

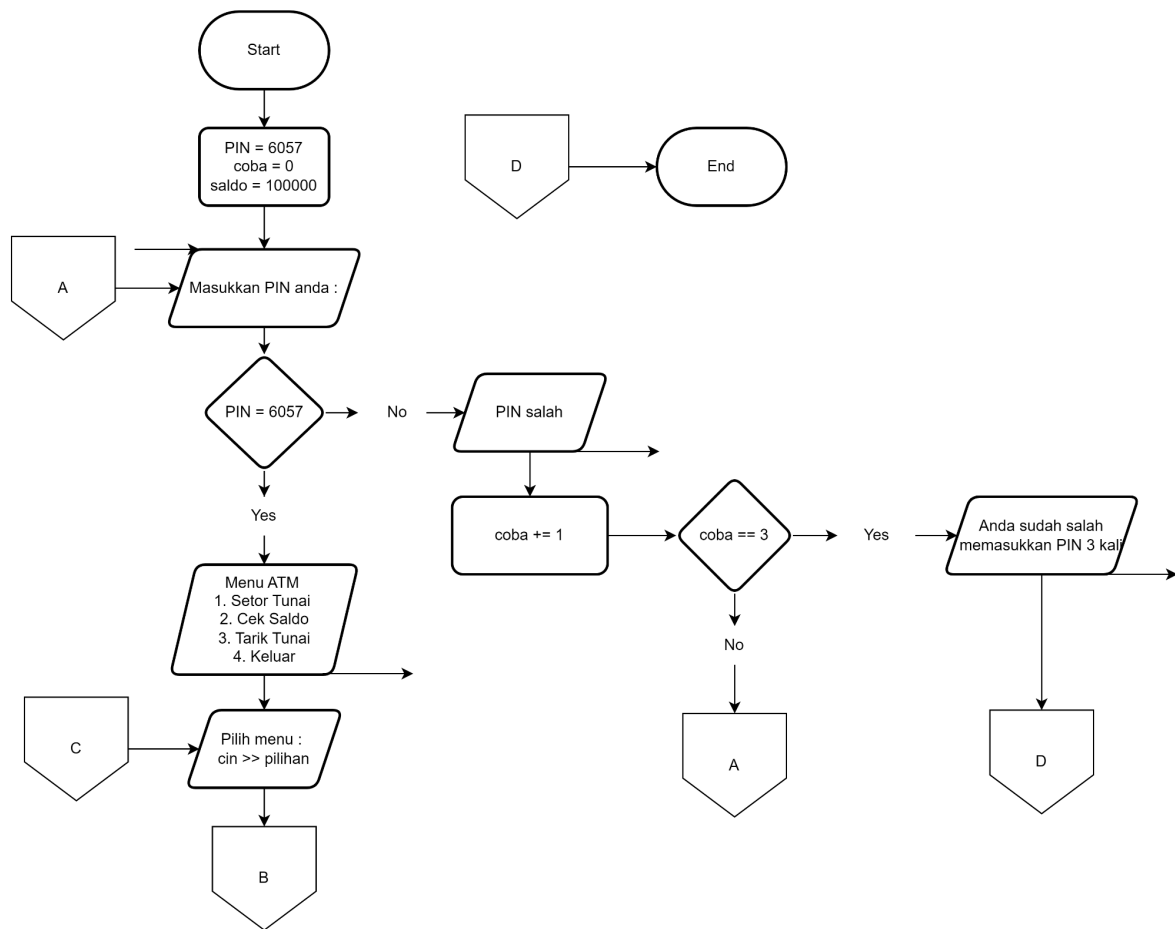
**LAPORAN PRAKTIKUM**  
**POSTTEST 1**  
**ALGORITMA PEMROGRAMAN LANJUT**



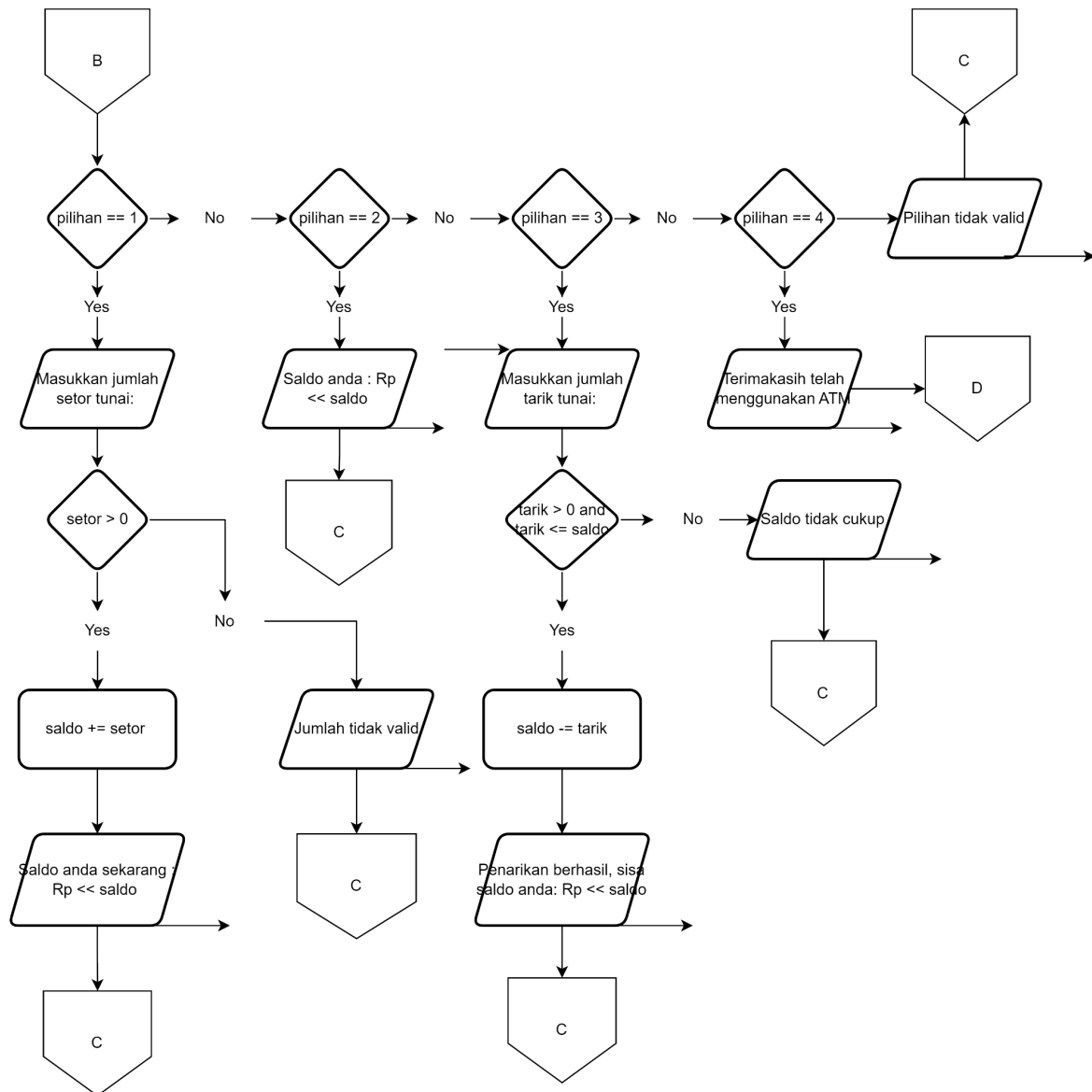
**Disusun oleh:**  
**Dimas Firjatullah Islamay (2409106057)**  
**Kelas (B1 '24)**

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

## 1. Flowchart



Gambar 1.1 Akses login



Gambar 1.2 Fitur tarik tunai, cek saldo, tarik tunai, keluar

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini merupakan program atm sederhana yang menerapkan beberapa fitur diantaranya seperti sistem login yang hanya memakai password, fitur create yang membuat kita dapat menambah nominal (setor tunai), fitur read untuk membaca data yang ada (cek saldo), fitur update untuk memperbarui data (tarik tunai, setor tunai), terdapat juga fitur looping yang dimana program akan mengulang terus kecuali user salah memasukkan pin sebanyak 3 kali atau user memilih keluar.

