

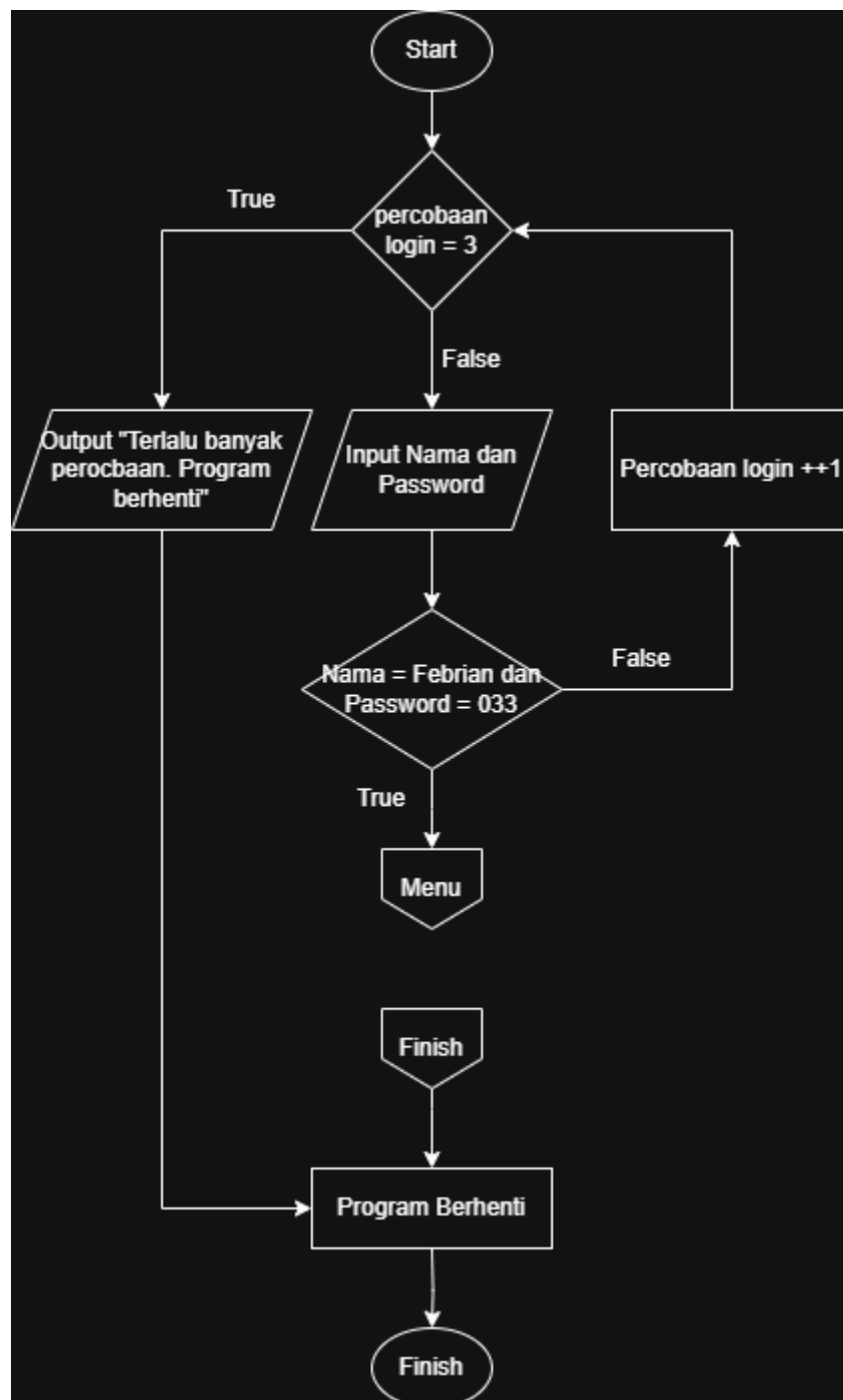
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



