

LAPORAN PRAKTIKUM
POSTTEST (2)
ALGORITMA PEMROGRAMAN LANJUT

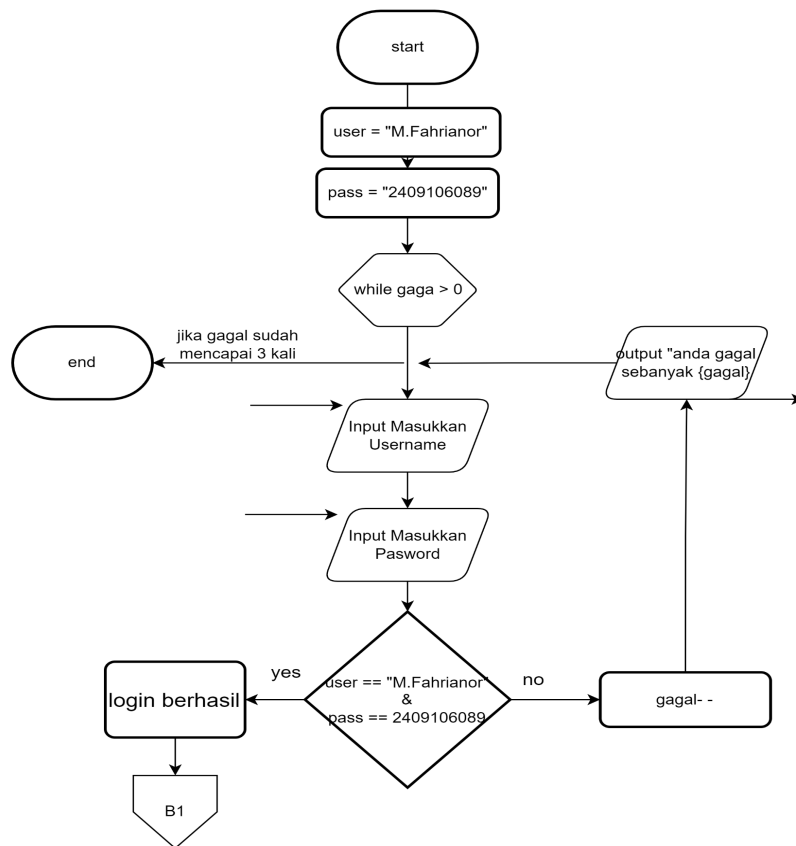


Disusun oleh:
M.Fahrianor (2409106089)
Kelas (B2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

