

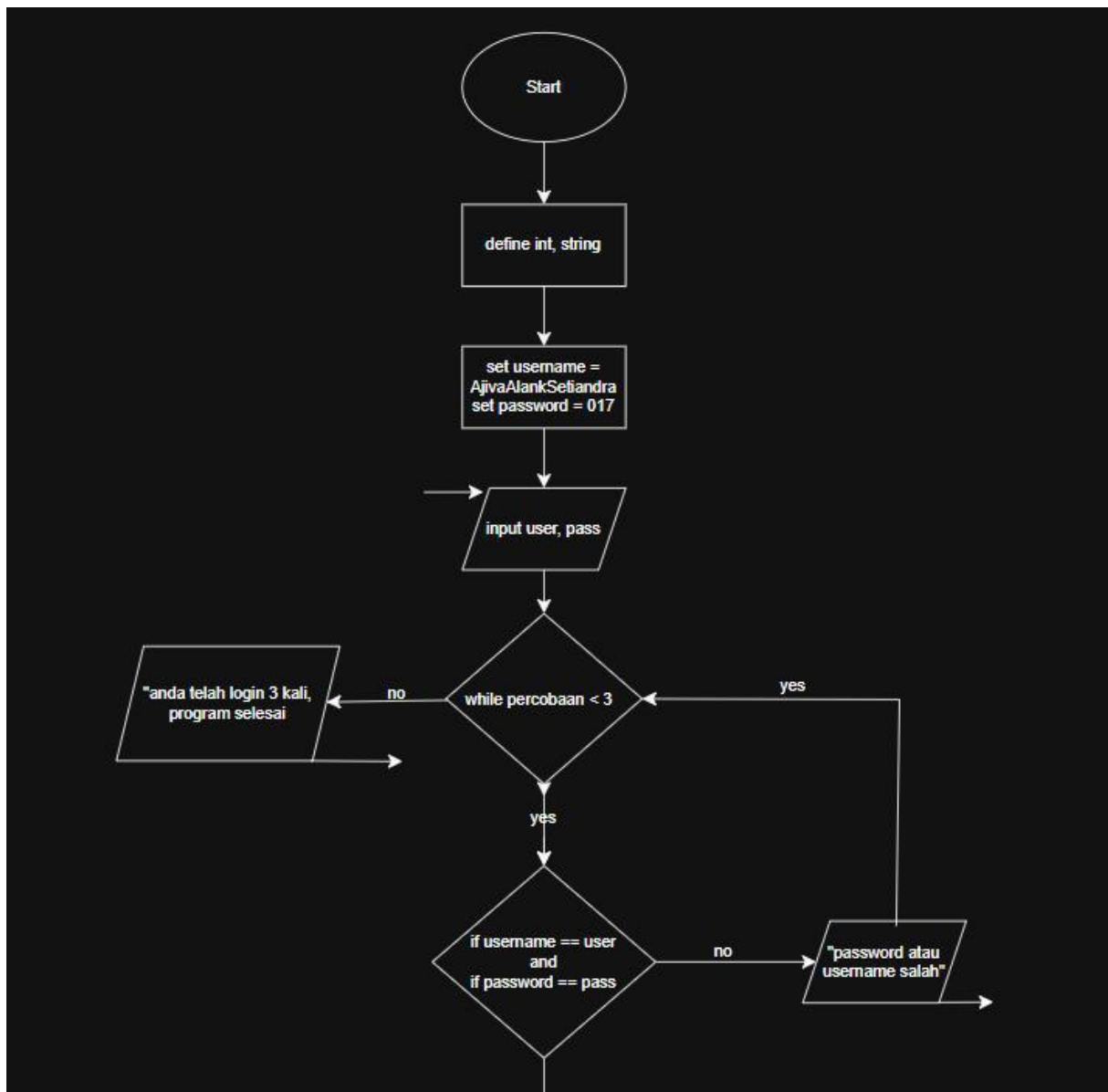
**LAPORAN PRAKTIKUM**  
**POSTTEST 3**  
**ALGORITMA PEMROGRAMAN LANJUT**



