

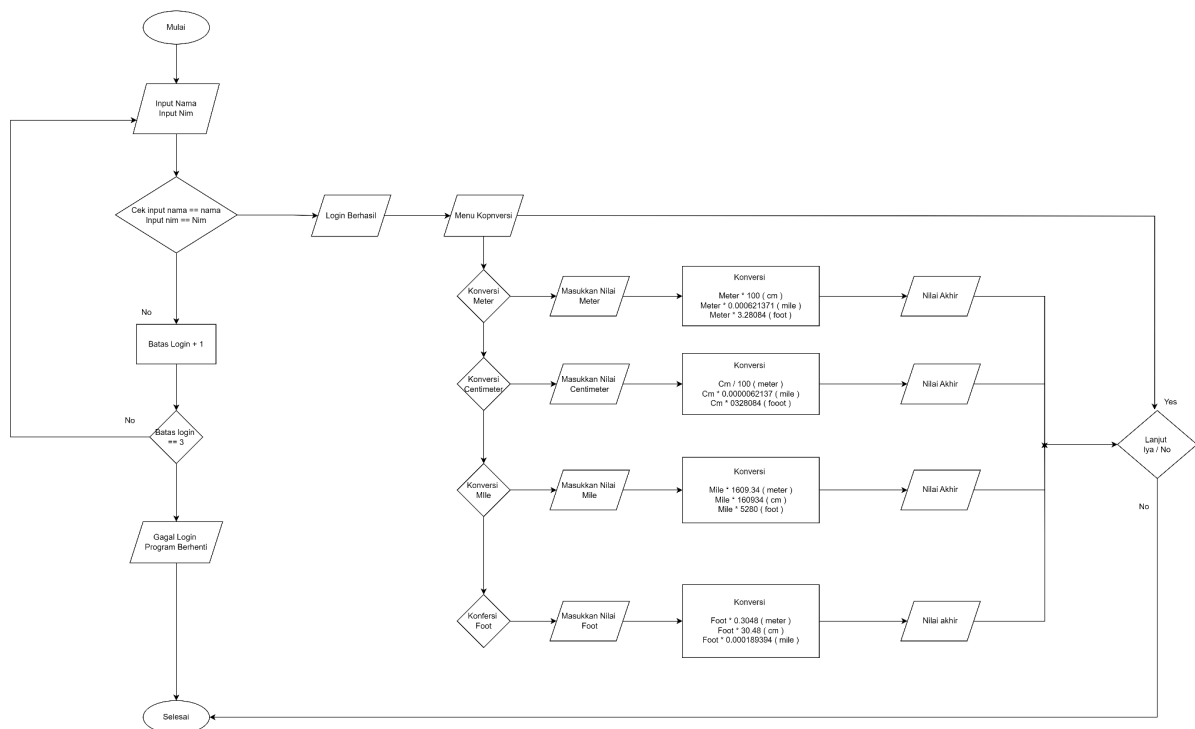
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Renaya Putri Alike (2409106002)
Kelas A1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart

Alur dan Penjelasan :

1) Mulai

Program dimulai dan menampilkan proses login.

2) Input Nama dan NIM

Pengguna diminta untuk memasukkan nama dan NIM

3) Cek Kesesuaian Nama dan NIM

Program memeriksa apakah input nama dan NIM sesuai dengan data yang telah ditentukan.

- **Jika benar** → Login berhasil dan pengguna diarahkan ke menu konversi.
- **Jika salah** → Jumlah percobaan login bertambah 1.

4) Cek Batas Percobaan Login

Jika jumlah percobaan mencapai 3 kali, program menampilkan pesan "Gagal Login, Program Berhenti" dan berakhir. Jika belum mencapai batas, pengguna dapat mencoba login kembali.

5) **Menu Konversi**

Jika login berhasil, pengguna diberikan beberapa pilihan konversi:

- Konversi **Meter**
- Konversi **Centimeter**
- Konversi **Mile**
- Konversi **Foot**

6) **Input Nilai**

Setelah memilih satuan yang akan dikonversi, pengguna diminta memasukkan nilai yang ingin dikonversi.

