# **PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**LAPORAN PROYEK AKHIR**

**MOVIE REVIEW**

****

**DISUSUN OLEH :**

|  |  |  |
| --- | --- | --- |
| **ANGGOTA** | **:** | **SATYA GHIFARI A 123180051**  **LIEK ALLYANDARU 123180054** |
| **KELAS/PLUG** | **:** | **E** |
| **NAMA ASISTEN** | **:** | **MONICA AMEILIANA ANDISTA** |
|  |  | **AMIRA SALSABILA** |

### PROGRAM STUDI INFORMATIKA

### JURUSAN TEKNIK INFORMATIKA

### FAKULTAS TEKNIK INDUSTRI

### UNIVERSITAS PEMBANGUNAN NASIONAL ”VETERAN”

#### YOGYAKARTA

#### 2020

# **HALAMAN PENGESAHAN**

**LAPORAN PROYEK AKHIR**

****

Disusun oleh :

*Satya Ghifari Adipratama* 123180051

*Liek Allyandaru* 123180054

Telah Diperiksa dan Disetujui oleh Asisten Praktikum Implementasi Basis Data

Pada Tanggal : …………………

|  |  |
| --- | --- |
| **Menyetujui,**  **Asisten Praktikum**  **Monica Ameiliana Andista M**  **NIM. 124180041** | **Asisten Praktikum**  **Amira Salsabila**  **NIM. 124170019** |

|  |
| --- |
| **Mengetahui,**  **Ka. Lab. Geoinformatka**  **Budi Santosa, S.Si.,M.T.**  **NIK. 2 7009 02 0234 1** |

# **KATA PENGANTAR**

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa yang senantiasa mencurahkan rahmat dan hidayah-Nya sehingga kami dapat menyelesaikan praktikum Pemrograman Berorientasi Objek serta laporan proyek akhir praktikum yang berjudul Movie Review. Adapun laporan ini berisi tentang proyek akhir yang kami pilih dari hasil pembelajaran selama praktikum berlangsung.

Tidak lupa ucapan terima kasih kepada asisten dosen yang selalu membimbing dan mengajari kami dalam melaksanakan praktikum dan dalam menyusun laporan ini. Laporan ini masih sangat jauh dari kesempurnaan, oleh karena itu kritik serta saran yang membangun kami harapkan untuk menyempurnakan laporan akhir ini.

Atas perhatian dari semua pihak yang membantu penulisan ini, kami ucapkan terima kasih. Semoga laporan ini dapat dipergunakan seperlunya.

Yogyakarta, Mei 2020

Penyusun

# **DAFTAR ISI**

[**HALAMAN PENGESAHAN** ii](#_Toc25323565)

[**KATA PENGANTAR** iii](#_Toc25323566)

[**DAFTAR ISI** iv](#_Toc25323567)

[**BAB I PENDAHULUAN** 1](#_Toc25323568)

[1.1 Latar Belakang 1](#_Toc25323569)

[1.2 Tujuan 1](#_Toc25323570)

[**BAB II PEMBAHASAN** 2](#_Toc25323571)

[2.1 Dasar Teori 2](#_Toc25323572)

[2.2 Deskripsi Umum Proyek Akhir 3](#_Toc25323573)

[2.3 Inti Pembahasan 4](#_Toc25323574)

[**BAB III JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS** 19](#_Toc25323575)

[3.1 Jadwal Pengerjaan 19](#_Toc25323576)

[3.2 Pembagian Tugas 19](#_Toc25323577)

[**BAB IV KESIMPULAN DAN SARAN** 20](#_Toc25323578)

[4.1 Kesimpulan 20](#_Toc25323579)

[4.2 Saran 20](#_Toc25323580)

[**DAFTAR PUSTAKA** 21](#_Toc25323581)

# **BAB I**

**PENDAHULUAN**

## **Latar Belakang**

Dunia perfilman mengalami kemajuan pesat selama satu dekade terakhir baik di Indonesia maupun internasional. Banyak faktor pendukungnya, salah satunya kemajuan teknologi mempermudah produksi dan distribusi film. Selain itu, adanya permintaan yang cukup tinggi dari masyarakat modern yang selalu butuh hiburan baru ditengah tempo kehidupan mereka yang sangat cepat. Oleh karena itu, selalu ada film baru yang rilis hampir setiap satu bulan sekali.

