LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

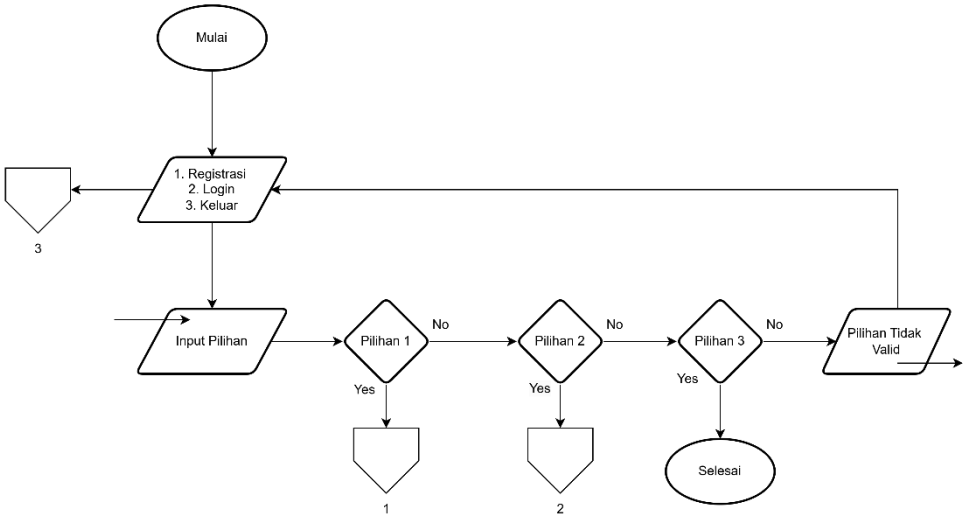
Disusun oleh:

Renaya Putri Alika (2409106002)

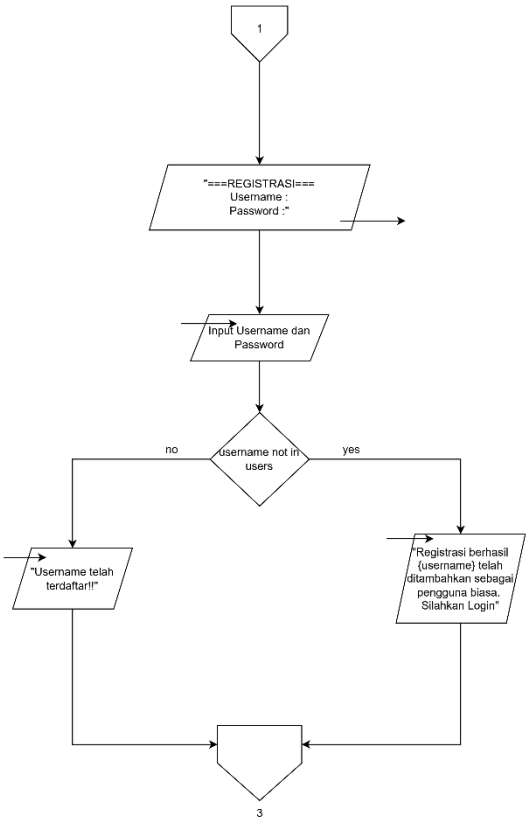
Kelas A1’24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN SAMARINDA
2025

