

LAPORAN ANALISA
PROYEK SISTEM INFORMASI PEMBELIAN TIKET
BIOSKOP
BERBASIS DESKTOP

Oleh :

SOYID WAHYU DARMAWAN/1741720151

RIFAUL HILAL SHABRIANSYAH/1741720195



PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2019

Analisa

Menentukan Kebutuhan Sistem

Dalam pembuatan Aplikasi pemesanan tiket bioskop berbasis Desktop ini, bertujuan untuk membantu penonton yang khususnya mengalami kesulitan dalam menonton film di bioskop karena kendala yang diantaranya tiket yang terbilang mahal dan waktu penayangan film yang tayang di jam tertentu saja. Dan tujuan lainnya adalah membangun sistem pembelian tiket bioskop secara terkomputerisasi.

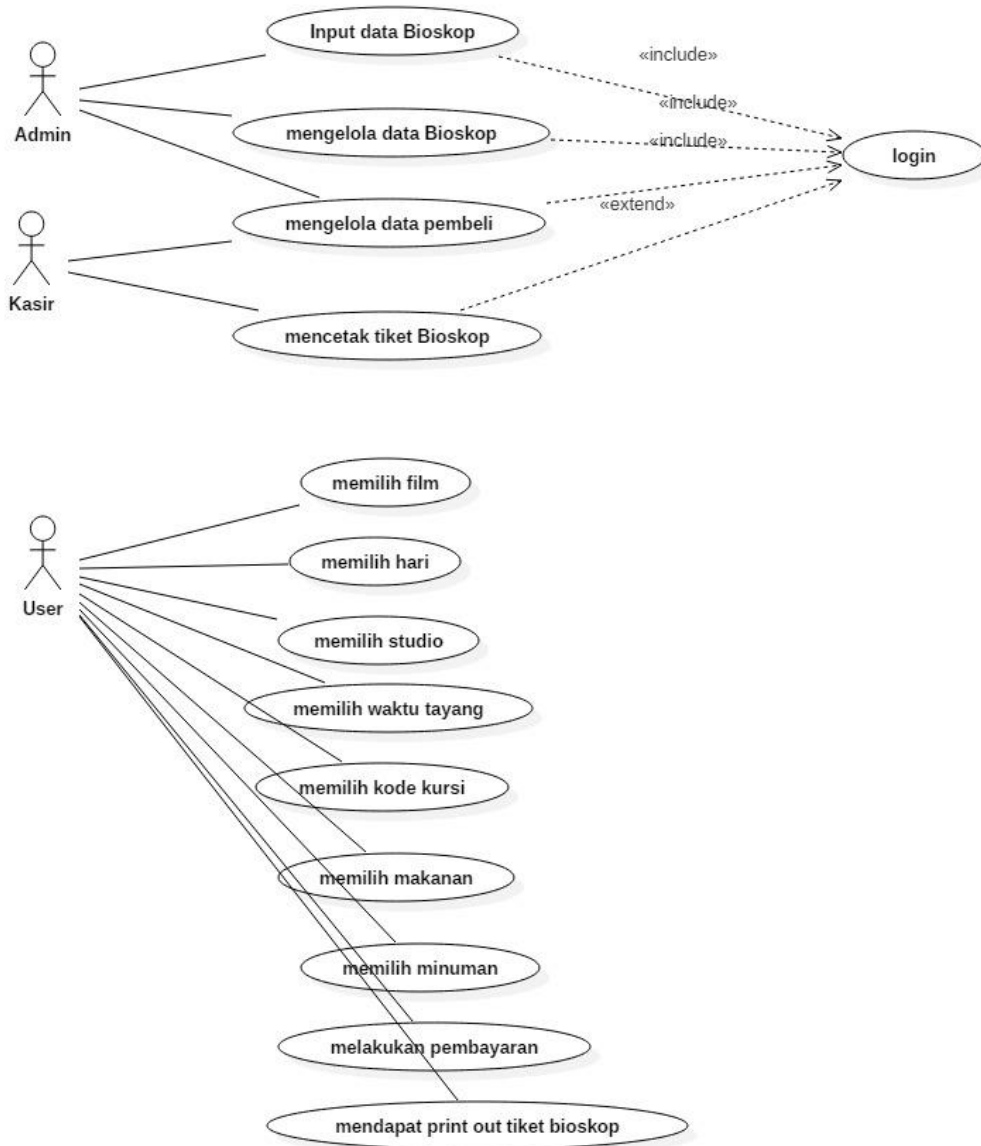
Metode yang dilakukan dalam melakukan pemesanan tiket bioskop ini adalah penggunaan user (pembeli). Kenapa penggunaan? Karena menurut kami dalam melakukan pemesanan tiket bioskop ini adalah untuk semua kalangan sehingga pembeli pasti menemui kendala dan kesulitan jika memesan tiket bioskop yang harga tiketnya mahal dan penayangannya di jam tertentu saja sehingga kita sebagai penonton merasa kesal karena kesulitan dalam menonton film di bioskop, dan khawatirnya segi keamanan dalam melakukan pembayaran

Oleh sebab itu, kami membuat itu menjadi metode yang cukup mudah dan aman. Dengan cara membuat Aplikasi pemesanan tiket bioskop berbasis Desktop. Yang bertujuan membantu penonton bioskop yang khususnya mengalami kesulitan dalam melakukan pemesanan tiket bioskop. Serta membantu penonton mengetahui keberadaan jadwal film penayangan di bioskop yang akan di pesan.

Dalam Aplikasi ini, tentunya diperlukan admin untuk selain melakukan pengolahan data serta membuat langkah pemrosesan data tentu saja membantu kasir dalam melakukan pengelolaan data penumpang. Pengolahan data ini terdiri dari input data, melakukan perubahan data berdasarkan jadwal, film, dan booking.