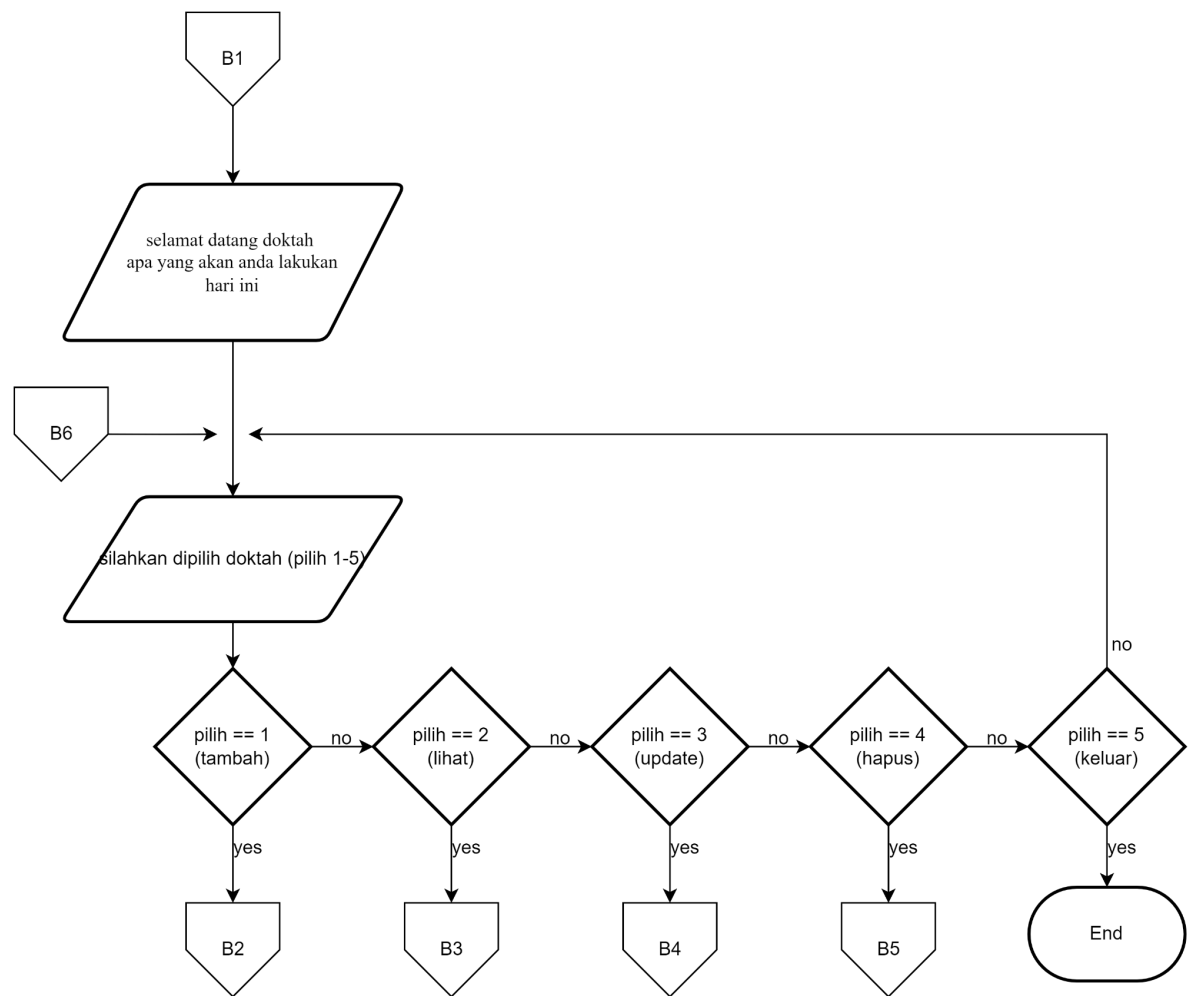
1. Flowchart

A. Login



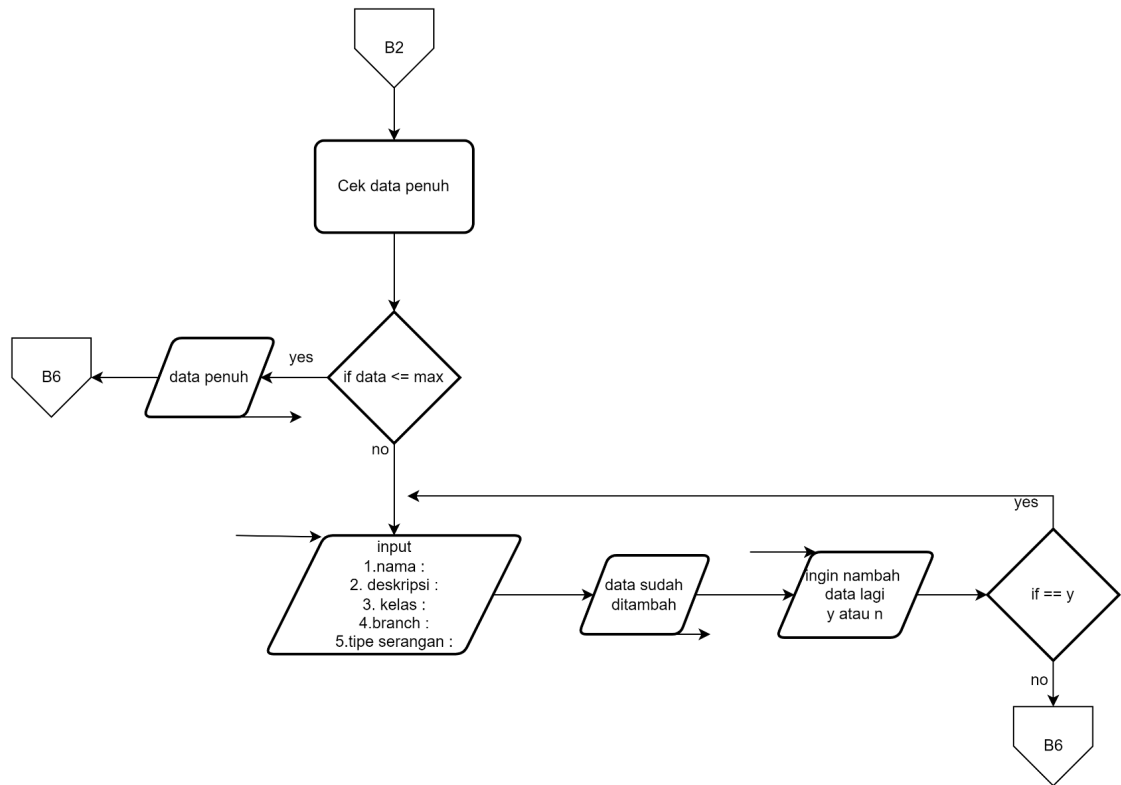
Gambar 1.1 Login

B. Menu Utama



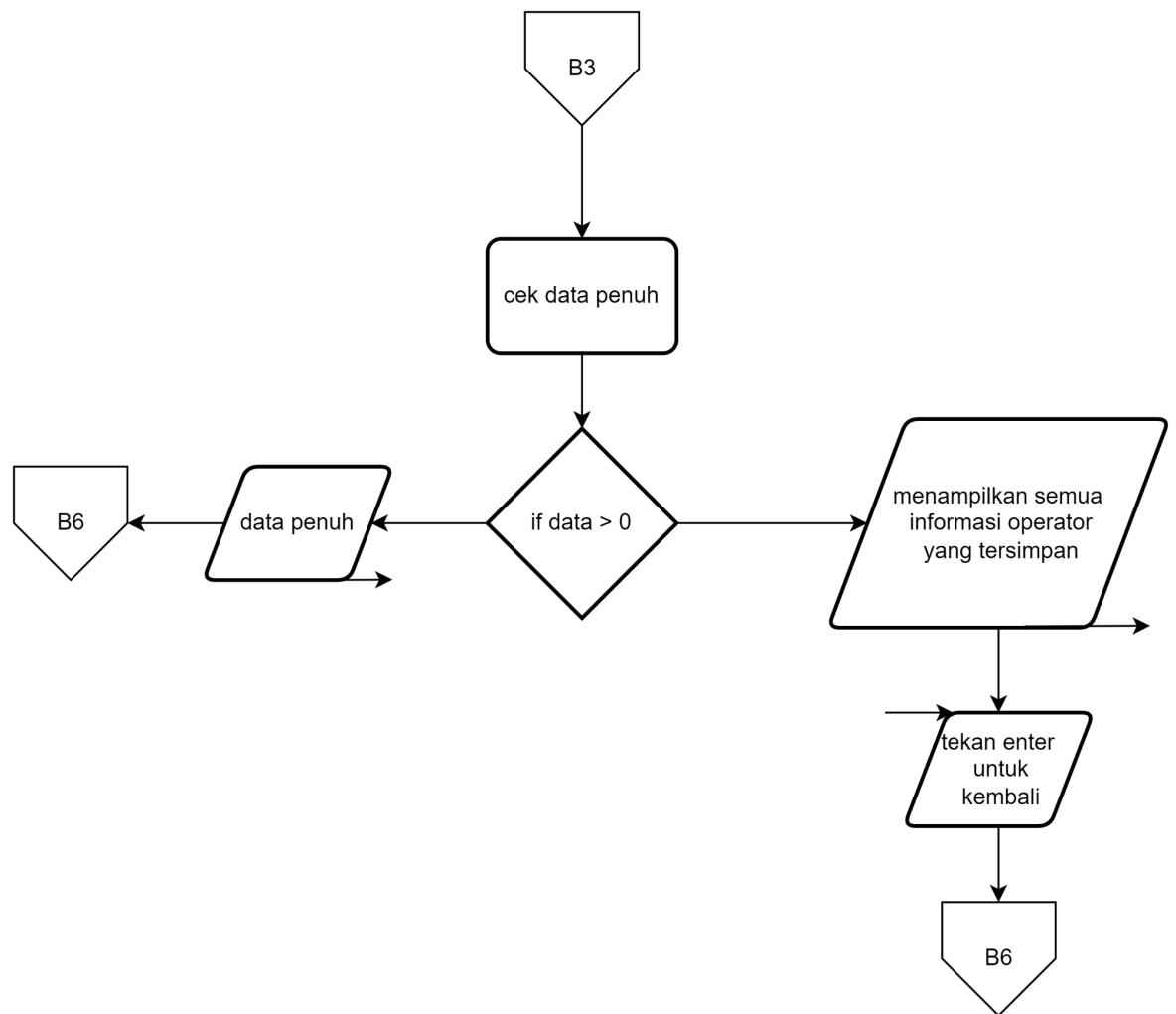
Gambar 1.2 Menu utama

C. Tambah data



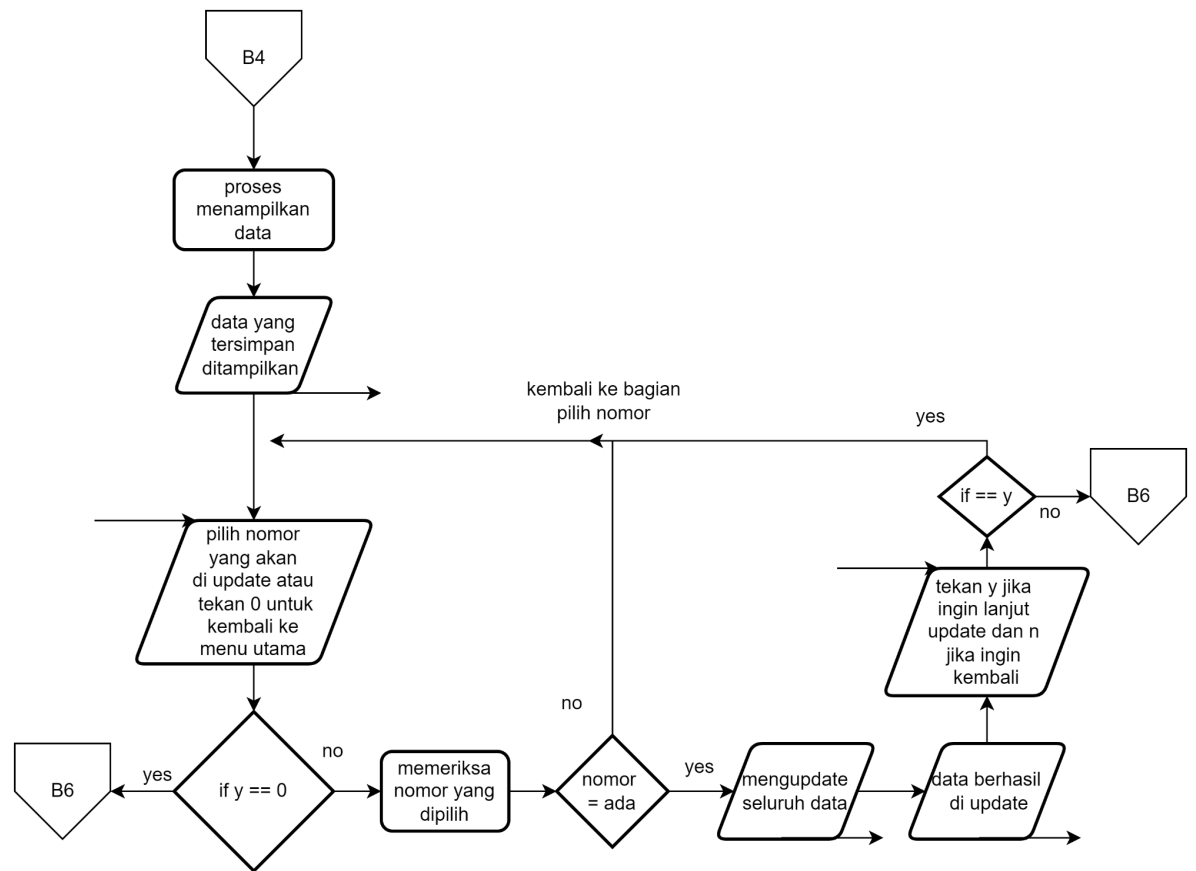
Gambar 1.3 Tambah data

D. Lihat data



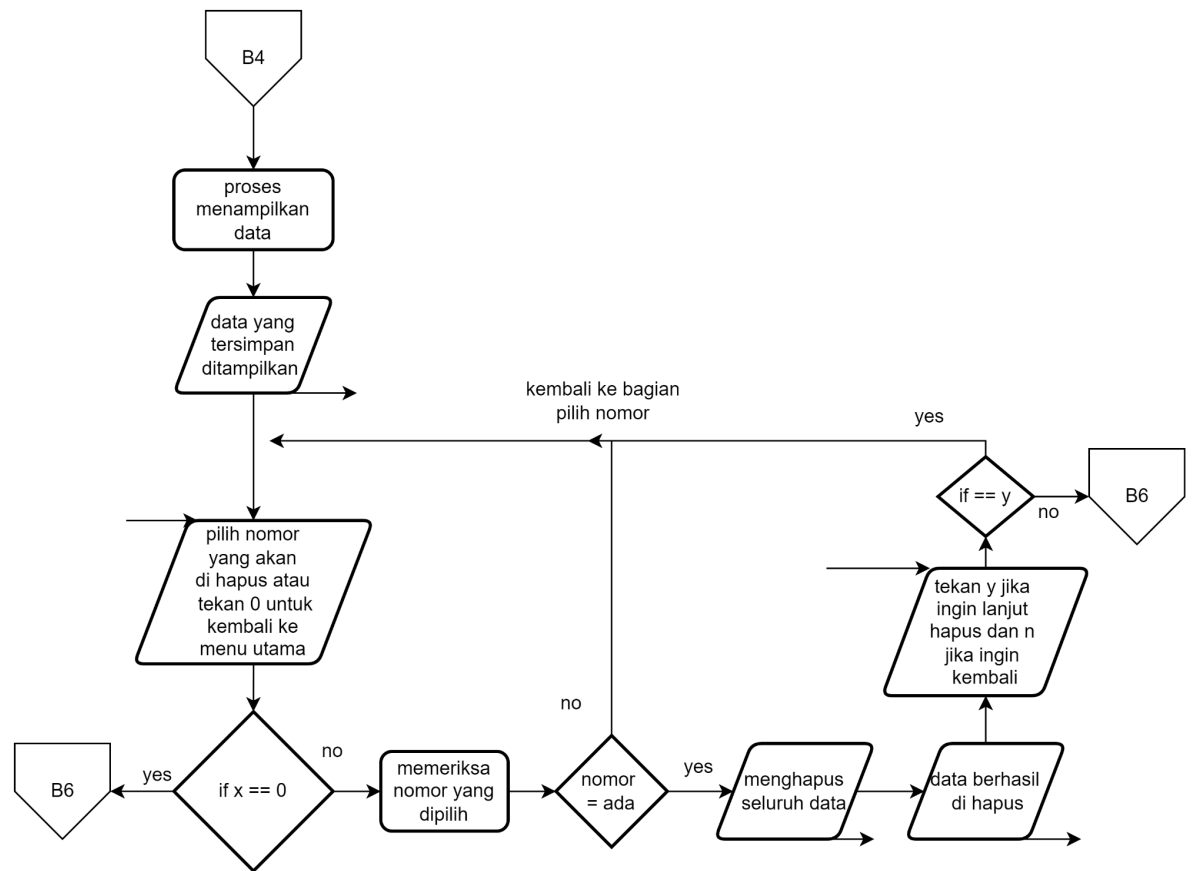
Gambar 1.4 Lihat data

E. Konversi Waktu



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”, 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.

2.2 Penjelasan Alur & Algoritma

