Perancangan system pemesanan tiket bioskop

# Praktikum Pemrograman Platform Web

Laporan ini dibuat sebagai salah satu syarat Tugas Besar Praktikum Pemrograman Platform Web

# Oleh :

|  |  |
| --- | --- |
| **1906007** | **Reza Hasnan Habib** |
| **1906012** | **Shantang Adhimusyari W** |
| **1906030** | **Ahmad Nurhalim** |
| **1906031** | **Aura Fitria Hazar** |
| **1906061** | **Ziadatun Khoirunisa** |
| **1906166** | **Via Nur Afifah** |



**PROGRAM STUDI INFORMATIKA INSTITUT TEKNOLOGI GARUT 2022**

# KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah Ta’ala. yang telah memberikan banyak nikmat, terutama nikmat kesehatan dan kesempatan sehingga proses pembuatan skripsi tentang kajian media ini dapat penulis laksanakan dengan baik. Begitupun atas rahmat Allah Swt penelitian dengan judul pemesanan tiket bioskop bisa selesaikan dengan baik

Penulis menyadari banyak pihak yang membantu dan berkontribusi dalam terselesaikannya penelitian ini. Segala bentuk bantuan, baik berupa dukungan moril dan materil sangat membantu penulis dalam mengumpulkan semangat dan keinginan untuk menyelesaikan studi. Dengan demikian penulis ucapkan terima kasih dengan ketulusan hati kepada pihak-pihak yang telah membantu dan membimbing penulis selama menyusun skripsi ini, yakni kepada:

Bapak Ade Sutedi,Amd.Kom,ST.,M.Kom. selaku pembingbing dalam dalam penelitian.

Semoga Allah Ta’ala memberikan pahala yang berlimpah atas segala bentuk bantuan yang telah diberikan kepada penulis. Selain itu penulis juga berharap agar skripsi ini dapat bermanfaat bagi para pembaca dari berbagai kalangan. Penulis kemudian mengucapkan permohonan maaf jika selama proses penyusunan penelitian banyak melakukan kesalahan, baik berbentuk lisan maupun tulisan, yang dilakukan secara disengaja maupun tidak disengaja. Salam.

# DAFTAR ISI

[KATA PENGANTAR i](#_TOC_250022)

[DAFTAR ISI ii](#_TOC_250021)

[DAFTAR GAMBAR iii](#_TOC_250020)

DAFTAR TABLE iv

1. [PENDAHULUAN](#_TOC_250019)
   1. [Latar Belakang 1](#_TOC_250018)
   2. [Masalah Penelitian 2](#_TOC_250017)
   3. [Tujuan Penelitian 2](#_TOC_250016)
   4. [Pertanyaan Penelitian 2](#_TOC_250015)
   5. [Cakupan Penelitian 2](#_TOC_250014)
   6. Mamfaat Penelitian 2
2. [TINJAUAN PUSTAKA](#_TOC_250013)
   1. [Definisi Operasional 3](#_TOC_250012)
      1. [HTML 3](#_TOC_250011)
      2. [CSS 3](#_TOC_250010)
      3. [BOOTSTARP 3](#_TOC_250009)
      4. [PHP 3](#_TOC_250008)
   2. Relation Unified Proses (RUP) 3
   3. [Unified Modeling Languange 5](#_TOC_250007)
      1. [Use Case 5](#_TOC_250006)
      2. Activity Diagram 6
      3. Squnce Diagram 7
      4. Class Diagram 8
3. HASIL DAN PEMBAHSAN
   1. [Perancangan Sistem 10](#_TOC_250005)
      1. [Use Case Diagram 10](#_TOC_250004)
      2. Scanario Use Case 10
      3. [Activity Diagram 14](#_TOC_250003)
      4. [Squence Diagram 18](#_TOC_250002)
      5. [Class Diagram 21](#_TOC_250001)
   2. [Pembahasan Hasil 21](#_TOC_250000)

**4. KESIMPULAN DAN SARAN DAFTAR PUSTAKA**

# DAFTAR GAMBAR

1. **TINJAUAN PUSTAKA**
   1. **Definisi Operasional 3**
      1. **HTML 3**
      2. **CSS 3**
      3. **BOOTSTARP 3**
      4. **PHP 3**
   2. **Relation Unified Proses (RUP) 3**
   3. **Unified Modeling Languange 5**
      1. **Use Case 5**
      2. **Activity Diagram 6**
      3. **Squnce Diagram 7**
      4. **Class Diagram 8**

**DAFTAR TABEL**

HASIL DAN PEMBAHSAN

* 1. [Perancangan Sistem 10](#_TOC_250005)
     1. [Use Case Diagram 10](#_TOC_250004)
     2. Scanario Use Case 10
     3. [Activity Diagram 14](#_TOC_250003)
     4. [Squence Diagram 18](#_TOC_250002)
     5. [Class Diagram 21](#_TOC_250001)
  2. [Pembahasan Hasil 21](#_TOC_250000)

# PENDAHULUAN

# Latar Belakang

Dengan banyaknya masyarakat yang ingin mencari hiburan yaitu salah satunya dengan cara menonton bioskop, maka dari itu dibutuhkan tiket untuk menonton film yang ada dibioskop. Dengan adanya bioskop tersebut masyarakat bisa terhibur dan juga dengan adanya bioskop ini bisa menghilangkan depresi, serta sarana relaksi yang cukup ampuh dalam menghadapi rasa tekanan saat dalam bekerja maupun aktifitas lainya. Kelebihan dari bioskop ini sensasi dalam menoton film yang dimuat dalam layar lebar bisa menguji adrenalin saat menonton film.

Dan seiring banyak nya yang suka menonton film di bioskop, petugas tiket kesulitan dalam antrian tiket yang begitu banyak dan memakan waktu yang lama.contohnya seperti petugas mendata orang yang mau pesan tiket tersebut. Oleh karena itu Salah satu yang menjadi masalah yaitu pada pemesanan tiket yang offline maka dari itu kami menawarkan solusi agar pemesanan tiket lebih mudah dilakukan secara online. Dengan cara itu lebih memudahkan customer dalam pembelian tiket dan memilih tempat duduk dengan mudah dan cepat. Pemesanan tiket secara online dibutuhkan sebuah software atau aplikasi untuk membantu mempermudah dalam pemesanan. Dengan adanya software ini pengunjung atau pelanggan tidak lagi mengantri diloker untuk memesan tiket.

Dalam mengatur semua kebutuhan yang diperlukan dalam hal penyedia jasa yang di lakukan dalam pembuatan aplikasi sebagai penyajian nya .pembuatan aplikasi bisa sangat membatu dalam hal pelayan secara pirtual dengan penyajian yang cukup komplek dan lebih akurat.dalam hal pembuatan aplikasi yang bisa mengurangi kesalahan teknis yang dilakukan secara manual. Berdasarkan hal yang sudah di uraikan di atas bahwa dalam pembuatan aplikasi yang dibutuhkan oleh konsumen bisa membatu dalam hal kebutuhan dalam membatu dalam pekerjaan nya.

# Masalah Penelitian

Sesuai dengan permasalahan yang telah disampaikan diatas maka perumusanya yang kami ajukan yaitu : “Bagaimana membuat aplikasi E- tiketing yang dapat mengurangi kerumunan serta mudah dan praktis dalam memesan ”

# Tujuan Penelitian

Tujuan yang akan dicapai pada pembuatan aplikasi ini yaitu diantaranya seperti : Membuat Aplikasi yang membantu dalam pemesanan tiket agar dengan adanya aplikasi tersebut bisa terhindarnya antrian yang begitu banyak dan juga agar cepat dalam pemesanan tiket. Dengan adanya aplikasi ini pelanggan tidak perlu dating untuk memesan tiket ke lobi.

# Pertanyaan Penelitian

Adapun pertanyaan terkait dengan penelitian ini adalah sebagai berikut:

* + 1. Apa yang bisa dibantu dalam permasalahan tentang pemesanan?
    2. Bagaimana sistem yang dibuat menjamin pemsalahan selesai?
    3. Apa saja keuntungan dari sistem yang di terapkan?

# Cakupan Penelitian

Berdasarkan masalah yang telah ditemukan kami selaku pembuat sistem bermaksud untuk memberikan solusi dari masalah yang telah dihadapi.dengan pembuatan sistem yang menggunakan metode RUP sebagai salah satu metode yang kami kembangkan .dengan menerapkan pemecahan masalah dan bertanggung jawab dalam kualitas sistem yang kami buat.

# Manfaat Penelitian

Bisa menambah wawasan bagi kami selaku peneliti agar dalam pembuatan sistem kedepan nya dapat lebih baik lagi dari segi apapun.

