

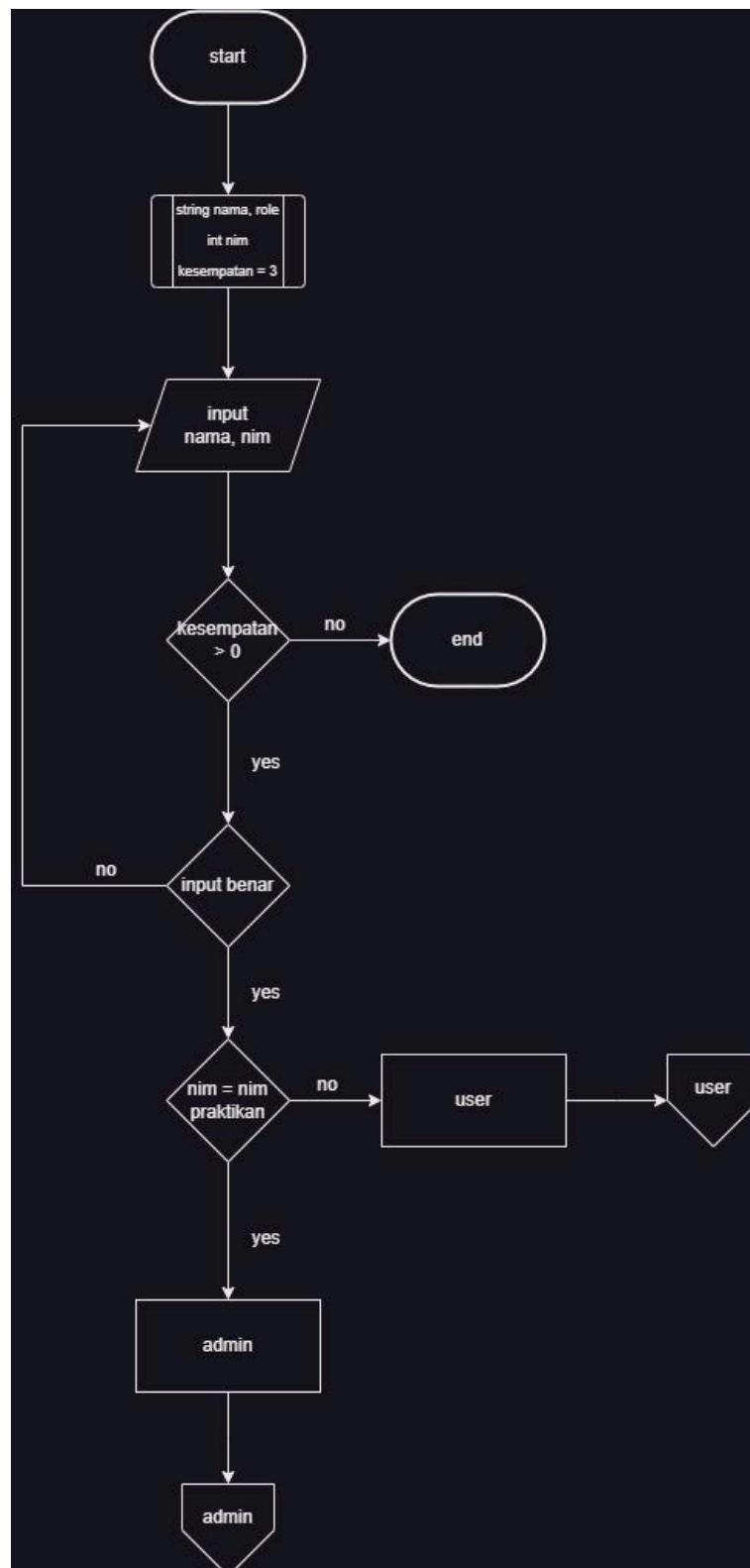
**LAPORAN PRAKTIKUM**  
**POSTTEST 3**  
**ALGORITMA PEMROGRAMAN LANJUT**



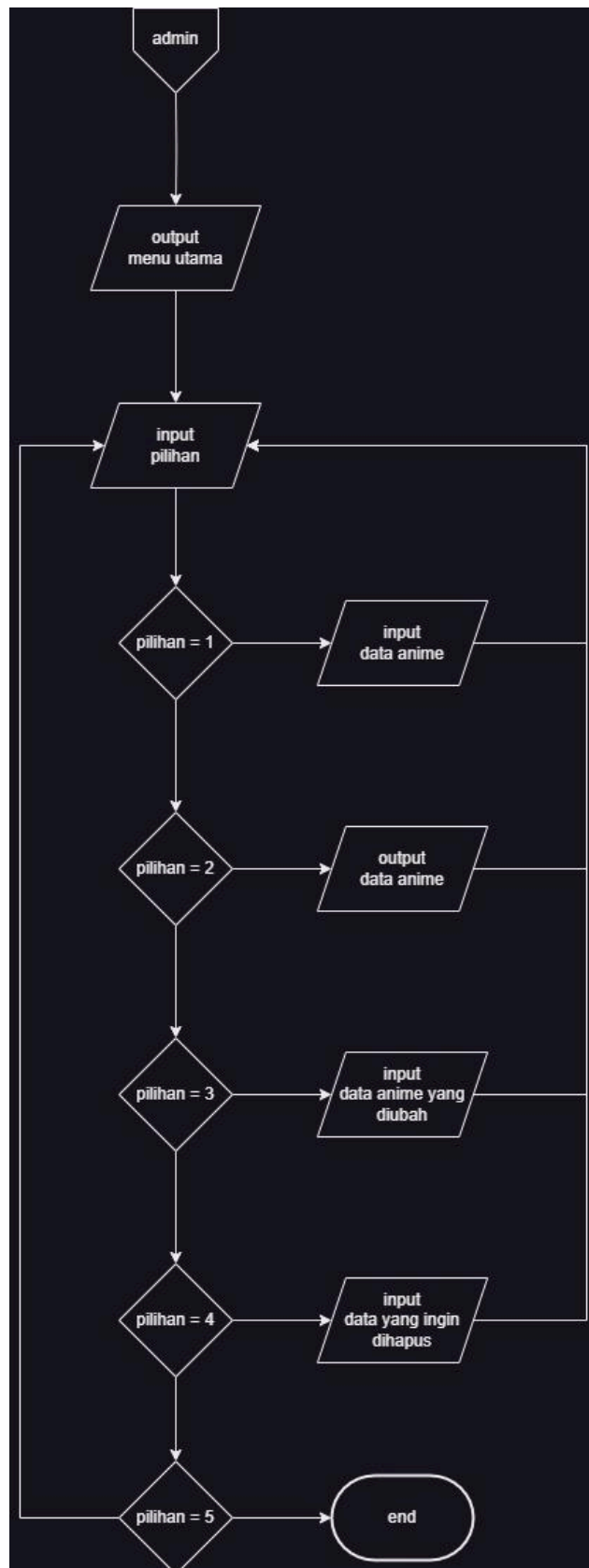
**Disusun oleh:**  
**Richo Anan Rizky Putra (2409106062)**  
**Kelas (B1'24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

