

# **SISTEM APLIKASI PEMESANAN TIKET BIOSKOP**

## **KELOMPOK 2**

**REJEKI H SIANTURI      231712096**

**PASKALIS SITUMORANG   231712090**

**JOHANES PANGGABEAN   231712100**

**GIDEON SIMANJUNTAK   231712097**

**YUBRADER SIDABALOK   231712094**

**KOM B2 '23**



**PROGRAM STUDI D-3 TEKNIK INFORMATIKA  
FAKULTAS VOKASI  
UNIVERSITAS SUMATERA UTARA  
MEDAN  
2025**

## **KATA PENGANTAR**

Puji dan syukur penulis panjatkan kepada Allah SWT, dengan limpah karunia-Nya penulis dapat menyelesaikan penyusunan makalah ini dengan judul Sistem Aplikasi Atm Bank Sederhana pada matakuliah Pemrograman Visual.

Terima kasih penulis sampaikan kepada setiap pihak yang sudah mendukung selama berlangsungnya pembuatan makalah ini. Terkhusus lagi penulis sampaikan terima kasih kepada Bapak Dr. Niskarto Zendrato. S.Kom., M.Kom. selaku dosen pengajar dan asisten laboratorium Nadira N Kuechler, semoga makalah ini bisa bermanfaat bagi setiap pembaca.

Disertai keseluruhan rasa rendah hati, kritik dan saran yang membangun amat penulis nantikan, agar nantinya penulis dapat meningkatkan dan merevisi kembali pembuatan makalah di tugas lainnya dan di waktu berikutnya.

Medan, ... November 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>iv</b>
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	1
1.3 Batasan Masalah .....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Komponen Penelitian .....	4
2.1.1 Pemrograman Visual.....	4
2.1.2 Visual Studio 2022.....	4
2.1.3 C#.....	5
2.1.4 Komponen Visual C#.....	5
2.1.5 MySQL.Data.....	6
2.2 Metode Pengerjaan .....	6
2.2.1 Metode Prototyping .....	6
2.2.2 Tahapan Metode Prototyping.....	7
<b>BAB III METODOLOGI PENELITIAN</b>	
3.1 Diagram Penelitian .....	10
3.1.1 Diagram Konteks .....	10
3. Aliran Data Utama .....	11
3.1.2 Flowchart .....	12
3.1.2 Diagram ERD Database.....	14
3.2 Jadwal Penelitian .....	15
<b>BAB IV HASIL DAN PEMBAHASAN</b>	
4.1 Pengertian dan Deskripsi Hasil Aplikasi.....	16
4.2 Tampilan & Panduan Pengguna .....	17
<b>BAB VPENUTUP</b>	
5.1 Kesimpulan.....	21
5.2 Saran .....	21
<b>DAFTAR PUSTAKA .....</b>	<b>22</b>
<b>LAMPIRAN.....</b>	<b>23</b>

## **DAFTAR GAMBAR**

Gambar 3.1.1 Diagram Konteks	10
Gambar 3.1.2 Flowchart	12
Gambar 3.1.3 Diagram ERD	14
Gambar 4.2.1 Form Login	17
Gambar 4.2.2 Form Registrasi	17
Gambar 4.2.3 Halaman Utama	18
Gambar 4.2.4 Halaman Pilih Jadwal	18
Gambar 4.2.5 Form Pemilihan Kursi	19
Gambar 4.2.6 Halaman Reviem dan Detail	20
Gambar 4.2.7 Halaman Riwayat Pemesanan	20

## **DAFTAR TABEL**

Tabel 3.1.1 Tabel Aliran data Diagram 11

Tabel 3.2 Tabel Jadwal Kegiatan 15

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi dan komunikasi (TIK) telah mengubah paradigma layanan dalam berbagai sektor, termasuk industri hiburan. Bioskop, sebagai salah satu sarana hiburan populer, dituntut untuk beradaptasi dengan kemudahan akses digital. Dalam konteks operasional konvensional, proses pembelian tiket secara fisik seringkali menimbulkan isu efisiensi, seperti antrian panjang di loket yang berujung pada penurunan kepuasan konsumen dan membuang waktu. Selain itu, calon penonton menghadapi ketidakpastian mengenai ketersediaan kursi, jadwal tayang, atau jenis studio tanpa harus datang langsung ke lokasi. Masalah ini semakin relevan dalam lingkungan akademis, di mana mahasiswa pengembang memerlukan studi kasus sistem informasi yang terintegrasi penuh—meliputi otentikasi pengguna (Login), manajemen data real-time (Film, Jadwal, Kursi), dan pencatatan transaksi (History)—sebagai media pembelajaran komprehensif. Oleh karena itu, dikembangkanlah Sistem Aplikasi Pemesanan Tiket Bioskop berbasis desktop dengan nama " CINEMATIX ". Aplikasi ini dirancang untuk mengatasi inefisiensi transaksi konvensional serta berfungsi sebagai sarana simulasi yang efektif bagi pengembang untuk memahami implementasi client-server sederhana, khususnya dalam visualisasi data menggunakan DataGridView dan ComboBox serta penanganan logika bisnis pemesanan dan riwayat transaksi.

### **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah diuraikan, perumusan masalah dalam pengembangan aplikasi " CINEMATIX " ini adalah:

1. Bagaimana merancang dan membangun antarmuka aplikasi desktop pemesanan tiket bioskop yang interaktif dan mudah digunakan oleh pengguna?
2. Bagaimana mengimplementasikan fitur otentikasi (login), manajemen data film dan jadwal, serta pemilihan kursi secara real-time pada aplikasi " CINEMATIX "?

3. Bagaimana menerapkan logika bisnis dan koneksi database untuk memastikan proses validasi pemesanan, pencatatan transaksi, dan penampilan riwayat pemesanan (History) berjalan akurat dan aman?

### 1.3 Batasan Masalah

Agar pengembangan aplikasi ini terfokus dan sesuai dengan tujuan, maka batasan-batasan masalah yang ditetapkan adalah sebagai berikut:

1. Aplikasi dikembangkan dalam platform desktop (bukan berbasis web atau mobile) menggunakan bahasa pemrograman Visual C#.
2. Database yang digunakan hanya mencakup entitas utama seperti User, Film, Jadwal Tayang, Kursi, dan Transaksi.
3. Fitur transaksi yang disimulasikan terbatas pada: Login User, Pilih Film, Pilih Jadwal/Studio/Jumlah Tiket, Pilih Kursi, Detail Tiket, dan History Pemesanan.
4. Aplikasi ini tidak membahas integrasi dengan sistem pembayaran eksternal (misalnya: payment gateway). Proses konfirmasi pesanan diasumsikan berhasil setelah data disimpan(insert) ke dalam tabel Transaksi dan Detail Transaksi ke database.

