

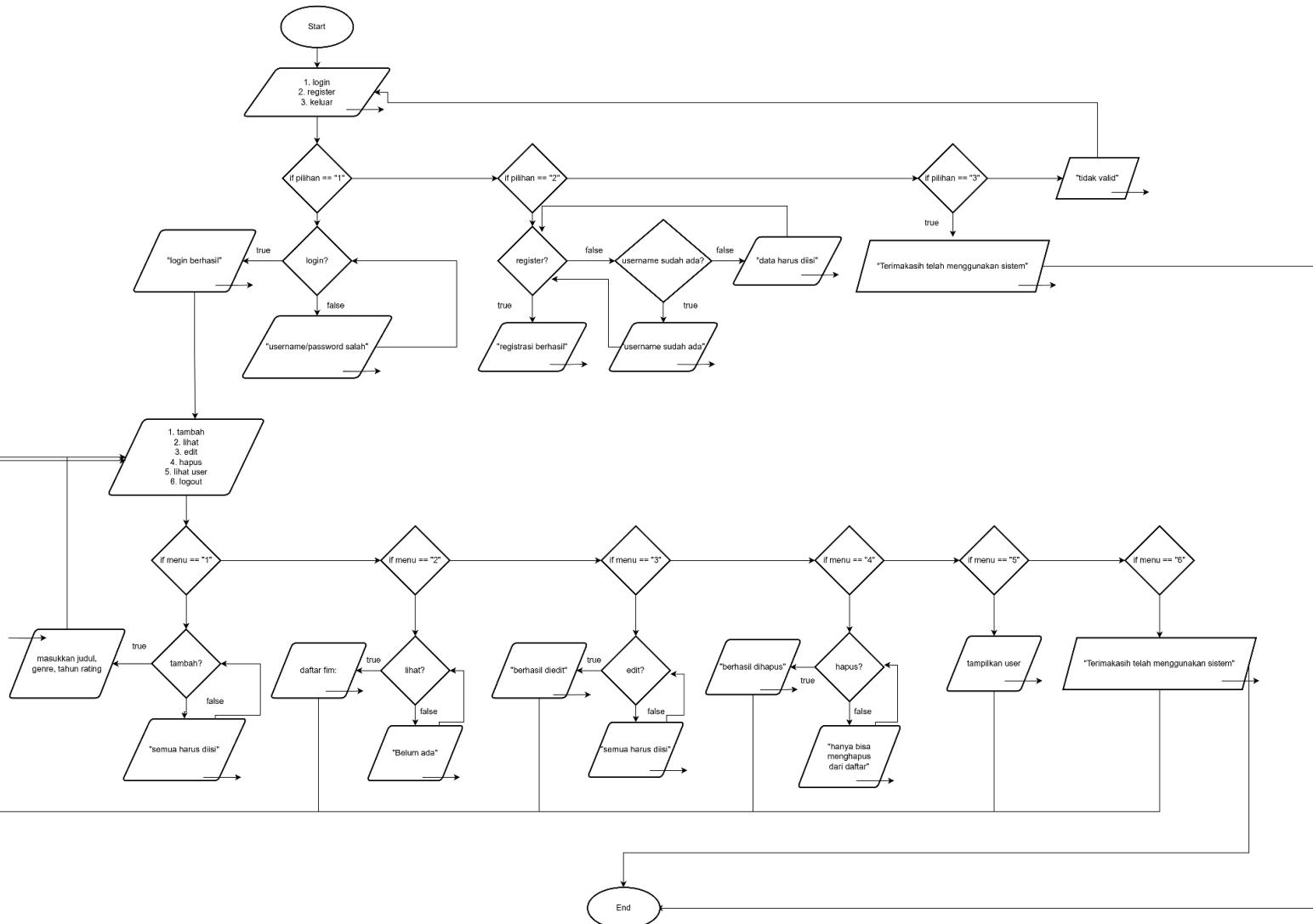
**LAPORAN PRAKTIKUM**  
**POSTTEST 8**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**  
**Ammara Fadia Fany Sarahman (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), menampilkan 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. Fungsi Utility

melakukan tugas-tugas kecil, umum, dan berulang yang mendukung fungsi utama program.

**Source Code:**

```
import os

def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')

def press_enter_to_continue():
    input("\nTekan Enter untuk melanjutkan...")

def view_all_users(users, PrettyTable):
    clear_screen()
    print("== DAFTAR USER ==")
    if not users:
        print("Belum ada user terdaftar.")
    else:
        table = PrettyTable()
        table.field_names = ["No.", "Username", "Role"]
        table.align["Username"] = "l"
        table.align["Role"] = "c"

        for i, user in enumerate(users, 1):
            table.add_row([i, user['username'], user['role']])

        print(table)
    press_enter_to_continue()
```

#### B. Fungsi user/auth

mengelola identitas dan hak akses pengguna dalam sistem (admin memiliki hak untuk menghapus dan mengedit).

