

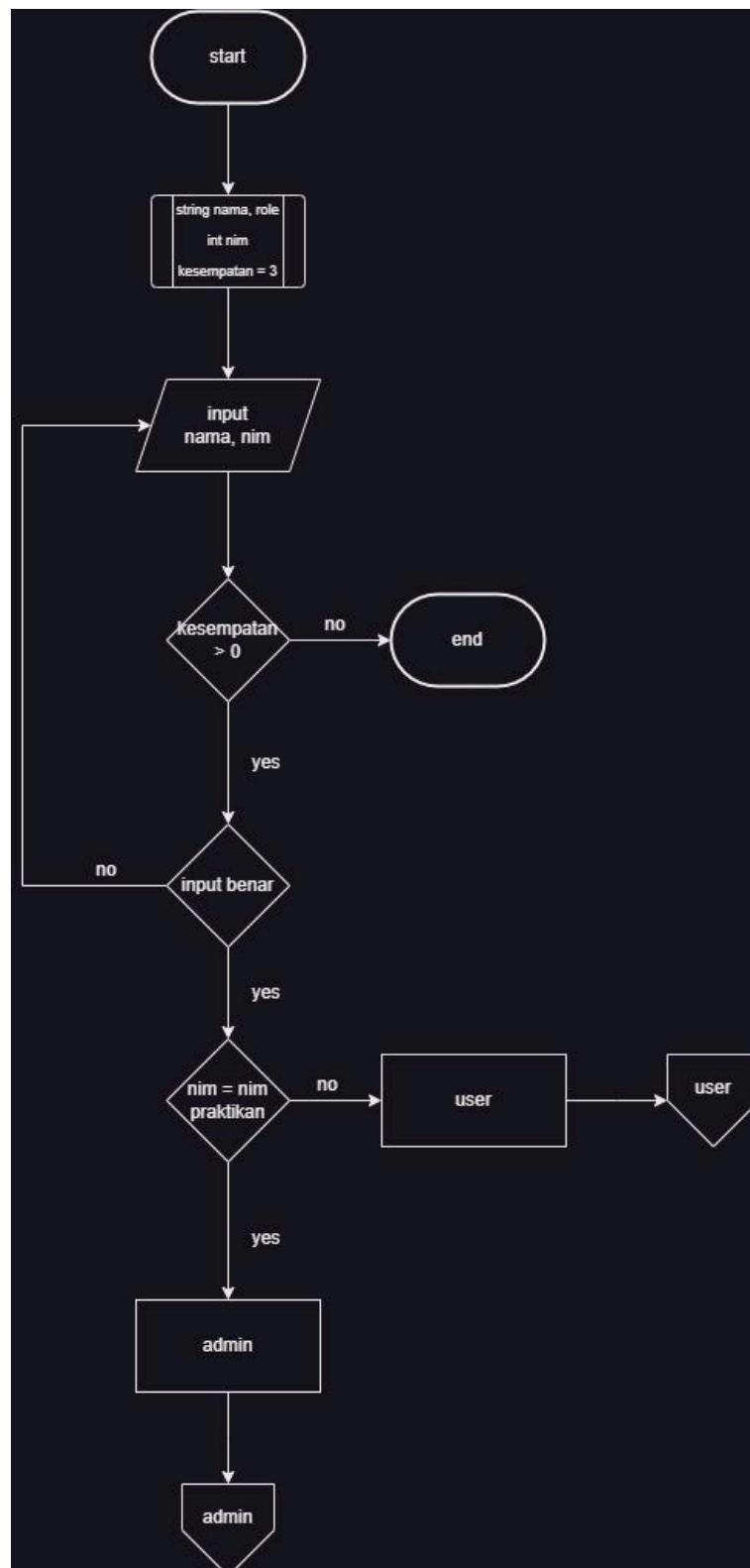
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