### 1.4 Tujuan

Tujuan utama dari pengembangan Sistem Aplikasi Pemesanan Tiket Bioskop "SCREAM" ini adalah sebagai berikut:

1. Membangun Aplikasi Fungsional: Menghasilkan aplikasi desktop Visual C# yang memiliki antarmuka yang ramah pengguna dan dapat memfasilitasi seluruh alur transaksi pemesanan tiket secara terstruktur.
2. Mengimplementasikan Fitur Inti: Mengimplementasikan fitur-fitur fungsional utama, meliputi Otentikasi User (Login), Pemilihan Film/Jadwal menggunakan DataGridView dan ComboBox, Pemilihan Kursi real-time, Konfirmasi Detail Tiket, dan Pencatatan Riwayat Transaksi (History).
3. Menguasai Koneksi Database: Menyediakan media studi kasus praktis untuk melatih pengembang dalam menguasai konsep koneksi database, khususnya operasi CRUD (Create, Read, Update, Delete) antara aplikasi desktop dan Database Server.

4. Menerapkan Logika Bisnis: Menguji dan menerapkan logika bisnis yang rigor, seperti validasi login, pengecekan ketersediaan kursi, dan perhitungan total harga tiket secara otomatis.

## **1.5 Manfaat**

Pengembangan aplikasi " CINEMATIX " ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Konsumen (Pengguna): Memberikan kemudahan, kecepatan, dan kepastian dalam proses pembelian tiket tanpa harus mengantri, serta akses instan terhadap riwayat pemesanan.
2. Bagi Bioskop: Menyediakan simulasi sistem yang dapat membantu meningkatkan efisiensi layanan, mengurangi human error dalam transaksi, dan menghasilkan data terstruktur untuk laporan manajemen.
3. Bagi Akademis (Pengembang): Berfungsi sebagai sarana studi kasus komprehensif untuk memahami siklus pengembangan sistem informasi, dari perancangan diagram (Flowchart, DFD) hingga implementasi kode Visual C# dan integrasi database.



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Komponen Penelitian**

Komponen penelitian ini mencakup konsep Pemrograman Visual sebagai dasar pembuatan antarmuka aplikasi melalui elemen GUI, penggunaan Visual Studio 2022 sebagai IDE utama berbasis 64-bit yang mendukung pengembangan aplikasi modern, pemanfaatan bahasa pemrograman C# sebagai bahasa inti yang fleksibel dan berorientasi objek, pemakaian komponen Visual C# seperti Code Editor, Form Designer, Solution Explorer, Toolbox, dan Properties Window untuk menunjang proses desain dan penulisan kode, serta penggunaan library MySQL.Data untuk menghubungkan aplikasi dengan database MySQL sehingga memungkinkan pengolahan data secara efektif.

##### **2.1.1 Pemrograman Visual**

Pemrograman visual merupakan sebuah paradigma dalam pengembangan perangkat lunak yang menekankan pembuatan aplikasi melalui antarmuka grafis atau GUI (*Graphical User Interface*). Konsep ini memungkinkan pengembang merancang tampilan aplikasi menggunakan komponen visual seperti tombol (*button*), label, *textbox*, panel, serta elemen grafis lainnya yang dapat disusun secara intuitif melalui proses *drag-and-drop* (Suhartono, n.d.). Dengan dukungan lingkungan IDE (*Integrated Development Environment*), pemrograman visual memberikan kemudahan bagi pengembang dalam membangun logika program, mengatur layout, dan mengelola alur aplikasi secara lebih efisien. Paradigma ini sangat membantu terutama bagi pemula karena struktur antarmuka dan alur aplikasi dapat terlihat secara langsung, sehingga proses belajar dan pengembangan menjadi lebih cepat serta terstruktur.

##### **2.1.2 Visual Studio 2022**

Visual Studio 2022 adalah sebuah *Integrated Development Environment* (IDE) yang dikembangkan oleh Microsoft untuk mendukung pembuatan berbagai tipe aplikasi seperti desktop, web, cloud, mobile, hingga game. Sebagai versi pertama yang

berbasis 64-bit, Visual Studio 2022 menawarkan performa lebih cepat, stabil, dan mampu menangani proyek berskala besar yang memerlukan kapasitas memori lebih tinggi (Microsoft, 2021). IDE ini dilengkapi fitur modern seperti IntelliSense yang semakin cerdas, debugging yang lebih komprehensif, integrasi Git, serta dukungan terhadap .NET 6/7 dan MAUI. Dengan keunggulan tersebut, Visual Studio 2022 menjadi alat penting bagi pengembang pemula hingga profesional dalam menghasilkan perangkat lunak berkualitas tinggi secara efisien.

### 2.1.3 C#

C# merupakan bahasa pemrograman modern yang dikembangkan oleh Microsoft dan berjalan pada platform .NET Framework maupun .NET Core. Bahasa ini dirancang dengan sintaks yang sederhana, aman, dan efisien sehingga mudah dipelajari oleh pemula namun tetap kuat digunakan dalam proyek skala besar (Microsoft, 2022). C# mendukung konsep *Object-Oriented Programming* (OOP), pengelolaan memori otomatis, dan dilengkapi pustaka (*library*) yang sangat luas. Dengan kemampuannya digunakan untuk berbagai jenis aplikasi—mulai dari desktop, web, mobile, cloud, hingga game menggunakan Unity—C# menjadi salah satu bahasa pemrograman yang paling populer dan fleksibel di lingkungan pengembangan modern.

### 2.1.4 Komponen Visual C#

Komponen Visual C# merupakan elemen-elemen penting dalam proses pengembangan aplikasi berbasis C# menggunakan Visual Studio. Komponen tersebut meliputi *Code Editor* untuk menulis dan mengelola kode, *Windows Form Designer* untuk merancang GUI secara visual, *Solution Explorer* untuk navigasi struktur proyek, serta *Properties Window* untuk pengaturan atribut objek. Selain itu, terdapat *Toolbox* yang berisi kontrol seperti *button*, *textbox*, *label*, dan komponen GUI lainnya yang dapat ditempatkan pada form secara *drag-and-drop*. Fitur Error List juga berperan dalam menampilkan pesan kesalahan dan membantu proses debugging (Microsoft, 2022). Keseluruhan komponen ini saling mendukung sehingga proses pembuatan aplikasi menjadi lebih terstruktur, cepat, dan mudah dipahami.

