

LAPORAN TUGAS BESAR PEMROGRAMAN BERBASIS OBJEK
CINEMA 5 TICKETING SYSTEM



Disusun oleh:

Kelompok 4

TT-44-06

1. Muhammad Fahri Tarigan 1101201323
2. Ikram Tauffiqul Hakim 1101204028
3. Bintang Kurniawan 1101204231
4. Endika Satrio Wibowo 1101204281

PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI
FAKULTAS TEKNIK ELEKTRO
UNIVERSITAS TELKOM

2022

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi pada era globalisasi sekarang ini makin meningkat serta didukung oleh perkembangan Ilmu Pengetahuan dan Teknologi (IPTEK) yang mendorong setiap orang untuk melakukan perbaikan dan pembaharuan dalam sistem penyampaian informasi yang telah mereka miliki. Keinginan masyarakat untuk memperoleh kemudahan dalam melakukan kegiatan sehari-hari mendorong pesatnya kemajuan teknologi. Banyak teknologi yang ditujukan untuk memberikan kemudahan yang diinginkan, seperti misalnya internet. Fakta yang tidak bisa ditolak saat ini adalah kenyataan hampir segala hal dapat diperoleh di internet. Dengan kemajuan teknologi internet memungkinkan seseorang untuk memperoleh informasi dan melakukan transaksi dengan bebas tanpa dibatasi oleh ruang dan waktu.

Di zaman yang semakin modern pada saat ini, perkembangan perfilman sudah semakin berkembang dengan cepat. Penjualan tiket bioskop pun diatur secara otomatis melalui media aplikasi dan sangat mudah untuk digunakan oleh kasir bioskop. Sebelum adanya transformasi digital, proses transaksi yang dilakukan oleh kasir bioskop terbilang masih menggunakan sistem manual dan dapat mengakibatkan terjadinya penumpukan antrian.

Berdasarkan permasalahan tersebut, kami yang beranggotakan empat mahasiswa Universitas Telkom merancang sebuah inovasi yang memudahkan kasir bioskop dalam mengatur dan memenuhi pesanan pelanggan bioskop dengan mudah dan praktis. Inovasi ini bertujuan untuk menjaga kualitas dari pemesanan tiket bioskop. Aplikasi ini dibangun melalui aplikasi Netbeans dengan memanfaatkan bahasa pemrograman Java. Pada aplikasi ini disediakan fasilitas *code editor* dan GUI. Dengan demikian aplikasi ini dapat meningkatkan pelayanan dan kepuasan pembeli tiket bioskop.

1.2 Tujuan

Tujuan dalam pembangunan aplikasi ini, sebagai berikut :

1. Membangun sebuah sistem pemesanan tiket yang paling efektif untuk mempermudah kasir
2. Mencegah terjadinya penumpukan antrian bioskop.

1.3 Batasan Masalah

Pada pembuatan laporan ini, kami membatasi masalahnya sebagai berikut :

1. Merancang aplikasi yang digunakan hanya untuk admin atau kasir bioskop saja.
2. Fitur yang ditawarkan pada aplikasi hanya dikhususkan untuk kasir.
3. Database yang ditampilkan pada aplikasi hanya memuat per hari sesuai dengan jam tayang pada list film yang tertera.
4. Membuat dan menghubungkan aplikasi ke database

BAB II

LANDASAN TEORI

2.1 OOP (Object Oriented Programming)

OOP (Object Oriented Programming) adalah sebuah istilah yang diberikan kepada bahasa pemrograman yang menggunakan teknik berorientasi atau berbasis pada sebuah objek dalam pembangunan program aplikasi, maksudnya bahwa orientasi pembuatan program tidak lagi menggunakan orientasi linear melainkan berorientasi pada objek-objek yang terpisah-pisah, atau lebih singkatnya adalah suatu metode pemrograman yang berorientasi kepada objek. Tujuan dari OOP diciptakan adalah untuk mempermudah pengembangan program dengan cara mengikuti model yang telah ada di kehidupan sehari-hari. Jadi setiap bagian dari suatu permasalahan adalah objek, nah objek itu sendiri merupakan gabungan dari beberapa objek yang lebih kecil.

Konsep OOP pertama kali muncul di MIT (Massachusetts Institute of Technology) pada era 1960-an. Lalu sekitar beberapa tahun kemudian antara 1962-1965, sebuah bahasa pemrograman yang mendasari konsep OOP diperkenalkan dengan nama bahasa pemrograman SIMULA 1, dikembangkan oleh Kristen Nygaard dan Ole-Johan yang merupakan warga negara Norwegia. Setelah itu pada tahun 1967 keluarlah SIMULA 67. Pada tahun 70an tercipta bahasa SMALLTALK, pada tahun 1980-an banyak Bahasa pemrograman menggunakan konsep OOP diperkenalkan kepada dunia, tapi hanya empat yang cukup populer yaitu ADA (US Department of Defense), PROLOG (the Japanese “Fifth Generation Computer project”), Eiffel dan C++. Dan pada tahun 90-an, bahasa pemrograman JAVA diperkenalkan yang mengaku terinspirasi oleh C++.

Mengacu pada pengertian OOP dan Apa itu OOP di atas, Fitur OOP meliputi:

- a. Encapsulation: Ini membuat struktur program lebih mudah untuk dikelola karena implementasi dan status setiap objek tersembunyi di balik batas yang ditentukan dengan baik.
- b. Polimorphism: Ini berarti entitas abstrak diimplementasikan dalam berbagai cara,
- c. Inheritance: Ini merujuk pada susunan hierarki implementasi fragmen.

2.2 Netbeans

Netbeans adalah Integrated Development Environment (IDE) berbasis Java dari Sun Microsystems yang berjalan di Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan diberbagai macam platform seperti Windows, Linux, Mac OS X dan Solaris. Suatu IDE adalah lingkup pemrograman yang diintegrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan pembangunan Graphical User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Software Netbeans sebagai platform memungkinkan berbagai aplikasi Java untuk dikembangkan dari suatu set modular komponen software yang disebut modul. Software Netbeans dan lainnya dapat diunduh secara gratis, bebas dan lengkap.

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

2.4 MySql

SQL adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Saat ini hampir semua server basis data yang ada mendukung bahasa ini untuk melakukan manajemen datanya. MySQL sebenarnya adalah turunan dari salah satu konsep utama dalam database yang ada; SQL (Structured Query Language). SQL adalah konsep basis data operasional, khususnya untuk memilih atau memasukkan data, yang memungkinkan untuk mengeksekusi data secara otomatis dengan cara yang sederhana. MySQL menurut Van der Lans, ... "MySQL adalah server basis data relasional yang mendukung basis data SQL yang terkenal (Structured Query Language)". Ini berarti bahwa MySQL adalah server database relasional yang mendukung bahasa database SQL (Structured Query Language) yang terkenal.

2.5 XAMPP

XAMPP adalah perangkat lunak bebas, yang banyak mendukung sistem operasi dan merupakan kompilasi berbagai program. Fungsi XAMPP seperti server berdiri sendiri (localhost), terdiri dari pada program Apache HTTP Server, Database dan penerjemah MySQL bahasa yang ditulis dalam bahasa tersebut PHP dan pemrograman Perl. nama XAMPP adalah singkatan dari X (empat sistem operasi apa pun), Apache, MySQL, PHP dan Perl. Program ini tersedia di GNU General Public Lisensi gratis, ini jaringan server yang mudah digunakan tampilan halaman web bisa digunakan dinamis.

Menurut (Waryanto, 2018) versi terbaru dari XAMPP (dengan PHP 7.2.1) memiliki komponen sebagai berikut :

1. Apache versi 2.4.29 adalah aplikasi web server default.
2. MariaDB versi 10.1.30 adalah sistem manajemen database.

3. PHP versi 7.2.1 adalah server-side scripting untuk membuat aplikasi berbasis web.
 4. phpMyAdmin versi 4.7.4 adalah tool untuk menggunakan MySQL berbasis aplikasi.
 5. OpenSSL versi 1.1.0g adalah implementasi open source dari dua protokol keamanan populer, yaitu SSL dan TLS.
 6. XAMPP Control Panel versi 3.2.2 adalah kontrol panel sederhana untuk mengatur komponen berbeda pada XAMPP.
 7. Webalizer versi 2.23-04 adalah sebuah tool analitik untuk user log dan metrik penggunaan.
 8. Mercury Mail Transport System versi 4.63 adalah email server open source.
 9. FileZilla FTP Server versi 0.9.41 berfungsi untuk melakukan transfer file.
 10. Tomcat versi 7.0.56 adalah java servlet freeware untuk aplikasi Java.
 11. Strawberry Perl 7.0.56 Portable berfungsi untuk melakukan distribusi Perl.
- XAMPP dikembangkan oleh tim proyek yang disebut Apache Teman, yang terdiri dari Tim Inti (Core Tim), Tim Pengembangan (Pengembangan Tim) dan tim pendukung (dukungan tim).

2.6 Objektif

Aplikasi ini dibangun agar dapat memudahkan kasir dalam melayani pengunjung yang ingin menonton film di bioskop. Dalam aplikasi ini nantinya pengunjung akan diminta untuk mengisi nama, jenis kursi, level kursi dan membayarnya. Rincinya sebagai berikut:

1. Welcome page
Pada halaman ini akan ada tampilan selamat datang dan tombol login yang digunakan oleh kasir.
2. Login page
Pada halaman ini admin akan login terlebih dahulu dengan menggunakan kombinasi *username* dan *password* (*username*: admin dan *password*: admin). Setelah memasukkan *username* dan *password* admin memilih tombol "Login"
3. Menu page
Pada halaman ini terdapat tampilan menu yang mempunyai fungsi masing-masing, seperti pesan tiket yang berfungsi untuk membeli tiket film, c'food yang berfungsi untuk memesan makanan dan minuman, lihat data yang berfungsi untuk melihat data pelanggan yang berhasil terpesan di aplikasi Cinema5, dan logout yang berfungsi untuk keluar dari aplikasi tersebut.
4. Pesan Tiket page
Pada halaman ini kita akan memasukkan data kita untuk memesan tiket. Masukkan nama pemesan, jenis kursi (A, B, C) dan jumlah kursi yang akan dipesan, Total bayar akan keluar setelah proses pemesanan selesai dilakukan. Selanjutnya, kasir akan melakukan proses pembayaran dan apabila uang dari pelanggan lebih dari total pembayaran, aplikasi cinema5 akan menampilkan jumlah uang kembalian untuk proses pembayaran. Setelah itu tekan tombol

“Next”. Tiket Film akan tercetak secara otomatis dan setelah itu klik “Save”.

