

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

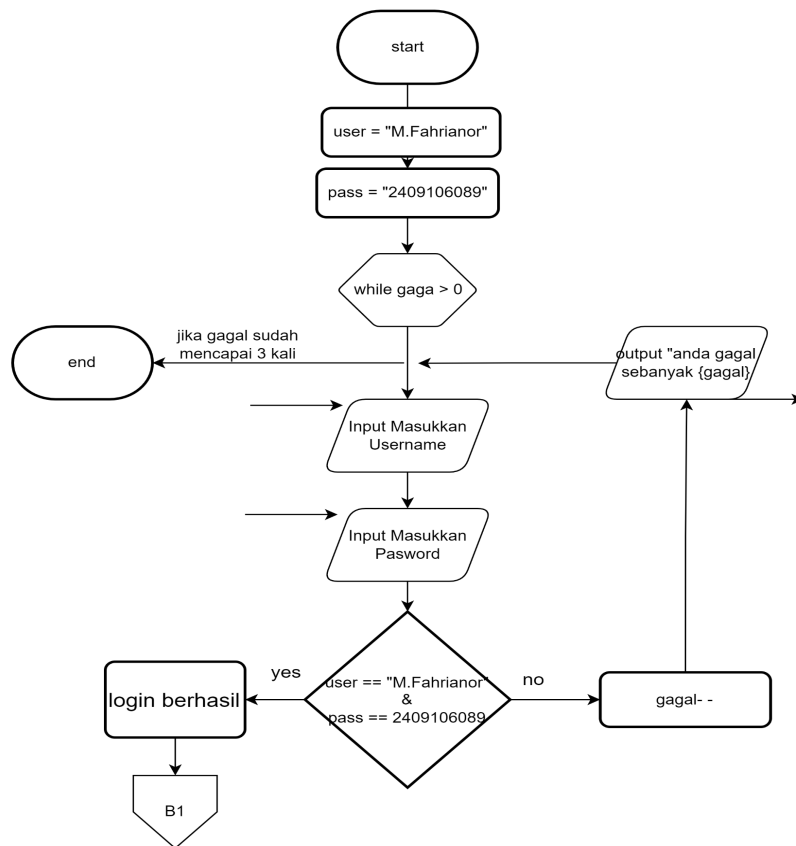


**Disusun oleh:**  
**M.Fahrianor (2409106089)**  
**Kelas (B2'24)**

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

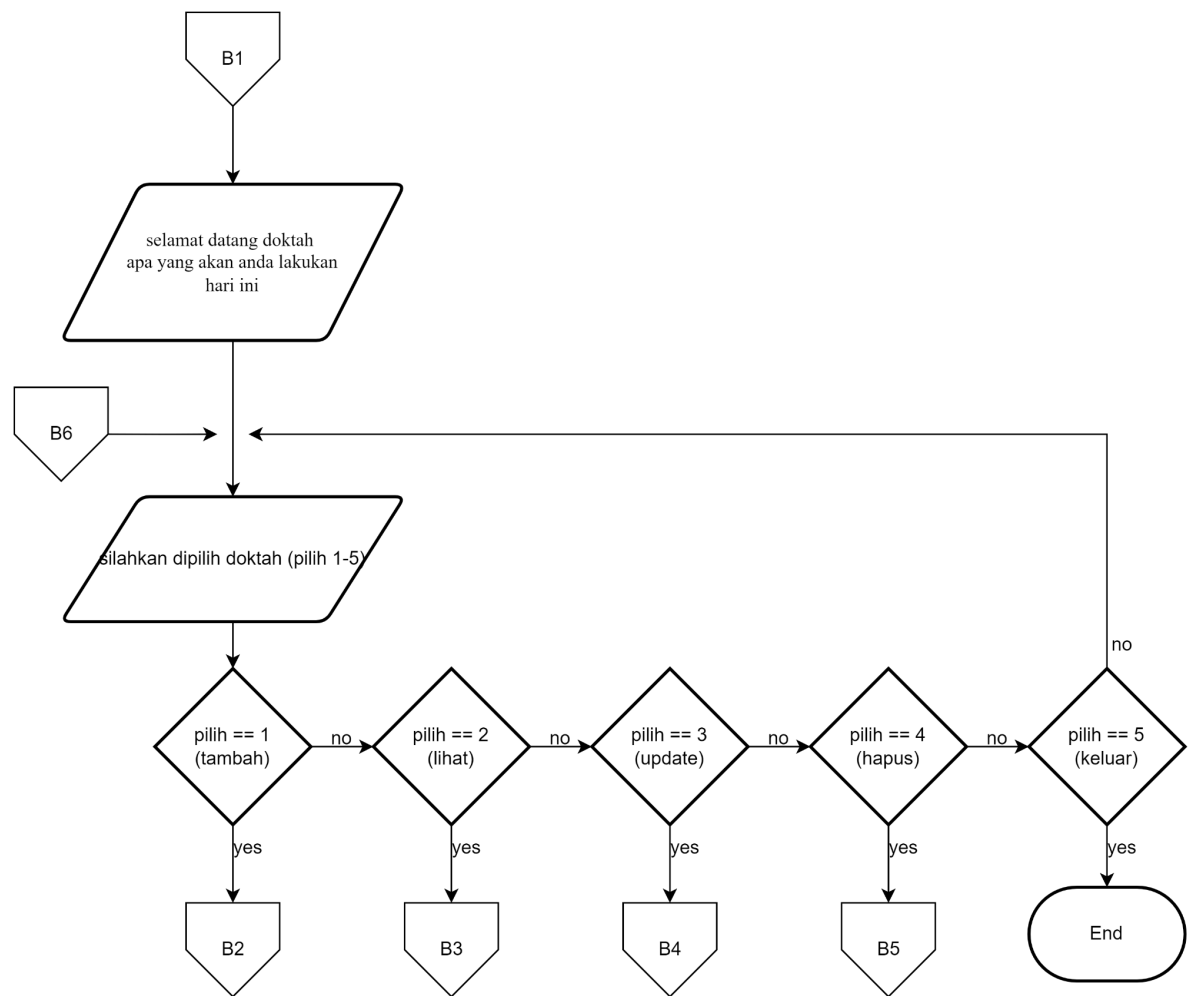
## 1. Flowchart

### A. Login



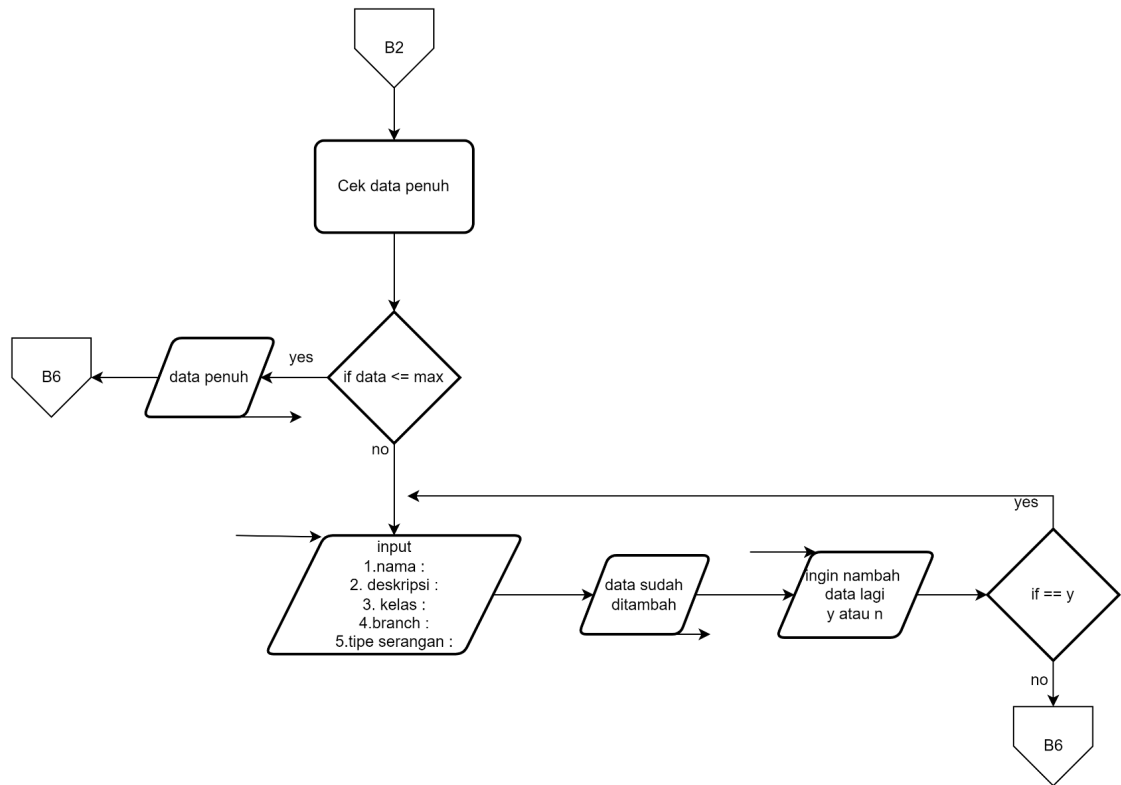