Berikut komponen-komponen C# (Windows Form / Visual C#) :

### 1.1 Form

Form adalah jendela atau wadah utama tempat meletakkan komponen visual dalam aplikasi Visual Basic. Form berfungsi sebagai area kerja pengguna untuk berinteraksi dengan aplikasi, tempat menempatkan objek objek seperti tombol, label, textbox, dan kontrol lainnya.

### 2.1 Command Button

Command Button adalah tombol perintah yang berfungsi untuk menjalankan aksi tertentu saat diklik oleh pengguna, misalnya tombol OK, Cancel, Exit, atau Apply.

### 3.1 Label

Label digunakan untuk menampilkan teks statis sebagai keterangan atau informasi pada form. Label biasanya berfungsi menjelaskan isi dari input atau komponen lain.

### 4.1 TextBox

TextBox adalah kotak teks yang memungkinkan pengguna memasukkan atau mengedit data.

## 2.1.5 MySQL.Data

MySQL.Data merupakan library resmi dari MySQL yang digunakan sebagai jembatan koneksi antara aplikasi C# dan database MySQL. Library ini menyediakan kelas penting seperti *MySqlConnection*, *MySqlCommand*, *MySqlDataReader*, hingga *MySqlDataAdapter*, yang memungkinkan aplikasi melakukan koneksi database, menjalankan query, membaca data, dan mengelola transaksi (Oracle, 2022). MySQL.Data biasanya dipasang melalui NuGet Package Manager di Visual Studio sehingga integrasinya mudah dilakukan. Library ini sangat penting terutama dalam pengembangan aplikasi yang memerlukan penyimpanan dan pengelolaan data secara terstruktur.

## 2.2 Metode Pengerjaan

### 2.2.1 Metode Prototyping

Metode Prototyping adalah metode pengembangan perangkat lunak yang dilakukan dengan cara membuat model awal (prototype) dari sistem yang akan dibangun, kemudian prototype tersebut diuji oleh pengguna, diperbaiki, dan

disempurnakan sampai sesuai kebutuhan. Prototype berfungsi sebagai gambaran awal tampilan, alur kerja, dan fungsi sistem sebelum dibuat versi finalnya. Metode ini cocok digunakan ketika kebutuhan sistem masih dapat berubah, atau ketika tampilan dan interaksi sangat penting seperti aplikasi Pemesanan Tiket Bioskop.

### 2.2.2 Tahapan Metode Prototyping

#### 1. Pengumpulan Kebutuhan (Requirement Gathering)

Pada tahap ini, dilakukan identifikasi kebutuhan fungsional dan non-fungsional aplikasi pemesanan tiket bioskop " CINEMATIX ".

- Login & Otentikasi: Menggunakan Username dan Password
- Menu Navigasi Utama: Tampilan untuk memilih film.
- Fitur Utama:
  - Pilih Film: Menampilkan daftar film yang sedang tayang (menggunakan DataGridView).
  - Pilih Jadwal: Memilih waktu tayang dan studio (ComboBox).
  - Pilih Kursi: Visualisasi layout kursi studio dan pemilihan kursi yang tersedia.
  - Riwayat Transaksi (History): Mencatat dan menampilkan tiket yang sudah dibeli.
- Tampilan Antarmuka: Tampilan form yang intuitif, modern, dan mudah dipahami, menyerupai aplikasi pemesanan tiket digital.
- Notifikasi: Notifikasi sukses/gagal pada proses Login dan Konfirmasi Pemesanan.

#### 2. Pembuatan Prototype

Tahap ini membuat rancangan awal sistem (simulasi) yang cepat berdasarkan kebutuhan di atas.

- Desain Form: Membuat rancangan tampilan form pada Visual C# (Form Login, Form Pemilihan Film, Form Kursi, Form Konfirmasi).
- Simulasi Alur: Membuat simulasi alur navigasi dari Login - Pilih Film - Pilih Jadwal - Pilih Kursi - Konfirmasi.
- Mock Data: Menggunakan data tiruan (dummy data) untuk daftar film, jadwal,

dan ketersediaan kursi.

- Visualisasi Data: Mengimplementasikan DataGridView untuk daftar film dan layout kursi, serta ComboBox untuk pilihan jadwal.
- Layout Antarmuka: Desain tata letak form dan tombol navigasi utama (Pesan Lagi, Lihat History, Logout).

### 3. Evaluasi Prototype oleh Pengguna

Pengguna (misalnya dosen pembimbing atau pengguna target) mencoba prototipe dan memberikan masukan kritis.

- Masukan Navigasi: Alur Pilih Kursi terasa terlalu panjang, navigasi kembali (Pesan Lagi) kurang jelas.
- Masukan Tampilan: Ukuran font di DataGridView daftar film terlalu kecil. Visualisasi kursi kurang jelas antara kursi yang sudah terisi dan kursi yang tersedia.
- Masukan Fungsional: Perlu ada validasi input jumlah tiket agar tidak melebihi kapasitas studio.
- Kesesuaian Fitur: Fitur History perlu ditambahkan tombol sortir berdasarkan tanggal.

### 4. Perbaikan Prototype (Refinement)

Developer memperbaiki prototipe berdasarkan feedback yang diterima pada tahap evaluasi.

- Penyempurnaan Layout: Menyesuaikan ukuran tampilan (UI) agar lebih responsif.
- Validasi Input: Menambahkan skrip C# sederhana untuk memastikan input jumlah tiket dan pemilihan kursi sudah sesuai.
- Optimalisasi Visual: Mengubah kode warna pada Layout Kursi untuk membedakan kursi tersedia dan terisi
- Finalisasi Tampilan: Menyempurnakan desain form agar menyerupai aplikasi pemesanan tiket profesional.

### 5. Pengembangan Sistem Sebenarnya

Jika prototipe sudah disetujui, sistem final yang terintegrasi penuh dibangun.

- Implementasi Logika C# Final: Menulis kode backend C# untuk semua logika

transaksi (penambahan Transaksi dan Detail Transaksi).

