

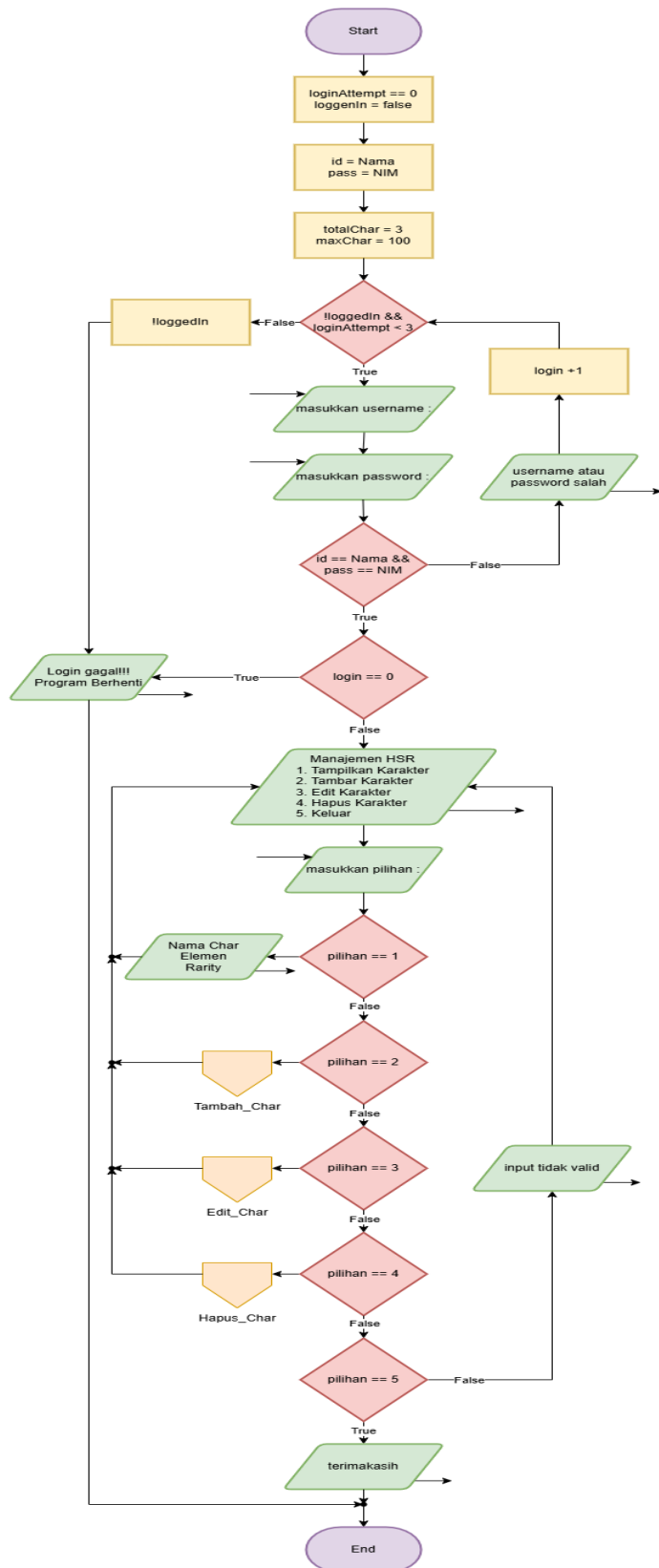
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT

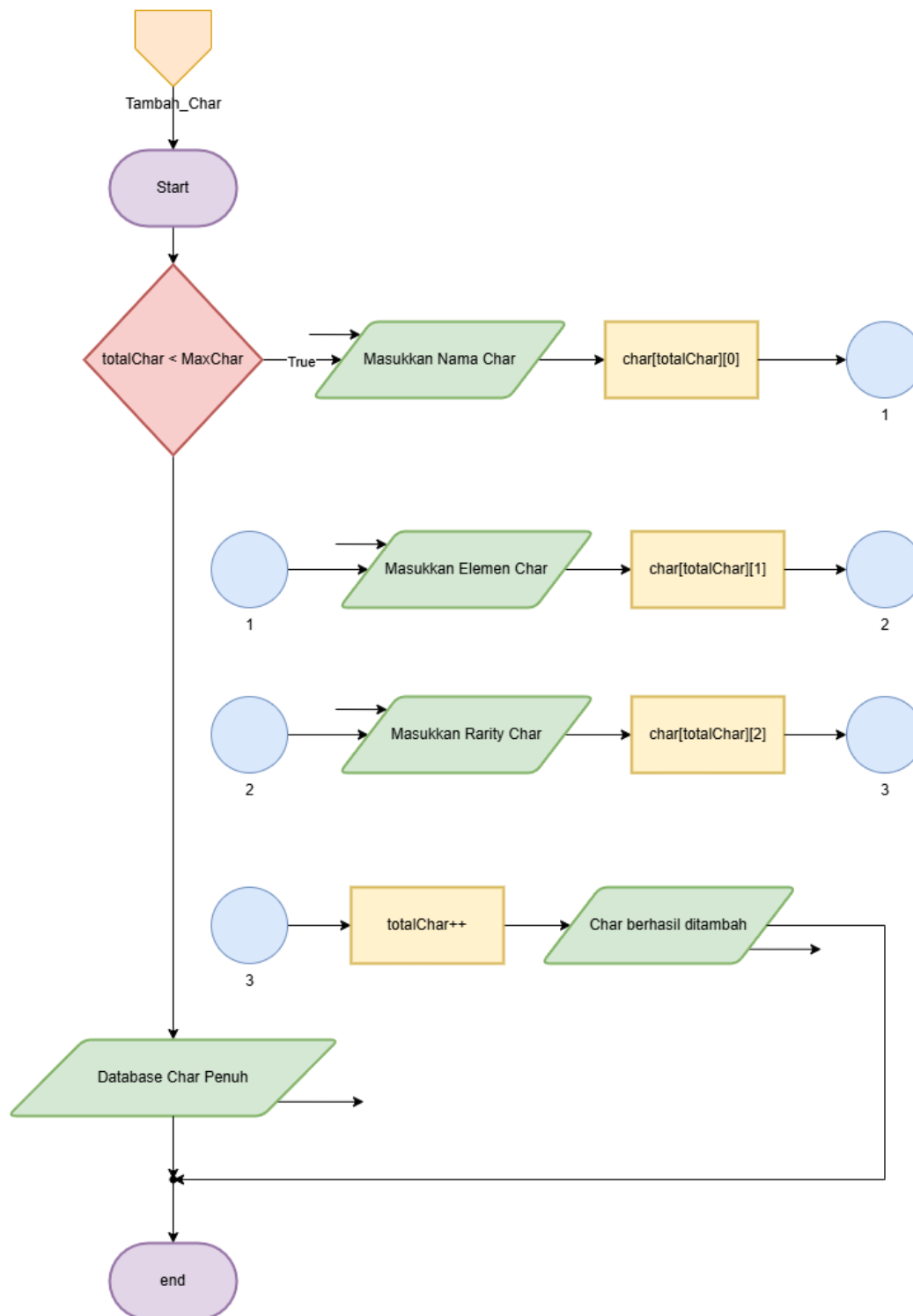


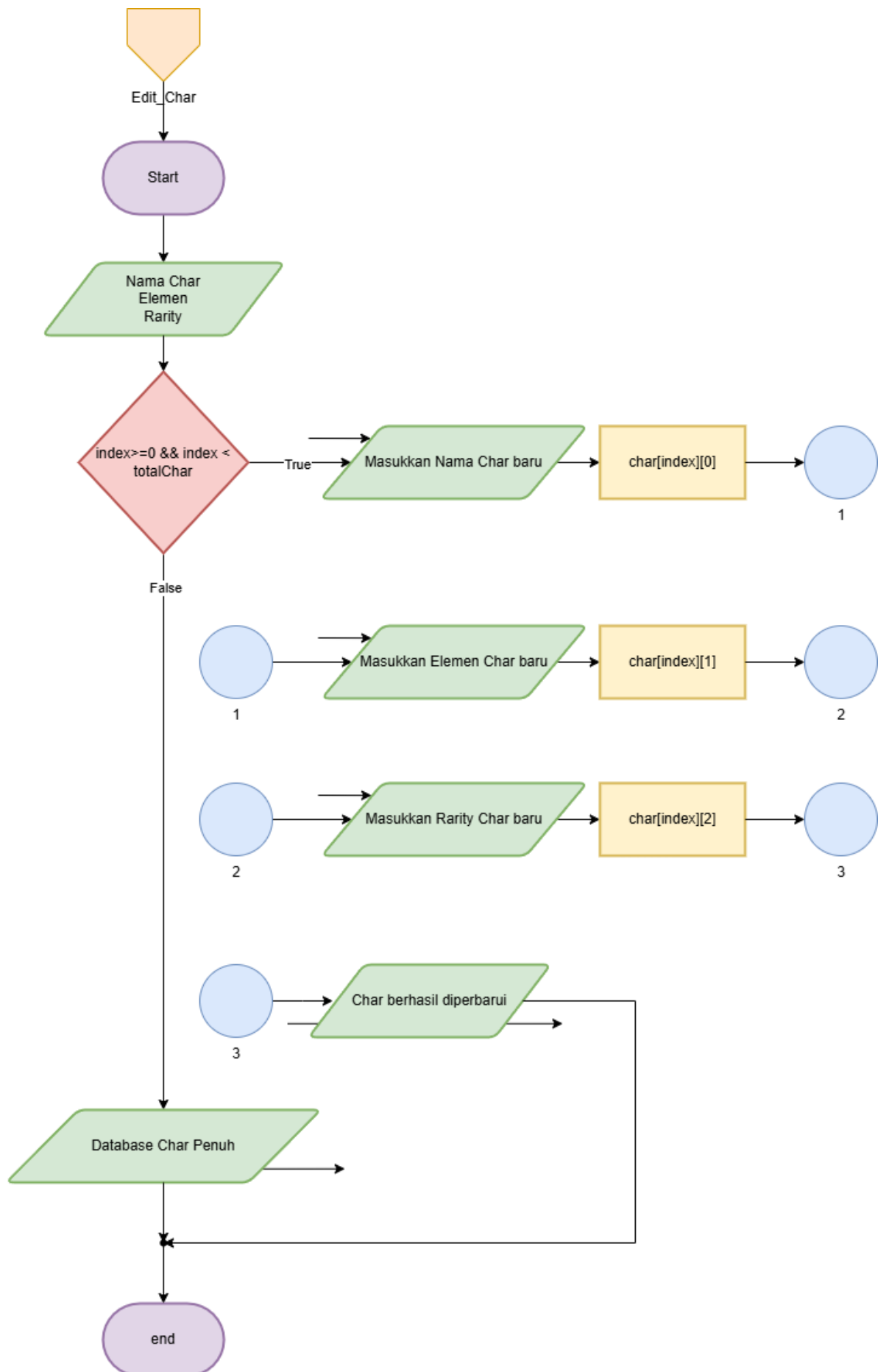
Disusun oleh:
Melchizedek Juliro Salomo Simangunsong (2409106117)
Kelas (C2 '24)