Dengan adanya penelitian ini bisa membuat suatu sarana ilmu yang bisa digunakan bagi yang membutuhkan dalam sebuah penelitian yang bersangkutan dengan proyek yang sama.

# TINJAUAN PUSTAKA

# Definisi Operasional

Dengan adanya permasalah pada bidang pemesanan kami berinisiatip membuat sebuah sistem yang berbasis web yang digunakan untuk membantu pekerjaan yang sedang dilakukan.

# HTML

Merupakan sebuah standar yang digunakan secara luas untuk dapat menampilkan halaman web.dengan menggunakan tag- tag tertentu untuk menyatakan kode –kode yang harus di tafsirkan oelh browser agar halaman tersebut bisa ditampilkan di web browser.

# CSS

Adalah sebuah singkatan dari cascanding style sheets yang berguna untuk menyederhanakan prosess pembuata wesite dengan mengatur elemen yang tertulis di Bahasa markup. Css dipakai untuk mendisain halaman depan atau tampilan website.

# BOOTSTARP

Merupakan sebuah ribrary framework css yang telah dibuat khusus untung mengembangkan fond end dari sebuah website.bootstarp juga dikenal sebagai salah satu framework css,html,javascript yang begitu populer dikalangan website devoleper atau pengembang website.

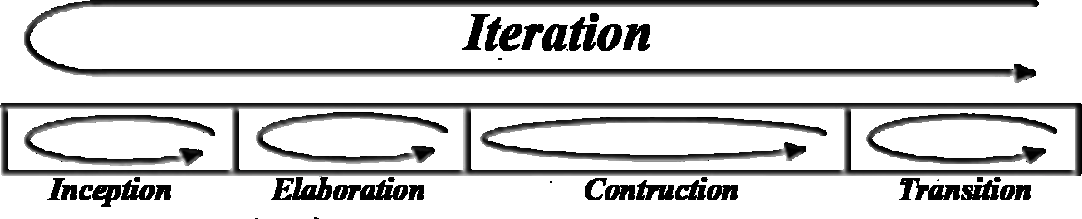
# PHP

Adalah Bahasa scripting server-ide,bahsa pemograman yang digunakan untuk mengembangkan situs web statis atau situs web dinamis atau aplikasi web.php mempunyai singkatan yaitu hypertext pre-processor,yang sebelumnya disebut personal home pages. Php merupan scripting server- side yang membuat Bahasa program yang akanh dijalankan /di proses oleh server.

# Relational Unified Process (RUP)

Merupakan sebuah metode yang digunakan dalam pengembangan perangkat lunak secara *iterative* (berulang) serta dengan tahapan proses

menaik dimana setiap iterasi memperbaiki iterasi berikutnya pada setiap tahapannya (Sukamto & Salahuddin, 2018). Berikut merupakan tahapan *RUP :*



*Gambar 2.1 Tahapan Rational Unified Process*

Penjelasan untuk setiap tahapan adalah sebagai berikut (Sukamto & Salahuddin, 2018):

1. *Inception* (Permulaan)

Aktivitas tahap pertama digunakan untuk mengumpulkan kebutuhan sistem dan pemodelan bisnis yang dimulai dari pendefinisian ruang lingkup proyek serta membangun kasus biisnis yang dibutuhkan. Hasiil yang diharapkan adalah umpan balik dari pendefinisian ruang lingkup, perikraan biaya, jadwal penentuan skala priioritas, risiko dan proses pengembangan. Kebutuhan yang sejalan dengan kasus primer dan dapat dimengerti. Terakhir membangun garis besar untuk perbandingan rencana aktual dan perencanaan. Pada tahap ini memetakan kebutuhan sistem, dengan menggunakan *Work Breakdown Structure* (*WBS*).

1. *Elaboration* (Perluasan/Perencanaan)

Menggambarkan rencana artsitektur sistem. Dilakukan pendeteksian terhadap arsitektur, resiko yang terjadi bagaimana. Lebih pada analiisis dan desaiin sistem yang menghasilkan pada purwarupa (*prototype*). Hasilnya berupa *use case* yang didalamnya terdefinisi kasus dan aktor yang terlibat dalam sistem. Telah membuat proses bisnis, daftar risiko yang telah diperbaiki, dan purwarupa (*prototype*) yang dapat di tunjukkan/dijalankan.

1. *Construction* (Konstruksi)

Dilakukan pengembangan komponen dan fitur. Implementasi yang fokus pada pembuatan program yang nantinya menghasilkans sebuah perangkat lunak yang menjadii syarat batas kemampuan operasional awal.

Selain itu tahapan ini juga memiliki tujuan untuk meminimalisir biaya pengembangan dengan mengoptimalkan sumber daya dan menghindari pembongkaran dan pengerjaan ulang, mencapai kualitas yang memadai secepat praktis, mecapai versi yang berguna (alfa, beta, dan rilis pengujian lainnya) secepat muungkin, menyelesaikan analisis, desain, pengembangan dan pengujian semua fungsionalitas yang diperlukan dan mendeskripsikan kasus penggunaan yang tersisa dan persyaratan lainnya, menyempurnakan desain, menyelesaikan implementasi dan menguji perangkat peranngkat lunak;

1. *Transition* (Transisi)

Dilakukan instalasi sistem agar dapat dimengerti oleh user. Hasilnya berupaa perangkat lunak yang menjadi syarat tonggak kemampuan operasional awal dan juga dilakukan latihan untu pengguna, sistem dipelihara, dan diuji agar memenuhi harapan pengguna.

## Unified Modeling Languange

UML adalah suatu [bahasa](https://id.wikipedia.org/wiki/Bahasa) yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu [sistem](https://id.wikipedia.org/wiki/Sistem_informasi) [informasi.](https://id.wikipedia.org/wiki/Sistem_informasi) UML dikembangkan sebagai suatu alat untuk [analisis](https://id.wikipedia.org/wiki/Analisis) dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi Penggunaan UML dalam [industri](https://id.wikipedia.org/wiki/Industri) terus meningkat Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri [peranti lunak](https://id.wikipedia.org/wiki/Peranti_lunak) dan pengembangan sistem.

## Use Case

*Use case Diagram* mendeskripsikan sebuah interaksi aktor dengan sistem yang akan dibuat, *Use case* dimanfaatkan untuk mengetahui fungsi apa saja dan siapa saja yang berhak menggunakan fungsi-fungsi yang terdapat pada sistem tersebut. Adapun simbol yang ada pada diagram *use case* adalah sebagai berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Deskripsi** |
| 1. |  | *Use Case* | Sistem telah menyediakan fasilitas untuk |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | memaparkan aktor yang saling bertukar  pesan. |
| 2. |  | *Actor* | Menjelaskan himpunan peran orang,  proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. |
| 3. |  | *Association* | Menjelaskan komunikasi antara *actor* dan  *use case.* |
| 4. | <<extend>> | *Extend* | Menjelaskan bahwa *use case* target meyebarkan perilaku dari *use case*  sumber dan dapat berdiri sendiri. |
| 5. |  | *Generalization* | Hubungan generalisasi (umum-khusus)  antara dua atau lebih *use case.* |
| 6. | <<include>> | *Include* | Menjelaskan sebuah *use case* tambahan  bergantung kepada *use case* utama. |

* + 1. ***Activity* Diagram**

Diagram *activity* diagram menjelaskan *workflow* atau aktifitas proses bisnis dari sebuah sistem, pada tahapan ini yang perlu diperhatikan adalah diagram aktifitas ini hanya menjelaskan aktifitas sistem bukan apa yang dilakukan aktor. Berikut adalah simbol-simbol yang ada pada diagram aktifitas:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Deskripsi** |
| 1. |  | *Initial Node* | Status awal aktivitas dari sebuah sistem. |
| 2. |  | *Activity* | Aktivitas (*activity*) dari sebuah sistem. |
| 3. |  | *Decision* | Menjelaskan beberapa langkah pada  aliran kerja. |
| 4. |  | *Join* | Menggabungkan dari beberapa pilihan  aktivitas. |
| 5. |  | *Activity Final*  *Node* | Menjelaskan status akhir (*activity final*  *node*) dari aktivitas sistem. |

|  |  |  |  |
| --- | --- | --- | --- |
| 6. |  | *Swimlane* | Memecahkan kumpulan bisnis yang bertanggung jawab untuk melaksanakan aktivitas tertentu yang terjadi. |