- Koneksi Database: Menghubungkan semua form dengan database (SQL Server/SQLite).
- Logika CRUD:
  - Validasi Login (Read data User).
  - Pengurangan Ketersediaan Kursi (Update status kursi setelah transaksi).
  - Pencatatan Transaksi (Create data baru di tabel Transaksi dan Detail Transaksi).
  - Tampilan History (Read data dari tabel Transaksi).

## 6. Pengujian Sistem

Dilakukan pengujian fungsional (functional testing) untuk memastikan seluruh fitur bekerja dengan benar sesuai logika bisnis.

- Pengujian Otentikasi: Tes Login dengan Username dan Password yang benar dan salah.
- Pengujian Transaksi Valid: Tes pemesanan tiket (pilih film, pilih kursi) dengan total harga yang dihitung dengan benar.
- Pengujian Transaksi Invalid: Mencoba memilih kursi yang sudah terisi atau memesan jumlah tiket melebihi kapasitas studio.
- Pengujian History: Memastikan riwayat transaksi baru tercatat dan dapat dilihat oleh User yang bersangkutan.
- Pengujian Tampilan: Memastikan layout DataGridView (Film dan Kursi) ditampilkan dengan benar di berbagai resolusi.

## 7. Implementasi dan Maintenance

Sistem dijalankan dalam bentuk aplikasi (.exe) yang dapat diinstal (atau dijalankan) oleh pengguna, kemudian dilakukan pemeliharaan.

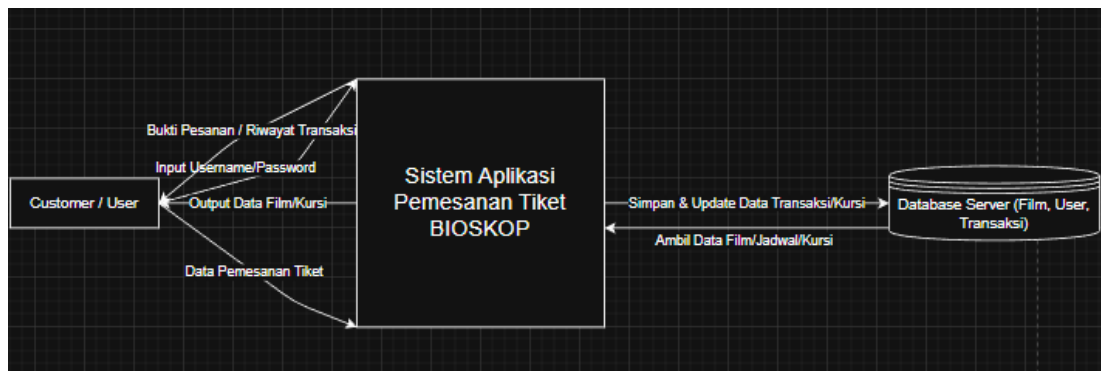
- Penerapan Sistem: Mendistribusikan aplikasi kepada pengguna target (misalnya, di lab komputer studi kasus).
- Perbaikan Bug: Mengatasi bug kecil yang mungkin muncul setelah sistem digunakan secara live.
- Penyesuaian Fitur: Melakukan penyesuaian minor jika ada perubahan kebijakan bisnis (misalnya penyesuaian skema harga).
- Pemeliharaan Database: Memastikan data transaksi tetap aman dan up-to-date.

## BAB III METODOLOGI PENELITIAN

### 3.1 Diagram Penelitian

Diagram penelitian digunakan untuk menggambarkan alur proses penelitian secara visual sehingga memudahkan pembaca memahami tahapan-tahapan yang dilakukan dalam pengembangan sistem. Diagram ini menjelaskan langkah-langkah mulai dari identifikasi masalah, pengumpulan data, perancangan sistem, implementasi, hingga pengujian. Dengan adanya diagram penelitian, alur kerja penelitian menjadi lebih terstruktur, sistematis, dan mudah dipahami. Pada subbagian ini ditampilkan dua diagram utama, yaitu flowchart yang menunjukkan proses penelitian secara berurutan, serta diagram konteks yang menggambarkan hubungan antara sistem yang dibangun dengan entitas luar yang terlibat.

#### 3.1.1 Diagram Konteks



Gambar 3.1.1 Diagram Konteks

Diagram Konteks adalah level tertinggi dari Data Flow Diagram (DFD). Tujuannya adalah untuk menunjukkan batasan sistem dan bagaimana sistem berinteraksi dengan dunia luar. Aplikasi Sistem Aplikasi Pemesanan Tiket BIOSKOP ini hanya berinteraksi dengan dua Entitas Eksternal melalui beberapa Aliran Data utama.

#### 1. Proses Sentral

- **Sistem Aplikasi Pemesanan Tiket BIOSKOP:** Ini adalah satu-satunya proses utama yang mewakili seluruh fungsionalitas aplikasi Anda, mulai dari login hingga penyimpanan transaksi.

#### 2. Entitas Eksternal

- Customer / User: Pihak yang menjalankan aplikasi dan merupakan sumber dan tujuan utama dari hampir semua data fungsional (seperti input pemesanan dan output bukti tiket).
- Database Server (Film, User, Transaksi): Repositori data utama. Ini adalah tempat semua data master (Film, User) dan data operasional (Transaksi) disimpan dan diambil.

### 3. Aliran Data Utama

Aliran data di Diagram Konteks ini menjelaskan interaksi *input* dan *output* sistem.