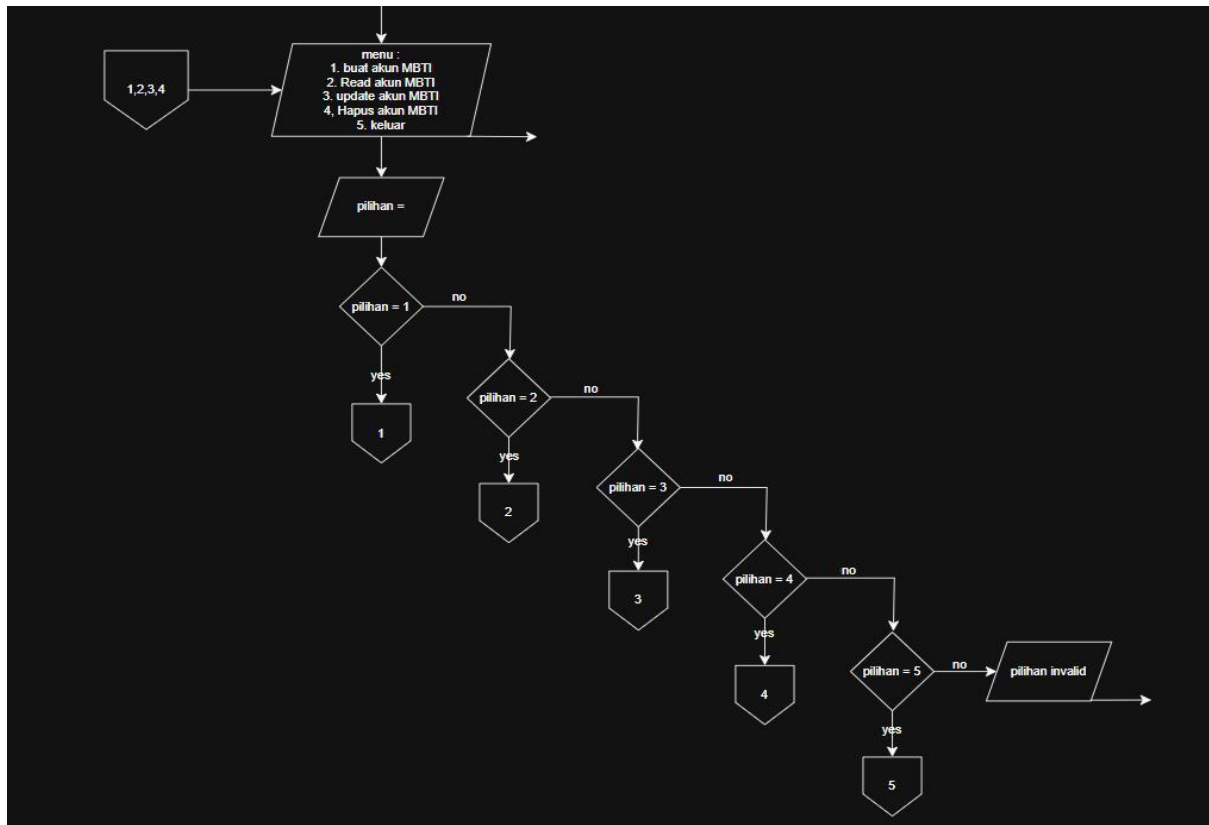
**Disusun oleh:**  
**Ajiva Alank Setiandra (2409106017)**  
**Kelas (A1 '24)**

