Project Business Application Development

Aplikasi Filmlogue

Rafael Febrian Kurniawan  
School of Information Systems, Information Systems Accounting and Auditing  
Binus UniversityJakarta, Indonesia  
rafael.kurniawan001@binus.ac.id

Dylan Nathanael  
School of Information Systems, Information Systems Accounting and Auditing  
Binus UniversityJakarta, Indonesia  
dylan.nathanael@binus.ac.id

Ernest Tan  
School of Information Systems, Information Systems Accounting and Auditing  
Binus UniversityJakarta, Indonesia  
ernest.tan@binus.ac.id

Pada paper ini akan membahas teori mengenai setiap kode dan metode yang digunakan untuk memperjelas unsur - unsur yang digunakan untuk membuat pemrograman aplikasi ini agar bisa berjalan dengan baik seperti penggunaan OOP pada kode, metode yang digunakan seperti enkapsulasi, *polymorphism*, GUI yang bisa masuk ke dalam unsur OOP, dan masih banyak lagi. Lalu, penjelasan akan dilanjutkan mengenai hal - hal yang digunakan dalam pembuatan aplikasi *Filmlogue* berdasarkan teori yang sudah dijelaskan. Unsur - unsur OOP yang digunakan akan dijelaskan tentang apa saja yang digunakan untuk membangun aplikasi *Filmlogue*. Pemrograman ini juga dihubungkan dengan database yang digunakan untuk menyimpan data dimana database dihubungkan melalui aplikasi lain yaitu *XAMPP Control* sehingga pemrograman tersebut bisa berjalan dan berguna untuk orang - orang. Serta, lengkap dengan penjelasan bagaimana pemrograman itu berjalan dan hasil evaluasi dari aplikasi pemrograman tersebut agar bisa mendapatkan sebuah kesan dan kesimpulan dari setiap tahap yang dijalankan.

Kata kunci — Pemrograman Berorientasi Objek, GUI, Basis Data, Pewarisan, Polimorfisme, Enkapsulasi, JavaFX

# **Pendahuluan**

**1.1 Latar belakang**

Film merupakan sebuah seni yang menggunakan sebuah gambar dengan gerakan yang bisa menyampaikan sebuah pesan ataupun cerita pada orang yang melihatnya. Pada era sekarang, film bisa menjadi sebuah alat komunikasi yang efektif dan film sendiri.

Begitu juga banyak jenis - jenis bioskop di Indonesia seperti bioskop modern di era sekarang seperti XXI, CGV, dan lain - lan. Ada juga bioskop tradisional yang masih menggunakan layar dan proyektor dan letaknya masih ada di pinggir jalan.

Melihat bioskop tradisional yang masih menggunakan cara tradisional untuk membeli tiket dan butuh waktu yang lama karena harus menulis satu - satu. Dari situ kami memiliki sebuah inovasi untuk membuat aplikasi tiket film yang bernama *FilmLogue*. Aplikasi ini dibuat memudahkan orang - orang yang menonton di bioskop tradisional agar mereka tidak perlu mengantri lama dan panjang. Aplikasi ini juga membantu orang yang memiliki usaha bioskop tradisional agar mereka lebih mudah menjual tiket mereka pada orang - orang yang sudah berlangganan.

**1.2 Tujuan**

Aplikasi ini dibuat dengan beberapa alasan berikut:

1. Untuk mengaplikasikan materi javaFX kedalam bentuk aplikasi yang bisa diintegrasikan dengan nyata
2. Memudahkan para pengusaha bioskop tradisional dalam menjualkan tiket mereka pada orang - orang yang ingin menonton
3. Menambah pengetahuan para mahasiswa akan penggunaan javaFX dalam kehidupan sehari - hari

# **Metodologi Penelitian**

Penelitian yang digunakan yaitu metode deskriptif bersama dengan R&D (*Research & Design*) yang bisa disebut juga Kualitatif. dimana metode ini menggunakan sebuah analisa pada aplikasi lain untuk mendapatkan fakta yang tepat selaras dengan tujuan agar mendapatkan sebuah gambaran dari aplikasi yang akan dibuat sehingga bisa menghasilkan produk baru yang lebih simple dan bisa digunakan untuk orang awam. tidak hanya dari aplikasi yang lain, tetapi bisa mencari referensi yang lain seperti lewat jurnal,artikel atau studi kasus yang berhubungan dengan projek. Untuk RAD, menggunakan metode 3 fase yaitu:

1. Kebutuhan yang direncanakan

Analisis kebutuhan yang diperlukan beserta kepentingan yang harus dimasukkan dalam aplikasi

1. Desain

Membuat sebuah desain pemrograman aplikasi dan database agar mudah dimengerti

1. Implementasi

Membuat sebuah prototype dari rencana yang sudah dibuat dengan menggunakan sebuah OOP dan GUI pada javaFX

1. **Dasar Teori**

**2.1 Pengertian O*bject Oriented Programming***

OOP adalah sebuah pemrograman yang menggunakan sebuah unsur dasar konsep objek. Pemrograman yang dijalankan sendiri bisa disebut dengan kumpulan objek yang saling terhubung dan berinteraksi satu dengan lainnya sehingga membuat pemrograman yang dibuat bisa berjalan. Setiap objek yang digunakan pada pemrograman ini memiliki karakteristik masing - masing yang bisa digunakan untuk membedakan mereka dan mengetahui fungsi dari mereka masing - masing. Setiap program yang dibentuk memiliki sebuah fungsi yang sama seperti cara manusia berpikir mengenai berjalannya sebuah program dan cara mereka berinteraksi antara satu dengan lainnya.

**2.2 Unsur Pada Object Oriented**

1. Object

Objek adalah sebuah bentuk konkrit yang bisa ditemukan pada setiap kelas yang kita buat. Objek ini juga menjadi sebuah hal yang penting dalam pemrograman karena menjadi dasar dalam berjalannya sebuah aplikasi. Objek ini memiliki keadaan dan perilaku yang berbeda - beda sesuai dengan kelas yang dibuat. Contohnya seperti objek “hewan” yang memiliki keadaan jenis kelamin dan jenis makanan, objek ini juga memiliki perilaku seperti bersembunyi atau metamorfosis yang membuat mereka memiliki ciri khas.

1. *Class(Kelas)*

Kelas merupakan sebuah blueprint yang menampung kumpulan objek yang berinteraksi satu dengan lainnya. Kelas ini selain bisa menampung objek - objek, kelas juga bisa mendefinisikan suatu atribut yang diimplementasikan dan metode yang digunakan untuk menjalankan para objek - objek tersebut sesuai dengan kelas yang dibentuk.

1. *Encapsulation*(Enkapsulasi)

Enkapsulasi merupakan sebuah prinsip yang ada di dalam OOP atau *Object Oriented Programming.* Enkapsulasi memiliki sebuah implementasi secara internal pada objek dan hanya menggunakan sesuai dengan fungsi. Enkapsulasi menjadi sebuah pembentuk batasan untuk mengurangi sebua ketergantungan dari satu objek mandiri sehingga objek lain bisa berdiri dengan fungsi masing - masing.

1. *Inheritance*(Pewarisan)

Pewarisan atau disebut juga *Inheritance* merupakan proses pewarisan data atau metode dari suatu class kepada class baru. Maksud nya adalah ketika suatu *class* yang merupakan Induk atau Parents memiliki suatu atribut atau method. maka akan diwariskan ke *class child* nya. class child nya akan berisi atribut dan method yang sama dengan parent nya tetapi pada masing pasing *child* memiliki atribut khusus yang membedakan dengan *class child* yang lain.

1. *Polymorphism*

*Polymorphism* adalah sebuah konsep OOP yang ada di java. berisikan class yang memiliki banyak nama method yang sama tetapi parameter yang berbeda. *Polymorphism* memiliki 2 jenis.