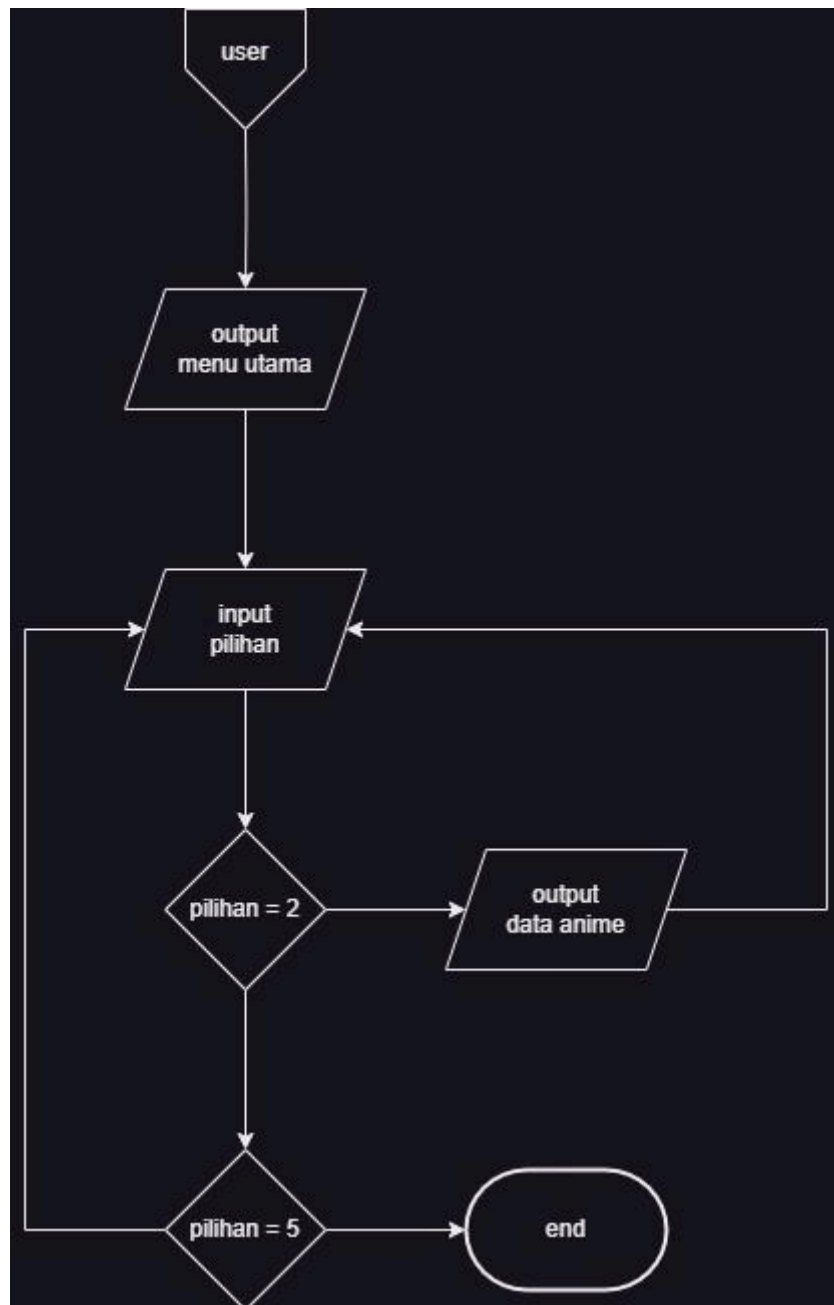
## 1. Flowchart



Gambar 1.1



Gambar 1.2



Gambar 1.3

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program yang saya buat ini adalah Program Pengelolaan Anime. Program ini akan terlihat seperti list/daftar judul Anime lengkap dengan genre, rating, jumlah episode dan jumlah seasonnya. Program ini dapat digunakan untuk Menambahkan data anime ke list/daftar, Membaca list/daftar anime tersebut, Mengubah data anime yang sudah ada di daftar, dan Menghapus data anime di daftar. Dan Program ini takkan berhenti/selesai sebelum pengguna memilih pilihan “Keluar” di menu utama

### 3. Source Code

#### A. Fitur Memasukkan NIM

Fitur ini digunakan untuk memasukkan NIM agar *user* dan *admin* dapat menggunakan fitur-fitur Sistem daftar Anime, *user* dan *admin* hanya diberi 3x kesempatan, jika melakukan kesalahan sebanyak 3x, maka *user* dan *admin* akan langsung keluar dari program. Jika memasukkan nim sesuai dengan *adminNIM* akan mendapat role *admin*, sedangkan jika memasukkan dengan *userNIM* akan mendapat role *user*

```
const string adminNIM = "2409106062";
const string userNIM = "2409106063";

int main() {
    string nama, nim, role;
    int kesempatan = 3;
    bool loginBerhasil = false;

    while (kesempatan > 0) {
        cout << "Masukkan Nama: ";
        cin.ignore();
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, nim);

        if (nim == adminNIM) {
            role = "admin";
        } else if (nim == userNIM) {
            role = "user";
        } else {
            kesempatan--;
            cout << "Login gagal! Kesempatan tersisa: " << kesempatan << endl;
            continue;
        }
        loginBerhasil = true;
        break;
    }

    if (!loginBerhasil) {
        cout << "\nTerlalu banyak percobaan. Program berhenti." << endl;
        return 0;
    }
}
```

Gambar 3.1

## B. Fitur menampilkan Menu Utama

Fitur ini akan menampilkan menu utama ATM dengan 5 pilihan, yaitu 1. Tambah Anime, 2. Lihat daftar Anime, 3. Update daftar Anime, 4. Hapus Anime, dan 5. Keluar. *User* dan *admin* akan diminta untuk memasukkan pilihan

```
int pilihan;
do {
    cout << "\n=== Sistem Manajemen Anime ===" << endl;
    cout << "1. Tambah Anime (Admin)" << endl;
    cout << "2. Lihat Anime" << endl;
    cout << "3. Update Anime (Admin)" << endl;
    cout << "4. Hapus Anime (Admin)" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";
    cin >> pilihan;
```

