

LAPORAN PRAKTIKUM
POSTTEST (4)
ALGORITMA PEMROGRAMAN LANJUT

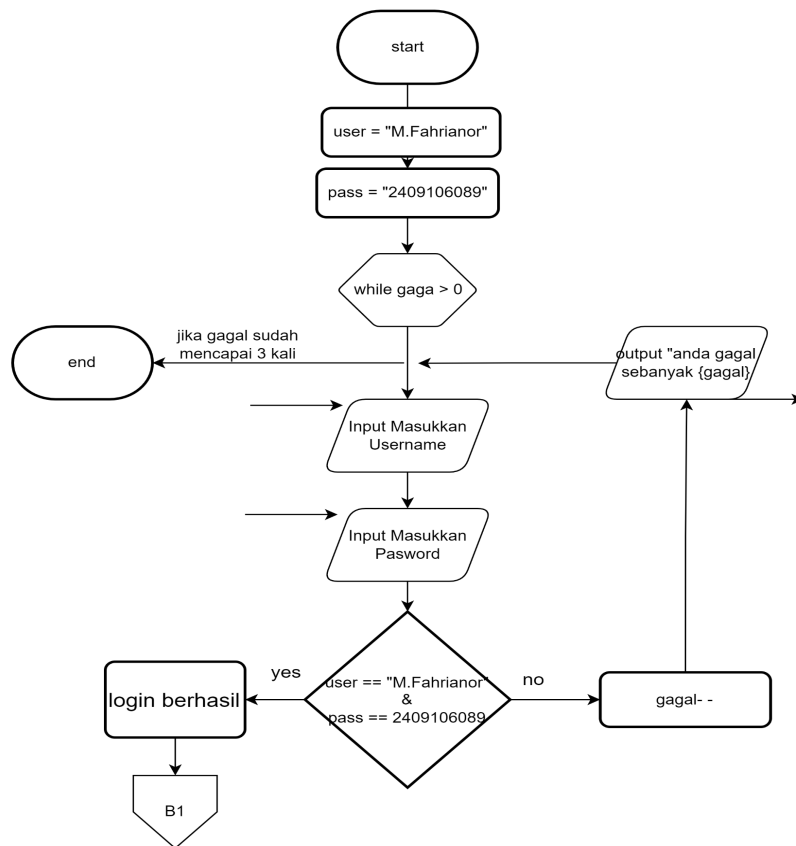


Disusun oleh:
M.Fahrianor (2409106089)
Kelas (B2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