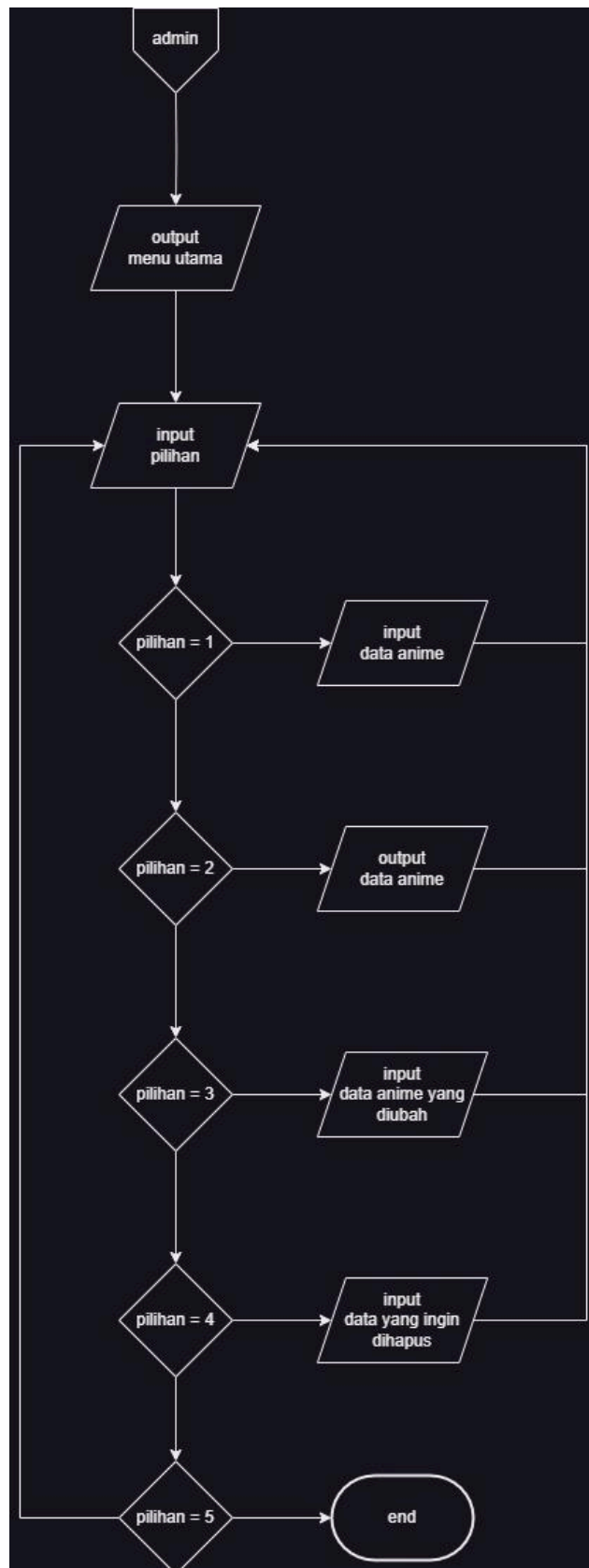
Disusun oleh:
Richo Anan Rizky Putra (2409106062)
Kelas (B1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