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

Berikut beberapa penjelasan alur dari program atm sederhana saya :

1. Saya mendeklarasikan beberapa variabel seperti PIN yang bernilai 6057 (integer) sebagai password agar dapat menjalankan program, MasukkanPin (integer) untuk menampung input PIN dari user, coba yang bernilai 0 (integer) untuk menghitung jumlah percobaan memasukkan PIN yang dimulai dari 0, pilihan (integer) untuk memilih menu di menu utama, dan saldo yang bernilai 100 ribu (double) sebagai saldo awal saya. variabel saldo memiliki tipe data double agar bisa diisi dengan angka desimal.
2. Selanjutnya saya membuat blok kode perulangan while yang didalamnya berfokus pada memasukkan PIN. Jadi jika if (MasukkanPin == PIN) maka akan muncul output login berhasil dan loop akan diberhentikan menggunakan break, ini berarti apa yang diinput user sudah sesuai dengan PIN awal yang sudah ditetapkan (yaitu 6057). Jika else atau salah maka dia akan terus looping dan coba++ yang berarti variabel coba yang nilai awalnya 0 akan bertambah 1.
3. Selanjutnya terdapat blok kode if (coba == 3) yang didalamnya berisi output dan return 0, artinya jika nilai variabel coba yang awalnya bernilai 0 telah bernilai 3 maka user sudah salah memasukkan PIN sebanyak 3 kali dan program akan otomatis diberhentikan.
4. Selanjutnya ada blok kode perulangan while (true) yang akan terus mengulang kode sampai bertemu dengan break. Lalu terdapat 4 pilihan menu utama atm seperti yang

sudah dijelaskan sebelumnya, user tinggal memilih mau masuk menu keberapa dengan menginput dan akan disimpan ke variabel Pilihan.

5. Jika variabel pilihan bernilai 1, maka user akan dimasukkan ke blok kode setor tunai. Didalamnya user akan memasukkan ingin menyetor berapa, lalu terdapat fungsi if (setor > 0) {saldo += setor}, artinya jika jumlah yang mau kita setor lebih besar dari 0, maka saldo kita di awal akan ditambah dengan jumlah setor yang sudah kita inputkan tadi lalu akan ditampilkan outputnya setelah di total. Jika else atau salah, maka akan muncul output jumlah tidak valid dan program akan mengulang kembali ke menu utama.
6. Jika variabel pilihan bernilai 2, maka program hanya akan menampilkan jumlah saldo saat ini dan akan mengulang kembali ke menu utama.
7. Jika variabel pilihan bernilai 3, maka user akan dimasukkan ke blok kode tarik tunai. Didalamnya user akan memasukkan ingin menarik jumlah uang berapa, lalu terdapat fungsi if (tarik > 0 and tarik <= saldo) {saldo -= tarik}, artinya jika jumlah yang mau kita tarik lebih besar dari 0 dan lebih kecil sama dengan dari jumlah saldo kita, maka saldo kita di awal akan dikurangi dengan jumlah penarikan yang sudah kita inputkan tadi lalu akan ditampilkan outputnya setelah di total. Jika else atau salah, maka akan muncul output jumlah tidak valid dan program akan mengulang kembali ke menu utama.
8. Jika variabel pilihan bernilai 4, maka user akan diperlihatkan output terimakasih, artinya user telah sepenuhnya keluar program dan perulangan akan terhenti karena ada break.
9. Jika variabel pilihan bernilai apapun selain 1, 2, 3, 4 maka dia akan masuk ke bagian else dari blok kode perulangan while (true) yang berada di langkah keempat. User akan diperlihatkan output pilihan tidak valid dan program akan mengulang kembali ke menu utama.