1. Static *Polymorphism* atau *Overloading* merupakan method class yang memiliki nama method yang sama tapi memiliki parameter dan tipe data yang berbeda.
2. Dynamic *Polymorphism* atau *Overriding* merupakan suatu kelas anak (child class) yang menyediakan implementasi ulang dari metode yang sudah didefinisikan di kelas induk (parent)

**2.3 Pengertian GUI**

GUI atau *Graphical User Interface* adalah sebuah interaksi yang dilakukan secara antarmuka dengan pengguna dengan komputer atau teknologi yang digunakan dimana bentuk dari GUI sendiri bisa seperti *Button*, *StackPane*, dan lain - lain. GUI memudahkan pengguna dalam menggunakan pemrograman atau aplikasi yang dibentuk sehingga pemrograman yang dibentuk bisa berinteraksi ketika pengguna memberikan respon seperti menekan tombol dan tampilan pemrograman bisa berpindah dari satu tampilan ke tampilan lainnya. GUI sendiri juga memiliki keterkaitan terhadap OOP seperti:

1. GUI dalam objek menjadi seperti tombol, label yang membantu untuk memahami jalannya pemrograman tersebut dan objek tersebut bisa menerima respon seperti tombol yang ditekan
2. GUI sendiri juga bisa menjadi bagian dari polymorphism dimana bisa diakses secara umum agar bisa saling berhubungan dengan kelas lainnya
3. GUI juga bisa menjadi bagian dari kapsulasi melalui penggunaan *Setter* dan *Getter* untuk memanggil sebuah objek yang diperlukan dalam menjalankan pemrograman

# **Detail Aplikasi**

**3.1 Unsur OOP yang digunakan**

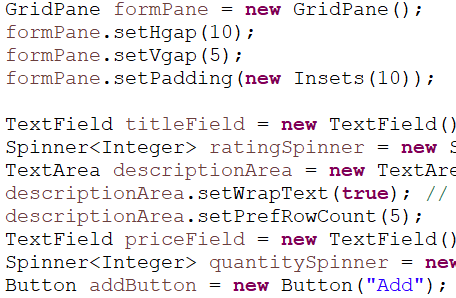
Pada aplikasi yang dibuat yaitu *FilmLogue* menggunakan oop berupa:

1. *Class*



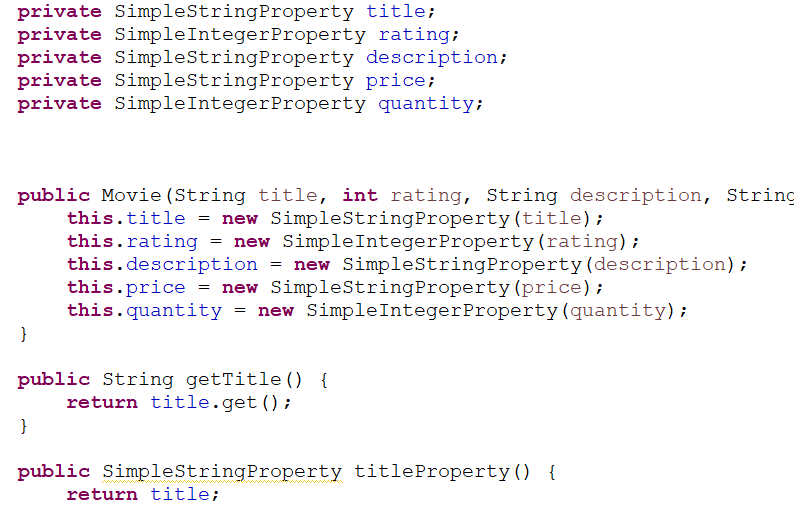
Aplikasi Filmlogue menggunakan kelas seperti kelas *Main* untuk menampung objek dan metode untuk mengarahkan pada tampilan *login* dan *register*. Kelas login dan register sendiri menampung objek dan metode yang berinteraksi agar kelas registrasi bisa mengarahkan pada tampilan login dan kelas *login* bisa mengarahkan pada tampilan kelas *Filmlogue* atau *Main menu* . Pda kelas *Filmlogue* ini menjadi tempat utama dari berjalannya aplikasi pemrograman karena berisi objek yang berhubungan dengan tiket dan metode untuk mengolah setiap objek yang ada. Serta, kelas *DatabaseManager* sendiri untuk menghubungkan setiap kelas yang ada dengan database untuk menyimpan setiap data yang di input

1. *Object*



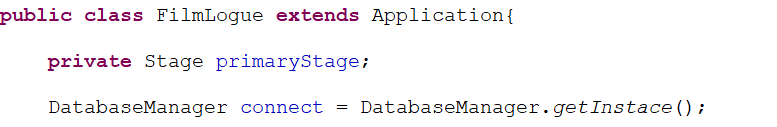
Object yang digunakan pada aplikasi *Filmlogue* ini cukup banyak mengingat ada 5 kelas yang dibentuk sehingga bentuk - bentuk objek yang digunakan seperti Button, Label, Textfield, dan masih banyak lagi

1. Encapsulation



Penggunaan encapsulation digunakan pada class movie yang kami buat untuk mendapatkan sebuah objek data yang sudah di input dimana dibuat menjadi private dan menggunakan Setter Getter untuk mengambil data atau objek tersebut.

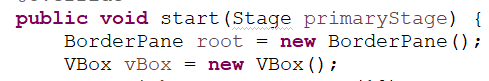
1. *Inheritance*



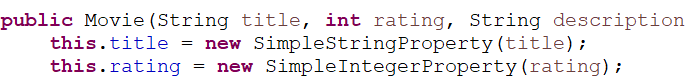
Penggunaan *inheritance* atau pewarisan digunakan pada kelas *Main*, *login*, *register*, dan *Filmlogue* dengan melakukan “*extends Application”* untuk mengakses setiap metode tanpa melakukan pengisian ulang yang membuat pemrograman menjadi tidak efektif

1. *Polymorphism*

*Polymorphism* yang digunakan pada pemrograman aplikasi ini menggunakan 2 jenis *polymorphism* yaitu metode *overriding* dimana mengimplementasikan “*primaryStage*” pada kelas induk agar bisa berpindah dari satu tampilan ke tampilan lain



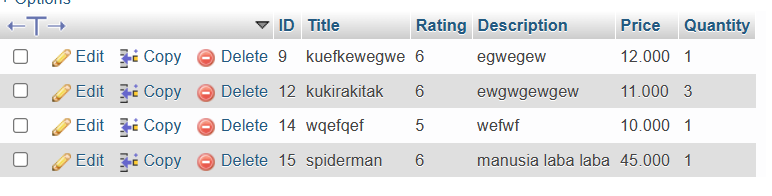
Pemrograman ini juga menggunakan metode *overloading* dalam polymorphism yaitu pada metode movie yang dibuat sama dengan kelas lain tetapi memiliki parameter objek yang berbeda

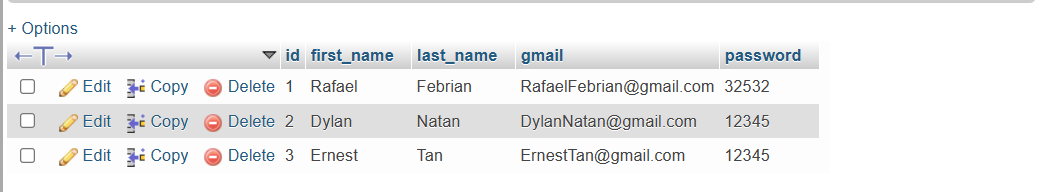


**3.2 Penggunaan Database Pada Aplikasi**

Database yang digunakan sendiri yaitu data *sql query* pada *phpmyAdmin* yang dihubungkan melalui *XAMPP control* panel agar setiap data yang diinput bisa tersimpan dan tidak hilang ketika aplikasi pemrograman ditutup. Database ini bekerja pada kelas *Filmlogue* untuk menyimpan data tiket yang diinput agar tidak hilang dan bisa tersimpan dengan baik,

Sedangkan pada *class login* dan *register*, database ini memiliki fungsi untuk menyimpan data dari sebuah *Gmail* dan *password* yang diinput pada kelas *register* dan akan divalidasi pada kelas *login* agar tidak sembarang orang bisa masuk aplikasi tersebut.





Sql query yang digunakan yaitu

CREATE TABLE Userss (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Users (

id INT PRIMARY KEY AUTO\_INCREMENT,

first\_name VARCHAR(50) NOT NULL,

last\_name VARCHAR(50) NOT NULL,

gmail VARCHAR(100) NOT NULL,

password VARCHAR(255) NOT NULL

);

CREATE TABLE Mainmenu (

ID INT PRIMARY KEY AUTO\_INCREMENT,

Title VARCHAR(255) NOT NULL,

Rating INT(11) NOT NULL,

Description VARCHAR(100),

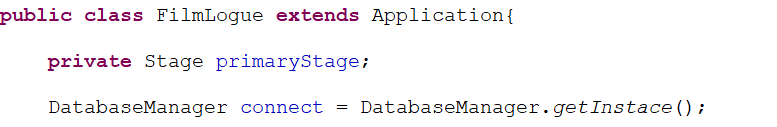
Price VARCHAR(200) NOT NULL,

Quantity INT(11) NOT NULL

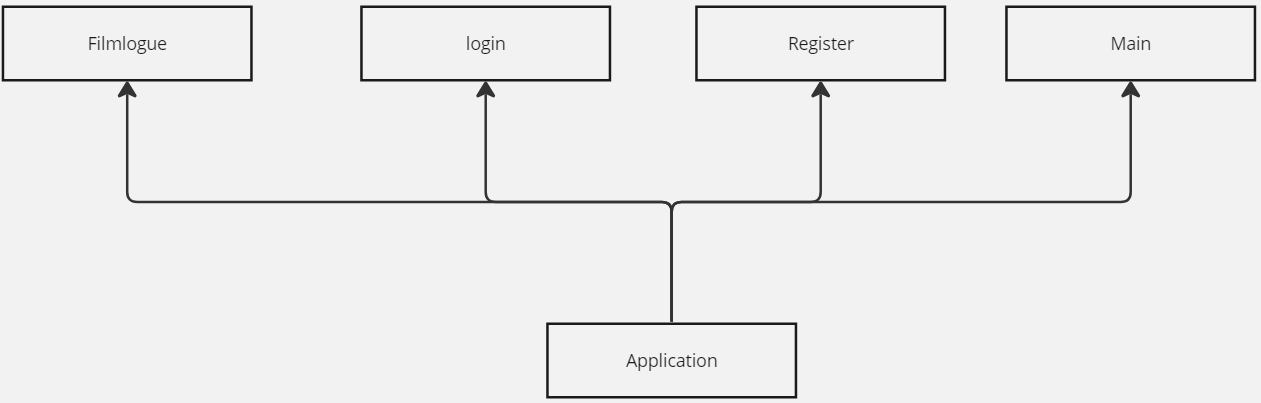
);

**3.3 *Inheritance* dan *Class Diagram***

Inheritance yang digunakan pada pemrograman yaitu ada pada “*extends Application*” yang diimplementasikan pada kelas *Main*, *login*, *register*, dan *Filmlogue* untuk menjalankan metode dan fungsi sehingga saat pemrograman itu dijalankan bisa berjalan dengan efektif tanpa memerlukkan implementasi kode yang sama lagi.