7) **Proses Konversi**

Program menghitung hasil konversi ke satuan lain :

- **Meter** → cm, mile, foot
- **Centimeter** → meter, mile, foot
- **Mile** → meter, cm, foot
- **Foot** → meter, cm, mile

8) **Tampilkan Hasil Konversi**

Program menampilkan hasil konversi berdasarkan nilai yang dimasukkan pengguna.

9) **Pilihan Lanjut atau Tidak**

Setelah melihat hasil konversi, pengguna diberikan pilihan:

- **Jika memilih "Yes"** → Kembali ke menu konversi untuk memilih satuan lain.
- **Jika memilih "No"** → Program berhenti.

10) **Selesai**

Jika pengguna memilih keluar atau gagal login tiga kali, program selesai.

2. Analisis Program

2.1 Deskripsi Singkat Program

Program yang saya buat berfungsi sebagai sistem login dan konversi satuan panjang. Awalnya, pengguna harus memasukkan **Nama** dan **NIM** yang sesuai dengan data yang telah ditentukan. Jika pengguna gagal login tiga kali, program akan berhenti. Setelah berhasil login, pengguna diberikan pilihan untuk mengkonversi satuan panjang antara meter, centimeter, mile, dan foot. Program memastikan bahwa input yang dimasukkan valid dan

memberikan hasil konversi yang sesuai. Selain itu, terdapat opsi untuk keluar dari program jika pengguna tidak ingin melanjutkan.

2.2 Penjelasan Alur & Algoritma

2.2.1 Alur Program

Program ini memiliki dua bagian utama: **Sistem Login** dan **Konversi Satuan Panjang**. Berikut adalah alur program dari awal hingga akhir :

1. **Program Dimulai** : mendeklarasikan variabel yang diperlukan, termasuk Nama dan NIM pengguna.
2. **Sistem Login** : Pengguna diminta memasukkan Nama dan NIM. Jika Nama dan NIM cocok dengan yang telah ditentukan, login berhasil dan program lanjut ke menu konversi. Jika salah, pengguna diberi kesempatan hingga tiga kali. Jika tetap gagal, program berhenti.
3. **Menu Konversi** : Menu konversi dengan 5 pilihan dari Meter, Centimeter, Mile, Foot, atau keluar dari program.
4. **Pengguna Memilih Salah Satu Opsi** : Pengguna memilih kemudian memasukkan nilai yang ingin dikonversi.
5. **Program Mengkonversi Nilai** : Program mengkonversi nilai sesuai satuan yang dipilih dan menampilkan hasilnya.
6. **Menu Kembali** : Menu ini Untuk memilih konversi lain atau keluar. Jika pengguna memilih keluar, Maka program berhenti.

2.2.2 Algoritma

1) Inisialisasi Variabel

- Deklarasikan variabel **nama** dan **nim** untuk input pengguna.
- Tetapkan konstanta **Nama** dan **NIM** sebagai data login yang valid.
- Gunakan variabel **percobaan** untuk menghitung jumlah percobaan login.

2) Sistem Login

- Selama jumlah percobaan login kurang dari 3:
 - Minta pengguna memasukkan **nama** dan **nim**.
 - Jika cocok dengan data yang valid, tampilkan pesan "Login berhasil!" dan lanjut ke menu.
 - Jika tidak cocok, tampilkan pesan "Login gagal. Coba lagi." dan tambah nilai **percobaan**.
- Jika pengguna gagal login setelah 3 kali percobaan, tampilkan pesan "Anda telah gagal login 3 kali. Program berhenti." lalu akhiri program.

3) Menampilkan Menu Konversi

- Tampilkan daftar pilihan konversi satuan panjang:
 - Konversi dari meter
 - Konversi dari centimeter
 - Konversi dari mile
 - Konversi dari foot
 - Keluar dari program
- Minta pengguna memilih menu.

4) Validasi Input Menu

- Jika input tidak valid (bukan angka atau di luar rentang 1-5), tampilkan pesan error dan minta input ulang.

