

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

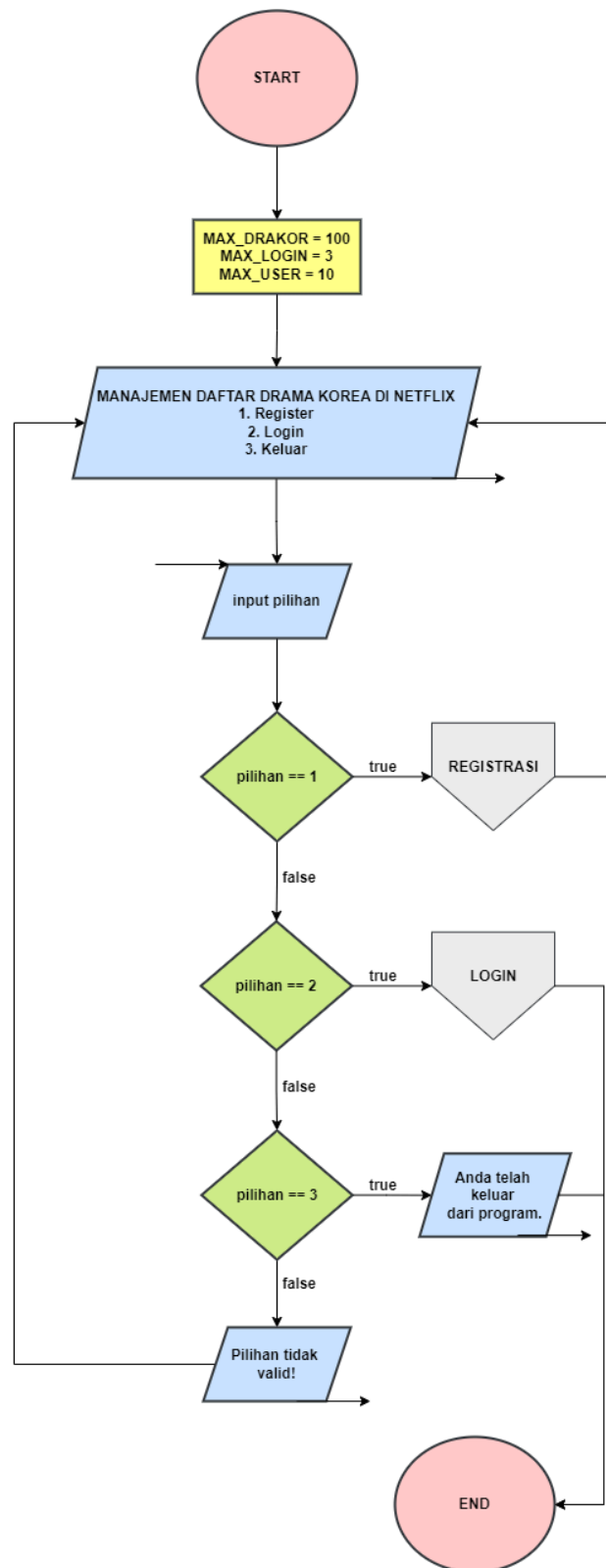


Disusun oleh:
Nabilah Alfa Rahmah (2409106004)
Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