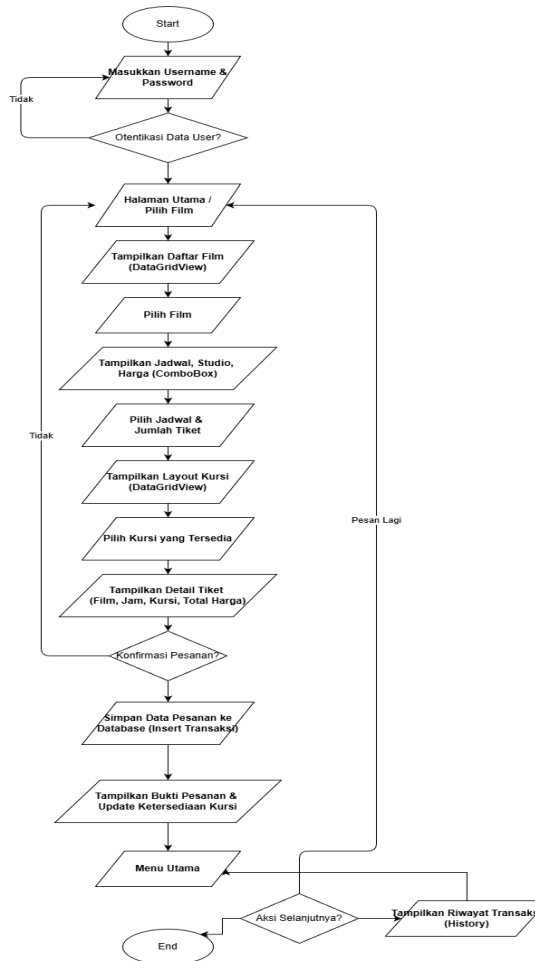
Arah	Aliran Data	Tujuan Fungsional
User > Sistem	Input Username/ Password	Data otentikasi yang diperlukan sistem untuk memvalidasi pengguna agar dapat mengakses fitur utama.
User > Sistem	Data Pemesanan Tiket	Data yang dikirimkan oleh pengguna setelah memilih film, jadwal, dan kursi. Ini adalah data mentah transaksi.
Sistem > User	Output Data Film/Kursi	Data yang dibaca sistem dari database dan ditampilkan ke pengguna (misalnya, daftar film di DataGridView atau layout ketersediaan kursi).
Sistem > User	Bukti Pesanan / Riwayat Transaksi	Konfirmasi keberhasilan pemesanan dan tampilan daftar tiket yang sudah dibeli (History).
Database > Sistem	Ambil Data Film/Jadwal/Kursi	Permintaan baca (Read) dari sistem ke database untuk menampilkan informasi yang up-to-date kepada pengguna.
Sistem > Database	Simpan & Update Data Transaksi/Kursi	Perintah tulis (Write) ke database. Ini mencakup Insert data transaksi baru dan Update status kursi yang telah dipesan.

Tabel 3.1.1 Tadel Aliran data Diagram



### 3.1.2 Flowchart

LinLogoutby



Gambar 3.1.2 Flowchart

Flowchart ini mendeskripsikan alur kerja aplikasi "CINEMATIX" secara sekuensial, yang terbagi dalam empat fase utama: Otentikasi, Pemilihan Tiket, Konfirmasi Transaksi, dan Penutupan Sistem.

a) FASE 1: Otentikasi Pengguna

1. Start: Aplikasi dimulai dan pengguna diarahkan ke Form Login.
2. Masukkan Username & Password: Pengguna melakukan input kredensial mereka.
3. Otentikasi Data User? (Keputusan):
  - Tidak: Jika data yang dimasukkan tidak ditemukan atau tidak cocok dengan database, pengguna dipaksa kembali ke langkah Masukkan Username & Password (alur loop di sisi kiri) hingga otentikasi berhasil.
  - Ya: Jika kredensial valid, proses berlanjut ke FASE 2.

## FASE 2: Pemilihan Tiket

4. Halaman Utama / Pilih Film: Pengguna berhasil masuk ke menu utama aplikasi.
5. Tampilkan Daftar Film (DataGridView): Sistem membaca data film yang sedang tayang dari Database Server dan menampilkannya di Form dalam komponen DataGridView (tabel).
6. Pilih Film: Pengguna memilih judul film yang diinginkan.
7. Tampilkan Jadwal, Studio, Harga (ComboBox): Sistem memuat dan menampilkan jadwal tayang, jenis studio (misalnya Reguler, VIP), dan harga tiket yang terkait dengan film yang dipilih, biasanya melalui ComboBox.
8. Pilih Jadwal & Jumlah Tiket: Pengguna memilih jadwal tertentu dan menentukan berapa jumlah tiket yang akan dipesan (ini akan mempengaruhi validasi kursi).
9. Tampilkan Layout Kursi (DataGridView): Sistem menampilkan representasi visual layout kursi studio, di mana status ketersediaan kursi dibaca dari database dan divisualisasikan (misalnya, kursi terisi berwarna abu-abu).
10. Pilih Kursi yang Tersedia: Pengguna melakukan pemilihan kursi satu per satu, dengan batasan jumlah tiket yang telah ditentukan.

## FASE 3: Konfirmasi dan Pencatatan Transaksi (Non-Pembayaran)

11. Tampilkan Detail Tiket (Film, Jam, Kursi, Total Harga): Sistem menampilkan ringkasan pesanan, menghitung total harga, dan menyajikan kode kursi yang dipilih.
12. Konfirmasi Pesanan? (Keputusan):
  - Tidak: Pengguna membatalkan transaksi dan diarahkan kembali ke Halaman Utama / Pilih Film untuk memulai proses dari awal.
  - Ya: Proses berlanjut ke pencatatan data.
13. Simpan Data Pesanan ke Database (Insert Transaksi): Pada titik ini, yang merupakan inti dari transaksi:
  - Sistem mencatat transaksi baru ke tabel TRANSAKSI dan DETAIL\_TRANSAKSI (operasi INSERT).
  - Perlu diingat: Tahap ini menandakan pesanan berhasil tanpa melibatkan payment gateway.

14. Tampilkan Bukti Pesanan & Update Ketersediaan Kursi: Setelah data berhasil disimpan:

- Sistem menampilkan notifikasi sukses dan bukti pesanan.
- Sistem menjalankan operasi UPDATE pada tabel KURSI di database untuk menandai kursi yang baru dibeli sebagai tidak tersedia.

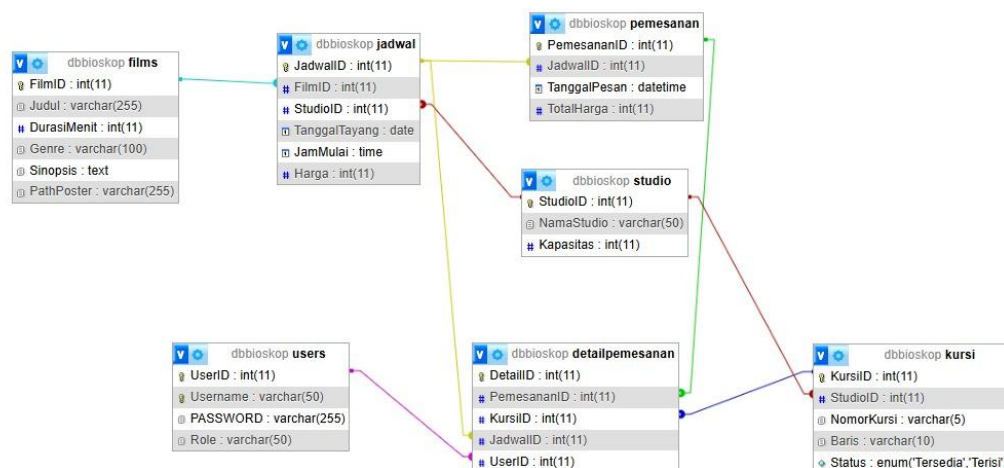
15. Menu Utama: Pengguna kembali ke menu setelah transaksi selesai.