Class diagram yang bisa kami bentuk sendiri yaitu:.



Dimana class diagram ini menggambarkan sebuah hubungan *inheritance* yang ada pada pemrograman aplikasi. *Application* menjadi sebuah dasar untuk berdirinya dan berjalannya pemrograman dan 8 kelas tersebut menjadi turunan dari *Application* dimana menggunakan metode untuk saling berinteraksi antara satu dengan lainnya.

**3.4 Hasil Evaluasi Aplikasi**

Evaluasi dari aplikasi ini adalah dimana aplikasi untuk cukup membantu para pengusaha bioskop sederhana dalam penjualan tiket mereka dan meningkatkan efektivitas dari penjualan tiket tersebut karena mereka tidak perlu melakukan penjualan secara manual. Dari beberapa hasil survei dimana kami menanyakan bagaimana pendapat mereka tentang menggunakan aplikasi pemrograman kami untuk menjual tiket film dan sekitar 70% dari orang yang ditanyakan berkata mau menggunakan aplikasi tersebut. Orang - orang yang kami tanyakan diantaranya teman kuliah dan sekolah kami serta orang - orang yang sering pergi ke bioskop sederhana.

Aplikasi ini juga akan dikembangkan lagi dimana aplikasi ini akan dikembangkan fiturnya agar lebih lengkap dan lebih praktis digunakan contohnya seperti menambahkan fitur *genre* untuk mengkategorikan film - film dan denah duduk yang memudahkan pengusaha bioskop dalam menentukan tempat duduk penonton mereka.

Aplikasi ini juga akan dioptimasi dalam berjalannya waktu agar lebih responsif dan mengurangi tingkat *bug* yang bisa membuat pemrograman menjadi *error* atau tidak sesuai dengan fungsi.

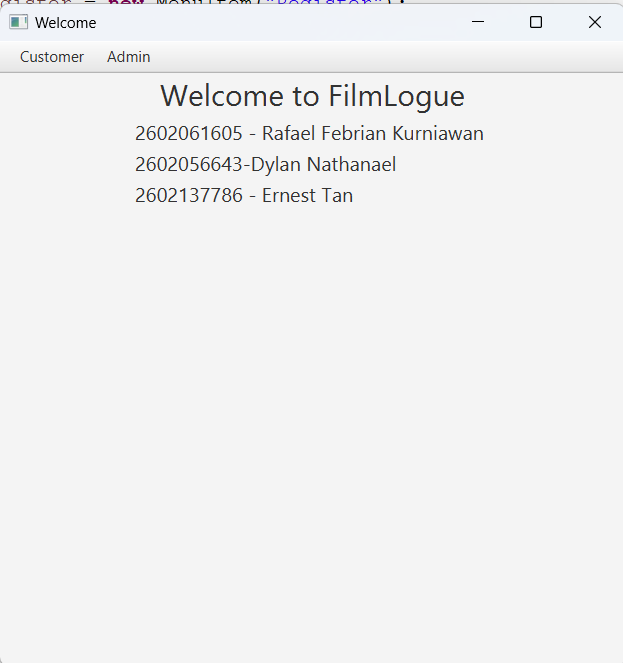
**3.5 Daya Tarik Aplikasi**

Tingkat kesulitan yang dialami yaitu mengetahui kode yang benar agar pemrograman bisa berjalan sesuai dengan tujuan dan menghindari *error* pada penggunaan objek karena kesalahan kode. *Syntax* juga menjadi salah satu kesulitan yang dialami karena jika pemrograman yang dijalankan mengalami kesalahan pada *syntax* maka pemrograman tersebut bisa saja mengalami *debugging*.Kesulitan yang sering dialami juga ada pada *error* yang muncul ketika pemrograman sedang dijalankan dimana *error* tersebut harus cepat - cepat ditangani dan mengetahui penyebab dari error tersebut agar kode tersebut bisa cepat diperbaiki agar bisa berjalan normal kembali.

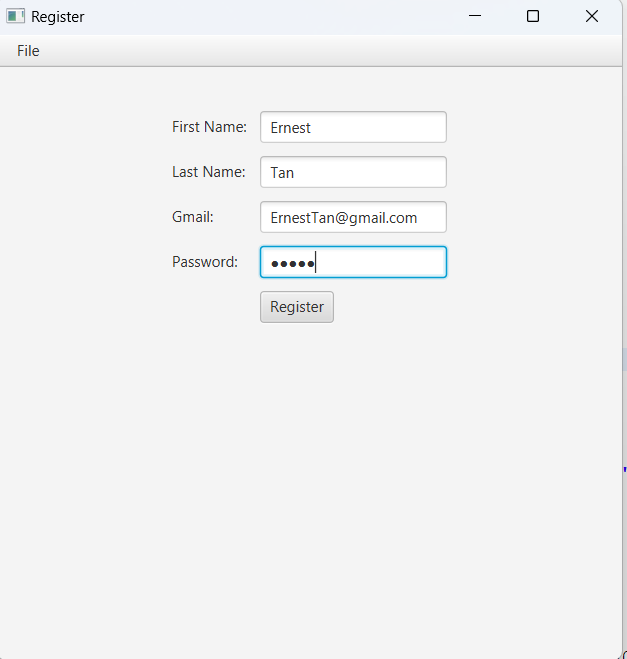
Manfaat dan *value* yang bisa didapatkan dari aplikasi *Filmlogue* adalah membantu para pengusaha bioskop sederhana dalam menjalankan usaha mereka agar penjualan tiket mereka menjadi lebih efektif dan bisa menjadi nilai tambah untuk pengusaha bioskop sederhana dalam mengimplementasi teknologi dalam usaha mereka. Pemrograman ini juga bisa membantu orang - orang untuk mendapatkan gambaran bahwa dengan adanya sebuah teknologi pemrograman maka setiap usaha menjadi lebih mudah untuk dijalankan.

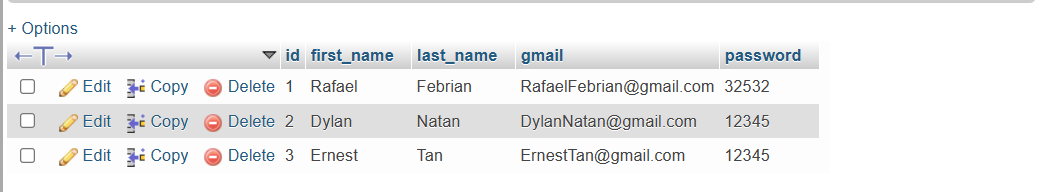
Pemrograman ini juga memiliki manfaat untuk orang - orang agar bisa menambah sebuah pengetahuan akan *javaFX* sehingga mereka bisa menjadi tertarik dan pemrograman dan mulai mempelajari materi tersebut agar bisa diimplementasikan di dunia nyata.