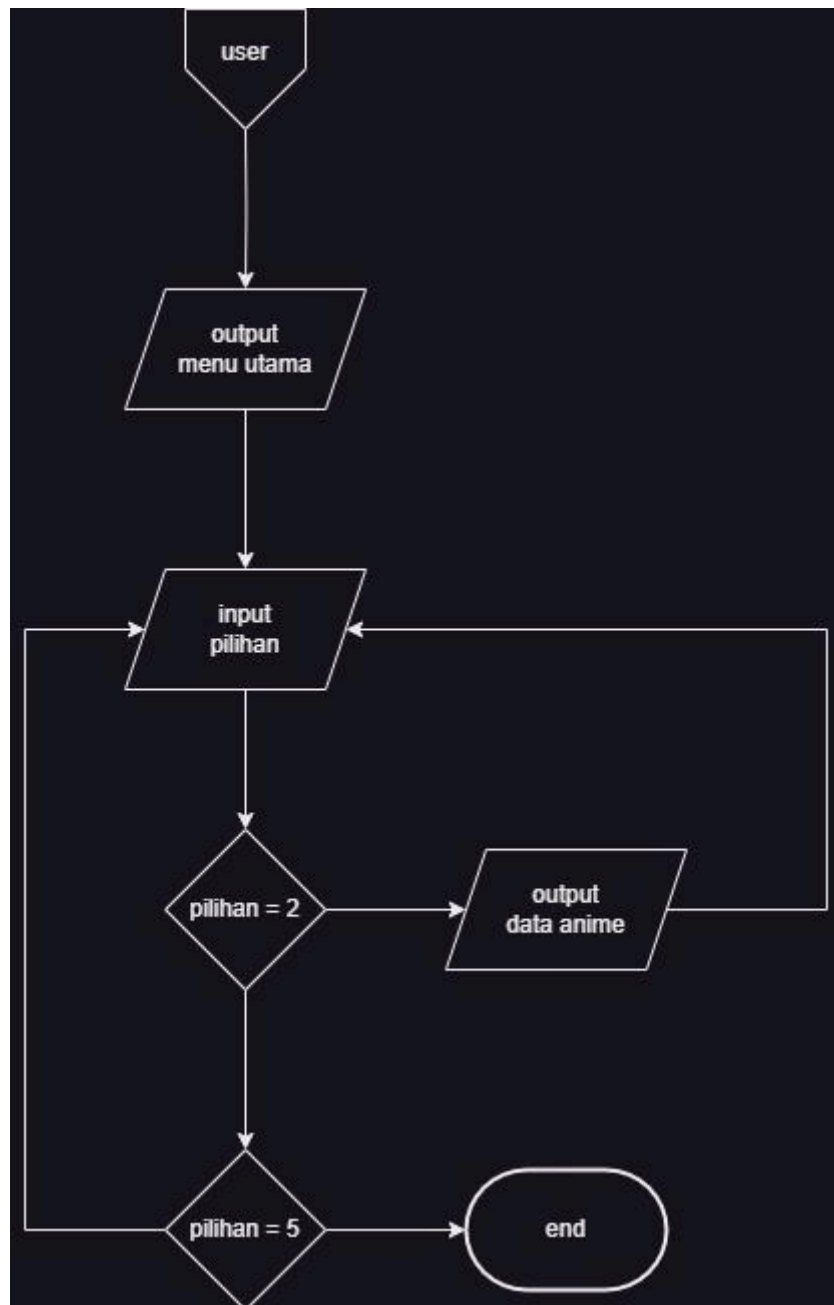
1. Flowchart



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

2.2 Penjelasan Alur & Algoritma

NIM : 2409106062 (NIM praktikan)

Total Kesempatan memasukkan NIM : 3

Jumlah awal Anime di daftar: 5 [Naruto, One Piece, Attack on Titan, Death Note, Demon Slayer]

Alur Memasukkan NIM

User dan *admin* akan diberi kesempatan sebanyak 3x untuk memasukkan NIM dengan benar, *admin* diminta untuk memasukkan NIM sesuai dengan milik praktikan, sedangkan untuk *user* dapat memasukkan angka berapapun, jika salah sebanyak 1 - 2x saat memasukkan NIM masih diberi kesempatan untuk memasukkan NIM, tapi jika salah sebanyak 3x maka akan langsung keluar dari program dengan pesan yang tertera di terminal “Terlalu banyak percobaan, Program berhenti”

Jika *user* dan *admin* berhasil memasukkan NIM dengan benar, maka *user* akan langsung masuk ke program dan *user* akan ditampilkan Menu Utama

Alur Pilihan Menu Utama

Setelah *admin* memasukkan NIM dengan benar, maka akan ditampilkan Menu Utama dengan 5 Pilihan, yaitu 1. Tambah Anime, 2. Lihat daftar Anime, 3. Update daftar Anime, 4. Hapus Anime dan 5. Keluar. Lalu *admin* akan diminta untuk memasukkan pilihan antara nomor 1,2,3,4 atau 5, Jika *user* memasukkan nomor/angka diluar yang telah diminta seperti nomor 6,7,8,9,atau 0, maka akan muncul pesan “Akses ditolak. Pilihan tidak valid” dan akan kembali ke tampilan Menu Utama dan diminta lagi memasukkan pilihan

