

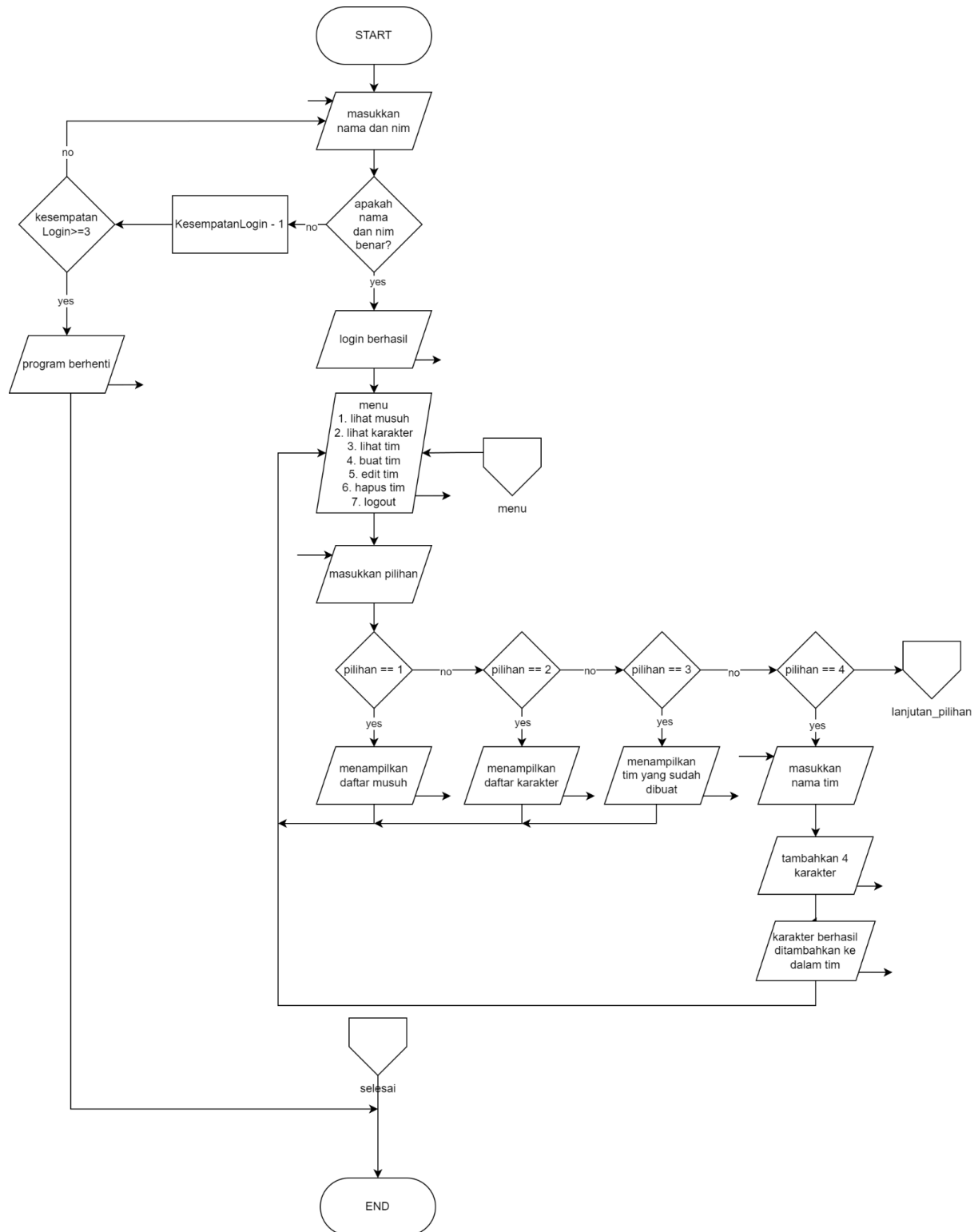
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