**3.6 Hasil Running *Code***

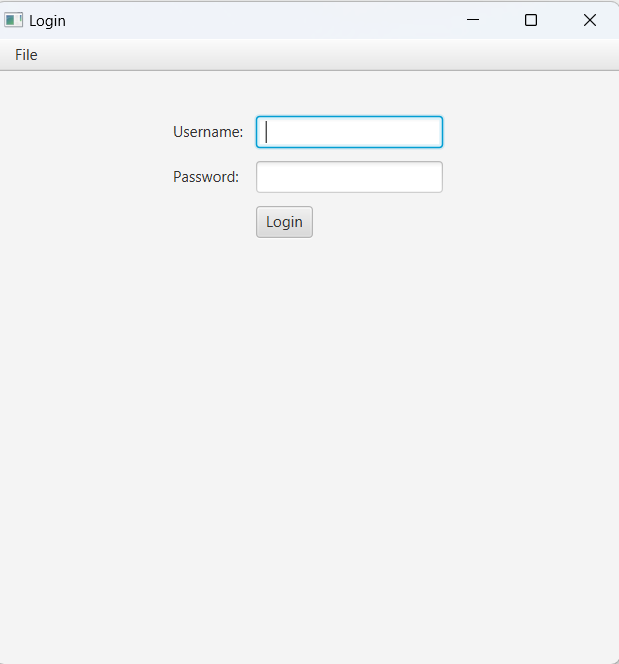


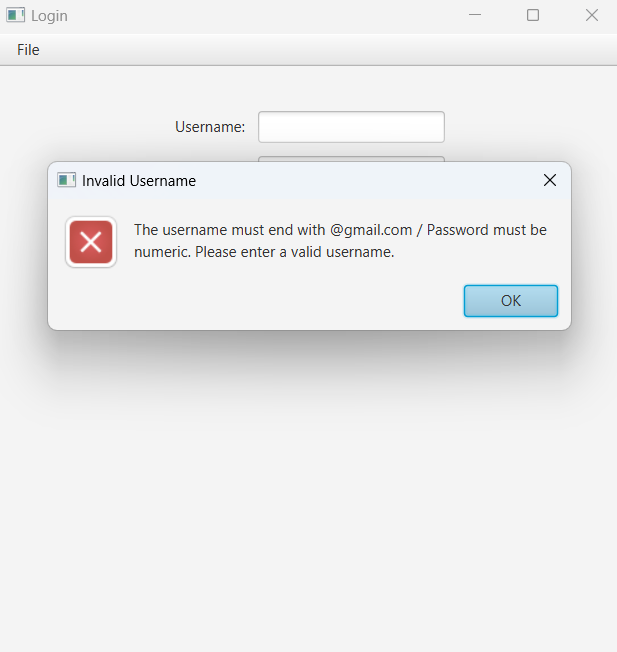
Tampilan pertama ketika kode dijalankan



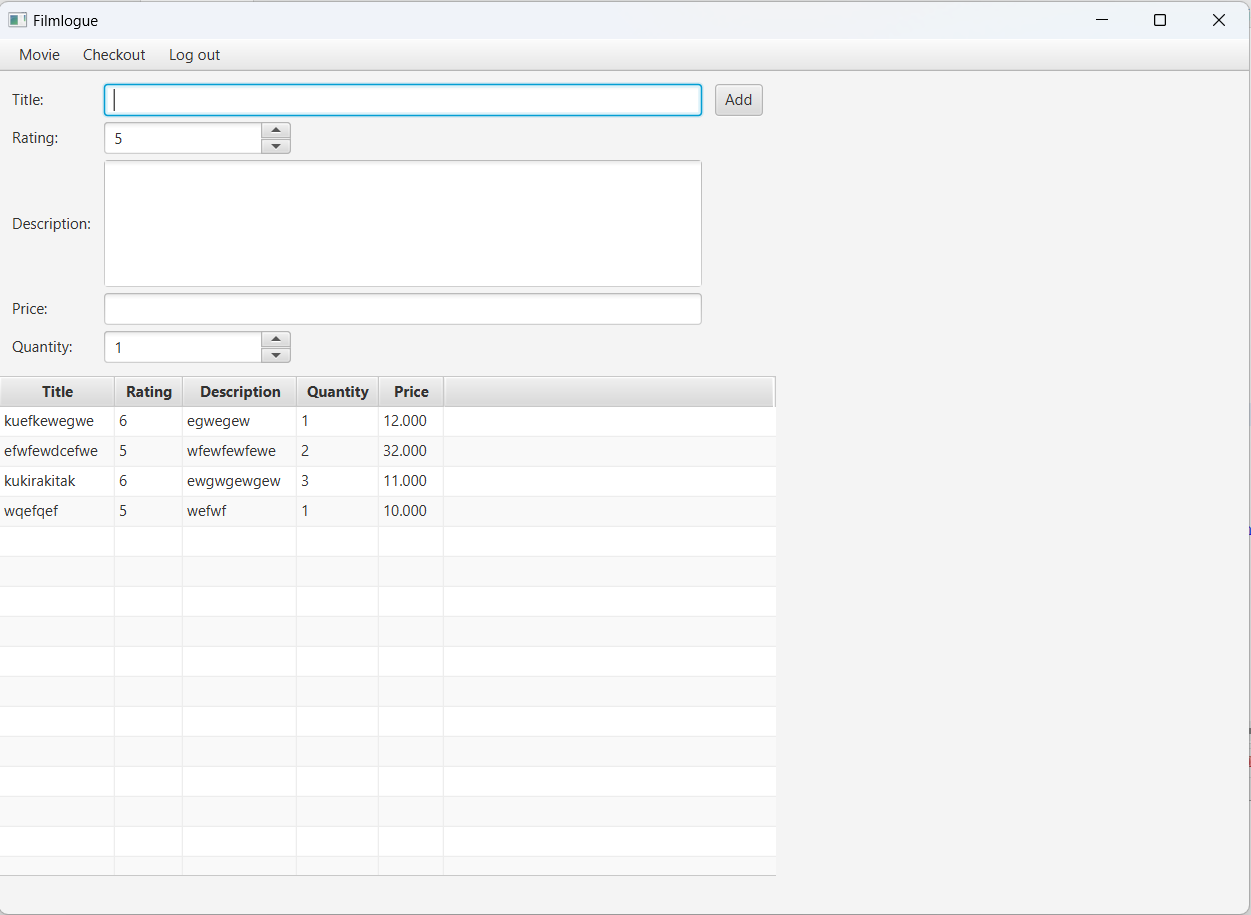


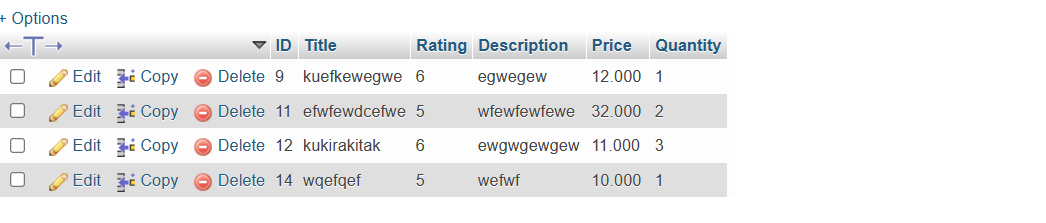
Tampilan kelas register dan tampilan *database* ketika kita input data



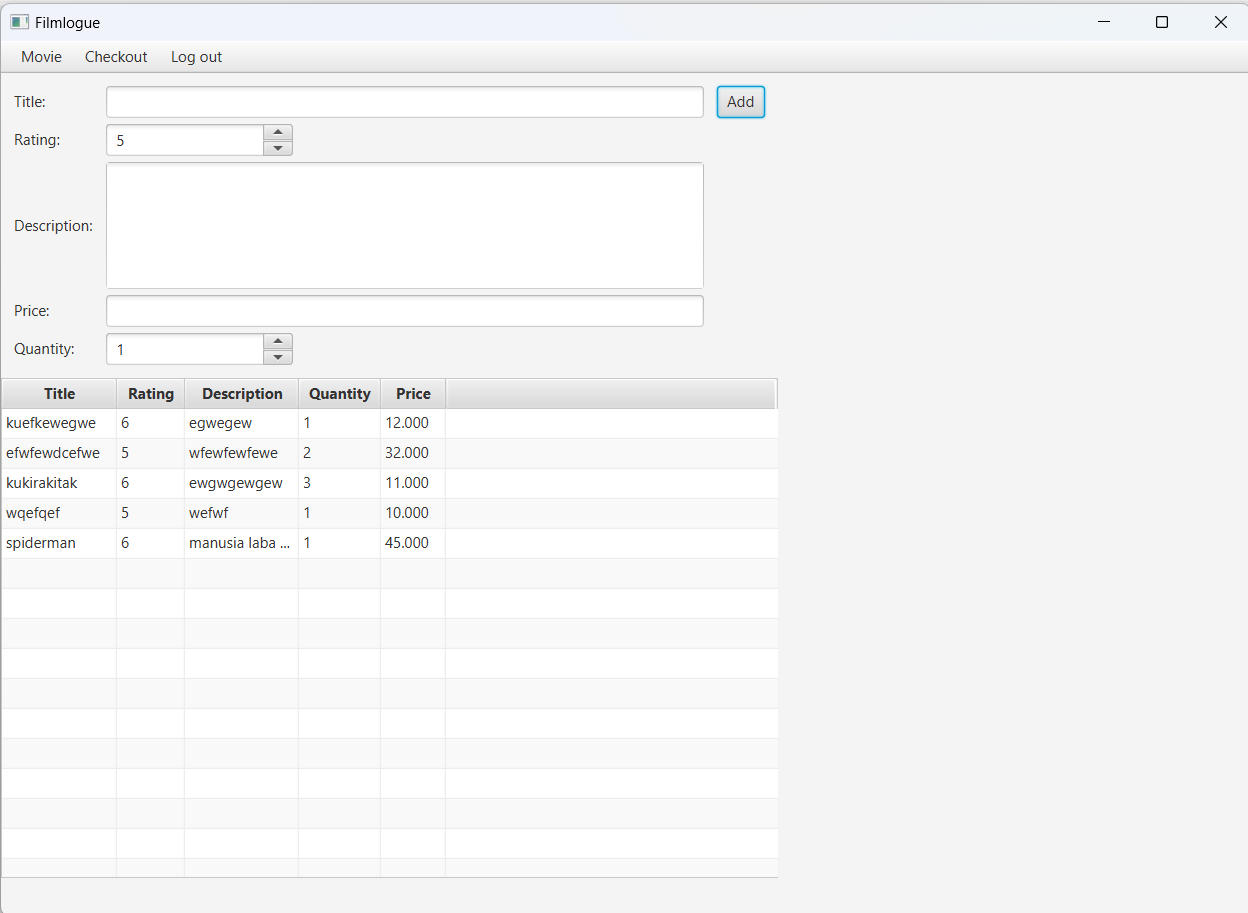


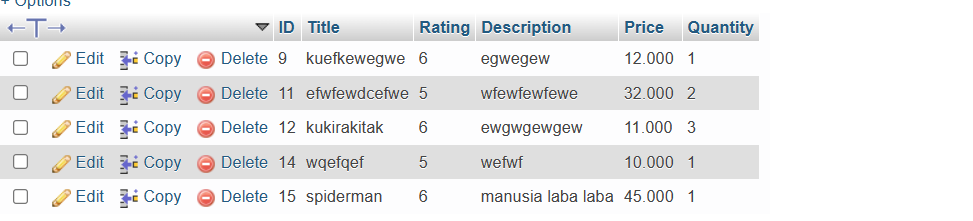
Tampilan kelas login dan ketika memasukkan data yang salah