### 3. Source Code

#### A. Login

Fitur ini digunakan untuk memvalidasi user yang ingin menggunakan aplikasi melalui proses autentikasi dengan PIN.

##### Source Code:

```
while (coba < 3)
{
    cout << "Masukkan PIN anda: ";
    cin >> MasukkanPin;
    if (MasukkanPin == PIN)
    {
        cout << "Login berhasil" << endl;
        break;
    }
    else
    {
        cout << "PIN salah" << endl;
        coba++;
    }
}

if (coba == 3)
{
    cout << "Anda sudah salah memasukkan PIN 3 kali";
    return 0;
}
```

Gambar 3.1

#### B. Menu Utama

Terdapat beberapa menu berisi fitur-fitur yang dapat dipilih oleh user setelah berhasil login.

##### Source Code:

```
cout << "=== MENU ATM ===" << endl;
cout << "1. Setor Tunai" << endl;
cout << "2. Cek Saldo" << endl;
cout << "3. Tarik Tunai" << endl;
cout << "4. Keluar" << endl;
cout << "Pilih menu: ";
```

Gambar 3.2

### C. Setor Tunai

Blok kode yang memiliki fitur untuk melakukan penjumlahan.

**Source Code:**

```
double setor;  
    cout << "Masukkan jumlah setor tunai: ";  
    cin >> setor;  
    if (setor > 0)  
    {  
        saldo += setor;  
        cout << "Saldo anda sekarang: Rp " << saldo << endl;  
    }  
    else  
    {  
        cout << "Jumlah tidak valid" << endl;  
    }  
}
```

Gambar 3.3

### D. Cek Saldo

Blok kode yang memiliki fitur read untuk memperlihatkan data sekarang.

**Source Code:**

```
cout << "Saldo anda: Rp " << saldo << endl;
```

Gambar 3.4

### E. Tarik Tunai

Blok kode yang memiliki fitur pengurangan.

### Source Code:

```
double tarik;  
    cout << "Masukkan jumlah tarik tunai: ";  
    cin >> tarik;  
    if (tarik > 0 and tarik <= saldo)  
    {  
        saldo -= tarik;  
        cout << "Penarikan berhasil, sisa saldo anda: Rp " << saldo << endl;  
    }  
    else  
    {  
        cout << "Saldo tidak cukup" << endl;  
    }  
}
```

Gambar 3.5

### F. Menu Keluar

Menu ini digunakan untuk menyelesaikan program dan mengakhiri looping.

### Source Code:

```
cout << "Terima kasih telah menggunakan ATM" << endl;  
break;
```

Gambar 3.6



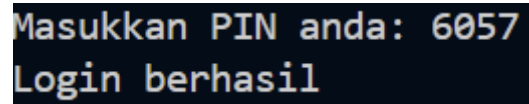
## **4. Uji Coba dan Hasil Output**

### **4.1 Uji Coba**

Berikut beberapa skenario yang akan digunakan dalam pengujian program saya:

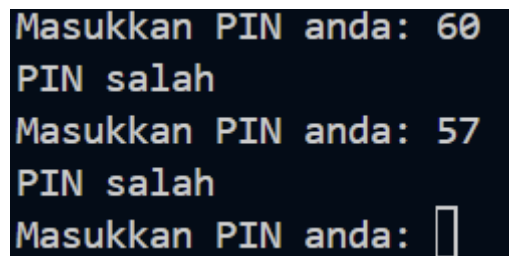
1. Ketika user berhasil memasukkan PIN dengan benar.
2. Ketika user salah memasukkan PIN kurang dari 3 kali percobaan maka program akan menampilkan pesan PIN salah, dan program akan mengulang.
3. Ketika user salah memasukkan PIN sebanyak 3 kali maka program akan berhenti.
4. Setelah user berhasil memasukkan PIN dengan benar, user akan dibawa ke halaman menu utama atm dan user akan diberi pilihan menu.
5. Jika user memilih pilihan nomor 1, maka user akan memasuki halaman setor tunai, user akan diminta untuk menginput berapa jumlah yang ingin disetor dan akan diperbarui di dalam saldo setelah ditambah, lalu program akan kembali ke menu utama.
6. Jika user memilih pilihan nomor 2, maka user akan memasuki halaman cek saldo, user akan diperlihatkan jumlah saldo saat ini, lalu program akan kembali ke menu utama.
7. Jika user memilih pilihan nomor 3, maka user akan memasuki halaman tarik tunai, user akan diminta untuk menginput berapa jumlah yang ingin ditarik dan akan diperbarui di dalam saldo setelah dikurangi, lalu program akan kembali ke menu utama.
8. Jika user memilih pilihan nomor 4, maka user akan keluar dan program dinyatakan selesai.
9. Jika user memilih pilihan selain nomor 1, 2, 3, 4 maka akan muncul pesan pilihan tidak valid, lalu program akan kembali ke menu utama.

## 4.2 Hasil Output



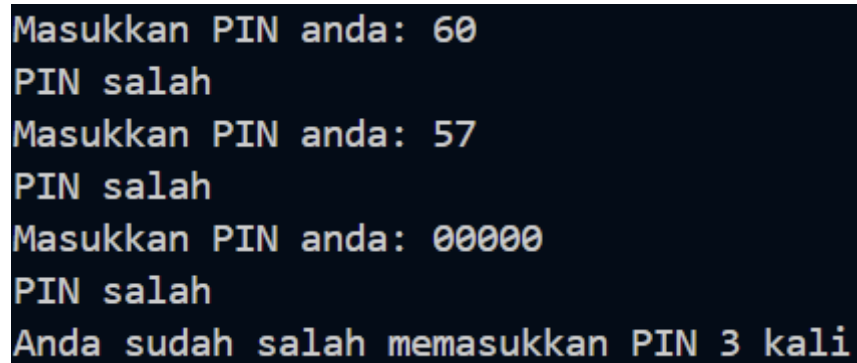
```
Masukkan PIN anda: 6057
Login berhasil
```

Gambar 4.1 Login berhasil



```
Masukkan PIN anda: 60
PIN salah
Masukkan PIN anda: 57
PIN salah
Masukkan PIN anda: 
```

Gambar 4.2 PIN salah kurang dari 3 kali



```
Masukkan PIN anda: 60
PIN salah
Masukkan PIN anda: 57
PIN salah
Masukkan PIN anda: 00000
PIN salah
Anda sudah salah memasukkan PIN 3 kali
```

Gambar 4.3 PIN salah sebanyak 3 kali

```
=== MENU ATM ===  
1. Setor Tunai  
2. Cek Saldo  
3. Tarik Tunai  
4. Keluar  
Pilih menu: 
```

Gambar 4.4 Menu utama

```
Pilih menu: 1  
Masukkan jumlah setor tunai: 200000  
Saldo anda sekarang: Rp 300000
```

Gambar 4.5 Setor tunai

```
Pilih menu: 2  
Saldo anda: Rp 300000
```

Gambar 4.6 Cek saldo

```
Pilih menu: 3  
Masukkan jumlah tarik tunai: 200000  
Penarikan berhasil, sisa saldo anda: Rp 100000
```