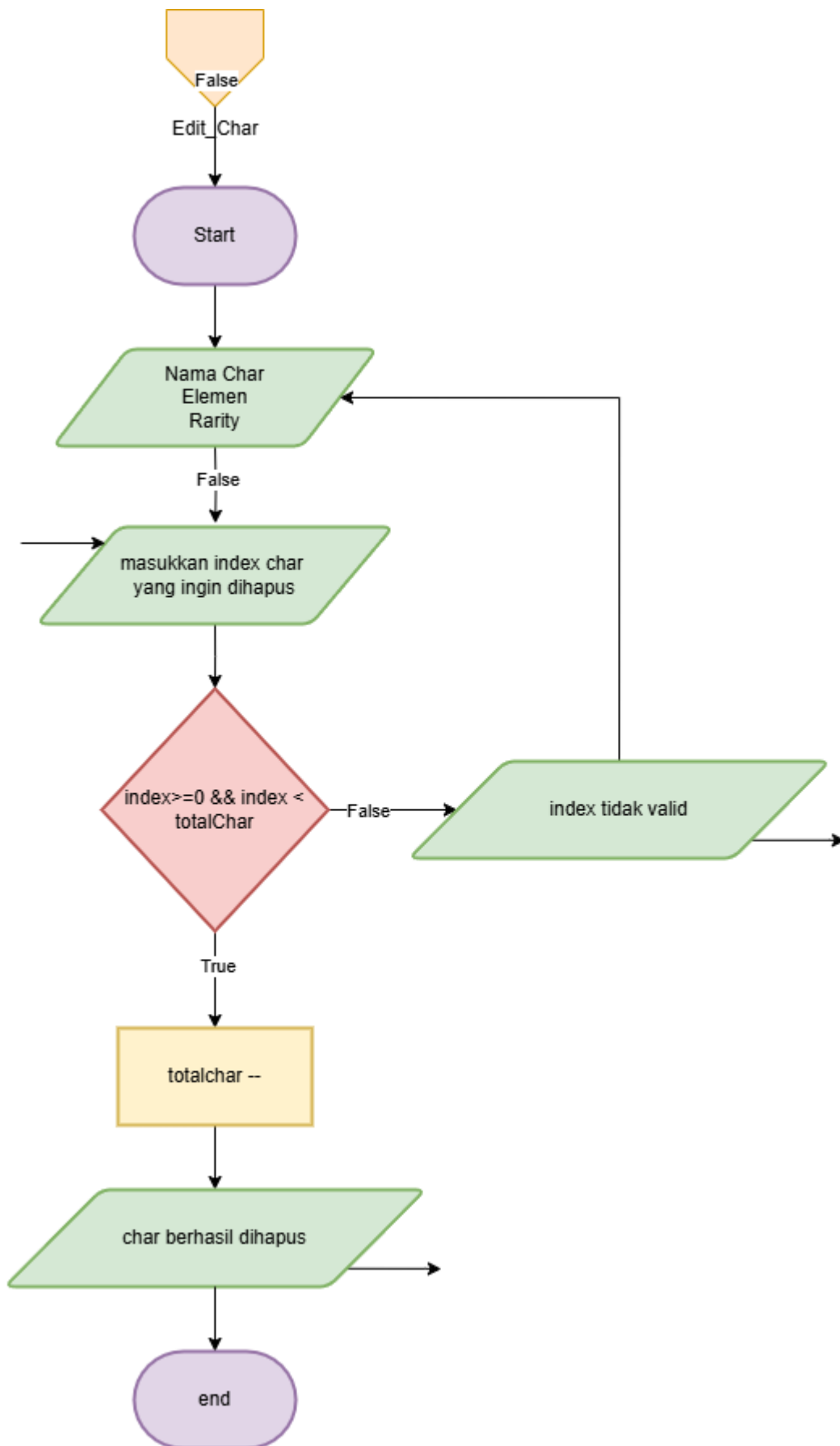
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart









2. Analisis Program

2.1 Deskripsi Singkat Program

Fitur utama program:

1. **Login Sistem** – Pengguna harus memasukkan username dan password yang benar untuk mengakses program.
2. **Tampilkan Karakter** – Menampilkan daftar karakter beserta elemen dan rarity-nya.
3. **Tambah Karakter** – Memungkinkan pengguna menambahkan karakter baru ke daftar.
4. **Edit Karakter** – Mengubah informasi karakter yang sudah ada berdasarkan indeks.
5. **Hapus Karakter** – Menghapus karakter dari daftar berdasarkan indeks.
6. **Keluar** – Menutup program setelah pengguna selesai menggunakan.

2.2 Penjelasan Alur & Algoritma

1. Library yang Digunakan

```
#include <iostream>
#include <string>
#include <iomanip>
```

- **#include <iostream>** → Digunakan untuk operasi input dan output (cin, cout).
- **#include <string>** → Digunakan agar bisa menggunakan tipe data string.
- **#include <iomanip>** → Digunakan untuk format output, seperti setw() untuk mengatur lebar kolom saat mencetak data

2. Deklarasi Variabel dan Array

```
int main() {  
    const int MAX_CHARACTERS = 100;  
    string characters[MAX_CHARACTERS][3] = {  
        {"Silver Wolf", "Quantum", "5"},  
        {"Kafka", "Lightning", "5"},  
        {"Dan Heng", "Wind", "4"}  
    };  
    int totalCharacters = 3;
```

- **MAX_CHARACTERS** → Konstanta yang menentukan jumlah maksimal karakter yang bisa disimpan.
- **characters[MAX_CHARACTERS][3]** → Array 2 dimensi untuk menyimpan daftar karakter.
 - **Kolom 0:** Nama karakter
 - **Kolom 1:** Elemen karakter
 - **Kolom 2:** Rarity karakter
- **totalCharacters** → Menyimpan jumlah karakter yang saat ini ada dalam daftar

3. Sistem Login

```
string username, password;  
int loginAttempts = 0;  
bool loggedIn = false;
```

- Variabel untuk menyimpan **username**, **password**, jumlah percobaan login, dan status login

```
while (true) {  
    loginAttempts = 0;  
    loggedIn = false;
```

- Loop utama yang akan **mengulang program setelah logout**.

```
if (!loggedIn) {  
    cout << "Anda telah gagal login 3 kali. Program berhenti.\n";  
    return 0;  
}
```

- Jika login gagal **3 kali berturut-turut**, program akan keluar (return 0).