5. C’Food

Pada halaman ini kasir dapat memilih makanan dan minuman sesuai dengan pemesanan pelanggan. Setelah proses pemesanan selesai, tampilan akan tertuju pada tampilan pembayaran dan tampilan ini hamper sama dengan tampilan yang terdapat pada proses pembayaran di tampilan pesan tiket.

6. Lihat Data

Pada tampilan ini, kasir dapat melihat data yang berhasil dipesan sesuai dengan permintaan pelanggan. Pada tampilan lihat data juga dapat menampilkan data C’Food dan Film yang berguna untuk mengoreksi permintaan pelanggan dan menghitung data pemasukan bioskop pada hari itu.

BAB III

HASIL DAN ANALISIS

3.1 Software dan Hardware

Dalam pembuatan aplikasi Cinema5 ini, penulis menggunakan beberapa software dan hardware sebagai pendukung dalam pembuatan aplikasi ini. Software dan hardware yang digunakan adalah :

1. NetBeans

Netbeans merupakan sebuah aplikasi *Integrated Development Environment* (IDE) berbasis Java dari *Sun Microsystems* yang berjalan di atas swing. Swing sendiri adalah sebuah teknologi Java untuk pengembangan aplikasi dekstop yang dapat berjalan di berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Pada pembuatan aplikasi ini, kami menggunakan NetBeans IDE 15 dan 16.

2. Perangkat komputer

Komputer berguna untuk pembuatan aplikasi dari segi hardware, kami menggunakan komputer dalam membuat pemrograman aplikasi berbasis java, membuat database aplikasi, dan menghubungkan database dengan aplikasi.

3. PhpMyAdmin

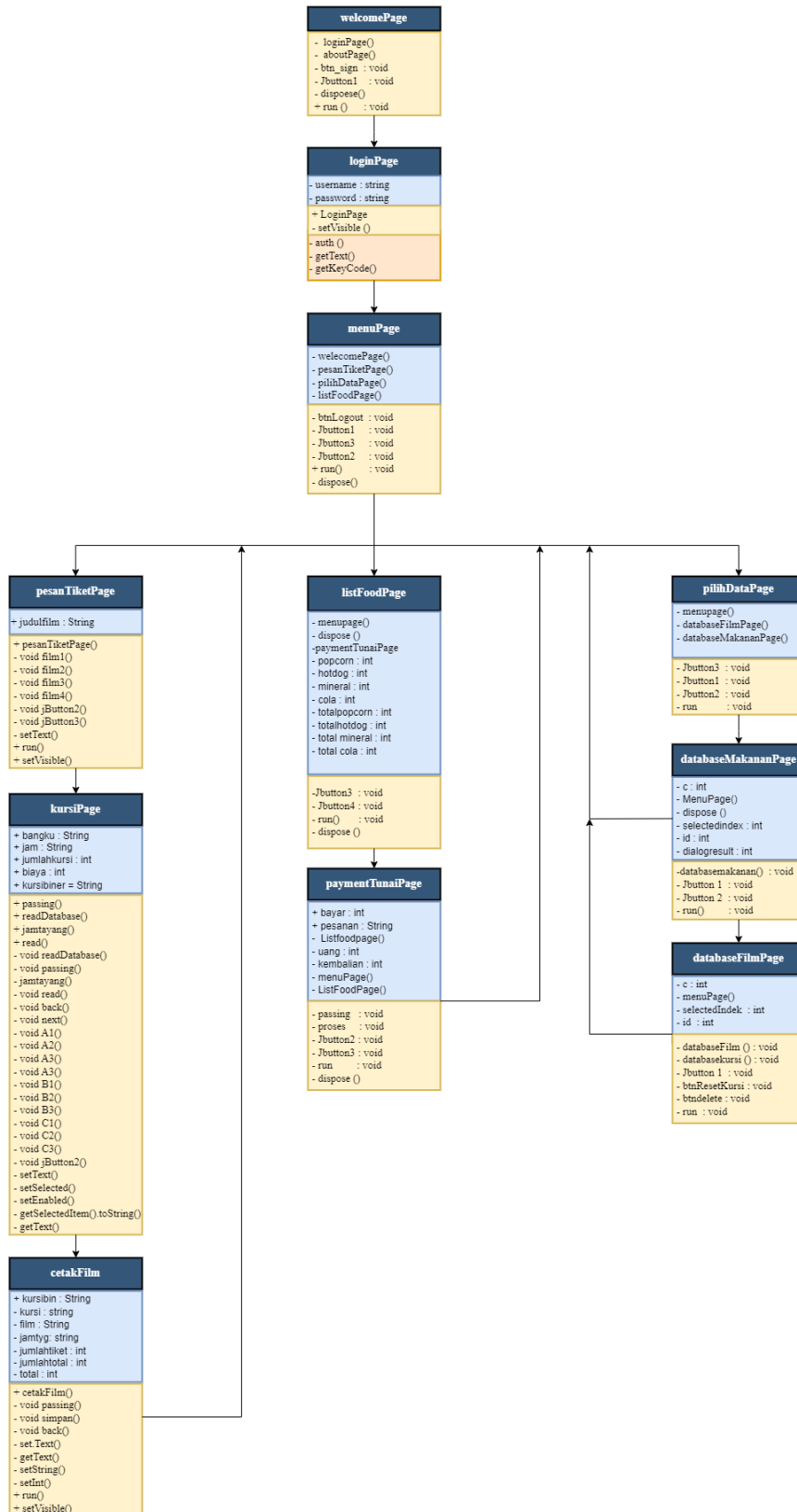
PhpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (World Wide Web). phpMyAdmin mendukung berbagai operasi MySQL, diantaranya (mengelola basisdata, table-tabel, bidang (fields), relasi (relation), indeks, pengguna (users), perijinan (permissions, dan lain-lain). Pada pembuatan aplikasi ini, PhpmyAdmin bertugas untuk mengelola database dengan aplikasi melalui jaringan local host didukung oleh aplikasi XAMPP.

4. XAMPP

Pada aplikasi ini, XAMPP berfungsi dalam mengatur jalannya koneksi dari aplikasi ke database melalui jaringan local host. Agar data aplikasi tersimpan dengan baik, penulis menyalakan mySQL dan Apache pada aplikasi XAMPP. Apabila kedua hal ini tidak dinyalakan, database pada aplikasi Cinema5 ini tidak dapat dimunculkan dan pemesanan pada aplikasi tidak dapat disimpan.

3.2 Flow Chart

CLASS DIAGRAM



3.3 Implementasi

Dibawah ini merupakan listing program pada aplikasi Cinema5, sebagai berikut.

1. Tampilan Skrip Welcome Page

```
import java.awt.Label;
import javax.swing.ImageIcon;
import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author ikram
 */
public class welcomePage extends javax.swing.JFrame {

    /**
     * Creates new form welcomePage
     */
    public welcomePage() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // Generated Code

    private void btn_signActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        loginPage lp = new loginPage();
        lp.show();
    }

    dispose();
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    aboutPage ap = new aboutPage();
    ap.setVisible(true);
    dispose();
    JOptionPane.showMessageDialog(parentComponent:null, message:"KELOMPOK 5\nTT-44-06\nIkram Tauffiqul Haki");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new welcomePage().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JButton btn_sign;
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel2;
// End of variables declaration
}
```

2. Tampilan skrip Login page

```
import java.awt.event.KeyEvent;
import javax.swing.JFrame;
import javax.swing.JOptionPane;

/**
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author Endika
 */
public class LoginPage extends javax.swing.JFrame {

    /**
     * Creates new form loginPage
     */
    public LoginPage() {
        initComponents();
    }

    private void auth() {
        menuPage mp = new menuPage();
        if (username.getText().equals("")) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Masukan username!");
        } else if (password.getText().equals("")) {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Masukan password");
        } else if (username.getText().contains(":admin") && password.getText().contains(":admin")) {
            mp.show();
            dispose();
        } else {
            JOptionPane.showMessageDialog(parentComponent: null, message: "Username atau password salah!", title: "Message", messageType: JOptionPane.ERROR_MESSAGE);
        }
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void btn_loginActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        auth();
    }

    private void passwordKeyTyped(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
    }

    private void passwordKeyPressed(java.awt.event.KeyEvent evt) {
        // TODO add your handling code here:
        if (evt.getKeyCode() == KeyEvent.VK_ENTER) {
            auth();
        }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new LoginPage().setVisible(true);
            }
        });
    }
}
```

```

private void passwordKeyTyped(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
}

private void passwordKeyPressed(java.awt.event.KeyEvent evt) {
    // TODO add your handling code here:
    if(evt.getKeyCode() == KeyEvent.VK_ENTER){
        auth();
    }
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new loginPage().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel bg_loginPage;
private javax.swing.JButton btn_login;
private javax.swing.JPasswordField password;
private javax.swing.JTextField username;
// End of variables declaration
}

```

3. Tampilan skrip menu Page

```

import javax.swing.JOptionPane;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author HP
 */
public class menuPage extends javax.swing.JFrame {

    /**
     * Creates new form menuPage
     */
    public menuPage() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void btnLogoutActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        JOptionPane.showMessageDialog(parentComponent=null, message:"Berhasil logout");
        welcomePage wp = new welcomePage();
        wp.show();
        dispose();
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        pesanTiketPage pt = new pesanTiketPage();
        pt.show();
        dispose();
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        pilihDataPage pdp = new pilihDataPage();
        pdp.show();
        dispose();
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        listFoodPage lf = new listFoodPage();
        lf.show();
        dispose();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new menuPage().setVisible(true);
            }
        });
    }
}

```

```

public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new menuPage().setVisible(true);
        }
    });
}

// Variables declaration - do not modify
private javax.swing.JLabel bg_menuPage;
private javax.swing.JButton btnLogout;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
// End of variables declaration
}

```

4. Tampilan Pesan Tiket

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */
/**
 *
 * @author ikram
 */
public class pesanTiketPage extends javax.swing.JFrame {

    public static String judulFilm;

    /**
     * Creates new form pesanTiketPage
     */
    public pesanTiketPage() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void film1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        judulFilm = "Fifty Shades Darker";
        judul.setText(text: judulFilm);
    }

    private void film2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        judulFilm = "The Conjuring 2";
        judul.setText(text: judulFilm);
    }

    private void film3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        judulFilm = "Cocomelon";
        judul.setText(text: judulFilm);
    }

    private void film4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        judulFilm = "Avengers Endgame";
        judul.setText(text: judulFilm);
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        kursiPage kp = new kursiPage();
        kp.show();
        dispose();
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        menuPage mp = new menuPage();
        mp.show();
        dispose();
    }