5) Proses Konversi Berdasarkan Pilihan Pengguna

- **Jika memilih meter** : Minta input dalam meter, validasi input, lalu hitung konversi ke cm, mile, dan foot.
- **Jika memilih centimeter** : Minta input dalam cm, validasi input, lalu hitung konversi ke meter, mile, dan foot.
- **Jika memilih mile** : Minta input dalam mile, validasi input, lalu hitung konversi ke meter, cm, dan foot.
- **Jika memilih foot** : Minta input dalam foot, validasi input, lalu hitung konversi ke meter, cm, dan mile.

6) Keluar dari Program

- Jika pengguna memilih opsi 5, tampilkan pesan "Terima kasih telah menggunakan program ini." dan akhiri program.
- Jika tidak, kembali ke menu utama dan ulangi proses.

2.3 Penjelasan Detail Blok Kode

A. Header dan Deklarasi Variabel

- `#include <iostream>` Digunakan untuk operasi input dan output.
- `#include <limits>` Digunakan untuk menangani error input.
- `using namespace std;` Memungkinkan penggunaan dari `std` tanpa harus menuliskan `std::` sebelum fungsi seperti `cout` atau `cin`.
- `nama` dan `nim` Digunakan untuk menyimpan input pengguna.
- `percobaan` Digunakan untuk menghitung jumlah percobaan login.
- `Nama` dan `NIM` Adalah konstanta yang digunakan sebagai data login yang valid.

B. Sistem Login

- `while (percobaan < 3)` Memungkinkan pengguna mencoba login maksimal tiga kali.
- `getline(cin, nama)` Digunakan agar input nama bisa mengandung spasi.
- Jika input sesuai dengan konstanta `Nama` dan `NIM`, login berhasil dan program lanjut ke menu konversi.
- Jika tidak sesuai, jumlah percobaan bertambah dan pengguna diminta mencoba lagi.
- Jika pengguna gagal login tiga kali, program menampilkan pesan kesalahan dan berhenti (`return 0;`).

C. Menu Konversi

- `do-while` Digunakan agar menu selalu muncul setelah login berhasil.
- Program menampilkan lima opsi menu.
- `cin >> pilihan;` Digunakan untuk menerima input pilihan dari pengguna.
- Memastikan input adalah angka dan berada dalam rentang 1–5.
- `cin.clear()` Menghapus error flag jika input tidak valid.
- `cin.ignore(...)` Mengabaikan karakter sisa agar input berikutnya dapat diterima dengan benar.

D. Pemrosesan Konversi

Setiap kasus dalam `switch` menangani satu satuan panjang tertentu.

Konversi Meter :

- `double meter`; Digunakan untuk menyimpan input dalam meter.
- Pengguna memasukkan nilai meter.
- Memastikan input adalah angka yang valid.
- Mengonversi meter ke cm, mile, dan foot, lalu mencetak hasilnya.

Konversi Centimeter :

- `cm / 100` untuk meter, `cm * 0.0000062137` untuk mile, dan `cm * 0.0328084` untuk foot.

Konversi Mile :

- `mile * 1609.34` untuk meter, `mile * 160934` untuk cm, dan `mile * 5280` untuk foot.

Konversi Foot :

- `foot * 0.3048` untuk meter, `foot * 30.48` untuk cm, dan `foot * 0.000189394` untuk mile.

E. Keluar dari Program

- Jika pengguna memilih `5`, program menampilkan pesan perpisahan dan keluar dari `do-while`.
- `do-while` memastikan program terus berjalan sampai pengguna memilih keluar.

