

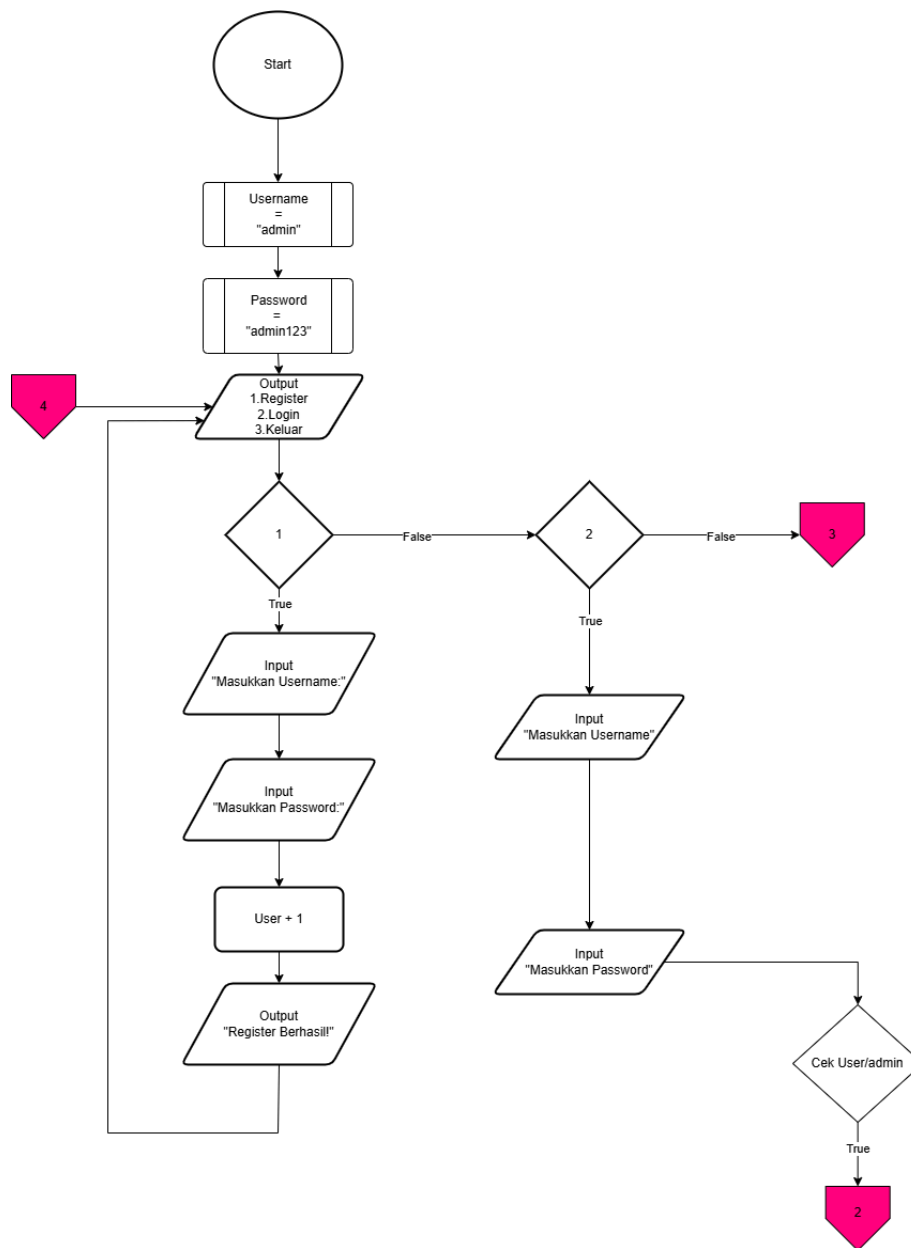
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