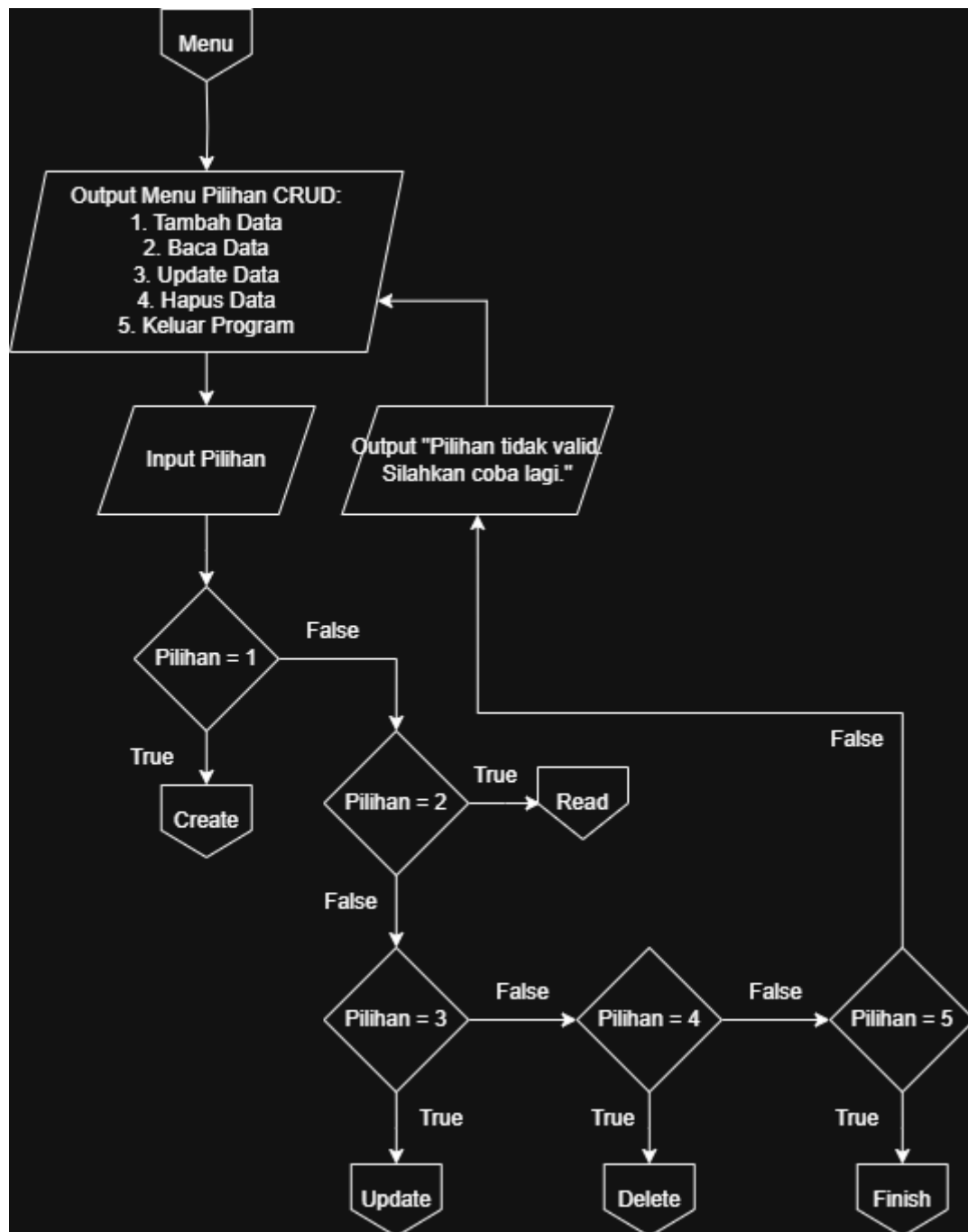
Disusun oleh:
Febrian Pratama Saputra (2409106033)
Kelas (A2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