Menyusun kebutuhan sistem

USE CASE



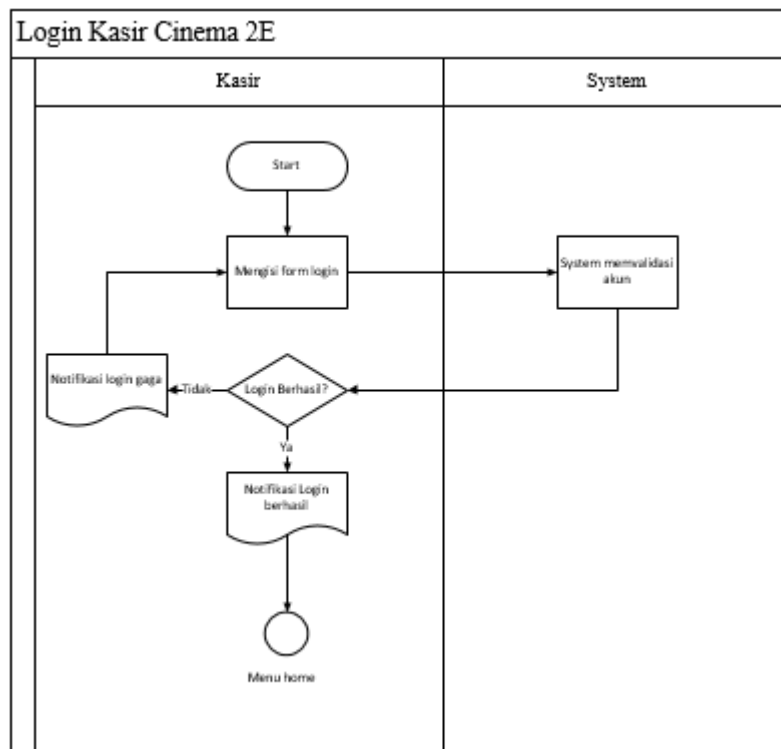
Berikut adalah use case dari aplikasi pemesanan tiket pesawat berbasis Desktop, di aplikasi ini kami menetapkan 3 (tiga) actor yaitu Admin, Kasir, dan Pembeli yang ketiganya memiliki tugas yang berbeda-beda. Untuk Admin, memiliki tugas input data bioskop, mengelola data bioskop, dan mengelola data pembeli. Maksud dari input data bioskop adalah admin dapat melakukan penambahan data jadwal dan film, studio, data makanan dan minuman yang terdapat di bioskop pada sistem ini, mengelola data penerbangan yaitu melakukan pengeditan pada data jika data mengalami kesalahan sebelum disimpan ke sistem, dan untuk mengelola data pembeli hanya sebatas pengecekan setelah pembeli berhasil membooking tiket.

Tugas kasir adalah mengelola data pembeli dan mencetak tiket bioskop. Maksud mengelola data pembeli sama seperti yang dilakukan admin pada tugas ini dan untuk mencetak tiket bioskop adalah setelah pembeli berhasil membooking tiket, maka kasir akan mencetak tiket untuk diserahkan ke pembeli sebelum film dimulai.

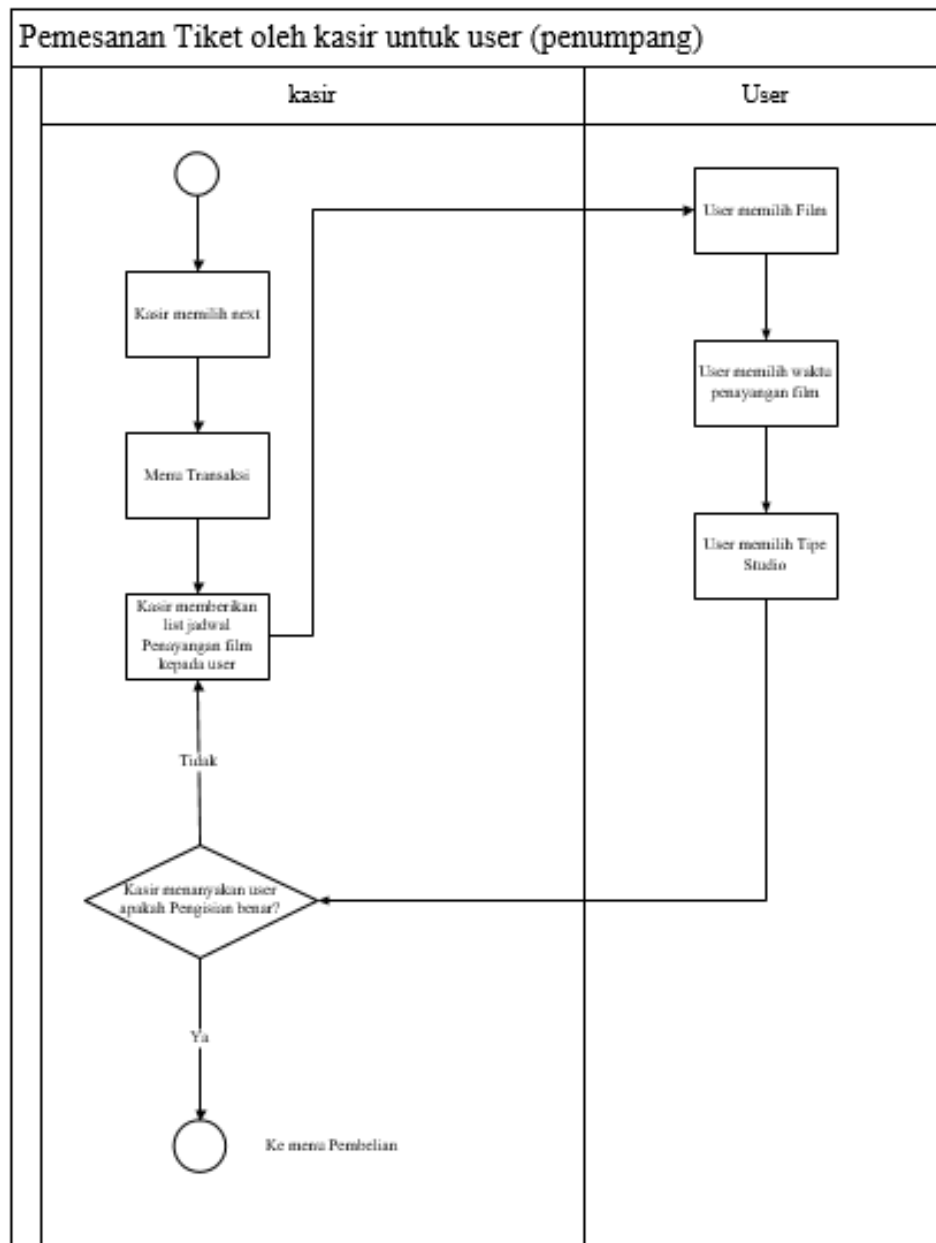
Dan tugas user adalah memilih film, memilih waktu tayang, memilih kode bangku, memilih menu makanan dan minuman, melakukan pembayaran, dan mendapat print out tiket bioskop. User pertama memilih jadwal bioskop berdasarkan judul film dan waktu tayang yang diinginkan pembeli, lalu memilih kode kursi yang diinginkan berdasar studio selanjutnya pembayaran dan yang terakhir adalah mendapat print out tiket bioskop yang sudah dicetak oleh kasir sebelumnya.