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

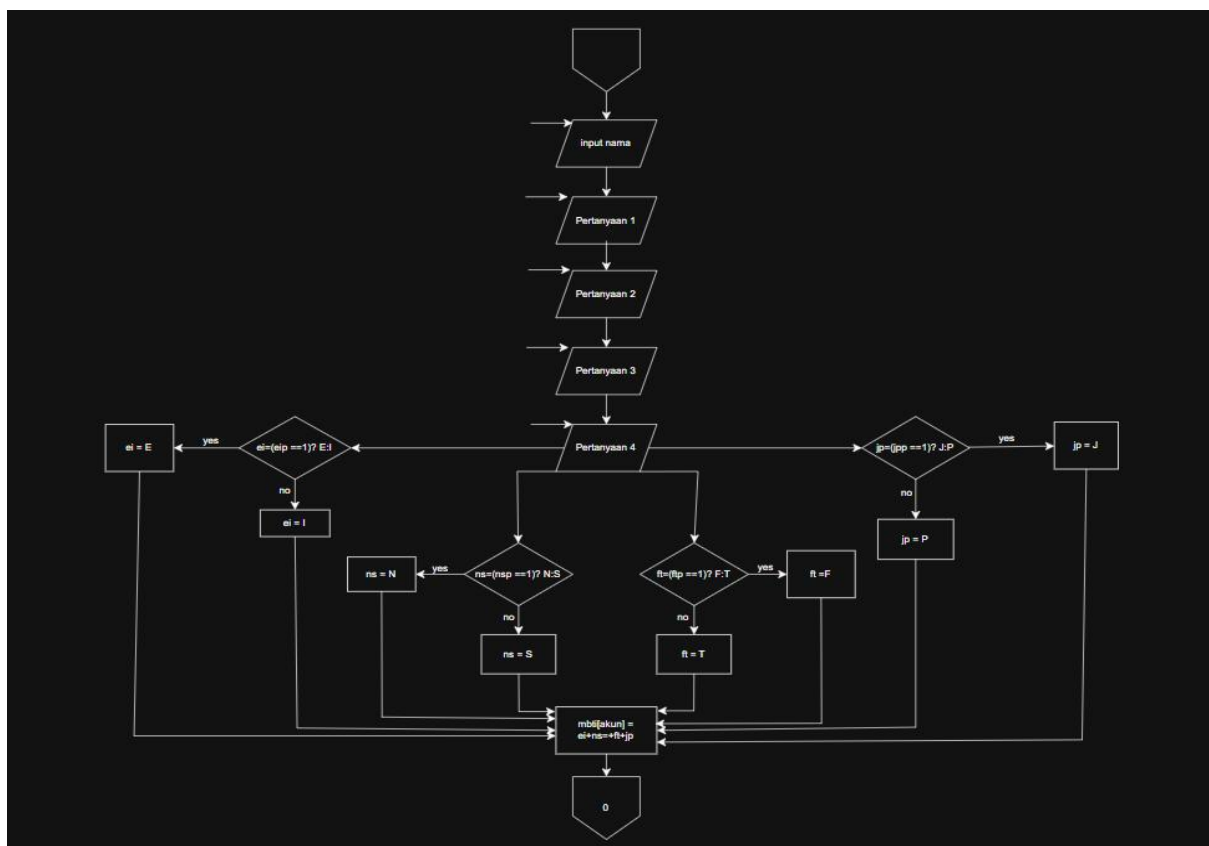
## 1. Flowchart



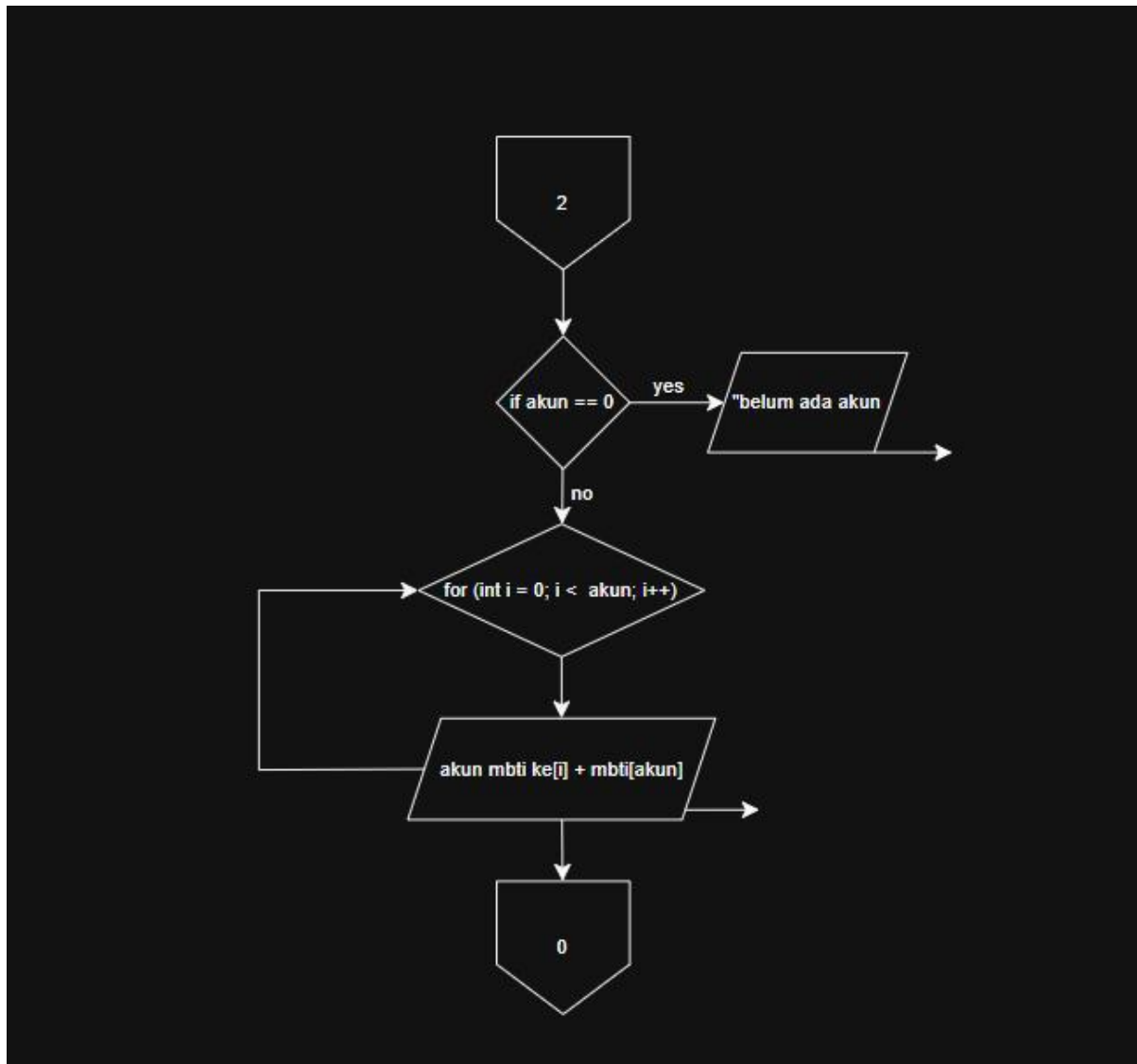
Gambar 1.1 flowchart bagian 1



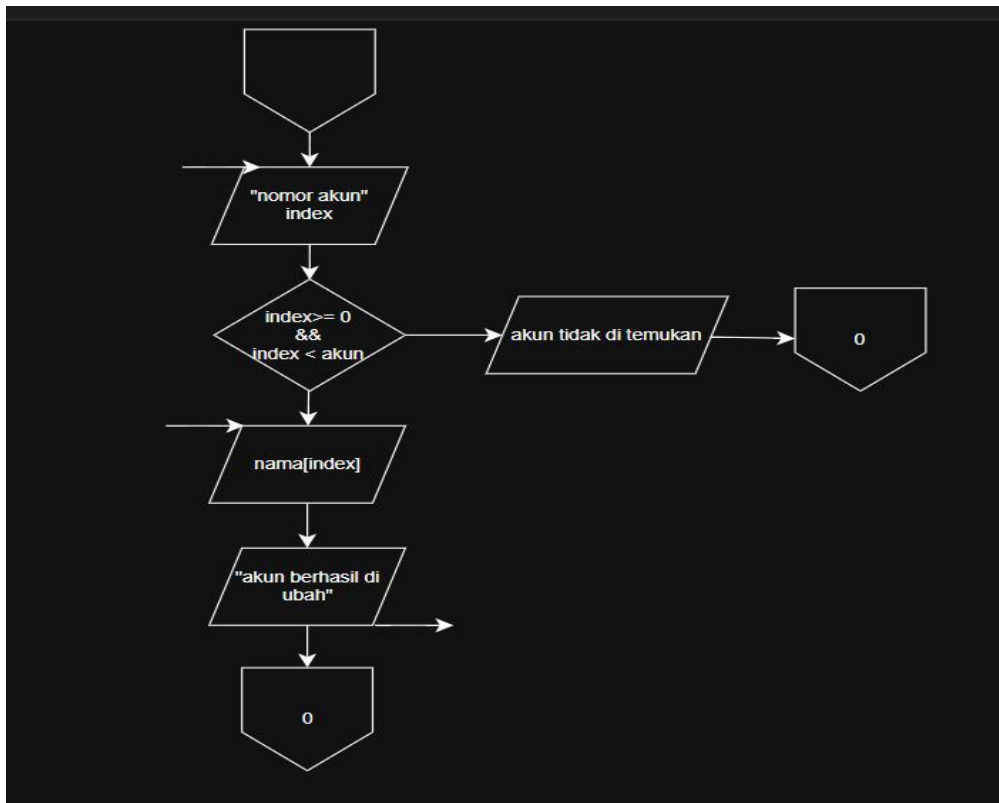
Gambar 1.2 flowchart bagian 2



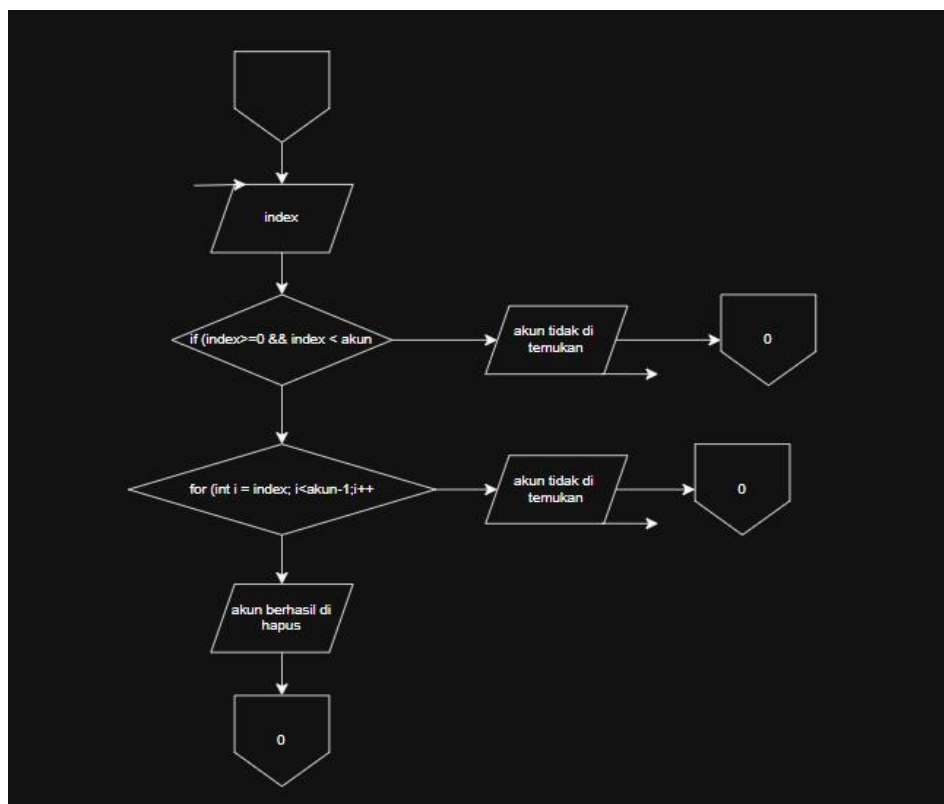
Gambar 1.3 flowchart fitur create



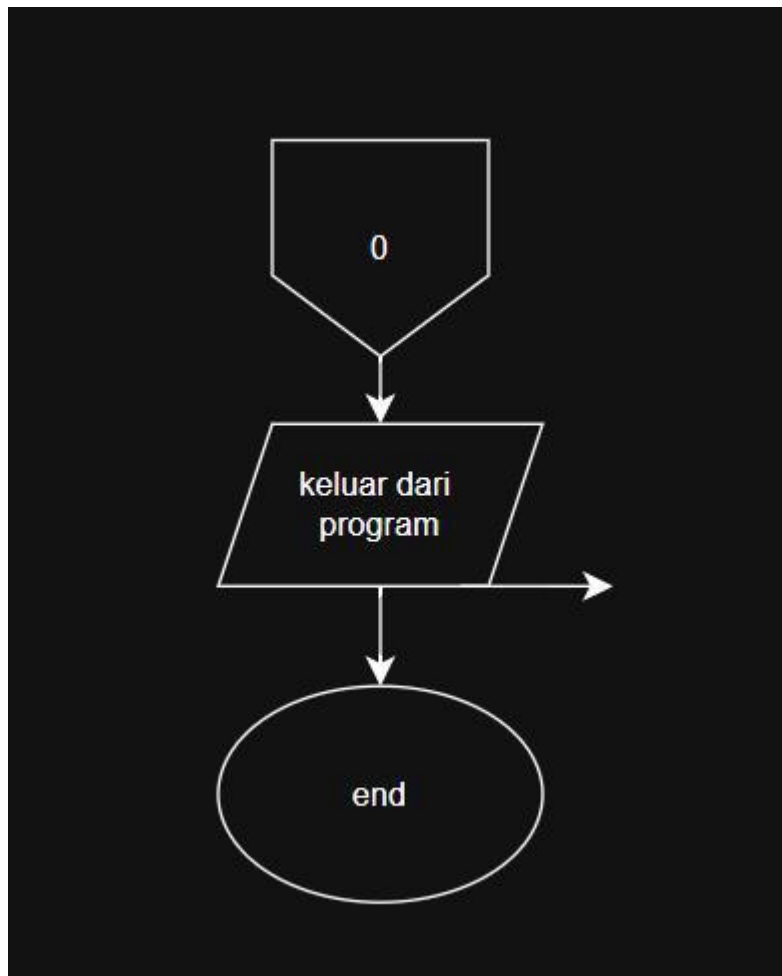
Gambar 1.4 flowchart fitur read



**Gambar 1.5 flowchart fitur update**



**Gambar 1.6 flowchart fitur delete**



**Gambar 1.7 flowchart fitur keluar**

## 2. Analisis Program

Program memiliki tujuan