

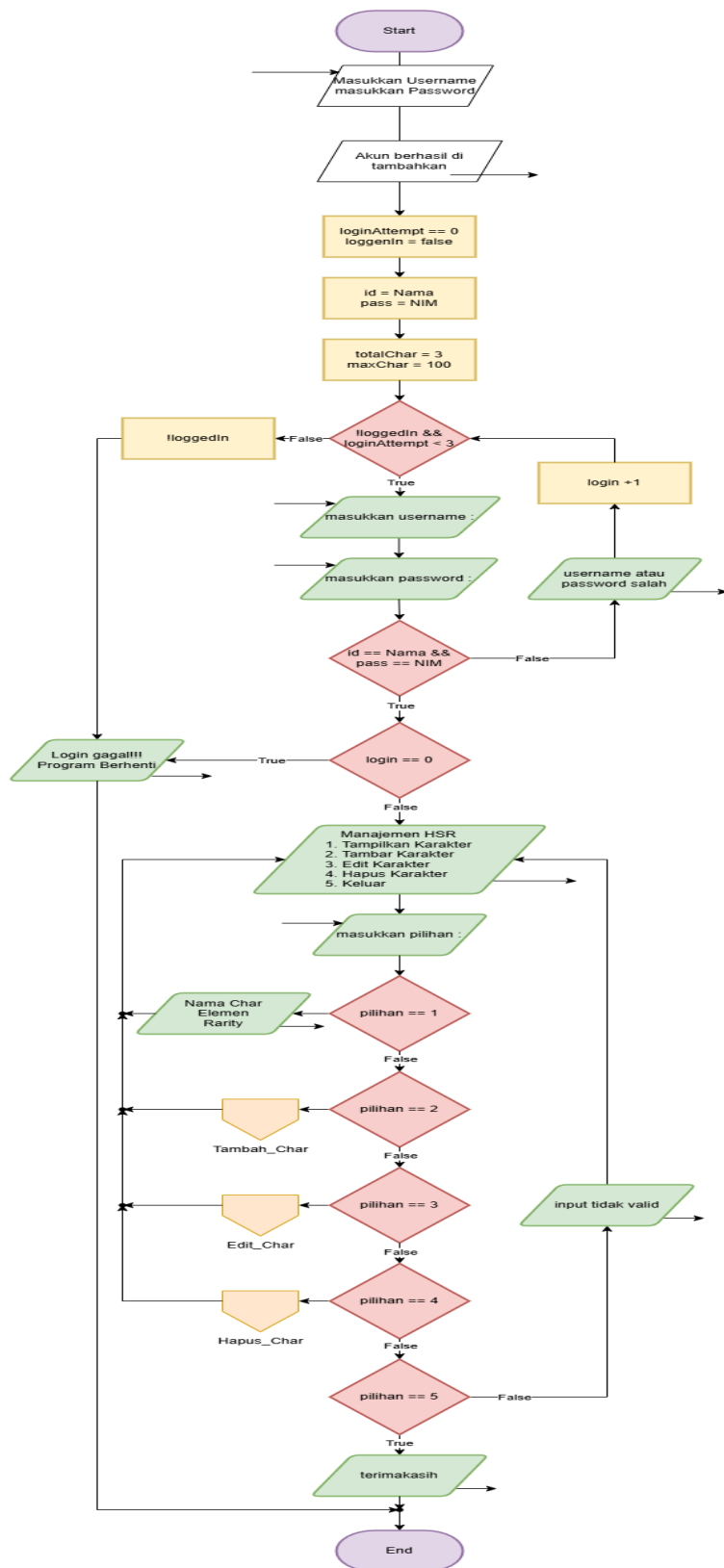
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT

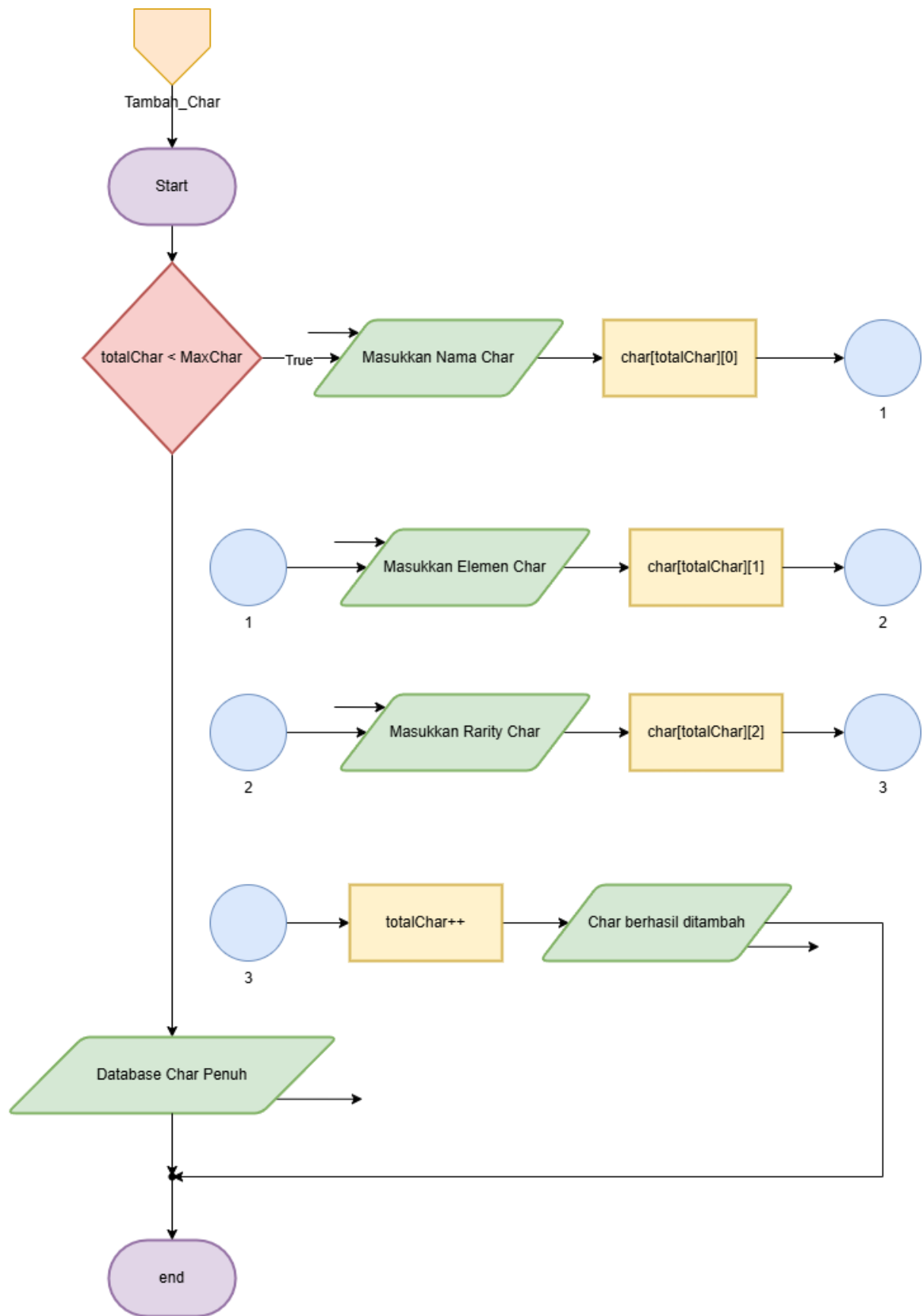


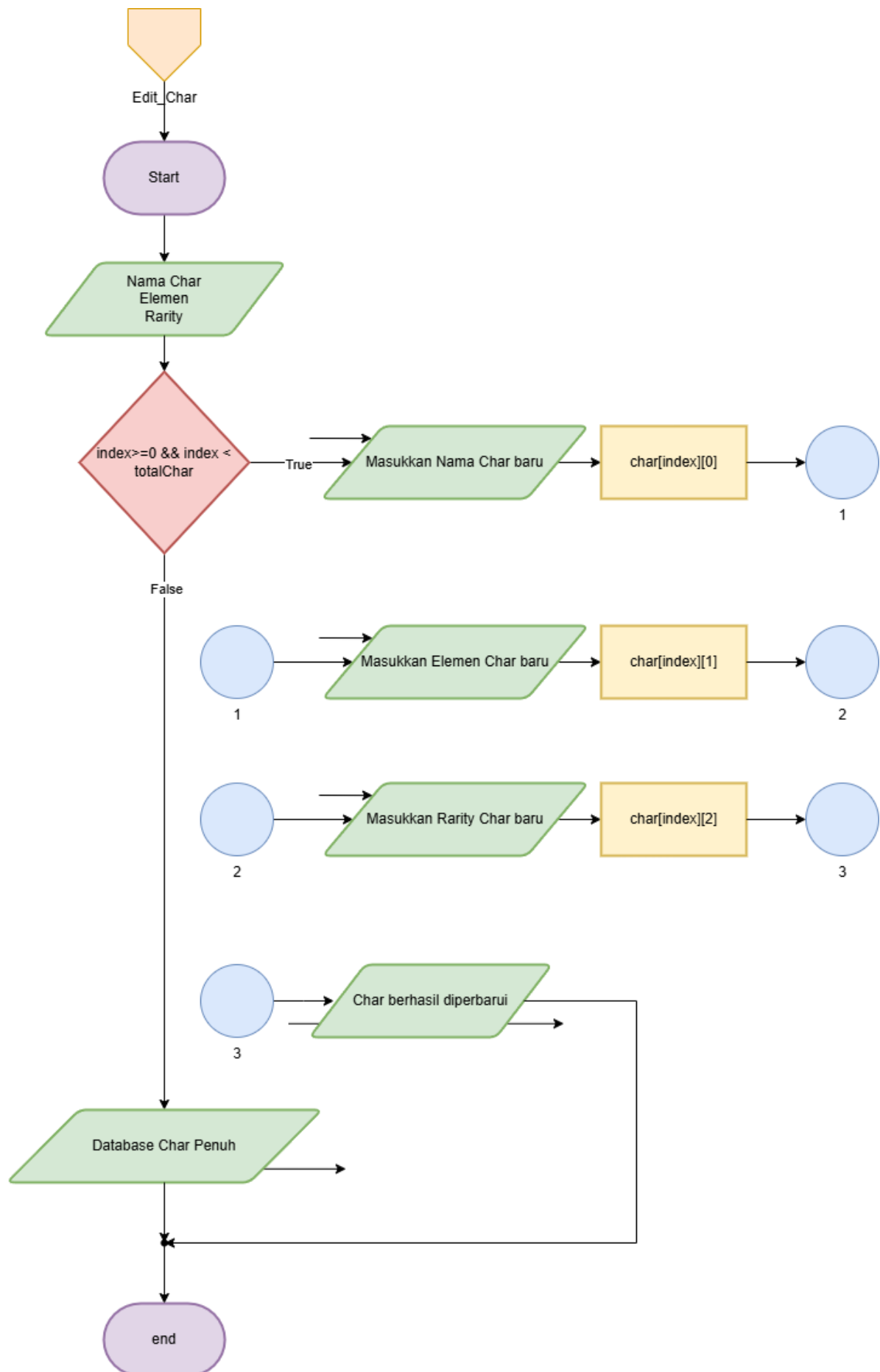
Disusun oleh:
Nama (2409106117)
Kelas (C2 '24)