3. Source Code

3.1 Sistem Login

Program meminta pengguna untuk memasukkan **Nama** dan **NIM** yang telah ditentukan. Pengguna diberi **maksimal 3 kali percobaan** untuk memasukkan data yang benar. Jika telah gagal 3 kali percobaan, maka program berhenti.

```

while (percobaan < 3) {
    cout << "Masukkan Nama: ";
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    getline(cin, nim);

    if (nama == Nama && nim == NIM) {
        cout << "Login berhasil!\n";
        break;
    } else {
        cout << "Login gagal. Coba lagi.\n";
        percobaan++;
    }
}

if (percobaan == 3) {
    cout << "Anda telah gagal login 3 kali. Program berhenti.\n";
    return 0;
}

```

Gambar 3.1 Sistem Login

3.2 Menu Konversi Satuan Panjang

Setelah login berhasil, program menampilkan menu pilihan konversi satuan panjang. Pengguna harus memilih menu dengan angka 1 hingga 5. Jika input tidak valid, program akan meminta input ulang.

```

int pilihan;
do {
    cout << "\nMenu Konversi Satuan Panjang\n";
    cout << "1. Konversi Meter\n";
    cout << "2. Konversi Centimeter\n";
    cout << "3. Konversi Mile\n";
    cout << "4. Konversi Foot\n";
    cout << "5. Keluar\n";
    cout << "Pilih menu: ";
}

```



```

cin >> pilihan;

// Validasi input menu
while (cin.fail() || pilihan < 1 || pilihan > 5) {
    cin.clear(); // Menghapus error
    cin.ignore(numeric_limits<streamsize>::max(), '\n');
    cout << "Pilihan tidak valid. Masukkan angka antara 1 dan 5: ";
    cin >> pilihan;
}

```

Gambar 3.2 Menu Konversi

3.3 Proses Konversi

Setiap pilihan menu akan meminta input angka dari pengguna, memvalidasi input, dan melakukan konversi seperti meter ke satuan lain dan begitu juga seterusnya.

```

switch (pilihan) {
    case 1: {
        double meter;
        cout << "Masukkan nilai dalam meter: ";
        cin >> meter;
        // Validasi input angka
        while (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Input tidak valid. Masukkan angka untuk meter: ";
            cin >> meter;
        }
        cout << meter << " meter = " << meter * 100 << " cm\n";
        cout << meter << " meter = " << meter * 0.000621371 << "
mile\n";

        cout << meter << " meter = " << meter * 3.28084 << " foot\n";
        break;
    }
    case 2: {
        double cm;
        cout << "Masukkan nilai dalam centimeter: ";

```

```

        cin >> cm;
        while (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Input tidak valid. Masukkan angka untuk centimeter: ";

            cin >> cm;
        }
        cout << cm << " cm = " << cm / 100 << " meter\n";
        cout << cm << " cm = " << cm * 0.0000062137 << " mile\n";
        cout << cm << " cm = " << cm * 0.0328084 << " foot\n";
        break;
    }
    case 3: {
        double mile;
        cout << "Masukkan nilai dalam mile: ";
        cin >> mile;
        while (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Input tidak valid. Masukkan angka untuk mile: ";
            cin >> mile;
        }
        cout << mile << " mile = " << mile * 1609.34 << " meter\n";
        cout << mile << " mile = " << mile * 160934 << " cm\n";
        cout << mile << " mile = " << mile * 5280 << " foot\n";
        break;
    }
    case 4: {
        double foot;
        cout << "Masukkan nilai dalam foot: ";
        cin >> foot;
        while (cin.fail()) {
            cin.clear();
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Input tidak valid. Masukkan angka untuk foot: ";
            cin >> foot;
        }
        cout << foot << " foot = " << foot * 0.3048 << " meter\n";
    }
}

```

```

        cout << foot << " foot = " << foot * 30.48 << " cm\n";
        cout << foot << " foot = " << foot * 0.000189394 << " mile\n";
        break;
    }

```

Gambar 3.3 Proses Konversi

3.4 Validasi Input

Program menggunakan `cin.fail()` untuk mendeteksi input yang tidak valid.