* + 1. ***Squence* Diagram**

Diagram sequen menjelaskan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada diagram sequen:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Simbol** | | **Nama** | | **Keterangan** | | |
| 1. |  | | *Actor* (Aktor) | | Menjelaskan himpunan peran orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan  dibuat. | | |
| 2. |  |  | *Lifeline* (Garis  Hidup) | | Mendeskripsikan  objek. | kehidupan | suatu |
|  | |
| 3. |  | | *Object*  (Objek) | | Mendeskripsikan objek yang saling  berinteraksi pesan. | | |
| 4. |  | | Waktu Aktif | | Menjelaskan bahwa objek sedang aktif  atau berinteraksi. | | |
| 5. | <<create>> | | Pesan  *Create* | Tipe | Memaparkan bahwa suatu objek  sedang membuat objek lain. | | |
| 6. | 1:  nama\_metode() | | Pesan  *Call* | Tipe | Suatu objek sedang memanggil operasi  yang terdapat pada dirinya sendiri atau objek lain. | | |
| 7. | 1: masukan | | Pesan  *Send* | Tipe | Mendeskripsikan suatu objek mengirim  informasi ke objek lain. | | |
| 8. | 1: keluaran | | Pesan  *Return* | Tipe | Mendeskripsikan suatu objek yang  telah berjalan menghasilkan keluaran ke objek tertentu. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 9. | <<destroy>> | Pesan Tipe  *Destroy* | Memaparkan suatu objek mengakhiri objek lain. |

* + 1. ***Class* Diagram**

Diagram kelas menjelaskan struktur sistem dari segi pendefinisian kelas- kelas untuk sistem yang akan dibuat. Kelas memiliki 3 bagian utama, yaitu *attribute*, *operation*, dan *name*. Susunan struktur kelas dinyatakan baik pada diagram kelas yaitu harus memiliki jenis-jenis kelas seperrti berikut berikut:

* + - 1. Kelas *main.* Yaitu kelas yang pertama kali dieksekusi ketika sistem dijalankan.
      2. Kelas *Interface*. Kelas yang mengatur dan menggambarkan tampilan ke pemakai. Biasanya juga disebut kelas *boundaries.*
      3. Kelas yang diambil dari pendefinisian *use case*. Merupakan kelas yang menangani fungsi – fungsi yang harus ada dan diambil dari pendefinisian *use case*.
      4. Kelas Entitas. Merupakan kelas yang digunakan untuk membalut data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke barisan data.

Adapun simbol yang ada pada diagram kelas tampak pada Tabel 2.9 berikut:

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Simbol** | **Nama** | **Deskripsi** |
| 1. |  | *Class* | Kelas pada struktur sistem |
| 2. |  | *Interface* | Sama dengan konsep *interface* dalam  pemrograman berorientasi objek |
| 3. |  | *Association* | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan  *multiplicity* |
| 4. |  | *Directed* | Relasi antarkelas dengan makna kelas |



|  |  |  |  |
| --- | --- | --- | --- |
|  |  | *Association* | yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai  dengan *multiplicity* |
| 5. |  | *Generalization* | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-  khusus) |
| 6. |  | *Dependency* | Relasi antarkelas dengan makna  ketergantungan antarkelas |
| 7. |  | *Aggregation* | Relasi antarkelas dengan makna  semua-bagian (*whole*-*part*) |

# HASIL DAN PEMBAHASAN

# 3.1 Work Breakdown Structure

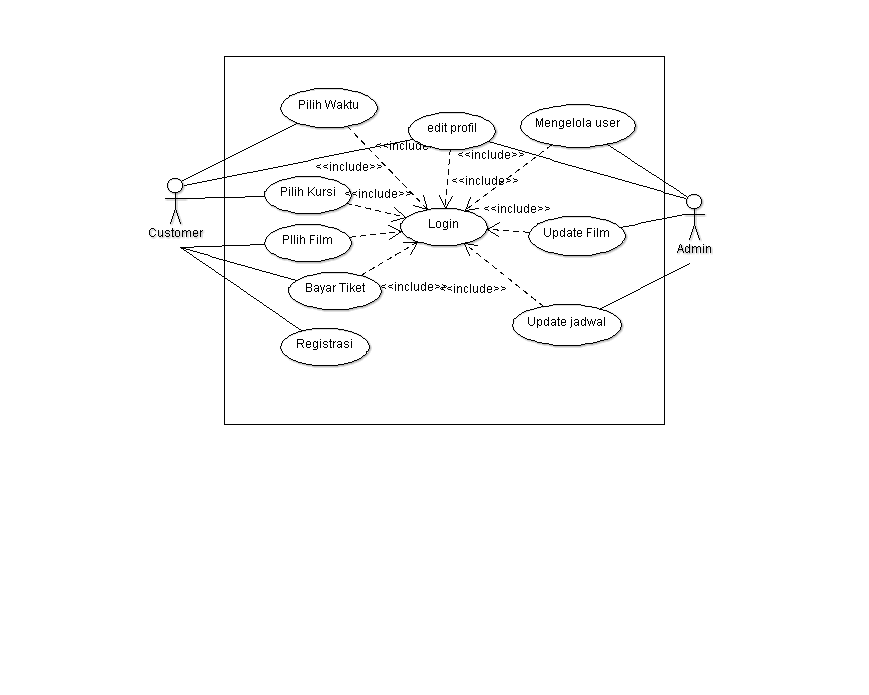
# Urutan aktivitas dari perencanaan hingga pengujian aplikasi,dituangkan menggunakan Work Breakdown Structure (WBS).Aktivitas yang terakhir adalah menyimpulkan seluruh aktivitas yang dilakukan penelitian.berikut ini merupakan gambaran WBS berdasarkan tujuan dan metode penelitian.

# 

# 3.2Perancangan Sistem

Pada perancangan aplikasi ini mencakup pembuatan diagram yang menggambarkan dan membantu menjelaskan tahapan-tahapan didalam aplikasi. Pemodelan yang digunakan adalah *Unified Modeling Language (UML)*. Setiap tahapan digambarkan dengan diagram yang berfungsi untuk membantu menjelaskan tahapan didalam aplikasi, diagram yang dibuat diantaranya *use case, activity, sequence* dan *class diagram*. Setiap iterasi yang menaik ini memperbaiki iterasi berikutnya.

**3.2.1 Use Case Diagram**

****

**3.2.2 Skenario Use case**

1. **Skenario Login**

Nama Usecase : Login

Tujuan : Untuk masuk kedalam system pemesanan tiket

Aktor : Admin dan Customer

Type : Primary

|  |  |  |
| --- | --- | --- |
| **Aktor** | | **Sistem** |
| **Skenario Normal** | | |
| 1. Mengakses Web Pemesanan  Tiket Bioskop | |  |
|  | | 2. Menampilkan halaman Login |
| 3. Memasukkan User Name dan  Password | |  |
|  | | 4. Memvalidasi data akun  pengguna |
|  | | 5. Jika berhasil, masuk kehalaman  utama |
| Skenario Alternatif | | | |
| 1. Memasukkan User Name dan  Password | |  | |
|  | | 2. Memvalidasi data dan menampilkan notifikasi *login*  gagal | |
|  | | 3. Kembali ke halaman login | |

1. **Skenario Registrasi**

Nama Usecase : Registrasi Akun

Tujuan : Untuk mendaftar akun

Aktor : Customer

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Masuk ke Website E-TIX |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih Registrasi |  |
|  | 4. Menampilkan Halaman Registrasi Customer |
| 5. Memasukan Identitas diri |  |
| 6. Klik Register |  |
|  | 7. Menampilkan halaman utama |

1. **Skenario Pilih Film**

Nama Usecase : Pilih Film

Tujuan : Untuk memilih Film

Aktor : Customer

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Melakukan Login sistem |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih menu Film |  |
|  | 4. Menampilkan Halaman Daftar  Film |
| 5. Memilih Film |  |
|  | 6. Menampilkan halaman Film  yang dipilih |

1. **Skenario Pilih Waktu**

Nama Usecase : Pilih Waktu

Tujuan : Untuk memilih waktu tayang Film

Aktor : Customer

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Memilih Film |  |
|  | 2. Menampilkan halaman Film  yang dipilih |
| 3. Pilih Waktu tayang film |  |
|  | 4. Menampilkan halaman waktu  film yang dipilih |

1. **Skenario Pilih Kursi**

Nama Usecase : Pilih Kursi

Tujuan : Untuk memilih tempat duduk saat nonton

Aktor : Customer

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Pilih Waktu film |  |
|  | 2. Menampilkan halaman Daftar  kursi |
| 3. Memilih no kursi |  |
|  | 4. Menampilkan halaman no  kursi yang dipilih |

