

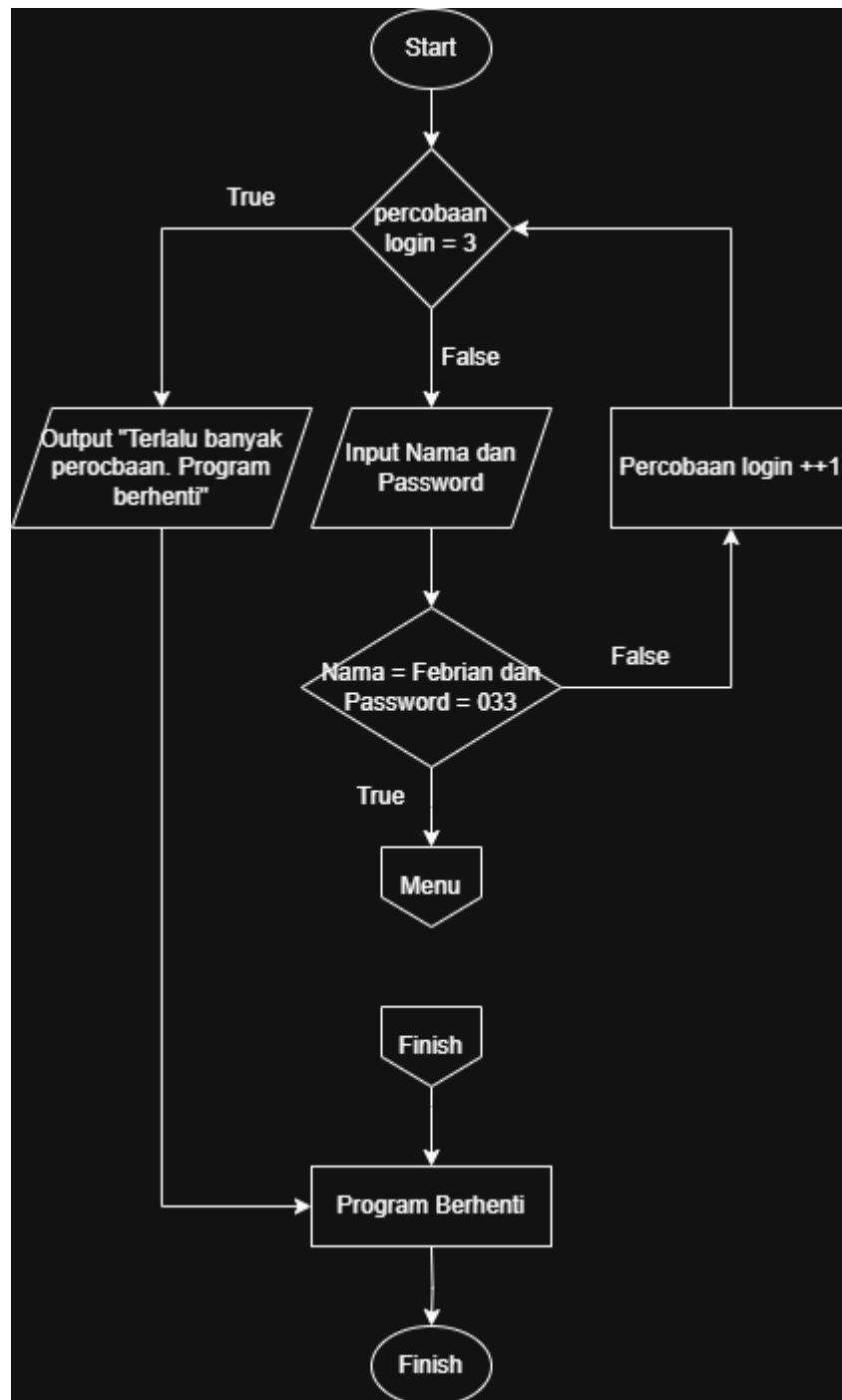
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