Gambar 1.1 Login

## B. Menu Utama



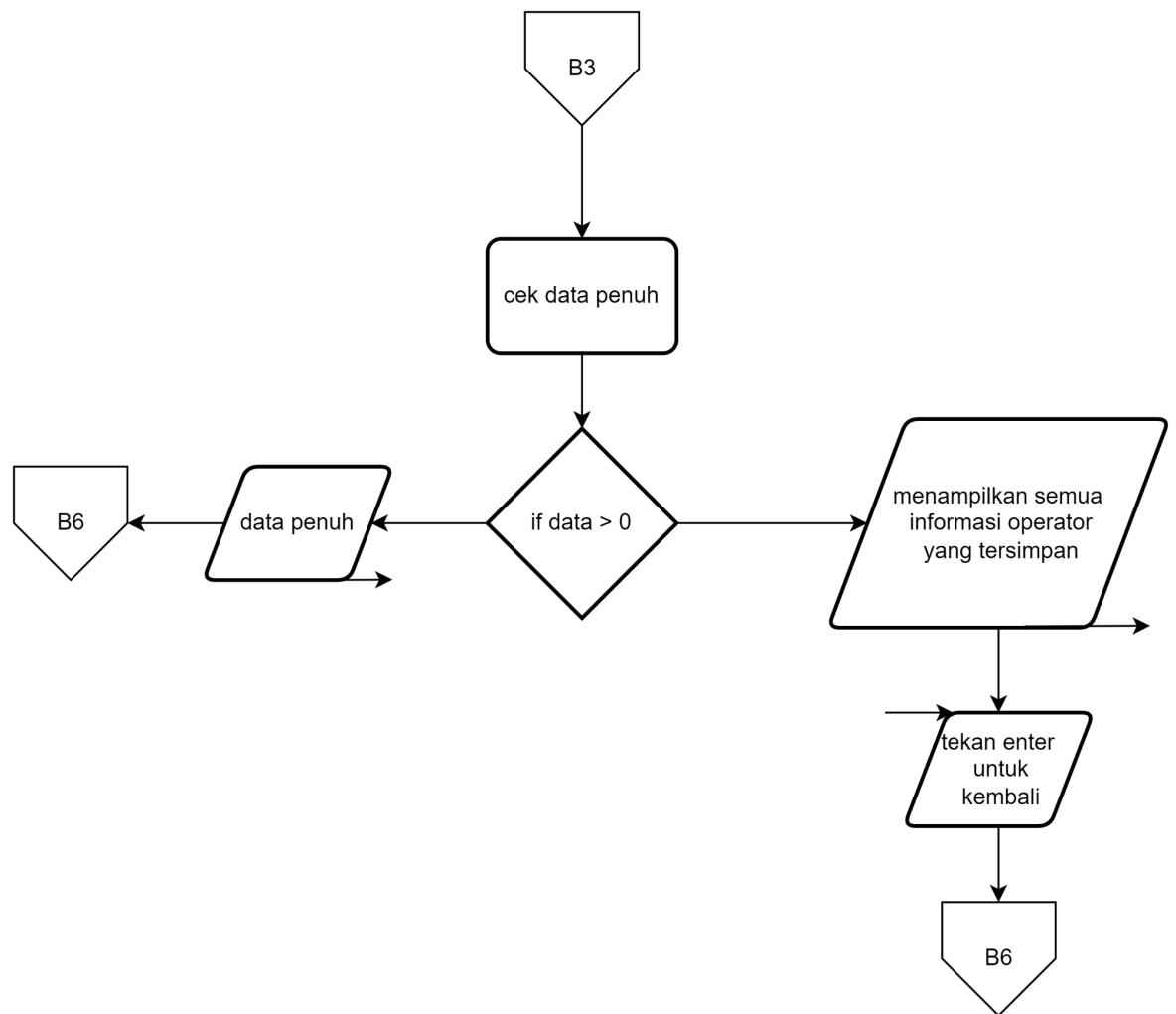
Gambar 1.2 Menu utama

### C. Tambah data



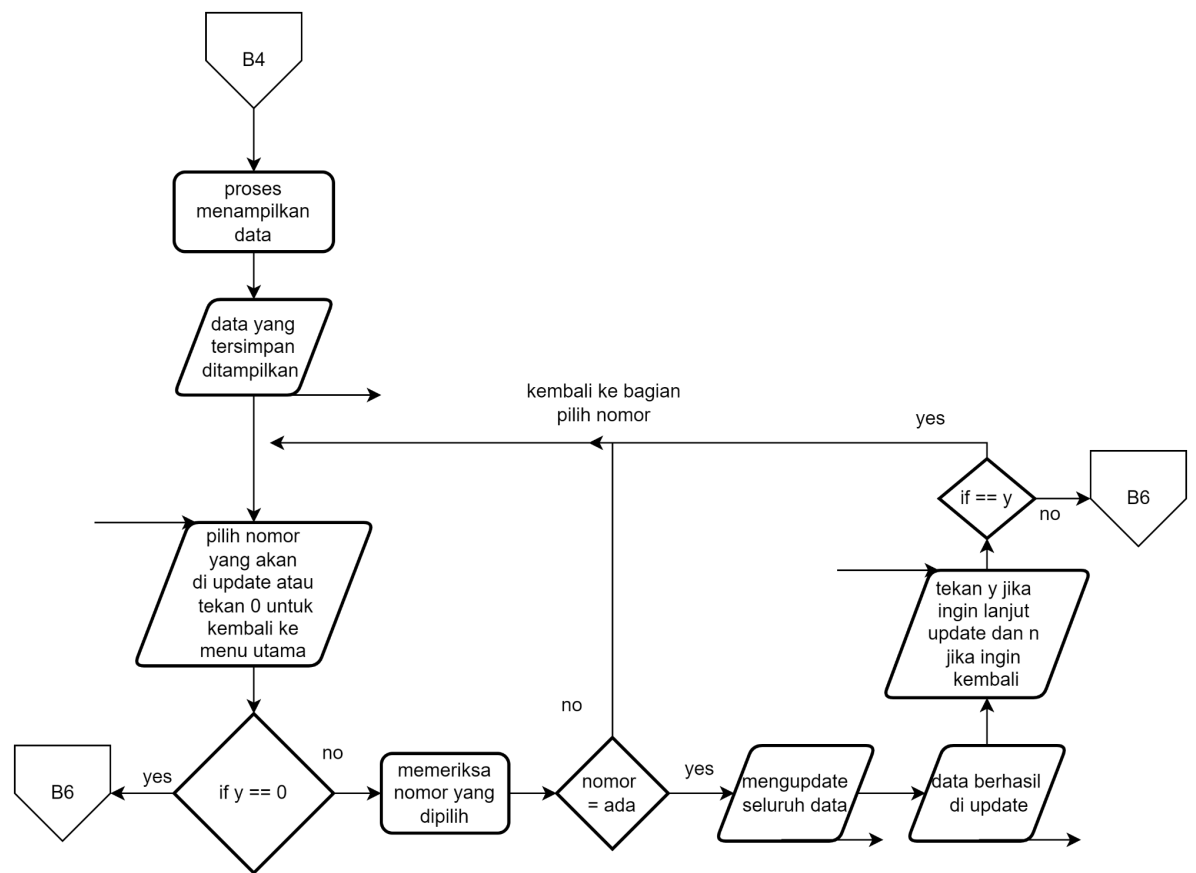
Gambar 1.3 Tambah data

#### D. Lihat data



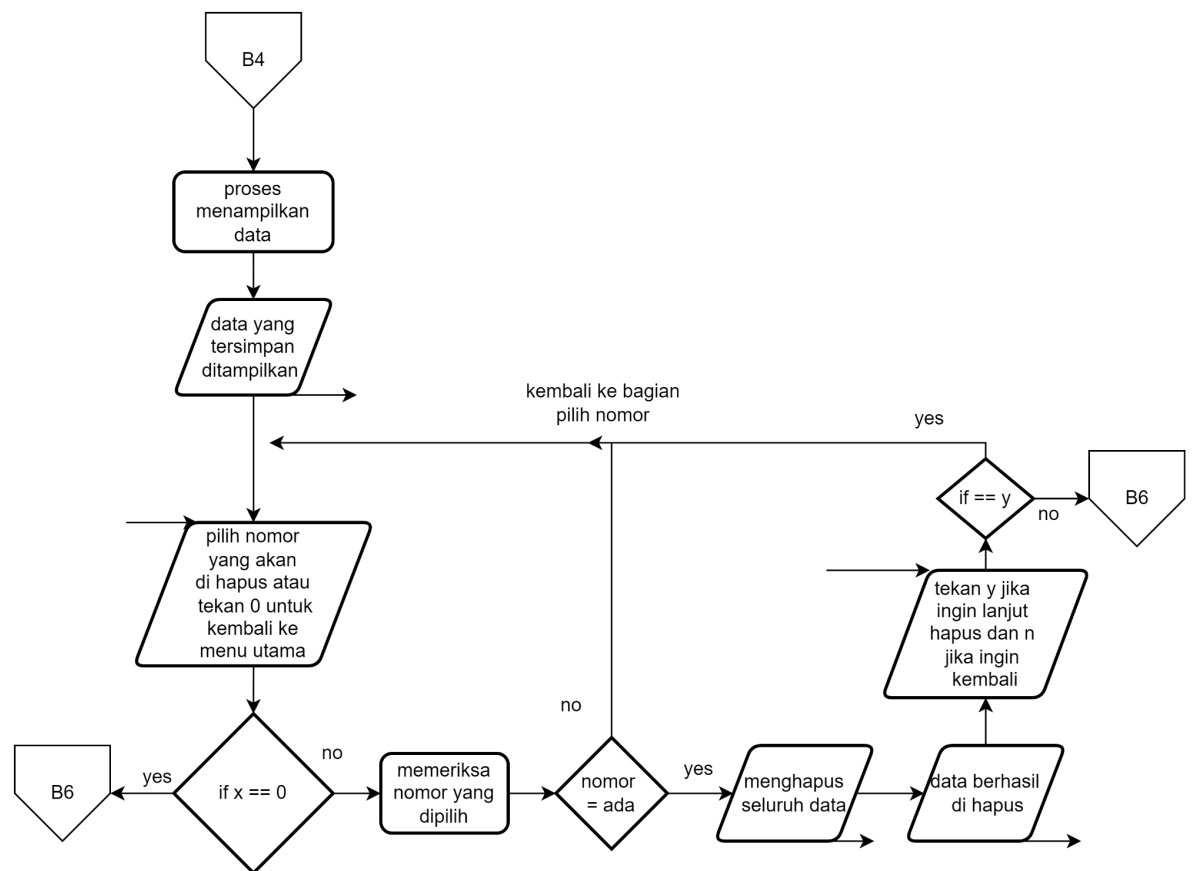
Gambar 1.4 Lihat data

## E. Update data



Gambar 1.5 Update data

## F. Hapus data



Gambar 1.6 Hapus data

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini merupakan program CRUD (create, read, update, delete) yang memiliki judul “Sistem informasi karakter game Arknights”, sistem ini dibuat dengan menggunakan pengelompokan variabel-variabel yaitu struct lebih tepatnya dibuat dengan array of struct yaitu array yang dimana setiap elemennya adalah sebuah struct yang sudah didefinisikan. Didalam program tepatnya sebelum masuk ke menu utama akan dilakukan login yang dimana pengguna akan diminta memasukkan username dan password, dan ketika pengguna salah sebanyak 3 kali maka program akan otomatis berhenti, tetapi jika berhasil pengguna akan

masuk ke dalam menu utama, yang di dalam menu utama ada beberapa pilihan yaitu, tambah data operator, lihat data operator, update data operator, hapus data operator, dan keluar. pada program ini pengguna tidak akan bisa keluar kecuali pengguna memilih opsi keluar.