    /**
     * @param args the command line arguments
     */
}

```

5. Tampilan Skrip Kursi Page

```
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.ResultSetMetaData;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
import javax.swing.JOptionPane;
import javax.swing.JToggleButton;
import javax.swing.table.DefaultTableModel;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GuiForms/JFrame.java to edit this template
 */

/**
 *
 * @author ikram
 */
public class kursiPage extends javax.swing.JFrame {

    public static String bangku;
    public static String jam;
    public static int jumlahkursi;
    public static int biaya;
    public static String kursibiner;

    public PreparedStatement ps;
    public ResultSet rs;
    Connection con = koneksi.koneksiDatabase.BukaKoneksi();

    public kursiPage() {
        initComponents();
        passing();
        readDatabase();
        jamtayang();
        read();
        hargafilm.setText("Rp. 50,000");
    }

    private void readDatabase() {
        try {
            ps = con.prepareStatement("SELECT * FROM databasekursi");
            rs = ps.executeQuery();
            ResultSetMetaData Rss = rs.getMetaData();

            Vector judulDatabase = new Vector();
            Vector binerDatabase = new Vector();

            while(rs.next()){
                judulDatabase.add(rs.getString("judulfilm"));
                binerDatabase.add(rs.getString("kursibiner"));
            }

            kursiDatabase = binerDatabase;

            film1 = (String) binerDatabase.get(index:0);
            film2 = (String) binerDatabase.get(index:1);
            film3 = (String) binerDatabase.get(index:2);
            film4 = (String) binerDatabase.get(index:3);

        } catch (Exception e) {
        }
    }
}
```

```

private void passing(){
    pesanTiketPage ptp = new pesanTiketPage();
    String film = ptp.judulFilm;
    judul.setText( text:film);
}

private void jamayang(){
    if(judul.getText() == "Fifty Shades Darker"){
        pilihjam.removeItemAt( anIndex:1);
        pilihjam.removeItemAt( anIndex:1);
        pilihjam.removeItemAt( anIndex:1);
        b = film1;
    } else if (judul.getText() == "The Conjuring 2"){
        pilihjam.removeItemAt( anIndex:0);
        pilihjam.removeItemAt( anIndex:1);
        pilihjam.removeItemAt( anIndex:1);
        b = film2;
    } else if (judul.getText() == "Cocomelon"){
        pilihjam.removeItemAt( anIndex:0);
        pilihjam.removeItemAt( anIndex:0);
        pilihjam.removeItemAt( anIndex:1);
        b = film3;
    } else if (judul.getText() == "Avengers Endgame"){
        pilihjam.removeItemAt( anIndex:0);
        pilihjam.removeItemAt( anIndex:0);
        pilihjam.removeItemAt( anIndex:0);
        b = film4;
    }
    String testing2;
    int testing[] = new int[b.length()];
    int testing3;
    for(int i=0; i<b.length(); i++){
        if (i!=b.length()){
            testing2 = b.substring( beginIndex:i, i+1);
        }else {

```

```

107         testing2 = b.substring( beginIndex:i, i+1);
108     }else {
109         testing2 = b.substring( beginIndex:i);
110     }
111     testing3 = Integer.parseInt( s:testing2);
112     testing[i] = testing3;
113 }
114 a = testing;
115
116
117 }
118 String b;
119
120 int a[]={0,0,0,0,0,0,0,0,0};
121 ArrayList<String> al = new ArrayList<String>();
122 Vector kursiDatabase = new Vector();
123 String film1,film2,film3,film4;
124
125
126 private void read() {
127
128     if (a[0]==1) {
129         A1.setEnabled( b:false);
130     } else {
131         A1.setSelected( b:false);
132     }
133     if (a[1]==1) {
134         A2.setEnabled( b:false);
135     } else {
136         A2.setSelected( b:false);
137     }
138     if (a[2]==1) {
139         A3.setEnabled( b:false);
140     } else {
141         A3.setSelected( b:false);

```

```

141         A3.setSelected( b:false);
142     }
143     if (a[3]==1) {
144         B1.setEnabled( b:false);
145     } else {
146         B1.setSelected( b:false);
147     }
148     if (a[4]==1) {
149         B2.setEnabled( b:false);
150     } else {
151         B2.setSelected( b:false);
152     }
153     if (a[5]==1) {
154         B3.setEnabled( b:false);
155     } else {
156         B3.setSelected( b:false);
157     }
158     if (a[6]==1) {
159         C1.setEnabled( b:false);
160     } else {
161         C1.setSelected( b:false);
162     }
163     if (a[7]==1) {
164         C2.setEnabled( b:false);
165     } else {
166         C2.setSelected( b:false);
167     }
168     if (a[8]==1) {
169         C3.setEnabled( b:false);
170     } else {
171         C3.setSelected( b:false);
172     }
173
174     if (a[0]==1&&a[1]==1&&a[2]==1&&a[3]==1&&a[4]==1&&a[5]==1&&a[6]==1&&a[7]==1&&a[8]==1) {
175         JOptionPane.showMessageDialog( parentComponent:null, message:"Kursi penuh!");

```

```

175     JOptionPane.showMessageDialog( parentComponent:null, message: "Kursi penuh!");
176 }
177
178 }
179
180
181
182 /**
183  * This method is called from within the constructor to initialize the form.
184  * WARNING: Do NOT modify this code. The content of this method is always
185  * regenerated by the Form Editor.
186  */
187 @SuppressWarnings("unchecked")
188 Generated Code
367
368 private void backActionPerformed(java.awt.event.ActionEvent evt) {
369     // TODO add your handling code here:
370     pesanTiketPage ptp = new pesanTiketPage();
371     ptp.show();
372     dispose();
373 }
374
375 private void nextActionPerformed(java.awt.event.ActionEvent evt) {
376     // TODO add your handling code here:
377     bangku = (String.valueOf(obj.a1));
378     jam = pilihjam.getSelectedIndex().toString();
379     biaya = (jumlahkursi * 50000);
380
381     kursibiner = "";
382     for (int i=0; i<a.length;i++){
383         kursibiner = kursibiner + a[i];
384     }
385
386     if(bangku == ""){
387         JOptionPane.showMessageDialog( parentComponent:null, message: "Silahkan pilih kursi terlebih dahulu");
388     }

```

```

388     if(bangku == ""){
389         JOptionPane.showMessageDialog( parentComponent:null, message: "Silahkan pilih kursi terlebih dahulu");
390     } else {
391         cetakFilm cf = new cetakFilm();
392         cf.show();
393         dispose();
394     }
395
396 }
397
398 private void A1ActionPerformed(java.awt.event.ActionEvent evt) {
399     // TODO add your handling code here:
400     if(A1.isSelected() && a[0]== 0){
401         a[0]=1;
402         al.add( e:A1.getText());
403         pilihankursi.setText( t:String.valueOf( obj.a1));
404         jumlahkursi+=1;
405     } else {
406         a[0]=0;
407         al.remove( e:A1.getText());
408         pilihankursi.setText( t:String.valueOf( obj.a1));
409         jumlahkursi-=1;
410     }
411 }
412
413 private void A2ActionPerformed(java.awt.event.ActionEvent evt) {
414     // TODO add your handling code here:
415     if(A2.isSelected() && a[1]== 0){
416         a[1]=1;
417         al.add( e:A2.getText());
418         pilihankursi.setText( t:String.valueOf( obj.a1));
419         jumlahkursi+=1;
420     } else {
421         a[1]=0;

```



```

422         al.remove(o:A2.getText());
423         pilihankursi.setText(t:String.valueOf(obj:a1));
424         jumlahkursi-=1;
425     }
426 }
427
428 private void A3ActionPerformed(java.awt.event.ActionEvent evt) {
429     // TODO add your handling code here:
430     if(A3.isSelected() && a[2]== 0){
431         a[2]=1;
432         al.add(e:A3.getText());
433         pilihankursi.setText(t:String.valueOf(obj:a1));
434         jumlahkursi+=1;
435     } else {
436         a[2]=0;
437         al.remove(o:A3.getText());
438         pilihankursi.setText(t:String.valueOf(obj:a1));
439         jumlahkursi-=1;
440     }
441 }
442
443 private void B1ActionPerformed(java.awt.event.ActionEvent evt) {
444     // TODO add your handling code here:
445     if(B1.isSelected() && a[3]== 0){
446         a[3]=1;
447         al.add(e:B1.getText());
448         pilihankursi.setText(t:String.valueOf(obj:a1));
449         jumlahkursi+=1;
450     } else {
451         a[3]=0;
452         al.remove(o:B1.getText());
453         pilihankursi.setText(t:String.valueOf(obj:a1));
454         jumlahkursi-=1;
455     }
456 }

```

```

457
458 private void B2ActionPerformed(java.awt.event.ActionEvent evt) {
459     // TODO add your handling code here:
460     if(B2.isSelected() && a[4]== 0){
461         a[4]=1;
462         al.add(e:B2.getText());
463         pilihankursi.setText(t:String.valueOf(obj:a1));
464         jumlahkursi+=1;
465     } else {
466         a[4]=0;
467         al.remove(o:B2.getText());
468         pilihankursi.setText(t:String.valueOf(obj:a1));
469         jumlahkursi-=1;
470     }
471 }
472
473 private void B3ActionPerformed(java.awt.event.ActionEvent evt) {
474     // TODO add your handling code here:
475     if(B3.isSelected() && a[5]== 0){
476         a[5]=1;
477         al.add(e:B3.getText());
478         pilihankursi.setText(t:String.valueOf(obj:a1));
479         jumlahkursi+=1;
480     } else {
481         a[5]=0;
482         al.remove(o:B3.getText());
483         pilihankursi.setText(t:String.valueOf(obj:a1));
484         jumlahkursi-=1;
485     }
486 }
487
488 private void C1ActionPerformed(java.awt.event.ActionEvent evt) {
489     // TODO add your handling code here:
490     if(C1.isSelected() && a[6]== 0){
491         a[6]=1;
492         al.add(e:C1.getText());

```