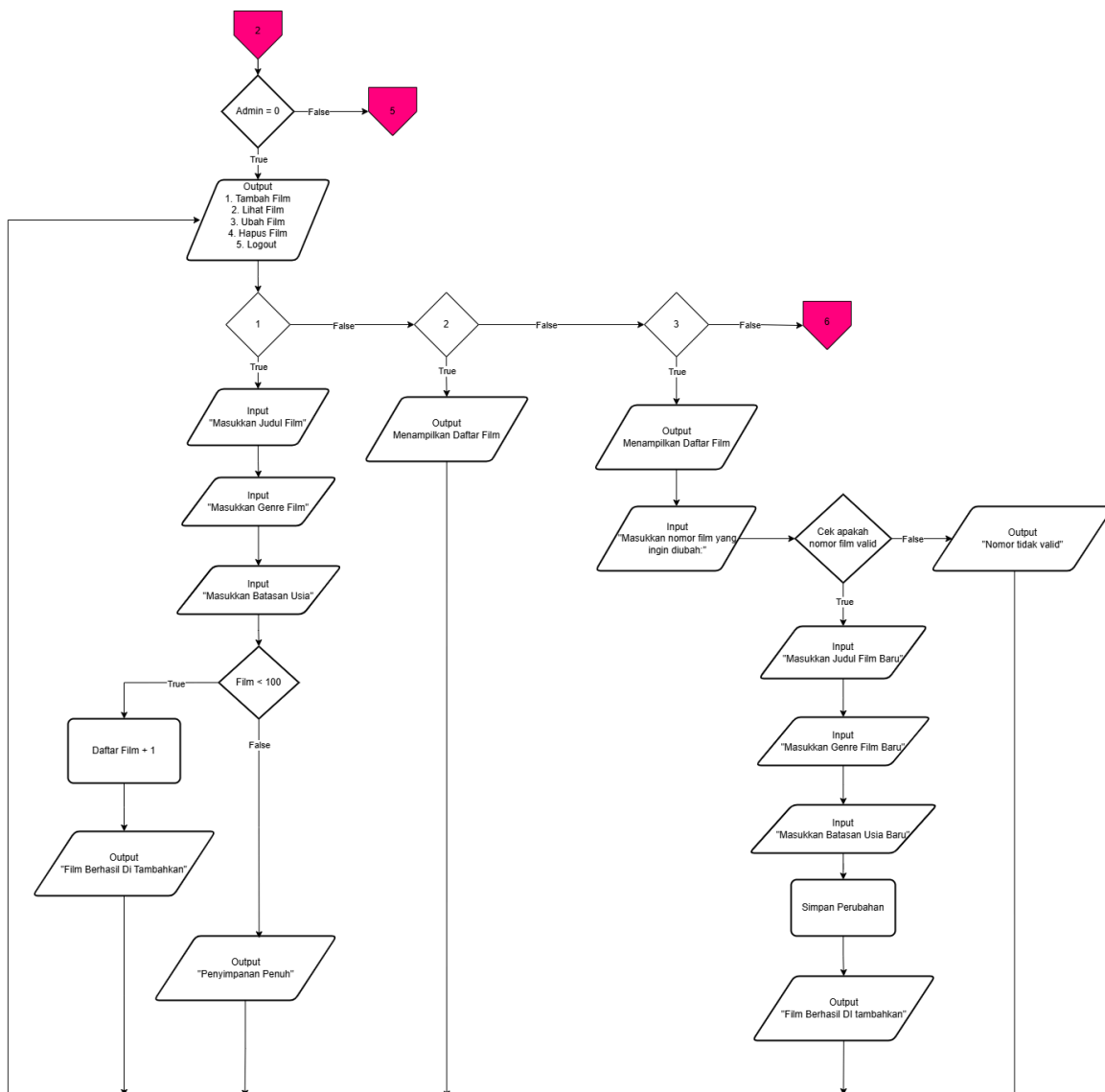
**Disusun oleh:**  
**Muhammad Rizky Adha Putra (2409106113)**  
**Kelas (C2'24)**