#### FASE 4: Penutupan Sistem

16. Aksi Selanjutnya? (Keputusan): Setelah kembali ke Menu Utama, pengguna memiliki tiga pilihan:

- Pesan Lagi (Loop Samping): Pengguna diarahkan kembali ke proses Halaman Utama / Pilih Film untuk memulai pemesanan baru.
- Tampilkan Riwayat Transaksi (History): Pengguna dapat melihat daftar semua tiket yang pernah dibeli. Setelah selesai, pengguna kembali ke Menu Utama.
- End: Pengguna memilih untuk keluar dari aplikasi (Logout).

### 3.1.2 Diagram ERD Database



Gambar 3.1.3 Diagram ERD

### 3.2 Jadwal Penelitian

Penelitian ini kami lakukan dalam kurun waktu kurang lebih selama satu bulan dengan detail penjadwalan sebagai berikut:

	Perancangan Sistem & Kebutuhan	Desain Antarmuka (UI) & Database	Implementasi Logika Bisnis (C#)	Integrasi & Pengujian Fungsional	Finalisasi & Penulisan Laporan
Minggu 1					
Minggu 2					
Minggu 3					
Minggu 4					

Tabel 3.2 Tabel Jadwal Kegiatan

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Pengertian dan Deskripsi Hasil Aplikasi**

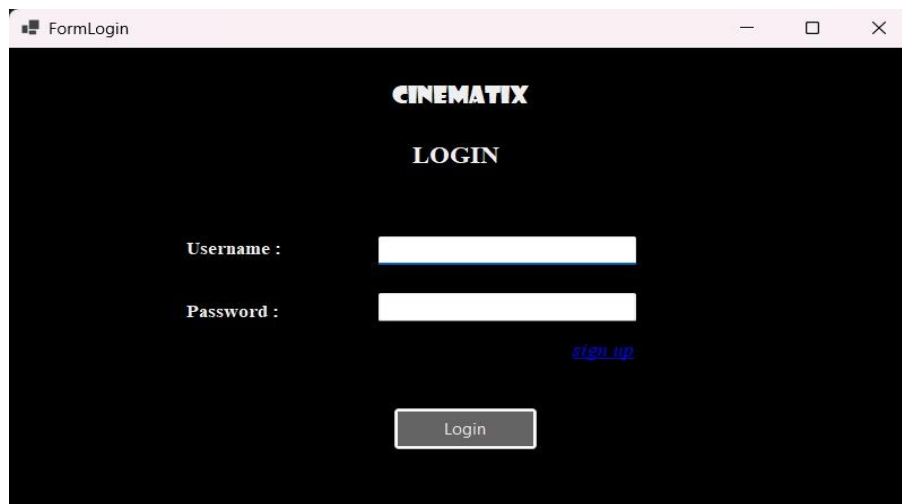
Hasil dari perancangan dan implementasi yang telah dilakukan adalah sebuah perangkat lunak aplikasi desktop bernama "CinemaTix". Aplikasi ini dibangun menggunakan bahasa pemrograman C# dengan kerangka kerja .NET 8.0 dan menggunakan MySQL sebagai sistem manajemen basis data. Aplikasi CinemaTix dirancang untuk mensimulasikan sistem pemesanan tiket bioskop modern. Secara garis besar, aplikasi ini memiliki karakteristik dan kemampuan sebagai berikut:

1. **Sistem Autentikasi:** Aplikasi memiliki sistem keamanan di mana pengguna harus melakukan Login untuk mengakses fitur utama. Tersedia pula fitur Sign Up untuk pendaftaran pengguna baru yang datanya langsung tersimpan di tabel Users database.
2. **Visualisasi Data Film:** Daftar film ditampilkan secara interaktif menggunakan komponen DataGridView yang telah dikustomisasi untuk menampilkan poster film, judul, durasi, dan genre secara rapi.
3. **Pemilihan Jadwal Terintegrasi:** Sistem mampu menampilkan jadwal tayang yang spesifik untuk film yang dipilih, lengkap dengan informasi waktu tayang dan lokasi studio.
4. **Manajemen Kursi Real-time:** Fitur unggulan aplikasi ini adalah denah kursi digital. Aplikasi dapat membedakan status kursi:
  - Tersedia (Hijau/Abu-abu): Kursi dapat dipilih oleh pengguna.
  - Terisi (Merah): Kursi telah dipesan pada transaksi sebelumnya dan tidak dapat dipilih kembali (dinonaktifkan secara otomatis).
  - Sedang Dipilih (Kuning): Kursi yang sedang diklik oleh pengguna saat ini.
5. **Pencatatan Transaksi (Struk):** Setiap pemesanan yang dikonfirmasi akan disimpan ke dalam dua tabel database yang saling berelasi (Pemesanan dan DetailPemesanan), memastikan integritas data penjualan.
6. **Riwayat Pemesanan:** Pengguna dapat melihat kembali tiket yang pernah dibeli melalui fitur History.