berupa untuk mengikuti suatu tes kecil untuk menentukan apa mbti sang pengguna dengan cara menjawab beberapa soal, pengguna dapat menggunakan multiple akun untuk sekali pakai namun maksimal 5 akun. Selain itu pengguna dapat update, Read dan delete akun dan MBTI yang terpaparkan berdasarkan hasil dari coding

## 3. Source Code

### A. Deklarasi tipe data

Disini tempat dimana setiap bentuk data akan di tampilkan disini, entah itu string, maupun interger, terdapat juga deklarasi array disini.

**SourceCode:**

```
int main() {
    string username = "AjivaAlankSetiandra";
    string password = "017";
    string user, pass;
    int percobaan = 0;
    #define MBTI 5
    int akun = 0;
    string nama[MBTI];
    string mbti[MBTI];
    int pilihan;

    cout << "Selamat datang di program Manajemen MBTI" << endl;
```

**Gambar 3.1 source code deklarasi tipe data**

### B. Fitur login

Fitur ini adalah fitur dimana pengguna harus mengisi password dan username dengan sesuai agar dapat lanjut kedalam program, jika salah sebanyak 3 kali, program akan tutup dengan sendirinya.

**SourceCode:**

```
while (percobaan < 3) {  
    cout << "Masukkan Username: ";  
    cin >> user;  
  
    if (user == username) {  
        cout << "Masukkan Password (NIM): ";  
        cin >> pass;  
  
        if (pass == password) {  
            cout << "Login sukses" << endl;  
        }  
    }  
}
```

**Gambar 3.2 source code fitur login**



### C. Fitur create

Fitur berisi suatu pertanyaan perntanyaan yang dimana pengguna harus menjawab agar pengguna dapat melihat hasil dari MBTI mereka. Berikut berupa salah satu source codenya (tidak memasuki semuanya karena akan terlalu banyak, dan ke 4 pertanyaan memiliki bentuk yang sama namun variabel yang berbeda saja).

