

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



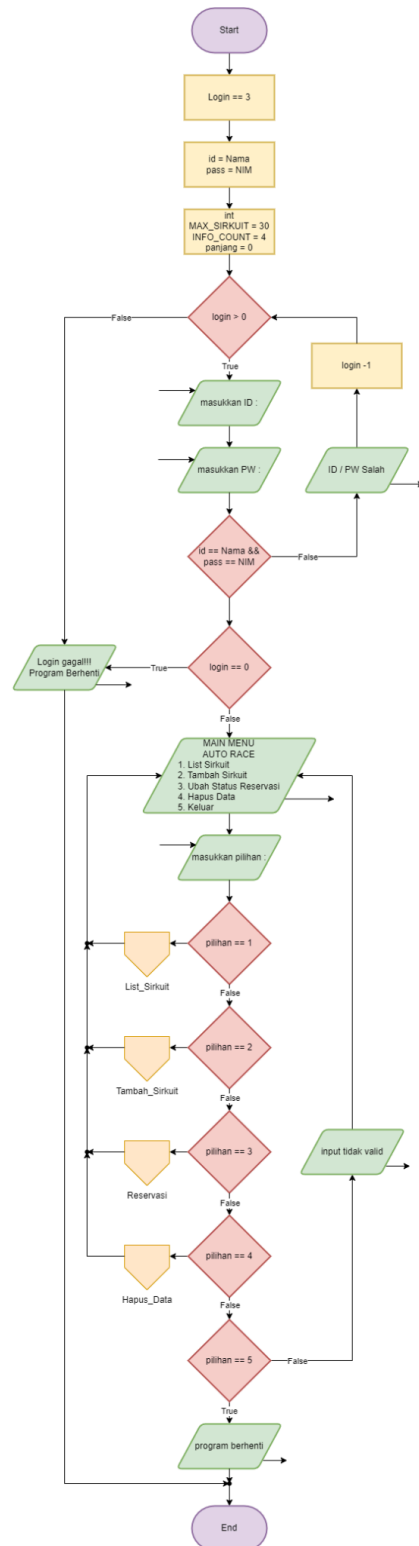
Disusun oleh:

Muhammad Ilma Yusrian Fahmi 2409106079

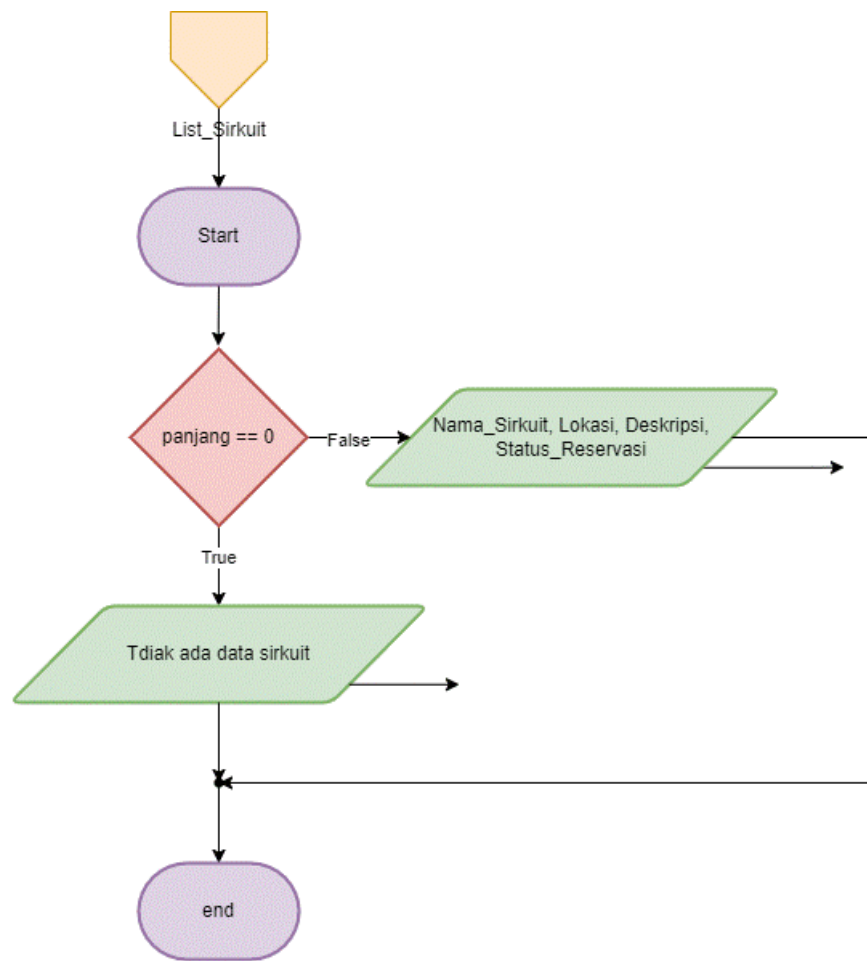
Kelas B2 2024

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