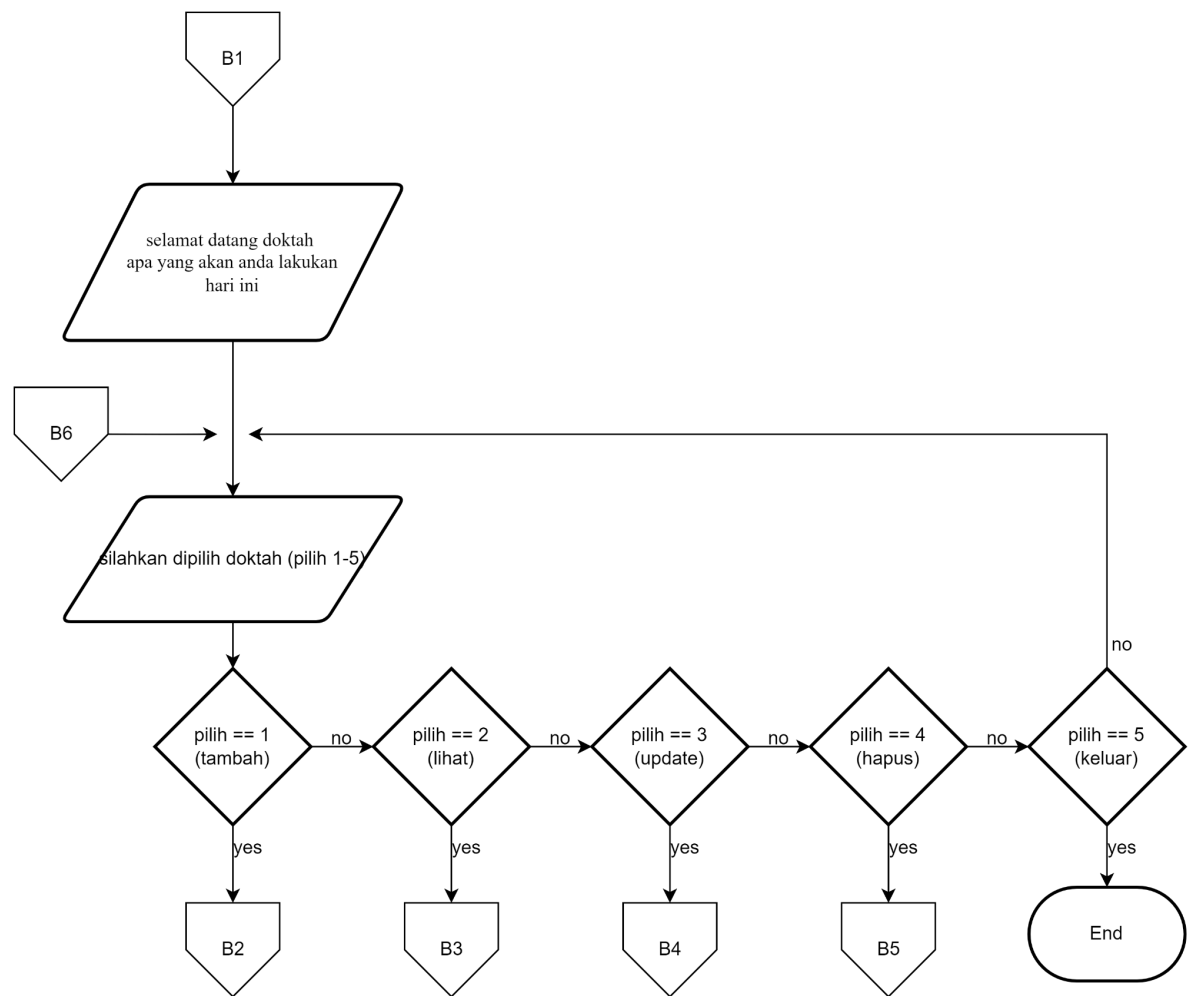
1. Flowchart

A. Login



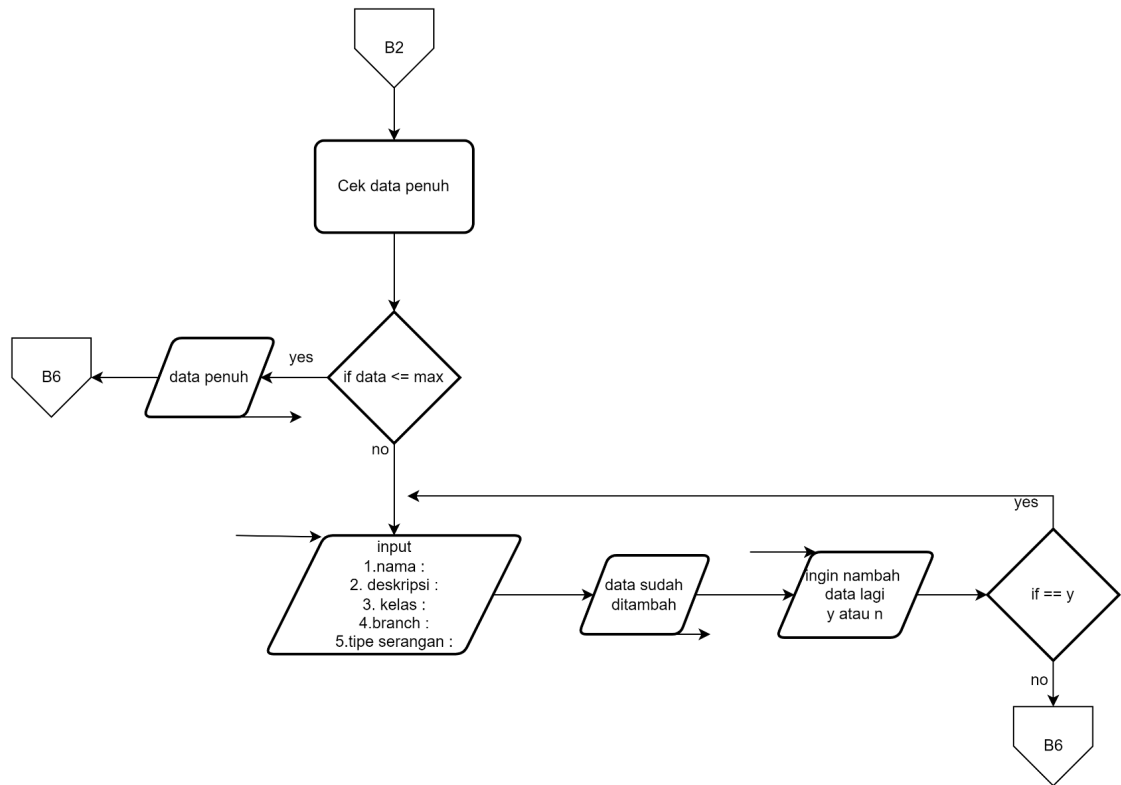
Gambar 1.1 Login

B. Menu Utama



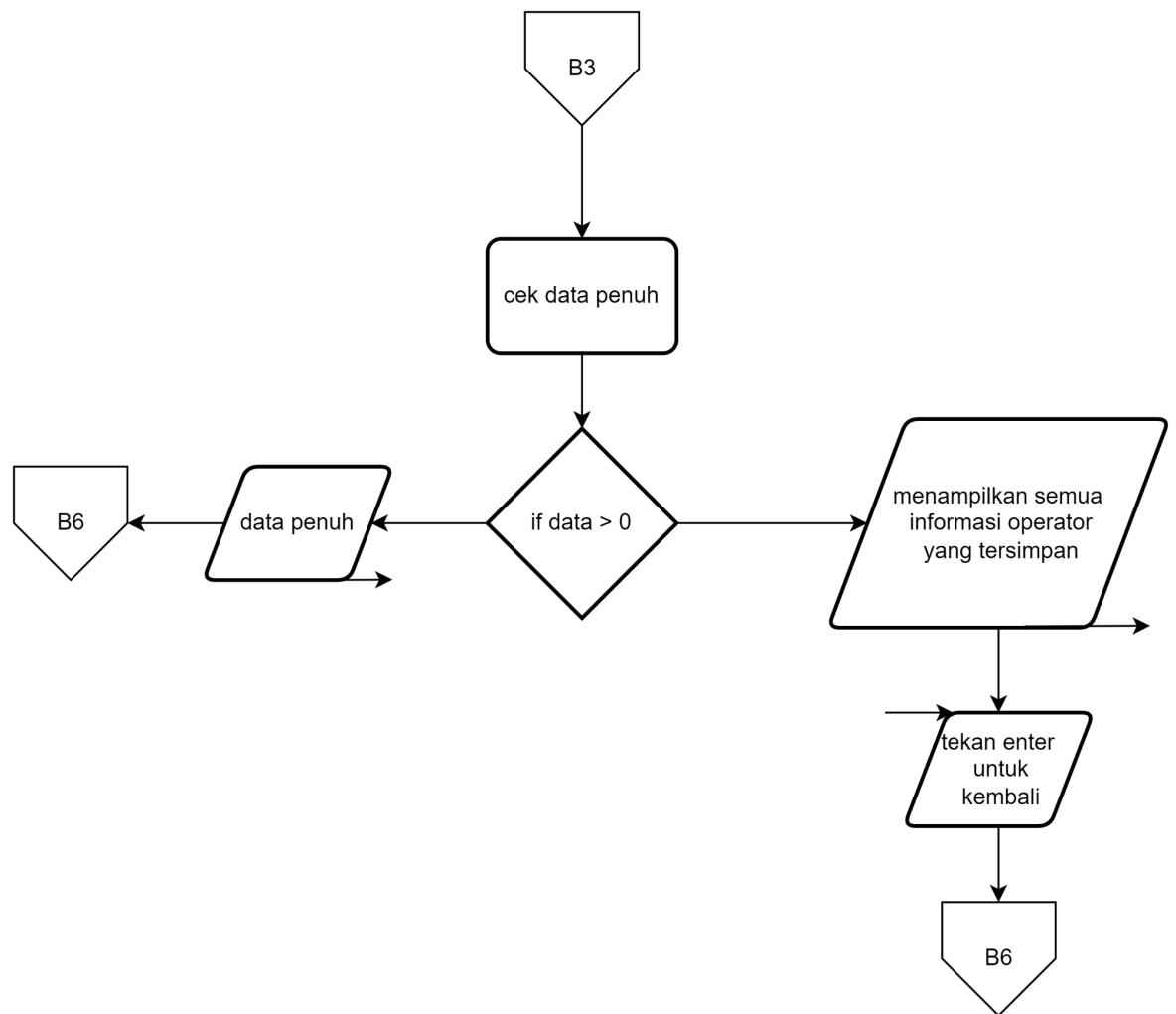
Gambar 1.2 Menu utama

C. Tambah data