Gambar 4.7 Tarik tunai

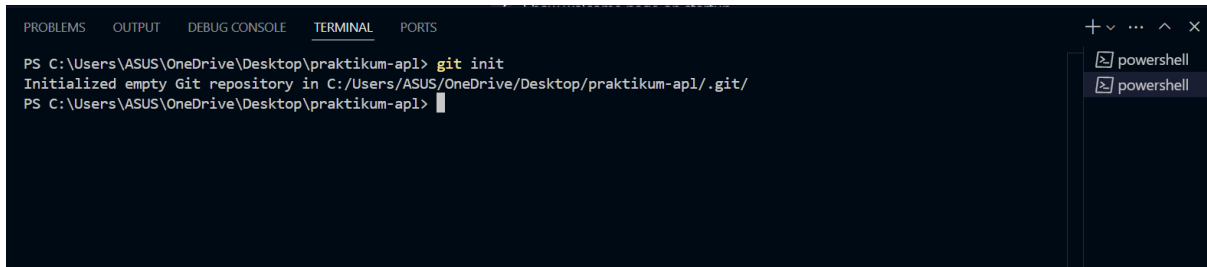
```
Pilih menu: 4  
Terima kasih telah menggunakan ATM
```

Gambar 4.8 Keluar program

```
Pilih menu: 5  
Pilihan tidak valid
```

Gambar 4.9 Pilihan tidak valid

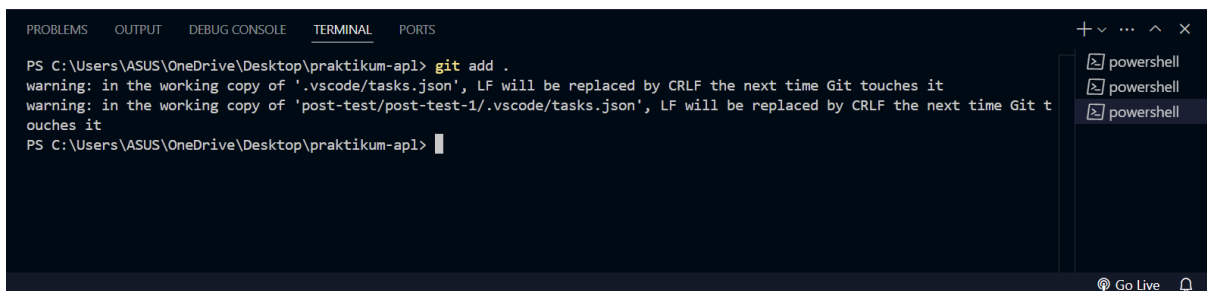
## 5. Langkah-langkah GIT



```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git init
Initialized empty Git repository in C:/Users/ASUS/OneDrive/Desktop/praktikum-apl/.git/
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl>
```

Gambar 5.1 Git init

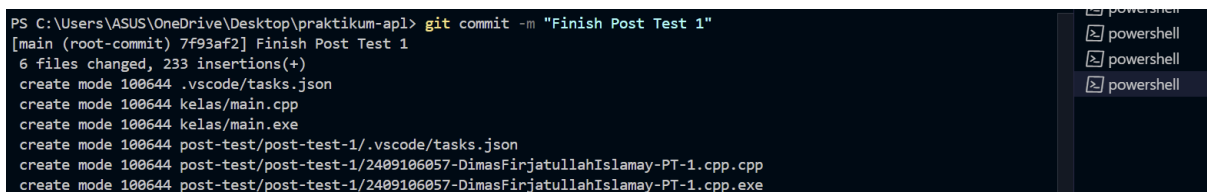
git init digunakan untuk membuat repository lokal yang kosong



```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git add .
warning: in the working copy of '.vscode/tasks.json', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'post-test/post-test-1/.vscode/tasks.json', LF will be replaced by CRLF the next time Git touches it
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl>
```

Gambar 5.2 Git add

Git add digunakan untuk menambahkan file yang ingin di commit di step selanjutnya



```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git commit -m "Finish Post Test 1"
[main (root-commit) 7f93af2] Finish Post Test 1
6 files changed, 233 insertions(+)
create mode 100644 .vscode/tasks.json
create mode 100644 kelas/main.cpp
create mode 100644 kelas/main.exe
create mode 100644 post-test/post-test-1/.vscode/tasks.json
create mode 100644 post-test/post-test-1/2409106057-DimasFirjatullahIslamay-PT-1.cpp.cpp
create mode 100644 post-test/post-test-1/2409106057-DimasFirjatullahIslamay-PT-1.cpp.exe
```

