

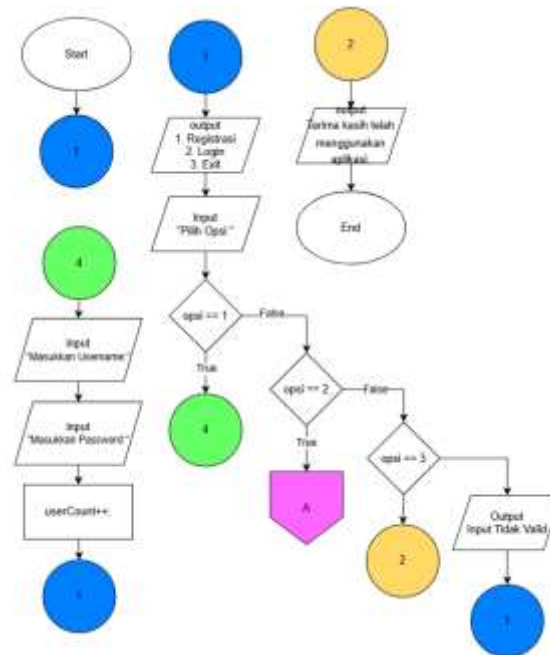
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



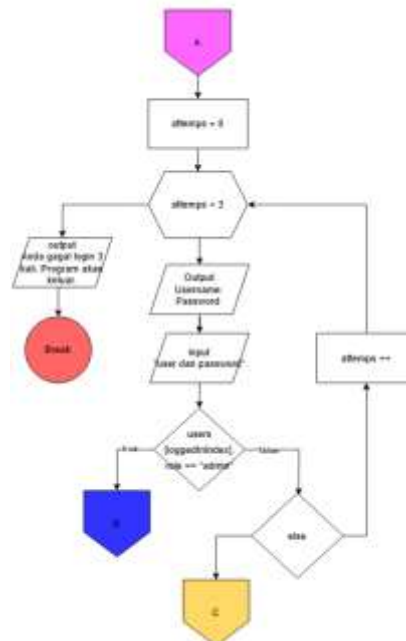
**Disusun oleh:**  
**Ikhsan (2409106118)**  
**Kelas (C2 '24)**