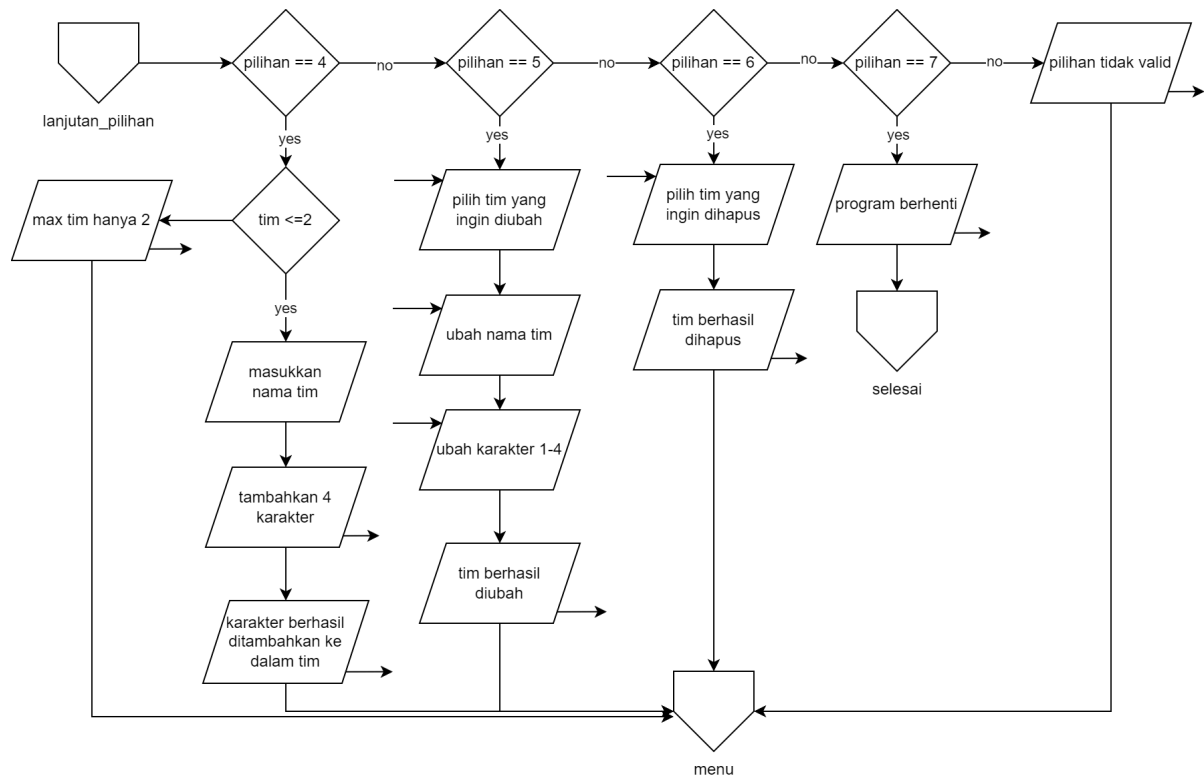
**Disusun oleh:**  
**Inayah Ramadhani (2409106068)**  
**Kelas (B1 '24)**

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

# 1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini merupakan Manajemen Tim Spiral Abyss pada game Genshin Impact yang memungkinkan pengguna dapat mensimulasikan cara kerja Spiral Abyss di mana pengguna dapat login menggunakan nama dan NIM, melihat daftar musuh yang akan dilawan, melihat daftar karakter yang tersedia, melihat daftar tim yang sudah dibuat, membuat tim, mengedit tim, menghapus tim, dan keluar dari program.

### **2.2 Penjelasan Alur & Algoritma**

#### **1. Inisialisasi variabel**

- nama: Menyimpan nama pengguna yang benar
- inputNama: menyimpan nama yang diinput oleh pengguna
- nim: Menyimpan nim pengguna yang benar
- inputNim : Menyimpan NIM yang diinput oleh pengguna
- saldo : Menyimpan nominal saldo pengguna, saldo awalnya Rp0
- kesempatanLogin : Menyimpan jumlah percobaan login yang dapat dilakukan oleh pengguna
- pilihan : Menyimpan pilihan menu yang diinput oleh pengguna
- musuh : Menyimpan daftar musuh
- level: menyimpan daftar level musuh
- karakter: Menyimpan daftar karakter
- namaTim: Menyimpan nama tim yang dibuat oleh pengguna

#### **2. Autentikasi login**

Pada tahap ini, pengguna diminta untuk menginput nama dan NIM. Fitur ini menggunakan perulangan for untuk membatasi pengguna login dengan batas sebanyak 3 kali. Jika pengguna memasukkan nama dan NIM dengan benar, maka pengguna akan masuk ke menu program. Jika pengguna salah memasukkan nama atau NIM maka akan menampilkan pesan untuk mencoba login lagi, namun jika gagal 3 kali berturut-turut, maka program akan langsung berhenti.