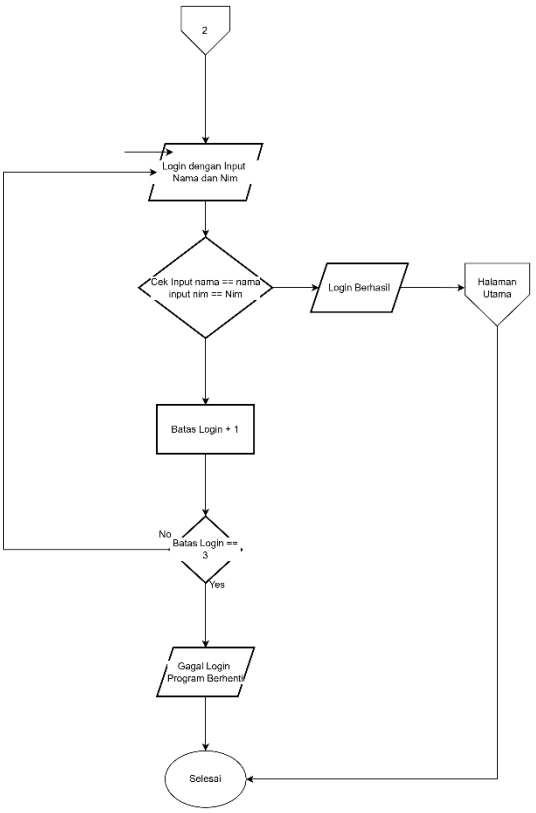
1. Flowchart



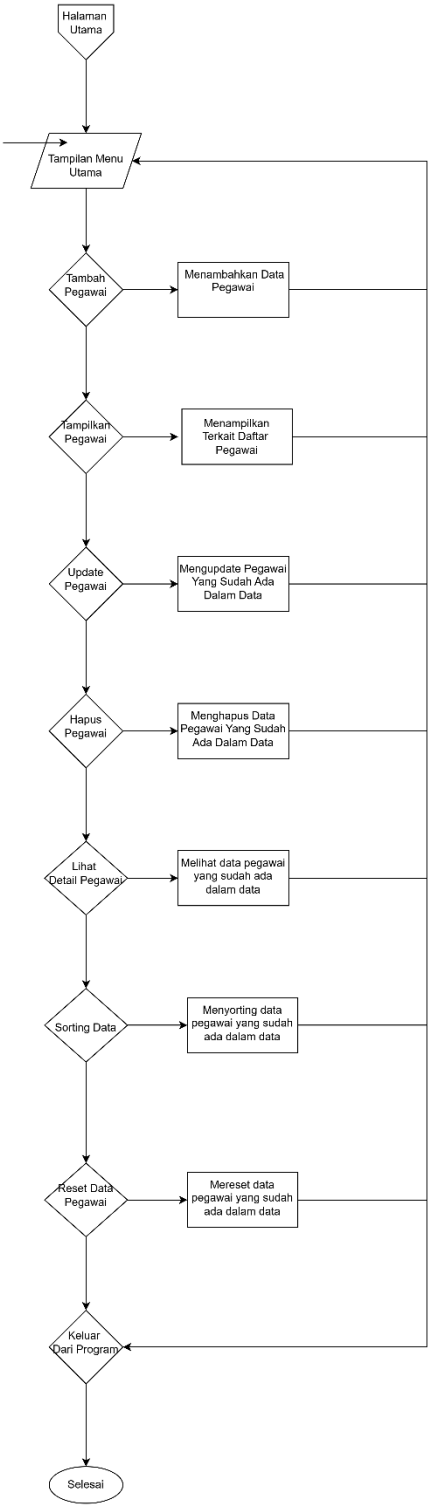
Flowchart 1.1 Pilihan Menu



Flowchart 1.2 Registrasi



Flowchart 1.3 Login



Flowchart 1.4 Halaman Utama

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini adalah sistem manajemen pegawai berbasis C++ yang memungkinkan pengguna untuk register akun baru, login, dan setelah berhasil login, mengelola data pegawai. Data pegawai mencakup nama, waktu masuk, waktu keluar, dan hasil kerja. Di dalam program, data user dan pegawai disimpan menggunakan array struct. Untuk login, sistem memverifikasi kecocokan nama dan NIM yang diregistrasi. Setelah login berhasil, tersedia berbagai fitur yaitu menambah pegawai baru, menampilkan data pegawai (baik secara keseluruhan maupun berdasarkan pencarian nama), mengupdate data pegawai, menghapus pegawai, melihat detail data pegawai, mengurutkan data berdasarkan kriteria tertentu, hingga mereset semua data.

Program juga menerapkan beberapa konsep penting seperti fungsi rekursif (saat menampilkan data pegawai), overloading fungsi (untuk menampilkan semua pegawai atau mencari berdasarkan nama), penggunaan pointer (saat mereset data), serta sorting manual. Struktur program dibuat agar setiap fitur dikelola dalam fungsi-fungsi terpisah untuk memudahkan pemeliharaan dan pengembangan.