- 1) Pada source code terdapat beberapa variabel untuk array 1 dimensi yaitu, string karakter, string deskripsi, string kelas, string branch, string tipe, kemudian terdapat `#define max 20` yang artinya maksimal data yang dapat ditampung ada 20 data.
- 2) Program diawali proses yang dimulai dengan deklarasi dari string yang mana string ini menampung variabel username, password, userbenar, dan passbenar, kemudian ada deklarasi dari integer dengan variabel gagal = 3. setelah itu masuk ke dalam proses login yang dimana terdapat perulangan while di dalamnya yang bertuliskan `while(gagal > 0)` yang mempunyai arti perulangan akan terus berjalan selama nilai variabel dari gagal masih lebih besar dari nol, karena awalnya gagal diatur ke 3 dan apabila pengguna memasukkan username atau password yang salah maka pengguna masih memiliki kesempatan untuk mencoba lagi hingga gagal tadi sudah menjadi nol maka program akan berhenti.
- 3) Setelah melewati proses login yang artinya pengguna memasukkan username dan password yang benar maka pengguna akan masuk ke bagian menu utama yang berisi pilihan-pilihan tambah data operator, lihat data operator, update data operator, hapus data operator, dan keluar. pada bagian menu utama ini terdapat deklarasi integer dengan variabel pilih, kemudian ada perulangan do while dengan `while(pilih != 5)` yang berfungsi agar program tidak berhenti sebelum pengguna memilih keluar, kemudian ada `system("cls")` ini berguna untuk membersihkan terminal pada saat output hasil agar tampilan terminal terlihat rapi.