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

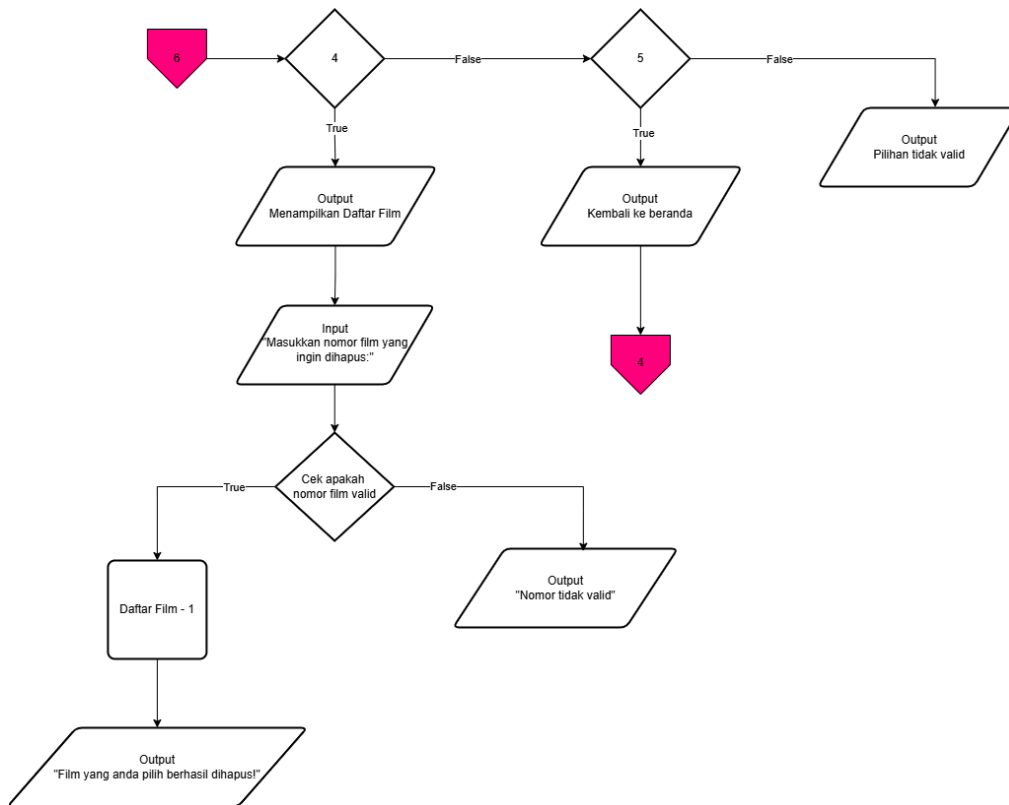
## 1. Flowchart



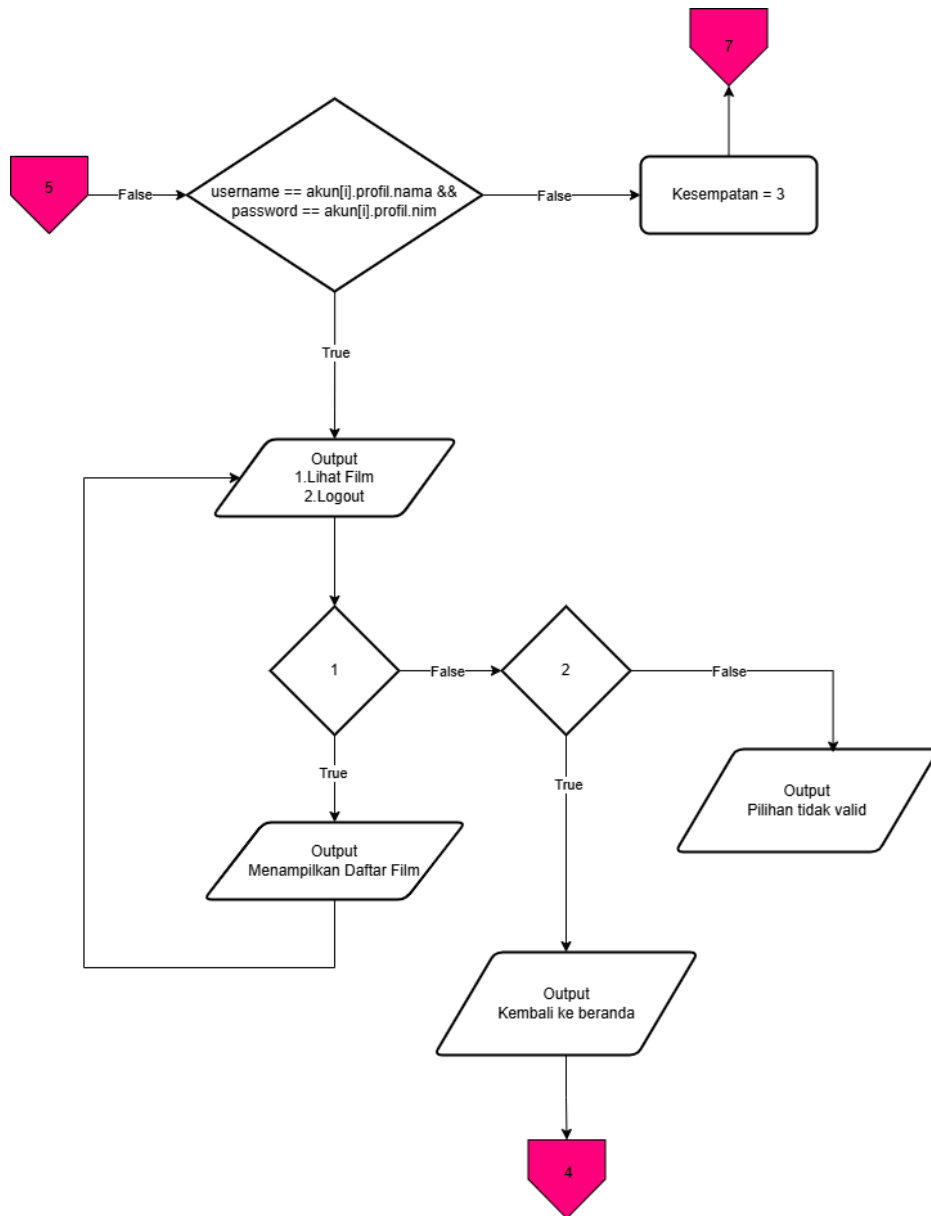
Gambar 1.1  
Flowchart



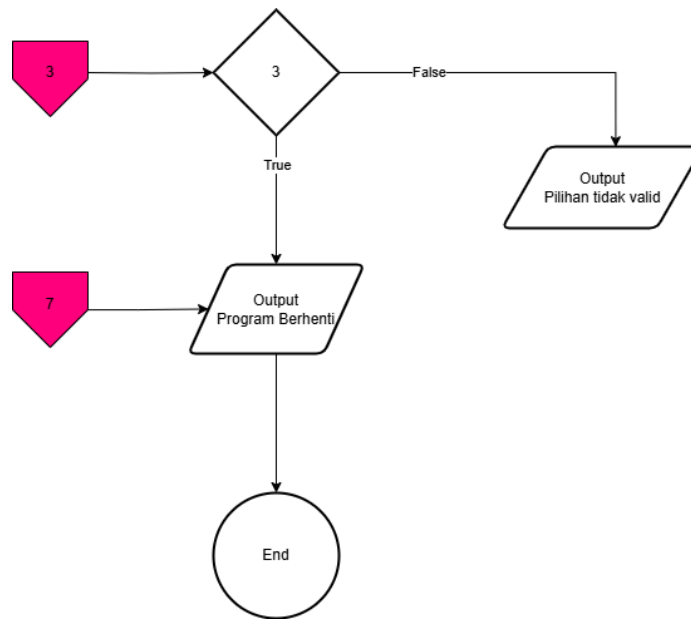
Gambar 1.2  
Flowchart



Gambar 1.3  
Flowchart



Gambar 1.4  
Flowchart



Gambar 1.5  
Flowchart

## **2. Analisis Program**

Pada post test 3 ini saya membuat APLIX sebuah program berbasis C++ yang digunakan untuk mengelola daftar film. Program ini memungkinkan dua jenis pengguna, yaitu Admin dan User, untuk berinteraksi dengan data film. Admin memiliki kontrol penuh atau CRUD yang dimana biasanya untuk menambahkan, mengedit, dan menghapus film, sedangkan User ada fitur tambahan yaitu register terlebih dahulu sebelum login dan ketika sudah selesai register maka baru bisa login lalu melihat daftar film yang tersedia di APLIX, lalu di codingan membuat menjadi rapi.

### 3. Source Code

#### A. Fitur Fungsi Tambah Film

Source Code:

```
void tambahFilm(Film daftarFilm[], int& filmC) {
    if (filmC < 100) {
        cout << "Masukkan Judul Film: ";
        getline(cin, daftarFilm[filmC].judul);
        cout << "Masukkan Genre Film: ";
        getline(cin, daftarFilm[filmC].genre);
        cout << "Masukkan Usia Minimal: ";
        cin >> daftarFilm[filmC].usia;
        cin.ignore();