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

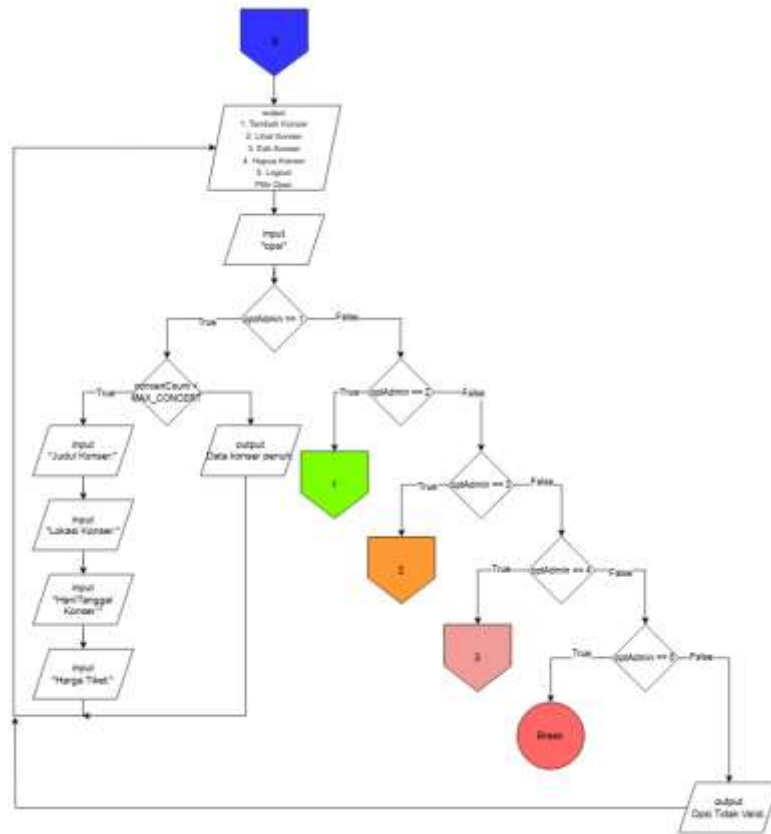
## 1. Flowchart



Gambar 1.1 Flowchart



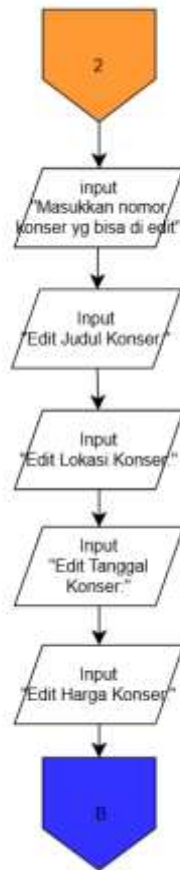
Gambar 1.2 Flowchart



Gambar 1.3 Flowchart



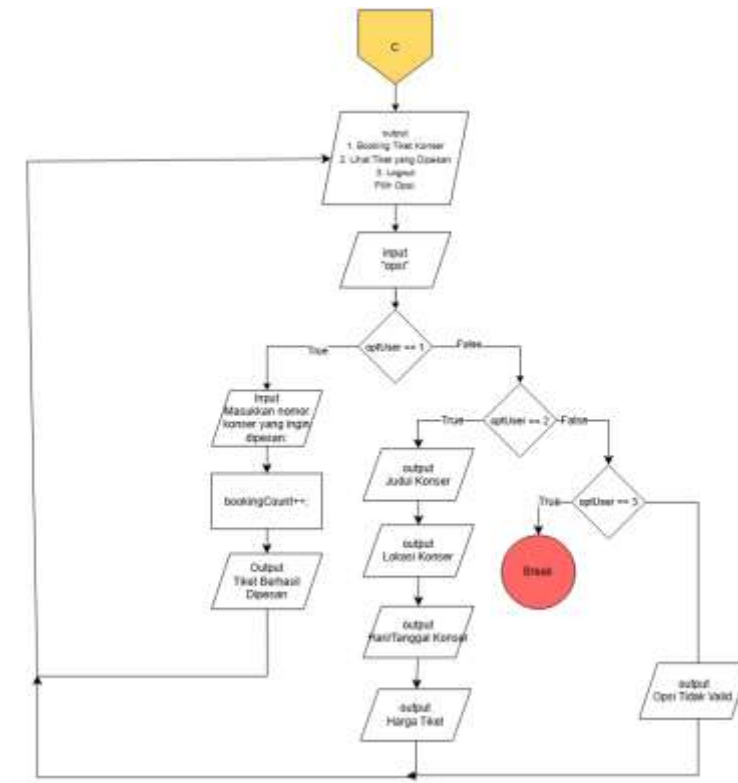
Gambar 1.4 Flowchart



Gambar 1.5 Flowchart



Gambar 1.6 Flowchart



Gambar 1.7 Flowchart

## 2. Analisis Program

### 2.1 Tujuan dan Fungsi Program

Program ini dibuat untuk mengelola pemesanan tiket konser secara sederhana dengan fitur:

1. **Registrasi & Login Pengguna**
  - Pengguna dapat mendaftar dan masuk ke sistem.
  - Tersedia dua jenis akun: **admin** (untuk manajemen data konser) dan **user** (untuk pemesanan tiket).
2. **Fitur Admin**
  - Menambahkan konser baru.
  - Menampilkan daftar konser dalam bentuk tabel.
  - Mengedit data konser yang sudah ada.
  - Menghapus konser.
3. **Fitur Pengguna**
  - Menampilkan daftar konser dalam bentuk tabel.
  - Memesan tiket konser.
  - Menampilkan daftar tiket yang telah dipesan dalam bentuk tabel.

### 2.2 Manfaat Utama

- **Mudah Digunakan:** Antarmuka berbasis teks yang sederhana, dengan menu yang jelas.
- **Manajemen Tiket Efektif:** Admin dapat mengelola data konser secara penuh, sedangkan pengguna cukup memesan dan melihat tiketnya.
- **Keamanan:** Mekanisme login dibatasi tiga kali percobaan untuk mencegah akses tidak sah.
- **Penyimpanan Data Sementara:** Semua data disimpan dalam array selama program berjalan, cocok untuk skala kecil atau simulasi.

### 2.3 Desain Modularitas dan Alur Program

Untuk menjaga **keterbacaan** dan **pemeliharaan** kode, setiap fitur menu dipisahkan menjadi **subprogram** (fungsi atau prosedur) yang spesifik:

- **registrasi(...):** Prosedur untuk pendaftaran pengguna baru.
- **login(...):** Fungsi yang mengembalikan indeks pengguna jika berhasil login, atau -1 jika gagal.
- **tambahKonser(...), lihatKonser(...), editKonser(...), hapusKonser(...):** Prosedur admin untuk manajemen konser.
- **bookingTiket(...), lihatBooking(...):** Prosedur user untuk pemesanan dan tampilan tiket.