1. **Skenario Bayar Tiket**

Nama Usecase : Bayar Tiket

Tujuan : Untuk melakukan pembayaran

Aktor : Customer

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
|  | 1. Menampilkan kursi yang telah  dipilih |
| 2. Memilih tombol Ringkasan Order |  |
|  | 3. Menampilkan Halaman  Ringkasan Order |
| 4. Memilih tombol Bayar Sekarang |  |
|  | 5. Memvalidasi dan menyimpan  data |

1. **Skenario Edit Profile**

Nama Usecase : Edit Profil

Tujuan : Untuk mengedit profile user

Aktor : Customer & Admin

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Melakukan Login sistem |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih menu User |  |
|  | 4. Menampilkan Halaman  User |
| 5. Klik menu Profile |  |
|  | 6. Menampilkan Halaman Profile |
| 7. Mengedit Informasi mengenai user |  |
| 8. Klik tombol Update |  |
|  | 9. Menampilkan Halaman Profile yang telah di update |

1. **Skenario Mengelola User**

Nama Usecase : Mengelola User

Tujuan : Untuk mengelola data customer

Aktor : Admin

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Melakukan Login sistem |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih menu User |  |
|  | 4. Menampilkan Halaman  User |
| 5. Melihat Data User |  |

1. **Skenario Update Film**

Nama Usecase : Update Film

Tujuan : Untuk update informasi mengenai Film

Aktor : Admin

Type : Primary

|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Melakukan Login sistem |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih menu Update Film |  |
|  | 4. Menampilkan Halaman Film |
| 5. Mengupdate film |  |
|  | 6. Menampilkan Halaman Film  yang sudah d update |

1. **Skenario Update Jadwal**

Nama Usecase : Update Jadwal

Tujuan : Untuk update informasi mengenai Jadwal

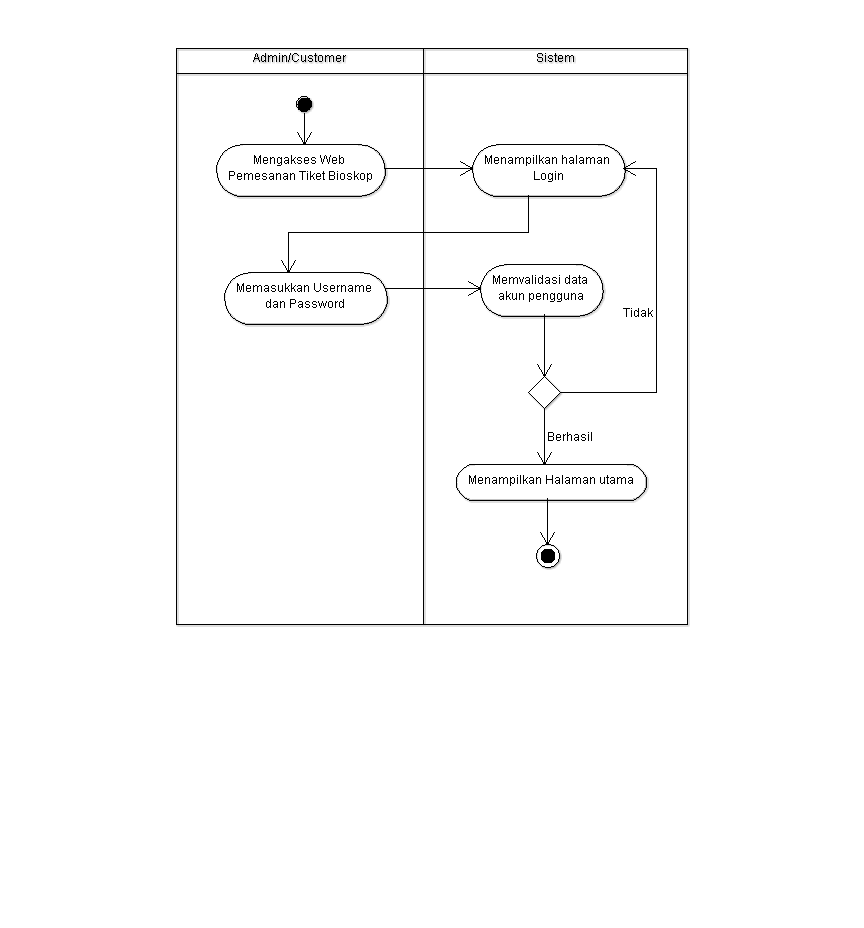
Aktor : Admin

Type : Primary

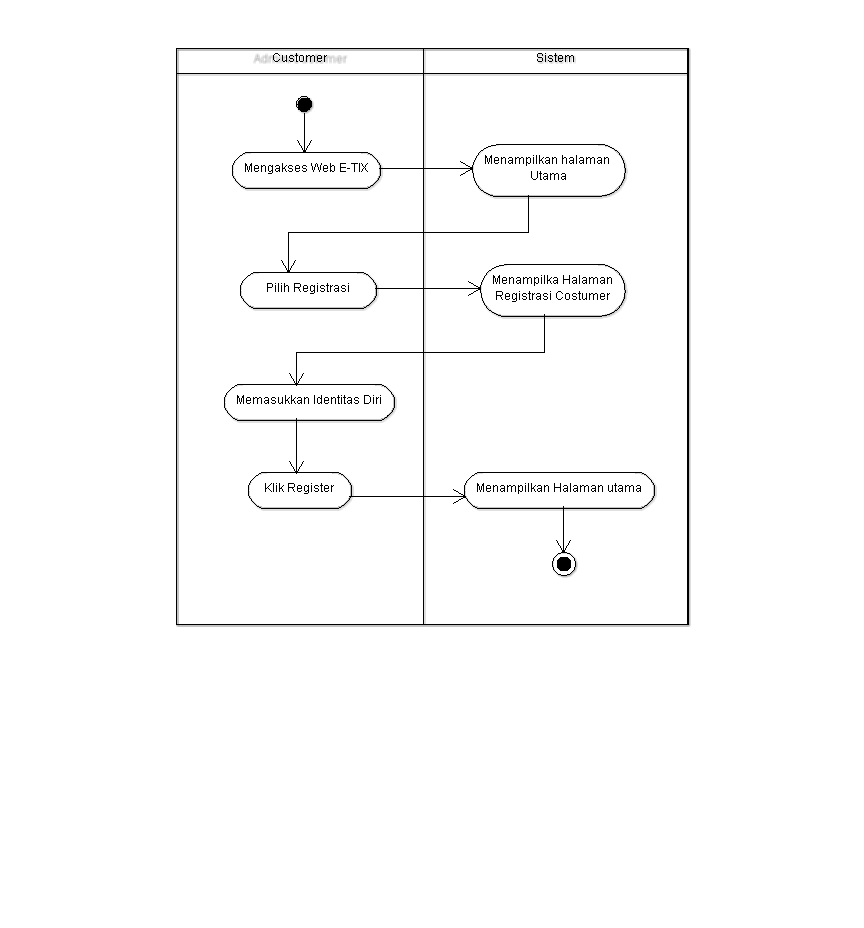
|  |  |
| --- | --- |
| **Aktor** | **Sistem** |
| **Skenario Normal** | |
| 1. Melakukan Login sistem |  |
|  | 2. Menampilkan halaman utama |
| 3. Pilih menu Update Jadwal |  |
|  | 4. Menampilkan Halaman Update Jadwal Tayang |
| 5. Mengupdate jadwal Tayang Film |  |
|  | 6. Menampilkan Halaman Film  yang sudah d update |