3. Source Code

3.1 Struktur Data Utama

Untuk menyimpan data login pengguna dan Menyimpan data pegawai (nama, waktu masuk/keluar, dan hasil kerja).

```
struct Pegawai {
    string nama;
    string waktu_masuk;
    string waktu_keluar;
    string hasil_kerja;
};

struct User {
    string username;
    string nim;
};
```

3.2 Register & Login User

Menambahkan akun baru (maksimal 10 user) dan Cek username & NIM, user bisa login maksimal 3 percobaan.

```
void registerUser(User users[], int &jumlah_user) {
    if (jumlah_user < 10) {
        getline(cin, users[jumlah_user].username);
        getline(cin, users[jumlah_user].nim);
        jumlah_user++;
    }
}

bool loginUser(User users[], int jumlah_user, string &loggedInUser) {
    string username, nim;
    for (int attempt = 0; attempt < 3; attempt++) {
        getline(cin, username);
        getline(cin, nim);
        for (int i = 0; i < jumlah_user; i++) {
            if (users[i].username == username && users[i].nim == nim) {
                loggedInUser = username;
                return true;
            }
        }
    }
    exit(0); // keluar setelah gagal login 3x
}
```

3.3 Menu Manajemen Pegawai

Menyediakan pilihan CRUD data pegawai, Sorting, dan Reset data.

```
void menuPegawai(Pegawai pegawai[], int &jumlah_pegawai) {
    int pilihan;
    do {
        cout << "1. Tambah\n2. Tampilkan\n3. Update\n4. Hapus\n5. Lihat
Detail\n6. Sorting\n7. Reset\n8. Logout\n";
        cin >> pilihan; cin.ignore();
        switch (pilihan) {
            case 1: tambahPegawai(pegawai, jumlah_pegawai); break;
            case 2: tampilkanPegawai(pegawai, jumlah_pegawai); break;
            case 3: updatePegawai(pegawai, jumlah_pegawai); break;
            case 4: hapusPegawai(pegawai, jumlah_pegawai); break;
            case 5: cariDanTampilkanPegawai(pegawai, jumlah_pegawai); break;
            case 6: /* menu sorting dipanggil */ break;
            case 7: resetDataPegawai(pegawai, &jumlah_pegawai); break;
        }
    } while (pilihan != 8);
}
```

3.4 CRUD Data Pegawai

Menginput data baru ke array, Read atau menampilkan nama pegawai, Mengupdate data pegawai tertentu, dan Hapus atau geser array untuk menghapus satu pegawai.

```
void tambahPegawai(Pegawai pegawai[], int &jumlah_pegawai);
void tampilkanPegawaiRekursif(Pegawai pegawai[], int index, int
jumlah_pegawai);
void updatePegawai(Pegawai pegawai[], int jumlah_pegawai);
void hapusPegawai(Pegawai pegawai[], int &jumlah_pegawai);
```

3.5 Fungsi Overloading

Fungsi ini memiliki dua versi :

- Menampilkan semua pegawai
- Menampilkan hanya pegawai tertentu berdasarkan nama yang dicari.

```
void tampilkanPegawai(Pegawai pegawai[], int jumlah_pegawai); // overloading 1
void tampilkanPegawai(Pegawai pegawai[], int jumlah_pegawai, string nama); //
overloading 2
```

3.6 Sorting Data Pegawai

Pegawai bisa diurutkan berdasarkan : Nama, Waktu Masuk, atau Panjang Nama.

```
void sortNamaDescending(Pegawai pegawai[], int jumlah_pegawai) { /* nama Z-A */
}
void sortWaktuMasukAscending(Pegawai pegawai[], int jumlah_pegawai) { /* waktu
masuk naik */ }
void sortPanjangNamaAscending(Pegawai pegawai[], int jumlah_pegawai) { /*
panjang nama naik */ }
```