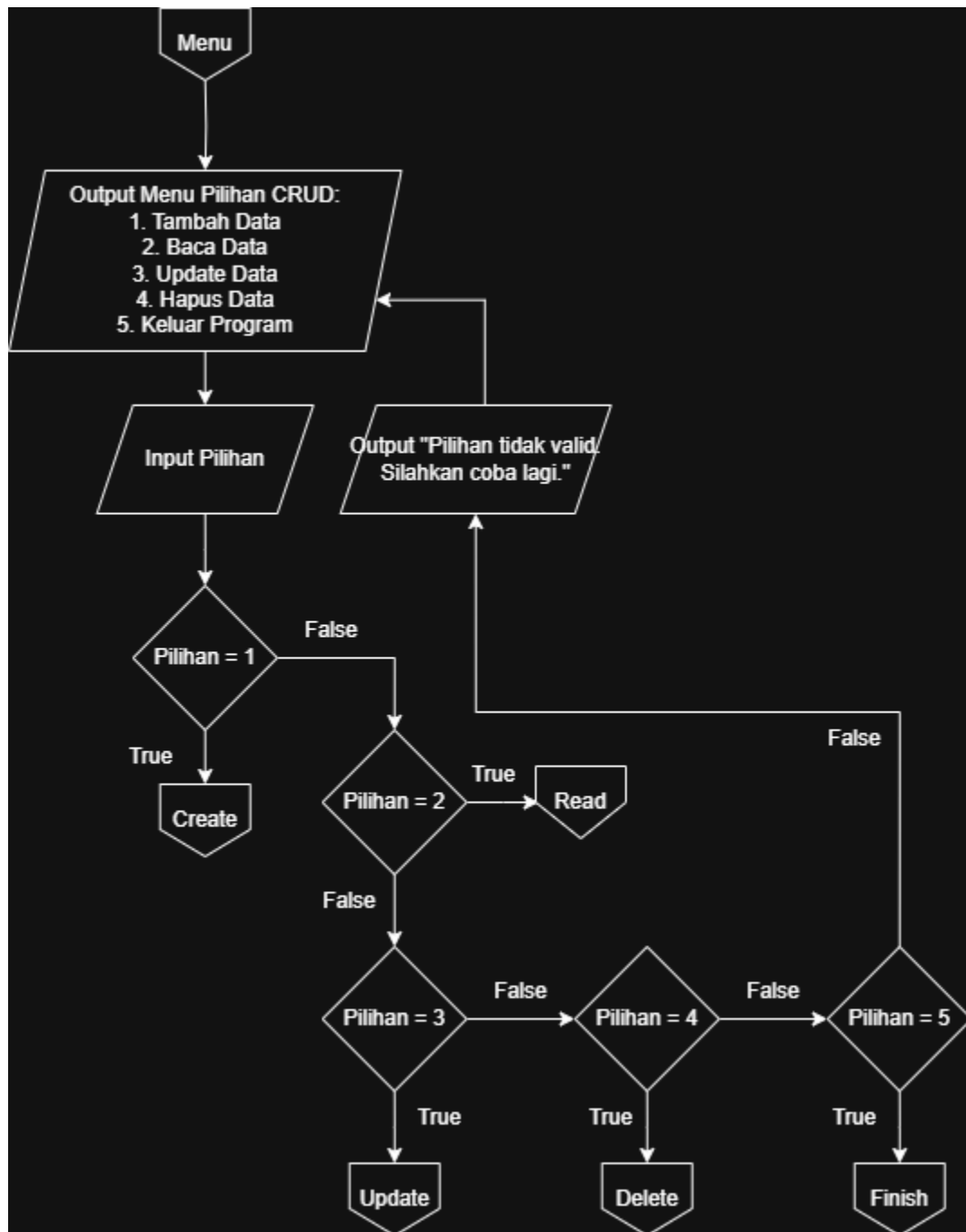
Disusun oleh:
Febrian Pratama Saputra (2409106033)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