PROSES BISNIS

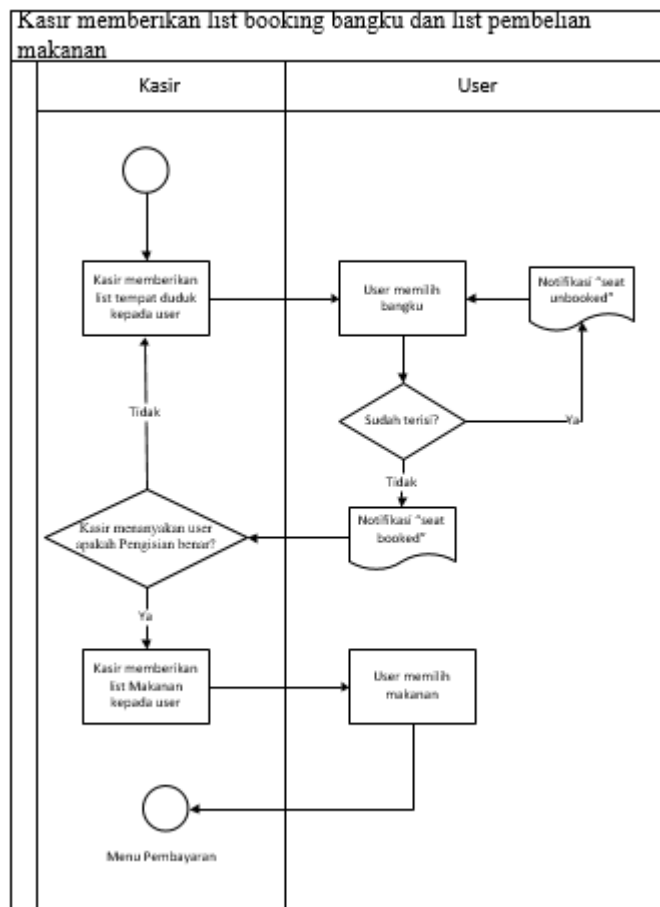
A. login kasir



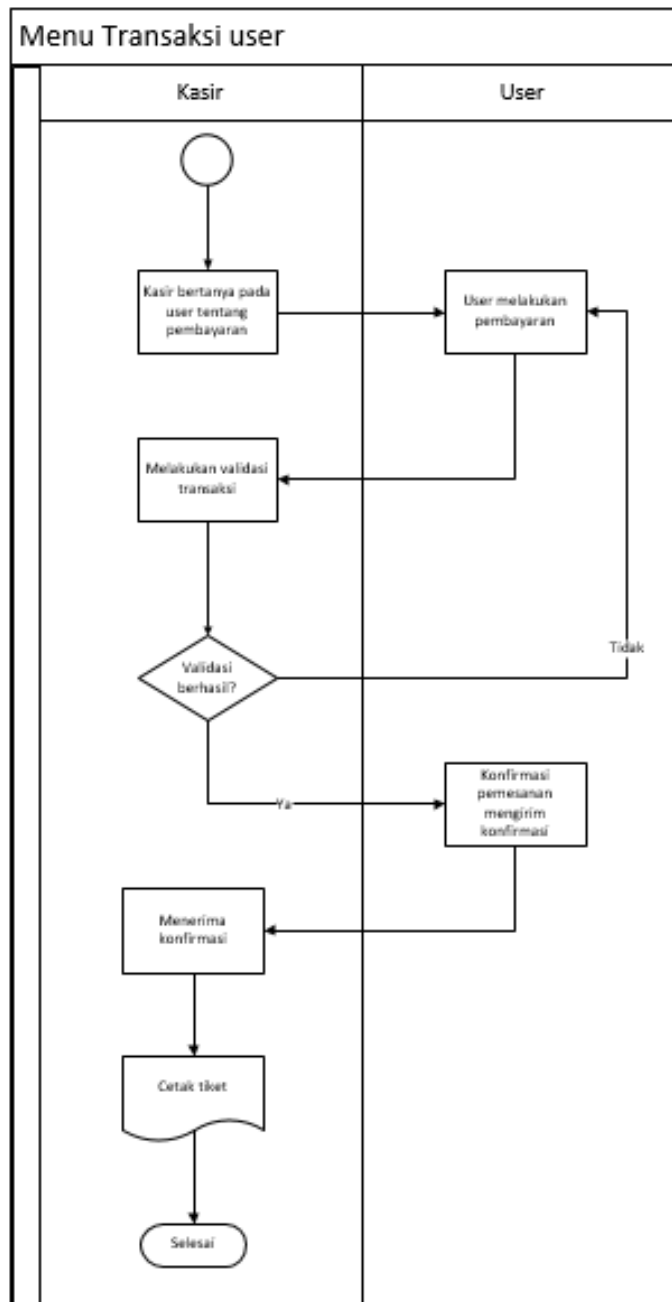
B. Pembelian tiket oleh kasir untuk user (penumpang)



C. Kasir memberikan list maskapai dan booking bangku



D. Menu transaksi user



Berikut adalah proses bisnis dari aplikasi pemesan tiket bioskop berbasis Desktop. Dari proses bisnis tersebut, terdapat langkah yang diantaranya validasi oleh kasir, pembelian tiket oleh kasir untuk penumpang, memberikan list film dan booking, serta menu transaksi yang sudah kami jelaskan sebelumnya di proposal. Dalam aplikasi ini, kami bisa mencetak satu tiket yang diinclude kan oleh beberapa user.

Analisis Kebutuhan Sistem

Proses analisis yang dilakukan secara pengamatan langsung terhadap apa yang terjadi di Theater Cinema.