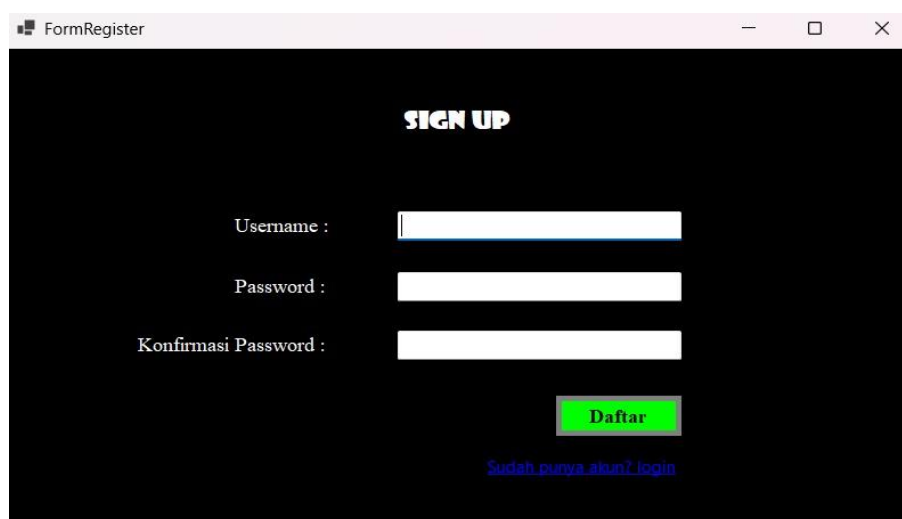
## 4.2 Tampilan & Panduan Pengguna

Bagian ini menjelaskan langkah-langkah penggunaan aplikasi CinemaTix mulai dari proses instalasi hingga penyelesaian transaksi.

1. Halaman Login dan Registrasi Saat aplikasi pertama kali dijalankan, pengguna akan diarahkan ke halaman Login.
  - Login: Masukkan Username dan Password yang sudah terdaftar, lalu klik tombol Login.
  - Registrasi: Jika belum memiliki akun, klik tautan "Daftar disini". Isi Username, Password, dan Konfirmasi Password pada form pendaftaran, lalu klik Daftar. Data akan tersimpan sebagai Customer.

The image shows a web browser window titled "FormLogin". The background is black. At the top center, the word "CINEMATIX" is written in white, bold, uppercase letters. Below it, the word "LOGIN" is also in white, bold, uppercase letters. There are two white input fields: the first is labeled "Username :" and the second is labeled "Password :". To the right of the password field, there is a blue link that says "sign up". Below the input fields is a grey button with the word "Login" in white text.

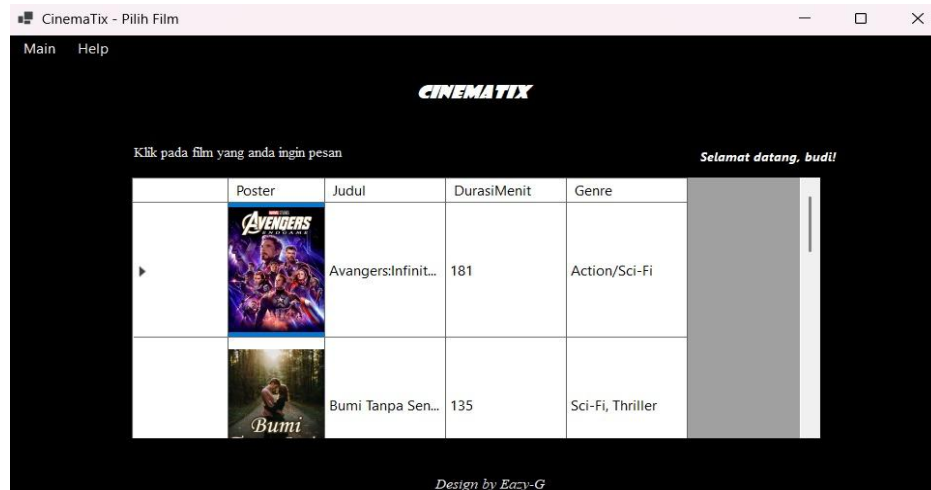
Gambar 4.2.1 Form Login

The image shows a web browser window titled "FormRegister". The background is black. At the top center, the words "SIGN UP" are written in white, bold, uppercase letters. There are three white input fields: the first is labeled "Username :", the second is labeled "Password :", and the third is labeled "Konfirmasi Password :". Below the input fields is a green button with the word "Daftar" in white text. At the bottom, there is a blue link that says "Sudah punya akun? login".

Gambar 4.2.2 Form Registrasi

2. Halaman Utama (Daftar Film) Setelah berhasil login, pengguna akan melihat Dashboard Utama.

- Halaman ini menampilkan daftar film yang sedang tayang (Now Playing).
- Setiap baris menampilkan Poster Film, Judul, Durasi, dan Genre.
- Pilih film yang diinginkan dengan mengklik pada baris film tersebut.



Gambar 4.2.3 Halaman Utama

3. Halaman Pemilihan Jadwal Pada halaman ini, pengguna menentukan detail pemutaran film.

- Film: Judul film yang dipilih sebelumnya akan tampil otomatis (Read-Only).
- Jadwal: Pilih waktu tayang pada ComboBox. Sistem akan otomatis menampilkan nama Studio yang sesuai dengan jam tersebut.
- Jumlah Tiket: Tentukan jumlah kursi yang ingin dipesan.
- Klik tombol Konfirmasi untuk lanjut ke pemilihan kursi.