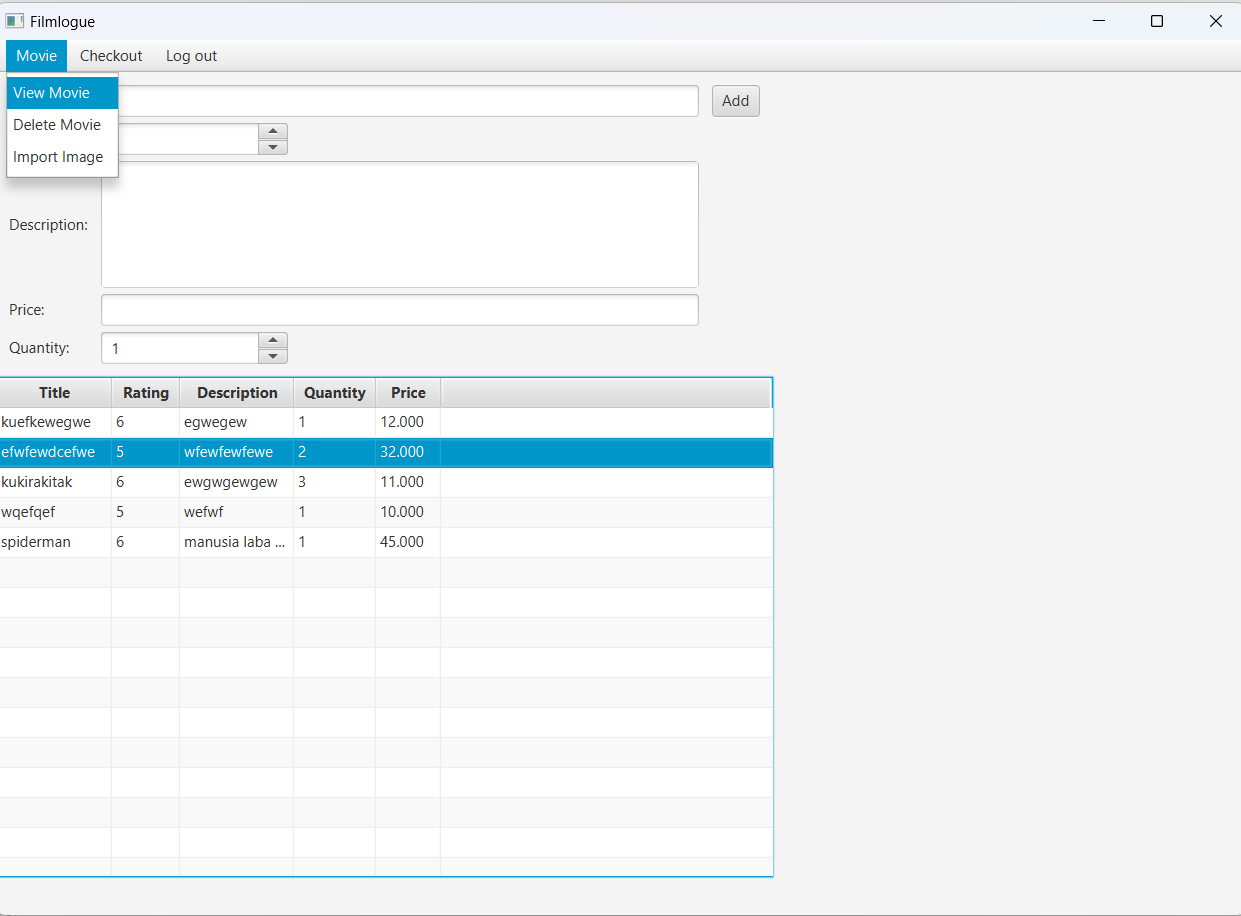


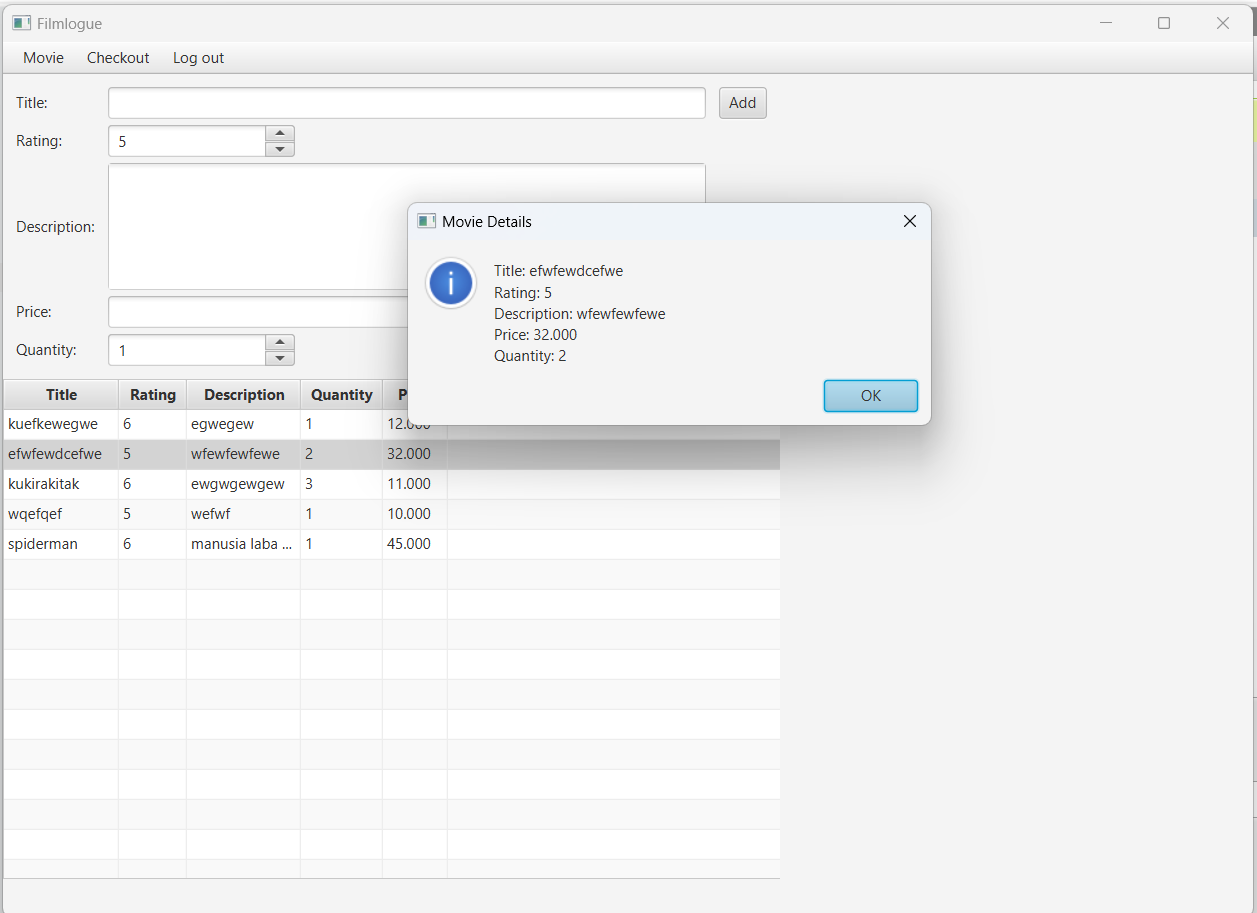
Tampilan kelas *main menu* atau *Filmlogue* dimana terdapat data yang sudah ada dalam *database*