Terdapat kejadian operasional sebagai berikut:

- a. Mencatat data Bioskop sebelum di pesan oleh penumpang
- b. Pembeli memesan tiket bioskop sesuai kebutuhan pembeli
- c. Kasir memberikan list film pada pembeli
- d. Kasir dan pembeli melakukan transaksi jual beli

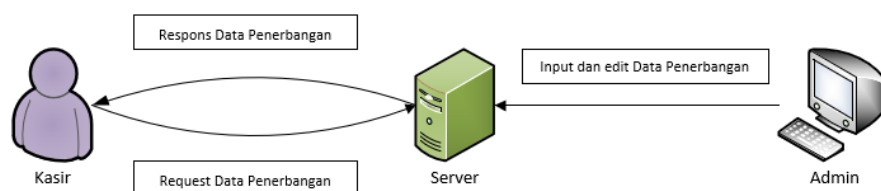
Kebutuhan Proses berbasis komputer berdasarkan kejadian operasional yaitu:

1. Menginputkan List bioskop (jadwal dan film) yang tersedia
2. Menginputkan pesanan pembeli
3. Mengupdate data penerbangan
4. Membuat laporan pemesanan berdasar pesanan pembeli

Kebutuhan Dokumen dan data (file) berdasarkan kejadian operasional adalah:

1. Formulir validasi kasir yang bisa dikelola oleh admin
2. Formulir booking pembeli dengan kasir yang dikelola kasir dan admin
3. Formulir jadwal dan maskapai
4. Form data passenger yang sudah diisi pembeli
5. Formulir booking berdasar pesanan pembeli

Arsitektur sistem



Berikut adalah Arsitektur sistem dari aplikasi pemesan tiket pesawat berbasis Desktop. Dari Arsitektur siste tersebut, terdapat penjelasan yang sudah kami jelaskan sebelumnya di proposal.

Menyusun kebutuhan sistem

Permodelan kebutuhan sistem yang dibutuhkan tidak dapat digambarkan tools. Melainkan hanya ditulis input proses dan output yang ada pada Sistem Informasi Persediaan.

Model kebutuhan sistem informasi pembelian tiket bioskop berbasis Desktop adalah sebagai berikut:

1. Proses : mengisi Jadwal film
Input : formulir Film
Output : file di form Film
Aktor : Admin
2. Proses : mengisi Studio
Input : formulir Studio
Output : file di form Studio
Aktor : Admin
3. Proses : mengelola data penonton
Input : formulir transaksi
Output : file di Form transaksi
Aktor : Admin
4. Proses : mengelola data booking
Input : formulir booking
Output : file di form booking
Aktor : Admin dan Kasir
5. Proses : mencetak tiket Bioskop
Input : formulir pembelian
Output : file di form pembelian
Aktor : Kasir
6. Proses : memilih hari, film, dan kode kursi, memilih makanan dan minuman, serta pembayaran
Input : Pembelian
Output : Inputan di pembelian
Aktor : User (Pembeli)

Desain

Metode Desain

Metode perancangan perangkat lunak terdapat beberapa pendekatan yaitu:

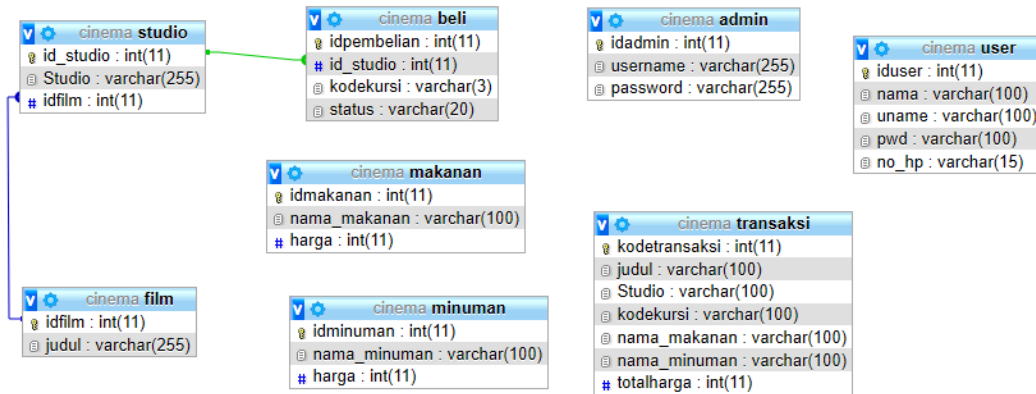
- *Structured Design method* (Constantie and Yourdon, 1979)
- *Structured systems Analysis* (Gane and Sarson, 1999)
- *Jackson System Development* (Jackson, 1983)
- *Object-oriented Design*(Robinson,1992; Booch,1994)

Dalam Metode Struktur melibatkan sejumlah aktifitas, notasi, format laporan , aturan dan panduan desain. Metode Struktural ini didukung oleh beberapa model sistem berikut:

1. Model Aliran data, dimana sistem dimodelkan menggunakan transformasi data yang ditempatkan sebagai proses. Hal ini digambarkan dengan *Data Flow Diagram*
2. Model *Entity-Relationship*, digunakan untuk menggambarkan struktur data secara logika digunakan.
3. Model Struktural, dimana komponen sistem dan antarmukanya didokumentasikan. Pendekatan ini ke arah modular dengan structure charts-nya.
4. Model *Object-oriented*, yang melibatkan model *inheritance* pada sistem. Model ini menggambarkan bagaimana objek disusun dan digunakan oleh objek lain.

DESAIN DATABASE

Dalam pembuatan Aplikasi pembelian tiket bioskop berbasis Desktop, kami menggunakan strategi desain terstruktur. Secara fungsional ini, kami melakukan penyempurnaan desain database yang sudah di relasikan sehingga lebih detail.

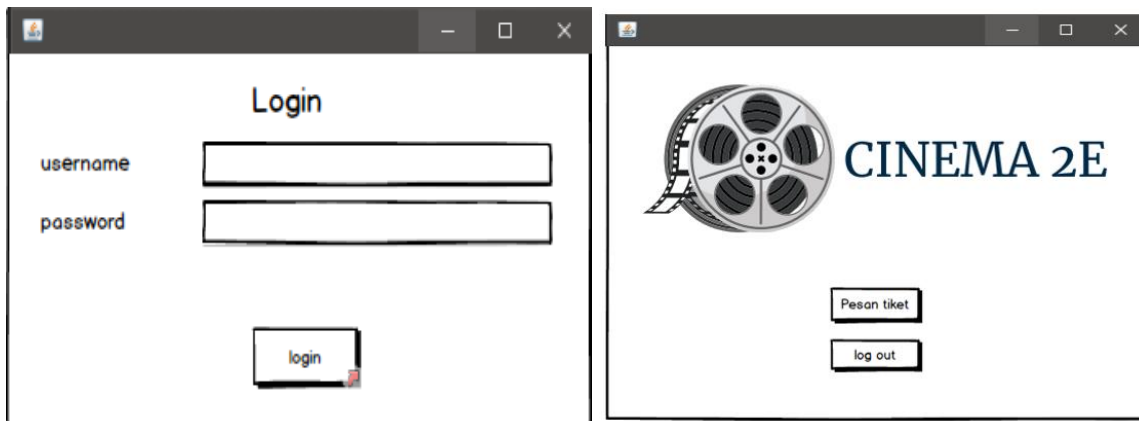


Berikut merupakan Desain database dari aplikasi kami yang sudah kami relasikan sehingga dalam melakukan pengolahan data lebih efektif. Kami membuat 8 tabel yang 3 diantaranya sudah kami relasikan yaitu tabel maskapai, tabel studio, tabel film, dan tabel beli lalu duanya hanya untuk kasir dan admin yaitu tabel user (user yang kami maksud ini adalah kasir) dan tabel admin