### 3. Menu utama

Setelah berhasil login, pengguna akan masuk ke menu utama program. Di menu terdapat 7 pilihan yaitu lihat musuh, lihat karakter, lihat tim, buat tim, edit tim, hapus tim, dan keluar. Menu ini menggunakan perulangan while(true) agar terus berjalan, jadi setelah selesai memproses salah satu pilihan menu, program akan kembali lagi ke menu. Program akan berhenti hanya jika hingga pengguna memilih untuk berhenti.

- Lihat musuh

Jika pengguna memilih pilihan menu 1, maka program akan menampilkan daftar musuh yang akan dilawan di first half dan second half chamber 1 sampai 3.

- Lihat karakter

Jika pengguna memilih pilihan menu 2, maka program akan menampilkan daftar karakter yang tersedia. Mengingat karakter Genshin Impact yang sangat banyak, jadi di sini saya hanya memasukkan 8 karakter saja sebagai contoh.

- Lihat tim

Jika pengguna memilih pilihan menu 3, maka program akan menampilkan tim yang sudah pernah dibuat oleh pengguna sebelumnya. Jika belum pernah membuat tim, program akan menampilkan pesan bahwa belum ada tim yang dibuat.

- Buat tim

Jika pengguna memilih pilihan menu 4, maka program akan meminta pengguna untuk memasukkan nama tim yang ingin dibuat lalu memasukkan 4 karakter yang ingin dimasukkan ke dalam tim. Namun pengguna hanya dapat membuat 2 tim, karena spiral abyss hanya ada 2 half, yaitu first half dan second half.

- Edit tim

Jika pengguna memilih pilihan menu 5, maka program akan meminta pengguna untuk memasukkan indeks tim yang ingin diubah, lalu pengguna dapat mengubah nama tim dan karakter.

- Hapus Tim

Jika pengguna memilih pilihan menu 6, maka program akan meminta pengguna untuk memasukkan indeks tim yang ingin dihapus, setelah itu program akan menampilkan pesan tim berhasil dihapus.

- Logout

Jika pengguna memilih pilihan menu 7, maka program akan langsung berhenti.

### 3. Source Code

#### A. Verifikasi Login

Fitur ini berguna untuk memungkinkan pengguna dapat masuk ke sistem dengan memasukkan nama dan NIM yang benar.

```
for (int percobaan = 0; percobaan < kesempatanLogin; percobaan++) {
    cout << "Nama: ";
    getline(cin, inputNama);

    cout << "NIM: ";
    cin >> inputNim;

    if (inputNama == nama && inputNim == nim) {
        cout << "Berhasil login!" << endl;
        cin.ignore();
        break;
    } else {
        cout << "Nama atau NIM yang Anda masukkan salah." << endl;
        cout << "Percobaan tersisa " << (kesempatanLogin - 1 - percobaan) << "
kali." << endl;
        cin.ignore();
    }

    if (percobaan == kesempatanLogin - 1) {
        cout << "Anda telah salah memasukkan Nama atau NIM 3 kali." << endl;
        cout << "Program telah berhenti." << endl;
        return 0;
    }
}
```

#### B. Menu

Program ini memiliki beberapa fitur yang dapat dipilih oleh pengguna.

```
cout << "[1] Lihat Musuh" << endl;
cout << "[2] Lihat Karakter" << endl;
cout << "[3] Lihat Tim" << endl;
cout << "[4] Buat Tim" << endl;
cout << "[5] Edit Tim" << endl;
cout << "[6] Hapus Tim" << endl;
cout << "[7] Logout" << endl;
cout << "Pilih menu: ";
cin >> pilihan;
```

### C. Lihat Musuh

Pada fitur ini, pengguna dapat melihat musuh apa saja yang akan dilawan.

```
cout << left << setw(10) << "Chamber" << setw(15) << "Half"
      << setw(50) << "Musuh" << "Lv." << endl;
cout << string(85, '-') << endl;

for (int i = 0; i < 3; i++) {
    cout << "Chamber " << i + 1 << endl;

    cout << setw(10) << "" << setw(15) << "First Half";
    cout << setw(50) << musuh[i][0] << level[i][0] << endl;
    cout << setw(10) << "" << setw(15) << "" << setw(50) << musuh[i][1] <<
    level[i][1] << endl;

    cout << setw(10) << "" << setw(15) << "Second Half";
    if (!musuh[i][2].empty()) {
        cout << setw(50) << musuh[i][2] << level[i][2] << endl;
    }
    if (!musuh[i][3].empty()) {
        cout << setw(10) << "" << setw(15) << "" << setw(50) << musuh[i][3] <<
        level[i][3] << endl;
    }
    cout << string(85, '-') << endl;
}
```