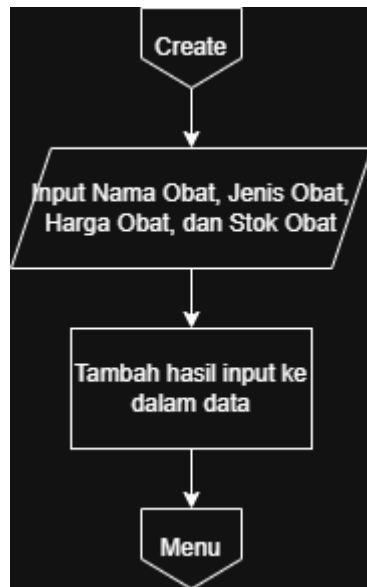
1. Flowchart



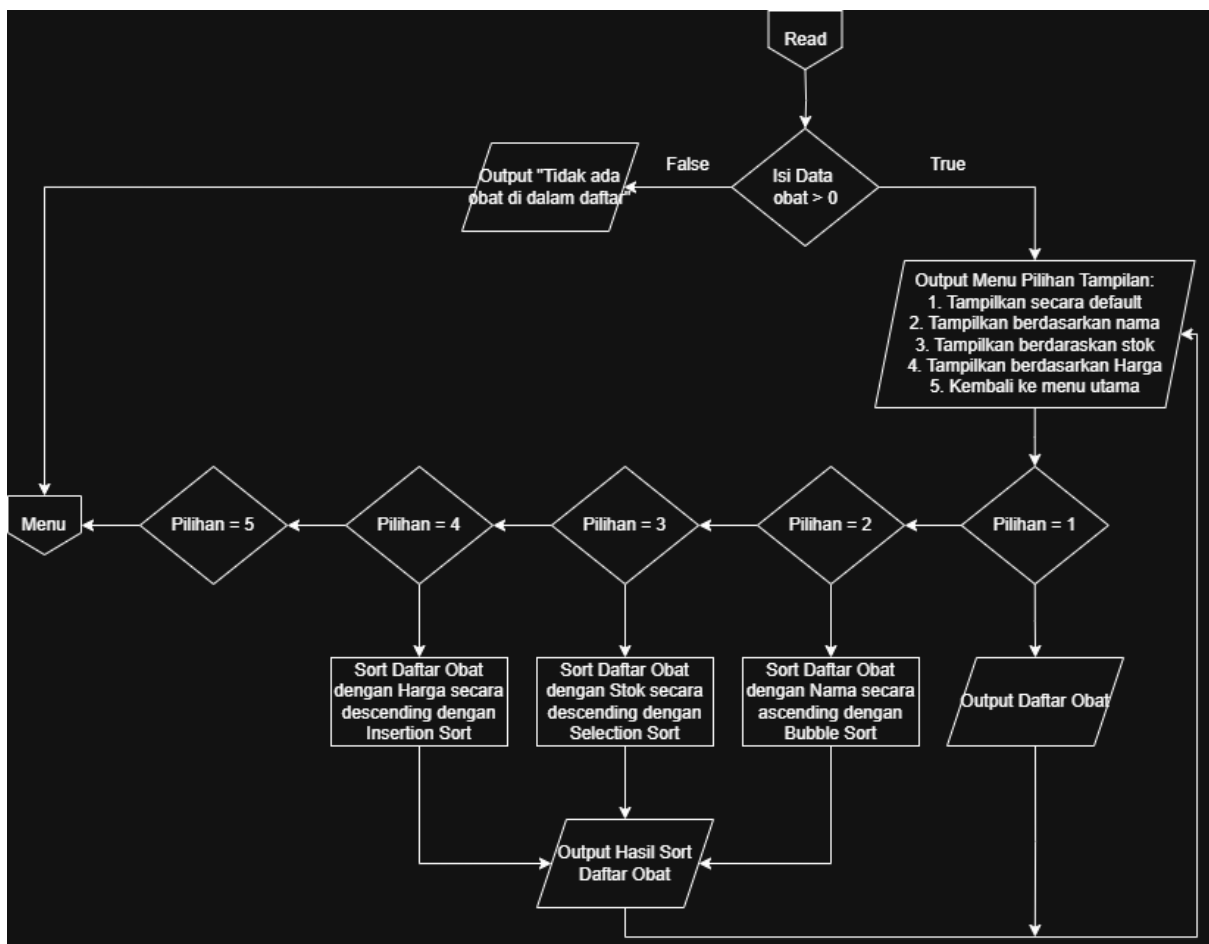
Gambar 1.1 Flowchart Start dan Login



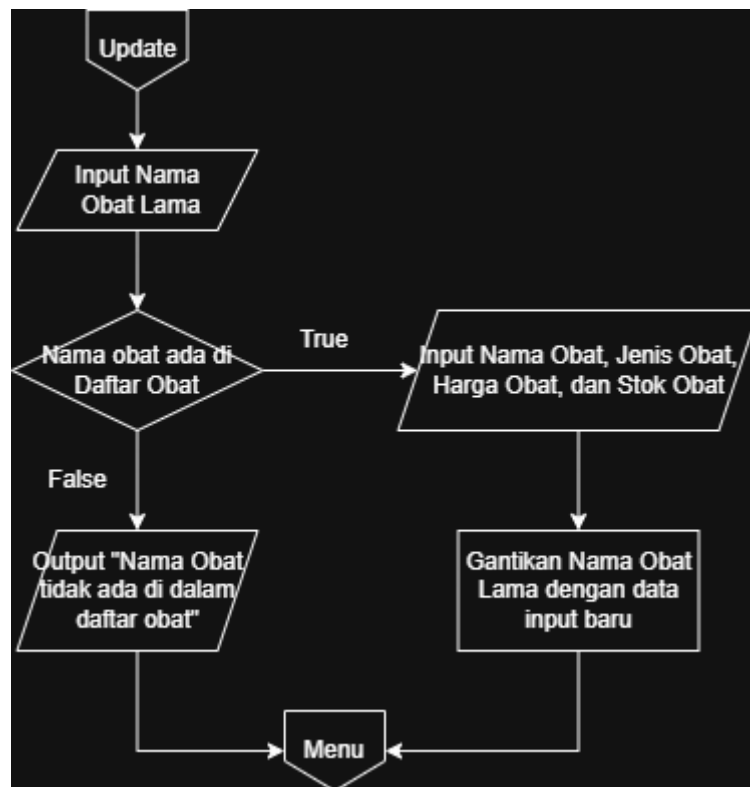
Gambar 1.2 Flowchart Menu