**Source Code:**

```
from utils import clear_screen, press_enter_to_continue

def check_credentials(users, username, password):
    return next((user for user in users if user["username"] == username and
```

```

user["password"] == password), None)

def register(users):
    clear_screen()
    print("==> REGISTER ===")
    username = input("Username: ")
    password = input("Password: ")

    if any(user["username"] == username for user in users):
        print("Error: Username sudah digunakan!")
    elif username and password:
        new_user = {
            "username": username,
            "password": password,
            "role": "user"
        }
        users.append(new_user)
        print("Registrasi berhasil! Silakan login.")
    else:
        print("Error: Username dan password harus diisi!")
    press_enter_to_continue()

def login(users):
    clear_screen()
    print("==> LOGIN ===")
    username = input("Username: ")
    password = input("Password: ")

    found_user = check_credentials(users, username, password)

    if found_user:
        print(f"Login berhasil! Selamat datang {username}")
        press_enter_to_continue()
        return found_user
    else:
        print("Error: Username atau password salah!")
        press_enter_to_continue()
        return None

```

## C. Fungsi Manajemen Film/Series

Digunakan untuk menambah, melihat, mengedit, dan menghapus film.

### Source Code:

```

def add_movie():

```

```

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

if not all([judul, genre, tahun, rating]):
    print("Error: Semuanya harus diisi!")
elif not is_title_unique(judul):
    print("Error: Film/series dengan judul tersebut sudah ada!")
elif not rating.isdigit() or not (1 <= int(rating) <= 10):
    print("Error: Rating harus angka antara 1-10!")
else:
    new_movie = {
        "judul": judul,
        "genre": genre,
        "tahun": tahun,
        "rating": rating,
        "added_by": current_user["username"]
    }
    movies.append(new_movie)
    print("Film/Series berhasil ditambahkan!")
press_enter_to_continue()

def view_movies():
    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['judul']} - {film['genre']} ({film['tahun']})")
- Rating: {film['rating']} - Ditambahkan oleh: {film['added_by']}")
    press_enter_to_continue()

def edit_movie():
    global movies
    clear_screen()
    print("== EDIT FILM/SERIES ==")

    def update_judul(film):
        film['judul'] = input(f"Judul baru ({film['judul']}): ") or
        film['judul']

    def update_genre(film):
        film['genre'] = input(f"Genre baru ({film['genre']}): ") or

```

```

film['genre']

def update_tahun(film):
    film['tahun'] = input(f"Tahun baru ({film['tahun']}): ") or
film['tahun']

def validate_and_update_rating(film):
    rating_baru = input(f"Rating baru ({film['rating']}): ") or
film['rating']
    if rating_baru.isdigit() and (1 <= int(rating_baru) <= 10):
        film['rating'] = rating_baru
        return True
    else:
        print("Error: Rating harus angka antara 1-10! Rating tidak
diubah.")
        return False

def check_admin_access():
    if current_user["role"] != "admin":
        print("Error: Hanya admin yang dapat mengedit film/series!")
        press_enter_to_continue()
        return False
    return True

if not movies:
    print("Belum ada film/series dalam koleksi.")
    press_enter_to_continue()
    return

if not check_admin_access():
    return

for i, film in enumerate(movies, 1):
    print(f"{i}. {film['judul']} - {film['genre']} ({film['tahun']}) -
Rating: {film['rating']}")

nomor = input("Pilih nomor film/series yang akan diedit: ")
if nomor.isdigit():
    index = int(nomor) - 1
    if 0 <= index < len(movies):
        film = movies[index]
        print(f"\nEdit film: {film['judul']}")

        update_judul(film)
        update_genre(film)
        update_tahun(film)

        success = validate_and_update_rating(film)

```

```

        if success:
            print("Film/Series berhasil diupdate!")

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

press_enter_to_continue()

def delete_movie():
    global movies
    clear_screen()
    print("== HAPUS FILM/SERIES ==")
    if not movies:
        print("Belum ada film/series dalam koleksi.")
        press_enter_to_continue()
        return

    if current_user["role"] != "admin":
        print("Error: Hanya admin yang dapat menghapus film/series!")
        press_enter_to_continue()
        return

    for i, film in enumerate(movies, 1):
        print(f"{i}. {film['judul']} - {film['genre']} ({film['tahun']}) - "
Rating: {film['rating']}")

nomor = input("Pilih nomor film/series yang akan dihapus: ")
if not nomor.isdigit():
    print("Error: Input harus angka!")
    press_enter_to_continue()
    return

index = int(nomor) - 1
if not (0 <= index < len(movies)):
    print("Error: Nomor film/series tidak valid!")
    press_enter_to_continue()
    return

film_to_delete = movies[index]
konfirmasi = input(f"Yakin hapus film/series
'{film_to_delete['judul']}'? (yes/no): ")
if konfirmasi.lower() == 'yes':
    movies.pop(index)
    print("Film/Series berhasil dihapus!")
else:
    print("Batal menghapus.")