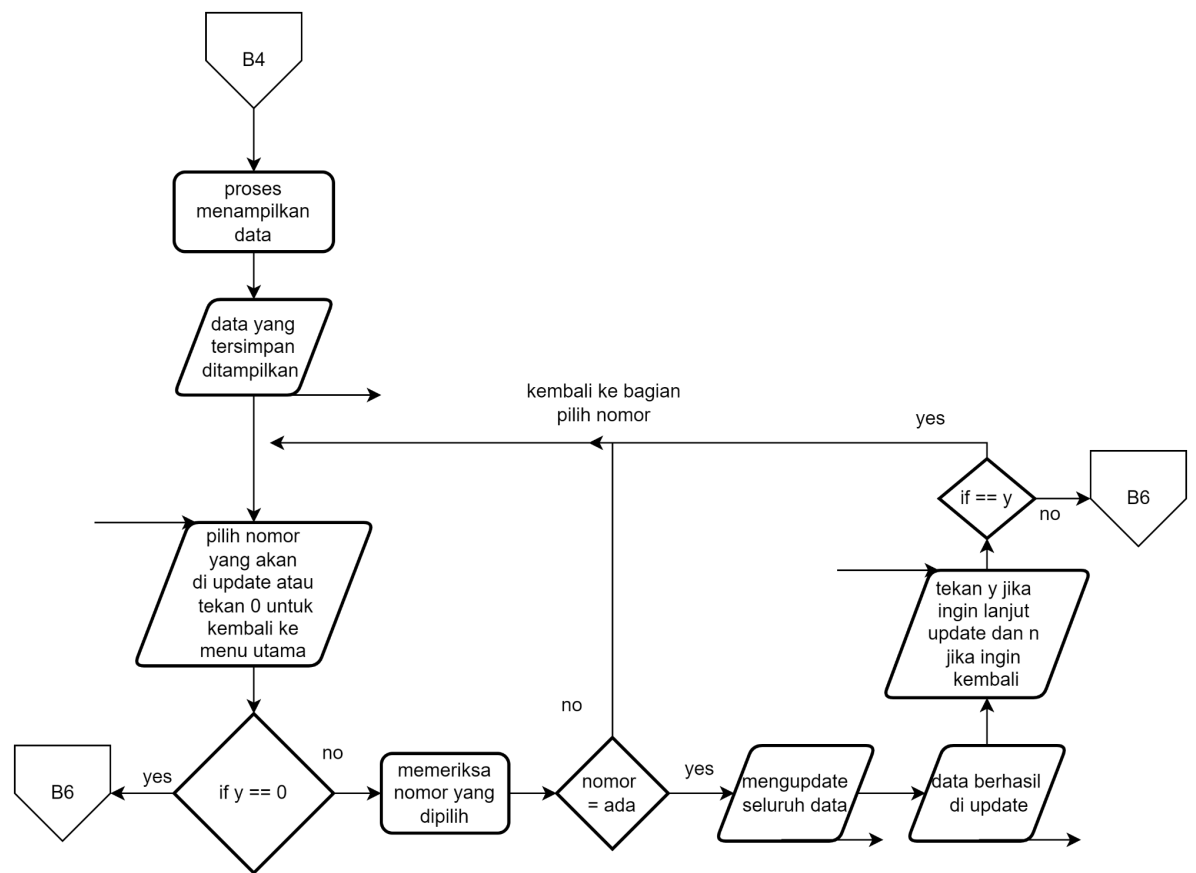
Gambar 1.3 Tambah data

D. Lihat data



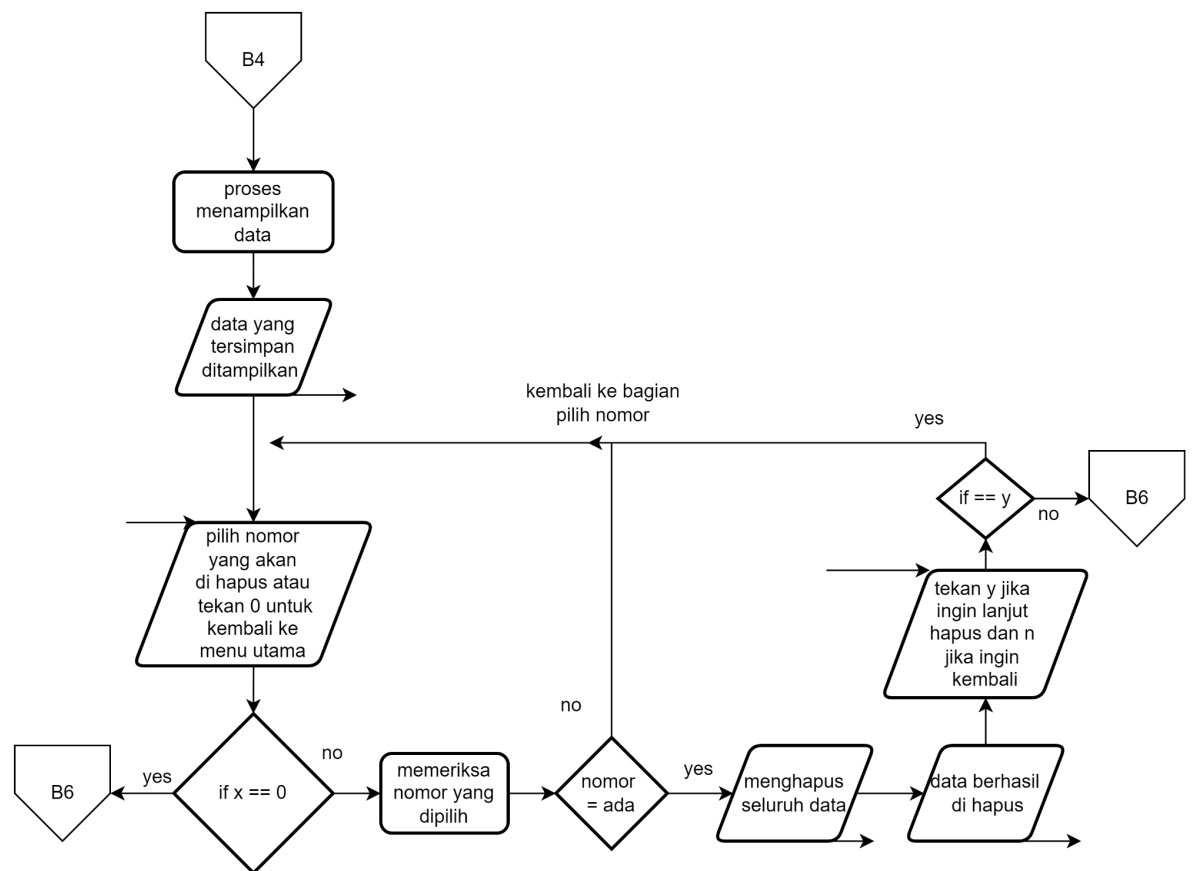
Gambar 1.4 Lihat data

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 fungsi, parameter, dan prosedur yang dimana pada program ini akan terdapat 5 sub program yang dibuat menggunakan fungsi. 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. Struct dan Data awal

Ini adalah program array of struct

Source Code:

```
#define max 20
int pos = 3;

// struct
struct Operator
{
    string karakter;
    string deskripsi;
    string kelas;
    string branch;
    string tipe;
};
Operator op[max];
void dataawal(){
    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";
}
```

