

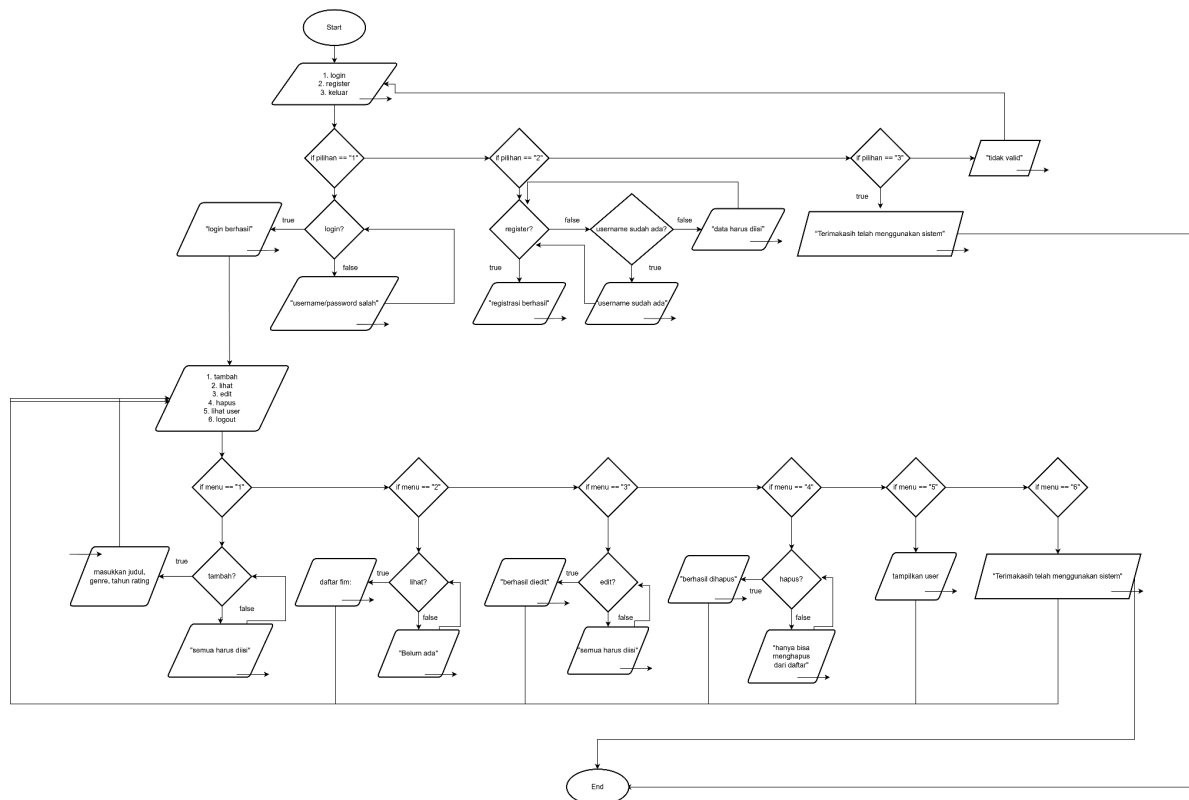
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Ammara Fadia FanySarahman (2509106035)
Kelas (A'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Deskripsi Singkat Program

Program ini merupakan sistem manajemen koleksi film/series sederhana yang memungkinkan pengguna mencatat film/series yang sudah mereka tonton (create), menaampilkan daftar film/series yang sudah mereka tonton (read), mengedit data film/series yang sudah mereka tonton (update), menghapus daftar film/series yang sudah ada (delete).

3. Source Code

A. Menu utama

Ini digunakan untuk memilih login, register, keluar dari program.

Source Code:

```
while True:
```

```

clear_screen()
print("=== SISTEM MANAJEMEN FILM/SERIES ===")
print("1. Login")
print("2. Register")
print("3. Keluar")

pilihan = input("Pilih menu (1-3): ")

```

B. Login (pilihan 1)

Ini digunakan untuk melakukan login dan mengecek status login.

Source Code:

```

if pilihan == "1":
    clear_screen()
    print("=== LOGIN ===")
    username = input("Username: ")
    password = input("Password: ")

    # Cek Login
    login_sukses = False
    for user in users:
        if user[0] == username and user[1] == password:
            current_user = user
            login_sukses = True
            break

    if login_sukses:
        print(f"Login berhasil! Selamat datang {username}")
        input("Tekan Enter untuk melanjutkan...")
    ...
    else:
        print("Error: Username atau password salah!")
        input("Tekan Enter untuk melanjutkan...")

```

C. Menu (setelah login)

Menampilkan 6 pilihan berupa tambah film/series, lihat film/series, edit film/series, hapus film/series, lihat user (khusus admin), logout.

Source Code:

```

while True:
    clear_screen()
    print(f"=== MENU UTAMA ({current_user[0]}) ===")
    print("1. Tambah Film/Series")
    print("2. Lihat Semua Film/Series")
    print("3. Edit Film/Series")
    print("4. Hapus Film/Series")
    if current_user[2] == "admin":
        print("5. Lihat Semua User")
    print("6. Logout")

    menu = input("Pilih menu (1-6): ")

```

D. Tambah

Untuk menambah daftar film/series baru.