DESAIN MOCKUP

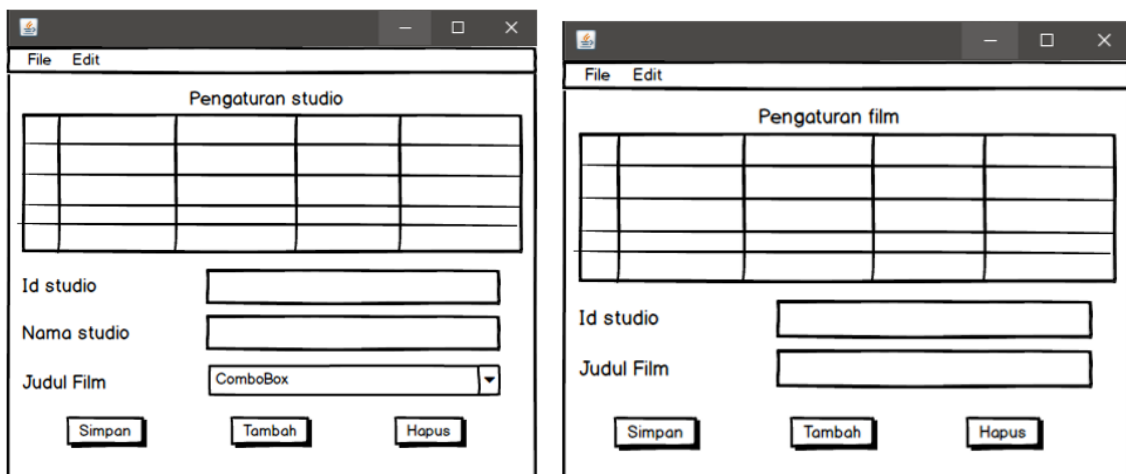
Dalam penerapan SDLC di bagian desain, tentu sebelum aplikasi diimplementasikan (tahap koding) maka dibuat desain interface itu dalam bentuk apapun bisa mockup atau yang lain. Disini kami membuat desain interface dengan mockup yang kami bagi menjadi 3 yaitu mockup form hanya untuk admin, mockup pembelian untuk kasir dan pembeli, dan mockup tampilan beranda, dan login. Mockup seperti ini :

1. Validasi dan beranda



Mockup ini berisi beranda dan validasi dari sistem ini dimulai dari beranda awal untuk kasir dan admin melakukan login yang dimana akan ditampilkan jika ada pembeli yang ingin memesan tiket bioskop

2. Admin



Id studio

Nama studio

Judul Film

Id studio

Judul Film

The image displays three separate window mockups for a management system, each with a 'File' and 'Edit' menu bar.

Pengaturan makanan

Id makanan

Nama makanan

Harga

Pengaturan minuman

Id minuman

Nama minuman

Harga

Pengaturan transaksi

Ini merupakan mockup form yang disediakan hanya untuk admin kami membuat menjadi 5 yaitu pengaturan studio, film, makanan, minuman, dan transaksi yang dimana keempatnya fungsinya mengolah bisa untuk menambah, mengedit atau menghapus data sebelum data tersebut masuk ke database

3. Kasir dan Pembeli

The mockup shows a movie booking application interface. It features a central seating chart with 8 rows and 8 columns of seats, each represented by a chair icon. To the right is a 'Booking' sidebar with dropdown menus for 'Film' (Avenger: Endgame), 'Hari', 'Jam' (11.15), and 'Studio' (Regular). Below these are input fields for 'Kursi' and 'Jumlah', followed by 'Pesan', 'Reset', and 'Cetak' buttons. At the bottom, there are three columns of input fields for pricing and food/drink orders, each with a 'Pilih' dropdown and a 'Total' label.

Layar

Booking

Film
Avenger: Endgame

Hari

Jam
11.15

Studio
Regular

Kursi

Jumlah

Pesan
Reset
Cetak

Harga Weekday :Rp.35000
Weekend :Rp.45000

Pilih makanan
Pilih minuman

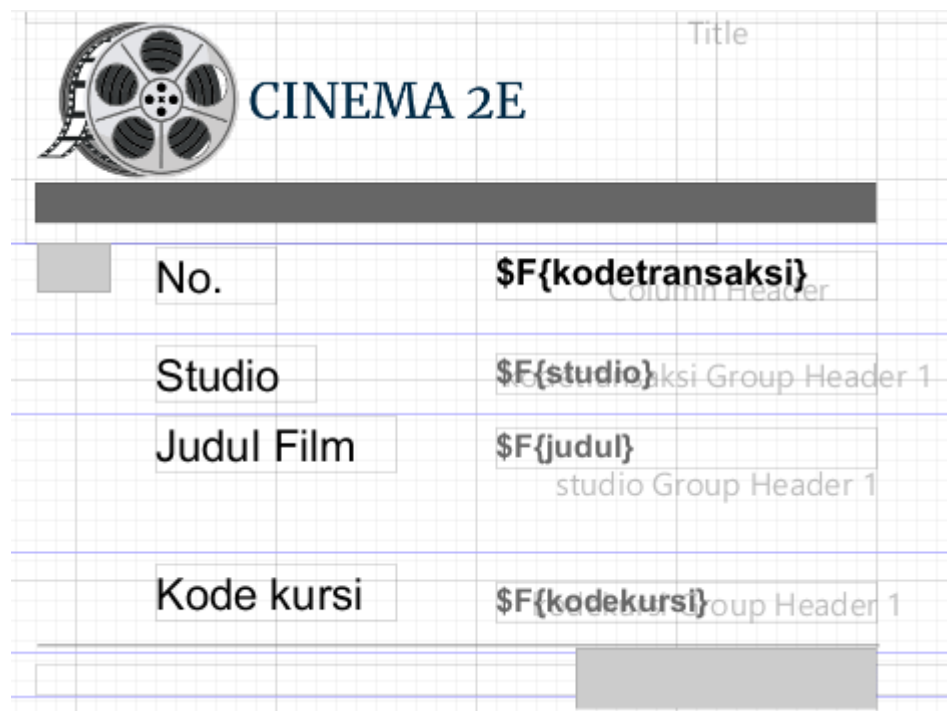
harga
Jumlah
Total

harga
Jumlah
Total

Harga Satuan Tiket
Total harga
Bayar
Kembali

Tampilan mockup ini jika pembeli ingin memesan lalu memilih film, studio, hari, jam, Studio, memilih kode kursi, memesan makanan, memesan minuman, dan melakukan pembayaran