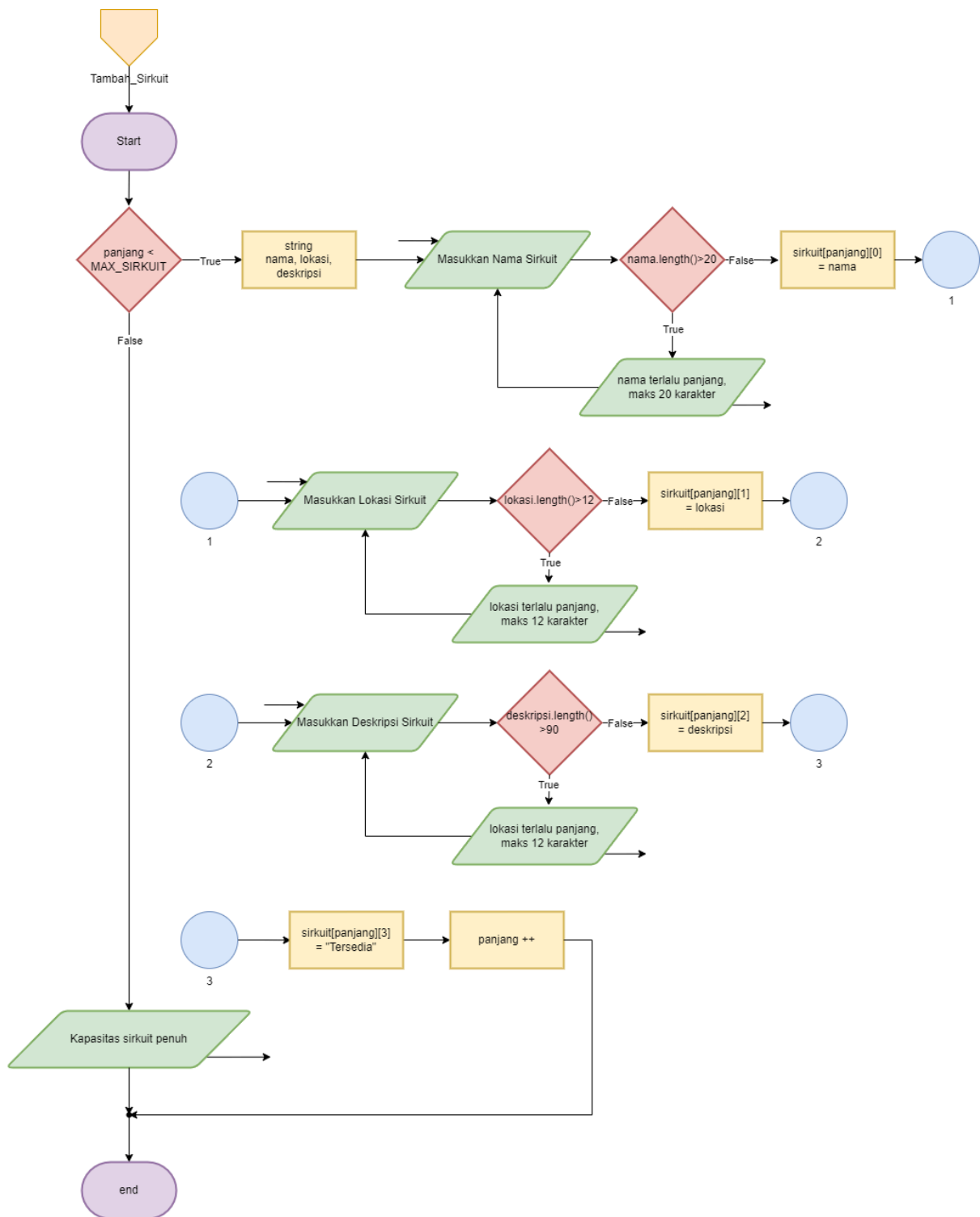
1. Flowchart



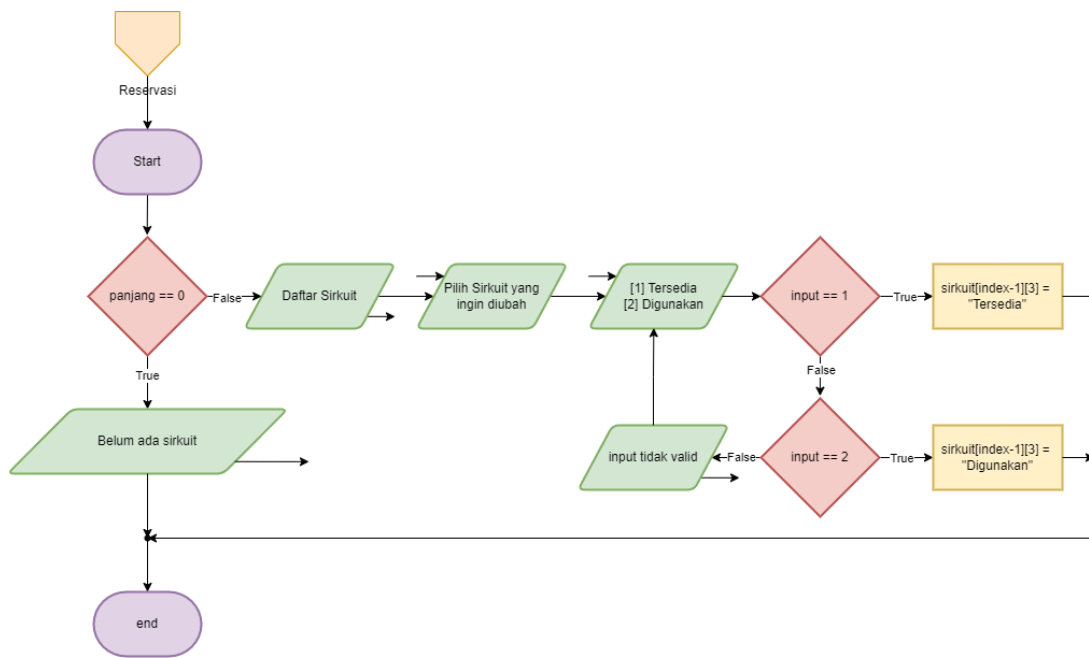
Gambar 1.1 Flowchart Login dan Main Menu