Gambar 5.3 Git commit

Git commit digunakan untuk menyimpan hasil file yang telah ditambahkan



```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git remote add https://github.com/EuphoriaBot/praktikum-apl.git
usage: git remote add [<options>] <name> <url>

    -f, --[no-]fetch          fetch the remote branches
    --[no-]tags              import all tags and associated objects when fetching
                             or do not fetch any tag at all (--no-tags)
    -t, --[no-]track <branch>
                             branch(es) to track
    -m, --[no-]master <branch>
                             master branch
    --[no-]mirror[=(push|fetch)]
                             set up remote as a mirror to push to or fetch from

PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git remote add origin https://github.com/EuphoriaBot/praktikum-apl.git
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl>
```

Gambar 5.4 Git remote

Git remote digunakan untuk menghubungkan ke server lain seperti Github

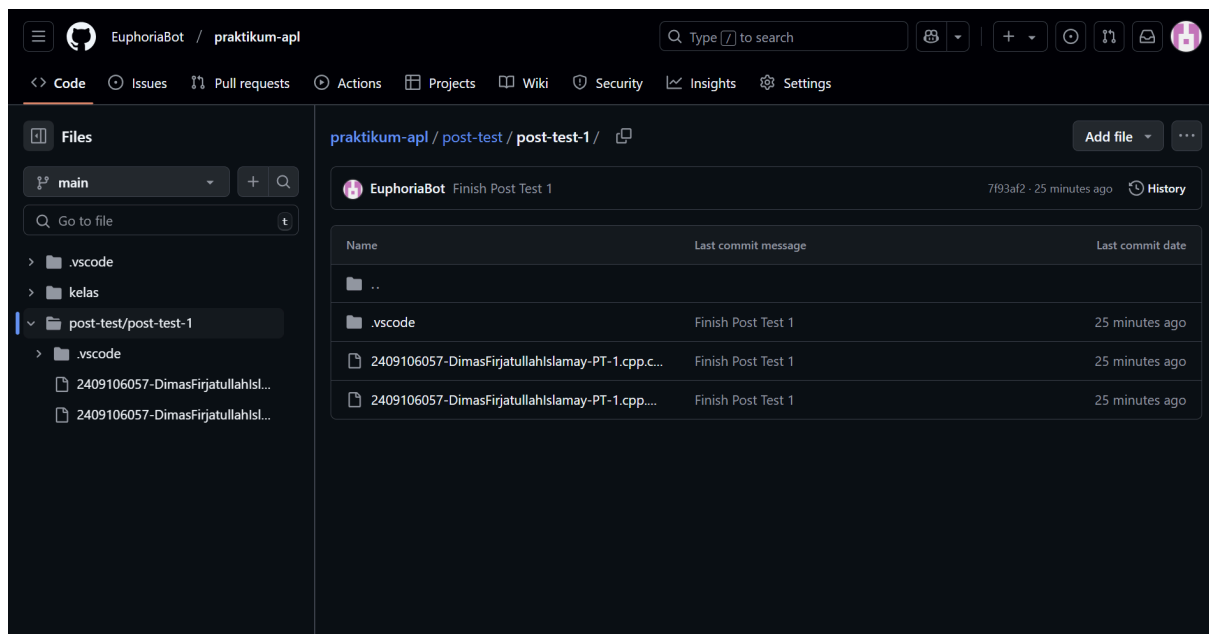
```

PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git status
On branch main
nothing to commit, working tree clean
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 22 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (11/11), 79.93 KiB | 2.50 MiB/s, done.
Total 11 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/EuphoriaBot/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl>

```

Gambar 5.5 Git push

Git push digunakan untuk mengupload seluruh commit yang sudah dilakukan dalam satu repository lokal



Gambar 5.6

Berikut contoh jika langkah-langkah Git berhasil dilaksanakan