4. Tiket



The mockup shows a cinema ticket for "CINEMA 2E". It includes a film reel logo and a table with the following fields:

Title	
No.	<code>\$F{kodetransaksi}</code>
Studio	<code>\$F{studio}</code>
Judul Film	<code>\$F{judul}</code>
Kode kursi	<code>\$F{kodekursi}</code>

Ini merupakan mockup dari tiket bioskop jika pembeli berhasil memesan tiket bioskop maka kasir akan mencetak tiket sebelum memberikannya ke pembeli dan film tersebut dimulai

IMPLEMENTASI

METODE IMPLEMENTASI

Dalam tahap implementasi kami membuat aplikasi ini menggunakan Bahasa pemrograman JAVA dengan Text Editor dan IDE ialah Netbean. Di aplikasi ini kami menggunakan GUI (Graphical User Interface). Semakin kompleks implementasi yang dibuat maka akan semakin banyak pula jumlah source code yang dibuat. Kriteria source code yang baik menurut kami yaitu :

- Format
- Dokumentasi
- Penggunaan #define
- Konsisten dalam penulisan

A. Format

Berikanlah indentasi yang konsisten di dalam kode program untuk membuat hubungan yang jelas antara kode blok, kelas, fungsi, serta loop dari mana mereka berasal.

```
package backEnd;
import java.util.ArrayList;
import java.sql.*;
/**
 *
 * @author Rifaul06
 */
public class Pembelian {
    private int idPembelian;
    private Kursi kursi = new Kursi();
    private Studio studio = new Studio();

    public Pembelian() {
    }

    public int getIdPembelian() {
        return idPembelian;
    }

    public void setIdPembelian(int idPembelian) {
        this.idPembelian = idPembelian;
    }
}
```

B. Dokumentasi

menulis komentar pada program saat Anda menulisnya. Namun, Anda sebaiknya tidak menulis komentar secara berlebihan ataupun terlalu sedikit serta tidak perlu bagi Anda menulis komentar untuk suatu baris program yang sudah jelas ataupun umum diketahui.

C. Penggunaan #define

Untuk mendefinisikan sesuatu nilai yang bernilai default, sebaiknya kita menggunakan #define. Hal itu lebih baik daripada kita menuliskannya secara manual ke beberapa baris program yang menggunakannya.

```
public Kursi getKursi() {  
    return kursi;  
}  
  
public void setKursi(Kursi kursi) {  
    this.kursi = kursi;  
}  
  
public Studio getStudio() {  
    return studio;  
}  
  
public void setStudio(Studio studio) {  
    this.studio = studio;  
}
```

D. Konsisten dalam penulisan

dalam penamaan variabel, namespace case sensitive, dan ukuran agar mudah dipahami. Untuk penulisan namespace ada dua cara yaitu semua huruf pertama dari setiap kata dikapitalisasi kecuali kata pertama seperti kataSatu atau digunakan underscores (garis bawah)

PENJELASAN IMPLEMENTASI

Dalam implementasi kami membuat BackEnd dan FrontEnd demi memudahkan dalam pengodingan aplikasi ini.

1. BackEnd

Untuk BackEnd kami membuat DBHelper yang gunanya untuk mengkoneksikan ke database sehingga data kami dapat tersimpan di database dengan aman

```

public class DBHelper {
    private static Connection koneksi;
    public static void bukaKoneksi(){
        if (koneksi == null) {
            try {
                String url = "jdbc:mysql://localhost/cinema";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi = DriverManager.getConnection(url, "root", "");
            } catch (SQLException e) {
                System.out.println("Error koneksi !");
            }
        }
    }
}

```

Lalu kami membuat BackEnd lagi dengan nama Film, Studio, kursi, makanan, minuman , pembelian, Studio, dan transaksi yang masing-masing memiliki kegunaan untuk saling berhubungan untuk dibuat FrontEnd nya dan tentu saja masing-masing BackEnd sudah saling berelasi berdasarkan database yang sudah kami buat sebelumnya

```

package backEnd;
import java.util.ArrayList;
import java.sql.*;

/**
 * @author Rifaul06
 */
public class Film {
    private int idFilm;
    private String judul;
    private String genre;

    public Film();

    public String getGenre() {
        return genre;
    }

    public void setGenre(String genre) {
        this.genre = genre;
    }

    public Film(String judul) {
        this.judul = judul;
    }
}

```

```

package backEnd;
import java.util.ArrayList;
import java.sql.*;

/**
 * @author Rifaul06
 */
public class Kursi {
    private String kode;
    private String status;

    public Kursi() {...2 lines }

    public Kursi(String kode, String status) {
        this.kode = kode;
        this.status = status;
    }

    public String getKode() {...3 lines }

    public void setKode(String kode) {...3 lines }

    public String getStatus() {
        return status;
    }
}