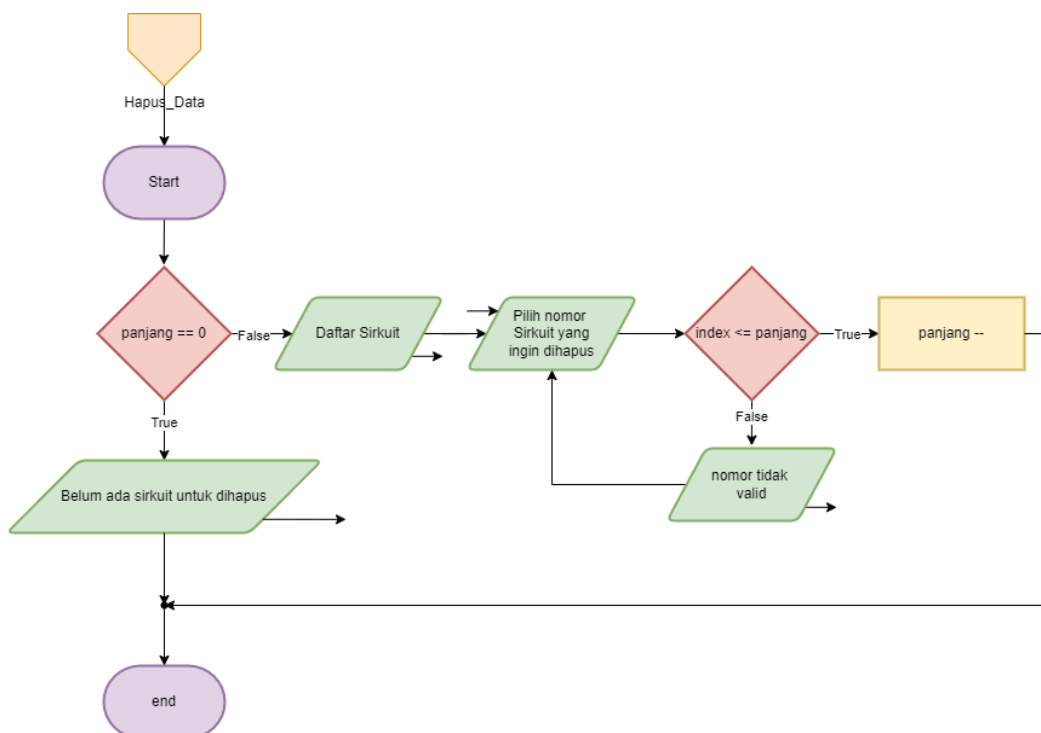
Gambar 1.2 Flowchart List Sirkuit



Gambar 1.3 Flowchart Tambah Sirkuit



Gambar 1.4 Flowchart Reservasi Sirkuit



Gambar 1.5 Flowchart Hapus Data Sirkuit

2. Analisis Program

2.1 Deskripsi Singkat Program

Program manajemen sirkuit ini dibuat untuk mempermudah pengguna dalam mengelola data sirkuit balap, seperti menambahkan, menampilkan, mengubah status reservasi, dan menghapus sirkuit. Program ini dilengkapi dengan fitur login untuk memastikan hanya pengguna yang memiliki akses yang dapat mengelola data, serta akan menghentikan program jika gagal login sebanyak tiga kali. Selain itu, sistem memastikan validasi input agar data yang dimasukkan sesuai, seperti memastikan deskripsi tidak melebihi batas karakter dan status reservasi hanya dapat diubah ke opsi yang valid. Namun, program ini masih memiliki kekurangan, yaitu belum adanya error handling jika pengguna menginput huruf ketika sistem meminta angka.

2.2 Penjelasan Alur & Algoritma

Alur Program

Ketika program dijalankan, pengguna pertama kali akan diminta untuk memasukkan username dan password sebagai langkah autentikasi. Jika pengguna salah memasukkan username atau password, program akan menampilkan pesan gagal login dan memberikan informasi mengenai sisa percobaan login yang tersedia. Program ini menggunakan sistem looping sederhana, di mana pengguna memiliki tiga kali percobaan login sebelum program secara otomatis berhenti.

Jika pengguna berhasil login, program akan menampilkan menu utama yang berisi beberapa pilihan untuk mengelola data sirkuit. Pengguna dapat memilih menu List Sirkuit, Tambah Sirkuit, Ubah Status Reservasi, Hapus Data dan Keluar.

Algoritma

1. Inisialisasi

- Program dimulai dengan deklarasi variabel utama seperti username, password, id, pass, serta array sirkuit untuk menyimpan informasi sirkuit.
- Variabel panjang digunakan untuk melacak jumlah sirkuit yang tersimpan.