Berdasarkan fakta diatas, dunia perfilman tentunya penuh dengan persaingan. Adanya persaingan ini mempunyai dua sisi layaknya koin. Di satu sisi, positif karena adanya persaingan meramaikan dunia perfilman dan membuat produser film bisa melakukan evaluasi dari film lain. Namun, di sisi yang lain, ada negatifnya karena persaingan ini terkadang membuat produksi film harus dilakukan secepat mungkin agar waktu rilis bisa dipercepat. Hal ini terkadang bisa mempengaruhi kualitas dari film. Selain itu, terjadi monopoli studio besar yang sudah memiliki banyak penonton sehingga persaingan yang terjadi banyak terfokus pada studio besar. Padahal, banyak juga studio yang kecil sampai menengah dengan kualitas film yang boleh diadu dengan studio besar dari semua sisi baik dari tokoh, audio & visual, jalan cerita , dan lainnya.

Sisi negative tersebut pada akhirnya juga berdampak pada penonton. Banyak kasus saat orang selesai menonton film mereka merasa kecewa telah menghabiskan waktunya menonton film tersebut entah karena visual yang buruk, penokohan yang tidak tepat, atupun jalan cerita yang tidak menarik. Selain itu, ada juga kasus penonton tidak bisa menikmati film karena tidak sesuai dengan mereka. Misalnya, filmnya terlalu banyak adegan kekerasan.

Hal tersebut akhirnya mendorong adanya website atau aplikasi yang digunakan untuk melakukan ulasan (*review*) film. Pada aplikasi ini orang bisa menuturkan bagiaman film ini dan menjadi refrensi bagi orang lain. Tidak jarang yang melakukan ulasan adalah kritikus film ternama ataupun media daring yang memiliki reputasi untuk melakukan ulasan film. Kemudia aplikasi ini dapat diakses siapapun secara gratis yang membuat setiap orang yang terkoneksi internet bisa mengaksesnya.

Aplikasi seperti ini membantu para penonton untuk memilih film apa yang ingin mereka tonton. Apalagi di tengah pandemi seperti sekarang, orang butuh banyak hiburan yang bisa diakses secara daring karena hiburan yang bersifat harus keluar rumah hampir tidak ada atau bahkan dilarang. Karena itu penulis membuat aplikasi untuk melakukan ulasan film untuk membantu banyak orang memilih film hiburannya di tengah pandemi.

## **Tujuan**

Tujuan dari proyek Praktikum Pemrograman Berorientasi Objek ini adalah untuk membuat sistem *review* film untuk membantu penonton memilih film yang ingin ditonton serta untuk menyelesaikan tugas akhir dari mata kuliah Praktikum Pemrograman Berorientasi Objek.

# **BAB II**

**PEMBAHASAN**

## **2.1 Dasar Teori**

**2.1.1 Bahasa Pemrograman Java**

Java adalah bahasa pemrograman berorientasi objek. Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas-kelas untuk mengorganisasikan kode program dalam modul-modul logikal.

**2.1.2 Pemrograman Berorientasi Objek**

Dalam bahasa pemrograman berorientasi objek, program dibagi menjadi objek-objek yang saling berinteraksi satu sama lain. Objek tersebut dapat memiliki wujud yang nyata, maupun tidak nyata (abstrak). Konsep-konsep dasar dari pemrograman berorientasi objek adalah inheritance (pewarisan), encapsulation (pengkapsulan), dan polymorphism.

**2.1.3 *Overriding***

Overriding adalah kemampuan suatu subclass untuk menulis code milik method() yang abstract milik super class, atau memodifikasi isi attribute atau isi method() yang tidak abstract, tidak private, tidak final, dan non static milik super class. Method-method yg overloading wajib berada di dalam class yg sama atau dalam satu keluarga (sebagian method()-method() tersebut di dalam superclass dan yg lain di dalam subclass).