Source Code:

```

if menu == "1": # Tambah film/series
    clear_screen()
    print("=== TAMBAH FILM/SERIES ===")
    judul = input("Judul film/series: ")
    genre = input("Genre: ")
    tahun = input("Tahun rilis: ")
    rating = input("Rating (1-10): ")

    if judul and genre and tahun and rating:
        film_baru = [judul, genre, tahun, rating,
current_user[0]]
        movies.append(film_baru)
        print("Film/Series berhasil ditambahkan!")
    else:
        print("Error: Semuanya harus diisi!")
        input("Tekan Enter untuk melanjutkan...")

elif menu == "2": # Lihat semua film/series
    clear_screen()
    print("=== DAFTAR FILM/SERIES ===")
    if not movies:
        print("Belum ada film/series dalam koleksi.")
    else:
        for i, film in enumerate(movies, 1):
            print(f"{i}. {film[0]} - {film[1]} ({film[2]}) -

```

```
Rating: {film[3]} - Ditambahkan oleh: {film[4]}")
    input("Tekan Enter untuk melanjutkan...")
```

E. Edit

Untuk mengedit daftar yang sudah ada.

Source Code:

```
elif menu == "3": # Edit film/series
    clear_screen()
    print("=== EDIT FILM/SERIES ===")
    if not movies:
        print("Belum ada film/series dalam koleksi.")
    else:
        for i, film in enumerate(movies, 1):
            print(f"{i}. {film[0]} - {film[1]} ({film[2]}) -
Rating: {film[3]}")

        nomor = input("Pilih nomor film/series yang akan
diedit: ")

        if nomor.isdigit():
            index = int(nomor) - 1
            if 0 <= index < len(movies):
                film = movies[index]
                # Cek hak akses
                if film[4] == current_user[0] or
current_user[2] == "admin":

                    print(f"Edit film: {film[0]}")
                    judul_baru = input(f"Judul baru
({film[0]}): ") or film[0]

                    genre_baru = input(f"Genre baru
({film[1]}): ") or film[1]

                    tahun_baru = input(f"Tahun baru
({film[2]}): ") or film[2]

                    rating_baru = input(f"Rating baru
({film[3]}): ") or film[3]

                    movies[index] = [judul_baru, genre_baru,
tahun_baru, rating_baru, film[4]]
                    print("Film/Series berhasil diupdate!")
                else:
                    print("Error: Anda hanya bisa mengedit
film/series yang Anda tambahkan!")
```

```

        else:
            print("Error: Nomor film/series tidak
valid!")

        else:
            print("Error: Input harus angka!")
            input("Tekan Enter untuk melanjutkan...")

```

F. Hapus

Untuk menghapus daftar yang sudah ada.

Source Code:

```

elif menu == "4": # Hapus film/series
    clear_screen()
    print("=== HAPUS FILM/SERIES ===")
    if not movies:
        print("Belum ada film/series dalam koleksi.")
    else:
        for i, film in enumerate(movies, 1):
            print(f"{i}. {film[0]} - {film[1]} ({film[2]}) -
Rating: {film[3]}")

        nomor = input("Pilih nomor film/series yang akan
dihapus: ")

        if nomor.isdigit():
            index = int(nomor) - 1
            if 0 <= index < len(movies):
                film = movies[index]
                # Cek hak akses
                if film[4] == current_user[0] or
current_user[2] == "admin":
                    konfirmasi = input(f"Yakin hapus
film/series '{film[0]}'? (yes/no): ")
                    if konfirmasi.lower() == 'yes':
                        movies.pop(index)
                        print("Film/Series berhasil
dihapus!")
                    else:
                        print("Error: Anda hanya bisa menghapus
film/series yang Anda tambahkan!")
                else:
                    print("Error: Nomor film/series tidak
valid!")

```

```
        else:
            print("Error: Input harus angka!")
            input("Tekan Enter untuk melanjutkan...")
```

G. Lihat user

Untuk melihat daftar user yang ada (khusus admin).

Source Code:

```
elif menu == "5" and current_user[2] == "admin": # Lihat semua user (admin only)

    clear_screen()
    print("=== DAFTAR USER ===")
    for i, user in enumerate(users, 1):
        print(f"{i}. {user[0]} - Role: {user[2]}")
    input("Tekan Enter untuk melanjutkan...")

    elif menu == "6": # Logout
        current_user = None
        print("Logout berhasil!")
        input("Tekan Enter untuk melanjutkan...")
        break

    else:
        print("Pilihan tidak valid!")
        input("Tekan Enter untuk melanjutkan...")

else:
    print("Error: Username atau password salah!")
    input("Tekan Enter untuk melanjutkan...")
```

H. Register (pilihan 2)

Ini digunakan untuk melakukan daftar pengguna baru.