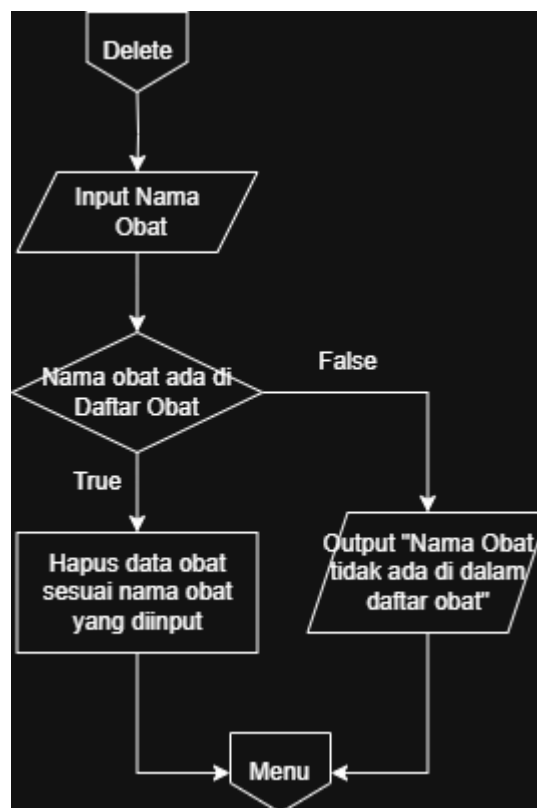
Gambar 1.3 Flowchart Create



Gambar 1.4 Flowchart Read dan Sorting



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 tipe data Struct untuk penyimpanan lebih efisien.