Setiap subprogram **menerima parameter** (misalnya array data dan variabel jumlah elemen) sehingga tidak bergantung pada variabel global. Prosedur bertipe void digunakan untuk aksi yang tidak perlu mengembalikan nilai, sedangkan fungsi (login) mengembalikan nilai penting (indeks pengguna).

## 2.4 Nilai Tambah: Rekursi dan Overloading

- **Fungsi Rekursif**

Kami menggunakan `tampilkanTabelKonserRekursif(...)` untuk mencetak tabel konser satu baris per panggilan fungsi hingga semua baris tercetak. Ini memperlihatkan penerapan rekursi dalam konteks iterasi data.

- **Fungsi Overloading**

Dua versi `tampilkanPesan(...)` disediakan:

```
void tampilkanPesan(const string &msg);  
void tampilkanPesan(const string &msg, int nilai);
```

Sehingga kita bisa memanggil `tampilkanPesan("Sisa percobaan: ", sisa);` atau `tampilkanPesan("Login berhasil!");` dengan antarmuka yang sama.

## 3. Source Code

```
#include <iostream>  
using namespace std;  
  
#define MAKS_KONSER 100  
#define MAKS_BOOKING 100  
#define MAKS_USER 100  
  
struct Konser {  
    string judul, lokasi, tanggal, harga;  
};  
  
struct Booking {  
    Konser konser;  
};  
  
struct Pengguna {  
    string nama, sandi, peran;  
    int jumlahBooking;  
    Booking daftarBooking[MAKS_BOOKING];  
};  
  
// Overloading tampilkanPesan  
void tampilkanPesan(const string &pesan) {  
    cout << pesan << "\n";  
}  
void tampilkanPesan(const string &pesan, int x) {  
    cout << pesan << x << "\n";  
}
```





```

        cout << "+-----+";
    }
    if (idx >= jumlah) {
        cout << "+-----+";
    }
    return;
}
auto &k = konser[idx];
printf("| %2d | %-20s | %-12s | %-15s | %8s |\n",
        idx+1, k.judul.c_str(), k.lokasi.c_str(), k.tanggal.c_str(),
        k.harga.c_str());
tampilkanTabelKonserRekursif(konser, idx+1, jumlah);
}

// Prosedur admin
void tambahKonser(Konser konser[], int &jumlah) {
    if (jumlah>=MAKS_KONSER) {
        tampilkanPesan("Data konser penuh.");
        return;
    }
    tampilkanPesan("\n=== Tambah Konser ===");
    cin.ignore();
    cout << "Judul Konser: ";    getline(cin, konser[jumlah].judul);
    cout << "Lokasi Konser: ";   getline(cin, konser[jumlah].lokasi);
    cout << "Tanggal Konser: ";  getline(cin, konser[jumlah].tanggal);
    cout << "Harga Tiket: ";     getline(cin, konser[jumlah].harga);
    jumlah++;
    tampilkanPesan("Konser berhasil ditambahkan.");
}

void lihatKonser(const Konser konser[], int jumlah) {
    tampilkanTabelKonserRekursif(konser, 0, jumlah);
}

void editKonser(Konser konser[], int jumlah) {
    if (!jumlah) { tampilkanPesan("Tidak ada konser untuk diedit."); return; }

    lihatKonser(konser, jumlah);
    cout << "Pilih nomor konser: "; int no; cin>>no; cin.ignore();
    if (no<1||no>jumlah) { tampilkanPesan("Nomor tidak valid."); return; }
    int i=no-1;
    cout<<"Edit Judul ("<<konser[i].judul<<"): "; getline(cin,
konser[i].judul);
    cout<<"Edit Lokasi ("<<konser[i].lokasi<<"): "; getline(cin,
konser[i].lokasi);
    cout<<"Edit Tanggal ("<<konser[i].tanggal<<"): "; getline(cin,
konser[i].tanggal);
}