**2.1.4 *Overloading***

Overriding adalah kemampuan suatu subclass untuk menulis code milik method() yang abstract milik super class, atau memodifikasi isi attribute atau isi method() yang tidak abstract, tidak private, tidak final, dan non static milik super class.

Modifikasi modifier akses pada attribute, method(), dan constructor() miliki super class, boleh dilakukan dengan cara memperluas daerah aksesnya.

**2.1.5 *Encapsulation***

Encapsulation atau pengkapsulan digunakan untuk menyembunyikan atribut atau method pada suatu objek, dari objek lainnya, atau dengan kata lain yakni Information hiding. Pengkapsulan dilakukan dengan memberikan modifier private pada atribut maupun method. Kemudian, untuk memberikan nilai pada atribut private, digunakan method ‘setter’, dan untuk mengambil nilai dari atribut private, digunakan method ‘getter’. Method setter dan getter ini sama seperti method biasa, hanya berbeda pada pemanfaatan yang dituliskan programmernya saja.

**2.1.6 *Inheritance***

Inheritance adalah penurunan atribut dan method dari suatu kelas induk (superclass) ke kelas anak (subclass). Pewarisan bersifat menyeluruh dan beruntun, sehingga jika A merupakan superclass dari B, B merupakan superclass dari C, maka A juga merupakan superclass dari C. Jika ada atribut atau method dari C dan B yang ingin dimodifikasi, kita hanya perlu memodifikasi atribut/method pada A. Pernyataan extends digunakan untuk melakukan pewarisan ini.

**2.1.7 *Interface***

Interface merupakan suatu kelas yang berisi method-method tanpa implementasi, namun tanpa modifier abstract. Apabila terdapat atribut, maka atribut tersebut berlaku sebagai konstanta (static final). Interface bersifat seperti kontrak yang bisa digunakan oleh setiap kelas yang nyata, dan penggunaannya bukan diwariskan (extends) melainkan diimplementasikan (implements).

**2.1.8 *Exception***

Exception merupakan suatu objek Error khusus yang dibuat ketika terjadi kesalahan pada suatu program. Setelah suatu objek exception dibuat oleh java, objek tersebut dikirimkan ke program, proses ini disebut throwing an exception. Objek exception tersebut kemudian diterima oleh program, dan penanganan exception selanjutnya terserah pada program.

Penanganan exception dalam java dibuat dengan menuliskan kode program exception-handling pada Source code. Exception-handling berguna untuk mencegah program menjadi “Hang”. Code program exception handling secara umum terdiri dari 3 bagian yaitu try, catch dan finally.

**2.1.9 SWING**

GUI merupakan suatu metode untuk antar-muka komputer berbasis grafis. GUI digunakan dalam pembuatan program aplikasi dengan mempertimbangkan dua aspek yaitu keindahan tampilan dan kemudahan penggunaan program. Swing adalah kumpulan class–class yang digunakan untuk membuat GUI. Swing adalah GUI yang lightweight, karena lebih multiplatform dan sedikit menggunakan Native class.

**2.1.10 Java *Database Connectivity***

JDBC (Java *DataBase Connectivity*) merupakan teknologi Java yang populer sebagai standar pengaksesan database. Java tidak membedakan cara pemrograman database yang satu dengan lainnya. Sebagai contoh: cara Java mengolah tabel di Oracle dan MySQL sama saja, yang berbeda hanyalah alamat URL koneksi dan driver JDBC-nya. Sehingga kita dapat membuat sebuah program yang sama di Java untuk mengolah database yang ada di MySQL, Oracle, SQL Server dan lain-lain asalkan nama dan struktur tabelnya sama.