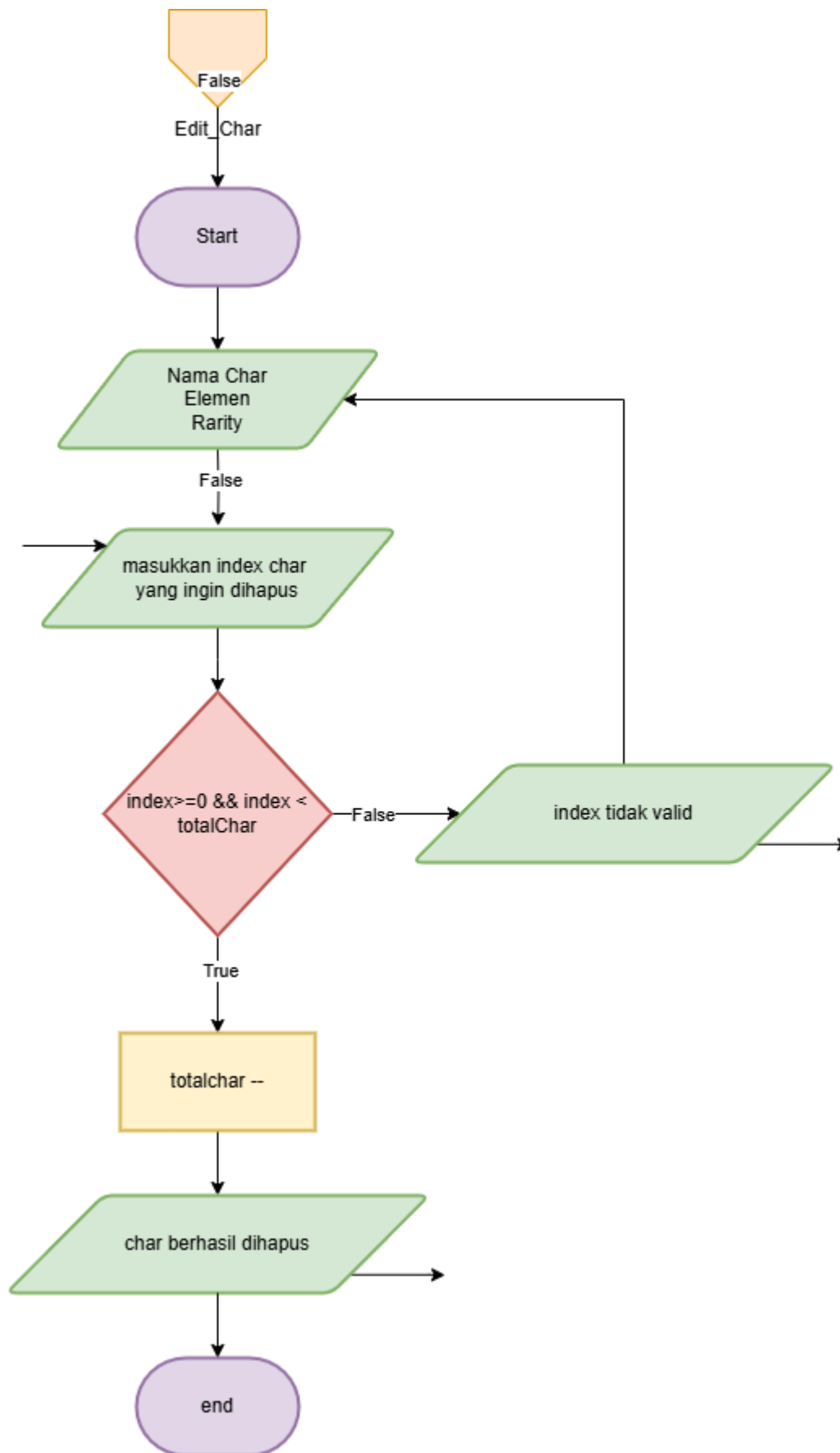
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart









2. Analisis Program

2.1 Deskripsi Singkat Program

Program "Sistem Manajemen Karakter Honkai: Star Rail" ini merupakan aplikasi berbasis C++ yang memungkinkan pengguna untuk mengelola data karakter dari game Honkai: Star Rail. Program ini bersifat multiuser, di mana pengguna dapat melakukan registrasi dan login, kemudian mengakses berbagai fitur manajemen karakter, seperti:

- Menampilkan daftar karakter
- Menambahkan karakter baru
- Mengedit karakter yang sudah ada
- Menghapus karakter tertentu

Program ini telah dimodifikasi agar sesuai dengan instruksi:

- Semua fitur utama dipisahkan menjadi subprogram menggunakan fungsi dan prosedur.
- Fungsi menggunakan parameter untuk pengolahan data.
- Menggunakan nested struct untuk menyimpan data pengguna dan karakter.
- Menambahkan fungsi rekursif untuk validasi login.
- Menambahkan fungsi overloading untuk menampilkan karakter (baik secara total maupun per rarity).

3. Source Code

A. Program Register Dan Login dengan Rekursif

```
void registrasi() {
    cout << "==== Registrasi Akun ====\\n";
    if (totalUsers < MAX_USERS) {
        cout << "Masukkan Username: ";
        getline(cin, users[totalUsers].username);
        cout << "Masukkan Password: ";
        getline(cin, users[totalUsers].password);
        totalUsers++;
        cout << "Akun berhasil didaftarkan!\\n";
    }
}

bool login(User &currentUser) {
    int loginAttempts = 0;
    bool loggedIn = false;

    while (!loggedIn && loginAttempts < 3) {
        cout << "\\n==== Login ====\\n";
        cout << "Masukkan Username: ";
        getline(cin, currentUser.username);
        cout << "Masukkan Password: ";
        getline(cin, currentUser.password);

        for (int i = 0; i < totalUsers; i++) {
            if (users[i].username == currentUser.username &&
                users[i].password == currentUser.password) {
                loggedIn = true;
                cout << "Berhasil login!\\n";
                break;
            }
        }

        if (!loggedIn) {
            cout << "Login gagal, coba lagi.\\n";
            loginAttempts++;
        }
    }

    return loggedIn;
}
```

```
}
```

B. Program Menu

```
void menu() {
    int pilihan;
    do {
        cout << "\n===== Menu 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.ignore();

        switch (pilihan) {
            case 1: tampilKarakter(); break;
            case 2: tambahKarakter(); break;
            case 3: editKarakter(); break;
            case 4: hapusKarakter(); break;
            case 5: break;
            default: cout << "Pilihan tidak valid!\n";
        }
    } while (pilihan != 5);
}
```

C. Program Tampilkan Karakter

```
void tampilKarakter(int index) {
    if (index == 0) {
        cout << "\nDaftar Karakter:\n";
        cout << left << setw(8) << "Indeks" << setw(20) << "Nama"
            << setw(15) << "Elemen" << setw(10) << "Rarity" << endl;
        cout << "-----\n";
    }

    if (index < totalCharacters) {
        cout << left << setw(8) << (index + 1)
            << setw(20) << characters[index].name
            << setw(15) << characters[index].element
            << setw(10) << characters[index].rarity << endl;
    }
}
```



```

        tampilKarakter(index + 1); // rekursif lanjut
    }
}

```

D. Program Tambah Karakter

```

void tambahKarakter() {
    if (totalCharacters < MAX_CHARACTERS) {
        cout << "Masukkan Nama Karakter: ";
        getline(cin, characters[totalCharacters].name);
        cout << "Masukkan Elemen Karakter: ";
        getline(cin, characters[totalCharacters].element);
        cout << "Masukkan Rarity Karakter: ";
        cin >> characters[totalCharacters].rarity;
        cin.ignore();
        totalCharacters++;
        cout << "Karakter berhasil ditambahkan!\n";
    } else {
        cout << "Database karakter penuh!\n";
    }
}