**3.2.3 Activity Diagram**

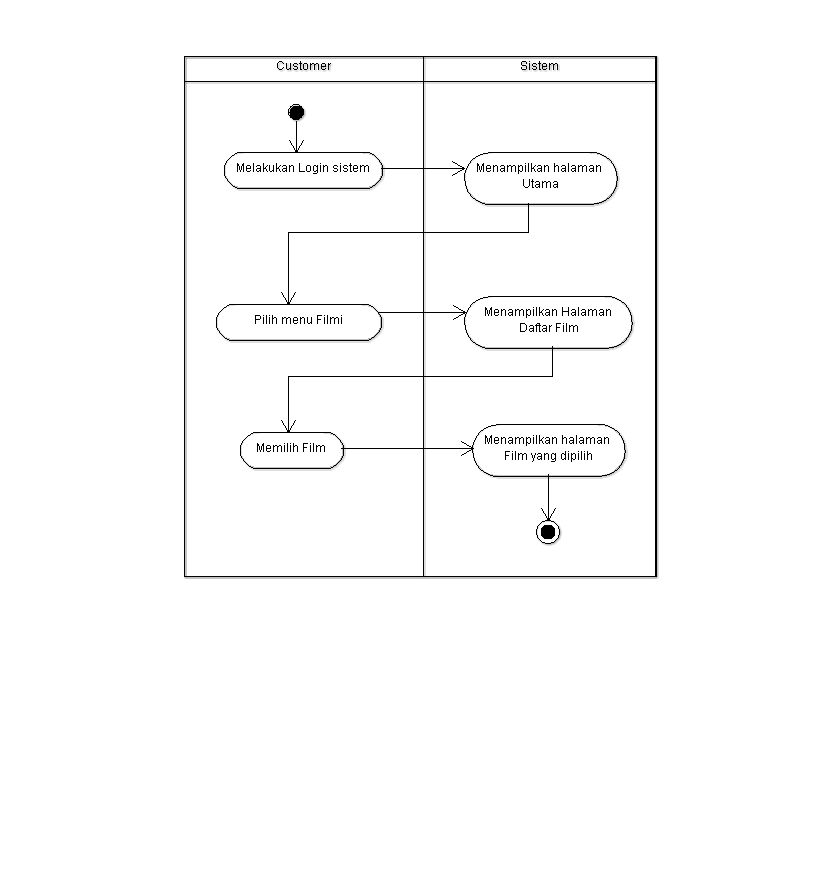
1. **Activity Login**

****

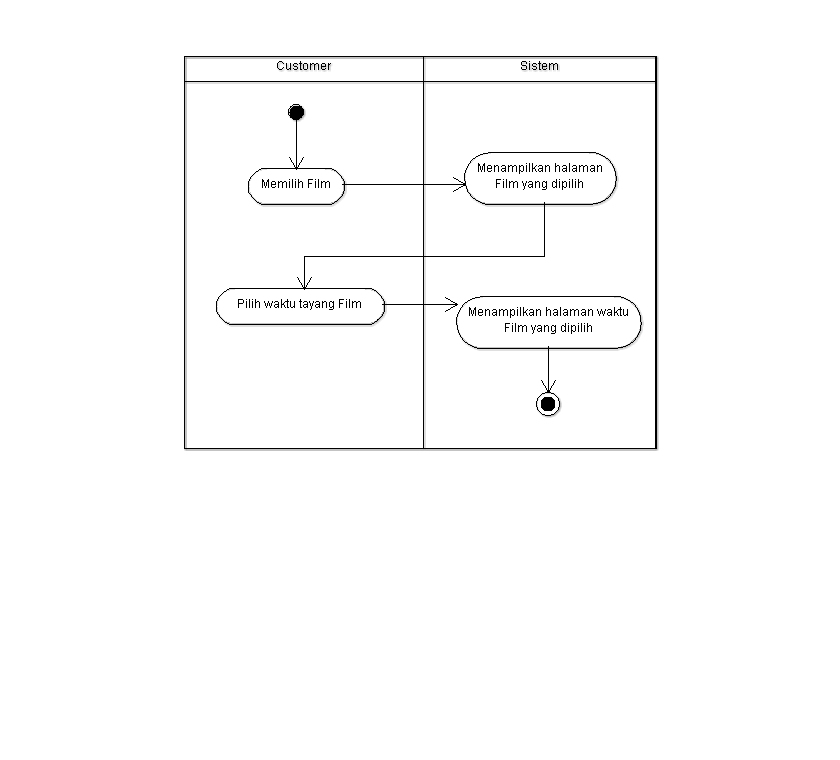
1. **Activity Registrasi**

****

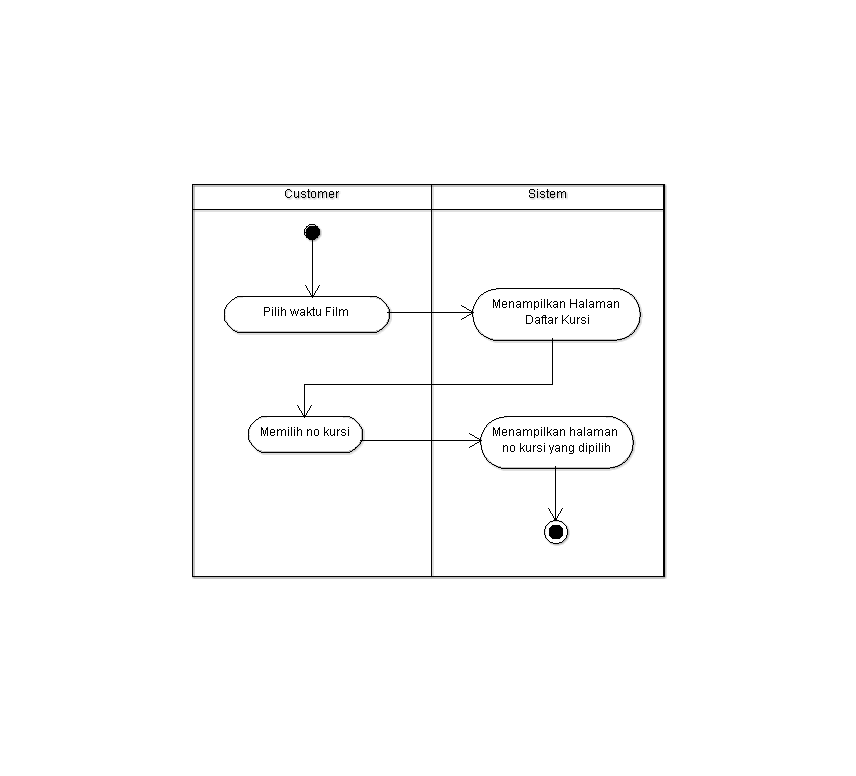
1. **Activity Pilih Film**

****

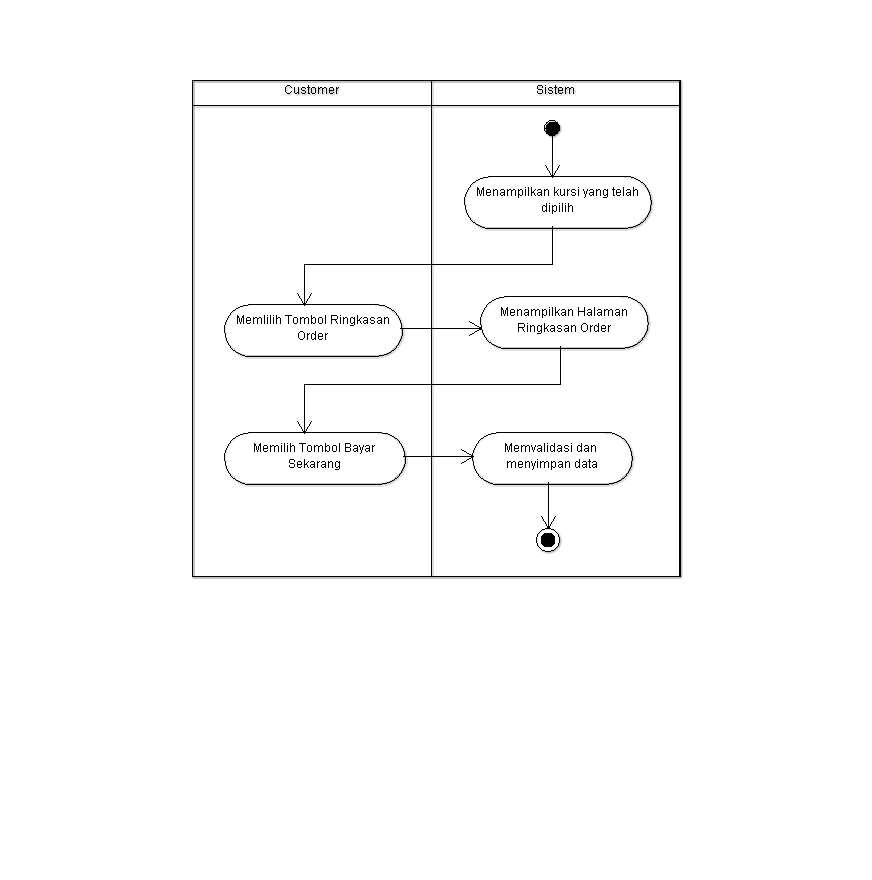
1. **Activity Pilih Waktu**

****

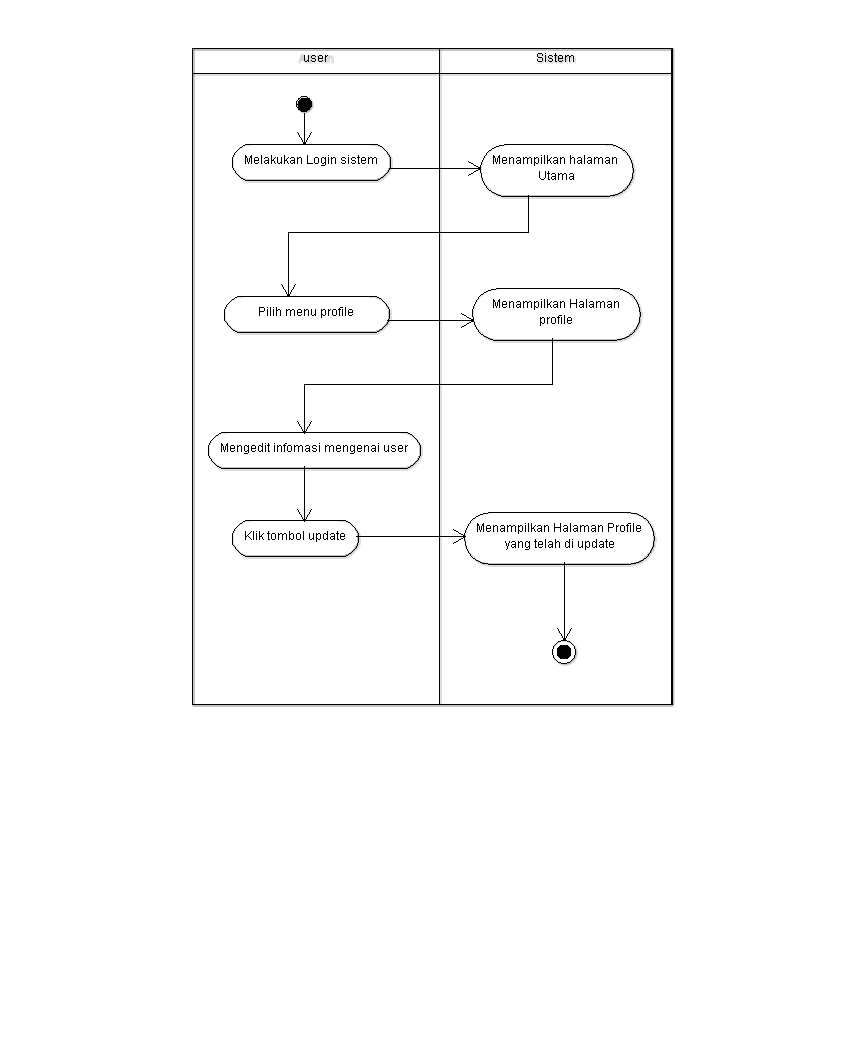
1. **Activity Pilih Kursi**

****

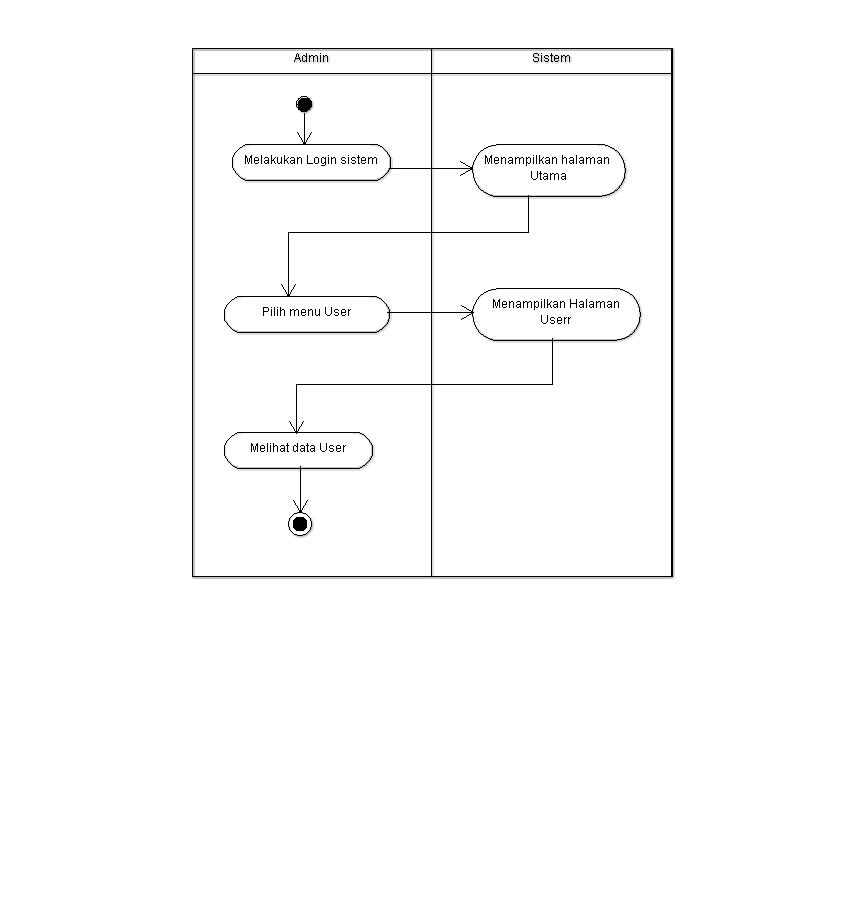
1. **Activity Bayar Tiket**

****

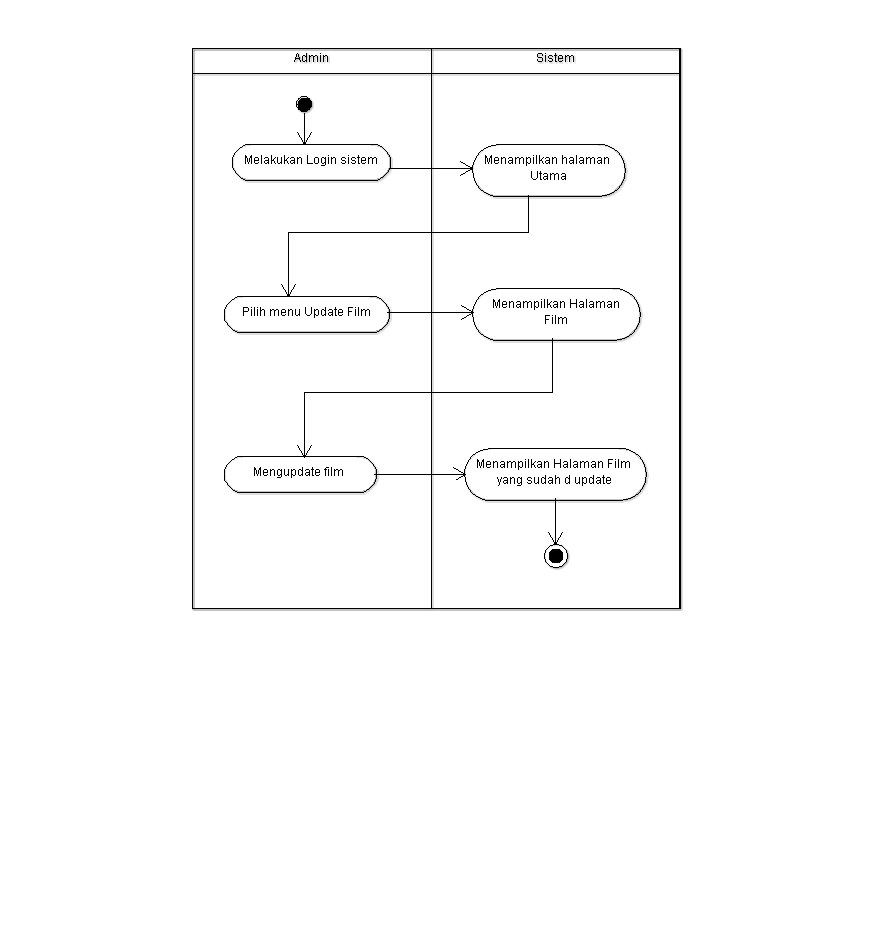
1. **Activity Edit Profile**

****

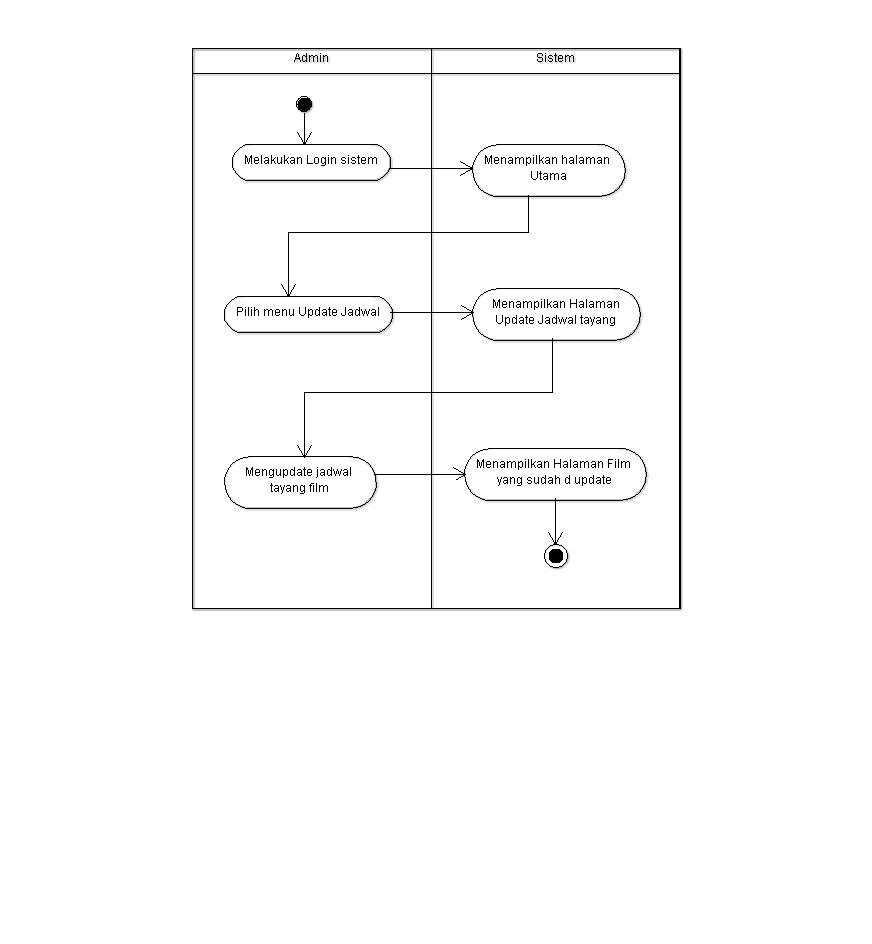
1. **Activity Mengelola User**

****

1. **Activity Update Film**

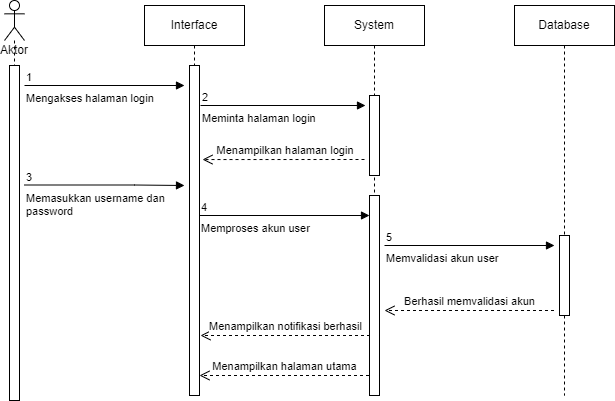
****

1. **Activity Update Jadwal**

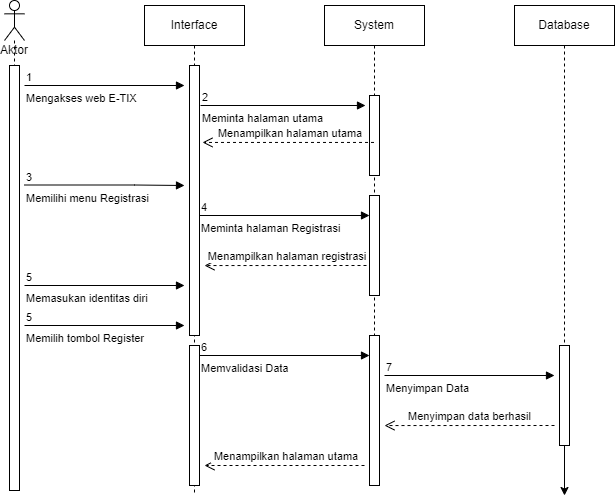
****

**3.2.4 Squence Diagram**

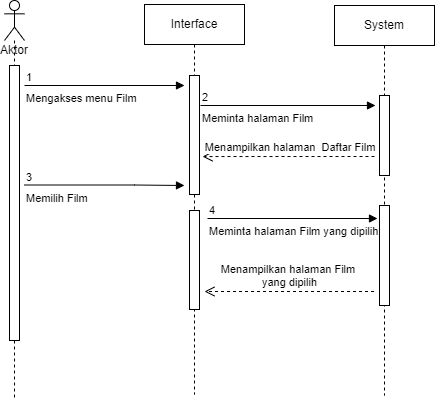
1. **Squence Login**

****

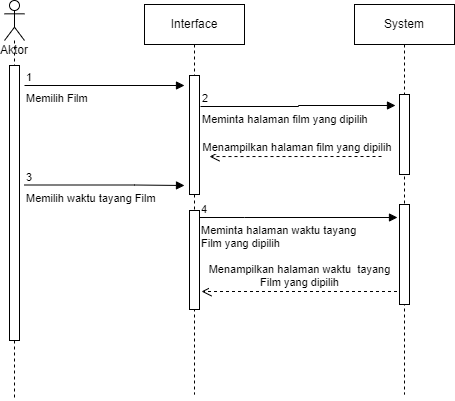
1. **Squence Registrasi**

****

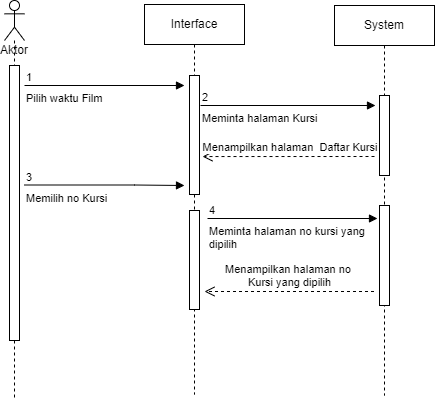