Sedangkan untuk *user* setelah memasukkan NIM dengan benar, maka akan ditampilkan Menu Utama dengan 4 Pilihan, yaitu 1. Tambah Anime, 2. Lihat daftar Anime, 3. Update daftar Anime, 4. Hapus Anime dan 5. Keluar. Tapi *user* hanya dapat memilih antara nomor 2 dan nomor 5, Jika *user* memasukkan nomor/angka diluar yang telah diminta seperti nomor 1,3,4,5,6,7,8,9,atau 0, maka akan muncul pesan “Akses ditolak. Pilihan tidak valid” dan akan kembali ke tampilan Menu Utama dan diminta lagi memasukkan pilihan

Alur Pilihan 1: Tambah Anime

Jika *admin* memasukkan pilihan 1 di Menu Utama, yaitu Tambah Anime, selanjutnya *admin* akan diminta untuk memasukkan data anime yang ingin ditambahkan, jika sudah memasukkannya.... akan ditampilkan pesan menambahkan “Anime berhasil ditambahkan” kemudian *admin* akan dibawa kembali ke Menu Utama

Alur Pilihan 2: Lihat daftar Anime

Jika *user* dan *admin* memasukkan pilihan 2 di Menu Utama, yaitu Lihat daftar Anime, selanjutnya *user* dan *admin* akan ditampilkan pesan yang berisi Daftar Anime, Lalu *user* dan *admin* akan dibawa kembali ke Menu Utama

Alur Pilihan 3: Update daftar Anime

Jika *admin* memasukkan pilihan 3 di Menu Utama, yaitu Update daftar Anime, selanjutnya *admin* akan diminta untuk memasukkan nomor anime di daftar yang ingin diubah, jika sudah memasukkannya.... akan ditampilkan data yang ingin diubah. Jika *admin* selesai memasukkan data, lalu akan muncul pesan “Anime berhasil di update”, kemudian *admin* akan dibawa kembali ke Menu Utama

Alur Pilihan 4: Hapus Anime

Jika *admin* memasukkan pilihan 4 di Menu Utama, yaitu Hapus Anime, selanjutnya *admin* akan diminta untuk memasukkan nomor anime di daftar yang ingin dihapus, jika sudah memasukkan nomornya, anime tersebut akan terhapus dari daftar... lalu akan muncul pesan “Anime berhasil dihapus”, kemudian *admin* akan dibawa kembali ke Menu Utama

Alur Pilihan 5: Keluar

Jika *user* dan *admin* memasukkan pilihan 5 di Menu Utama, yaitu Keluar, selanjutnya *user* dan *admin* akan dibawa keluar dari Program dengan mendapatkan pesan “Terima kasih telah menggunakan program ini”
Program Selesai

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

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

    if (nim == "2409106062") { // NIM admin
        role = "admin";
        break;
    } else if (nim == "2409106063") { // NIM user
        role = "user";
        break;
    } else {
        kesempatan--;
        cout << "Login gagal! Kesempatan tersisa: " << kesempatan << endl;
        continue;
    }
    LoginBerhasil = true;
    break;
}
if (kesempatan == 0) {
    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 daftar Anime" << endl;
    cout << "3. Update daftar 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][0]);
        cout << "Genre: ";
        getline(cin, anime[jumlahAnime][1]);
        cout << "Jumlah Episode: ";
        cin >> anime[jumlahAnime][2];
        cout << "Jumlah Season: ";
        cin >> anime[jumlahAnime][3];
        cout << "Rating: ";
        cin >> anime[jumlahAnime][4];
        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][0] << endl;
        cout << "Genre: " << anime[i][1] << endl;
        cout << "Jumlah Episode: " << anime[i][2] << endl;
        cout << "Jumlah Season: " << anime[i][3] << endl;
        cout << "Rating: " << anime[i][4] << 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][0]);
        cout << "Genre: ";
        getline(cin, anime[index][1]);
        cout << "Jumlah Episode: ";
        cin >> anime[index][2];
        cout << "Jumlah Season: ";
        cin >> anime[index][3];
        cout << "Rating: ";
        cin >> anime[index][4];
        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][0] = anime[i + 1][0];
            anime[i][1] = anime[i + 1][1];
            anime[i][2] = anime[i + 1][2];
            anime[i][3] = anime[i + 1][3];
            anime[i][4] = anime[i + 1][4];
        }
        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 program 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 Uji Coba