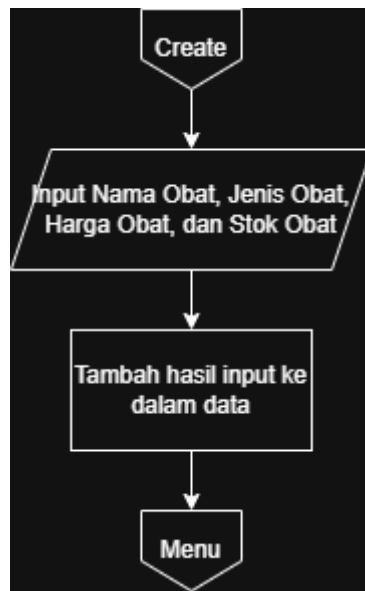
1. Flowchart



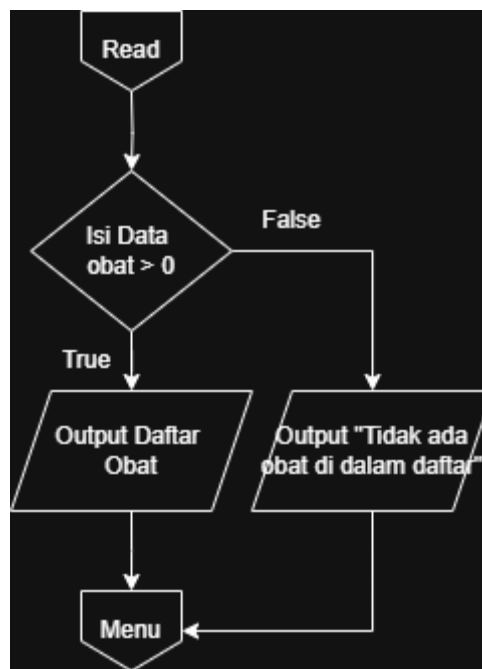
Gambar 1.1 Flowchart Login



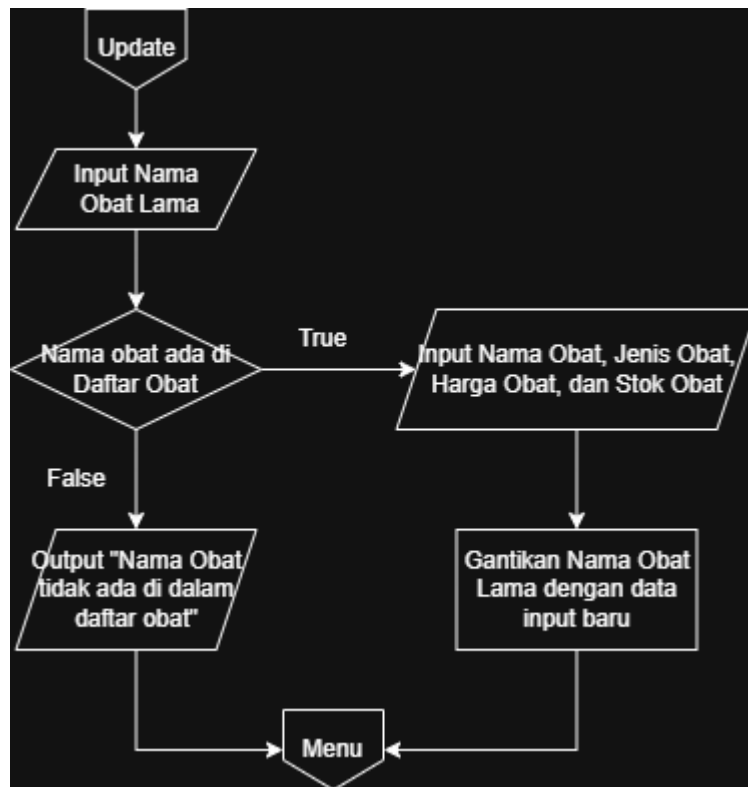
Gambar 1.2 Flowchart Menu



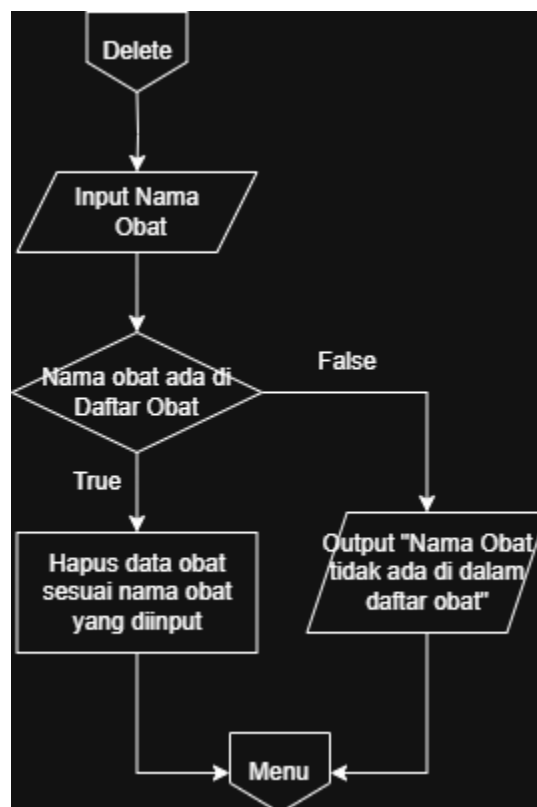
Gambar 1.3 Flowchart Create



Gambar 1.4 Flowchart Read



Gambar 1.5 Flowchart Update



Gambar 1.6 Flowchart Delete

2. Analisis Program

Program berbasis C++ ini berfungsi untuk mengurus data stok obat di sebuah apotek dengan fitur CRUD (Create, Read, Update, Delete). Program ini telah diamankan dengan fitur login untuk mencegah akses tidak sah. Program ini menggunakan m

3. Source Code

A. Fitur Login

Fitur ini berfungsi untuk mengamankan program dari akses tidak sah. Program akan berhenti jika user salah memasukkan kredensial login 3x.

```
string namaLogin, passwordLogin;
string namaBenar = "Febrian", passwordBenar = "033";
int percobaanLogin = 0;

while (percobaanLogin < 3) {
    cout << "Masukkan Username: "; cin >> namaLogin;
    cout << "Masukkan Password: "; cin >> passwordLogin;

    if (namaLogin == namaBenar && passwordLogin == passwordBenar) {
        cout << "Login berhasil! Selamat datang pengguna program.\n";
        break;
    } else {
        cout << "Username atau Password salah! Silahkan coba lagi.\n";
        percobaanLogin++;
    }
}

if (percobaanLogin == 3) {
    cout << "Maaf! Terlalu banyak percobaan gagal. Program akan segera
berhenti.\n";
    return 0;
}
```

B. Fitur Menu

Fitur ini berfungsi sebagai menu utama user untuk melakukan CRUD data. Program akan selalu looping kembali ke fitur ini untuk setiap fitur CRUD.

```
cout << "\nManajemen Stok Obat Apotek";
cout << "\n1. Tambah Obat";
cout << "\n2. Tampilkan Obat";
cout << "\n3. Ubah Obat";
cout << "\n4. Hapus Obat";
cout << "\n5. Keluar";
cout << "\nPilih menu: ";
cin >> pilihan;
```