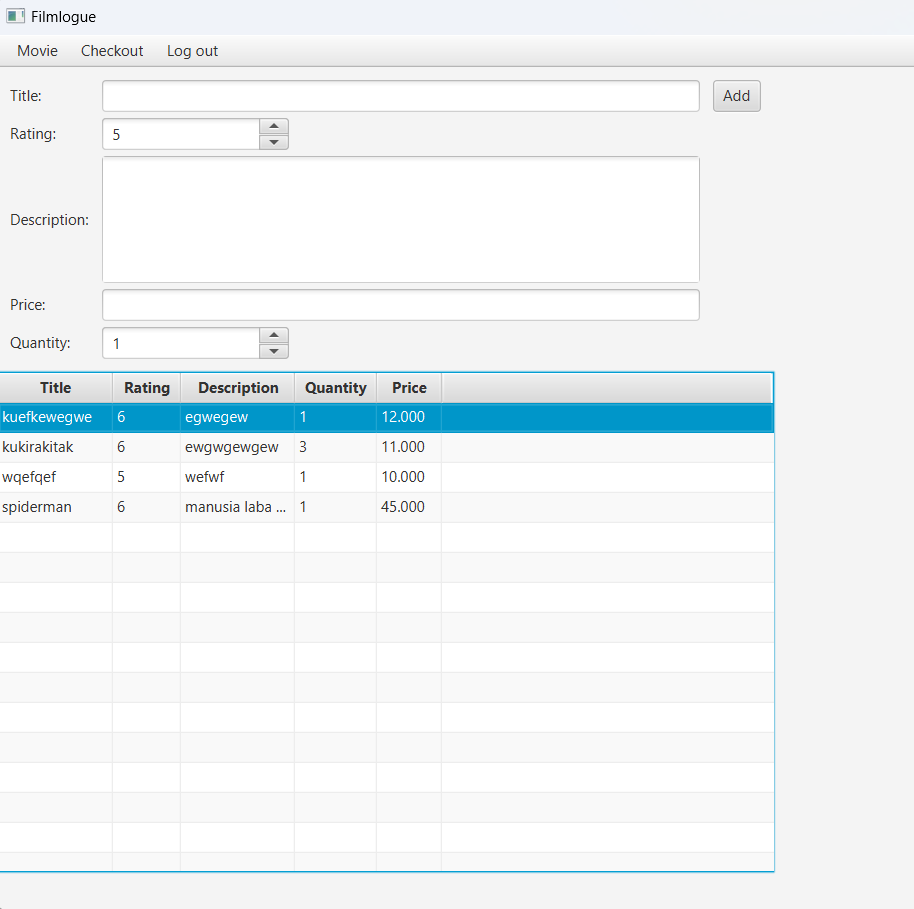


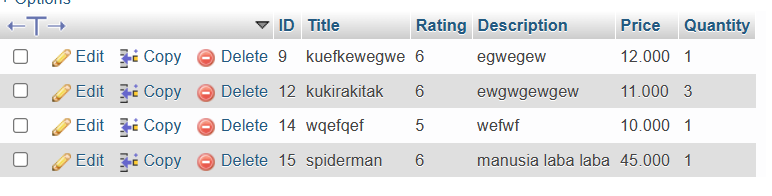
Berikut tampilan data yang baru diinput terlihat pada *tableview* yang nambah beserta *database* yang bertambah data