- 4) kemudian terdapat percabangan `if(pilih == 1)` yang artinya pengguna memilih pilihan 1 yaitu tambah data operator, maka program akan menampilkan isi dari pilihan 1 yaitu beberapa inputan untuk menambah data informasi dari operator seperti masukkan nama operator, masukkan deskripsi operator, masukkan kelas operator, masukkan branch operator, dan masukkan tipe serangan operator. jika pengguna sudah selesai memasukkan semua itu maka program akan selesai menambah informasi dan program akan menampilkan pesan “Apakah anda ingin menambah operator lagi doktah? (y/n):” yang artinya jika pengguna ingin menambah lagi pengguna harus menekan y yang artinya yes dan jika tidak pengguna harus menekan n yang artinya no, dan bagaimana cara ini dapat berjalan, hal ini dikarenakan pada pilihan 1 ini menggunakan perulangan `do while` yang dimana `while(loop == “y”)`.
- 5) selanjutnya ada pilihan 2 yaitu lihat data operator program ini juga menggunakan percabangan `if(pilih == 2)`. di dalam program ini pertama-tama program akan mengecek apakah data di dalam program ada atau tidak dengan menggunakan percabangan lagi yaitu `if (pos > 0)` jika ada maka program akan menampilkan isi dari data informasi karakter yang tersimpan dengan menggunakan perulangan `for`, yang dimana jika terdapat 3 data maka perulangan ini akan berjalan sebanyak 3 kali untuk menampilkan data yang tersimpan
- 6) selanjutnya ada pilihan 3 yaitu update data operator, program ini juga menggunakan percabangan `if(pilih == 3)`. program ini berjalan dengan menampilkan data operator terlebih dahulu, tetapi data yang ditampilkan hanya nama operator saja jadi pengguna hanya akan memilih nomor berapa pada nama operator yang tersedia. proses melihat data ini sama dengan proses pada pilihan 2 yaitu menggunakan `if (pos > 0)` dan perulangan `for`, tetapi yang ditampilkan hanya nama operator. setelah melihat nama daftar operator, pengguna dapat memilih nomor mana yang akan di update, tapi sebelum itu pada program akan menampilkan pesan jika ingin kembali ke menu utama maka tekan 0, artinya jika pengguna berubah pikiran dan tidak jadi untuk update operator maka pengguna cukup menginput 0 dan program akan kembali ke menu utama, tetapi jika pengguna tetap lanjut maka pengguna harus memasukkan nomor operator yang ingin di update, jika sudah dipilih maka tampilan program akan sama seperti tambah data, tetapi disini gunanya akan mengupdate seluruh data operator. jika pengguna sudah selesai memasukkan semua itu maka program akan selesai mengupdate informasi dan program akan menampilkan pesan “Apakah anda ingin menambah operator lagi doktah? (y/n):” yang artinya jika pengguna ingin

menambah lagi pengguna harus menekan y yang artinya yes dan jika tidak pengguna harus menekan n yang artinya no, dan bagaimana cara ini dapat berjalan, hal ini dikarenakan pada pilihan 3 ini juga menggunakan perulangan do while yang dimana while(loop == "y").

- 7) selanjutnya ada pilihan 4 yaitu hapus data operator, program ini juga menggunakan percabangan if(pilih == 4). program ini sama seperti program pilihan 3, pertama-tama data akan ditampilkan dengan if(pos > 0) dan perulangan for, dan pengguna diminta untuk memilih nomor operator yang ingin dihapus, tapi sebelum itu pada program akan menampilkan pesan jika ingin kembali ke menu utama maka tekan 0, artinya jika pengguna berubah pikiran dan tidak jadi untuk update operator maka pengguna cukup menginput 0 dan program akan kembali ke menu utama, tetapi jika pengguna tetap lanjut maka pengguna harus memasukkan nomor operator yang ingin dihapus. jika sudah dipilih maka program akan menghapus data informasi operator tersebut. jika program sudah selesai menghapus data maka program akan menampilkan pesan "Apakah anda ingin menambah operator lagi doktah? (y/n):" yang artinya jika pengguna ingin menambah lagi pengguna harus menekan y yang artinya yes dan jika tidak pengguna harus menekan n yang artinya no, dan bagaimana cara ini dapat berjalan, hal ini dikarenakan pada pilihan 4 ini juga menggunakan perulangan do while yang sama seperti pilihan 1 dan 3 yang dimana while(loop == "y").
- 8) terakhir ada pilihan 5 yaitu keluar yang artinya jika pengguna memilih ini maka program akan berhenti dengan menampilkan pesan " selamat tinggal doktah".