```

Tampilan BackEnd Lainnya :

```
package backEnd;

import java.sql.ResultSet;
import java.util.ArrayList;

/**
 *
 * @author Rifaul06
 */
public class LoginKasir {
    private int id;
    private String username;
    private String password;
    private String nama;
    private String noTlp;
    private String email;

    public LoginKasir() {
    }

    public int getId() {
        return id;
    }
}

package backEnd;

import java.util.ArrayList;
import java.sql.*;

/**
 *
 * @author soyidwahyud
 */
public class Makanan {
    private int idmakanan;
    private int harga = 0;
    private String nama_makanan;

    public Makanan(){};

    public int getIdmakanan() {
        return idmakanan;
    }

    public void setIdmakanan(int idmakanan) {
        this.idmakanan = idmakanan;
    }

    public int getHarga() {
        return harga;
    }
}

package backEnd;

import java.util.ArrayList;
import java.sql.*;

/**
 *
 * @author soyidwahyud
 */
public class Minuman {
    private int idminuman, harga = 0;
    private String nama_minuman;

    public Minuman(){};

    public int getIdminuman() {
        return idminuman;
    }

    public void setIdminuman(int idminuman) {
        this.idminuman = idminuman;
    }

    public int getHarga() {
        return harga;
    }
}

package backEnd;

import java.util.ArrayList;
import java.sql.*;

/**...4 lines */
public class Pembelian {
    private int idPembelian;
    private Kursi kursi = new Kursi();
    private Studio studio = new Studio();

    public Pembelian() {
    }

    public int getIdPembelian() {
        return idPembelian;
    }

    public void setIdPembelian(int idPembelian) {
        this.idPembelian = idPembelian;
    }

    public Kursi getKursi() {
        return kursi;
    }
}
```

```

package backEnd;
import java.util.ArrayList;
import java.sql.*;

/**
 *
 * @author Rifaul06
 */
public class Studio {
    private int idStudio;
    private String Studio;
    private Film film = new Film();

    public Studio() {
    }

    public Studio(String Studio) {
        this.idStudio = idStudio;
        this.Studio = Studio;
    }

    public int getIdStudio() {
        return idStudio;
    }
}

```

```

package backEnd;
import java.util.ArrayList;
import java.sql.*;

/**
 *
 * @author soyidwahyud
 */
public class Transaksi
{
    private int idTransaksi;
    private String judul, Studio, kodekursi, nama_makanan, nama_minuman;
    private int total;

    public Transaksi()
    {
    }

    public String getNama_makanan() {
        return nama_makanan;
    }

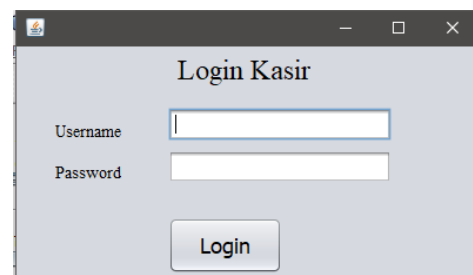
    public void setNama_makanan(String nama_makanan) {
        this.nama_makanan = nama_makanan;
    }
}

```

2. FrontEnd

Setelah pembuatan BackEnd maka kami membuat tampilan FrontEnd nya untuk dilihat oleh pembeli. Kami membuatnya melalui JFrame berdasarkan mockup yang sudah kami buat di desain Mockup

A. FrontEnd Validasi



FrontEnd ini berisi beranda dan validasi dari sistem ini dimulai dari beranda awal untuk kasir dan admin melakukan login atau registrasi jika kasir belum terdaftar di database lalu ada tampilan beranda jika kasir sudah login yang dimana akan ditampilkan jika ada pembeli yang ingin memesan tiket bioskop.

B. FrontEnd Admin

Pengaturan Studio

ID	Studio	Film
6	Reguler	Pokemon Detectiv...
7	Deluxe	Pokemon Detectiv...
8	Reguler	John Wick 3:Parab...
9	Reguler	Avenger: EndGame
10	Deluxe	Long Shot
11	Reguler	Long Shot
12	Deluxe	Avenger: EndGame

id Studio:

nama Studio:

Judul Film:

Simp... Hapus Tambah

Pengaturan Film

Id	Judul
1	Pokemon Detective Pikachu
2	Avenger: EndGame
3	John Wick 3:Parabellum
4	Long Shot

id Film:

Judul Film:

Add Delete Save

Pengaturan Makanan

idmakanan	nama_makanan	harga
1	Pop Corn (small)	35000
2	Pop Corn (Medium)	50000
3	Pop Corn (Large)	65000
4	Fish n Chips	40000

Id Makanan:

Nama Makanan:

Harga:

Add Delete Save

Pengaturan Minuman

idminuman	nama_minuman	harga
1	Java Tea	20000
2	Iced Lychee Tea	35000
3	Milo Dinosaurius	35000
4	Iced Mocchacino	40000

Id Minuman:

Nama Minuman:

Harga:

Add Delete Save

Transaksi

Bayar Delete

ID	Studio	Film	kursi	Total harga
18	Reguler	A Star is B...	C1	35000
19	Reguler	Avenger: E...	C6	35000
20	Reguler	Avenger: E...	F8	35000
21	Reguler	Pokemon ...	B1	45000

Ini merupakan FrontEnd form yang disediakan hanya untuk admin kami membuat menjadi 5 yaitu pengaturan studio, film, transaksi, makanan, dan minuman yang dimana keempatnya fungsinya mengolah bisa untuk menambah, mengedit atau menghapus data sebelum data tersebut masuk ke database

C. FrontEnd Kasir dan Pembeli

The screenshot displays a movie booking application interface. On the left, a grid of 24 seats is arranged in 6 rows (A-F) and 4 columns (1-4). Each seat is represented by a chair icon and a label (e.g., A1, A2, ..., F8). Below the grid, there are input fields for 'Harga' (Price) with 'Weekday : Rp. 35000' and 'Weekend : Rp. 45000', 'Harga Satuan Tiket', 'Total Harga', and 'Total Keseluruhan'. There are also buttons for 'Bayar' (Pay) and 'Kembali' (Back). In the center, there are dropdown menus for 'Pilih Makanan' (Pop Corn (small)) and 'Pilih Minuman' (Java Tea), along with input fields for 'Harga' and 'Jumlah' for each. On the right, a 'BOOKING' section contains dropdown menus for 'Film' (Pokemon Detective Pikachu), 'Hari', 'jam' (11.15), 'Studio', and 'Item 1'. There is also an input field for 'Kursi' (Seat) and a 'Jumlah' input field. At the bottom of the booking section are buttons for 'Pesanan' (Order), 'Reset', and 'Cetak' (Print).