Jika input bukan angka, program akan meminta input ulang hingga pengguna memasukkan angka yang benar.

```

cin >> foot;
    while (cin.fail()) {
        cin.clear();
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Input tidak valid. Masukkan angka untuk foot: ";
        cin >> foot;
    }

```

Gambar 3.4 Validasi Input

3.5 Keluar dari Program

Jika pengguna memilih **opsi 5**, program akan menampilkan pesan terima kasih dan keluar dari program.

```

} while (pilihan != 5);

    return 0;
}

```

Gambar 3.5 Keluar Program

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1 : Login Berhasil dan Menggunakan Fitur Konversi

- Jalankan program.
- Masukkan **Nama**: Renaya Putri Alika.
- Masukkan **NIM**: 2409106002.
- Pilih menu **Konversi Satuan Panjang**.
- Pilih opsi **Konversi Meter** dan masukkan angka, misal : 10
- Program menampilkan hasil konversi dari angka 10:
 - 10 meter = 1000 cm
 - 10 meter = 0.00621371 mile
 - 10 meter = 32.8084 foot
- Keluar dari program.

Skenario 2 : Gagal Login 3 Kali dan Program Berhenti

- Jalankan program.
- Masukkan **Nama**: User Salah.
- Masukkan **NIM**: 1234567890.
- Program menampilkan "**Login gagal. Coba lagi.**"
- Ulangi langkah 2-3 sebanyak 3 kali dengan nama/NIM yang salah.
- Program menampilkan "**Anda telah gagal login 3 kali. Program berhenti.**"
- Program otomatis berhenti.

4.2 Hasil Output

```
Masukkan Nama: Renaya Putri Alika
Masukkan NIM: 2409106002
Login berhasil!

Menu Konversi Satuan Panjang
1. Konversi Meter
2. Konversi Centimeter
3. Konversi Mile
4. Konversi Foot
5. Keluar
Pilih menu: 1
Masukkan nilai dalam meter: 10
10 meter = 1000 cm
10 meter = 0.00621371 mile
10 meter = 32.8084 foot

Menu Konversi Satuan Panjang
1. Konversi Meter
2. Konversi Centimeter
3. Konversi Mile
4. Konversi Foot
5. Keluar
Pilih menu: 2
Masukkan nilai dalam centimeter: 20
20 cm = 0.2 meter
20 cm = 0.000124274 mile
20 cm = 0.656168 foot
```

Gambar 4.1 Output skenario 1

```
Menu Konversi Satuan Panjang
1. Konversi Meter
2. Konversi Centimeter
3. Konversi Mile
4. Konversi Foot
5. Keluar
Pilih menu: 3
Masukkan nilai dalam mile: 30
30 mile = 48280.2 meter
30 mile = 4.82802e+06 cm
30 mile = 158400 foot

Menu Konversi Satuan Panjang
1. Konversi Meter
2. Konversi Centimeter
3. Konversi Mile
4. Konversi Foot
5. Keluar
Pilih menu: 4
Masukkan nilai dalam foot: 40
40 foot = 12.192 meter
40 foot = 1219.2 cm
40 foot = 0.00757576 mile

Menu Konversi Satuan Panjang
1. Konversi Meter
2. Konversi Centimeter
3. Konversi Mile
4. Konversi Foot
5. Keluar
Pilih menu: 5
Terima kasih telah menggunakan program ini.
```