4. Menu Utama

```
int pilihan;  
do {  
    cout << "\n==== Sistem Manajemen Karakter Honkai: Star Rail  
====\n";  
    cout << "1. Tampilkan Karakter\n";  
    cout << "2. Tambah Karakter\n";  
    cout << "3. Edit Karakter\n";  
    cout << "4. Hapus Karakter\n";  
    cout << "5. Keluar\n";  
    cout << "Pilihan: ";  
    cin >> pilihan;  
    cin.clear();  
    cin.ignore();  
}
```

- Menampilkan menu utama.
- **Pengguna memilih fitur dengan angka 1-5.**

5. Menampilkan Daftar Karakter

```
if (pilihan == 1) {
    cout << "\nDaftar Karakter:\n";
    cout << left << setw(8) << "Indeks" << setw(20) << "Nama" <<
    setw(15) << "Elemen" << setw(10) << "Rarity" << endl;
    cout << "-----\n";
    for (int i = 0; i < totalCharacters; i++) {
        cout << left << setw(8) << (i + 1) << setw(20) <<
        characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
        characters[i][2] << endl;
    }
}
```

- Menampilkan daftar karakter dalam **bentuk tabel** dengan format rapi menggunakan setw().

6. Menambah Karakter Baru

```
else if (pilihan == 2) {
    if (totalCharacters < MAX_CHARACTERS) {
        cout << "Masukkan Nama Karakter: ";
        getline(cin, characters[totalCharacters][0]);
        cout << "Masukkan Elemen Karakter: ";
        getline(cin, characters[totalCharacters][1]);
        cout << "Masukkan Rarity Karakter: ";
        getline(cin, characters[totalCharacters][2]);
        totalCharacters++;
        cout << "Karakter berhasil ditambahkan!\n";
    } else {
        cout << "Database penuh! Tidak bisa menambahkan karakter
        lagi.\n";
    }
}
```

- Jika jumlah karakter belum mencapai **batas maksimum**, pengguna dapat **memasukkan nama, elemen, dan rarity karakter**.
- Setelah input selesai, jumlah karakter bertambah (totalCharacters++).

7. Mengedit Karakter

```
else if (pilihan == 3) {
    int index;
    for (int i = 0; i < totalCharacters; i++) {
        cout << left << setw(8) << (i + 1) << setw(20) <<
characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
characters[i][2] << endl;
    }
    cout << "Masukkan indeks karakter yang ingin diedit (1 - "
<< totalCharacters << "): ";
    cin >> index;
    cin.ignore();
    index--;
    if (index >= 0 && index < totalCharacters) {
        cout << "Masukkan Nama Baru: ";
        getline(cin, characters[index][0]);
        cout << "Masukkan Elemen Baru: ";
        getline(cin, characters[index][1]);
        cout << "Masukkan Rarity Baru: ";
        getline(cin, characters[index][2]);
        cout << "Karakter berhasil diperbarui!\n";
    } else {
        cout << "Indeks tidak valid!\n";
    }
}
```

- Menampilkan daftar karakter.
- Meminta pengguna memilih karakter **berdasarkan indeks** untuk diedit.
- Jika indeks valid, pengguna dapat **mengubah nama, elemen, dan rarity** karakter.

8. Menghapus Karakter

```
        else if (pilihan == 4) {
            int index;
            for (int i = 0; i < totalCharacters; i++) {
                cout << left << setw(8) << (i + 1) << setw(20) <<
characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
characters[i][2] << endl;
            }
            cout << "Masukkan indeks karakter yang ingin dihapus (1 - "
<< totalCharacters << "): ";
            cin >> index;
            index--;
            if (index >= 0 && index < totalCharacters) {
                for (int i = index; i < totalCharacters - 1; i++) {
                    characters[i][0] = characters[i + 1][0];
                    characters[i][1] = characters[i + 1][1];
                    characters[i][2] = characters[i + 1][2];
                }
                totalCharacters--;
                cout << "Karakter berhasil dihapus!\n";
            } else {
                cout << "Indeks tidak valid!\n";
            }
        }
    }
```

- Meminta indeks karakter untuk dihapus.
- Karakter yang dipilih akan dihapus **dan karakter lainnya digeser ke atas**.