1. **Squence Pilih Film**

****

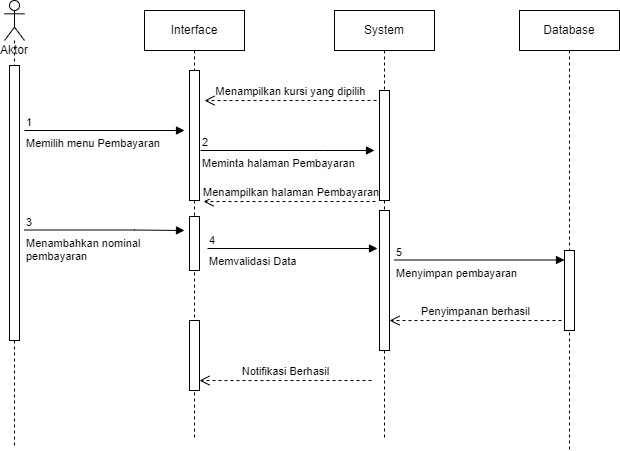
1. **Squence Pilih Waktu**

****

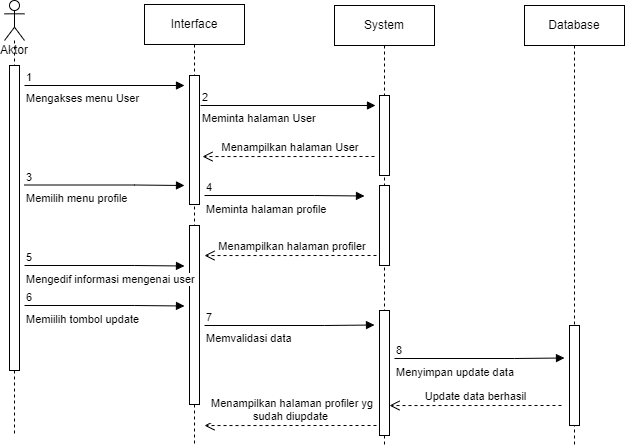
1. **Squence Pilih Kursi**

****

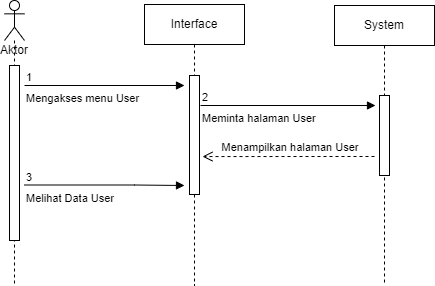
1. **Squence Bayar Tiket**

****

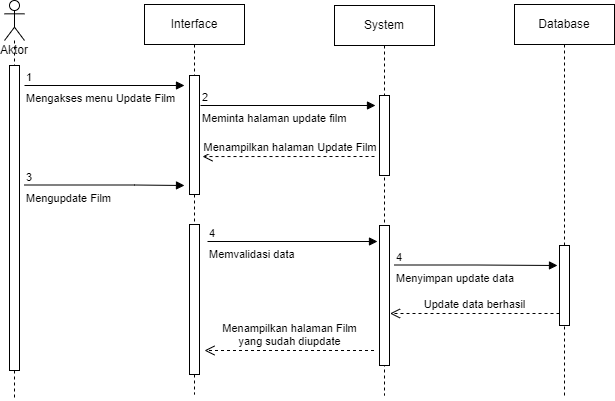
1. **Squence Edit Profile**

****

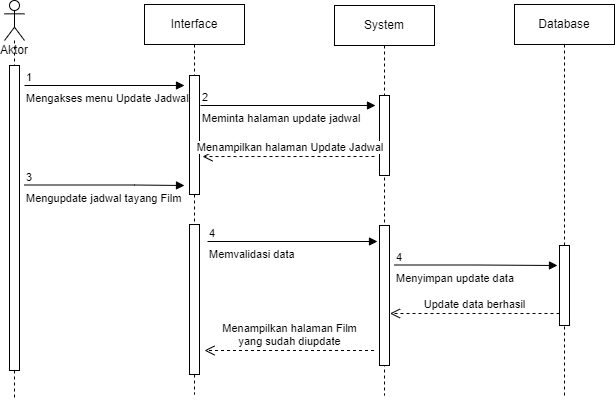
1. **Squence Mengelola User**

****

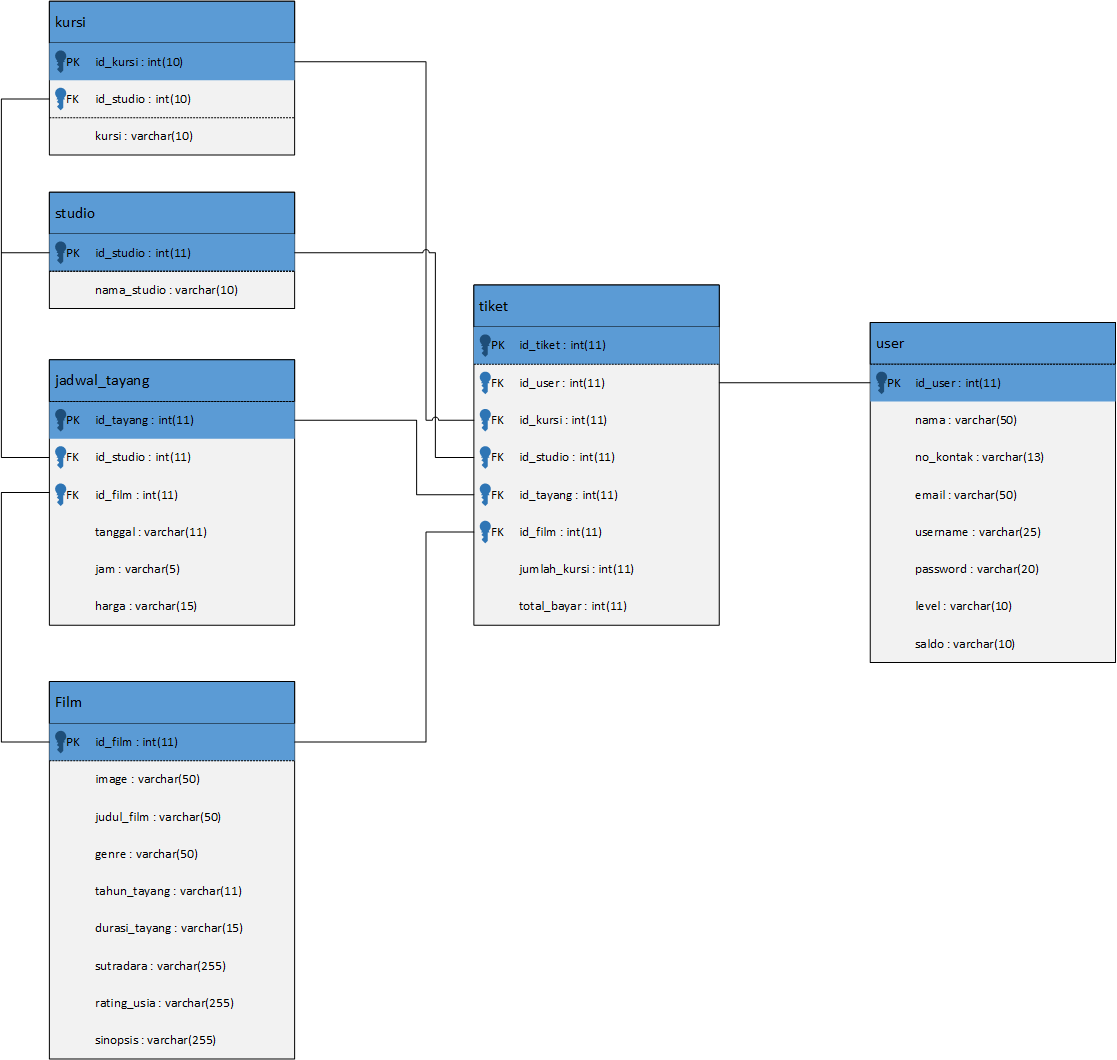
1. **Squence Update Film**

****

1. **Squence Update Jadwal**

****

**3.2.5 Class Diagram**

****

# 3.3 Pembahasan Hasil

# Custamer

# 3.3.1 Tampilan awal aplikasi

# 

# 3.3.2 Halaman login dan register

# 

# 

# 3.3.3 Halaman masuk

# 

# 

# 3.3.4 Pemilihan Film dan Jam Tayang

# 

# 3.3.5 pemilihan kursi

# 

# 3.3.6 konfirmasi pemesanan

# 

# 3.3.7 Hasil Pemesan Tiket

# 

# Admin

# 3.3.8 Login Admin

# 

# 3.3.9 kelola data film

# 

# 3.3.10 Kelola Data User

# 

# 3.3.11 kelola jadwal tayang

# 

# 3.3.12 