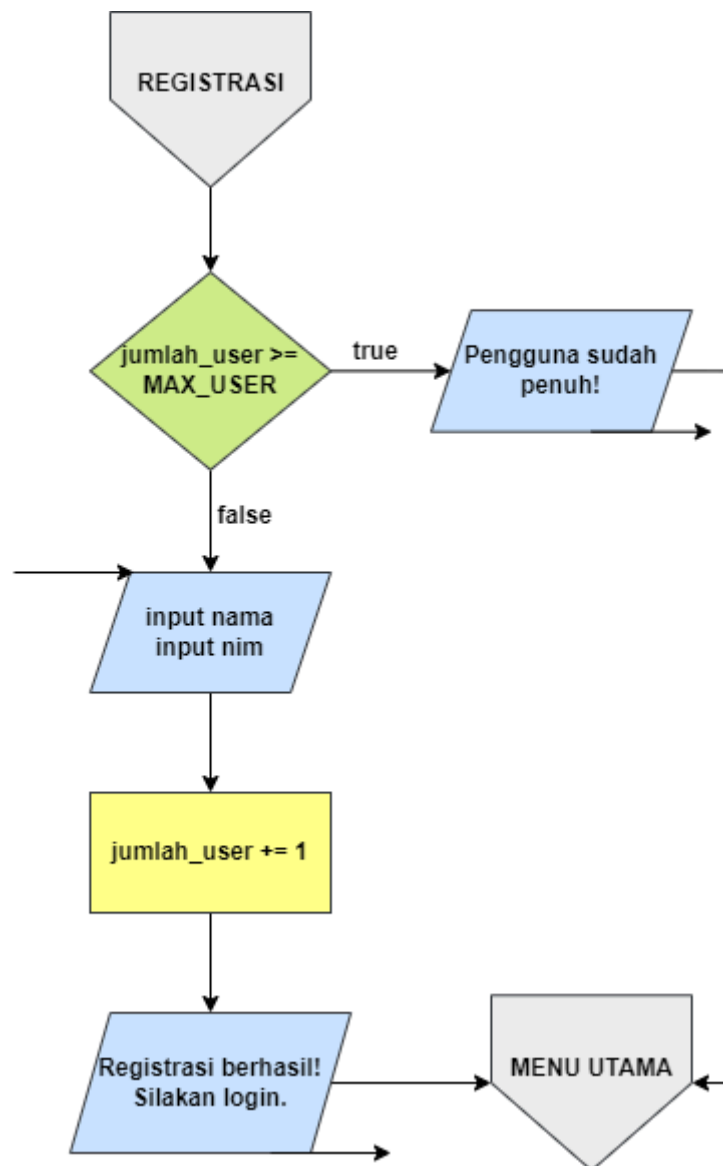
1. Flowchart

1.1 Menu Utama



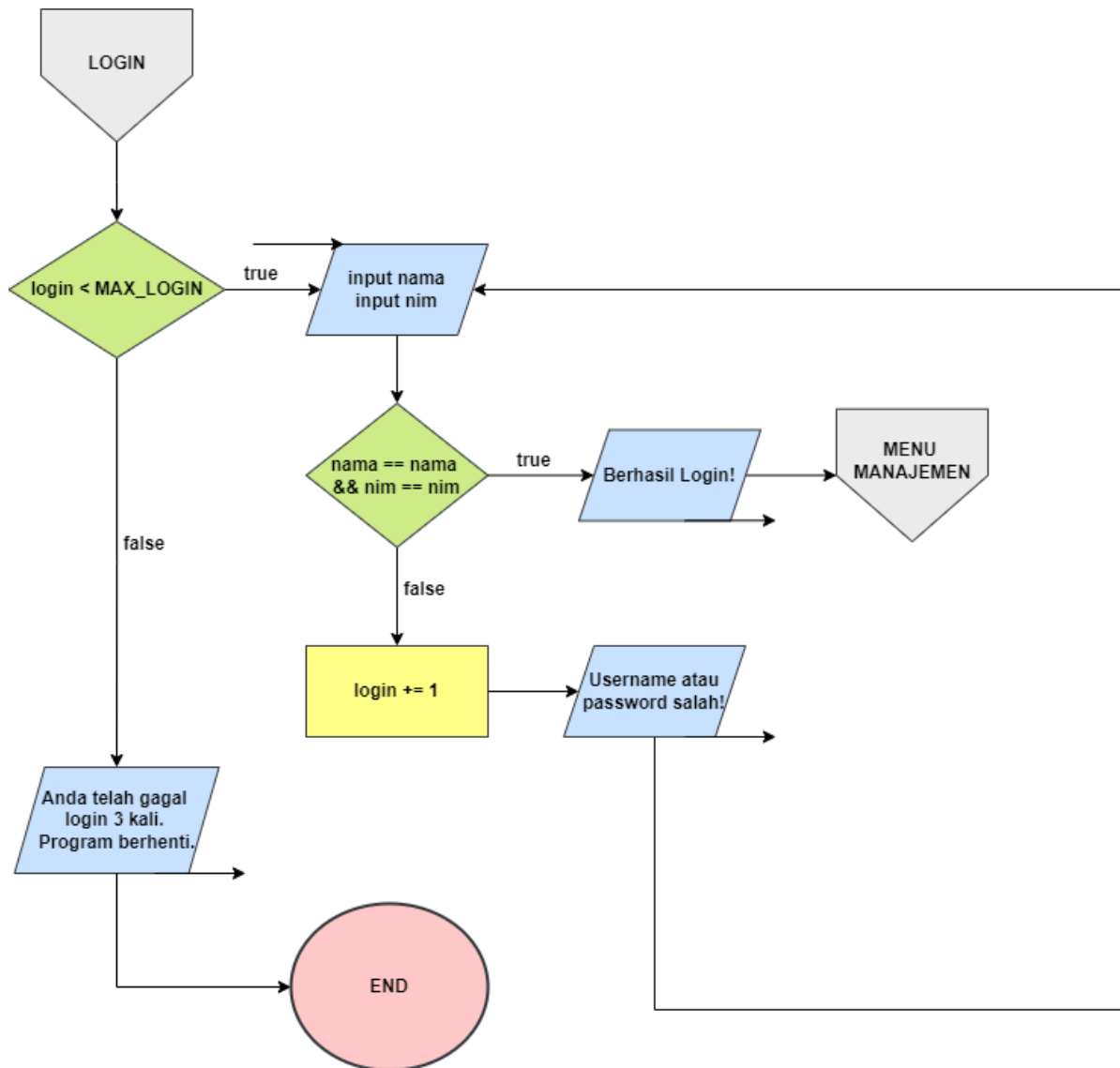
Gambar 1.1 Flowchart Menu Utama

1.2 Registrasi



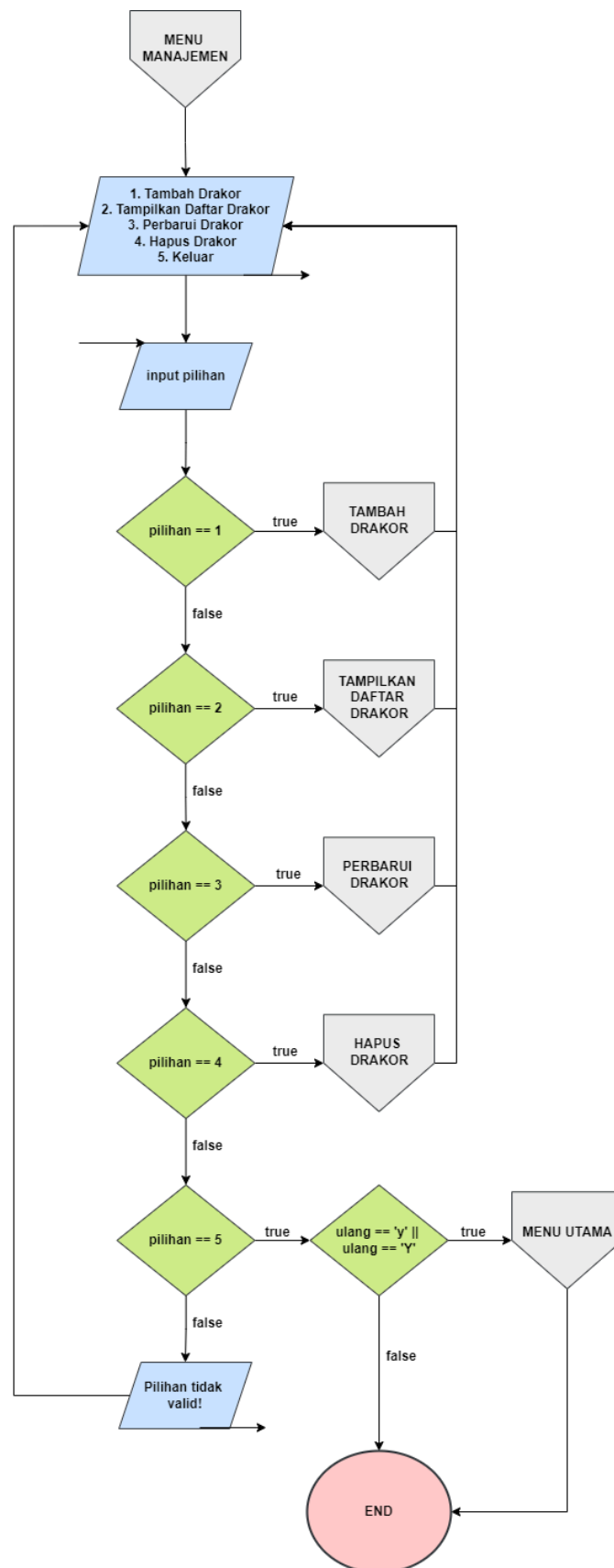
Gambar 1.2 Flowchart Registrasi

1.3 Login



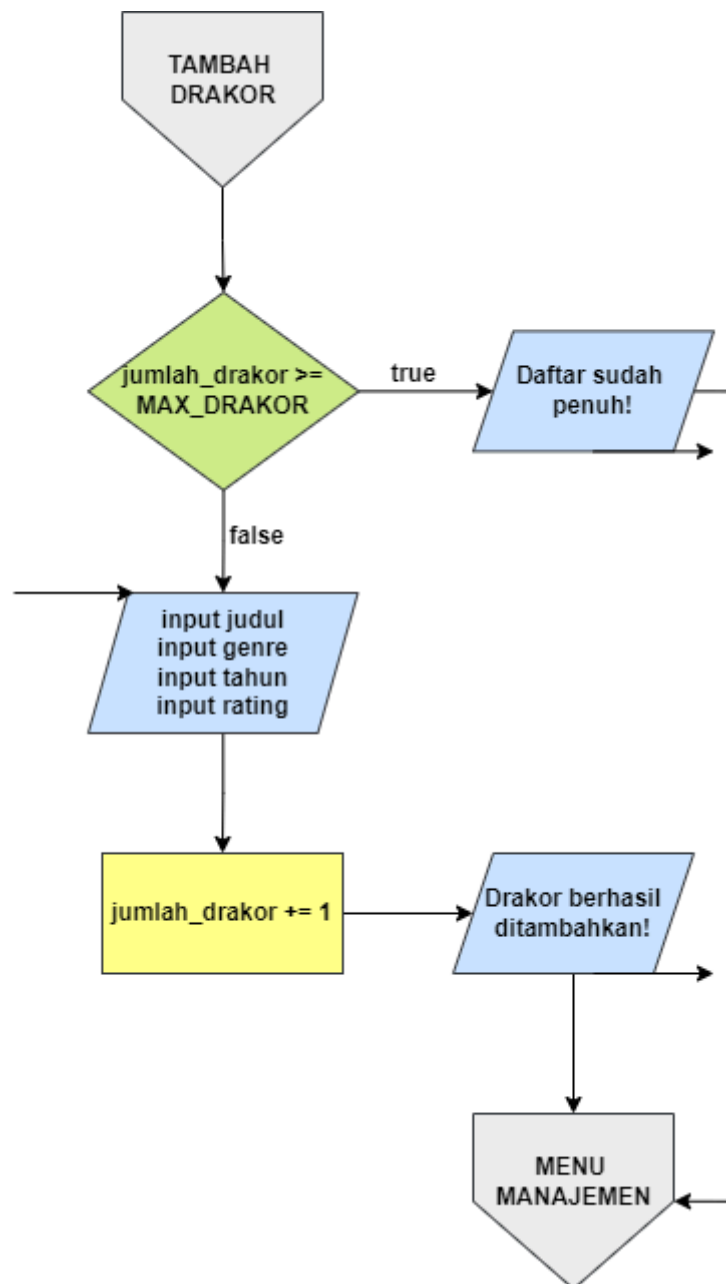
Gambar 1.3 Flowchart Login

1.4 Menu Manajemen



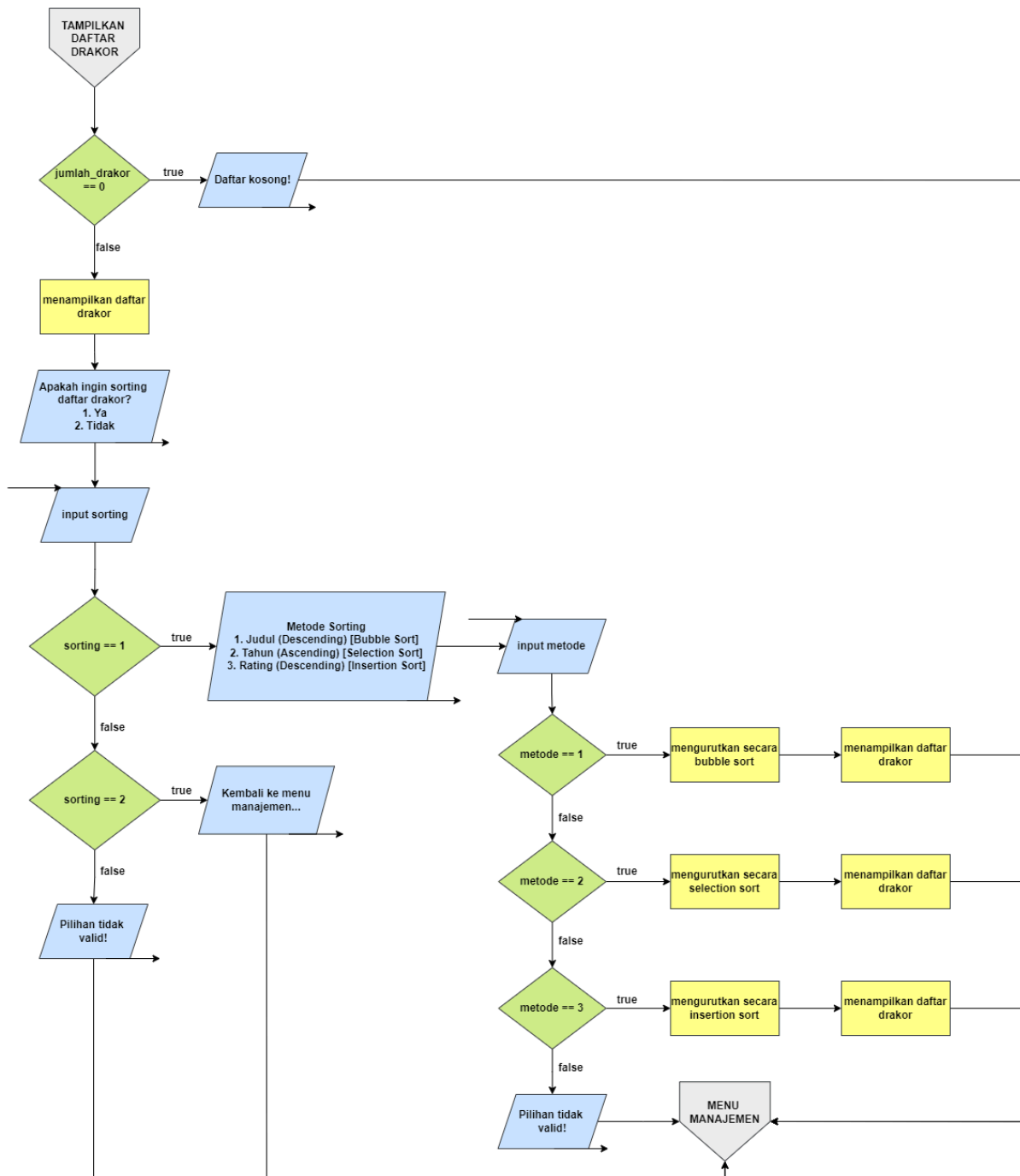
Gambar 1.4 Flowchart Menu Manajemen

1.5 Tambah Drakor



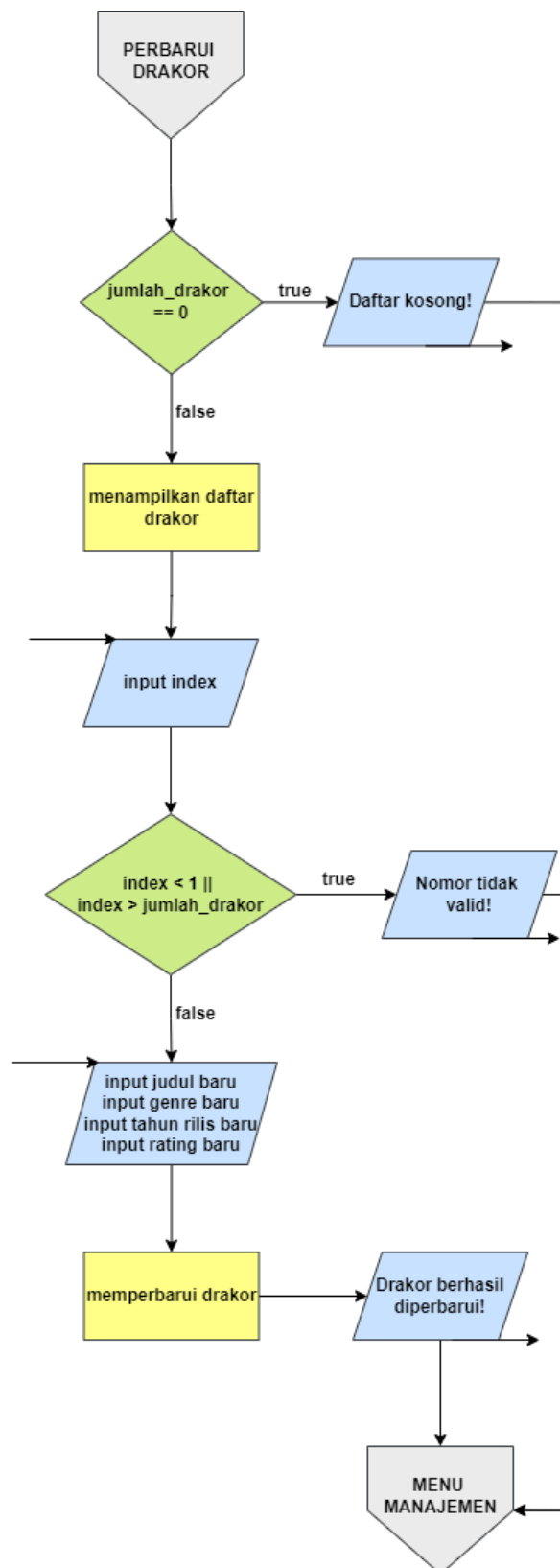
Gambar 1.5 Flowchart Tambah Drakor

1.6 Tampilkan Daftar Drakor



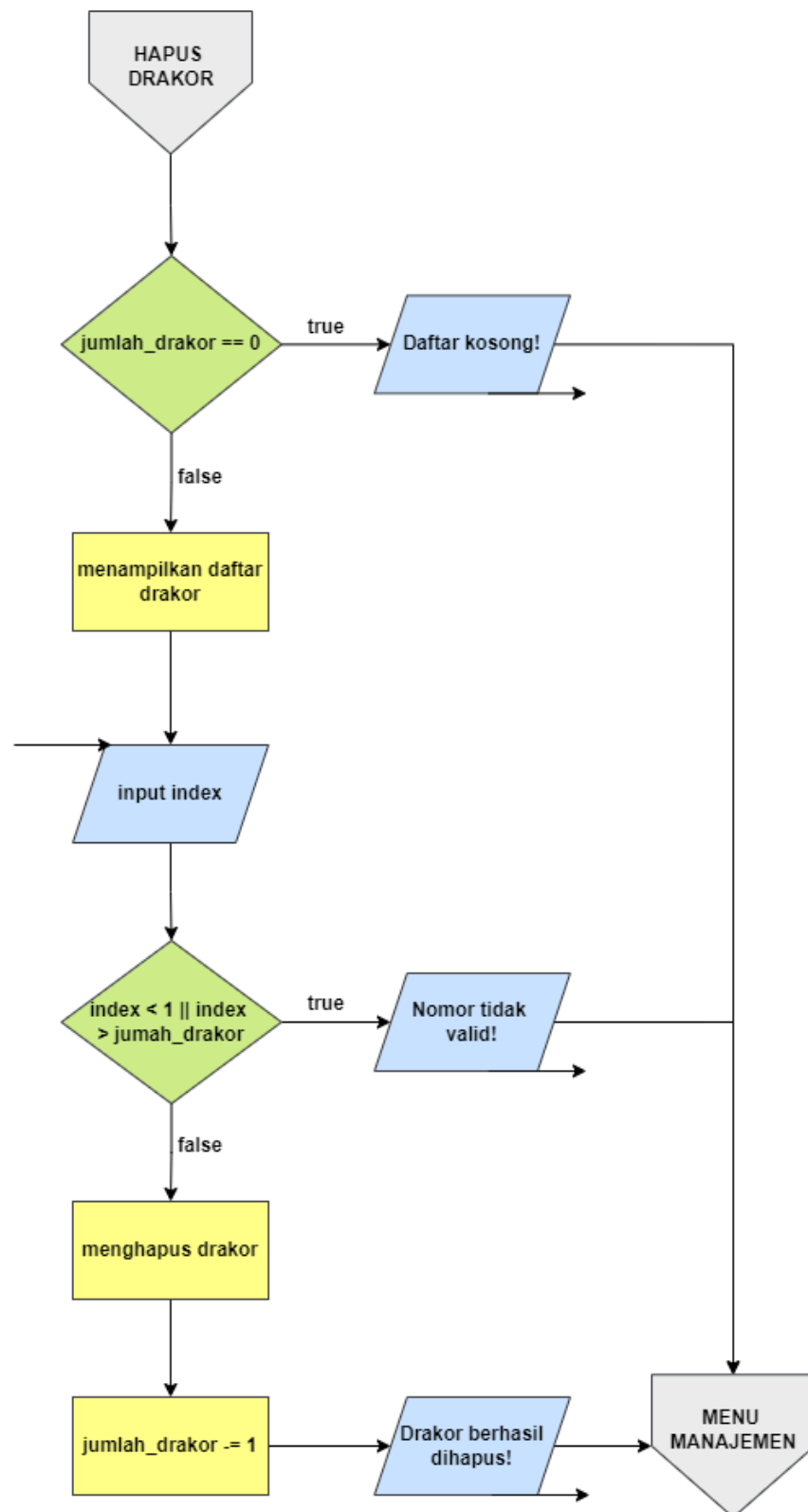
Gambar 1.6 Flowchart Tampilkan Daftar Drakor

1.7 Perbarui Drakor



Gambar 1.7 Flowchart Perbarui Drakor

1.8 Hapus Drakor



Gambar 1.8 Flowchart Hapus Drakor

2. Analisis Program