Terdapat variabel asing yang bernama (ei,ns,ft,jp) dan (eip, nsp, ftp, jpp), variabel ini berupa singkatan dari Extrovert/Introvert, Intuition/Sensing, Feeling/Thinking, Judgement/perceiving. Untuk yag berakhiran p sama saja, tapi itu menandakan pilihan.

#### SourceCode:

```
if (akun < MBTI) {
    string ei, ns, ft, jp;
    int eip, nsp, ftp, jpp;

    cout << "Masukkan nama Anda: ";
    cin >> nama[akun];
    cout << " " << endl;
    cout << "<----->" << endl;

    cout << "1. Apakah anda suka keluar atau sendiri di kamar? (1/2)" << endl;
    cout << " " << endl;
    cout << "a. Keluar (1)\nb. Di kamar (2)" << endl;
    cout << " " << endl;
    cout << "Jawaban anda = ";
    cin >> eip;
    cout << " " << endl;
    ei = (eip == 1) ? "E" : "I";
    cout << "<----->" << endl;
```

Gambar 3.3 source code fitur create

#### D. Fitur read

Fitur ini berisikan pembacaan akun akun yang telah di ciptakan di fitur create

**SourceCode:**

```
case 2:
    if (akun == 0) {
        cout << "Belum ada akun" << endl;
    } else {
        for (int i = 0; i < akun; i++) {
            cout << "Akun MBTI ke-" << i + 1 << " (" << nama[i] << ") : " << mbti[i] << endl;
        }
    }
    break;
```

**Gambar 3.4 source code fitur read**

#### E. Fitur update

Fitur hanya berisikan fitur yang bertujuan untuk mengubah MBTI dari suatu akun yang ada, jika seandainya pengguna salah mengisi jawaban.

**SourceCode:**

```
case 3: {
    int index;
    cout << "Masukkan nomor akun yang ingin diupdate: ";
    cin >> index;
    index--;

    if (index >= 0 && index < akun) {
        cout << "Masukkan MBTI baru untuk " << nama[index] << ": ";
        cin >> mbti[index];
        cout << "MBTI berhasil diperbarui!" << endl;
    } else {
        cout << "Akun tidak ditemukan" << endl;
    }
    break;
}
```

**Gambar 3.5 source code fitur update**

## F. Fitur delete

Fitur yang berguna untuk menghapus akun akun yang telah di buat, jika tidak ada akun yang terbuat maka program akan kembali ke menu utama.

SourceCode:

```
case 4: {
    int index;
    cout << "Masukkan nomor akun yang ingin dihapus: ";
    cin >> index;
    index--;

    if (index >= 0 && index < akun) {
        for (int i = index; i < akun - 1; i++) {
            nama[i] = nama[i + 1];
            mbti[i] = mbti[i + 1];
        }
        akun--;
        cout << "Akun berhasil dihapus" << endl;
    } else {
        cout << "Akun tidak ditemukan" << endl;
    }
    break;
}
```

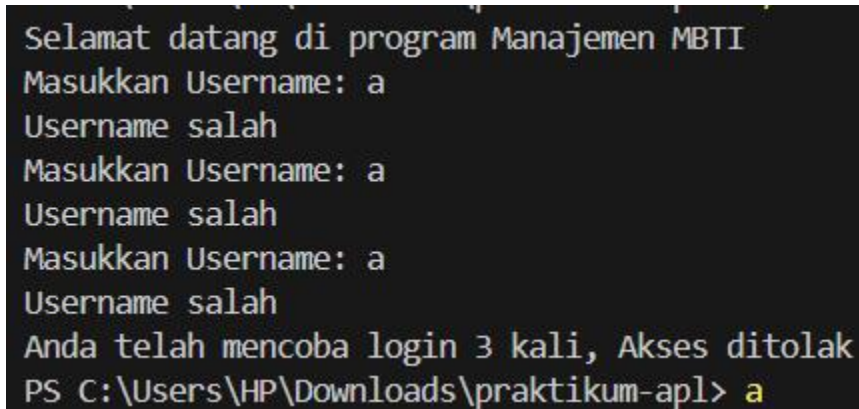
Gambar 3.6 source code fitur delete

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

1. Jika pengguna memasukan password salah sebanyak 3 kali, maka pengguna akan dipaksa berhenti dengan program dengan cara program tidak melanjutkan programnya dan memberi output.
2. Jika pengguna memilih fitur 2,3,4 tanpa memiliki akun, program akan membawa pengguna kembali ke fitur menu utama