```

```

        cout<<"Edit Harga ("<<konser[i].harga<<"): "; getline(cin,
konser[i].harga);
        tampilkanPesan("Konser berhasil diubah.");
    }

void hapusKonser(Konser konser[], int &jumlah) {
    if (!jumlah) { tampilkanPesan("Tidak ada konser untuk dihapus.");
return; }
    lihatKonser(konser, jumlah);
    cout<<"Pilih nomor konser: "; int no; cin>>no;
    if (no<1||no>jumlah) { tampilkanPesan("Nomor tidak valid."); return; }
    for(int i=no-1;i<jumlah-1;i++) konser[i]=konser[i+1];
    jumlah--; tampilkanPesan("Konser berhasil dihapus.");
}

// Prosedur user
void bookingTiket(Pengguna &u, const Konser konser[], int jumlah) {
    lihatKonser(konser, jumlah);
    cout<<"Pilih nomor konser: "; int no; cin>>no;
    if(no<1||no>jumlah){ tampilkanPesan("Nomor tidak valid."); return; }
    u.daftarBooking[u.jumlahBooking++].konser = konser[no-1];
    tampilkanPesan("Tiket berhasil dipesan.");
}

void lihatBooking(const Pengguna &u) {
    if (!u.jumlahBooking) { tampilkanPesan("Anda belum memesan tiket.");
return; }
    cout << "+----+-----+-----+-----+-----+-----+-----+-----+
-----+\n";
    cout << "| No | Judul | Lokasi | Tanggal |
Harga | \n";
    cout << "+----+-----+-----+-----+-----+-----+-----+-----+
-----+\n";
    for(int i=0;i<u.jumlahBooking;i++){
        auto &k=u.daftarBooking[i].konser;
        printf("| %2d | %-20s | %-12s | %-15s | %8s | \n",
            i+1, k.judul.c_str(), k.lokasi.c_str(), k.tanggal.c_str(),
k.harga.c_str());
    }
    cout << "+----+-----+-----+-----+-----+-----+-----+-----+
-----+\n";
}

// MAIN
int main(){
    Konser konser[MAKS_KONSER] = {
        {"Midnigh Serenade", "Samarinda", "11-Oktober-2024", "200000"},
        {"Arctic Monkey", "Balikpapan", "12-Oktober-2024", "500000"}
    }
}

```

```

};
int jumlahKonser = 2;

Pengguna pengguna[MAKS_USER];
int jumlahUser = 2;
pengguna[0] = {"admin","admin","admin",0};
pengguna[1] = {"Ikhsan","2409106118","user",0};

while(true){
    tampilkanPesan("\n=== Pemesanan Tiket Konser ===");
    tampilkanPesan("1. Register");
    tampilkanPesan("2. Login");
    tampilkanPesan("3. Exit");
    tampilkanPesan("Pilih opsi: ");
    int opsi; cin>>opsi;
    if(opsi==1) registrasi(pengguna, jumlahUser);
    else if(opsi==2){
        int idx = login(pengguna, jumlahUser);
        if(idx<0) break;
        if(pengguna[idx].peran=="admin"){
            while(true){
                tampilkanPesan("\n-- Menu Admin --");
                tampilkanPesan("1.Tambah");
                tampilkanPesan("2.Lihat");
                tampilkanPesan("3.Edit");
                tampilkanPesan("4.Hapus");
                tampilkanPesan("5.Logout");
                tampilkanPesan("Pilih: ");
                int o; cin>>o;
                if(o==1) tambahKonser(konser, jumlahKonser);
                else if(o==2) lihatKonser(konser, jumlahKonser);
                else if(o==3) editKonser(konser, jumlahKonser);
                else if(o==4) hapusKonser(konser, jumlahKonser);
                else if(o==5) break;
            }
        } else {
            while(true){
                tampilkanPesan("\n-- Menu User --");
                tampilkanPesan("1.Booking Tiket Konser");
                tampilkanPesan("2.Lihat Tiket yang Dipesan");
                tampilkanPesan("3.Logout");
                tampilkanPesan("Pilih: ");
                int o; cin>>o;
                if(o==1) bookingTiket(pengguna[idx], konser,
jumlahKonser);
                else if(o==2) lihatBooking(pengguna[idx]);
                else if(o==3) break;
            }
        }
    }
}