3.7 Reset Semua Data Pegawai

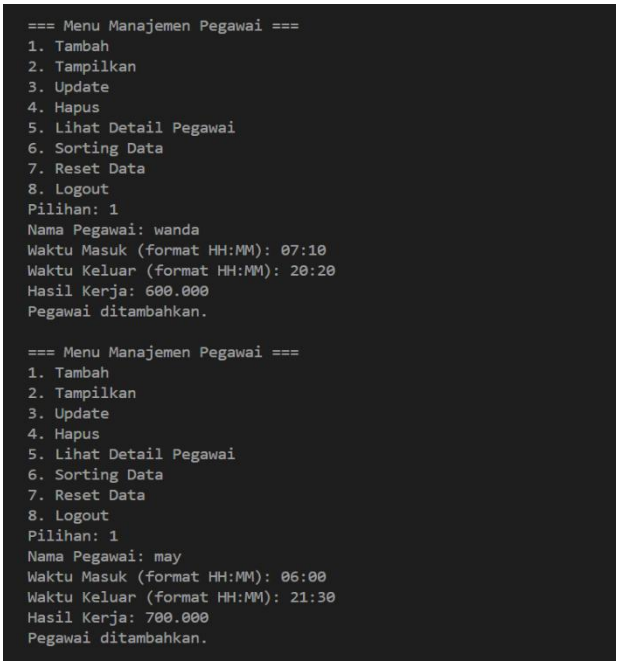
Untuk menghapus semua data pegawai dengan konfirmasi pengguna.

```
void resetDataPegawai(Pegawai *pegawai, int *jumlah_pegawai) {
    char konfirmasi;
    cin >> konfirmasi;
    if (konfirmasi == 'y' || konfirmasi == 'Y') {
        *jumlah_pegawai = 0;
    }
}
```

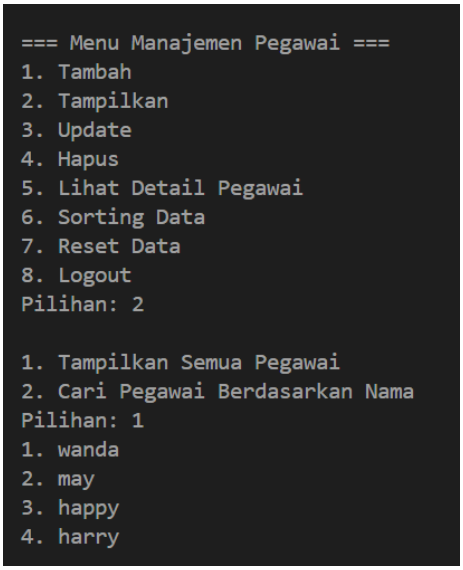
4 Uji Coba dan Hasil Output

```
=== Sistem Manajemen Pegawai ===
1. Register
2. Login
3. Keluar
Pilihan: 1

=== Register Akun ===
Masukkan Nama: Renaya
Masukkan NIM: 002
Registrasi berhasil! Silakan login.

=== Sistem Manajemen Pegawai ===
1. Register
2. Login
3. Keluar
Pilihan: 2

=== Login ===
Masukkan Nama: Renaya
Masukkan NIM: 002
```

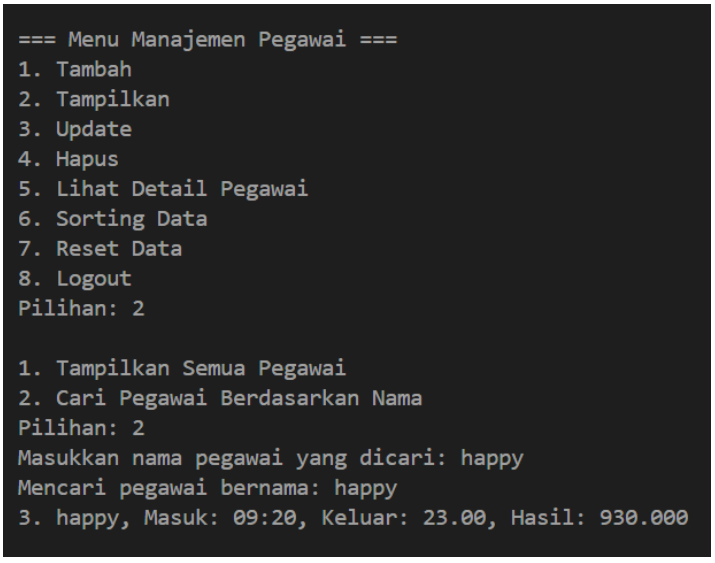
Output 4.1 Register & Login



Output 4.2 Tambah Pegawai (Create)