```

492         al.add(e:C1.getText());
493         pilihankursi.setText(t:String.valueOf(obj:al));
494         jumlahkursi+=1;
495     } else {
496         a[6]=0;
497         al.remove(o:C1.getText());
498         pilihankursi.setText(t:String.valueOf(obj:al));
499         jumlahkursi-=1;
500     }
501 }
502
503 private void C2ActionPerformed(java.awt.event.ActionEvent evt) {
504     // TODO add your handling code here:
505     if(C2.isSelected() && a[7]== 0){
506         a[7]=1;
507         al.add(e:C2.getText());
508         pilihankursi.setText(t:String.valueOf(obj:al));
509         jumlahkursi+=1;
510     } else {
511         a[7]=0;
512         al.remove(o:C2.getText());
513         pilihankursi.setText(t:String.valueOf(obj:al));
514         jumlahkursi-=1;
515     }
516 }
517
518 private void C3ActionPerformed(java.awt.event.ActionEvent evt) {
519     // TODO add your handling code here:
520     if(C3.isSelected() && a[8]== 0){
521         a[8]=1;
522         al.add(e:C3.getText());
523         pilihankursi.setText(t:String.valueOf(obj:al));
524         jumlahkursi+=1;
525     } else {
526         a[8]=0;
527         al.remove(o:C3.getText());
528         pilihankursi.setText(t:String.valueOf(obj:al));
529         jumlahkursi-=1;
530     }
531 }
532
533 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
534     // TODO add your handling code here:
535     biaya = (jumlahkursi * 50000);
536     total.setText(t:String.valueOf(i:biaya));
537 }
538
539 /**
540  * @param args the command line arguments
541  */
542 public static void main(String args[]) {
543     /* Set the Nimbus look and feel */
544     Look and feel setting code (optional)
545
546     /* Create and display the form */
547     java.awt.EventQueue.invokeLater(new Runnable() {
548         public void run() {
549             new kursiPage().setVisible(true);
550         }
551     });
552 }
553
554 // Variables declaration - do not modify
555 private javax.swing.JToggleButton A1;
556 private javax.swing.JToggleButton A2;
557 private javax.swing.JToggleButton A3;
558 private javax.swing.JToggleButton B1;
559 private javax.swing.JToggleButton B2;
560 private javax.swing.JToggleButton B3;
561 private javax.swing.JToggleButton C1;

```

```

569     java.awt.EventQueue.invokeLater(new Runnable() {
570         public void run() {
571             new kursiPage().setVisible(true);
572         }
573     });
574
575     // Variables declaration - do not modify
576     private javax.swing.JToggleButton A1;
577     private javax.swing.JToggleButton A2;
578     private javax.swing.JToggleButton A3;
579     private javax.swing.JToggleButton B1;
580     private javax.swing.JToggleButton B2;
581     private javax.swing.JToggleButton B3;
582     private javax.swing.JToggleButton C1;
583     private javax.swing.JToggleButton C2;
584     private javax.swing.JToggleButton C3;
585     private javax.swing.JButton back;
586     private javax.swing.JLabel bg_kursiPage;
587     private javax.swing.JTextField hargafilam;
588     private javax.swing.JButton jButton2;
589     private javax.swing.JLabel jamTayang;
590     private javax.swing.JLabel judul;
591     private javax.swing.JLabel kursi;
592     private javax.swing.JButton next;
593     private javax.swing.JTextField pilihankursi;
594     private javax.swing.JComboBox<String> pilihjam;
595     private javax.swing.JTextField total;
596     private javax.swing.JLabel txt_harga;
597     private javax.swing.JLabel txt_total;
598     // End of variables declaration
599 }

```

6. Tampilan Skrip Cetak Film Page

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4  */
5
6  /**
7   *
8   * @author ikram
9   */
10 import java.sql.*;
11 import javax.swing.JOptionPane;
12 import javax.swing.table.DefaultTableModel;
13 public class cetakFilm extends javax.swing.JFrame {
14
15     /**
16      * Creates new form NewJFrame
17      */
18
19     public PreparedStatement ps;
20     Connection con = koneksi.koneksiDatabase.BukaKoneksi();
21     String kursibin;
22
23     public cetakFilm() {
24         initComponents();
25         passing();
26     }
27
28     private void passing() {
29         pesanTiketPage ptp = new pesanTiketPage();
30         String film = ptp.judulFilm;
31         jdlFilm.setText(text:film);
32
33
34         kursiPage kp = new kursiPage();
35         String kursi = kp.bangku;
36         String jam = kp.jam;

```

```

34      kursiPage kp = new kursiPage();
35      String kursi = kp.bangku;
36      String jamtyg = kp.jam;
37      int jumlahtiket = kp.jumlahkursi;
38      int jumlahtotal = kp.biaya;
39      kursibin = kp.kursibiner;
40
41      jamTayang.setText( text:jamtyg);
42      nilaiKursi.setText( text:kursi);
43      jmlhTiket.setText( text:String.valueOf(i:jumlahtiket));
44      total.setText( text:String.valueOf(i:jumlahtotal));
45
46  }
47
48  // private void test(){
49  //     String judulfilm = "";
50  //
51  //     if(jdlFilm.getText() == "Fifty Shades Darker"){
52  //         judulfilm = "Fifty Shades Darker";
53  //     } else if (jdlFilm.getText() == "The Conjuring 2"){
54  //         judulfilm = "The Conjuring 2";
55  //     }
56  // }
57
58  StringBuilder sb = new StringBuilder();
59
60  /**
61   * This method is called from within the constructor to initialize the form.
62   * WARNING: Do NOT modify this code. The content of this method is always
63   * regenerated by the Form Editor.
64   */
65  @SuppressWarnings("unchecked")
66  Generated Code
166
167  private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
168      // TODO add your handling code here:

```

```

168      private void simpanActionPerformed(java.awt.event.ActionEvent evt) {
169          // TODO add your handling code here:
170
171          int id = 0;
172
173          if(jdlFilm.getText() == "Fifty Shades Darker"){
174              id = 1;
175          } else if (jdlFilm.getText() == "The Conjuring 2"){
176              id = 2;
177          } else if (jdlFilm.getText() == "Cocomelon"){
178              id = 3;
179          } else if (jdlFilm.getText() == "Avengers Endgame"){
180              id = 4;
181          }
182
183          try {
184              ps = con.prepareStatement( sql:"INSERT INTO databasefilm(jamtayang, judulfilm, kursi, total) VALUES(?,?,?,?)");
185              ps.setString( parameterIndex:1, x:jamTayang.getText());
186              ps.setString( parameterIndex:2, x:jdlFilm.getText());
187              ps.setString( parameterIndex:3, x:nilaiKursi.getText());
188              ps.setString( parameterIndex:4, x:total.getText());
189
190              ps.executeUpdate();
191
192              JOptionPane.showMessageDialog( parentComponent:null, message:"data berhasil disimpan");
193
194
195              ps = con.prepareStatement( sql:"update databasekursi set judulfilm=?,kursibiner=? where id=?");
196              ps.setString( parameterIndex:1, x:jdlFilm.getText());
197              ps.setString( parameterIndex:2, x:kursibin);
198              ps.setInt( parameterIndex:3, x:id);
199
200              ps.executeUpdate();
201

```

```
205         } catch (Exception e) {
206             JOptionPane.showMessageDialog( parentComponent:null, message:e);
207             System.out.print( obj:e);
208         }
209
210
211         //         } else if (jdlFilm.getText() == "Cocomelon"){
212         //             judulfilm = "Cocomelon";
213         //         } else if (jdlFilm.getText() == "Avengers Endgame"){
214         //             judulfilm = "Avengers Endgame";
215         //         }
216
217
218
219
220         kursiPage kp = new kursiPage();
221         kp.jumlahkursi = 0;
222         menuPage mp = new menuPage();
223         mp.show();
224         dispose();
225
226     }
227
228     private void backActionPerformed(java.awt.event.ActionEvent evt) {
229         // TODO add your handling code here:
230         kursiPage kp = new kursiPage();
231         kp.jumlahkursi = 0;
232         kp.show();
233         dispose();
234
235     }
236
237     /**
238      * @param args the command line arguments
239      */
```

```
238     * @param args the command line arguments
239     */
240     public static void main(String args[]) {
241         /* Set the Nimbus look and feel */
242         Look and feel setting code (optional)
243         //</editor-fold>
244         //</editor-fold>
245         //</editor-fold>
246         //</editor-fold>
247         //</editor-fold>
248         //</editor-fold>
249         //</editor-fold>
250
251         /* Create and display the form */
252         java.awt.EventQueue.invokeLater(new Runnable() {
253             public void run() {
254                 new cetakFilm().setVisible(true);
255             }
256         });
257     }
258
259     // Variables declaration - do not modify
260     private javax.swing.JButton back;
261     private javax.swing.JLabel bg_cetakFilm;
262     private javax.swing.JLabel jamTayang;
263     private javax.swing.JLabel jdlFilm;
264     private javax.swing.JLabel jmlhTiket;
265     private javax.swing.JLabel judulFilm;
266     private javax.swing.JLabel nilaiKursi;
267     private javax.swing.JButton simpan;
268     private javax.swing.JLabel total;
269     private javax.swing.JLabel txtjamTayang;
270     private javax.swing.JLabel txtjmlhTiket;
271     private javax.swing.JLabel txtkursi;
272     private javax.swing.JLabel txttotal;
```

```

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        new cetakFilm().setVisible(true);
    }
});
}

// Variables declaration - do not modify
private javax.swing.JButton back;
private javax.swing.JLabel bg_cetakFilm;
private javax.swing.JLabel jamTayang;
private javax.swing.JLabel jdlFilm;
private javax.swing.JLabel jmlhTiket;
private javax.swing.JLabel judulFilm;
private javax.swing.JLabel nilaiKursi;
private javax.swing.JButton simpan;
private javax.swing.JLabel total;
private javax.swing.JLabel txtjamTayang;
private javax.swing.JLabel txtjmlhTiket;
private javax.swing.JLabel txtkursi;
private javax.swing.JLabel txttotal;
// End of variables declaration
}