### 3. Source Code

#### A. Array of Struct

Ini adalah program array of struct

#### Source Code:

```
// maksimal data 20
#define max 20
// struct
struct Operator
{
    string karakter;
    string deskripsi;
    string kelas;
    string branch;
    string tipe;
};
int main(){
    string loop;
    int pos = 3;
    int pilih;
    Operator op[max];
    op[0].karakter = "Eyjafjalla";
    op[0].deskripsi = "Dengan nama asli Adele Nauman, dan seorang putri dari mendiang Katia dan Magna Nauman yaitu pasangan vulcanologist yang terkenal dari Leithanien";
    op[0].kelas = "Caster";
    op[0].branch = "Core caster";
    op[0].tipe = "Arts";

    op[1].karakter = "Ines";
    op[1].deskripsi = "Ines adalah tentara bayaran yang aktif selama perang kazdel, bersama dengan W dan Hoederer dalam satu regu";
    op[1].kelas = "Vanguard";
    op[1].branch = "Agent";
    op[1].tipe = "Physical";
```



```

    op[2].karakter = "Nymph";
    op[2].deskripsi = "Seorang warga kazdel yang bergabung dengan Rhodes
Island, memiliki keahlian dalam originium arts dan bakat alaminya sebagai
djall";
    op[2].kelas = "Caster";
    op[2].branch ="Primal caster";
    op[2].tipe = "Arts";

```

Gambar 3.1 Array of Struct

## B. Fitur Login

Ini adalah fitur login untuk pengguna

### Source Code:

```

// proses login
string username;
string password;
string userbenar = "M.Fahrianor";
string passbenar = "2409106089";
int gagal = 3;

while (gagal > 0) {
    cout << "Masukkan username anda doktah: ";
    cin >> username;
    cout << "Masukkan password anda doktah: ";
    cin >> password;

    if (username == userbenar && password == passbenar) {
        break;
    } else {
        gagal--;
        cout << "" << endl;
        cout << "Kesempatan kamu untuk masuk hanya tersisa " << gagal << ".
Jika kamu gagal Login maka kamu bukanlah doktah" << endl;
    }

    if (gagal == 0) {
        system("cls");
        cout << "=====" << endl;
        cout << "Kamu gagal. Kamu diblokir dari PRTS" << endl;
    }
}

```

```

        cout << "===== " << endl;
        return 0;
    }

```

Gambar 3.2 Fitur Login

### C. Menu Utama

Ini merupakan tampilan menu utama jika pengguna berhasil login

**Source Code:**

```

// menu utama
do
{
    system("cls");
    cout << "===== " << endl;
    cout << "Selamat datang doktah apa yang akan anda lakukan hari ini" <<
endl;

    cout << "===== " << endl;
    cout << "" << endl;
    cout << "===== " << endl;
    cout << "1. Tambah data operator\n";
    cout << "2. Lihat data operator\n";
    cout << "3. Update data operator\n";
    cout << "4. Hapus data operator\n";
    cout << "5. Keluar" << endl;
    cout << "===== " << endl;
    cout << "" << endl;
    cout << "Siahskan dipilih doktah: ";
    cin >> pilih;
    cin.ignore();
}

```

Gambar 3.3 Menu utama

#### D. Tambah Data Operator

Pada gambar ini merupakan tampilan tambah data informasi

**Source Code:**

```
else if (pos < max){
    system("cls");
    cout << "======" << endl;
    cout << "Masukkan nama operator: ";
    getline(cin, op[pos].karakter);
    cout << "Masukkan deskripsi operator: ";
    getline(cin, op[pos].deskripsi);
    cout << "Masukkan class operator: ";
    getline(cin, op[pos].kelas);
    cout << "Masukkan branch operator: ";
    getline(cin, op[pos].branch);
    cout << "Masukkan tipe serangan: ";
    getline(cin, op[pos].tipe);
    cout << "======" << endl;

    pos++;
    cout << "" << endl;
    cout << "Apakah anda ingin menambah operator lagi doktah? (y/n): ";
    cin >> loop;
    cin.ignore();
}
```