C. Fitur CRUD

Fitur utama program. Fitur ini memberikan user kemampuan untuk melaksanakan CRUD (Create, Read, Update, Delete) pada sebuah data.

```
if (pilihan == 1) {
    cout << "\nMasukkan Nama Obat: "; getline(cin,
daftarObat[jumlah].nama); // getline(cin, ...) agar input dengan spasi tidak
merusak program
    cout << "Masukkan Jenis Obat: "; getline(cin,
daftarObat[jumlah].jenis); // getline(cin, ...) agar input dengan spasi tidak
merusak program
    cout << "Masukkan Harga Obat: "; cin >> daftarObat[jumlah].harga;
    cout << "Masukkan Stok Obat: "; cin >> daftarObat[jumlah].stok;
    cin.ignore(); // agar newline (\n) dari cin sebelumnya tidak
terbaca, hingga merusak program saat menjumpai getline berikutnya
    jumlah++;
    cout << "Obat berhasil ditambahkan!\n";
} else if (pilihan == 2) {
    if (jumlah == 0) {
        cout << "Tidak ada obat dalam sistem.\n";
    } else {
        cout << "\nDaftar Obat:\n";
        for (int i = 0; i < jumlah; i++) {
            cout << "Nama: " << daftarObat[i].nama
                << ", Jenis: " << daftarObat[i].jenis
                << ", Harga: " << daftarObat[i].harga
                << ", Stok: " << daftarObat[i].stok << "\n";
        }
    }
} else if (pilihan == 3) {
    string ubah;
    cout << "\nMasukkan Nama Obat yang ingin diubah: "; getline(cin,
ubah);

    bool ditemukan = false;
    for (int i = 0; i < jumlah; i++) {
        if (daftarObat[i].nama == ubah) {
            cout << "Masukkan Nama Baru: "; getline(cin,
daftarObat[i].nama); // getline(cin, ...) agar input dengan spasi tidak merusak
program
            cout << "Masukkan Jenis Baru: "; getline(cin,
daftarObat[i].jenis); // getline(cin, ...) agar input dengan spasi tidak merusak
program
            cout << "Masukkan Harga Baru: "; cin >> daftarObat[i].harga;
            cout << "Masukkan Stok Baru: "; cin >> daftarObat[i].stok;
            cin.ignore(); // agar newline (\n) dari cin sebelumnya
tidak terbaca, hingga merusak program saat menjumpai getline berikutnya
            cout << "Obat berhasil diperbarui!\n";
            ditemukan = true;
            break;
        }
    }
}
```



```

        }
    }
    if (!ditemukan) {
        cout << "Nama obat tidak ditemukan!\n";
    }
} else if (pilihan == 4) {
    string hapus;
    cout << "\nMasukkan Nama Obat yang ingin dihapus: "; getline(cin,
hapus);

    bool ditemukan = false;
    for (int i = 0; i < jumlah; i++) {
        if (daftarObat[i].nama == hapus) {
            for (int j = i; j < jumlah - 1; j++) {
                daftarObat[j] = daftarObat[j + 1];
            }
            jumlah--;
            cout << "Obat berhasil dihapus!\n";
            ditemukan = true;
            break;
        }
    }
    if (!ditemukan) {
        cout << "Nama obat tidak ditemukan!\n";
    }
} else if (pilihan == 5) {
    cout << "\nTerima kasih telah menggunakan program ini.\n";
} else {
    cout << "\nPilihan tidak valid! Silahkan coba lagi.\n";
}
}

```

4. Uji Coba dan Hasil Output

```
Masukkan Username: Febrian
Masukkan Password: 033
Login berhasil! Selamat datang pengguna program.

Manajemen Stok Obat Apotek
1. Tambah Obat
2. Tampilkan Obat
3. Ubah Obat
4. Hapus Obat
5. Keluar
Pilih menu: 1

Masukkan Nama Obat: Xanax Alprazolam
Masukkan Jenis Obat: Sedatif
Masukkan Harga Obat: 400000
Masukkan Stok Obat: 50
Obat berhasil ditambahkan!

Manajemen Stok Obat Apotek
1. Tambah Obat
2. Tampilkan Obat
3. Ubah Obat
4. Hapus Obat
5. Keluar
Pilih menu: 2

Daftar Obat:
Nama: Xanax Alprazolam, Jenis: Sedatif, Harga: 400000, Stok: 50
```

Gambar 4.1 Screenshot Terminal Login, Menu, Create, dan Read

```
Masukkan Nama Obat yang ingin diubah: Xanax Alprazolam
Masukkan Nama Baru: Tylenol Asetaminofen
Masukkan Jenis Baru: Analgesik
Masukkan Harga Baru: 30000
Masukkan Stok Baru: 30
Obat berhasil diperbarui!

Manajemen Stok Obat Apotek
1. Tambah Obat
2. Tampilkan Obat
3. Ubah Obat
4. Hapus Obat
5. Keluar
Pilih menu: 4

Masukkan Nama Obat yang ingin dihapus: Tylenol Asetaminofen
Obat berhasil dihapus!

Manajemen Stok Obat Apotek
1. Tambah Obat
2. Tampilkan Obat
3. Ubah Obat
4. Hapus Obat
5. Keluar
Pilih menu: 2
Tidak ada obat dalam sistem.

Manajemen Stok Obat Apotek
1. Tambah Obat
2. Tampilkan Obat
3. Ubah Obat
4. Hapus Obat
5. Keluar
Pilih menu: 5

Terima kasih telah menggunakan program ini.
```