        filmC++;
        cout << "Film berhasil ditambahkan!" << endl;
    } else {
        cout << "Penyimpanan penuh!" << endl;
    }
}
```

#### B. Fitur Fungsi Lihat Film

Source Code:

```
void lihatFilm(const Film daftarFilm[], int filmC) {
    cout << "Daftar Film:" << endl;
    cout << "+-----+" << endl;
    cout << "| No | Judul          | Genre | Usia |" << endl;
    cout << "+-----+" << endl;
    for (int i = 0; i < filmC; ++i) {
        cout << "| " << setw(2) << i + 1 << " | "
              << setw(14) << daftarFilm[i].judul << " | "
              << setw(6) << daftarFilm[i].genre << " | "
              << setw(4) << daftarFilm[i].usia << " |" << endl;
    }
    cout << "+-----+" << endl;
}
```



### C. Fitur Fungsi Ubah Film

#### Source Code:

```
void ubahFilm(Film daftarFilm[], int filmC) {
    lihatFilm(daftarFilm, filmC);
    int index;
    cout << "Masukkan nomor film yang ingin diubah: ";
    cin >> index;
    cin.ignore();
    if (index > 0 && index <= filmC) {
        cout << "Masukkan judul baru: ";
        getline(cin, daftarFilm[index - 1].judul);
        cout << "Masukkan genre baru: ";
        getline(cin, daftarFilm[index - 1].genre);
        cout << "Masukkan usia minimal baru: ";
        cin >> daftarFilm[index - 1].usia;
        cin.ignore();
        cout << "Film berhasil diubah!" << endl;