3. Source Code

A. Variabel Dalam Array

Ini adalah seluruh variabel dari array yang ada pada program ini

Source Code:

```
// maksimal data 20
#define max 20

int main(){
    // variabl array
    int pos = 3;
    int pilih;
    string karakter[max] = {"Eyjafjalla", "Ines", "Nymph" };
    string deskripsi[max] = {"Dengan nama asli Adele Nauman, dan seorang putri
dari mendiang Katia dan Magna Nauman yaitu pasangan vulcanologist yang terkenal
dari Leithanien",
        "Ines adalah tentara bayaran yang aktif selama perang kazdel,bersama
dengan W dan Hoederer dalam ",
        "Seorang warga kazdel yang bergabung dengan Rhodes Island,dan memilik
keahlian dalam originium arts dan bakat alaminya sebagai djall"};
    string kelas[max] = {"Caster", "Vanguard","Caster"};
    string branch[max] = {"Core caster","Agent","Primal caster"};
    string tipe[max] = {"art","pyshical","art"};
    string loop;
```

Gambar 3.1 Variabel Array

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 << "Siahhkan 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, karakter[pos]);
    cout << "Masukkan deskripsi operator: ";
    getline(cin, deskripsi[pos]);
    cout << "Masukkan class operator: ";
    getline(cin, kelas[pos]);
    cout << "Masukkan branch operator: ";
    getline(cin, branch[pos]);
    cout << "Masukkan tipe serangan: ";
    getline(cin, tipe[pos]);
    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      : " << karakter[a] << endl;
            cout << "    Deskripsi    : " << deskripsi[a] << endl;
            cout << "    Kelas       : " << kelas[a] << endl;
        }
    }
}
```

```

        cout << "    Branch      : " << branch[a] << endl;
        cout << "    Tipe Serangan: " << tipe[a] << 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 << ". " << karakter[a] << 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();
}

```

```

else if(y > 0 && y <= pos)
{
    system("cls");
    cout << "===== " << endl;
    cout << "Update nama operator menjadi: ";
    getline(cin, karakter[y-1]);
    cout << "Update deskripsi operator menjadi: ";
    getline(cin, deskripsi[y-1]);
    cout << "Update class operator menjadi: ";
    getline(cin, kelas[y-1]);
    cout << "Update branch operator menjadi: ";
    getline(cin, branch[y-1]);
    cout << "Update tipe serangan operator menjadi: ";
    getline(cin, tipe[y-1]);
    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 << ". " << karakter[a] << 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();
}

```



```

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

4.1 Uji Coba

Beberapa uji coba yang digunakan dalam program ini adalah:

1. Pengguna berhasil masuk ke menu utama dengan username dan password yang benar
2. Pengguna salah memasukkan username atau password, ada kesempatan 3 kali tetapi ketika sudah gagal sebanyak 3 kali maka program akan berhenti
3. Jika pengguna berhasil masuk atau login maka pengguna akan berada ke dalam menu utama, yang didalam menu utama ada 5 pilihan yang bisa pengguna pilih.
4. Jika pengguna memilih pilihan 1 maka akan masuk ke dalam menu dari pilihan 1 yaitu tambah data operator, di dalam pilihan ini pengguna akan diminta untuk menambah informasi operator mulai dari nama, deskripsi, kelas, branch, dan tipe serangan, jika sudah selesai maka pengguna harus memilih apakah mau lanjut menambah atau berhenti dan kembali ke menu utama.
5. Jika pengguna memilih pilihan 2 maka akan masuk ke dalam menu dari pilihan 2 yaitu lihat data operator, di dalam pilihan ini pengguna hanya akan melihat data informasi operator yang telah tersimpan dan apabila kosong maka akan ada pemberitahuan data kosong, setelah melihat pengguna akan diminta untuk menekan enter untuk kembali.
6. Jika pengguna memilih pilihan 3 maka akan masuk ke dalam program update data operator, di dalam program ini pengguna dapat memilih nomor berapa pada operator yang ingin di update, dan jika pengguna memilih untuk keluar dari update data pengguna cukup menginput angka 0 untuk keluar, tetapi jika pengguna tidak ingin keluar pengguna harus menginput nomor operator yang ada dan pengguna harus menginput apa yang diperintahkan program maka program akan mengupdate data dari operator.
7. Jika pengguna memilih pilihan 4 maka akan masuk ke dalam program hapus data operator, di dalam program ini pengguna dapat memilih nomor berapa pada operator yang ingin di hapus, dan jika pengguna memilih untuk keluar dari update data pengguna cukup menginput angka 0 untuk keluar, tetapi jika pengguna tidak ingin keluar pengguna harus menginput nomor operator yang ada dan pengguna harus menginput apa yang diperintahkan program maka program akan menghapus data dari operator.