Gambar 3.4 Tambah Data

#### E. Lihat Data Operator

Pada gambar ini merupakan tampilan dan proses melihat data

**Source Code:**

```
// proses lihat data
else if (pilih == 2) {
    system("cls");
    if (pos > 0) {
        cout << "Data tersimpan:\n";
        for (int a = 0; a < pos; a++) {
            cout << a + 1 << ". Nama : " << op[a].karakter << endl;
            cout << "    Deskripsi : " << op[a].deskripsi << endl;
            cout << "    Kelas : " << op[a].kelas << endl;
        }
    }
}
```

```

        cout << "    Branch      : " << op[a].branch << endl;
        cout << "    Tipe Serangan: " << op[a].tipe << endl;
        cout << "-----" << endl;
    }
} else {
    cout << "Data kosong\n";
}
cout << "Tekan enter untuk kembali: ";
cin.get();

```

Gambar 3.5 Lihat Data Operator

## F. Update Data Operator

Pada gambar yang pertama merupakan proses dari menampilkan data yang tersimpan tapi hanya berupa nama operator saja dan pilihan untuk kembali atau memilih operator, dan pada gambar yang kedua merupakan proses update data

### Source Code:

```

do
{
    system("cls");
    cout << "===== " << endl;
    cout << "Daftar Operator yang Tersimpan:\n";
    cout << "===== " << endl;
    for (int a = 0; a < pos; a++) {
        cout << a + 1 << ". " << op[a].karakter << endl;
    }

    cout << "===== " << endl;
    cout << "Tekan 0 jika ingin kembali ke menu utama" << endl;
    cout << "===== " << endl;
    cout << "" << endl;
    cout << "\nMasukkan nomor operator yang ingin diupdate: ";
    cin >> y;
    cin.ignore();

    if (y == 0 )
    {
        cout << "Kita kembali ke menu utama doktah\n";
        break;
    } else if (y > 0 && y <= pos)
    {
        system("cls");
    }
}

```

```

        cout << "======" << endl;
        cout << "Update nama operator menjadi: ";
        getline(cin, op[y-1].karakter);
        cout << "Update deskripsi operator menjadi: ";
        getline(cin, op[y-1].deskripsi);
        cout << "Update class operator menjadi: ";
        getline(cin, op[y-1].kelas);
        cout << "Update branch operator menjadi: ";
        getline(cin, op[y-1].branch);
        cout << "Update tipe serangan operator menjadi: ";
        getline(cin, op[y-1].tipe);
        cout << "Data operator berhasil di update silahkan lihat pada
bagian lihat data\n";
        cout << "======" << endl;

```

Gambar 3.6 Update Data Operator

## G. Hapus Data Operator

Pada gambar yang pertama merupakan proses dari menampilkan data yang tersimpan tapi hanya berupa nama operator saja dan pilihan untuk kembali atau memilih operator, dan pada gambar yang kedua merupakan proses hapus data

### Source Code:

```

do
{
    system("cls");
    cout << "======" << endl;
    cout << "Daftar Operator yang Tersimpan:\n";
    cout << "======" << endl;
    for (int a = 0; a < pos; a++) {
        cout << a + 1 << ". " << op[a].karakter << endl;
    }

    cout << "======" << endl;
    cout << "Tekan 0 jika ingin kembali ke menu utama" << endl;
    cout << "======" << endl;
    cout << "" << endl;
    cout << "\nMasukkan nomor operator yang ingin dihapus: ";
    cin >> x;
    cin.ignore();

    if (x == 0)
    {
        cout << "Kita kembali ke menu utama doktah\n";
        break;
    }
}

```

```

}else if (x > 0 && x <= pos)
{
    for (int i = x; i < pos; i++)
    {
        op[i-1].karakter = op[i].karakter;
        op[i-1].deskripsi = op[i].deskripsi;
        op[i-1].kelas = op[i].kelas;
        op[i-1].branch = op[i].branch;
        op[i-1].tipe = op[i].tipe;
    }
    pos--;

    system("cls");
    cout << "=====" << endl;
    cout << "Data operator berhasil dihapus" << endl;
    cout << "=====" << endl;

```

Gambar 3.7 Hapus Data Operator

## H. Log Out

Gambar ini adalah fitur log out

**Source Code:**

```

//proses log out
}else if (pilih == 5)
{
    system("cls");
    cout << "=====" << endl;
    cout << "Selamat tinggal doktah" << endl;
    cout << "=====" << endl;
    break;
}

```

Gambar 3.8 Log Out

## 4. Hasil Output

### 4.1 Hasil Output

```
PS C:\Users\ACER\Documents\praktikum  
Masukkan username anda: M.Fahrianor  
Masukkan password anda: 2409106089
```