    } else {
        cout << "Nomor film tidak valid!" << endl;
    }
}
```

### D. Fitur Fungsi Hapus Film

#### Source Code:

```
void hapusFilm(Film daftarFilm[], int& filmC) {
    lihatFilm(daftarFilm, filmC);
    int index;
    cout << "Masukkan nomor film yang ingin dihapus: ";
    cin >> index;
    cin.ignore();
    if (index > 0 && index <= filmC) {
        for (int i = index - 1; i < filmC - 1; ++i) {
            daftarFilm[i] = daftarFilm[i + 1];
        }
        filmC--;
        cout << "Film berhasil dihapus!" << endl;
    } else {
        cout << "Nomor film tidak valid!" << endl;
    }
}
```

## E. Fitur Fungsi Login

Source Code:

```
bool login(User akun[], int userC, bool& isAdmin, int& loginIndex) {
    int kesempatan = 3;
    while (kesempatan > 0) {
        string username, password;
        cout << "Masukkan Nama: ";
        getline(cin, username);
        cout << "Masukkan Password: ";
        getline(cin, password);

        for (int i = 0; i < userC; i++) {
            if (username == akun[i].profil.nama && password ==
akun[i].profil.nim) {
                isAdmin = (i == 0);
                loginIndex = i;
                return true;
            }
        }

        kesempatan--;
        cout << "User atau password salah! Sisa percobaan: " << kesempatan
<< endl;
    }
}
```