### D. Lihat karakter

Pada fitur ini, pengguna dapat melihat karakter apa saja yang tersedia.

```
cout << left << setw(5) << "No" << setw(15) << "Nama"
      << setw(10) << "Elemen" << setw(10) << "Senjata" << endl;
cout << string(40, '-') << endl;

for (int i = 0; i < 8; i++) {
    cout << left << setw(5) << karakter[i][0] << setw(15) << karakter[i][1]
        << setw(10) << karakter[i][2] << setw(10) << karakter[i][3] << endl;
}
cout << "\n";
```

### E. Lihat Tim

Fitur ini memungkinkan pengguna untuk melihat tim yang sudah dibuat.

```
if (namaTim[0] == "" && namaTim[1] == "") {
    cout << "Belum ada tim yang dibuat." << endl;
} else {
    for (int i = 0; i < 2; i++) {
```



```

        if (namaTim[i] != "") {
            cout << "Tim " << namaTim[i] << ": ";
            for (int j = 0; j < 4; j++) {
                if (timKarakter[i][j] != "") {
                    cout << timKarakter[i][j] << " ";
                }
            }
            cout << "\n";
        }
    }
    cout << "\n";
}

```

## F. Buat Tim

Fitur ini memungkinkan pengguna untuk membuat tim baru.

```

int timIndex = (namaTim[0] == "") ? 0 : (namaTim[1] == "") ? 1 :
-1;
if (timIndex == -1) {
    cout << "Sudah ada 2 tim, tidak bisa menambah lagi." << endl;
} else {
    cout << "Masukkan nama tim: ";
    cin.ignore();
    getline(cin, namaTim[timIndex]);

    cout << "\nDaftar Karakter :" << endl;
    for (int i = 0; i < 8; i++) {
        cout << karakter[i][0] << ". " << karakter[i][1] << " ("
<< karakter[i][2] << ", " << karakter[i][3] << ")" << endl;
    }

    cout << "\nMasukkan nomor karakter yang ingin dipilih :" <<
endl;
    for (int j = 0; j < 4; j++) {
        int pilihIndex;
        cout << "Karakter " << j + 1 << ": ";
        cin >> pilihIndex;
        if (pilihIndex >= 1 && pilihIndex <= 8) {
            timKarakter[timIndex][j] = karakter[pilihIndex -
1][1];
        } else {

```

```

        cout << "Nomor karakter tidak valid." << endl;
        j--;
    }
}
cout << "Tim " << namaTim[timIndex] << " berhasil dibuat!" <<
endl;
}

```

## G. Edit Tim

Fitur ini memungkinkan pengguna untuk mengganti nama tim dan karakter.

```

int pilihTim;
cout << "Pilih tim yang ingin diubah: " << endl;
for (int i = 0; i < 2; i++) {
    if (!namaTim[i].empty()) {
        cout << i + 1 << ". " << namaTim[i] << endl;
    }
}
cin >> pilihTim;

if (pilihTim >= 1 && pilihTim <= 2 && !namaTim[pilihTim - 1].empty()) {
    cout << "Masukkan nama baru untuk tim: ";
    cin.ignore();
    getline(cin, namaTim[pilihTim - 1]);

    cout << "Daftar Karakter: " << endl;
    for (int i = 0; i < 8; i++) {
        cout << karakter[i][0] << ". " << karakter[i][1] << " (" <<
karakter[i][2] << ", " << karakter[i][3] << ")" << endl;
    }

    for (int j = 0; j < 4; j++) {
        int pilihKarakter;
        cout << "Karakter " << j + 1 << ": ";
        cin >> pilihKarakter;

        if (pilihKarakter >= 1 && pilihKarakter <= 8) {
            timKarakter[pilihTim - 1][j] = karakter[pilihKarakter - 1][1];
        } else {
            cout << "Nomor karakter tidak valid." << endl;
            j--;
        }
    }
}

```

```

        cout << "Tim berhasil diubah!" << endl;
    } else {
        cout << "Pilihan tim tidak valid." << endl;
    }
}