**2.1.3 *Model View Controller***

Model View Control pertama kali diperkenalkan oleh Trygve Reenskaug, seorang pengembang perangkat lunak Smalltalk pada Xerox Palo Alto Research Center pada tahun 1979. Konsep ini memisahkan antara akses data, logika bisnis, dan antarmuka pada perangkat lunak, secara detail MVC dibagi 3, yaitu :

1. Model, merupakan bagian dari MVC yang bertugas menghubungkan langsung antara tugas-tugas dengan proses di Database (Create, Read, Update, Delete). Sebagai contoh pada Java, dibuat kelas sendiri antara koneksi dengan query akses data ke database. Jadi untuk menggunakan fungsi Insert, Update, Delete, dan Select kita tinggal memanggil method-method yang menangani hal tersebut.
2. View, merupakan bagian dari MVC yang bertugas menghubungkan antara pengguna dengan tugas perangkat lunak, dengan kata lain View menangani bagian Interface atau Antarmuka untuk mempermudah si pengguna menggunakan perangkat lunak.
3. Controller, menjadi tulang punggung perangkat lunak. Controller menghubungkan antara View dan Model yang meng-handle semua proses-proses yang terjadi didalamnya. Sebagai contoh kasus User Login, tugas Controller adalah mencocokkan data yang di-inputkan user dengan data yang ada di database

## **2.2 Deskripsi Umum Proyek Akhir**

Program ini merupakan program yang digunakan untuk membuat ulasan sebuah film agar penonton dapat dengan mudah menentukan pilihannya.

Program ini dimulai dari menu *login* untuk pengguna yang ingin mengulas sebuah film. Pada halaman *login*, pengguna dapat mengisi *username* dan *password* untuk masuk ke halaman menu. Jika belum memiliki akun, pengguna dapat melakukan registrasi terlebih dahulu dengan menekan tombol registrasi pada halaman *login*.

Halaman menu menampilkan empat pilihan yaitu input, *review* anda, *profile*, dan daftar film. Menu input digunakan untuk menginputkan sebuah film yang akan diulas, jika film belum ada, maka film otomatis akan ditambahkan dalam *database*.

Menu *review* anda digunakan untuk menampilkan *review* atau ulasan yang sudah pernah user inputkan. Tampilan data berupa table yang dapat diklik untuk meng-*update* datau menghapus data ulasan.

Menu *profile* digunakan untuk menampilkan identitas pengguna dan menampilkan tingatan ulasan yang dilihat dari jumlah ulasan yang dibuat. Selain itu terdapat tombol ubah password untuk mengubah password pengguna.

Menu Daftar film digunakan untuk menampilkan semua film yang telah diulas oleh semua pengguna. Data ditampilkan dalam bentuk table yang dapat diklik untuk melihat detail dari film tersebut.

## **2.3 Inti Pembahasan**

**2.3.1 *Listing* Program**

#include <iostream>

#include <windows.h>

#include <malloc.h>

using namespace std;

typedef struct Queuenode\_1 \*queueptr\_1;

typedef struct Queuenode\_2 \*queueptr\_2;

typedef struct Historynode\_1 \*historyptr\_1;

typedef struct Historynode\_2 \*historyptr\_2;

queueptr\_1 qstart\_1, qend\_1;

queueptr\_2 qstart\_2, qend\_2;

historyptr\_1 hstart\_1, hend\_1;

historyptr\_2 hstart\_2, hend\_2;

struct Customer

{

char name[30], address[50], NIK[20];

};

struct Queuenode\_1{

Customer customer;

queueptr\_1 next;

};

struct Queuenode\_2{

Customer customer;

queueptr\_2 next;

};

struct Historynode\_1{

***Listing* Program 2.1** Program Antrian Bank

**2.3.2 Penjelasan Program**

1. **Deklarasi**

Pada program ini, penyusun mendeklarasikan empat pointer, satu *struct* Customer, dua *struct* *pointer* untuk antrian, dua *struct pointer*. Empat pointer diantaranya terdiri dari dua *pointer* untuk antrian dan dua *pointer* untuk penyimpanan *history*. *Struct* Customer berisikan *field* yang digunakan untuk menyimpan data nama, alamat, dan nik bertipe data char. Selain itu, penyusun mendeklarasikan berbagai macam void yang digunakan untuk membuat *linked list*, pemeriksaan *linked list* kosong, *log in*, pemasukan data, pengeluaran data, dan pencarian data.