## F. Fitur Fungsi Login

Source Code:

```
void registrasi(User akun[], int& userC) {
    if (userC < 10) {
        cout << "----- Registrasi Akun Baru -----" << endl;
        cout << "Masukkan Username: ";
        getline(cin, akun[userC].profil.nama);
        cout << "Masukkan Password: ";
        getline(cin, akun[userC].profil.nim);
        userC++;
        cout << "Registrasi berhasil! Silahkan login untuk melanjutkan." <<
endl;
    } else {
        cout << "Maksimum user tercapai!" << endl;
    }
}
```

## 4. Uji Coba dan Hasil Output

### A.Login Gagal

```
+-----+
|      APLIX      |
+-----+
| 1. Register Akun |
| 2. Login         |
| 3. Keluar        |
+-----+
Pilih: 2
Masukkan Nama: admin
Masukkan Password: 11234
User atau password salah! Sisa percobaan: 2
Masukkan Nama: ad123
Masukkan Password: admin12
User atau password salah! Sisa percobaan: 1
Masukkan Nama: admin
Masukkan Password: admin112
User atau password salah! Sisa percobaan: 0
Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 4.1  
Login Gagal

## B. Login Berhasil Admin

```
+-----+
|      APLIX      |
+-----+
| 1. Register Akun |
| 2. Login         |
| 3. Keluar        |
+-----+
Pilih: 2
Masukkan Nama: admin
Masukkan Password: admin123
Login berhasil! Selamat datang, admin!
+-----+
|      MENU      |
+-----+
| 1. Tambah Film  |
| 2. Lihat Film   |
| 3. Ubah Film    |
| 4. Hapus Film   |
| 5. Logout       |
+-----+
```

Gambar 4.2  
Login Berhasil Admin

### C. Register dan Login berhasil untuk user

```
+-----+
|      APLIX      |
+-----+
| 1. Register Akun |
| 2. Login         |
| 3. Keluar        |
+-----+
Pilih: 1
----- Registrasi Akun Baru -----
Masukkan Username: Muhammad Rizky Adha Putra
Masukkan Password: 2409106113
Registrasi berhasil! Silahkan login untuk melanjutkan.
+-----+
|      APLIX      |
+-----+
| 1. Register Akun |
| 2. Login         |
| 3. Keluar        |
+-----+
Pilih: 2
Masukkan Nama: Muhammad Rizky Adha Putra
Masukkan Password: 2409106113
Login berhasil! Selamat datang, Muhammad Rizky Adha Putra!
+-----+
|      MENU      |
+-----+
| 1. Lihat Film   |
| 2. Logout       |
+-----+
```

Gambar 4.3  
Register dan Login berhasil untu user

## 5. Langkah-Langkah Git pada VSCode

```
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git add .

Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git commit -m "Finish Posttest 4"
[main 69eec51] Finish Posttest 4
2 files changed, 218 insertions(+)
create mode 100644 Post-test/Post-test-4/2409106113-MuhammadRizkyAdhaPutra-PT-4.cpp
create mode 100644 Post-test/Post-test-4/2409106113-MuhammadRizkyAdhaPutra-PT-4.exe

Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git push -origin main
fatal: 'main' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 25.00 KiB | 2.78 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/113Rizky/praktikum-apl.git
aflae8e..69eec51 main -> main
```

1. **Git add .** Menambahkan semua perubahan file di direktori kerja ke staging area.
2. **Git commit -m “Finish Posttest 4”** Menyimpan perubahan yang sudah di-*add* ke dalam riwayat git dengan pesan commit.
3. **Git push** Mengirim commit dari repository lokal ke repository di GitHub (remote).