kelola data tambah user

# 

# 3.3.13 kelola data tambah film

# 

# 3.3.14 kelola data tambah jam tayang

# 

# 3.3.15 update film

# 

# 3.3.16 update jam tayang

# 

* 1. **KESIMPULAN DAN SARAN**

**4.1 Kesimpulan**

Berdasarkan hasil di atas bahwa perancangan sistem pemesanan tiket bioskop bisa di simpulkan sebagai berikut :

A. aplikasi yang dibuat bisa membantu dalam proses pemesanan tiket agar terhindar dari antrian dan pembeli tidak perlu datang langsung ke bioskop.

B. sistem yang dibangun sebagai sarana informasi bagi yang senang menonton di bioskop.

C. sistem ini bertujuan mempermudah dalam hal pemesanan.

**4.2 Saran**

Kami selaku pembuat sistem meminta saran dan keritik yang membangun agar kedepannya jauh lebih baik lagi dalam pembuatan nya.

**DAFTAR PUSTAKA**

Referensi:

[1] I. Oktaviani and A. T. Pamungkas, “BIOSKOP Intan Oktaviani , Adhe Try Pamungkas,” *J. Inf. Politek. Indonusa Surakarta*, vol. 3, no. 2442–7942, pp. 57–62, 2017.

[2] J. Prof, D. Surya, and S. No, “Aplikasi Pemesanan Tiket Bioskop Berbasis Mobile Timotius Witono , Raphael Susanto Jurusan Teknik Informatika , Fakultas Teknologi Informasi Universitas Kristen Maranatha,” no. 65, pp. 143–151.

[3] E. Hartati, “Sistem Pemesanan Dan Pembelian Tiket Bioskop Secara Online Dengan menggunakan Active Server Pages . Net Berbasis Web Dan Wireless Application Protocol,” *J. Ilm. STMIK GI MDP*, vol. 4, no. 3, pp. 19–28, 2008.

[4] J. Hutahaean and E. A. Purba, “Rancangan Bangun E-Ticket Bioskop Dengan Metode Waterfall Berbasis Web,” *Riau J. Comput. Sci.*, vol. 2, no. 2, pp. 51–58, 2016.

[5] A. D. Pinangkis and T. Andreansyah, “Rancang Bangun Aplikasi Pemesanan Tiket Pada Masa Pandemi Covid-19,” vol. 2, no. 3, pp. 1–13, 2022.

[6] Y. E. Achyani1, “Tugas Akhir,” *175.45.187.195*, p. 31124, 2018.

[7] O. Nurisma Putra and J. Fadhilah, “Pembuatan Aplikasi Pemesanan Tiket Bioskop Berbasis Web | Jurnal Wahana Informatika,” *J. Wahana Inform.*, vol. 1, no. 1, pp. 13–26, 2022.