Gambar 3.2

## C. Fitur Pilihan 1: Tambah Anime

Fitur ini akan muncul jika *admin* memasukkan angka 1 saat berada di Menu Utama, fitur ini digunakan untuk melakukan penambahan data anime ke daftar

```
if (pilihan == 1 && role == "admin") {
    if (jumlahAnime >= MAX_ANIME) {
        cout << "Data anime sudah penuh!" << endl;
    } else {
        cout << "\nMasukkan data anime:" << endl;
        cout << "Judul: ";
        cin.ignore();
        getline(cin, anime[jumlahAnime].judul);
        cout << "Genre: ";
        getline(cin, anime[jumlahAnime].genre);
        cout << "Jumlah Episode: ";
        cin >> anime[jumlahAnime].jumlahEpisode;
        cout << "Jumlah Season: ";
        cin >> anime[jumlahAnime].jumlahSeason;
        cout << "Rating: ";
        cin >> anime[jumlahAnime].rating;
        jumlahAnime++;
        cout << "Anime berhasil ditambahkan!" << endl;
    }
}
```

Gambar 3.3

#### D. Fitur Pilihan 2: Lihat daftar Anime

Fitur ini akan muncul jika *user* dan *admin* memasukkan angka 2 saat berada di Menu Utama, fitur ini digunakan untuk melihat anime yang ada di daftar

```
} else if (pilihan == 2) {
    for (int i = 0; i < jumlahAnime; i++) {
        cout << "\nAnime ke-" << i + 1 << ":" << endl;
        cout << "Judul: " << anime[i].judul << endl;
        cout << "Genre: " << anime[i].genre << endl;
        cout << "Jumlah Episode: " << anime[i].jumlahEpisode <<
endl;

        cout << "Jumlah Season: " << anime[i].jumlahSeason << endl;
        cout << "Rating: " << anime[i].rating << endl;
    }
}
```

Gambar 3.4

#### E. Fitur Pilihan 3: Update daftar Anime

Fitur ini akan muncul jika *user* memasukkan angka 3 saat berada di Menu Utama, fitur ini digunakan untuk melakukan penarikan secara tunai

```
} else if (pilihan == 3 && role == "admin") {
    int index;
    cout << "Masukkan nomor anime yang ingin diupdate: ";
    cin >> index;
    if (index < 1 || index > jumlahAnime) {
        cout << "Nomor tidak valid!" << endl;
    } else {
        index--;
        cout << "\nMasukkan data baru untuk anime:" << endl;
        cout << "Judul: ";
        cin.ignore();
        getline(cin, anime[index].judul);
        cout << "Genre: ";
        getline(cin, anime[index].genre);
        cout << "Jumlah Episode: ";
        cin >> anime[index].jumlahEpisode;
        cout << "Jumlah Season: ";
        cin >> anime[index].jumlahSeason;
        cout << "Rating: ";
        cin >> anime[index].rating;
        cout << "Anime berhasil diupdate!" << endl;
    }
}
```



Gambar 3.5

#### F. Fitur Pilihan 4: Hapus Anime

Fitur ini akan muncul jika *admin* memasukkan angka 4 saat berada di Menu Utama, fitur ini digunakan agar *admin* dapat mengubah atau meng-update data anime yang ada di daftar

```
} else if (pilihan == 4 && role == "admin") {  
    int index;  
    cout << "Masukkan nomor anime yang ingin dihapus: ";  
    cin >> index;  
    if (index < 1 || index > jumlahAnime) {  
        cout << "Nomor tidak valid!" << endl;  
    } else {  
        index--;  
        for (int i = index; i < jumlahAnime - 1; i++) {  
            anime[i] = anime[i + 1];  
        }  
        jumlahAnime--;  
        cout << "Anime berhasil dihapus!" << endl;  
    }  
}
```

Gambar 3.6

#### F. Fitur Pilihan 5: Keluar

Fitur ini akan muncul jika *user* dan *admin* memasukkan angka 5 saat berada di Menu Utama, fitur ini digunakan agar *user* dan *admin* dapat keluar dari program ATM Sederhana

```
} else if (pilihan == 5) {  
    cout << "Terima kasih telah menggunakan sistem ini!" << endl;  
} else {  
    cout << "Akses ditolak atau pilihan tidak valid!" << endl;  
}  
} while (pilihan != 5);  
return 0;  
}
```