Program ini bertujuan untuk mengelola daftar drama Korea di Netflix dengan fitur utama berupa registrasi pengguna, login, serta manajemen drakor dengan CRUD (Create, Read, Update, dan Delete). Ketika program dijalankan, pengguna ditampilkan menu utama berisi registrasi jika ingin mendaftarkan akun, login jika ingin masuk ke sistem, dan keluar dari program. Di sistem manajemen, pengguna dapat memilih menu tambah, tampilkan, perbarui, hapus drakor, maupun keluar dari program. Pada pilihan keluar, pengguna bisa menentukan lagi ingin mengulang program atau tidak.

Secara keseluruhan, program ini dirancang untuk membantu pengguna dalam mengelola daftar drama Korea di Netflix dengan cara yang sistematis dan terstruktur.

3. Source Code

3.1 Fitur Registrasi

Fitur ini digunakan untuk melakukan registrasi akun baru dengan memasukkan nama lengkap sebagai username dan NIM sebagai password.

Source Code:

```
void registrasi(Data &dt) {
    if (dt.jumlah_user >= MAX_USER) {
        cout << "Pengguna sudah penuh!" << endl;
    } else {
        cout << "Masukkan nama lengkap sebagai username: ";
        getline(cin, dt.users[dt.jumlah_user].nama);
        cout << "Masukkan NIM sebagai password: ";
        getline(cin, dt.users[dt.jumlah_user].nim);
        dt.jumlah_user++;
        cout << "Registrasi berhasil! Silakan login.\n";
    }
}
```

3.2 Fitur Login

Fitur ini digunakan untuk melakukan login dengan memasukkan nama dan NIM sebagai username dan password maksimal 3 kali percobaan.

Source Code:

```
bool login_user(Data *dt, string &user_sekarang) {
    string nama, nim;
    int login = 0;
    while (login < MAX_LOGIN) {
        cout << "Username: ";
        getline(cin, nama);
        cout << "Password: ";
        getline(cin, nim);

        bool valid = false;
        for (int i = 0; i < dt->jumlah_user; i++) {
            if (dt->users[i].nama == nama && dt->users[i].nim == nim) {
                user_sekarang = nama;
                valid = true;
                break;
            }
        }
        if (valid) {
            cout << "Berhasil Login!" << endl;
        }
    }
}
```

```

        return true;
    } else {
        login++;
        cout << "Username atau password salah! Percobaan " << login << "
dari 3.\n";
    }
}

cout << "Anda telah aggal login 3 kali. Program berhenti.\n";
return false;
}

```

3.3 Menu Manajemen

Menu pilihan dengan fitur-fitur yang dapat dipilih sesuai keinginan pengguna.