Tampilan frontend ini jika pembeli ingin memesan lalu memilih film, studio, hari, jam, Studio, memilih kode kursi, memesan makanan, memesan minuman, dan melakukan pembayaran

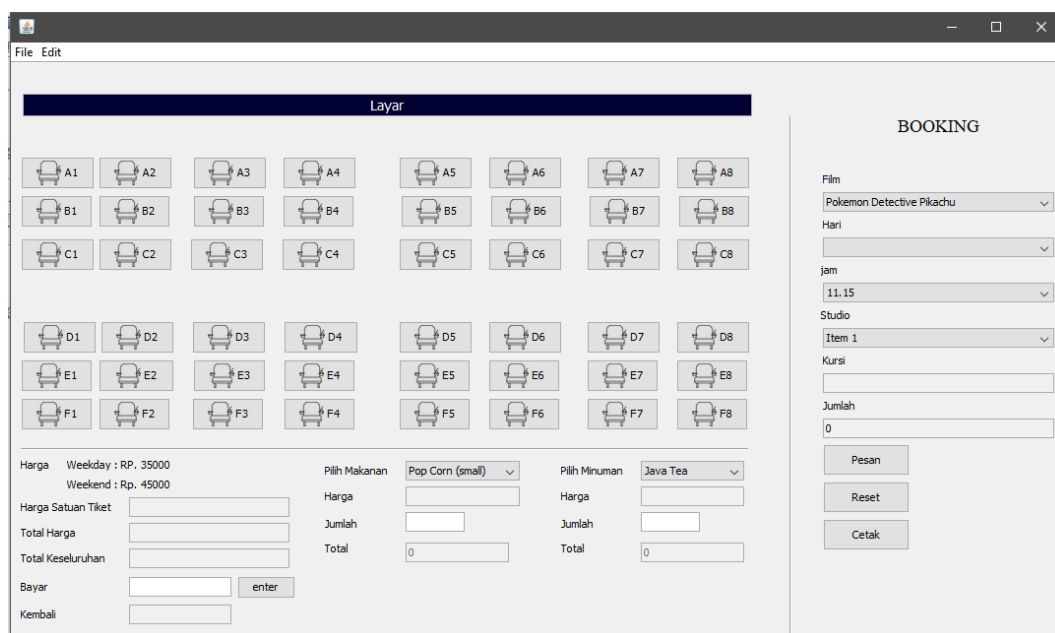
JALANNYA PROGRAM

The screenshot shows a 'Login Kasir' (Clerk Login) screen. It features a title 'Login Kasir' at the top. Below the title, there are two input fields: 'Username' with the text 'soyid' and 'Password' with masked characters '*****'. A 'Login' button is positioned below the password field.

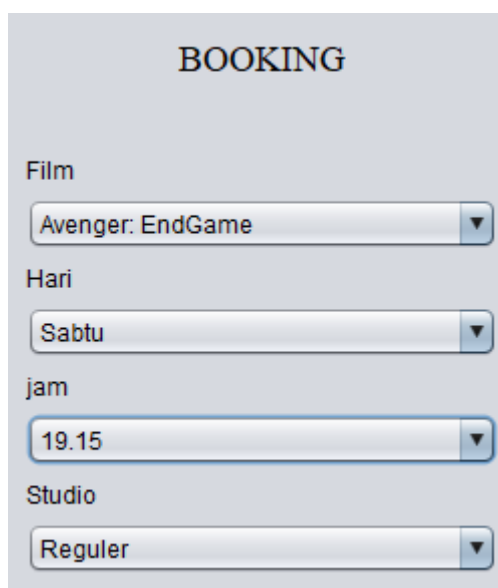
Pertama dengan melakukan login sebagai kasir. Maka klik button login. Di menu login kita mengisi username dan password yang sudah ada di database



Setelah login maka akan muncul tampilan seperti ini. Maka jika ingin melakukan pemesanan maka klik button pesan tiket



Muncul tampilan pembelian maka kita ikuti langkah-langkahnya terlebih dahulu yaitu :



Disini sebagai contoh saya memesan film berjudul Avenger: Endgame yang tayang di hari sabtu dan mulai pukul 19.15 dengan tipe studio yang saya pesan adalah reguler



Lalu pembeli memilih kode kursi berdasarkan denah bioskop pada gambar di atas.



Tampilan ini muncul jika penonton lain sudah memesan setelah saya maka muncul notifikasi “Sudah dibooking”

Kursi	
	E5 E6
Jumlah	
	2
Harga Weekday : RP. 35000 Weekend : Rp. 45000	
Harga Satuan Tiket	45000.0
Total Harga	90000.0

Saya memesan dua tiket, karena saya memesan dengan teman saya maka tampilan seperti ini :

Kursi yang tersedia ada dua yaitu E5 dan E6

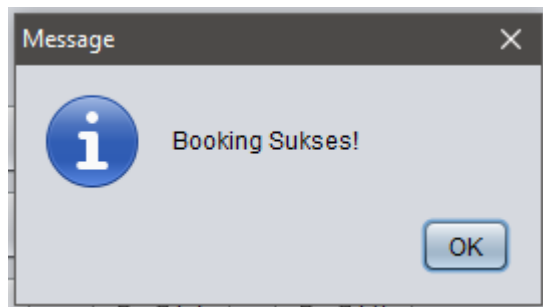
Harga satuan tiket Rp. 45000, karena saya memesan 2 maka total harga adalah Rp. 90000

Pilih Makanan	Pop Corn (Large) ▼	Pilih Minuman	Milo Dinosaurus ▼
Harga	65000	Harga	35000
Jumlah	2	Jumlah	2
Total	130000	Total	70000

Ini tampilan untuk memesan makanan dan minuman, ada harga satuan dan karena saya menonton dengan teman saya maka harga satuannya di kali 2 maka harga totalnya muncul.

Bayar	300000	enter
Kembali	110000.0	

Bayar ini merupakan total dari harga tiket, harga makanan dan harga minuman. Jika sudah, maka klik pesan dan muncul tampilan “Booking Sukses!”



Setelah booking berhasil, maka kasir akan mencetak tiket bioskop untuk diberikan ke pembeli. Tampilan tiket bioskop adalah seperti berikut



CINEMA 2E



No.

22

Studio

Reguler

Judul Film

Avenger: EndGame

Kode kursi

E5 E6

May 16, 2019

Untuk pengecekan apakah sudah disimpan di database apa belum, maka admin melihat di form transaksi dan setelah dilihat bahwa pemesan dengan kode kursi E5 dan , film Avenger: EndGame, dan tipe studio reguler sudah memiliki status book dengan total harga sesuai harga tiket dan tipe payment sesuai yang dipilih pembeli

Transaksi				
<div>Bayar Delete</div>				
ID	Studio	Film	kursi	Total harga
19	Reguler	Avenger: EndGame	C6	35000
20	Reguler	Avenger: EndGame	F8	35000
21	Reguler	Pokemon Detective Pikachu	B1	45000
22	Reguler	Avenger: EndGame	E5 E6	90000