9. Keluar dari Program

```
        else if (pilihan == 5) {
            cout << "Terima kasih sudah mau memakai program gw next main
Honkai Star Rail aja!\n";
            return 0;
        }
    }
```

3. Source Code

```
#include <iostream>
#include <string>
#include <iomanip>

using namespace std;

int main() {
    const int MAX_CHARACTERS = 100;
    string characters[MAX_CHARACTERS][3] = {
        {"Silver Wolf", "Quantum", "5"},
        {"Kafka", "Lightning", "5"},
        {"Dan Heng", "Wind", "4"}
    };
    int totalCharacters = 3;
    string username, password;
    int loginAttempts = 0;
    bool loggedIn = false;

    while (true) {
        loginAttempts = 0;
        loggedIn = false;

        while (!loggedIn && loginAttempts < 3) {
            cout << "Masukkan Username: ";
            getline(cin, username);
            cout << "Masukkan Password: ";
            getline(cin, password);

            if (username == "Melchizedek" && password == "2409106117") {
                cout << "Berhasil login!\n";
                loggedIn = true;
            } else {
                cout << "Login gagal! Username atau password salah.\n";
                loginAttempts++;
            }
        }

        if (!loggedIn) {
            cout << "Anda telah gagal login 3 kali. Program berhenti.\n";
            return 0;
        }

        int pilihan;
        do {
```

```

cout << "\n===== Sistem Manajemen Karakter Honkai: Star Rail
=====\\n";

cout << "1. Tampilkan Karakter\\n";
cout << "2. Tambah Karakter\\n";
cout << "3. Edit Karakter\\n";
cout << "4. Hapus Karakter\\n";
cout << "5. Keluar\\n";
cout << "Pilihan: ";
cin >> pilihan;
cin.clear();
cin.ignore();

if (pilihan == 1) {
    cout << "\\nDaftar Karakter:\\n";
    cout << left << setw(8) << "Indeks" << setw(20) << "Nama" <<
setw(15) << "Elemen" << setw(10) << "Rarity" << endl;
    cout << "-----\\n";

    for (int i = 0; i < totalCharacters; i++) {
        cout << left << setw(8) << (i + 1) << setw(20) <<
characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
characters[i][2] << endl;
    }
}
else if (pilihan == 2) {
    if (totalCharacters < MAX_CHARACTERS) {
        cout << "Masukkan Nama Karakter: ";
        getline(cin, characters[totalCharacters][0]);
        cout << "Masukkan Elemen Karakter: ";
        getline(cin, characters[totalCharacters][1]);
        cout << "Masukkan Rarity Karakter: ";
        getline(cin, characters[totalCharacters][2]);
        totalCharacters++;
        cout << "Karakter berhasil ditambahkan!\\n";
    } else {
        cout << "Database penuh! Tidak bisa menambahkan karakter
lagi.\\n";
    }
}
else if (pilihan == 3) {
    int index;
    cout << "\\nDaftar Karakter:\\n";
    cout << left << setw(8) << "Indeks" << setw(20) << "Nama" <<
setw(15) << "Elemen" << setw(10) << "Rarity" << endl;
    cout << "-----\\n";

    for (int i = 0; i < totalCharacters; i++) {

