjut ke **Tiket** |

#### **Usecase Scenario #5**

Nama use case : **TIKET**

Aktor : **USER**

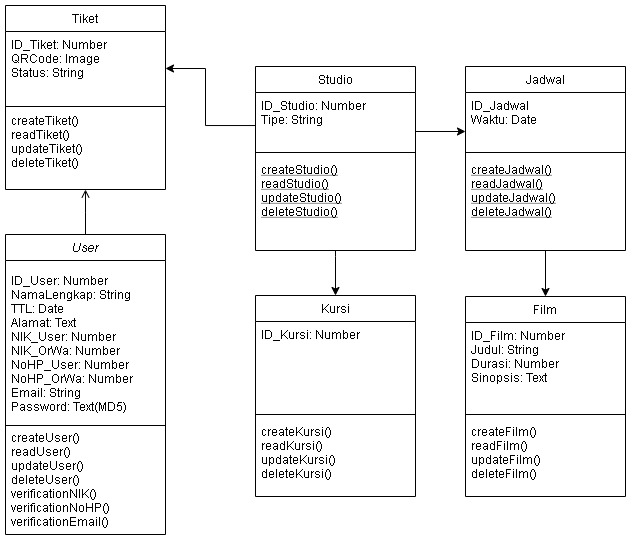
Pre-condition : **Aktor sudah memesan tiket dan belum melakukan pembayaran**

Post-condition : **Aktor telah berhasil menyelesaikan pembayaran**

Deskripsi : **Aktor menyelesaikan pembayaran**

|  |  |
| --- | --- |
| USER | SYSTEM |
|  | 1. Menampilkan E-Tiket dan Nota Pembayaran |
|  | 1. Menampilkan opsi **Simpan ke Galeri** dan **Selesai.** |
| 1. Memilih **Simpan ke Galeri** atau **Selesai** |  |
|  | 1. Jika **Simpan ke Galeri**, lanjut ke Nomor 6 |
|  | 1. Jika **Selesai**, lanjut ke Nomor 7 |
|  | 1. Menyimpan E-Tiket dan Nota Pembayaran ke dalam Galeri HP |
|  | 1. Selesai |

### Class Diagram:



# Kebutuhan Antarmuka Eksternal

## Antarmuka Pengguna

Kebutuhan antarmuka pengguna dengan perangkat lunak ini dibuat dengan menggunakan desktop. Tampilan desktop dengan menu yang dapat diakses memiliki berbagai fungsi yang disediakan.

## Antarmuka Perangkat Keras

Perangkat keras yang digunakan minimum:

1. PC/ Laptop (membutuhkan internet)

## Antarmuka Perangkat Lunak

Perangkat lunak yang digunakan dalam perangkat lunak tersebut:

1. Desktop.

## Antarmuka Komunikasi

Pemakai berinterkasi dengan aplikasi pemesanan tiket bioskop ini dengan menggunakan handphone yang memiliki jaringan internet untuk mengakses aplikasi.

# Requirements Lain

Lampiran A: Daftar Kata-Kata Sukar

Homepage : menu home aplikasi.

Orderpage : menu pesan didalam aplikasi

**Lampiran B: Analysis Models**

*Entity Relational Diagram*