3. Source Code

A. Fitur Login

Fitur ini berfungsi untuk mengamankan program dengan membutuhkan kredensial user.

```
bool login() {
    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";
            return true;
        }
        else if (percobaanLogin < 2) {
            cout << "Username atau Password salah! Silahkan coba lagi.\n";
            percobaanLogin++;
        } else {
            percobaanLogin++;
        }
    }

    cout << "Maaf! Terlalu banyak percobaan gagal. Program akan segera berhenti.\n";
    return false;
}
```

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. Program menggunakan switch case untuk pengalihan fitur lebih efisien.

```
int main() {
    if (!login()) return 0;

    Obat daftarObat[100];
    int jumlah = 0, pilihan;

    do {
        cout << "\n--- Manajemen 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;
    cin.ignore();

    switch (pilihan) {
        case 1:
            tambahObat(&daftarObat[jumlah]);
            jumlah++;
            break;
        case 2:
            tampilkanObat(daftarObat, jumlah);
            break;
        case 3:
            ubahObat(daftarObat, jumlah);
            break;
        case 4:
            hapusObat(daftarObat, jumlah);
            break;
        case 5:
            cout << "\nTerima kasih telah menggunakan program ini.\n";
            break;
        default:
            cout << "\nPilihan tidak valid! Silahkan coba lagi.\n";
    }
    while (pilihan != 5);

    return 0;
}

```

C. Fitur Create Data

Fitur ini berfungsi untuk menambahkan data ke dalam daftar data.

```

void tambahObat(Obat* obat) {
    cout << "\nMasukkan Nama Obat: "; getline(cin, obat->nama);
    cout << "Masukkan Jenis Obat: "; getline(cin, obat->jenis);
    cout << "Masukkan Harga Obat: "; cin >> obat->harga;
    cout << "Masukkan Stok Obat: "; cin >> obat->stok;
    cin.ignore();
    cout << "Obat berhasil ditambahkan!\n";
}

```


D. Fitur Read Data

Fitur ini berfungsi untuk menampilkan isi dari daftar data jika ada isi di dalamnya. Jika nama obat tidak ada, maka user akan langsung dikembalikan ke menu. Fitur ini juga menggunakan beberapa algoritma sorting untuk mengurut data sesuai kebutuhan user.

```
void tampilkanObat(Obat daftarObat[], int jumlah) {
    if (jumlah == 0) {
        cout << "Tidak ada obat dalam sistem.\n";
        return;
    }

    int pilihanTampil;
    do {
        cout << "\n--- Menu Tampilkan Obat ---\n";
        cout << "1. Tampilkan default (tanpa sorting)\n";
        cout << "2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)\n";
        cout << "3. Tampilkan berdasarkan Stok (descending - Selection Sort)\n";
        cout << "4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)\n";
        cout << "5. Kembali ke menu utama\n";
        cout << "Pilih: ";
        cin >> pilihanTampil;
        cin.ignore();

        Obat salinanObat[100];
        for (int i = 0; i < jumlah; i++) {
            salinanObat[i] = daftarObat[i];
        }

        switch (pilihanTampil) {
            case 1:
                cout << "\nDaftar Obat (Default):\n";
                break;
            case 2:
                bubbleSortNama(salinanObat, jumlah);
                cout << "\nDaftar Obat (Nama Ascending):\n";
                break;
            case 3:
                selectionSortStokDescending(salinanObat, jumlah);
                cout << "\nDaftar Obat (Stok Descending):\n";
                break;
            case 4:
                insertionSortHarga(salinanObat, jumlah);
                cout << "\nDaftar Obat (Harga Ascending):\n";
                break;
            case 5:
                cout << "Kembali ke menu utama.\n";
                return;
            default:
                cout << "Pilihan tidak valid.\n";
                continue;
        }
    } while (true);
}
```

```

    }

    for (int i = 0; i < jumlah; i++) {
        cout << "Nama: " << salinanObat[i].nama
            << ", Jenis: " << salinanObat[i].jenis
            << ", Harga: " << salinanObat[i].harga
            << ", Stok: " << salinanObat[i].stok << "\n";
    }

    } while (pilihanTampil != 5);
}