8. Jika pengguna memilih pilihan 5 maka program akan berhenti karena pilihan 5 adalah keluar.

4.2 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> git init
Reinitialized existing Git repository in C:/Users/ACER/Documents/praktikum-apl/.git/
PS C:\Users\ACER\Documents\praktikum-apl>
```

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 .
warning: in the working copy of 'post-test/post-test-1/2409106089-M.Fahrianor-PT-1.drawio', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'post-test/post-test-2/.2409106089-M.Fahrianor-PT-2.drawio.bkp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'post-test/post-test-2/2409106089-M.Fahrianor-PT-2.drawio', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.2 Git add

C. Git commit

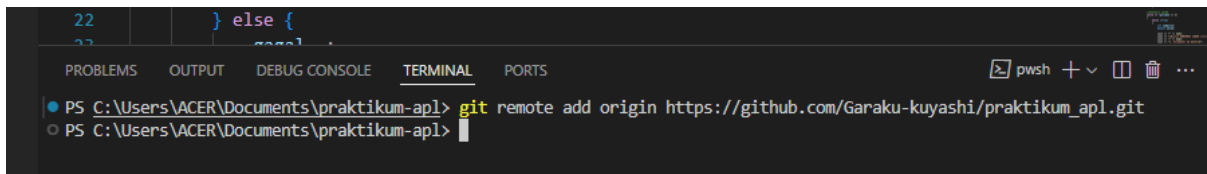
Perintah git commit digunakan untuk menyimpan perubahan yang telah di tambahkan ke dalam repository git

```
PS C:\Users\ACER\Documents\praktikum-apl> git commit -m 'finish'
[main f3e5d6d] finish
7 files changed, 1823 insertions(+), 38 deletions(-)
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-2/pertemuan-2.cpp
create mode 100644 kelas/pertemuan-2/pertemuan-2.exe
create mode 100644 post-test/post-test-2/.2409106089-M.Fahrianor-PT-2.drawio.bkp
create mode 100644 post-test/post-test-2/2409106089-M.Fahrianor-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106089-M.Fahrianor-PT-2.drawio
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

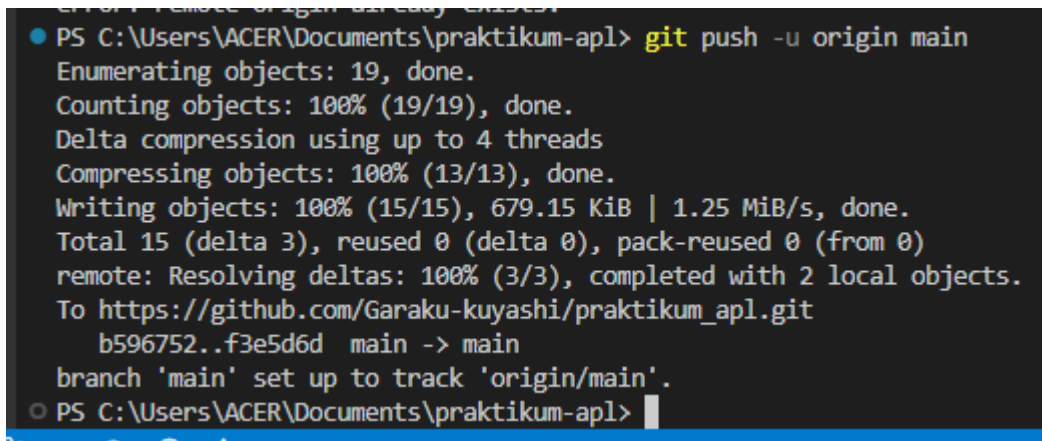


```
22 } else {  
23     ...  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\ACER\Documents\praktikum-apl> git remote add origin https://github.com/Garaku-kuyashi/praktikum_apl.git  
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.4 Git remote add origin

E. Git push

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



```
error: Remote origin already exists.  
PS C:\Users\ACER\Documents\praktikum-apl> git push -u origin main  
Enumerating objects: 19, done.  
Counting objects: 100% (19/19), done.  
Delta compression using up to 4 threads  
Compressing objects: 100% (13/13), done.  
Writing objects: 100% (15/15), 679.15 KiB | 1.25 MiB/s, done.  
Total 15 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.  
To https://github.com/Garaku-kuyashi/praktikum_apl.git  
b596752..f3e5d6d main -> main  
branch 'main' set up to track 'origin/main'.  
PS C:\Users\ACER\Documents\praktikum-apl>
```

Gambar 5.5 Git push