Gambar 3.1 Struct dan Data awal

B. Fitur Login

Ini adalah fitur login untuk pengguna

Source Code:

```
bool Login(string userbenar, string passbenar) {
    string username;
    string password;
    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) {
            return true;
        }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;
        }
    }
    return false;
}
```

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 << "Siahhkan dipilih doktah: ";
    cin >> pilih;
    cin.ignore();
}
```

Gambar 3.3 Menu utama

D. Tambah Data Operator

Pada gambar ini merupakan tampilan tambah data informasi yang menggunakan prosedur dan parameter di dalamnya

Source Code:

```
void tambahdata(Operator op[], int &pos){
    string loop;
    system("cls");
    do
    {
        if (pos >= max)
        {
            system("cls");
            cout << "===== " << endl;
        }
    }
}
```

```

        cout << "Data sudah max doktah\n";
        cout << "=====" << endl;
        cout << "" << endl;
        cout << "Tekan enter untuk kembali: ";
        cin.get();
        break;
    }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();
    }

    } while (loop == "y");
}

```

Gambar 3.4 Tambah Data

E. Lihat Data Operator

Pada gambar ini merupakan tampilan dan proses melihat data, sama seperti tambah data lihat data juga menggunakan prosedur dan parameter di dalamnya

Source Code:

```
void Lihatope(Operator op[], int &pos){
    string loop;
    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 ini merupakan fitur update data yang dimana saat memilih fitur ini di dalam menu maka yang pertama akan dilakukan sistem adalah menampilkan data yang tersimpan dan kemudian pengguna dapat memilih data nomor berapa yang akan di update.

Source Code:

```
void updateope(Operator op[], int pos){
    string loop;
    if (pos > 0)
```

```

{
    int y;
    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;
        } else {
            cout << "======" << endl;
            cout << "Data operator tidak ada mohon untuk memasukkan
nomor yang benar" << endl;
            cout << "======" << endl;
        }
        cout << "" << endl;
        cout << "Apakah doktah ingin update operator lagi? (y/n): ";
        cin >> loop;
    } while (loop == "y");
}
}

```

;

Gambar 3.6 Update Data Operator

G. Hapus Data Operator

pada gambar ini merupakan fitur hapus data yang dimana saat memilih fitur ini di dalam menu maka yang pertama akan dilakukan sistem adalah menampilkan data yang tersimpan dan kemudian pengguna dapat memilih data nomor berapa yang akan di hapus

Source Code:

```
void hapusope(Operator op[], int &pos){
    string loop;
    if (pos > 0)
    {
        int x;

        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;

    }else{
        cout << "======" << endl;
        cout << "Data operator tidak ada mohon untuk memasukkan
nomor yang benar"<< endl;
        cout << "======" << endl;
    }
    cout << "" << endl;
    cout << "Apakah doktah ingin menghapus lagi operator lagi?
(y/n): ";

    cin >> loop;
    } while (loop == "y");
    }
}

```

Gambar 3.7 Hapus Data Operator

H. Log Out

Gambar ini adalah fitur log out

Source Code:

```

void logout(){
    system("cls");
    cout << "======" << endl;
    cout << "Selamat tinggal doktah"<< endl;
    cout << "======" << endl;
}

```

Gambar 3.8 Log Out

J. Penulisan pada menu utama

Gambar ini adalah tampilan dari penulisan prosedur yang sudah dibuat di luar int main

Source Code:

```
if (pilih == 1)
{
    tambahdata(op, pos);
}else if (pilih == 2)
{
    lihatope(op, pos);
}else if (pilih == 3)
{
    updateope(op, pos);
}else if (pilih == 4)
{
    hapusope(op, pos);
}else if (pilih == 5)
{
    logout();
    break;
}
```

Gambar 3.9 Penulisan pada menu utama

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

```
Masukkan username anda doktah: M.F  
Masukkan password anda doktah: ss  
  
Kesempatan kamu untuk masuk hanya tersisa 2. Jika kamu gagal login maka kamu bukanlah doktah  
Masukkan username anda doktah: s
```

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: Priestess
Masukkan deskripsi operator: Priestess was considered the creator and likened to a god, but she considered Doctor to be the genius behind its creation
Masukkan class operator: ??????
Masukkan branch operator: ???
Masukkan tipe serangan: ???
=====