```

7. Tampilan Skrip ListFoodPage

```

import java.util.ArrayList;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author HP
 */
public class listFoodPage extends javax.swing.JFrame {

    public static int totalPembayaran;
    public static String totalpesanan;

    /**
     * Creates new form listFoodPage
     */
    public listFoodPage() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    Generated Code

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        ...
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        menuPage mp = new menuPage();
        mp.show();
        dispose();
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        int popcorn, hotdog, mineral, cola;
        int totalpopcorn, totalhotdog, totalmineral, totalcola;
        ArrayList<String> pesanan = new ArrayList<String>();

        popcorn = (int) jmlhPopcorn.getValue();
        totalpopcorn = popcorn * 25000;

        hotdog = (int) jmlhHotdog.getValue();
        totalhotdog = hotdog * 20000;

        mineral = (int) jmlhMineral.getValue();
        totalmineral = mineral * 10000;

        cola = (int) jmlhCola.getValue();
        totalcola = cola * 15000;

        if (popcorn > 0 || hotdog > 0 || mineral > 0 || cola > 0) {
            pesanan.add(e: "Popcorn");
            pesanan.add(e: String.valueOf(i: popcorn));
            if (popcorn == 0) {
                pesanan.remove(e: "Popcorn");
                pesanan.remove(e: String.valueOf(i: popcorn));
            }
            pesanan.add(e: "Hotdog");
            pesanan.add(e: String.valueOf(i: hotdog));
            if (hotdog == 0) {

```

```

155         if (hotdog == 0) {
156             pesanan.remove(o: "Hotdog");
157             pesanan.remove(o: String.valueOf(i: hotdog));
158         }
159         pesanan.add(e: "Mineral");
160         pesanan.add(e: String.valueOf(i: mineral));
161         if (mineral == 0) {
162             pesanan.remove(o: "Mineral");
163             pesanan.remove(o: String.valueOf(i: mineral));
164         }
165         pesanan.add(e: "Cola");
166         pesanan.add(e: String.valueOf(i: cola));
167         if (cola == 0) {
168             pesanan.remove(o: "Cola");
169             pesanan.remove(o: String.valueOf(i: cola));
170         }
171     }
172
173     totalpesanan = String.valueOf(obj.pesanan);
174
175
176     totalPembayaran = totalpopcorn+totalhotdog+totalmineral+totalcola;
177
178     paymentTunaiPage ptp = new paymentTunaiPage();
179     ptp.show();
180     dispose();
181 }
182
183 private void jmlhPopcornAncestorAdded(javax.swing.event.AncestorEvent evt) {
184     // TODO add your handling code here:
185 }
186
187 private void jmlhPopcornAncestorMoved(javax.swing.event.AncestorEvent evt) {
188     // TODO add your handling code here:
189 }
190

```

```

186 private void jmlhPopcornAncestorMoved(javax.swing.event.AncestorEvent evt) {
187     // TODO add your handling code here:
188 }
189
190
191 /**
192  * @param args the command line arguments
193  */
194 public static void main(String args[]) {
195     /* Set the Nimbus look and feel */
196     Look and feel setting code (optional)
197
198     /* Create and display the form */
199     java.awt.EventQueue.invokeLater(new Runnable() {
200         public void run() {
201             new listFoodPage().setVisible(true);
202         }
203     });
204 }
205
206 // Variables declaration - do not modify
207 private javax.swing.JLabel bg_listFoodPage;
208 private javax.swing.JButton jButton3;
209 private javax.swing.JButton jButton4;
210 private javax.swing.JLabel jLabel1;
211 private javax.swing.JLabel jLabel2;
212 private javax.swing.JLabel jLabel3;
213 private javax.swing.JLabel jLabel4;
214 private javax.swing.JSpinner jmlhCola;
215 private javax.swing.JSpinner jmlhHotdog;
216 private javax.swing.JSpinner jmlhMineral;
217 public static javax.swing.JSpinner jmlhPopcorn;
218 // End of variables declaration
219 }

```

8. Tampilan Skrip Pembayaran Makanan

```
2 import java.sql.Connection;
3 import java.sql.PreparedStatement;
4 import java.sql.ResultSet;
5 import java.util.ArrayList;
6 import javax.swing.JLabel;
7 import javax.swing.JOptionPane;
8 import javax.swing.JTextField;
9
10 /**
11  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
12  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
13  */
14
15 /**
16  *
17  * @author ikram
18  */
19 public class paymentTunaiPage extends javax.swing.JFrame {
20
21     /**
22      * Creates new form paymentTunaiPage
23      */
24
25     int bayar;
26     String pesanan;
27     public PreparedStatement ps;
28     public ResultSet rs;
29     Connection con = koneksi.koneksiDatabase.BukaKoneksi();
30
31     public paymentTunaiPage() {
32         initComponents();
33         passing();
34     }
35
36     private void passing(){
37         listFoodPage lfp = new listFoodPage();
38     private void passing(){
39         listFoodPage lfp = new listFoodPage();
40         bayar = lfp.totalPembayaran;
41         pesanan = lfp.totalPesanan;
42
43         totalbayar.setText( text:Integer.toString( i:bayar));
44         hasilpesanan.setText( text:pesanan);
45     }
46
47     /**
48      * This method is called from within the constructor to initialize the form.
49      * WARNING: Do NOT modify this code. The content of this method is always
50      * regenerated by the Form Editor.
51      */
52     @SuppressWarnings("unchecked")
53     Generated Code
54
55     private void prosesActionPerformed(java.awt.event.ActionEvent evt) {
56         // TODO add your handling code here:
57         int uang;
58         int kembalian;
59
60         uang = Integer.parseInt( s:jumlahuang.getText());
61
62         kembalian = (uang - bayar);
63
64         hasil.setText( text:String.valueOf( i:kembalian));
65     }
66
67     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
68         // TODO add your handling code here:
69         try {
70             ps = con.prepareStatement( sql:"INSERT INTO databasemakanan(pesanan, totalpembayaran) VALUES(?,?)");
71             ps.setString( parameterIndex:1, x:hasilpesanan.getText());
72         } catch (SQLException ex) {
73             JOptionPane.showMessageDialog(this, ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
74         }
75     }
76 }
```



```

149 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
150     // TODO add your handling code here:
151     try {
152         ps = con.prepareStatement( sql:"INSERT INTO databasemakanan (pesanan, totalpembayaran) VALUES(?,?)");
153         ps.setString( parameterIndex:1, x:hasilpesanan.getText());
154         ps.setString( parameterIndex:2, x:totalbayar.getText());
155
156         ps.executeUpdate();
157
158         JOptionPane.showMessageDialog( parentComponent:null, message: "data berhasil disimpan");
159     } catch (Exception e) {
160     }
161
162
163
164     menuPage mp = new menuPage();
165     mp.show();
166     dispose();
167 }
168
169 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
170     // TODO add your handling code here:
171     listFoodPage lpp = new listFoodPage();
172     lpp.show();
173     dispose();
174 }
175
176 private void jumlahuangActionPerformed(java.awt.event.ActionEvent evt) {
177     // TODO add your handling code here:
178 }
179
180 /**
181  * @param args the command line arguments
182  */
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215 // Variables declaration - do not modify
216 private javax.swing.JLabel hasil;
217 private javax.swing.JLabel hasilpesanan;
218 private javax.swing.JButton jButton2;
219 private javax.swing.JButton jButton3;
220 private javax.swing.JLabel jLabel13;
221 private javax.swing.JTextField jumlahuang;
222 private javax.swing.JButton proses;
223 private javax.swing.JLabel rp;
224 private javax.swing.JLabel rpl;
225 private javax.swing.JLabel totalbayar;
226 // End of variables declaration
227 }
228

```

9. Tampilan Skrip Pilih Data Aplikasi

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4  */
5
6  /**
7   *
8   * @author ikram
9   */
10 public class pilihDataPage extends javax.swing.JFrame {
11
12     /**
13      * Creates new form pilihDataPage
14      */
15     public pilihDataPage() {
16         initComponents();
17     }
18
19     /**
20      * This method is called from within the constructor to initialize the form.
21      * WARNING: Do NOT modify this code. The content of this method is always
22      * regenerated by the Form Editor.
23      */
24     @SuppressWarnings("unchecked")
25     // Generated Code
26
27     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
28         // TODO add your handling code here:
29         menuPage mp = new menuPage();
30         mp.show();
31         dispose();
32     }
33
34     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
35         // TODO add your handling code here:
36         databaseFilmPage dfp = new databaseFilmPage();
37
38     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
39         // TODO add your handling code here:
40         databaseFilmPage dfp = new databaseFilmPage();
41         dfp.show();
42         dispose();
43     }
44
45     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
46         // TODO add your handling code here:
47         databaseMakananPage dmp = new databaseMakananPage();
48         dmp.show();
49         dispose();
50     }
51
52     /**
53      * @param args the command line arguments
54      */
55     public static void main(String args[]) {
56         /* Set the Nimbus look and feel */
57         /* Look and feel setting code (optional) */
58
59         /* Create and display the form */
60         java.awt.EventQueue.invokeLater(new Runnable() {
61             public void run() {
62                 new pilihDataPage().setVisible(true);
63             }
64         });
65     }
66
67     // Variables declaration - do not modify
68     private javax.swing.JLabel bg_pilihdata;
69     private javax.swing.JButton jButton1;
70     private javax.swing.JButton jButton2;
71     private javax.swing.JButton jButton3;
72     // End of variables declaration
73 }
```

10. Tampilan Skrip Database Film Aplikasi

```
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
4   */
5
6  /**
7   *
8   * @author ikram
9   */
10 import java.sql.*;
11 import java.util.Vector;
12 import javax.swing.JOptionPane;
13 import javax.swing.table.DefaultTableModel;
14 public class databaseFilmPage extends javax.swing.JFrame {
15
16     /**
17      * Creates new form dataPelangganPage
18      */
19
20     public PreparedStatement ps;
21     public ResultSet rs;
22     Connection con = koneksi.koneksiDatabase.BukaKoneksi();
23
24     public databaseFilmPage() {
25         initComponents();
26         databaseFilm();
27         databaseKursi();
28     }
29
30     private void databaseFilm() {
31         int c;
32         try {
33             ps = con.prepareStatement("SELECT * FROM databasefilm");
34             rs = ps.executeQuery();
35             ResultSetMetaData Rss = rs.getMetaData();
36
37             c = Rss.getColumnCount();
38
39             DefaultTableModel Df = (DefaultTableModel)tblDatabaseFilm.getModel();
40             Df.setRowCount(0);
41
42             while(rs.next()) {
43                 Vector v = new Vector();
44                 for (int a=1; a<=c; a++){
45                     v.add(e:rs.getString(columnLabel:"id"));
46                     v.add(e:rs.getString(columnLabel:"jamtayang"));
47                     v.add(e:rs.getString(columnLabel:"judulfilm"));
48                     v.add(e:rs.getString(columnLabel:"kursi"));
49                     v.add(e:rs.getString(columnLabel:"total"));
50                 }
51                 Df.addRow(rowData:v);
52             }
53
54             } catch (Exception e) {
55             }
56         }
57     }
58
59     private void databaseKursi() {
60         int c;
61         try {
62             ps = con.prepareStatement("SELECT * FROM databasekursi");
63             rs = ps.executeQuery();
64             ResultSetMetaData Rss = rs.getMetaData();
65
66             c = Rss.getColumnCount();
67
68             DefaultTableModel Df = (DefaultTableModel)tblDataKursi.getModel();
69             Df.setRowCount(0);
70
```