Source Code:

```

cout << "\n===== \n";
cout << "          MENU MANAJEMEN          \n";
cout << "===== \n";
cout << "1. Tambah Drakor\n";
cout << "2. Tampilkan Daftar Drakor\n";
cout << "3. Perbarui Drakor\n";
cout << "4. Hapus Drakor\n";
cout << "5. Keluar\n";
cout << "----- \n";
cout << "Pilihan (1-5): ";
cin >> pilihan;

```

3.4 Fitur Tambah Drakor

Fitur ini menyimpan data drakor ke dalam array.

Source Code:

```

// Address-of
void tambah_drakor (Data &dt, string judul, string genre, int tahun, float
rating) {
    dt.daftar_drakor[dt.jumlah_drakor++] = {judul, genre, tahun, rating};
    cout << "Drakor berhasil ditambahkan!\n";
}

// Dereference
void tambah_drakor(Data *dt) {
    string judul, genre;
    int tahun;
    float rating;

    cout << "Masukkan Judul: ";

```

```

getline(cin, judul);
cout << "Masukkan Genre: ";
getline(cin, genre);
cout << "Masukkan Tahun Rilis: ";
cin >> tahun;
cout << "Masukkan Rating: ";
cin >> rating;
cin.ignore();

tambah_drakor(*dt, judul, genre, tahun, rating);
}

```

3.5 Fitur Tampilkan Daftar drakor

Fitur ini menampilkan data dalam bentuk tabel.

Source Code:

```

// Dereference
void tampilkan_drakor(Drakor *drakors, int index, int total) {
    if (index >= total) return;
    cout << left << setw(5) << index + 1
         << setw(30) << drakors[index].judul
         << setw(25) << drakors[index].genre
         << setw(10) << drakors[index].tahun
         << setw(10) << drakors[index].rating << endl;
    tampilkan_drakor(drakors, index + 1, total);
}

```

3.6 Fitur Bubble Sort

Fitur ini mengurutkan data (judul) secara descending.

Source Code:

```

// Bubble Sort untuk judul descending
void sort_bubble(Drakor drakors[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (drakors[j].judul < drakors[j + 1].judul) {
                Drakor temp = drakors[j];
                drakors[j] = drakors[j + 1];
                drakors[j + 1] = temp;
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan judul (descending)!\n";
}

```

3.7 Fitur Selection Sort

Fitur ini mengurutkan data (tahun rilis) secara ascending.

Source Code:

```
// Selection Sort untuk tahun ascending
void sort_selection(Drakor drakors[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int min_index = i;
        for (int j = i + 1; j < n; j++) {
            if (drakors[j].tahun < drakors[min_index].tahun) {
                min_index = j;
            }
        }
        if (min_index != i) {
            Drakor temp = drakors[i];
            drakors[i] = drakors[min_index];
            drakors[min_index] = temp;
        }
    }
    cout << "Data berhasil diurutkan berdasarkan tahun (ascending)!\n";
}
```

3.8 Fitur Insertion Sort

Fitur ini mengurutkan data (rating) secara descending.

Source Code:

```
// Insertion Sort untuk rating descending
void sort_insertion(Drakor drakors[], int n) {
    for (int i = 1; i < n; i++) {
        Drakor temp = drakors[i];
        int j = i - 1;
        while (j >= 0 && drakors[j].rating < temp.rating) {
            drakors[j + 1] = drakors[j];
            j--;
        }
        drakors[j + 1] = temp;
    }
    cout << "Data berhasil diurutkan berdasarkan rating (descending)!\n";
}
```

3.9 Fitur Perbarui Drakor

Fitur ini memperbarui data berdasarkan nomor urut.

Source Code:

```

void perbarui_drakor (Data &dt) {
    if (dt.jumlah_drakor == 0) {
        cout << "Daftar kosong!" << endl;
    } else {
        cout << fixed << setprecision(1);
        cout << "\nDaftar Drama Korea:\n";
        cout << left << setw(5) << "No"
            << setw(30) << "Judul"
            << setw(25) << "Genre"
            << setw(10) << "Tahun"
            << setw(10) << "Rating" << endl;
        cout << string(78, '-') << endl;
        tampilkan_drakor(dt.daftar_drakor, 0, dt.jumlah_drakor);

        int index;
        cout << "Masukkan nomor drakor yang ingin diperbarui: ";
        cin >> index;
        if (index < 1 || index > dt.jumlah_drakor) {
            cout << "Nomor tidak valid!" << endl;
        } else {
            cin.ignore();
            cout << "Masukkan Judul Baru: ";
            getline(cin, dt.daftar_drakor[index - 1].judul);
            cout << "Masukkan Genre Baru: ";
            getline(cin, dt.daftar_drakor[index - 1].genre);
            cout << "Masukkan Tahun Rilis Baru: ";
            cin >> dt.daftar_drakor[index - 1].tahun;
            cout << "Masukkan Rating Baru: ";
            cin >> dt.daftar_drakor[index - 1].rating;
            cin.ignore();
            cout << "Drakor berhasil diperbarui!\n";
        }
    }
}

```

3.10 Fitur Hapus Drakor

Fitur ini menghapus data berdasarkan nomor urut.

Source Code:

```

// Dereference
void hapus_drakor(Data *dt) {
    if (dt->jumlah_drakor == 0) {
        cout << "Daftar kosong!" << endl;
    } else {
        cout << fixed << setprecision(1);
        cout << "\nDaftar Drama Korea:\n";
        cout << left << setw(5) << "No"
            << setw(30) << "Judul"

```

```

        << setw(25) << "Genre"
        << setw(10) << "Tahun"
        << setw(10) << "Rating" << endl;
    cout << string(78, '-') << endl;
    tampilkan_drakor(dt->daftar_drakor, 0, dt->jumlah_drakor);

    int index;
    cout << "Masukkan nomor drakor yang ingin dihapus: ";
    cin >> index;
    cin.ignore();
    if (index < 1 || index > dt->jumlah_drakor) {
        cout << "Nomor tidak valid!" << endl;
    } else {
        for (int i = index - 1; i < dt->jumlah_drakor - 1; i++) {
            dt->daftar_drakor[i] = dt->daftar_drakor[i + 1];
        }
        dt->jumlah_drakor--;
        cout << "Drakor berhasil dihapus!\n";
    }
}
}

```

3.11 Fitur Keluar

Fitur ini program keluar dari loop while (true) dengan break dan kembali ke opsi ulang program.