```

## H. Hapus Tim

Fitur ini memungkinkan pengguna untuk menghapus tim yang ada.

```

int pilihTim;
cout << "Pilih tim yang ingin dihapus: " << endl;
for (int i = 0; i < 2; i++) {
    if (!namaTim[i].empty()) {
        cout << i + 1 << ". " << namaTim[i] << endl;
    }
}
cin >> pilihTim;

if (pilihTim >= 1 && pilihTim <= 2 && !namaTim[pilihTim - 1].empty()) {
    namaTim[pilihTim - 1] = "";
    for (int j = 0; j < 4; j++) {
        timKarakter[pilihTim - 1][j] = "";
    }
    cout << "Tim berhasil dihapus!" << endl;
} else {
    cout << "Pilihan tim tidak valid." << endl;
}
}

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

#### 1. Skenario 1

Uji coba verifikasi PIN di mana pengguna memasukkan nama dan NIM salah sebanyak 3 kali lalu program akan langsung berhenti.

#### 2. Skenario 2

Uji coba jika pengguna memilih pilihan selain yang tersedia di menu (1-7), maka akan muncul pesan pilihan tidak valid.

#### 3. Skenario 3

Uji coba pengguna membuat lebih dari 2 tim, maka akan menampilkan pesan max tim yang dibuat adalah 2.

#### 4. Skenario 4

Uji coba pengguna berhasil mengubah tim.

### 4.2 Hasil Output

```
=====
      Selamat Datang
      Silahkan Login
=====
Nama: Inayah ramadhani
NIM: 2409106068
Nama atau NIM yang Anda masukkan salah.
Percobaan tersisa 2 kali.
Nama: a
NIM: a
Nama atau NIM yang Anda masukkan salah.
Percobaan tersisa 1 kali.
Nama: a
NIM: a
Nama atau NIM yang Anda masukkan salah.
Percobaan tersisa 0 kali.
Anda telah salah memasukkan Nama atau NIM 3 kali.
Program telah berhenti.
```

Gambar 4.1 Hasil uji coba skenario 1

```
=====
Manajemen Tim Spiral Abyss
=====
[1] Lihat Musuh
[2] Lihat Karakter
[3] Lihat Tim
[4] Buat Tim
[5] Edit Tim
[6] Hapus Tim
[7] Logout
Pilih menu: 8

Pilihan tidak tersedia. Coba lagi.
```

Gambar 4.2 Hasil uji coba skenario 2

```
=====
Manajemen Tim Spiral Abyss
=====
[1] Lihat Musuh
[2] Lihat Karakter
[3] Lihat Tim
[4] Buat Tim
[5] Edit Tim
[6] Hapus Tim
[7] Logout
Pilih menu: 4

Sudah ada 2 tim, tidak bisa menambah lagi.
```

Gambar 4.2 Hasil uji coba skenario 3

```
Pilih tim yang ingin diubah:
1. Apalah
2. Yaudah
1
Masukkan nama baru untuk tim: Buset
Daftar Karakter:
1. Arlecchino (Pyro, Polearm)
2. Furina (Hydro, Sword)
3. Ororon (Electro, Bow)
4. Yelan (Hydro, Bow)
5. Xingqiu (Hydro, Sword)
6. Chasca (Anemo, Bow)
7. Bennett (Pyro, Sword)
8. Xiangling (Pyro, Polearm)
Karakter 1: 2
Karakter 2: 3
Karakter 3: 4
Karakter 4: 5
Tim berhasil diubah!
```

Gambar 4.2 Hasil uji coba skenario 4

## 5. Git

### 5.1 Git Add

Git add digunakan untuk menandai file untuk dimasukkan ke staging area sebelum dicommit.

```
ADVAN@choi3an MINGW64 ~/praktikum-apl/post-test/post-test-2 (main)
$ git add .
```

Gambar 5.2 Git add

### 5.3 Git Commit

Git commit berfungsi untuk menyimpan perubahan ke repositori lokal, -m “...” digunakan sebagai catatan perubahan.

```
ADVAN@choi3an MINGW64 ~/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "final posttest2"
[main (root-commit) 6e66997] final posttest2
 2 files changed, 249 insertions(+)
 create mode 100644 240910068-InayahRamadhani-PT-2.cpp
 create mode 100644 240910068-InayahRamadhani-PT-2.exe
```

Gambar 5.3 Git commit

### 5.4 Remote Add Origin

Git remote add origin berfungsi untuk menghubungkan repository lokal ke repositori yang ada di GitHub.

```
ADVAN@choi3an MINGW64 ~/praktikum-apl/post-test/post-test-2 (main)
$ git remote add origin https://github.com/naiversal/praktikum-apl.git
```

Gambar 5.4 Remote add origin

### 5.5 Git Push -u Main Origin

Git push -u main origin berfungsi untuk mengunggah commit yang telah dibuat ke repositori GitHub.

Gambar 5.5 Git push -u main origin