```

72         while(rs.next()){
73             Vector v = new Vector();
74             for (int a=1; a<=c; a++){
75                 v.add( e:rs.getString( columnLabel:"id"));
76                 v.add( e:rs.getString( columnLabel:"judulfilm"));
77                 v.add( e:rs.getString( columnLabel:"kursibiner"));
78             }
79             Df.addRow( rowData:v);
80         }
81
82         } catch (Exception e) {
83         }
84     }
85 }
86
87 /**
88  * This method is called from within the constructor to initialize the form.
89  * WARNING: Do NOT modify this code. The content of this method is always
90  * regenerated by the Form Editor.
91  */
92 @SuppressWarnings("unchecked")
93 Generated Code
210
211 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
212     // TODO add your handling code here:
213     menuPage mp = new menuPage();
214     mp.show();
215     dispose();
216 }
217
218 private void btnResetKursiActionPerformed(java.awt.event.ActionEvent evt) {
219     // TODO add your handling code here:
220
221     DefaultTableModel Df = (DefaultTableModel)tblDataKursi.getModel();
222     int selectedIndex = tblDataKursi.getSelectedRow();

```

```

224 String reset = "000000000";
225
226 try {
227     int id = Integer.parseInt(s:Df.getValueAt(row:selectedIndex, column:0).toString());
228
229     ps = con.prepareStatement(sql:"update databasekursi set kursibiner=? where id=?");
230     ps.setString(parameterIndex:1, x:reset);
231     ps.setInt(parameterIndex:2, x:id);
232
233     ps.executeUpdate();
234
235     JOptionPane.showMessageDialog(parentComponent:null, message:"kursibiner berhasil di update");
236
237     databaseKursi();
238
239 } catch (Exception e) {
240     JOptionPane.showMessageDialog(parentComponent:null, message:e);
241     System.out.print(obj:e);
242
243 }
244
245
246
247
248 private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
249     // TODO add your handling code here:
250     DefaultTableModel Df = (DefaultTableModel)tblDatabaseFilm.getModel();
251     int selectedIndex = tblDatabaseFilm.getSelectedRow();
252
253     try {
254         System.out.println(selectedIndex);
255         int id = Integer.parseInt(s:Df.getValueAt(row:selectedIndex, column:0).toString());
256         int dialogResult = JOptionPane.showConfirmDialog(parentComponent:null, message:"Do you want to delete?", title:"Warning", optionType:JOptionPane.YES_NO_OPTION);
257         if(dialogResult == JOptionPane.YES_OPTION){
258             ps = con.prepareStatement(sql:"delete from databasefilm where id=?");

```

```

258         ps = con.prepareStatement(sql:"delete from databasefilm where id=?");
259         ps.setInt(parameterIndex:1, x:id);
260
261         ps.executeUpdate();
262
263
264         JOptionPane.showMessageDialog(parentComponent:null, message:"Data berhasil di hapus");
265
266         databaseFilm();
267     }
268
269
270
271 } catch (Exception e) {
272     JOptionPane.showMessageDialog(parentComponent:null, message:e);
273     System.out.print(obj:e);
274
275 }
276
277
278 /**
279  * @param args the command line arguments
280  */
281 public static void main(String args[]) {
282     /* Set the Nimbus look and feel */
283     Look and feel setting code (optional)
284     //</editor-fold>
285
286     /* Create and display the form */
287     java.awt.EventQueue.invokeLater(new Runnable() {
288         public void run() {
289             new databaseFilmPage().setVisible(true);
290         }
291     });
292 }
293
294
295

```

```

278  /**
279   * @param args the command line arguments
280   */
281  public static void main(String args[]) {
282      /* Set the Nimbus look and feel */
283      Look and feel setting code (optional)
284      //</editor-fold>
285
286      /* Create and display the form */
287      java.awt.EventQueue.invokeLater(new Runnable() {
288          public void run() {
289              new databaseFilmPage().setVisible(true);
290          }
291      });
292  }
293
294  // Variables declaration - do not modify
295  private javax.swing.JLabel bg_databaseFilmPage;
296  private javax.swing.JButton btnDelete;
297  private javax.swing.JButton btnResetKursi;
298  private javax.swing.JButton jButton1;
299  private javax.swing.JScrollPane jScrollPane1;
300  private javax.swing.JScrollPane jScrollPane2;
301  private javax.swing.JTable tblDataKursi;
302  private javax.swing.JTable tblDatabaseFilm;
303  // End of variables declaration
304  }

```

11. Tampilkan Skrip Database Makanan

```

2  import java.sql.Connection;
3  import java.sql.PreparedStatement;
4  import java.sql.ResultSet;
5  import java.sql.ResultSetMetaData;
6  import java.util.Vector;
7  import javax.swing.JOptionPane;
8  import javax.swing.table.DefaultTableModel;
9
10  /**
11   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
12   * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
13   */
14
15  /**
16   *
17   * @author ikram
18   */
19  public class databaseMakananPage extends javax.swing.JFrame {
20
21      /**
22       * Creates new form databaseMakananPage
23       */
24      public databaseMakananPage() {
25          initComponents();
26          databasemakanan();
27      }
28
29      public PreparedStatement ps;
30      public ResultSet rs;
31      Connection con = koneksi.koneksiDatabase.BukaKoneksi();
32
33
34      private void databasemakanan() {
35          int c;
36          try {

```

```

37         ps = con.prepareStatement( sql:"SELECT * FROM databasemakanan");
38         rs = ps.executeQuery();
39         ResultSetMetaData Rss = rs.getMetaData();
40
41         c = Rss.getColumnCount();
42
43         DefaultTableModel Df = (DefaultTableModel)tbl1databasemakanan.getModel();
44         Df.setRowCount( rowCount:0);
45
46         while(rs.next()){
47             Vector v = new Vector();
48             for (int a=1; a<=c; a++){
49                 v.add( e:rs.getString( columnLabel:"id"));
50                 v.add( e:rs.getString( columnLabel:"pesanan"));
51                 v.add( e:rs.getString( columnLabel:"totalpembayaran"));
52             }
53             Df.addRow( rowData:v);
54         }
55     } catch (Exception e) {
56     }
57 }
58
59 /**
60  * This method is called from within the constructor to initialize the form.
61  * WARNING: Do NOT modify this code. The content of this method is always
62  * regenerated by the Form Editor.
63  */
64 @SuppressWarnings("unchecked")
65 // Generated Code
66
132
133 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
134     // TODO add your handling code here:
135     menuPage mp = new menuPage();
136     mp.show();
137     dispose();

```

```

137     dispose();
138 }
139
140 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
141     // TODO add your handling code here:
142     DefaultTableModel Df = (DefaultTableModel)tbl1databasemakanan.getModel();
143     int selectedIndex = tbl1databasemakanan.getSelectedRow();
144
145     try {
146         System.out.println(selectedIndex);
147         int id = Integer.parseInt( s:Df.getValueAt( row:selectedIndex, column:0).toString());
148         int dialogResult = JOptionPane.showConfirmDialog( parentComponent:null, message:"Do you want to delete?", title:"Warning", optionType
149
150         if(dialogResult == JOptionPane.YES_OPTION){
151             ps = con.prepareStatement( sql:"delete from databasemakanan where id=?");
152             ps.setInt( parameterIndex:1, x:id);
153
154             ps.executeUpdate();
155
156
157             JOptionPane.showMessageDialog( parentComponent:null, message:"Data berhasil di hapus");
158
159             databasemakanan();
160         }
161
162     } catch (Exception e) {
163         JOptionPane.showMessageDialog( parentComponent:null, message:e);
164         System.out.print( obj:e);
165     }
166 }
167
168 }
169
170

```

```

170
171
172     /**
173      * @param args the command line arguments
174      */
175     public static void main(String args[]) {
176         /* Set the Nimbus look and feel */
177         Look and feel setting code (optional)
178
179         /* Create and display the form */
180         java.awt.EventQueue.invokeLater(new Runnable() {
181             public void run() {
182                 new databaseMakananPage().setVisible(true);
183             }
184         });
185     }
186
187     // Variables declaration - do not modify
188     private javax.swing.JLabel bg_databasemakanan;
189     private javax.swing.JButton jButton1;
190     private javax.swing.JButton jButton2;
191     private javax.swing.JScrollPane jScrollPane1;
192     private javax.swing.JTable tbl1databasemakanan;
193     // End of variables declaration
194 }

```

12. Tampilan koneksi Database MySQL

