# **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 Struktur Data Antrian Bank ini adalah untuk membuat sistem antrian bank yang memudahkan nasabah menggunakan nomor antrian serta menyelesaikan tugas akhir dari mata kuliah Struktur Data.

Tujuan dari proyek Struktur Data Antrian Bank ini adalah untuk membuat sistem antrian bank yang memudahkan nasabah menggunakan nomor antrian serta menyelesaikan tugas akhir dari mata kuliah Struktur Data.

# **BAB II**

**PEMBAHASAN**

## **2.1 Dasar Teori**

**2.1.1 Bahasa Pemrograman Java**

C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan *class-class* yang merupakan anak class yang dibuat sebelumnya sebagai abstraksi dari *object-object* fisik, *Class* tersebut berisi keadaan *object*, anggota-anggotanya dan kemampuan dari *object*-nya, Setelah beberapa *Class* dibuat kemudian masalah dipecahkan dengan *Class*. Bahasa C adalah bahasa pemrograman prosedural yang memungkinkan kita untuk membuat prosedur dalam menyelesaikan suatu masalah. Bahasa pemrograman C++ adalah bahasa pemrograman yang berorientasi pada objek.

Perbedaan Antara Bahasa pemrograman C dan C++ meskipun bahasa-bahasa tersebut menggunakan sintaks yang sama tetapi mereka memiliki perbedaan, C merupakan bahasa pemrograman prosedural, dimana penyelesaian suatu masalah dilakukan dengan membagi-bagi masalah tersebut kedalam sub-submasalah yang lebih kecil, sedangkan untuk C++ merupakan bahasa pemrograman yang memiliki sifat Pemrograman berorientasi objek, Untuk menyelesaikan masalah, C++ melakukan langkah pertama dengan menjelaskan *class*-*class* yang merupakan anak *class* yang dibuat sebelumnya sebagai abstraksi dari *object*-*object* fisik, Class tersebut berisi keadaan *object*, anggota-anggotanya dan kemampuan dari *object*-nya, Setelah beberapa Class dibuat kemudian masalah dipecahkan dengan *Class*.

**2.1.2 Pemrograman Berorientasi Objek**

*Pointer* (variabel penunjuk) adalah suatu variabel yang berisi alamat memori dari suatu variabel lain. Alamat ini merupakan lokasi dari objek lain (biasanya variabel lain) di dalam memori. Contoh, jika sebuah variabel berisi alamat dari variabel lain, variabel pertama dikatakan menunjuk ke variabel kedua.

tipe\_data \*nama\_pointer;

**Contoh 2.1** *Pointer*

Tipe data *pointer* mendefinisikan tipe dari objek yang ditunjuk oleh *pointer*. Secara teknis, tipe apapun dari *pointer* dapat menunjukkan lokasi (dimanapun) dalam memori. Bahkan operasi *pointer* dapat dilaksanakan relatif terhadap tipe dasar apapun yang ditunjuk. Contoh, ketika kita mendeklarasikan pointer dengan tipe int\*, *compiler* akan menganggap alamat yang ditunjuk menyimpan nilai *integer* - walaupun sebenarnya bukan (sebuah *pointer* int\* selalu menganggap bahwa ia menunjuk ke sebuah objek bertipe *integer*, tidak peduli isi sebenarnya).

**2.1.3 *Model View Controller***

*Linked* *list* adalah salah satu cara untuk menyimpan sekumpulan elemen. Sama halnya dengan *array*, elemen yang disimpan dapat berupa karakter atau *integer*. Masing-masing elemen dalam *linked* *list* disimpan dalam bentuk sebuah *node*. *Node* adalah gabungan dua sub-elemen atau komponen. Bagian data yang menyimpan elemen dan bagian *next* yang menyimpan *link* ke *node* selanjutnya.

*Linked* *list* terbentuk saat terdapat banyak *node* yang saling tertaut dan membentuk sebuah rantai. Setiap *node* menunjuk pada *node* selanjutnya sesuai urutan. *Node* pertama yang selalu digunakan sebagai referensi saat melakukan *traversing* pada *list* (senarai; daftar) disebut *HEAD*. *Node* terakhir akan menunjuk ke *NULL*.

*Linked* *list* memiliki banyak jenis seperti *Linked* *list* *Array* dan *Linked* *List* Kepala Berekor. Selain itu, *Linked* *list* dapat diterapkan dalam *Stack*, *Queue*, dan *Graph* dan Pohon Telusur Biner.

## **2.2 Deskripsi Umum Proyek Akhir**

Program ini merupakan program yang digunakan untuk mengambil nomor antrian nasabah bank serta untuk memanggil antrian nasabah kemudian yang selanjutnya dimasukkan ke sejarah antrian.

Program ini dimulai dari menu *log in*. User dapat *log in* sebagai admin yang memerlukan *username* dan *password* atau *log in* sebagai nasabah untuk mengambil nomor antrian. Pada menu *user*, antrian dibagi menjadi dua yaitu menu deposit ditujukan untuk *user* yang ingin mengantri untuk melakukan setoran uang dan menu *customer* *service* yang ditujukan untuk *user* yang ingin mengantri untuk mendapatkan *customer* *service*, *user* akan memasuki antrian dengan mengisi beberapa data seperti nama, NIK, dan alamat.

Pada sisi admin, terdapat menu untuk memanggil nasabah yang mengantri untuk melakukan deposit dan *customer* *service*. Di dalam tiap menu tersebut, terdapat menu *call* *queue* untuk memanggil antrian, *show* *queue* untuk melihat antrian yang masih ada, serta *queue* *history* untuk melihat sejarah antrian yang telah dipanggil yang diurutkan berdasarkan NIK nasabah. Pengurutan dapat dilakukan secara *ascending* maupun *descending*. Selain itu terdapat menu *search* untuk mencari data nasabah yang kemudian dapat melakukan *update* data nik dan alamat.

## **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 *project* antrian bank yang dibuat dapat melakukan pengambilan nomor antrian sehingga nasabah tidak perlu lagi untuk berdiri mengantri. Dengan menggunakan *linked* *list* dan *queue* dalam pembuatan tugas ini, pengguna (nasabah dan admin) dapat dengan mudah melakukan aktivitas perbankan. *User* dapat dengan mudah mengantri serta admin dapat dengan mudah memanggil nasabah, melihat antrian yang masih ada, menampilkan sejarah antrian yang ada, serta mengubah data yang ada. Dengan begitu, sistem antrian bank akan lebih terstruktur dan lebih efisien dari pada sebelumnya.

## **4.2 Saran**

Adapun saran yang dapat dipertimbangkan untuk penyempurnaan dan pengembangan program Antrian Bank lebih lanjut adalah sebagai berikut:

1. Sistem Antrian Bank diharapkan dapat bekerja dengan lebih optimal dengan menyediakan fitur-fitur maupun menu-menu yang lebih banyak serta lebih kompleks, sehingga *user* dan admin dapat lebih nyaman dalam melakukan aktivitas.
2. Penambahan fitur untuk membatalkan antrian user ketika user mempunyai suatu kepentingan khusus yang membuat user harus meninggalkan antrian.
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.