```

```
    }  
    } else break;  
}  
return 0;  
}
```

#### 4. Uji Coba dan Hasil Output

```
=== Pemesanan Tiket Konser ===
1. Register
2. Login
3. Exit
Pilih opsi:
3. Exit
Pilih opsi:
1

=== Registrasi Pengguna Baru ===
Masukkan nama pengguna: Jamal
Masukkan kata sandi: 12345678
Registrasi berhasil! Silakan login.

=== Pemesanan Tiket Konser ===
1. Register
2. Login
3. Exit
Pilih opsi:
2
Nama pengguna: Jamal
Kata sandi: 12345678
Login berhasil!

-- Menu User --
1.Booking Tiket Konser
2.Lihat Tiket yang Dipesan
3.Logout
Pilih:
1
+-----+-----+-----+-----+
| No | Judul                | Lokasi      | Tanggal        | Harga    |
+-----+-----+-----+-----+
| 1  | Midnigth Serenade    | Samarinda   | 11-Oktober-2024 | 200000   |
| 2  | Arctic Monkey        | Balikpapan  | 12-Oktober-2024 | 500000   |
+-----+-----+-----+-----+
Pilih nomor konser: █
```

Gambar 4.1 Output Regis, Login dan Beli Tiket

```
-- Menu User --
1.Booking Tiket Konser
2.Lihat Tiket yang Dipesan
3.Logout
Pilih:
2
```

No	Judul	Lokasi	Tanggal	Harga
1	Arctic Monkey	Balikpapan	12-Oktober-2024	500000

```
-- Menu User --
1.Booking Tiket Konser
2.Lihat Tiket yang Dipesan
3.Logout
Pilih:
3

=== Pemesanan Tiket Konser ===
1. Register
2. Login
3. Exit
Pilih opsi:

```

Gambar 4.2 Output Lihat Tiket Konser Acc Baru

```
-- Menu Admin --
1.Tambah
2.Lihat
3.Edit
4.Hapus
5.Logout
Pilih:
1

=== Tambah Konser ===
Judul Konser: FUR
Lokasi Konser: Tanah Grogot
Tanggal Konser: 17-Agustus-2026
Harga Tiket: 150000
Konser berhasil ditambahkan.
```

Gambar 4.3 Output Admin Tambah Konser

```

-- Menu Admin --
1.Tambah
2.Lihat
3.Edit
4.Hapus
5.Logout
Pilih:
    2

+---+-----+-----+-----+-----+
| No | Judul           | Lokasi    | Tanggal       | Harga    |
+---+-----+-----+-----+-----+
| 1  | Midnigth Serenade | Samarinda | 11-Oktober-2024 | 200000   |
| 2  | Arctic Monkey     | Balikpapan | 12-Oktober-2024 | 500000   |
| 3  | FUR                | Tanah Grogot | 17-Agustus-2026 | 150000   |
+---+-----+-----+-----+-----+

-- Menu Admin --
1.Tambah
2.Lihat
3.Edit
4.Hapus
5.Logout
Pilih:
    3

+---+-----+-----+-----+-----+
| No | Judul           | Lokasi    | Tanggal       | Harga    |
+---+-----+-----+-----+-----+
| 1  | Midnigth Serenade | Samarinda | 11-Oktober-2024 | 200000   |
| 2  | Arctic Monkey     | Balikpapan | 12-Oktober-2024 | 500000   |
| 3  | FUR                | Tanah Grogot | 17-Agustus-2026 | 150000   |
+---+-----+-----+-----+-----+

Pilih nomor konser: 3
Edit Judul (FUR): Wali
Edit Lokasi (Tanah Grogot): Banjarmasin
Edit Tanggal (17-Agustus-2026): 17-Agustus-2026
Edit Harga (150000): 190000
Konser berhasil diubah.