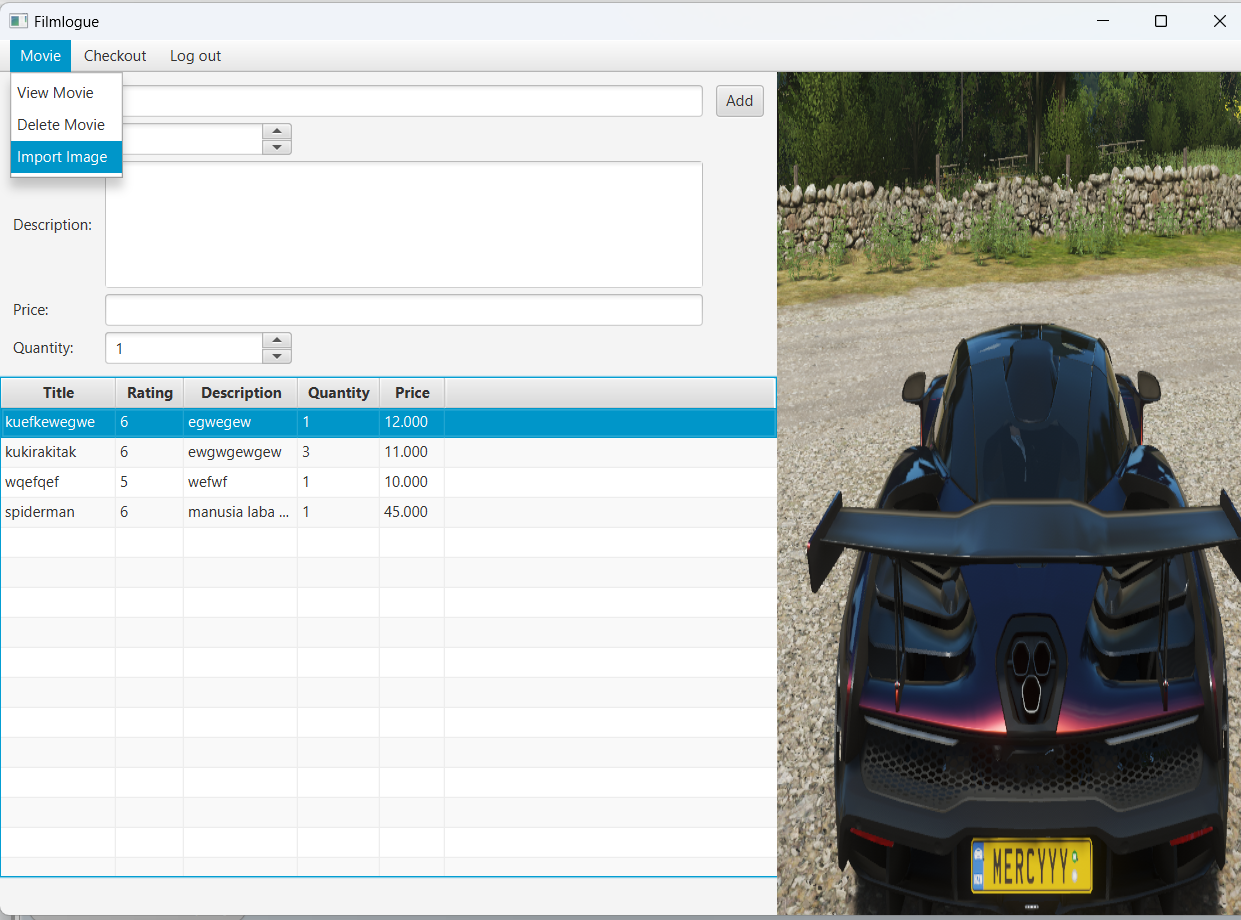


Tampilan ketika *view*

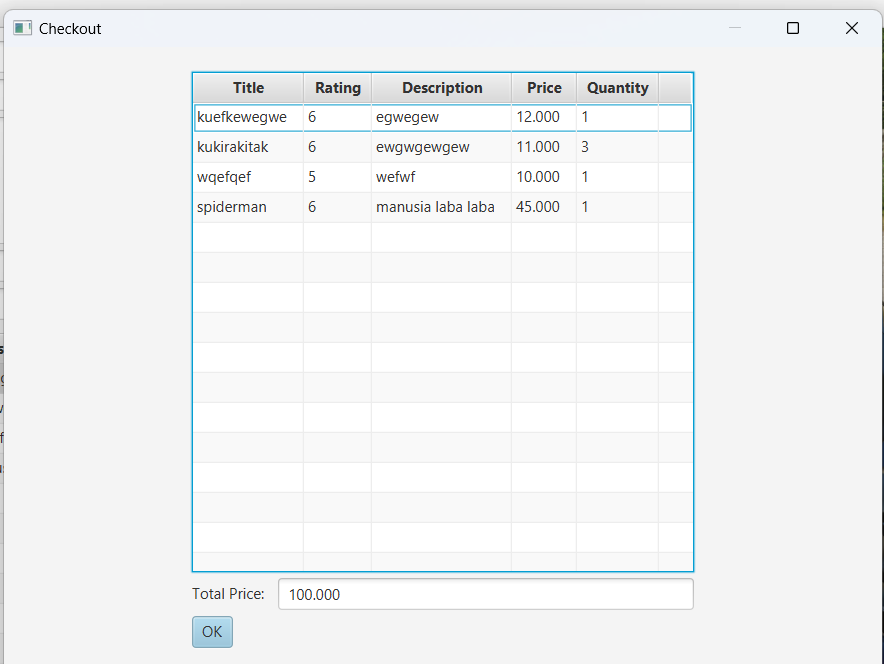




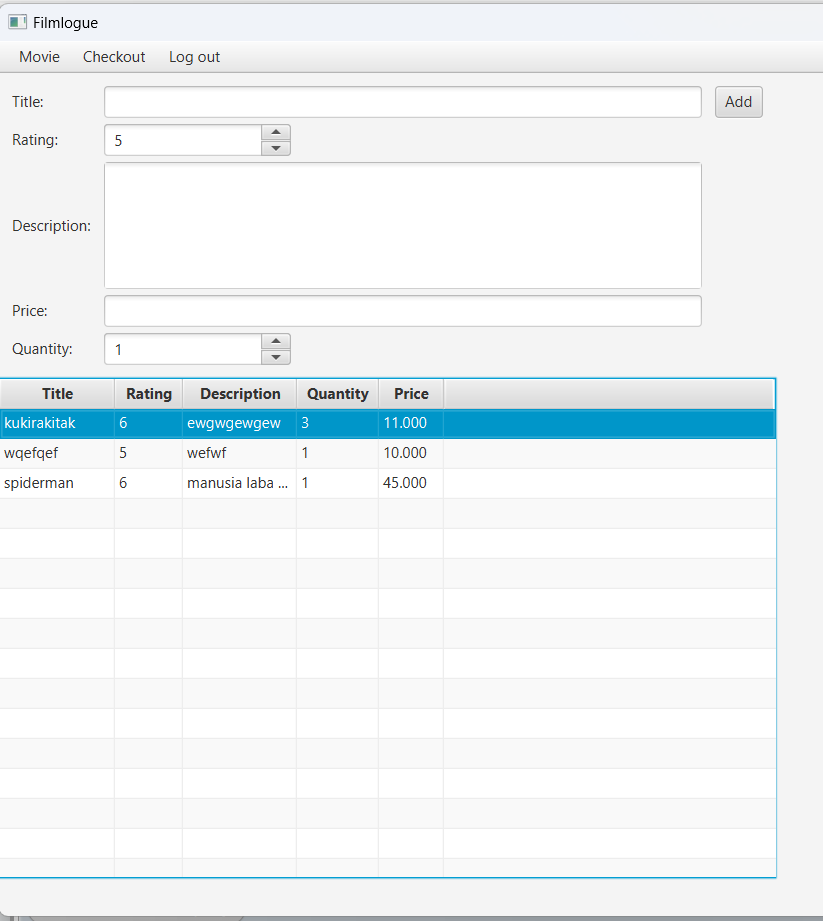
Tampilan ketika *delete* data sehingga data pada *tableview* berkurang dan data pada *database* ikut berkurang



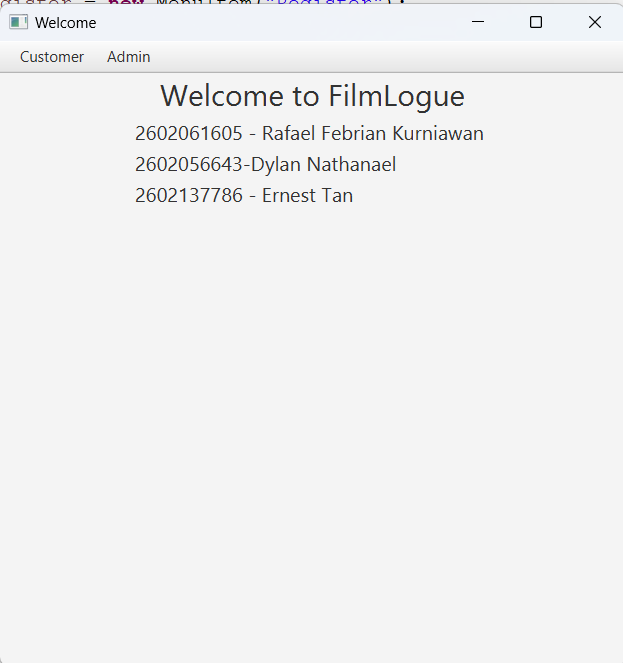
Tampilan ketika *import image*



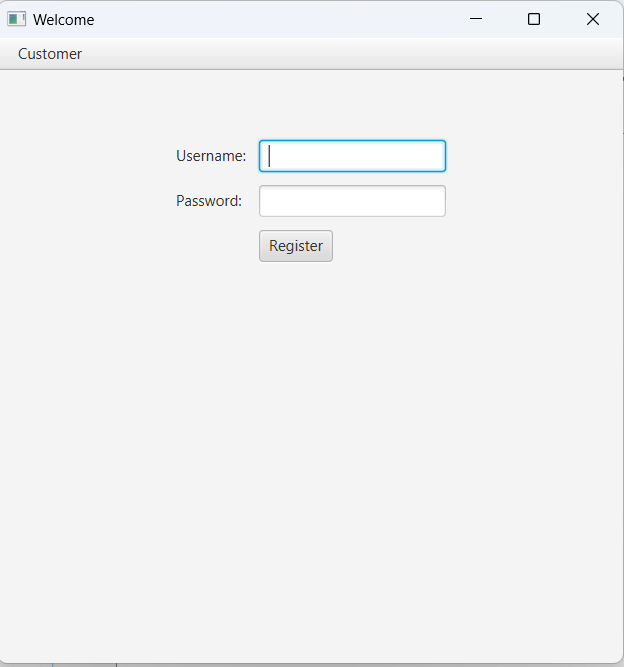
Tampilan ketika *checkout*

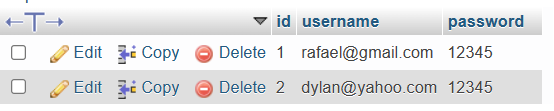


Tampilan ketika data di *checkout* maka berkurang dalam tableview tetapi tidak hilang di *database* agar data tersebut bisa dilihat sebagai *history*