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

Gambar 4.5 Tambah data

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

Masukkan nomor operator yang ingin diupdate: █
```

```
=====
Update nama operator menjadi: Mon3tr
Update deskripsi operator menjadi: We're all familiar with this name, but we need to get to know her again
Update class operator menjadi: Medic
Update branch operator menjadi: Chain
Update tipe serangan operator menjadi: Arts
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

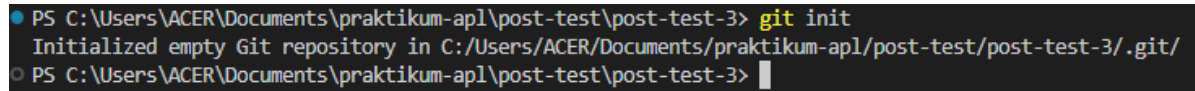
```
=====
Selamat tinggal doktah
=====
PS C:\Users\ACER\Documents\praktikum-apl> 
```

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.

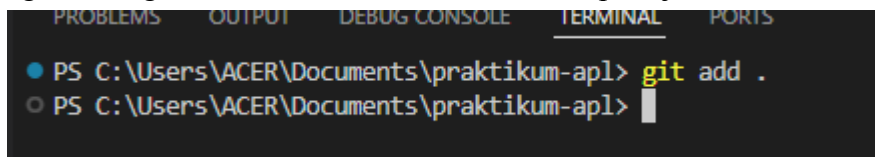


```
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3> git init
Initialized empty Git repository in C:/Users/ACER/Documents/praktikum-apl/post-test/post-test-3/.git/
PS C:\Users\ACER\Documents\praktikum-apl\post-test\post-test-3>
```

Gambar 5.1 Git init

B. Git add

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

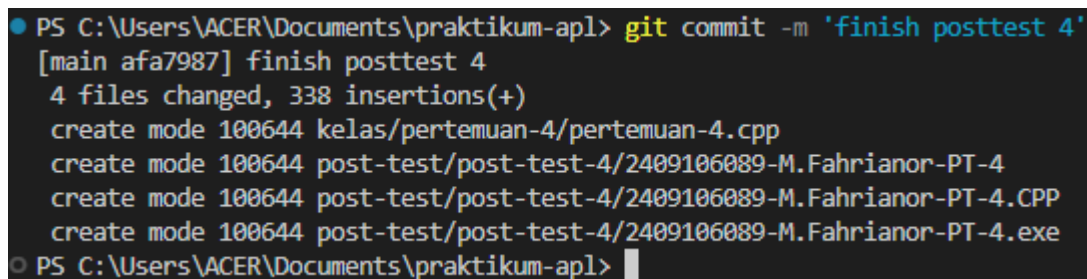


```
PS C:\Users\ACER\Documents\praktikum-apl> git add .
PS C:\Users\ACER\Documents\praktikum-apl>
```

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> git commit -m 'finish posttest 4'
[main afa7987] finish posttest 4
4 files changed, 338 insertions(+)
create mode 100644 kelas/pertemuan-4/pertemuan-4.cpp
create mode 100644 post-test/post-test-4/2409106089-M.Fahrianor-PT-4
create mode 100644 post-test/post-test-4/2409106089-M.Fahrianor-PT-4.CPP
create mode 100644 post-test/post-test-4/2409106089-M.Fahrianor-PT-4.exe
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.3 Git commit

D. Git remote add origin

Perintah git ini digunakan untuk menghubungkan antara repository lokal dengan github, tetapi pada gambar tertulis “error: remote origin already exists” yang artinya repository lokal dengan github sudah terhubung.

```
Create mode 100044 post-test/post-test-4/z409100089-M.Panrianor-Pr4.exe
PS C:\Users\ACER\Documents\praktikum-apl> git remote add origin https://github.com/Garaku-kuyashi/praktikum_apl.git
error: remote origin already exists.
PS C:\Users\ACER\Documents\praktikum-apl>
```

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: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 888.83 KiB | 2.15 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Garaku-kuyashi/praktikum_apl.git
 d35f21f..afa7987 main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.5 Git push