Output 4.3 Tampilkan Pegawai 1



Output 4.4 Tampilkan Pegawai 2

```
7. Reset Data
8. Logout
Pilihan: 3
1. wanda
2. may
3. happy
4. harry
Masukkan nomor pegawai: 4
Nama Baru: electra
Waktu Masuk Baru: 10.30
Waktu Keluar Baru: 23.40
Hasil Kerja Baru: 980.000
Data diperbarui.

=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 5

=== Detail Data Pegawai ===
No   Nama                Waktu Masuk    Waktu Keluar   Hasil Kerja
--------------------------------------------------------------------------------
1    wanda               07:10          20:20          600.000
2    may                 06:00          21:30          700.000
3    happy               09:20          23.00          930.000
4    electra             10.30          23.40          980.000
```

Output 4.5 Update Pegawai

```
=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 4
1. wanda
2. may
3. happy
4. electra
Masukkan nomor pegawai: 3
Data berhasil dihapus.
```

Output 4.6 Delete Pegawai

```
=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 6

=== Menu Sorting ===
1. Sort Nama (Z-A)
2. Sort Waktu Masuk (Ascending)
3. Sort Panjang Nama (Pendek ke Panjang)
Pilihan: 1
Data pegawai telah diurutkan berdasarkan nama (Z-A).
```

Output 4.7 Menu Sorting Data Pegawai

```
=== Menu Sorting ===
1. Sort Nama (Z-A)
2. Sort Waktu Masuk (Ascending)
3. Sort Panjang Nama (Pendek ke Panjang)
Pilihan: 1
Data pegawai telah diurutkan berdasarkan nama (Z-A).

=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 5

=== Detail Data Pegawai ===
No   Nama               Waktu Masuk    Waktu Keluar   Hasil Kerja
-------------------------------------------------------------------------
1    wanda              07:10          20:20          600.000
2    may                06:00          21:30          700.000
3    electra            10.30          23.40          980.000
```

Output 4.8 Sorting Nama Berdasarkan Z-A

```
=== Menu Sorting ===
1. Sort Nama (Z-A)
2. Sort Waktu Masuk (Ascending)
3. Sort Panjang Nama (Pendek ke Panjang)
Pilihan: 2
Data pegawai telah diurutkan berdasarkan waktu masuk.

=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 5

=== Detail Data Pegawai ===
No   Nama               Waktu Masuk    Waktu Keluar   Hasil Kerja
-------------------------------------------------------------------------
1    may                06:00          21:30          700.000
2    wanda              07:10          20:20          600.000
3    electra            10.30          23.40          980.000
```

Output 4.9 Sorting Ascending

```
=== Menu Sorting ===
1. Sort Nama (Z-A)
2. Sort Waktu Masuk (Ascending)
3. Sort Panjang Nama (Pendek ke Panjang)
Pilihan: 3
Data pegawai telah diurutkan berdasarkan panjang nama.

=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 5

=== Detail Data Pegawai ===
No   Nama              Waktu Masuk    Waktu Keluar   Hasil Kerja
-------------------------------------------------------------------------
1    may               06:00          21:30          700.000
2    wanda             07:10          20:20          600.000
3    electra           10.30          23.40          980.000
```

Output 4.10 Sorting Berdasarkan Nama (Pendek ke Panjang)

```
=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 7
Apakah Anda yakin ingin mereset semua data pegawai? (y/n): y
Seluruh data pegawai berhasil di-reset.
```

Output 4.11 Reset Data Pegawai

```
=== Menu Manajemen Pegawai ===
1. Tambah
2. Tampilkan
3. Update
4. Hapus
5. Lihat Detail Pegawai
6. Sorting Data
7. Reset Data
8. Logout
Pilihan: 8

=== Sistem Manajemen Pegawai ===
1. Register
2. Login
3. Keluar
Pilihan: 3
Program berhenti.
```

Output 4.12 Logout & Keluar Program

5. GIT

1. Git Init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah folder. Cukup ketik “git init” pada terminal VSCode.

```
PS C:\Users\LENOVO\Downloads\Praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/LENOVO/Downloads/Praktikum-apl/.git/
```

Gambar 5.1 Git Init

2. Git Add adalah perintah Git untuk menandai semua perubahan (file baru, diubah, atau dihapus) di folder saat ini agar siap disimpan ke dalam versi berikutnya di Git.

```
PS C:\Users\LENOVO\Downloads\Praktikum-apl> git add .
```