```

1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package koneksi;
6  import java.sql.*;
7  import javax.swing.JOptionPane;
8
9  /**
10   *
11   * @author ikram
12   */
13  public class koneksiDatabase {
14      Connection con;
15      public static Connection BukaKoneksi(){
16          try {
17              Class.forName("com.mysql.jdbc.Driver");
18              Connection con = DriverManager.getConnection("jdbc:mysql://localhost/cinema5", "root", "root");
19              return con;
20          } catch (Exception e) {
21              JOptionPane.showMessageDialog(null, message: e);
22              return null;
23          }
24      }
25
26  }
27
28

```


13. Tampilan XAMPP pada Aplikasi

XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

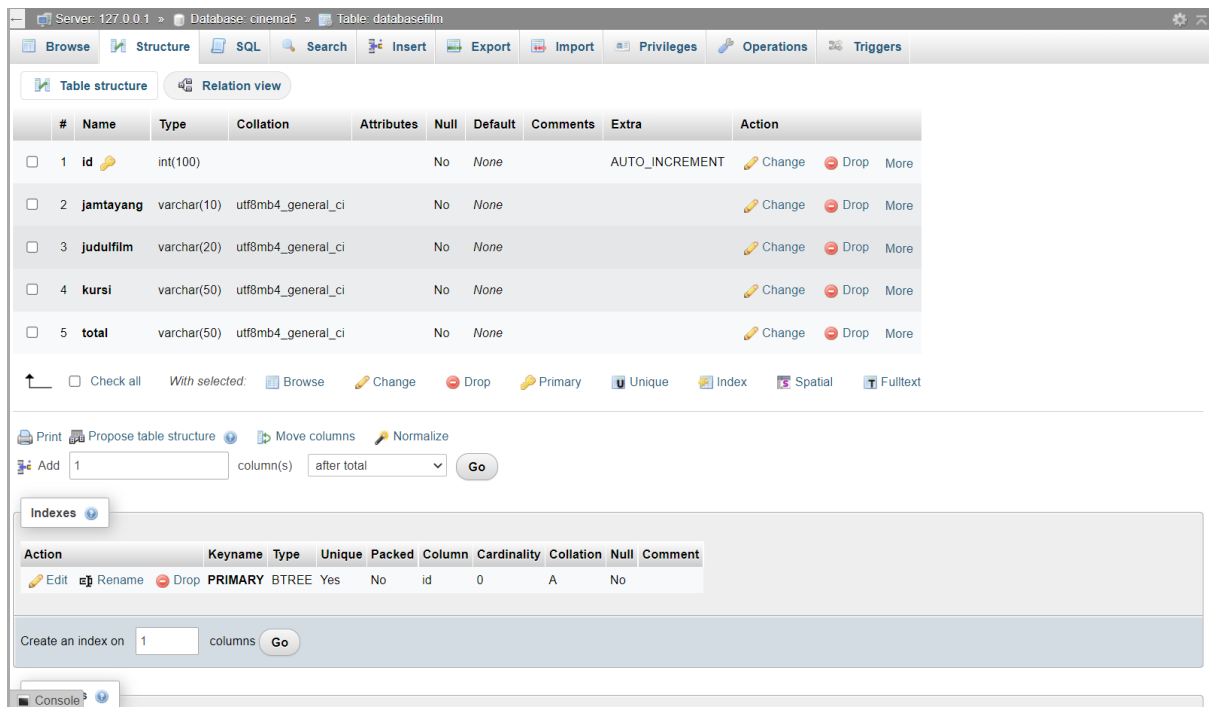
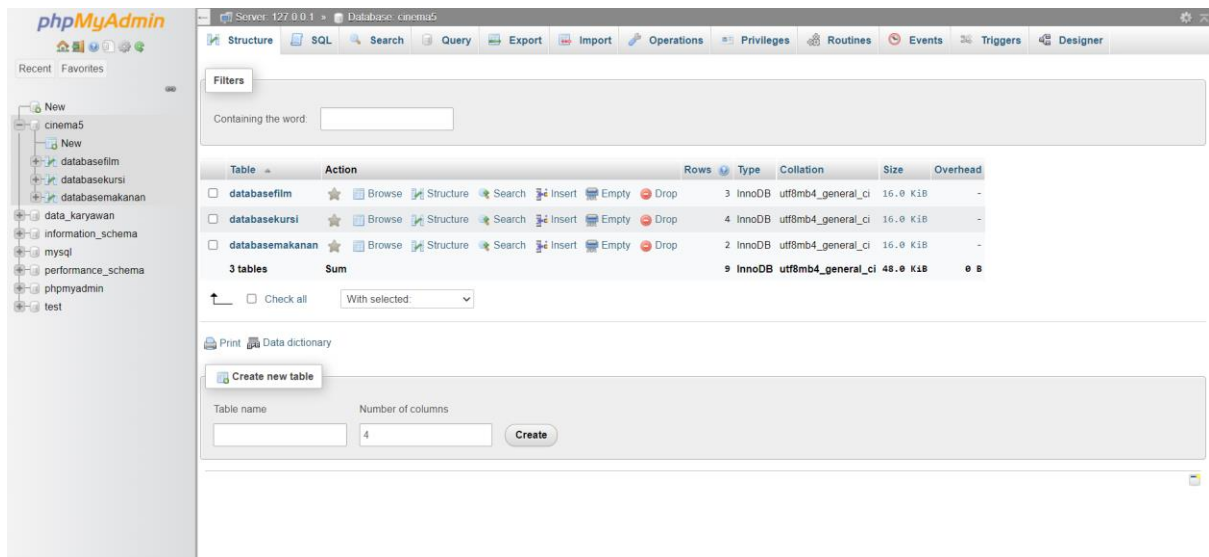
XAMPP Control Panel v3.3.0

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	15368 17396	80, 443	Stop Admin Config Logs
<input type="checkbox"/>	MySQL	14732	3306	Stop Admin Config Logs
<input type="checkbox"/>	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury			Start Admin Config Logs
<input type="checkbox"/>	Tomcat			Start Admin Config Logs

23.45.48 [main] All prerequisites found
23.45.48 [main] Initializing Modules
23.45.48 [main] Starting Check-Timer
23.45.48 [main] Control Panel Ready
23.45.54 [Apache] Attempting to start Apache app...
23.45.54 [Apache] Status change detected: running
23.45.55 [mysql] Attempting to start MySQL app...
23.45.55 [mysql] Status change detected: running

Config Netstat Shell Explorer Services Help Quit

14. Tampilan phpMyAdmin Database Cinema5



Server: 127.0.0.1 » Database: cinema5 » Table: databasেকursi

Browser Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

```
SELECT * FROM `databasেকursi`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	judulfilm	kursibiner
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	Fifty Shades Darker	000111000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	The Conjuring 2	000000000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	Cocomelon	000110000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	4	Avengers Endgame	000001000

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

Server: 127.0.0.1 » Database: cinema5 » Table: databasemakanan

Browser Structure SQL Search Insert Export Import Privileges Operations Triggers

Showing rows 0 - 1 (2 total, Query took 0.0002 seconds.)

```
SELECT * FROM `databasemakanan`
```

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

	id	pesanan	totalpembayaran
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	5	[Popcorn, 1, Hotdog, 1]	45000
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	[Popcorn, 2, Hotdog, 1, Mineral, 2, Cola, 1]	105000

☐ Check all | With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

☐ Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

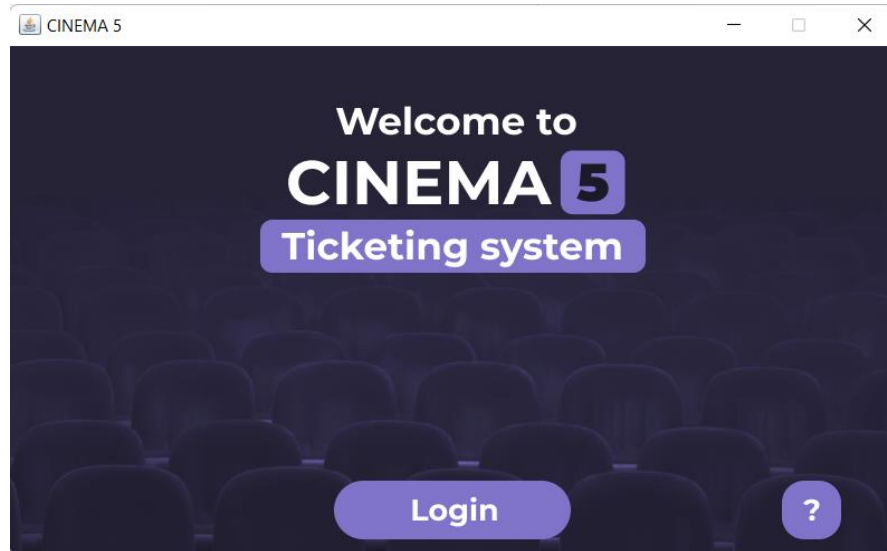
☐ Print ☐ Copy to clipboard ☐ Export ☐ Display chart ☐ Create view

Console

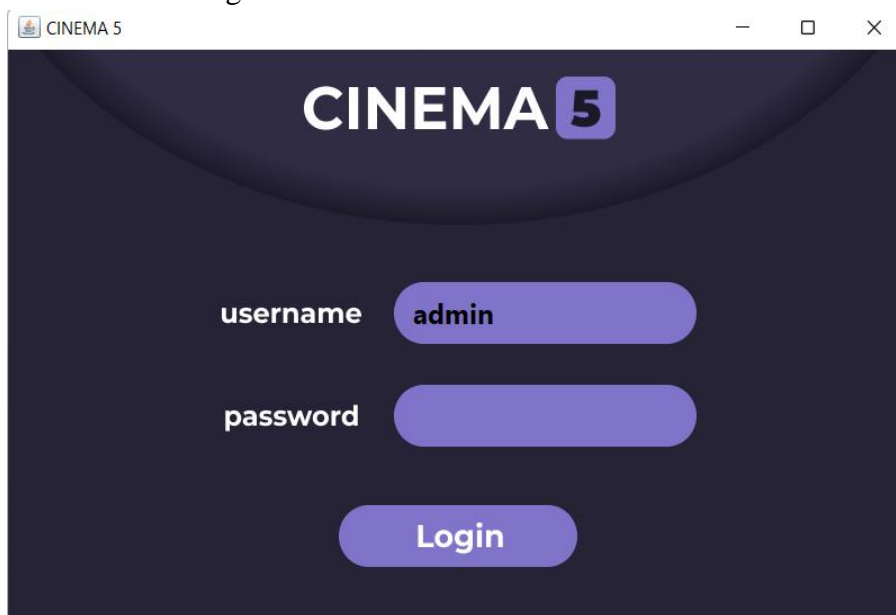
3.4 Dokumentasi Aplikasi

Dokumentasi pada aplikasi Cinema5, sebagai berikut.