```

Gambar 4.4 Output Lihat Konser dan Edit

```
-- Menu Admin --
1.Tambah
2.Lihat
3.Edit
4.Hapus
5.Logout
Pilih:
4
+---+-----+-----+-----+-----+
| No | Judul                | Lokasi    | Tanggal    | Harga    |
+---+-----+-----+-----+-----+
| 1  | Midnighth Serenade   | Samarinda | 11-Oktober-2024 | 200000   |
| 2  | Arctic Monkey        | Balikpapan | 12-Oktober-2024 | 500000   |
| 3  | Wali                 | Banjarmasin | 17-Agustus-2026 | 190000   |
+---+-----+-----+-----+-----+
Pilih nomor konser: 3
Konser berhasil dihapus.

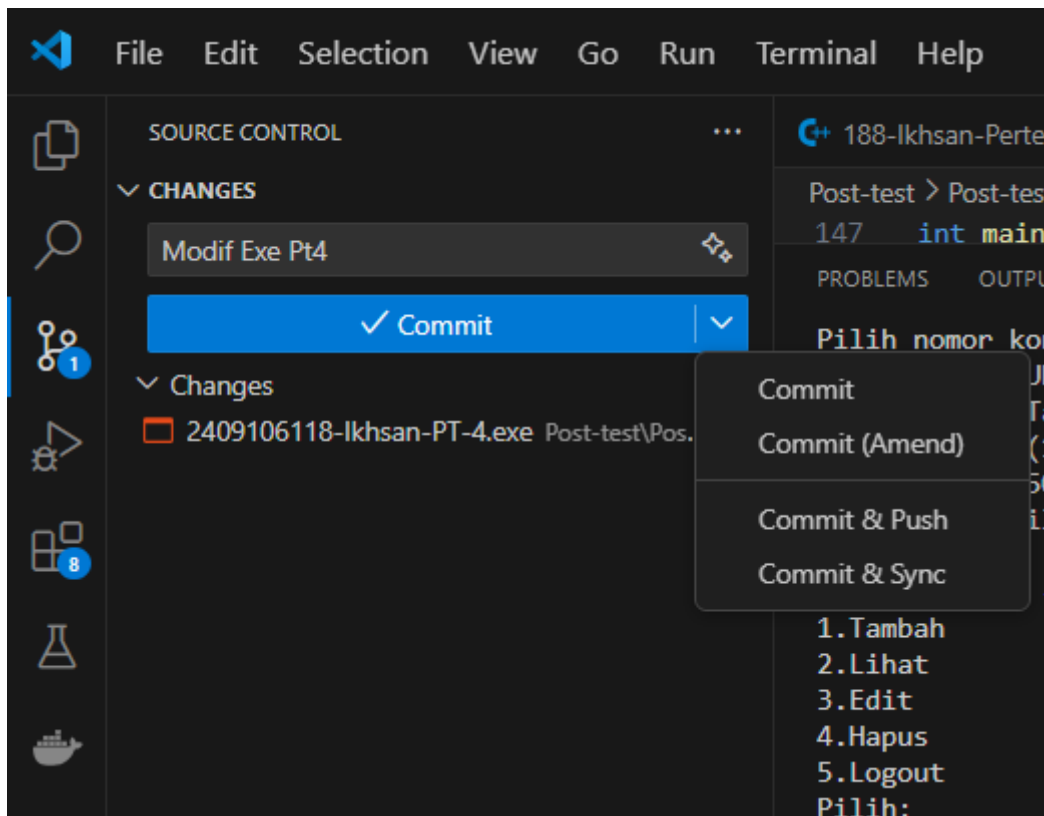
-- Menu Admin --
1.Tambah
2.Lihat
3.Edit
4.Hapus
5.Logout
Pilih:
5

=== Pemesanan Tiket Konser ===
1. Register
2. Login
3. Exit
Pilih opsi:
3
PS D:\GITHUB\Praktikum-Apl\Post-test\Post-test-4>
```

Gambar 4.5 Output Hapus hingga Exit

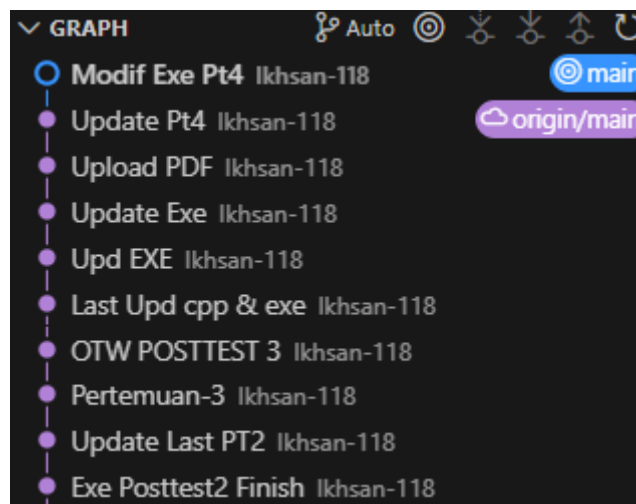


## 5. Langkah-Langkah Git pada VSCode



Gambar 5.1 Source Control Git Visual Code Studio

Pertama buka source control pada bagian sidebar di aplikasi Visual Code Studio, lalu input pesan yang ingin dipasang di message commit cintuhnya seperti punya saya “Update Exe”, lalu pencet tanda panah kebawah untuk mencari opsi “Commit & Push” agar programnya terupdate dan tersimpan di Github Cloud.



Gambar 5.2 Hasil apabila berhasil masuk Cloud