```

E. Fitur Update Data

Fitur ini berfungsi untuk memperbarui data yang telah berubah. Jika nama obat tidak ada, maka user akan langsung dikembalikan ke menu.

```

void ubahObat(Obat daftarObat[], int jumlah) {
    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);
            cout << "Masukkan Jenis Baru: "; getline(cin, daftarObat[i].jenis);
            cout << "Masukkan Harga Baru: "; cin >> daftarObat[i].harga;
            cout << "Masukkan Stok Baru: "; cin >> daftarObat[i].stok;
            cin.ignore();
            cout << "Obat berhasil diperbarui!\n";
            ditemukan = true;
            break;
        }
    }
    if (!ditemukan) {
        cout << "Nama obat tidak ditemukan!\n";
    }
}

```

F. Fitur Hapus Data

Fitur ini berfungsi untuk menghapus data jika tidak dibutuhkan lagi. Jika nama obat tidak ada, maka user akan langsung dikembalikan ke menu.

```
void hapusObat(Obat daftarObat[], int &jumlah) {
    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";
    }
}
```

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: 2

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 1

Daftar Obat (Default):
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
```

Gambar 4.1 Demonstrasi Login, Menu, dan Data sebelum dimanipulasi

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

Masukkan Nama Obat: Aspirin-PT6
Masukkan Jenis Obat: Analgesik
Masukkan Harga Obat: 16000
Masukkan Stok Obat: 80
Obat berhasil ditambahkan!
```

Gambar 4.2 Demonstrasi Create

```

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 1

Daftar Obat (Default):
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
Nama: Aspirin-PT6, Jenis: Analgesik, Harga: 16000, Stok: 80

```

Gambar 4.3 Demonstrasi Hasil Create

```

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 2

Daftar Obat (Nama Ascending):
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Aspirin-PT6, Jenis: Analgesik, Harga: 16000, Stok: 80
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55

```

Gambar 4.4 Demonstrasi Sorting berdasarkan Nama secara ascending

```

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 3

Daftar Obat (Stok Descending):
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Aspirin-PT6, Jenis: Analgesik, Harga: 16000, Stok: 80
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30

```

Gambar 4.5 Demonstrasi Sorting berdasarkan Stok secara descending

```

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 4

Daftar Obat (Harga Ascending):
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Aspirin-PT6, Jenis: Analgesik, Harga: 16000, Stok: 80
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40

```

Gambar 4.6 Demonstrasi Sorting berdasarkan Harga secara ascending

```

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

Masukkan Nama Obat yang ingin diubah: Aspirin-PT6
Masukkan Nama Baru: Azithromycin-PT6
Masukkan Jenis Baru: Antibiotik-PT6
Masukkan Harga Baru: 35000
Masukkan Stok Baru: 45
Obat berhasil diperbarui!

```

Gambar 4.7 Demonstrasi Update

```

--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 1

Daftar Obat (Default):
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
Nama: Azithromycin-PT6, Jenis: Antibiotik-PT6, Harga: 35000, Stok: 45

```

Gambar 4.8 Demonstrasi Hasil Update

```
--- 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: Azithromycin-PT6
Obat berhasil dihapus!
```

Gambar 4.9 Demonstrasi Delete

```
--- Menu Tampilkan Obat ---
1. Tampilkan default (tanpa sorting)
2. Tampilkan berdasarkan Nama (ascending - Bubble Sort)
3. Tampilkan berdasarkan Stok (descending - Selection Sort)
4. Tampilkan berdasarkan Harga (ascending - Insertion Sort)
5. Kembali ke menu utama
Pilih: 1

Daftar Obat (Default):
Nama: Paracetamol, Jenis: Analgesik, Harga: 20000, Stok: 100
Nama: Amoxicillin, Jenis: Antibiotik, Harga: 15000, Stok: 50
Nama: Cetirizine, Jenis: Antihistamin, Harga: 12000, Stok: 70
Nama: Metformin, Jenis: Antidiabetik, Harga: 30000, Stok: 40
Nama: Omeprazole, Jenis: Antasida, Harga: 25000, Stok: 60
Nama: Ibuprofen, Jenis: Analgesik, Harga: 18000, Stok: 90
Nama: Salbutamol, Jenis: Bronkodilator, Harga: 22000, Stok: 30
Nama: Simvastatin, Jenis: Antikolesterol, Harga: 28000, Stok: 55
Nama: Loratadine, Jenis: Antihistamin, Harga: 14000, Stok: 65
Nama: Ciprofloxacin, Jenis: Antibiotik, Harga: 27000, Stok: 35
```