Source Code:

```

        case 5:
            cout << "Keluar dari program...\n";
            break;

```

3.12 Opsi Ulang Program

Pengguna diberikan opsi untuk mengulang program atau keluar menggunakan perulangan do-while.

Source Code:

```

    do {
        cout << "Apakah Anda ingin mengulang program? (y/n): ";
        cin >> ulang;
        cin.ignore();
        if (ulang != 'y' && ulang != 'Y' && ulang != 'n' && ulang != 'N') {
            cout << "Input tidak valid! Masukkan 'y' untuk ya atau 'n' untuk tidak.\n";
        }
    } while (ulang != 'y' && ulang != 'Y' && ulang != 'n' && ulang != 'N');

```


4. Uji Coba dan Hasil Output

```
=====
|| MANAJEMEN DAFTAR DRAMA KOREA DI NETFLIX ||
=====

=====
                MENU UTAMA
=====
1. Register
2. Login
3. Keluar
Pilih (1-3): 1
Masukkan nama lengkap sebagai username: Nabilah Alfa Rahmah
Masukkan NIM sebagai password: 2409106004
Registrasi berhasil! Silakan login.

=====
                MENU UTAMA
=====
1. Register
2. Login
3. Keluar
Pilih (1-3): 2
Username: Nabilah Alfa Rahmah
Password: 2409106004
Berhasil Login!

=====
                MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
```

Gambar 4.1 Register dan Login

```

-----
Pilihan (1-5): 1
Masukkan Judul: Study Group
Masukkan Genre: Laga, Komedi
Masukkan Tahun Rilis: 2025
Masukkan Rating: 9
Drakor berhasil ditambahkan!

=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 1
Masukkan Judul: Squid Game 2
Masukkan Genre: Drama, Thriller, Horror
Masukkan Tahun Rilis: 2024
Masukkan Rating: 8.2
Drakor berhasil ditambahkan!

=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 1
Masukkan Judul: Crash Landing on You
Masukkan Genre: Romance, Drama

```

Gambar 4.2 Tambah Drakor

```

Masukkan Tahun Rilis: 2019
Masukkan Rating: 9
Drakor berhasil ditambahkan!

=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): █

```

Gambar 4.2 Tambah Drakor

```

=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 2

Daftar Drama Korea:

```

No	Judul	Genre	Tahun	Rating
1	Study Group	Laga, Komedi	2025	9.0
2	Squid Game 2	Drama, Thriller, Horror	2024	8.2
3	Crash Landing on You	Romance, Drama	2019	9.0

Gambar 4.3 Tampilkan Daftar Drakor

```

Apakah ingin sorting daftar drakor?
1. Ya
2. Tidak
Pilihan (1/2): 1

=====
METODE SORTING
=====
1. Judul (Descending) [Bubble Sort]
2. Tahun (Ascending) [Selection Sort]
3. Rating (Descending) [Insertion Sort]
-----
Pilihan (1-3): 1
Data berhasil diurutkan berdasarkan judul (descending)!

== Daftar Drakor Setelah Sorting ==

```

No	Judul	Genre	Tahun	Rating
1	Study Group	Laga, Komedi	2025	9.0
2	Squid Game	Drama, Thriller, Horror	2024	8.2
3	Crash Landing on You	Romance, Drama	2019	9.0

Gambar 4.4 Sorting Drakor (Judul)

```
Apakah ingin sorting daftar drakor?
1. Ya
2. Tidak
Pilihan (1/2): 1

=====
METODE SORTING
=====
1. Judul (Descending) [Bubble Sort]
2. Tahun (Ascending) [Selection Sort]
3. Rating (Descending) [Insertion Sort]
-----
Pilihan (1-3): 2
Data berhasil diurutkan berdasarkan tahun (ascending)!

== Daftar Drakor Setelah Sorting ==
No    Judul                Genre                Tahun    Rating
-----
1     Crash Landing on You    Romance, Drama       2019     9.0
2     Squid Game              Drama, Thriller, Horror 2024     8.2
3     Study Group             Laga, Komedi         2025     9.0
```

Gambar 4.5 Sorting Drakor (Tahun Rilis)

```
Apakah ingin sorting daftar drakor?
1. Ya
2. Tidak
Pilihan (1/2): 1

=====
METODE SORTING
=====
1. Judul (Descending) [Bubble Sort]
2. Tahun (Ascending) [Selection Sort]
3. Rating (Descending) [Insertion Sort]
-----
Pilihan (1-3): 3
Data berhasil diurutkan berdasarkan rating (descending)!

== Daftar Drakor Setelah Sorting ==
No    Judul                Genre                Tahun    Rating
-----
1     Crash Landing on You    Romance, Drama       2019     9.0
2     Study Group             Laga, Komedi         2025     9.0
3     Squid Game              Drama, Thriller, Horror 2024     8.2
```

Gambar 4.6 Sorting Drakor (Rating)

```
=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 3

Daftar Drama Korea:
No    Judul                      Genre                      Tahun    Rating
-----
1     Study Group                 Laga, Komedi              2025     9.0
2     Squid Game 2                 Drama, Thriller, Horror   2024     8.2
3     Crash Landing on You         Romance, Drama            2019     9.0
Masukkan nomor drakor yang ingin diperbarui: 2
Masukkan Judul Baru: Squid Game 2
Masukkan Genre Baru: Drama, Thriller, Horror
Masukkan Tahun Rilis Baru: 2024
Masukkan Rating Baru: 8.7
Drakor berhasil diperbarui!
```