**PESAN TIKET**

Film : Avengers:Infinity War

Jadwal : 12/10/2025 14:00

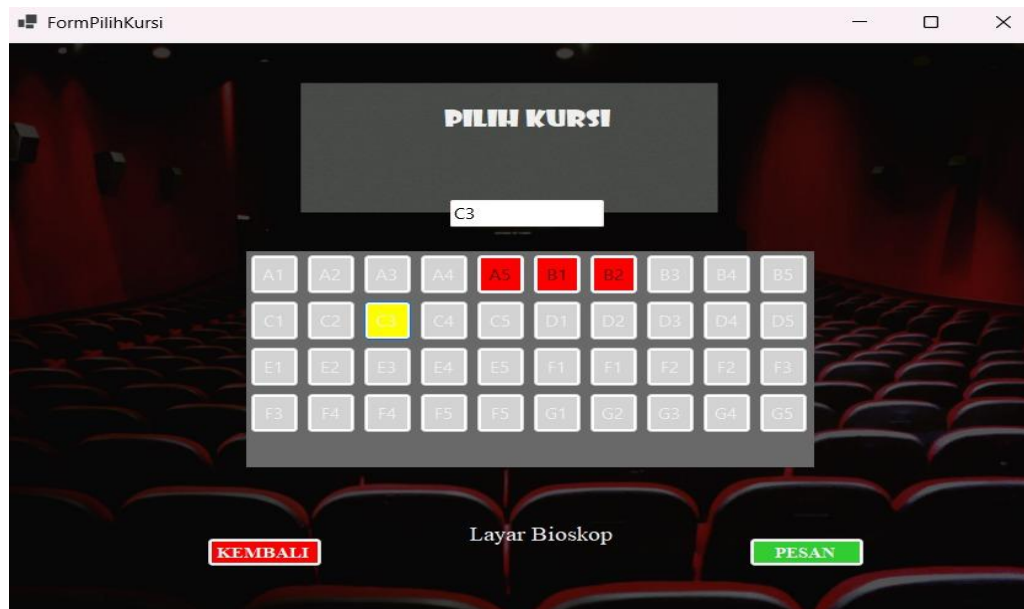
Jumlah : 1

Studio : Epic

**KEMBALI** **KONFIRMASI**

Gambar 4.2.4 Form Pilih Jadwal

4. Halaman Pemilihan Kursi Halaman ini menampilkan denah kursi sesuai dengan kapasitas studio (misal: Studio Epic 40 kursi, Studio Legend 30 kursi).
  - Pilih Kursi: Klik tombol kursi yang berwarna Hijau/Abu-abu. Kursi yang diklik akan berubah warna menjadi Kuning (Terpilih).
  - Validasi: Pengguna hanya dapat memilih kursi sejumlah tiket yang dimasukkan sebelumnya.
  - Kursi Terisi: Kursi yang berwarna Merah menandakan kursi tersebut sudah dibeli oleh orang lain dan tidak dapat diklik.
  - Klik tombol Pesan untuk melanjutkan ke pembayaran.



Gambar 4.2.5 Form Pemilihan Kursi

5. Halaman Detail dan Konfirmasi Halaman ini berfungsi sebagai review pesanan sebelum disimpan.
  - Periksa kembali Judul Film, Studio, Waktu, Nomor Kursi, dan Total Harga.
  - Jika data sudah benar, klik tombol KONFIRMASI.
  - Aplikasi akan menyimpan data ke database dan menampilkan pesan "Pemesanan Berhasil". Pengguna akan diarahkan kembali ke Halaman Utama.



**FormDetail**

**Detail Pemesanan**

**CINEMATIX**

Avangers:Infinity War      Studio : **1**

Jadwal : 2025-12-10

Waktu : **14:00**

Kursi : **C3**      Tipe : Epic

Harga : Rp. 50,000      Customer : budi

**KEMBALI**      **KONFIRMASI**

Gambar 4.2.6 Halaman Reviem dan Detail

6. Menu Histori dan Logout Melalui menu bar di bagian atas aplikasi:

- Histori: Klik menu Histori untuk melihat tabel riwayat tiket yang pernah dipesan oleh akun yang sedang login.
- Logout: Klik menu Logout untuk keluar dari akun dan kembali ke halaman Login.

**Riwayat Pemesanan - budi**

**RIWAYAT**

	PemesananID	TanggalPesanan	Films	Studio	TanggalTayang	JamMulai	TotalHarga
▶	5	11/30/2025 1:...	Avangers:Infin...	Epic	12/10/2025	14:00:00	50,000
	4	11/29/2025 5:...	Paw Patrol Th...	Legend	12/13/2025	15:30:00	300,000
	3	11/29/2025 1:...	Avangers:Infin...	Epic	12/10/2025	14:00:00	50,000
	2	11/28/2025 1:...	Avangers:Infin...	Epic	12/10/2025	14:00:00	50,000
	1	11/28/2025 1:...	Avangers:Infin...	Epic	12/10/2025	14:00:00	50,000

Gambar 4.2.7 Halaman Riwayat Pemesanan

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Berdasarkan seluruh tahapan perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa pengembangan Sistem Aplikasi Pemesanan Tiket Bioskop CINEMATIX telah berhasil mencapai seluruh tujuan yang ditetapkan. Aplikasi *desktop* ini telah berhasil dibangun menggunakan Visual C# dan menyediakan antarmuka yang ramah pengguna untuk memfasilitasi alur transaksi pemesanan tiket secara terstruktur. Secara fungsional, sistem telah mengimplementasikan fitur inti meliputi otentikasi pengguna melalui *Login*, visualisasi data film dan jadwal menggunakan DataGridView dan ComboBox, serta mekanisme pemilihan kursi *real-time*. Yang terpenting, aplikasi ini berhasil menguasai koneksi database melalui operasi CRUD untuk menerapkan logika bisnis yang rigor, memastikan validasi pemesanan dan pencatatan riwayat transaksi (History) berjalan akurat dan aman, sesuai dengan batasan yang telah ditetapkan (tidak melibatkan sistem pembayaran eksternal). Dengan demikian, CINEMATIX berhasil berfungsi sebagai media studi kasus praktis yang lengkap bagi pengembang.

#### **5.2 Saran**

Meskipun aplikasi CINEMATIX telah membuktikan fungsionalitasnya sebagai sistem simulasi pemesanan tiket, terdapat beberapa saran yang dapat diajukan sebagai bahan pertimbangan dan pengembangan di masa depan guna meningkatkan nilai guna aplikasi. Disarankan untuk melakukan pengembangan pada aspek fungsionalitas transaksi dengan mengintegrasikan sistem pembayaran virtual atau simulasi pembayaran menggunakan saldo internal, sehingga memberikan pengalaman transaksi yang lebih utuh. Selain itu, untuk meningkatkan efisiensi pengguna, perlu ditambahkan fitur pencarian dan filter pada daftar film di menu utama. Terakhir, pengembangan lebih lanjut dapat mencakup penambahan modul administrasi (Admin) untuk manajemen data *master* film, jadwal, dan pengguna, sehingga aplikasi dapat berfungsi sebagai sistem informasi terpadu, tidak hanya terbatas pada sisi *Customer*.

## DAFTAR PUSTAKA

<https://visualstudio.microsoft.com/vs/>  
<https://learn.microsoft.com/en-us/dotnet/csharp/>  
<https://learn.microsoft.com/en-us/visualstudio/designers/windows-forms-designer>  
<https://dev.mysql.com/doc/connector-net/en/>  
[https://github.com/Asisten-Laboratorium-Teknik-Informatika/Semester5\\_Vispro\\_2025-2026/tree/main/2\\_B2](https://github.com/Asisten-Laboratorium-Teknik-Informatika/Semester5_Vispro_2025-2026/tree/main/2_B2)

## **LAMPIRAN**