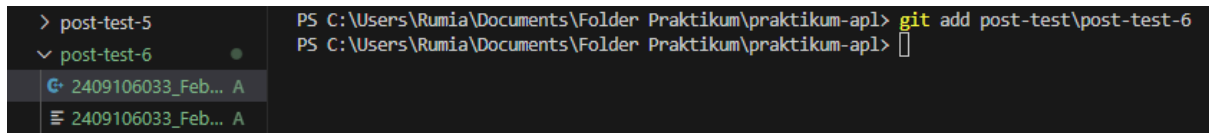
Gambar 4.10 Demonstrasi Hasil Update

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.



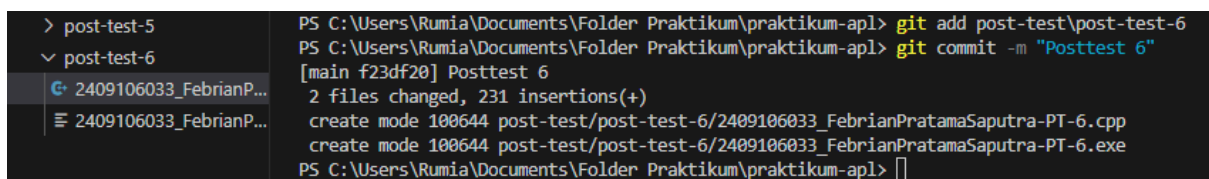
```
> post-test-5
└─ post-test-6
   └─ 2409106033_Feb... A
      └─ 2409106033_Feb... A

PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git add post-test\post-test-6
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.



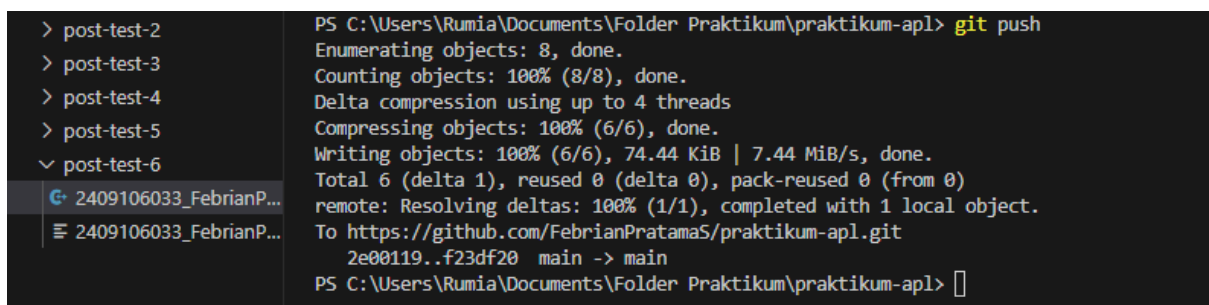
```
> post-test-5
└─ post-test-6
   └─ 2409106033_FebrianP...
      └─ 2409106033_FebrianP...

PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git add post-test\post-test-6
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git commit -m "Posttest 6"
[main f23df20] Posttest 6
2 files changed, 231 insertions(+)
create mode 100644 post-test/post-test-6/2409106033_FebrianPratamaSaputra-PT-6.cpp
create mode 100644 post-test/post-test-6/2409106033_FebrianPratamaSaputra-PT-6.exe
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.



```
> post-test-2
> post-test-3
> post-test-4
> post-test-5
└─ post-test-6
   └─ 2409106033_FebrianP...
      └─ 2409106033_FebrianP...

PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl> git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 74.44 KiB | 7.44 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/FebrianPratamaS/praktikum-apl.git
2e00119..f23df20 main -> main
PS C:\Users\Rumia\Documents\Folder Praktikum\praktikum-apl>
```

Gambar 5.3 Git Push