- Variabel login digunakan untuk membatasi jumlah percobaan login.

2. Login

- Pengguna memiliki tiga kali kesempatan untuk memasukkan username dan password yang benar.
- Menggunakan looping while, program akan terus meminta input login hingga pengguna berhasil atau gagal tiga kali.
- Jika pengguna gagal login sebanyak tiga kali, program akan menampilkan pesan "Gagal Login! Program Terhenti." dan berhenti.
- Jika login berhasil, program akan menampilkan menu utama.

3. Menu Utama

- Setelah login berhasil, program menampilkan menu utama dengan lima opsi utama.
- Pengguna diminta untuk memasukkan angka sesuai dengan opsi menu.
- Program menggunakan switch-case untuk menangani pilihan pengguna.

4. Menu Tambah Sirkuit

- Program akan meminta input Nama Sirkuit, Lokasi Sirkuit, dan Deskripsi.
- Deskripsi dibatasi hingga 90 karakter agar tidak terlalu panjang.
- Status reservasi otomatis diatur ke "Tersedia" saat sirkuit ditambahkan.

5. Keluar dari Program

- Jika pengguna memilih opsi "Keluar", program akan menampilkan pesan "Program Berhenti" dan berhenti.

Penjelasan Blok Kode

- Looping While digunakan pada proses login agar pengguna hanya memiliki tiga kali kesempatan untuk memasukkan username dan password yang benar.
- If-Else pada Login digunakan untuk mengecek apakah pengguna masih memiliki kesempatan login atau tidak. Jika kesempatan habis, program akan berhenti.
- Switch-Case pada Menu Utama digunakan untuk menangani pilihan pengguna agar dapat mengakses fitur yang diinginkan.
- Looping dan Validasi Input digunakan saat menambahkan sirkuit atau mengubah status reservasi agar pengguna tidak memberikan input kosong atau tidak valid.
- Perulangan Do-While pada status reservasi memastikan pengguna hanya dapat memilih antara "Tersedia" dan "Digunakan", serta meminta ulang jika input salah.

3. Source Code

A. Fitur Login

Fitur ini digunakan untuk melakukan proses login, pengguna hanya memiliki kesempatan maksimal 3 kali, jika salah memasukkan username atau password sebanyak 3 kali maka program akan secara otomatis berhenti. Jika pengguna berhasil login maka program akan menampilkan beberapa menu konversi.

Source Code:

```
while (login > 0) {  
    system("cls");  
    cout << "=====\n";  
    cout << "          LOGIN          \n";  
    cout << "=====\n";  
    cout << "Masukkan Username: ";  
    getline (cin, username);  
    cout << "Masukkan Password: ";  
    getline (cin, password);  
  
    if (username == id && password == pass) {  
        cout << "\nLogin Berhasil!\n";  
        system("pause");  
        break;  
    }  
    else {  
        login--;  
        cout << "\nGagal! Percobaan tersisa: " << login << "\n";  
        system("pause");  
    }  
}  
  
if (login == 0) {  
    cout << "\nGagal Login! Program Terhenti.\n";  
    return 0;  
}
```

B. Main Menu

Program akan menampilkan Main Menu ketika pengguna berhasil login kemudian pengguna akan diminta untuk memasukkan pilihan menu apa yang ingin digunakan

```
int pilihan = 0;
while (true) {
    do {
        system("cls");
        cout << "=====\n";
        cout << "          MAIN MENU          \n";
        cout << "          AUTO RACE          \n";
        cout << "=====\n";
        cout << "1. List Sirkuit\n";
        cout << "2. Tambah Sirkuit\n";
        cout << "3. Ubah Status Reservasi\n";
        cout << "4. Hapus Data\n";
        cout << "5. Keluar\n";
        cout << "\nPilih menu: ";
        cin >> pilihan;
        cin.ignore();
    }
```