1. ***Struct* Queuenode**

Terdapat dua *struct* Queuenode, yaitu Queuenode\_1 dan Queuenode\_2. Queuenode\_1 digunakan untuk menyimpan data *customer* yang ingin melakukan deposit. Queuenode\_2 digunakan untuk menyimpan data *customer* yang memlilih menu *customer service*.

**2.3.3 Output Program**

# **BAB III**

**JADWAL PENGERJAAN DAN PEMBAGIAN TUGAS**

## **3.1 Jadwal Pengerjaan**

Jadwal pengerjaan proyek Praktikum Pemrograman Berorientasi Objek *Movie Review* ini sebagai berikut:

**Tabel 3.1** Tabel Jadwal Pengerjaan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| no | Nama Tugas | Minggu ke- | | | |
| April | | Mei | |
| 3 | 4 | 1 | 2 |
| 1 | Menentukan Judul Proyek |  |  |  |  |
| 2 | Perancangan Konsep Proyek |  |  |  |  |
| 3 | Desain Skenario Program |  |  |  |  |
| 4 | Halaman Login |  |  |  |  |
| 5 | Register Akun |  |  |  |  |
| 6 | Halaman Input Data |  |  |  |  |
| 7 | Halaman Review Anda |  |  |  |  |
| 8 | Halaman Profile |  |  |  |  |
| 9 | Halaman Daftar Film |  |  |  |  |
| 10 | Pembuatan Laporan |  |  |  |  |

## **3.2 Pembagian Tugas**

Pembagian tugas proyek Praktikum Pemrograman Berorientasi Objek *Movie Review* ini sebagai berikut:

**Tabel 3.2** Tabel Pembagian Tugas

|  |  |  |
| --- | --- | --- |
| no | aktivitas | penanggung jawab |
| 1 | Pembahasan Judul | Satya & Allyandaru |
| 2 | Perancangan Konsep | Satya & Allyandaru |
| 3 | Desain Skenario | Satya & Allyandaru |
| 4 | Halaman Login | Satya & Allyandaru |
| 5 | Halaman Register | Allyandaru |
| 6 | Halaman Input Data | Allyandaru |
| 7 | Halaman Review Anda | Allyandaru |
| 8 | Halaman Profile | Satya |
| 9 | Halaman Daftar Film | Satya |
| 10 | Pembuatan Laporan | Satya & Allyandaru |

# **BAB IV**

**KESIMPULAN DAN SARAN**

## **4.1 Kesimpulan**

Kesimpulan yang dapat diambil dari pengerjaan ini adalah proyek *Movie Review* yang telah dibuat dapat melakukan penginputan ulasan film sehingga pengguna dapat membaca ulasan yang ada sebelum menonton sebuah film untuk menentukan pilihan film yang akan ditonton pengguna. Selain itu program dapat digunakan untuk meng-*update* data ulasan jika sewaktu-waktu memerlukan perubahan, menampilak *profile* pengguna dan menampilkan semua data ulasan dari semua pengguna untuk mempermudah pengguna sebelum menonton film.

## **4.2 Saran**

Adapun saran yang dapat dipertimbangkan untuk penyempurnaan dan pengembangan program *Movie Review* lebih lanjut adalah sebagai berikut:

1. Sistem *Movie Review* diharapkan dapat bekerja dengan lebih optimal dengan menyediakan fitur-fitur maupun menu-menu yang lebih banyak serta lebih kompleks, sehingga pengguna dapat lebih nyaman dalam melakukan aktivitas.
2. Tampilan pada yang diharapkan lebih baik dan kedepannya bisa diperbarui sesuai dengan teori UI UX.
3. Sejarah antrian dapat dibuat lebih kompleks dengan menambahkan data tanggal dan waktu ketika *user* selesai melakukan antrian, agar admin dapat lebih mudah melakukan pendataan terhadap *user*.

# **DAFTAR PUSTAKA**

Anonim.2019.*Modul Praktikum Pemrograman Berorientasi Objek Program Studi Teknik Infromatika Jurusan Teknik Informatika Fakultas Tekik Industri UPN “Veteran” Yogyakarta*. Yogyakarta: UPN “Veteran” Yogyakarta.