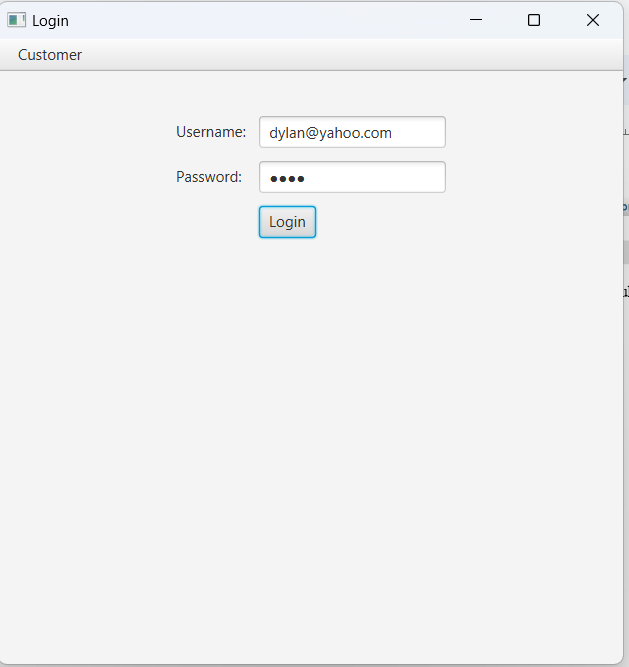
**

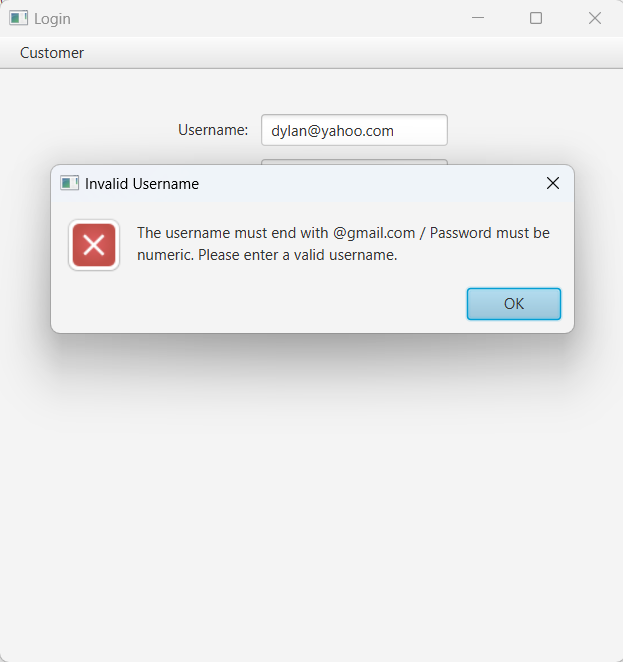
Tampilan awal aplikasi



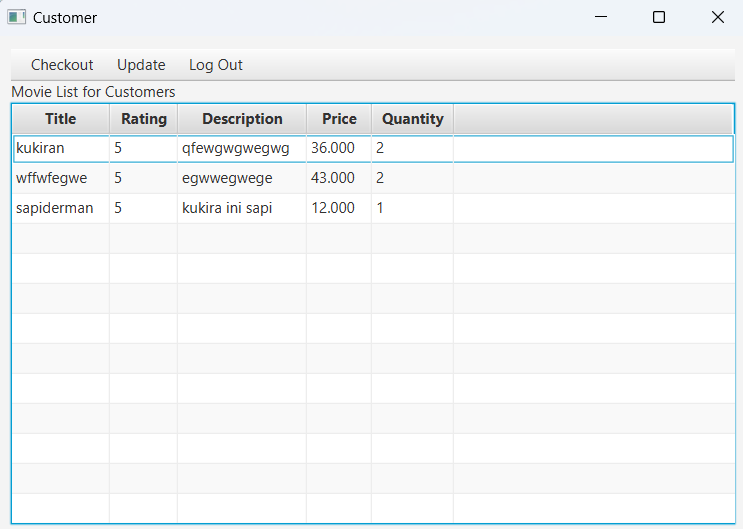


Tampilan halaman register dan data yang masuk untuk customer

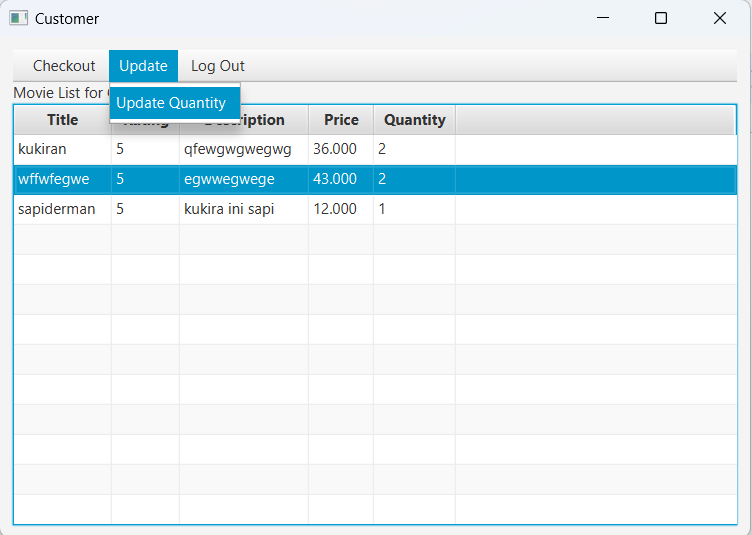


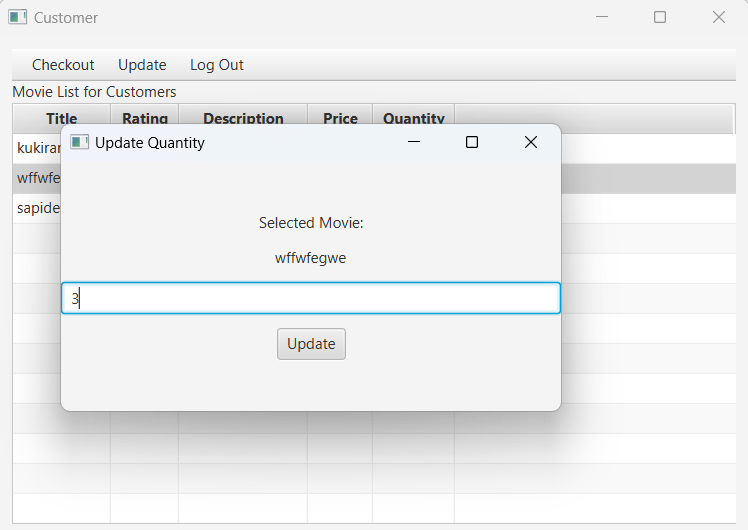


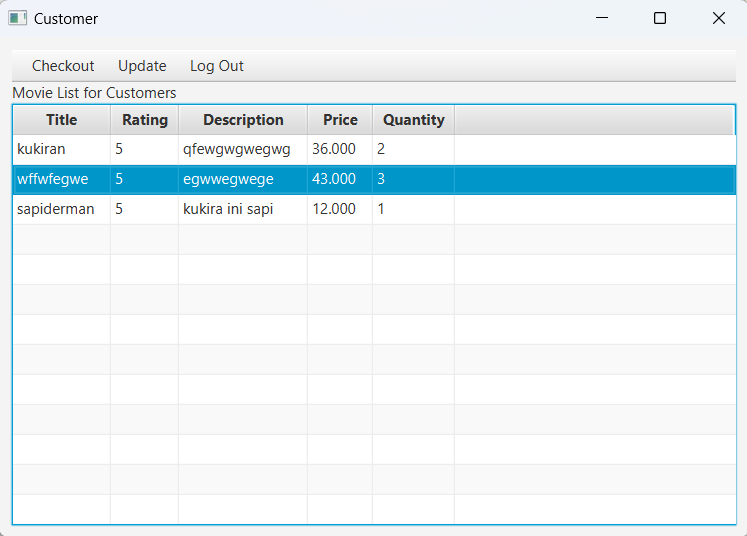
Tampilan *login* dan ketika salah input pada *login page* untuk customer

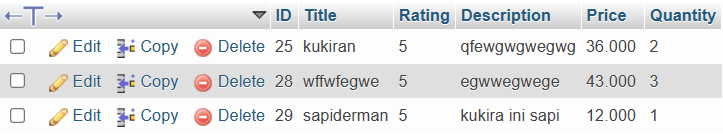


Tampilan main menu pada customer

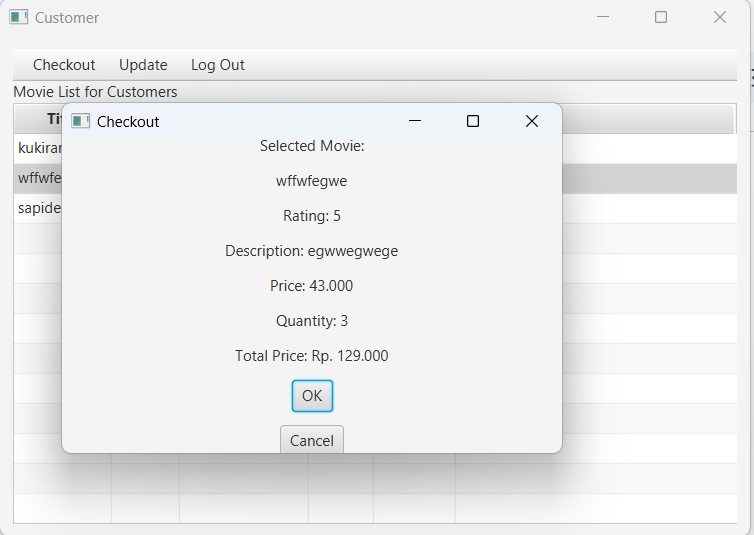








Tampilan untuk fitur *update* yang bisa *update quantity* tiket pada *tableview* menu customer begitu juga pada *database*





Tampilan menu *checkout* pada customer dan ketika tombol “ok” ditekan akan menghapus data pada *tableview* begitu juga pada *database*

# **Penutup**

Kesimpulan yang dibuatnya dari paper ini adalah Dengan pembuatan aplikasi Pemesanan Tiket ini dapat mempermudah para pengusaha atau startup bioskop sederhana yang ingin menjual Tiket Film berbasis aplikasi dengan desain yang simpel dan harga yang terjangkau. Tidak hanya para pengusaha bioskop sederhana tetapi para pengguna pun lebih mudah untuk mengakses nya karena aplikasi ini dapat digunakan oleh semua kalangan masyarakat.

Pembuatan Aplikasi tersebut menggunakan bahasa Pemrograman Java dengan Tampilan GUI *JavaFX.* Java merupakan bahasa pemrograman yang dapat diakses oleh seluruh device sehingga tidak ada ketentuan untuk mengaksesnya. Walau design nya simpel tetapi pada fungsi nya jelas sangat ter *highlight* karena menggunakan code code yang sudah disempurnakan oleh developernya.

**References**

1. Chrisna Putra, R. R., and D.Y. Sylfania. "Aplikasi Pemesanan Tiket Bioskop Berbasis Android Pada Bes cinema Pangkalpinang | Putra | It (INFORMATIC technique) journal." *PORTAL JURNAL Universitas Potensi Utama*, 2 Oct. 2018,
2. Mawarni, D., Andrianto, R., Liana, D., Azis, A., Rusadi, D., Rahmansyah, F., & Fikriyan, F. (2023, September 3). View of Aplikasi Pembelian Tiket Bioskop berbasis flutter: Integrasi Pilihan film, Bangku, Dan Pembayaran. https://jurnal.itbsemarang.ac.id/index.php/JPTIS/article/view/822/765
3. Emka Humam. Cara Pesan Tiket Bioskop Online Antiribet: Lewat Situs sampai Aplikasi. (2022, Juli 10).

<https://www.gramedia.com/best-seller/cara-pesan-tiket-bioskop-online/>

1. Ova Nurisma Putra, Jihan Fadhilah. Pembuatan Aplikasi Pemesanan Tiket Bioskop Berbasis Web. (Tahun 2022).

<https://journal.stmikjabar.ac.id/index.php/i/article/download/11/8>

1. AJENG SINDI E, RD RANI ROBI’AH, RIFFA NUR FATIAH, SRI ROHAYATI. SOFTWARE PENJUALAN TIKET BIOSKOP BERBASIS WEB . January 2020.

<https://www.researchgate.net/publication/338500529_Aplikasi_Tiket_Bioskop_Berbasis_WEB>

1. Bambang Sutrisno, Wahyu Sindu Prasetya. Rancang Bangun Mobile Apps E-tiket Bioskop Dengan Penerapan QR Code. (Agustus 2018). <https://sisfotenika.stmikpontianak.ac.id/index.php/enter/article/view/813>