Source Code:

```
elif pilihan == "2":
    clear_screen()
    print("=== REGISTER ===")
```

```

username = input("Username: ")
password = input("Password: ")

# Cek apakah username sudah ada
username_ada = False
for user in users:
    if user[0] == username:
        username_ada = True
        break

if username_ada:
    print("Error: Username sudah digunakan!")
elif username and password:
    user_baru = [username, password, "user"]
    users.append(user_baru)
    print("Registrasi berhasil! silahkan login.")
else:
    print("Error: Username dan password harus diisi!")
input("Tekan Enter untuk melanjutkan...")

```

I. Logout (pilihan 3)

Ini digunakan untuk logout.

Source Code:

```

elif pilihan == "3":
    print("Terima kasih telah menggunakan sistem!")
    break

```

4. Hasil Output

```

=== SISTEM MANAJEMEN FILM/SERIES ===
1. Login
2. Register
3. Keluar
Pilih menu (1-3): █

```

Gambar 4.1 Output pertama
Menu utama.


```
=== LOGIN ===  
Username: admin  
Password: akuadminkausiapa  
Login berhasil! Selamat datang admin  
Tekan Enter untuk melanjutkan...
```

Gambar 4.2 Output kedua
Login.

```
=== MENU UTAMA (admin) ===  
1. Tambah Film/Series  
2. Lihat Semua Film/Series  
3. Edit Film/Series  
4. Hapus Film/Series  
5. Lihat Semua User  
6. Logout  
Pilih menu (1-6): 1
```

Gambar 4.3 Output ketiga
Menu setelah login.

```
=== TAMBAH FILM/SERIES ===  
Judul film/series: Fast X  
Genre: Action, Adventure, Crime  
Tahun rilis: 2023  
Rating (1-10): 9  
Film/Series berhasil ditambahkan!  
Tekan Enter untuk melanjutkan...
```

Gambar 4.4 Output keempat
Menambah.

```
=== DAFTAR FILM/SERIES ===  
1. Fast X - Action, Adventure, Crime (2023) - Rating: 9 - Ditambahkan oleh: admin  
Tekan Enter untuk melanjutkan...
```

Gambar 4.5 Output kelima
Cek daftar.

```
=== EDIT FILM/SERIES ===
1. Fast X - Action, Adventure, Crime (2023) - Rating: 9
Pilih nomor film/series yang akan diedit: 1
Edit film: Fast X
Judul baru (Fast X): Fast X
Genre baru (Action, Adventure, Crime): Action, Adventure, Crime
Tahun baru (2023): 2023
Rating baru (9): 9.5
Film/Series berhasil diupdate!
Tekan Enter untuk melanjutkan...
```

Gambar 4.6 Output keenam
Update.

```
=== DAFTAR FILM/SERIES ===
1. Fast X - Action, Adventure, Crime (2023) - Rating: 9.5 - Ditambahkan oleh: admin
Tekan Enter untuk melanjutkan...
```

Gambar 4.7 Output ketujuh
Cek setelah update.

```
=== HAPUS FILM/SERIES ===
1. Fast X - Action, Adventure, Crime (2023) - Rating: 9.5
Pilih nomor film/series yang akan dihapus: 1
Yakin hapus film/series 'Fast X'? (yes/no): yes
Film/Series berhasil dihapus!
Tekan Enter untuk melanjutkan...
```

Gambar 4.8 Output kedelapan
Menghapus.

```
=== DAFTAR FILM/SERIES ===
Belum ada film/series dalam koleksi.
Tekan Enter untuk melanjutkan...
```

Gambar 4.9 Output kesembilan
Cek setelah dihapus.

```
=== DAFTAR USER ===
1. admin - Role: admin
Tekan Enter untuk melanjutkan...
```

Gambar 4.10 Output kesepuluh

Daftar user.

```
=== MENU UTAMA (admin) ===
1. Tambah Film/Series
2. Lihat Semua Film/Series
3. Edit Film/Series
4. Hapus Film/Series
5. Lihat Semua User
6. Logout
Pilih menu (1-6): 6
Logout berhasil!
Tekan Enter untuk melanjutkan...
```

Gambar 4.11 Output kesebelas
Logout.

5. Langkah-langkah GIT

```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS\Documents\praktikum-apd>git add .

C:\Users\ASUS\Documents\praktikum-apd>git commit -m "PT 5"
[main be70efd] PT 5
2 files changed, 270 insertions(+)
create mode 100644 kelas/pertrmuan-5/main.py
create mode 100644 post-test/post-test-apd-5/2509106035-Ammara Fadia-PT-5.py

C:\Users\ASUS\Documents\praktikum-apd>git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 2.75 KiB | 312.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/ammaraFadia/praktikum-apd.git
   bc86f71..be70efd  main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\ASUS\Documents\praktikum-apd>
```

Gambar 5.1
GIT.

5.1 GIT Add

Add adalah perintah yang berguna untuk staging atau untuk menambahkan perubahan atau registrasi perubahan pada git proyek yang ada.

5.2 GIT Commit

Commit adalah perintah yang digunakan untuk menyimpan perubahan pada repository git. Kita juga bisa mengembalikan file yang kita commit ke revisi sebelumnya.

5.3 GIT Push

Push adalah perintah yang digunakan untuk mengunggah konten pada repositori lokal.