```

```

        cout << left << setw(8) << (i + 1) << setw(20) <<
characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
characters[i][2] << endl;
    }
    cout << "Masukkan indeks karakter yang ingin diedit (1 - "
<< totalCharacters << "): ";
    cin >> index;
    cin.ignore();
    index--;
    if (index >= 0 && index < totalCharacters) {
        cout << "Masukkan Nama Baru: ";
        getline(cin, characters[index][0]);
        cout << "Masukkan Elemen Baru: ";
        getline(cin, characters[index][1]);
        cout << "Masukkan Rarity Baru: ";
        getline(cin, characters[index][2]);
        cout << "Karakter berhasil diperbarui!\n";
    } else {
        cout << "Indeks tidak valid!\n";
    }
}
else if (pilihan == 4) {
    int index;
    cout << "\nDaftar Karakter:\n";
    cout << left << setw(8) << "Indeks" << setw(20) << "Nama" <<
setw(15) << "Elemen" << setw(10) << "Rarity" << endl;
    cout << "-----\n";

    for (int i = 0; i < totalCharacters; i++) {
        cout << left << setw(8) << (i + 1) << setw(20) <<
characters[i][0] << setw(15) << characters[i][1] << setw(10) <<
characters[i][2] << endl;
    }
    cout << "Masukkan indeks karakter yang ingin dihapus (1 - "
<< totalCharacters << "): ";
    cin >> index;
    index--;
    if (index >= 0 && index < totalCharacters) {
        for (int i = index; i < totalCharacters - 1; i++) {
            characters[i][0] = characters[i + 1][0];
            characters[i][1] = characters[i + 1][1];
            characters[i][2] = characters[i + 1][2];
        }
        totalCharacters--;
        cout << "Karakter berhasil dihapus!\n";
    } else {
        cout << "Indeks tidak valid!\n";
    }
}

```

```
    }  
    else if (pilihan == 5) {  
        cout << "Terima kasih sudah mau memakai program gw next main  
Honkai Star Rail aja\n";  
        return 0;  
    }  
    else {  
        cout << "Pilihan tidak valid!\n";  
    }  
} while (pilihan != 5);  
}  
return 0;  
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 4.2.1 Pengujian Login (gagal)

Disini Jika Pemogram salah memasukkan username dan password sebanyak 3 kali maka otomatis program berhenti sendiri.

Skenario 4.2.2 Pengujian Login (Berhasil)

Disini jika pemogram memasukkan username dan password dengan benar maka pemogram langsung diarahkan untuk masuk ke bagian menu.

Skenario 4.2.3 Pengujian Menu Tampilkan Karakter

Disini ketika pemogram memilih opsi 1 pemogram diarahkan untuk melihat list list dari karakter Honkai Star Rail.

Skenario 4.2.4 Pengujian Menu Tambah Karakter dan Tampilan Karakter setelah di tambah

Disini ketika pemogram memilih opsi 2 pemogram disuruh untuk menambahkan karakter isi dari karakter tersebut ada 3 yaitu ada Nama, Element, dan Ratiry, ketika sudah di tambahkan pemogram harus balik ke menu Tampilkan Karakter untuk memastikan apakah karakter yang di tambahkan pemogram tersebut ada di list.

Skenario 4.2.5 Pengujian Menu Edit Karakter dan Tampilan Karakter setelah di edit

Disini ketika pemogram memilih opsi 3, pemogram disuruh untuk mengedit atau mengubah karakter, isi dari karakter tersebut ada 3 yaitu ada Nama, Element, dan Ratiry, ketika sudah di edit atau di ubah pemogram harus balik ke menu Tampilkan Karakter untuk memastikan apakah karakter yang di edit atau di ubah pemogram tersebut ada di list.

Skenario 4.2.6 Pengujian Menu Hapus Karakter dan Tampilan Karakter setelah di hapus

Disini ketika pemogram memilih opsi 4, pemogram disuruh untuk menghapus karakter mana yang pengen di hapus, ketika sudah di hapus, pemogram harus balik ke menu Tampilkan Karakter untuk memastikan apakah karakter yang di hapus pemogram tersebut hilang atau masi tetap.

Skenario 4.2.7 Pengujian Menu Keluar

Disini ketika pemogram memilih opsi 5 atau keluar, program bakalan langsung berhenti.

4.2 Hasil Output

```
Masukkan Username: Haqi
Masukkan Password: 2409106081
Login gagal! Username atau password salah.
Masukkan Username: Fahmi
Masukkan Password: 2409106079
Login gagal! Username atau password salah.
Masukkan Username: Devon
Masukkan Password: 2409106055
Login gagal! Username atau password salah.
Anda telah gagal login 3 kali. Program berhenti.
PS C:\Users\melch\OneDrive\Documents\Praktikum-APD\pertemuan 4> |
```

Gambar 4.2.1 Pengujian Login (gagal)

```
Masukkan Username: Melchizedek
Masukkan Password: 2409106117
Berhasil login!