```

E. Program Edit Karakter

```

void editKarakter() {
    tampilKarakter();
    int index;
    cout << "Masukkan indeks karakter yang ingin diedit: ";
    cin >> index; cin.ignore();
    index--;

    if (index >= 0 && index < totalCharacters) {
        cout << "Masukkan Nama Baru: ";
        getline(cin, characters[index].name);
        cout << "Masukkan Elemen Baru: ";
        getline(cin, characters[index].element);
        cout << "Masukkan Rarity Baru: ";
        cin >> characters[index].rarity;
        cin.ignore();
        cout << "Karakter berhasil diperbarui!\n";
    } else {
        cout << "Indeks tidak valid!\n";
    }
}

```

F. Program Hapus Karakter

```
void hapusKarakter() {
    tampilKarakter();
    int index;
    cout << "Masukkan indeks karakter yang ingin dihapus: ";
    cin >> index; cin.ignore();
    index--;

    if (index >= 0 && index < totalCharacters) {
        for (int i = index; i < totalCharacters - 1; i++) {
            characters[i] = characters[i + 1];
        }
        totalCharacters--;
        cout << "Karakter berhasil dihapus!\n";
    } else {
        cout << "Indeks tidak valid!\n";
    }
}
```

G. Program Keluar dengan Overloading

```
void keluar() {
    cout << "Terima kasih sudah menggunakan program ini!\n";
}

void keluar(string username) {
    cout << "Sampai jumpa lagi, " << username << "! Terima kasih sudah login.\n";
}
```

4. Uji coba dan Hasil Output

4.2 Hasil Output

```
PS C:\Users\melch\OneDrive\Documents\APL> cd "c:\Users\melch\OneDrive\Documents\APL\" ; if ($?) { g++ 2409106117-MelchizedekJulirosalomoSimangunsong-PT-3.cpp -o 2409106117-MelchizedekJulirosalomoSimangunsong-PT-3 } ; if ($?) { .\2409106117-MelchizedekJulirosalomoSimangunsong-PT-3 }  
==== Registrasi Akun ====  
Masukkan Username: Melchi  
Masukkan Password: 117  
Akun berhasil didaftarkan!  
  
==== Login ====  
Masukkan Username: Melchi  
Masukkan Password: 117  
Berhasil login!  
  
==== 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  
  
==== Sistem Manajemen Karakter Honkai: Star Rail ====  
1. Tampilkan Karakter  
2. Tambah Karakter  
3. Edit Karakter  
4. Hapus Karakter  
5. Keluar  
Pilihan: 
```

```

===== 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: Yan Qing
Masukkan Elemen Karakter: Ice
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: 3

```

```

Daftar Karakter:
Indeks  Nama          Elemen  Rarity
-----
1      Silver Wolf      Quantum  5
2      Kafka             Lightning 5
3      Dan Heng          Wind     4
4      Yan Qing          Ice      5
Masukkan indeks karakter yang ingin diedit (1 - 4): 4
Masukkan Nama Baru: Bailu
Masukkan Elemen Baru: Lightning
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: 4

```

```

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

```

```
===== 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 melihat program ini, nextim kita buat program Wuthering Waves kalo di bolehkan Bang Ade!  
PS C:\Users\melch\OneDrive\Documents\APL>
```

5. Langkah-Langkah Git pada VSCode

```
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git add .
warning: in the working copy of 'Post-Test/Post-Test-4/2409106117-MelchizedekJuliroSalomoSimangunsong-PT-4.drawio', L
F will be replaced by CRLF the next time Git touches it
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git commit "Pernan Ganteng"
error: pathspec 'Pernan Ganteng' did not match any file(s) known to git
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git commit -m "Pernan Ganteng"
[master 396d61b] Pernan Ganteng
 2 files changed, 1053 insertions(+)
 create mode 100644 Post-Test/Post-Test-4/2409106117-MelchizedekJuliroSalomoSimangunsong-PT-4.cpp
 create mode 100644 Post-Test/Post-Test-4/2409106117-MelchizedekJuliroSalomoSimangunsong-PT-4.drawio
PS C:\Users\melch\OneDrive\Documents\Praktikum-APL> git push origin master
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 7.75 KiB | 1.29 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/Melchi050706/Praktikum-APL.git
 fd4c1dc..396d61b 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