```

```
press_enter_to_continue()
```

#### D. Fungsi Menu

Untuk menampilkan menu.

Source Code:

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

        menu = input("Pilih menu: ")

        if menu == "1"
```

#### D. Fungsi Menu

Untuk menampilkan menu.

Source Code:

```
from prettytable import PrettyTable
from auth import login, register
from movie_manager import add_movie, view_movies, edit_movie, delete_movie
from utils import clear_screen, press_enter_to_continue, view_all_users

users = [
    {"username": "admin", "password": "akuadmindinkausiapa", "role": "admin"},
]
movies = []
current_user = None

def main_menu_logged_in():
    global current_user, movies, users
```

```

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

menu = input("Pilih menu: ")

if menu == "1":
    add_movie(movies, current_user)
elif menu == "2":
    view_movies(movies)
elif menu == "3":
    edit_movie(movies, current_user)
elif menu == "4":
    delete_movie(movies, current_user)
elif menu == "5" and current_user["role"] == "admin":
    view_all_users(users, PrettyTable) # Meneruskan users dan
PrettyTable
elif menu == "6":
    current_user = None
    print("Logout berhasil!")
    press_enter_to_continue()
    break
else:
    print("Pilihan tidak valid!")
    press_enter_to_continue()

def main_menu():
    global current_user, users

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

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

if pilihan == "1":
    logged_in_user = login(users)

```

```
if logged_in_user:
    current_user = logged_in_user
    main_menu_logged_in()
elif pilihan == "2":
    register(users)
elif pilihan == "3":
    print("Terima kasih telah menggunakan sistem!")
    break
else:
    print("Pilihan tidak valid!")
    press_enter_to_continue()

if __name__ == "__main__":
    main_menu()
```

#### 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 ===
+-----+-----+-----+-----+-----+
| No. | Judul | Genre | Tahun | Rating | Ditambahkan Oleh |
+-----+-----+-----+-----+-----+
| 1 | Fast X | Action | 2023 | 9 | admin |
+-----+-----+-----+-----+-----+
Tekan Enter untuk melanjutkan...■
```

Gambar 4.5 Output kelima  
Cek daftar.

```
==== EDIT FILM/SERIES ====
+-----+-----+-----+-----+
| No. | Judul | Genre | Tahun | Rating |
+-----+-----+-----+-----+
| 1   | Fast X | Action | 2023 | 9      |
+-----+-----+-----+-----+
Pilih nomor film/series yang akan diedit: 1

Edit film: Fast X
Judul baru (Fast X):
Genre baru (Action):
Tahun baru (2023):
Rating baru (9): 10
Film/Series berhasil diupdate!
```

Tekan Enter untuk melanjutkan... █

Gambar 4.6 Output keenam

Update.

```
==== DAFTAR FILM/SERIES ====
+-----+-----+-----+-----+-----+
| No. | Judul | Genre | Tahun | Rating | Ditambahkan Oleh |
+-----+-----+-----+-----+-----+
| 1   | Fast X | Action | 2023 | 10    | admin       |
+-----+-----+-----+-----+-----+
```

Tekan Enter untuk melanjutkan... █

Gambar 4.7 Output ketujuh

Cek setelah update.

```
==== HAPUS FILM/SERIES ====
+-----+-----+-----+-----+
| No. | Judul | Genre | Tahun | Rating |
+-----+-----+-----+-----+
| 1 | Fast X | Action | 2023 | 10 |
+-----+-----+-----+-----+
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

```
C:\Users\ASUS\Documents\praktikum-apd>git push -u origin main
Enumerating objects: 22, done.
Counting objects: 100% (22/22), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (17/17), 9.15 KiB | 852.00 KiB/s, done.
Total 17 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/ammarafadia/praktikum-apd.git
  b2b527b..e2ff41b  main -> main
branch 'main' set up to track 'origin/main'.

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

```
C:\Users\ASUS\Documents\praktikum-apd>git add .
C:\Users\ASUS\Documents\praktikum-apd>git commit -m "pt8"
[main e2ff41b] pt8
9 files changed, 349 insertions(+), 1 deletion(-)
create mode 100644 kelas/pertemuan-8/main.py
create mode 100644 post-test/post-test-apd-8/__pycache__/auth.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/__pycache__/movie_manager.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/__pycache__/utils.cpython-313.pyc
create mode 100644 post-test/post-test-apd-8/auth.py
create mode 100644 post-test/post-test-apd-8/main.py
create mode 100644 post-test/post-test-apd-8/movie_manager.py
create mode 100644 post-test/post-test-apd-8/utils.py
```

Gambar 5.1
GIT.

### 5.1 GIT Add

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