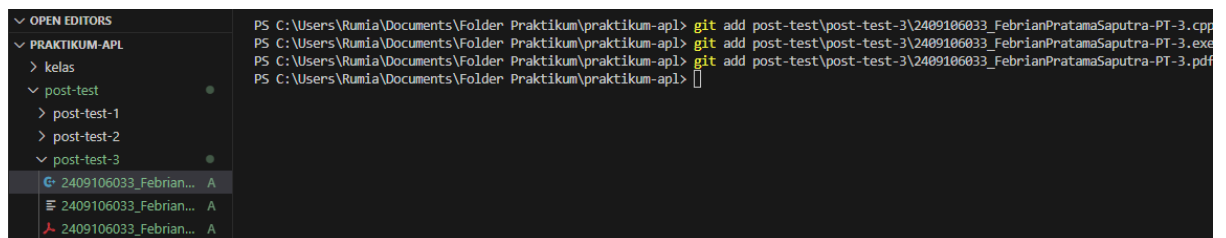
Gambar 4.2 Screenshot Terminal Update, Delete, dan Keluar

5. Langkah-Langkah Git pada VSCode

Dikarenakan git telah diinisiasi, dan remote telah tersambung ke github, kita bisa melewati langkah tersebut.

A. Git Add

Menambahkan file-file yang akan dicommit.

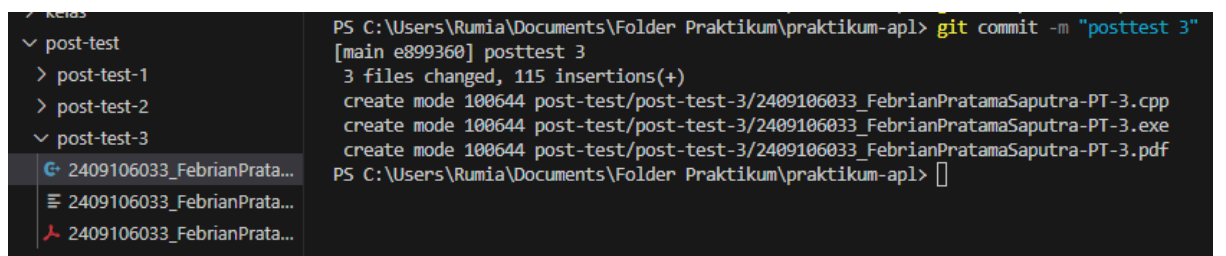


```
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git add post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.*
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git add post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.exe
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git add post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.pdf
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl>
```

Gambar 5.1 Git Add

B. Git Commit

Mengcommit file yang telah ditambahkan agar dijadikan menjadi sebuah checkpoint.

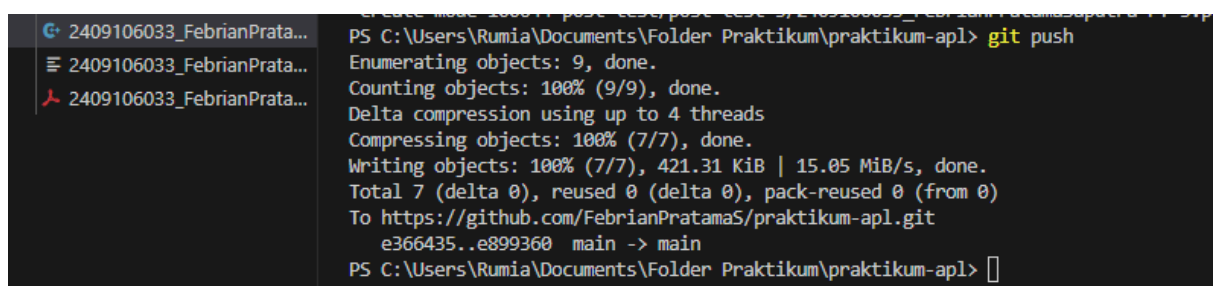


```
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git commit -m "posttest 3"
[main e899360] posttest 3
3 files changed, 115 insertions(+)
create mode 100644 post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.cpp
create mode 100644 post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.exe
create mode 100644 post-test/post-test-3/2409106033_FebrianPratamaSaputra-PT-3.pdf
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl>
```

Gambar 5.2 Git Commit

C. Git Push

Mengupload commit yang telah dilakukan ke repositori github.



```
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 421.31 KiB | 15.05 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/FebrianPratamaS/praktikum-apl.git
e366435..e899360 main -> main
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl>
```

Gambar 5.3 Git Push