### 4.2 Hasil Output



```
Selamat datang di program Manajemen MBTI
Masukkan Username: a
Username salah
Masukkan Username: a
Username salah
Masukkan Username: a
Username salah
Anda telah mencoba login 3 kali, Akses ditolak
PS C:\Users\HP\Downloads\praktikum-apl> a
```

Gambar 4.1 hasil output scenario 1

```

Menu Program
1. Tambah akun MBTI
2. Tampilkan MBTI
3. Update MBTI
4. Hapus akun MBTI
5. Keluar
Pilihan: 2

Belum ada akun

Menu Program
1. Tambah akun MBTI
2. Tampilkan MBTI
3. Update MBTI
4. Hapus akun MBTI
5. Keluar
Pilihan: 3

Masukkan nomor akun yang ingin diupdate: 4
Akun tidak ditemukan

Menu Program
1. Tambah akun MBTI
2. Tampilkan MBTI
3. Update MBTI
4. Hapus akun MBTI
5. Keluar
Pilihan: █

```

**Gambar 4.2 Hasil output scenario**

## **5. Langkah-langkah Git**

### **5.1 Penjelasan**

Pada awalnya buatlah terlebih dahulu repository untuk menampung file file yang akan di taruh disana, kedua buatlah folder untuk tempat menaruh file-filenya.

```

PS C:\Users\HP\Downloads\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/HP/Downloads/praktikum-apl/.git/

```

**Gambar 5.1.1 Git init**

Pergi terminal dari vscode lalu ketik git init (git init adalah suatu command yang dapat di lakukan di terminal untuk menginisiasi repository git.)



```

PS C:\Users\HP\Downloads\praktikum-apl> git add .
PS C:\Users\HP\Downloads\praktikum-apl> git commit -m "Yayyyy selesai #2"
[main cebb916] Yayyyy selesai #2
7 files changed, 261 insertions(+)
create mode 100644 kelas/int angka [5] = {1,2,3,4,5};.cpp
delete mode 100644 post-test/2409106017-AjivaAlankSetiandra-PT-1.exe
rename post-test/{ => PT-1}/2409106017-AjivaAlankSetiandra-PT-1.cpp (100%)
rename post-test/{ => PT-1}/2409106017-AjivaAlankSetiandra-PT-1.pdf (67%)
create mode 100644 post-test/PT-2/2409106017-AjivaAlankSetiandra-PT-2.cpp
create mode 100644 post-test/PT-2/2409106017-AjivaAlankSetiandra-PT-2.exe
create mode 100644 post-test/PT-2/2409106017-AjivaAlankSetiandra-PT-2.pdf

```

**Gambar 5.1.2 Git add dan commit**

lalu ketikan git add (Git add adalah command di terminal yang digunakan untuk menambahkan file apa saja yang akan di commit nantinya), lalu ketik git commit untuk memastikan sebagai checkpoint jika program akan di tandai selesai maupn tidak.

```

PS C:\Users\HP\Downloads\praktikum-apl> git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.10 MiB | 304.00 KiB/s, done.
Total 10 (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/Ajiva-Alank-Setiandra/praktikum-apl.git
   d554fd8..cebb916  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\HP\Downloads\praktikum-apl>

```

**Gambar 5.1.2 Git push**

Git push berisikan suatu command yang dapat membuat semua file apapun yang ada di folder akan masuk ke repository yang sudah disediakan pada awalan.

Ada beberapa langkah yang dilompati dikarenakan sudah terdaptarnya beberapa hal pada git, seperti git remote.