Gambar 4.1 Login

```
PS C:\Users\ACER\Documents\praktikum-apl> ./tes15.exe  
Masukkan username anda doktah: M.FA  
Masukkan password anda doktah: tt  
Kesempatan kamu untuk masuk hanya tersisa 2. Jika kamu gagal login maka kamu bukanlah doktah
```

Gambar 4.2 Gagal kurang dari 3 kali

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  
=====  
Kamu gagal. Kamu diblokir dari PRTS  
=====  
PS C:\Users\ACER\Documents\praktikum-apl> |
```

Gambar 4.3 Gagal sudah mencapai 3 kali

```
=====
Selamat datang doktah apa yang akan anda lakukan hari ini
=====

=====
1. Tambah data operator
2. Lihat data operator
3. Update data operator
4. Hapus data operator
5. Keluar
=====

Siajukan dipilih doktah: █
```

Gambar 4.4 Login berhasil dan menu utama

```
=====
Masukkan nama operator: irene
Masukkan deskripsi operator: berasal dari iberia
Masukkan class operator: guard
Masukkan branch operator: swordmaster
Masukkan tipe serangan: pyhsical
=====

Apakah anda ingin menambah operator lagi doktah? (y/n): █
```

Gambar 4.5 Tambah data



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

=====
Daftar Operator yang Tersimpan:
=====
1. Eyjafjalla
2. Ines
3. Nymph
4. Lin Yuxia
=====
Tekan 0 jika ingin kembali ke menu utama
=====

Masukkan nomor operator yang ingin diupdate: █
```

```
=====
Update nama operator menjadi: irene
Update deskripsi operator menjadi: berasal dari iberia
Update class operator menjadi: Guard
Update branch operator menjadi: Swordmaster
Update tipe serangan operator menjadi: Pyhsical
Data operator berhasil di update silahkan lihat pada bagian lihat data
=====

Apakah doktah ingin update operator lagi? (y/n): █
```

Gambar 4.6 Menu update dan hasil update

```
=====
Daftar Operator yang Tersimpan:
=====
1. Eyjafjalla
2. Ines
3. Nymph
4. irene
=====
Tekan 0 jika ingin kembali ke menu utama
=====

Masukkan nomor operator yang ingin dihapus: █
```

```
=====
Data operator berhasil dihapus
=====

Apakah doktah ingin menghapus lagi operator lagi? (y/n):
```

Gambar 4.7 Menu hapus dan hasil hapus

```
=====
Selamat tinggal doktah
=====
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 4.8 Log out

## 5. Langkah-langkah Git

### A. Git init

Perintah git init digunakan untuk menginisialisasi repository git dalam sebuah folder, ini berarti folder tersebut akan diubah menjadi repository git.

```
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> git init
Initialized empty Git repository in C:/Users/ACER/Documents/praktikum-apl/post-test/post-test-3/.git/
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> █
```

Gambar 5.1 Git init

### B. Git add

Perintah git add digunakan untuk menambahkan file apa saja sebelum dilakukannya commit.

```
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> git add .
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> █
```

Gambar 5.2 Git add

### C. Git commit

Perintah git commit digunakan untuk menyimpan perubahan yang telah di tambahkan ke dalam repository git

```
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> git commit -m 'finish post test 3'
[master (root-commit) 2ffd30a] finish post test 3
2 files changed, 273 insertions(+)
create mode 100644 2409106089-M.Fahrianor-PT-3.cpp
create mode 100644 2409106089-M.Fahrianor-PT-3.exe
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> █
```

Gambar 5.3 Git commit

#### D. Git remote add origin

Perintah git ini digunakan untuk menghubungkan antara repository lokal dengan github

```
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> git remote add origin https://github.com/Garaku-kuyashi/praktikum_apl.git
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3>
```

Gambar 5.4 Git remote add origin

#### E. Git push

perintah git push digunakan untuk mengunggah perubahan dari repository lokal ke github.

```
PS C:\Users\ACER\Documents\praktikum-apl> git push -u origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (17/17), 906.61 KiB | 2.16 MiB/s, done.
Total 17 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 2 local objects.
To https://github.com/Garaku-kuyashi/praktikum_apl.git
 1f466b8..cf92101  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.5 Git push