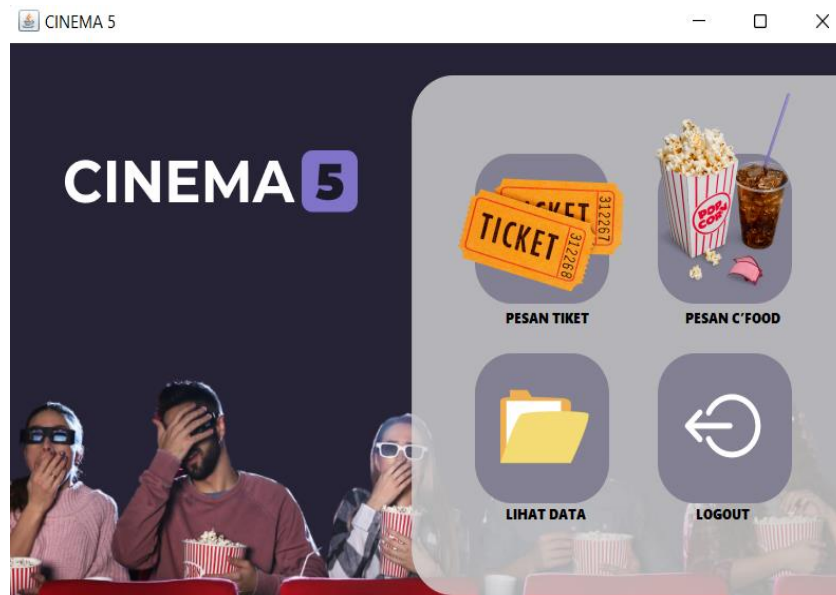
1. Halaman Awal



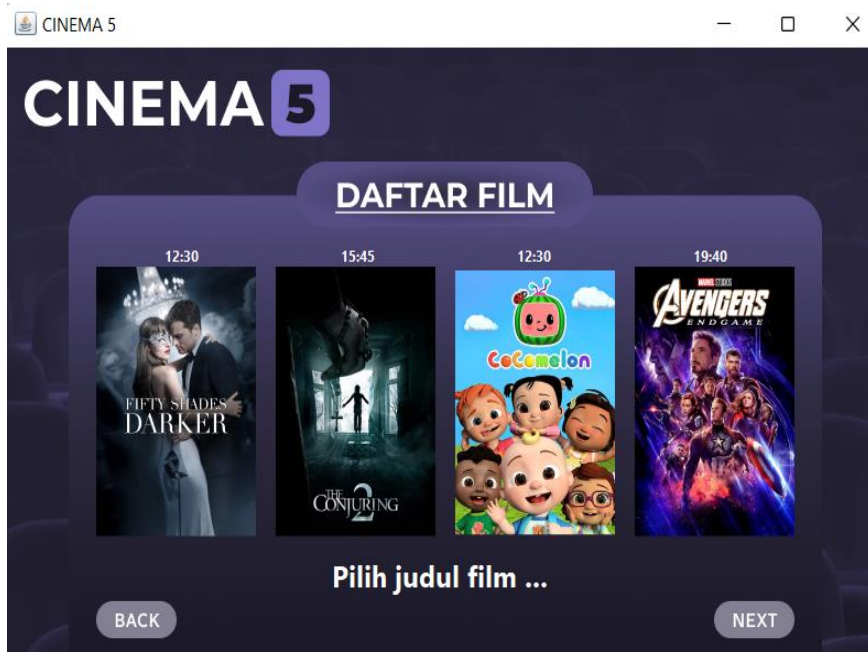
2. Halaman Login



3. Halaman Menu



4. Halaman Pesan Tiket



5. Halaman Pilih Kursi

Layar Bioskop

CINEMA 5

The Conjuring 2

Jam Tayang: 15:45

Kursi: [A1, B2, B1]

Harga: Rp. 50,000

Total: 150000

jumlah

BACK NEXT

6. Halaman Cetak Tiket Film

CINEMA 5

Judul Film: **The Conjuring 2**

Jam Tayang: **15:45**

Kursi: **[A1, B2, B1]**

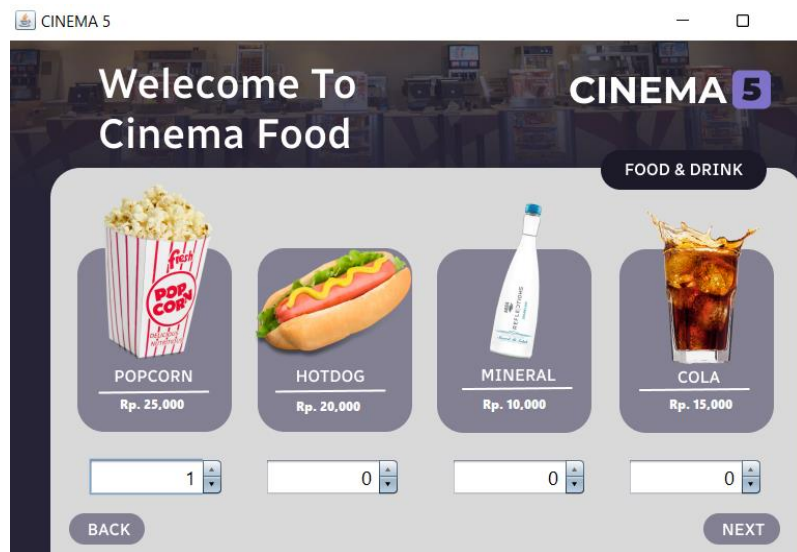
Jumlah Tiket: **3**

Total: **150000**

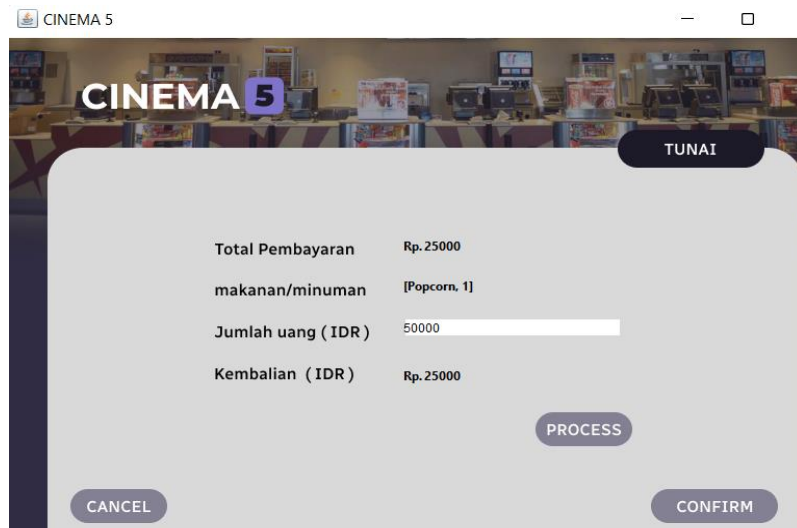
CINEMA 5

BACK SAVE

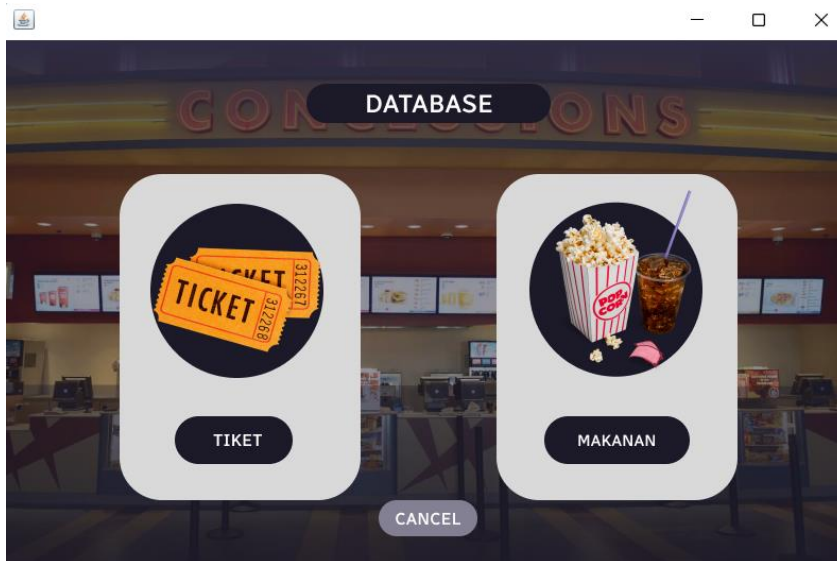
7. Halaman List Food



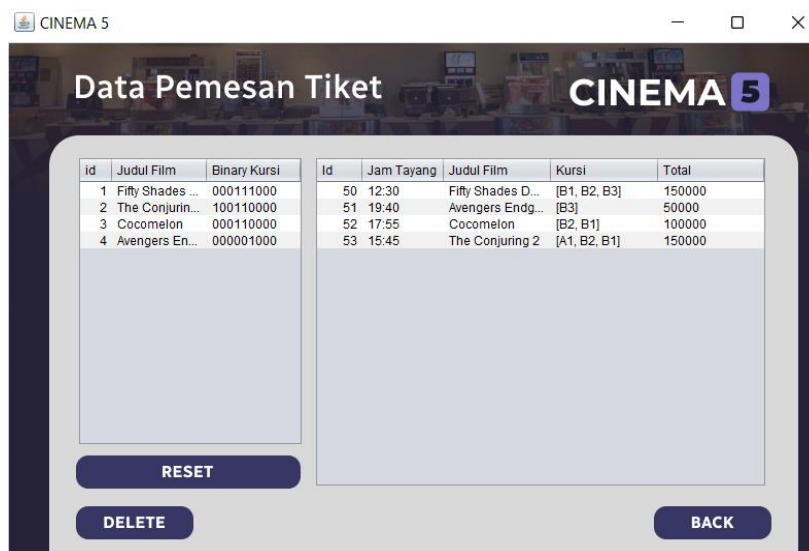
8. Halaman Pembayaran C'Food



9. Halaman Pilih Data



10. Halaman Database Film



11. Halaman Database Makanan



The screenshot shows a web application interface for managing food orders. The title is "Data Pemesan Makanan" and the logo is "CINEMA 5". The table lists the following data:

Id	Pesanan	Total Pembayaran
5	[Popcorn, 1, Hotdog, 1]	45000
7	[Popcorn, 2, Hotdog, 1, Mineral, 2, Cola, 1]	105000
8	[Popcorn, 1]	25000

At the bottom of the table area, there are two buttons: "DELETE" and "BACK".

BAB IV

PENUTUP

4.1 Kesimpulan

Dengan adanya program yang telah dibuat oleh kelompok kami, maka dapat diambil kesimpulan yaitu Aplikasi yang telah dibuat akan mempermudah dalam melakukan transaksi dan waktu yang terpakai dalam proses transaksi jauh lebih efisien pada pembelian tiket maupun pembelian c'food pada bioskop cinema 5

4.2 Kesimpulan

Program yang dibuat Oleh kelompok kami masih dapat dikembangkan menjadi lebih besar, karena kami menyadari bahwa program yang dibuat masih terdapat banyak kekurangan.

Beberapa saran yang dapat penulis sampaikan diantaranya:

1. Jenis film yang ditayangkan dapat ditambah menjadi lebih banyak
2. Kursi yang tersedia pada bioskop Cinema5 perlu ditingkatkan lagi
3. Jenis makanan atau minum pada cinema dapat ditambah lagi dengan menu bervariasi.
4. Program ini dapat didesain dengan tampilan yang lebih baik dan lebih menarik lagi.
5. Aplikasi dikembangkan dengan menyediakan fitur untuk pelanggan.
6. Database yang ditampilkan pada aplikasi dapat memuat hingga data tahunan pemasukan bioskop.