Beberapa Uji coba yang saya lakukan :

1. *User* dan *admin* berhasil masuk kedalam menu utama dengan NIM yang benar
2. *User* dan *admin* salah memasukkan NIM, ada kesempatan 3 kali tetapi ketika sudah gagal sebanyak 3 kali maka program akan berhenti
3. Jika *user* dan *admin* berhasil masuk maka *user* dan *admin* akan berada ke dalam menu utama, yang didalamnya terdapat 5 pilihan yang bisa *user* dan *admin* pilih
4. Jika admin memilih pilihan 1 maka akan masuk ke dalam menu dari pilihan 1 yaitu tambah anime, di dalam menu pilihan 1 *admin* akan diminta untuk memasukkan data anime yang ingin ditambahkan, jika berhasil akan kembali ke menu utama
5. Jika *user* dan *admin* memilih pilihan 2 maka akan masuk ke dalam menu dari pilihan 2 yaitu lihat daftar anime, di dalam menu pilihan 2 *user* dan *admin* diperlihatkan anime yang ada di daftar, lalu *user* dan *admin* akan kembali ke tampilan menu utama
6. Jika admin memilih pilihan 3 maka akan masuk ke dalam menu dari pilihan 3 yaitu update daftar anime, di dalam menu pilihan 3 *admin* akan diminta untuk memasukkan data yang ingin diubah, setelah selesai akan kembali ke menu utama
7. Jika admin memilih pilihan 4 maka akan masuk ke dalam menu dari pilihan 4 yaitu hapus anime, di dalam menu pilihan 4 *admin* akan memasukkan nomor anime sesuai dengan yang ada di daftar yang ingin dihapus
8. Jika *user* dan *admin* memilih pilihan 5 maka akan langsung keluar dari program
9. Jika pengguna memilih pilihan 6,7,8,9, atau 0 maka akan error dan akan kembali ke tampilan Menu Utama

4.2 Hasil Output

```
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 4.1 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-2> git add .
```

Gambar 5.1 Git add .

B. Git commit -m "Posttest 2"

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-2> git commit -m "Posttest 2"
[main 539d00b] Posttest 2
 7 files changed, 709 insertions(+)
 create mode 100644 post-test/post-test-2/.$2409106062_RichoAnanRizkyPutra_PT-2.drawio.bkp
 create mode 100644 post-test/post-test-2/.$2409106062_RichoAnanRizkyPutra_PT-2.drawio.dtmp
 create mode 100644 post-test/post-test-2/2409106062_RichoAnanRizkyPutra_PT-2.cpp
 create mode 100644 post-test/post-test-2/2409106062_RichoAnanRizkyPutra_PT-2.drawio
 create mode 100644 post-test/post-test-2/2409106062_RichoAnanRizkyPutra_PT-2.exe
 create mode 100644 post-test/post-test-2/2409106062_RichoAnanRizkyPutra_PT-2.pdf
```

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-2> git push origin main
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 1.20 MiB | 758.00 KiB/s, done.
Total 11 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 1 local object.
To https://github.com/Richo-ux/praktikum-apl.git
 569b849..539d00b main -> main
```

Gambar 5.3 Git push

E. Git branch dan push

perintah git branch digunakan untuk mengelola branch dalam repository git, dan perintah git push digunakan untuk mengunggah perubahan dari repository lokal ke github

```
PS D:\Praktikum APL\praktikum-apl\post-test\post-test-apl-1> git branch -M main
PS D:\Praktikum APL\praktikum-apl\post-test\post-test-apl-1> git push -u origin main
branch 'main' set up to track 'origin/main'.
Everything up-to-date
```

Gambar 5.5 Git branch dan push