Gambar 4.7 Perbarui Drakor

```
=====
MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 4

Daftar Drama Korea:
No    Judul                      Genre                      Tahun    Rating
-----
1     Study Group                 Laga, Komedi              2025     9.0
2     Squid Game 2                 Drama, Thriller, Horror   2024     8.7
3     Crash Landing on You         Romance, Drama            2019     9.0
Masukkan nomor drakor yang ingin dihapus: 3
Drakor berhasil dihapus!
```

Gambar 4.8 Hapus Drakor

```
=====
                        MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 5
Keluar dari program...
Apakah Anda ingin mengulang program? (y/n): y
=====
|| MANAJEMEN DAFTAR DRAMA KOREA DI NETFLIX ||
=====

=====
                        MENU UTAMA
=====
1. Register
2. Login
3. Keluar
Pilih (1-3): █
```

Gambar 4.9 Keluar dan Ulang Program

```
=====
                        MENU MANAJEMEN
=====
1. Tambah Drakor
2. Tampilkan Daftar Drakor
3. Perbarui Drakor
4. Hapus Drakor
5. Keluar
-----
Pilihan (1-5): 5
Keluar dari program...
Apakah Anda ingin mengulang program? (y/n): n
=====
                        TERIMA KASIH! BYE!
=====
PS D:\alfa\praktikum-apl\post-test\post-test-apl-3> █
```

Gambar 4.10 Keluar dan Tidak Ulang Program

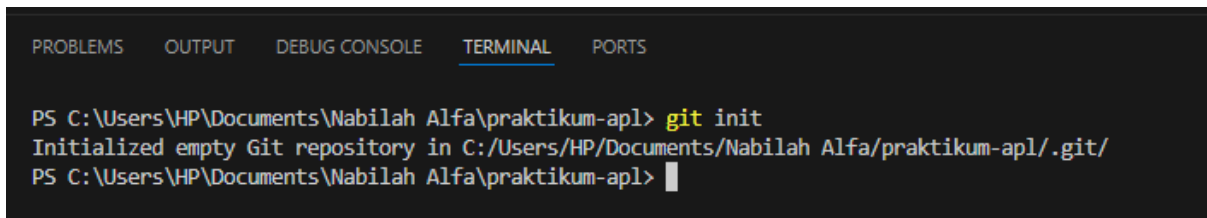
```
=====
                        MENU UTAMA
=====
1. Register
2. Login
3. Keluar
Pilih (1-3): 2
Username: nabila
Password: 4
Username atau password salah! Percobaan 1 dari 3.
Username: bil
Password: 2
Username atau password salah! Percobaan 2 dari 3.
Username: nab
Password: 3
Username atau password salah! Percobaan 3 dari 3.
Anda telah gagal login3 kali. Program berhenti.
PS D:\alfa\praktikum-apl\post-test\post-test-apl-3>
```

Gambar 4.11 Login Gagal 1-3 Kali

5. Langkah-Langkah Git

5.1 git init

Digunakan untuk menginisialisasi repository Git di dalam folder proyek. Setelah perintah ini dilakukan, folder .git akan dibuat di dalam direktori proyek.



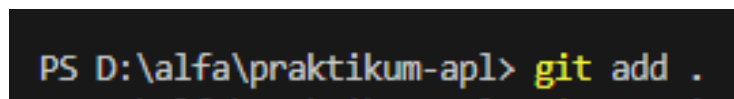
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\HP\Documents\Nabilah Alfa\praktikum-apl> git init
Initialized empty Git repository in C:/Users/HP/Documents/Nabilah Alfa/praktikum-apl/.git/
PS C:\Users\HP\Documents\Nabilah Alfa\praktikum-apl> 
```

Gambar 5.1 Git Init

5.2 git add

Digunakan untuk menambahkan perubahan ke dalam staging area sebelum commit, dapat digunakan untuk menambahkan file tertentu maupun semua file dalam proyek.

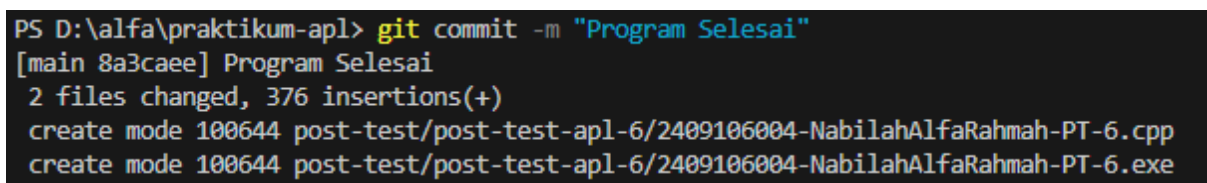


```
PS D:\alfa\praktikum-apl> git add .
```

Gambar 5.2 Git Add

5.3 git commit -m “Pesan Commit”

Digunakan untuk menyimpan perubahan di dalam repository Git dengan pesan deskriptif.



```
PS D:\alfa\praktikum-apl> git commit -m "Program Selesai"
[main 8a3caee] Program Selesai
2 files changed, 376 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106004-NabilahAlfaRahmah-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106004-NabilahAlfaRahmah-PT-6.exe
```

Gambar 5.3 Git Commit

5.4 git remote add origin link repo git

Digunakan untuk menghubungkan repository lokal dengan repository di GitHub.


```
PS C:\Users\HP\Documents\Wabilah Alfa\praktikum-apl> git remote add origin https://github.com/nblalfaaar/praktikum-apl.git
PS C:\Users\HP\Documents\Wabilah Alfa\praktikum-apl>
```

Gambar 5.4 Git Remote

5.5 git push -u origin main

Digunakan untuk mengunggah perubahan dari repository lokal ke repository di GitHub. -u origin main berarti mengatur branch main sebagai default untuk push ke origin.

```
PS D:\alfa\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 681.87 KiB | 5.13 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/nblalfaaar/praktikum-apl.git
    87d5b57..8a3caee  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\alfa\praktikum-apl>
```

Gambar 5.5 Git Push