Gambar 3.7

## 4. Uji Coba dan Hasil Output

### 4.1 Hasil Output

```
Masukkan Nama: admin
Masukkan NIM: 2409106062

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat Anime
3. Update Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: █
```

Gambar 4.1.1 berhasil login

```
Masukkan Nama: salah
Masukkan NIM: 12345
Login gagal! Kesempatan tersisa: 2
Masukkan Nama: salah 2
Masukkan NIM: 54321
Login gagal! Kesempatan tersisa: 1
Masukkan Nama: █
```

Gambar 4.2 gagal kurang dari 3x

```
Masukkan Nama: salah terakhir
Masukkan NIM: 13579
Login gagal! Kesempatan tersisa: 0

Terlalu banyak percobaan. Program berhenti.
```

Gambar 4.3 Gagal sebanyak 3x

```
Masukkan Nama: admin
Masukkan NIM: 2409106062

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: █
```

Gambar 4.4 Login berhasil dan Menu Utama

```
=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 1

Masukkan data anime:
Judul: Anohana
Genre: Movie
Jumlah Episode: 1
Jumlah Season: 0
Rating: 8.12
Anime berhasil ditambahkan!
```

Gambar 4.5 Menu 1: menambahkan data anime ke daftar

```
=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 2

Anime ke-1:
Judul: Naruto
Genre: Action
Jumlah Episode: 220
Jumlah Season: 9
Rating: 8.5

Anime ke-2:
Judul: One Piece
Genre: Adventure
Jumlah Episode: 1100
Jumlah Season: 20
Rating: 9.0
```

Gambar 4.6 Tampilan menu 2: lihat daftar anime

```

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 3
Masukkan nomor anime yang ingin diupdate: 6

Masukkan data baru untuk anime:
Judul: Fate/Heaven Feel I
Genre: Movie
Jumlah Episode: 1
Jumlah Season: 0
Rating: 9.21
Anime berhasil diupdate!

```

Gambar 4.7 Menu 3: mengubah/meng-update data anime

```

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 4
Masukkan nomor anime yang ingin dihapus: 6
Anime berhasil dihapus!

```

Gambar 4.8 Menu 4: menghapus anime dari daftar

```

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 5
Terima kasih telah menggunakan program ini. Sayonara :)

```

Gambar 4.9 Menu 5: Keluar dari program

```
Masukkan Nama: user
Masukkan NIM: 2409106063

=== Sistem Manajemen Anime ===
1. Tambah Anime (Admin)
2. Lihat daftar Anime
3. Update daftar Anime (Admin)
4. Hapus Anime (Admin)
5. Keluar
Pilihan: 1
Akses ditolak atau pilihan tidak valid!
```

Gambar 4.10 user memilih pilihan yang salah

## 5. Langkah-Langkah Git

### A. Git add .

Perintah ini digunakan untuk menginisialisasi repository git dalam sebuah folder, ini berarti folder tersebut akan diubah menjadi repository git

```
PS D:\Praktikum APL\praktikum-apl\post-test\praktikum-apl\post-test\post-test-3> git add .
```

Gambar 5.1 Git add .

### B. Git commit -m "Posttest 3"

Perintah git add digunakan untuk menambahkan file apa saja sebelum dilakukannya commit

```
PS D:\Praktikum APL\praktikum-apl\post-test\praktikum-apl\post-test\post-test-3> git commit -m "Posttest 3"
[main 047f989] Posttest 3
2 files changed, 140 insertions(+)
create mode 100644 post-test/post-test-3/2409106062-RichoAnanRizkyPutra-PT-3.cpp
create mode 100644 post-test/post-test-3/2409106062-RichoAnanRizkyPutra-PT-3.exe
```

Gambar 5.2 Git commit

### C. Git push origin main

Perintah ini digunakan untuk menyimpan perubahan yang telah di tambahkan ke dalam repository git

```
PS D:\Praktikum APL\praktikum-apl\post-test\praktikum-apl\post-test\post-test-3> git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 680.71 KiB | 3.35 MiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 1 local object.
To https://github.com/Richo-ux/praktikum-apl.git
0887622..46c37b9 main -> main
```

Gambar 5.3 Git push