C. Menu List Sirkuit

Ketika pengguna memilih salah satu menu untuk menampilkan list sirkuit, program akan menampilkan list sirkuit yang ada, jika data sirkuit tidak tersedia maka program akan menampilkan output tidak ada data sirkuit, tetapi jika data tersedia maka program akan menampilkan dalam bentuk tabel.

```
case 1:
    system("cls");
    if (panjang == 0) {
        cout << "Tidak ada data sirkuit" << endl;
        system("pause");
    } else {
        cout << tableHeader;
```

```

    for (int i = 0; i < panjang; i++) {
        cout << " | " << setw(1) << i+1 << " | "
            << setw(22) << left << sirkuit[i][0] << " | "
            << setw(14) << left << sirkuit[i][1] << " | "
            << setw(90) << left << sirkuit[i][2] << " | "
            << setw(16) << left << sirkuit[i][3] << " |\n";
    }
    cout << tableFooter;
    system("pause");
}
break;

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Menambahkan list sirkuit
2. Menampilkan list sirkuit
3. Menghapus list sirkuit

4.2 Hasil Output

```
Pilih menu: 2
Masukkan Nama Sirkuit: Silverstone
Masukkan Lokasi Sirkuit: Inggris
Masukkan Deskripsi Sirkuit (max 100 karakter): Silverstone adalah sirkuit ikonik di Inggris, tuan rumah F1 sejak 1950

Sirkuit berhasil ditambahkan!
Press any key to continue . . .
```

Gambar 4.1 Menambah List Sirkuit

No	Nama Sirkuit	Lokasi	Deskripsi	Status Reservasi
1	Mandalika	Pulau Lombok	Sirkuit modern dengan pemandangan indah, memiliki lintasan cepat dan tikungan menantang	Digunakan
2	Internasional Sentul	Bogor	Sirkuit balap di Indonesia dengan kombinasi lintasan lurus dan tikungan teknika	Tersedia
3	Jaya Ancol	Jakarta	Sirkuit jalan raya yang menantang dengan tikungan tajam dan pemandangan perkotaan	Tersedia
4	Silverstone	Inggris	Silverstone adalah sirkuit ikonik di Inggris, tuan rumah F1 sejak 1950	Tersedia

Press any key to continue . . .

Gambar 4.2 Menampilkan List Sirkuit

```
===== Daftar Sirkuit =====
1. Mandalika (Pulau Lombok)
2. Internasional Sentul (Bogor)
3. Jaya Ancol (Jakarta)
4. Silverstone (Inggris)

Masukkan nomor sirkuit yang akan dihapus: 4
Sirkuit berhasil dihapus!
Press any key to continue . . .
```

Gambar 4.3 Menghapus List Sirkuit

5. GIT

5.1 Git Add

Git add digunakan untuk merubah status file yang diinginkan, file akan memiliki status Untracked apabila belum pernah di-add, dan akan berubah menjadi staged yang artinya file sudah di-add ke repository tetapi belum di-commit.

```
Lenovo@LAPTOP-1M0LURIM MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-2 (main)
$ git add .
```

Gambar 5.1 Git Add

5.2 Git Commit

Git commit berfungsi untuk menyimpan perubahan file di repository setelah ditambahkan, -m “...” untuk memberikan pesan yang diinginkan.

```
Lenovo@LAPTOP-1M0LURIM MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "upload cpp"
[main 63f8930] upload cpp
2 files changed, 238 insertions(+)
create mode 100644 post-test/post-test-2/2409106079-MuhammadILmaYusrianFahmi-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106079-MuhammadILmaYusrianFahmi-PT-2.exe
```

Gambar 5.2 Git Commit

5.3 Git Push

git push -u origin main digunakan untuk meng-upload perubahan yang ada pada repository lokal ke repository github

```
Lenovo@LAPTOP-1M0LURIM MINGW64 ~/Desktop/praktikum-apl/post-test/post-test-2 (main)
$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 680.14 KiB | 4.28 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MeFahme/post-test-apl.git
14505b2..63f8930 main -> main
```

Gambar 5.3 Git Push