Gambar 5.2 Git Add

3. Git commit adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository. Setiap commit akan memiliki hash unik, pesan commit, dan menyimpan snapshot dari perubahan yang dilakukan.

```
PS C:\Users\LENOVO\Downloads\Praktikum-apl> git commit -m "Finish PostTest 6"
[main 81490f0] Finish PostTest 6
 4 files changed, 349 insertions(+)
 create mode 100644 post-test/post-test6/2409106002-RenayaPutriAlika-PT-6.cpp
 create mode 100644 post-test/post-test6/2409106002-RenayaPutriAlika-PT-6.exe
 create mode 100644 post-test/post-test6/2409106002-RenayaPutriAlika-PT-6.pdf
PS C:\Users\LENOVO\Downloads\Praktikum-apl>
```

Gambar 5.3 Git Commit

4. Git remote adalah perintah Git yang digunakan untuk menghubungkan repository lokal dengan repository remote (misalnya di GitHub, GitLab, atau Bitbucket). Dengan perintah ini, kita bisa mengelola koneksi ke repository jarak jauh, memungkinkan push, pull, dan fetch dari repository tersebut.

```
PS C:\Users\LENOVO\Downloads\Praktikum-apl> git remote set-url origin https://github.com/Renaya0207/praktikum-apl.git
```

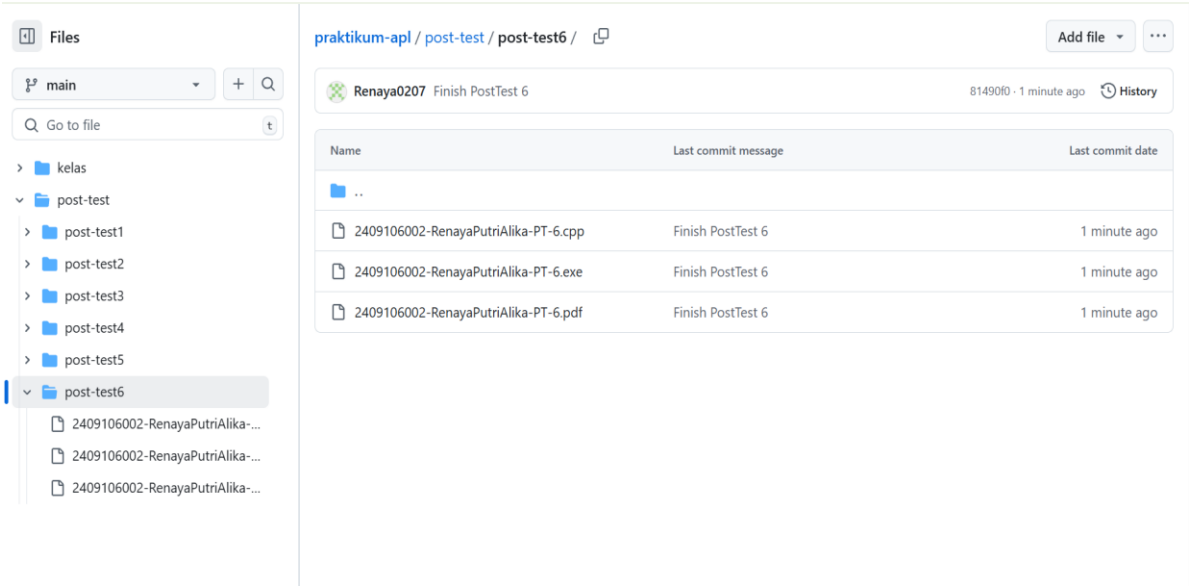
Gambar 5.4 Git Remote

5. Git Push adalah perintah Git yang digunakan untuk mengirim (mengunggah) perubahan dari repository lokal ke repository remote (seperti GitHub, GitLab, atau Bitbucket). Perintah ini memastikan bahwa perubahan yang sudah dikomit di lokal tersedia di repository jarak jauh sehingga bisa diakses oleh orang lain atau untuk cadangan.

```
PS C:\Users\LENOVO\Downloads\Praktikum-apl> git push origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 679.90 KiB | 3.43 MiB/s, done.
Total 9 (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/Renaya0207/praktikum-apl.git
   292c5eb..6b8b555  main -> main
PS C:\Users\LENOVO\Downloads\Praktikum-apl>
```

Gambar 5.5 Git Push

6. Berhasil di Up



Gambar 5.6 Berhasil