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: |
```

Gambar 4.2.2 Pengujian Login (Berhasil)

```
===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 1

Daftar Karakter:
Indeks  Nama                Elemen  Rarity
-----
1       Silver Wolf            Quantum  5
2       Kafka                   Lightning 5
3       Dan Heng                 Wind     4
```

Gambar 4.2.3 Pengujian Menu Tampilkan Karakter

```

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 2
Masukkan Nama Karakter: Himeko
Masukkan Elemen Karakter: Fire
Masukkan Rarity Karakter: 5
Karakter berhasil ditambahkan!

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 1

Daftar Karakter:
Indeks  Nama                Elemen  Rarity
-----
1      Silver Wolf             Quantum  5
2      Kafka                    Lightning 5
3      Dan Heng                 Wind     4
4      Himeko                   Fire     5

```

Gambar 4.2.4 Pengujian Menu Tambah Karakter dan Tampilan Karakter setelah di tambah

```
===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 3

Daftar Karakter:
Indeks  Nama                Elemen  Rarity
-----
1       Silver Wolf         Quantum  5
2       Kafka               Lightning 5
3       Dan Heng            Wind     4
4       Himeko              Fire     5
Masukkan indeks karakter yang ingin diedit (1 - 4): 3
Masukkan Nama Baru: Dan Heng
Masukkan Elemen Baru: Imaginary
Masukkan Rarity Baru: 5
Karakter berhasil diperbarui!

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 1

Daftar Karakter:
Indeks  Nama                Elemen  Rarity
-----
1       Silver Wolf         Quantum  5
2       Kafka               Lightning 5
3       Dan Heng            Imaginary 5
4       Himeko              Fire     5
```

Gambar 4.2.5 Pengujian Menu Edit Karakter dan Tampilan Karakter setelah di edit

```

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 4

```

Daftar Karakter:

| Indeks | Nama | Elemen | Rarity |
|--------|-------------|-----------|--------|
| 1 | Silver Wolf | Quantum | 5 |
| 2 | Kafka | Lightning | 5 |
| 3 | Dan Heng | Imaginary | 5 |
| 4 | Himeko | Fire | 5 |

Masukkan indeks karakter yang ingin dihapus (1 - 4): 4
Karakter berhasil dihapus!

```

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 1

```

Daftar Karakter:

| Indeks | Nama | Elemen | Rarity |
|--------|-------------|-----------|--------|
| 1 | Silver Wolf | Quantum | 5 |
| 2 | Kafka | Lightning | 5 |
| 3 | Dan Heng | Imaginary | 5 |

Gambar 4.2.6 Pengujian Menu Hapus Karakter dan Tampilan Karakter setelah di hapus

```

===== Sistem Manajemen Karakter Honkai: Star Rail =====
1. Tampilkan Karakter
2. Tambah Karakter
3. Edit Karakter
4. Hapus Karakter
5. Keluar
Pilihan: 5
Terima kasih sudah mau memakai program gw next main Honkai Star Rail aja

```

Gambar 4.2.7 Pengujian Menu Keluar

5. Langkah Langkah Git

```
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git add .
warning: in the working copy of 'Post-Test/Post-Test-2/2409106117-MelchizedekJulirosalomoSimangunsong-PT-2.drawio', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git commit -m "rusdi ganteng"
[master 3b86123] rusdi ganteng
5 files changed, 968 insertions(+)
create mode 100644 Post-Test/Post-Test-1/2409106117-MelchizedekJulirosalomoSimangunsong-PT-1-Melchizedek.pdf
create mode 100644 Post-Test/Post-Test-1/2409106117-MelchizedekJulirosalomoSimangunsong-PT-1.docx
create mode 100644 Post-Test/Post-Test-2/2409106117-MelchizedekJulirosalomoSimangunsong-PT-2.cpp
create mode 100644 Post-Test/Post-Test-2/2409106117-MelchizedekJulirosalomoSimangunsong-PT-2.drawio
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'https://github.com/Melchi050706/Praktikum-APL.git'
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git push origin master
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 532.47 KiB | 6.74 MiB/s, done.
Total 9 (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/Melchi050706/Praktikum-APL.git
6ccadc5..3b86123 master -> master
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> 
```

1. Git add & commit

Melakukan git add untuk menambahkan file yang akan kita commit, dan melakukan git commit untuk membuat checkpoint

2. Git Push

Melakukan git push untuk mengupload semua yang ada pada repository kita