Gambar 4.2 Output Skenario 1

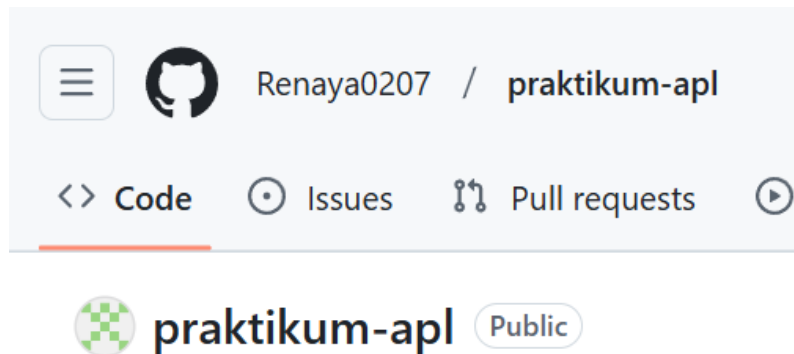
- Jika Nama dan NIM pengguna salah:

```
Masukkan Nama: arham miraj
Masukkan NIM: 18
Login gagal. Coba lagi.
Masukkan Nama: arham
Masukkan NIM: 02
Login gagal. Coba lagi.
Masukkan Nama: cintaku
Masukkan NIM: 01
Login gagal. Coba lagi.
Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 4.3 Output Skenario 2

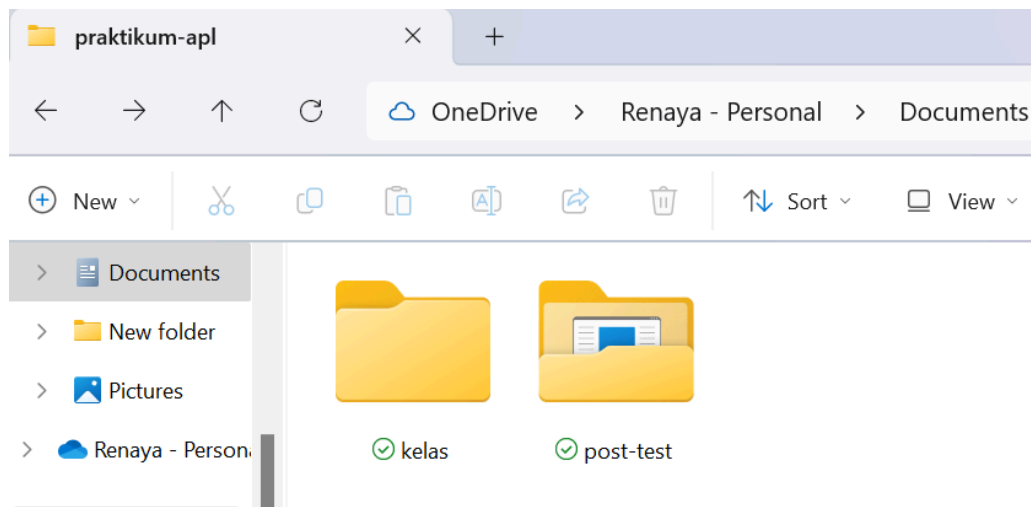
5. GIT

5.1 Membuat Repository Publik di GitHub



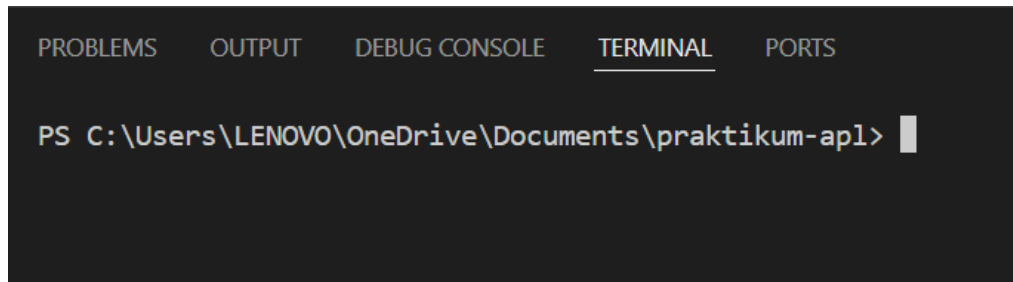
Gambar 5.1 Membuat Repository GitHub

5.2 Membuat Folder Praktikum di File Explorer



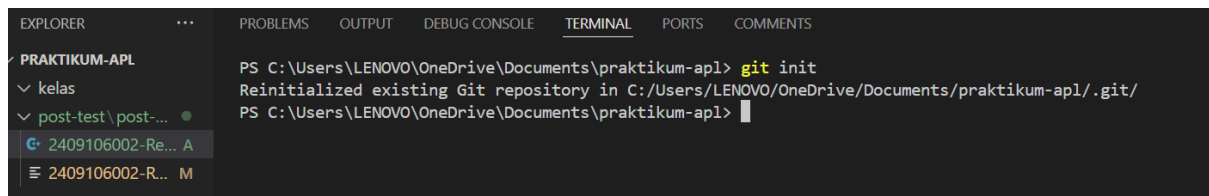
Gambar 5.2 Membuat Folder Praktikum

5.3 Penyesuaian Path di VSCode



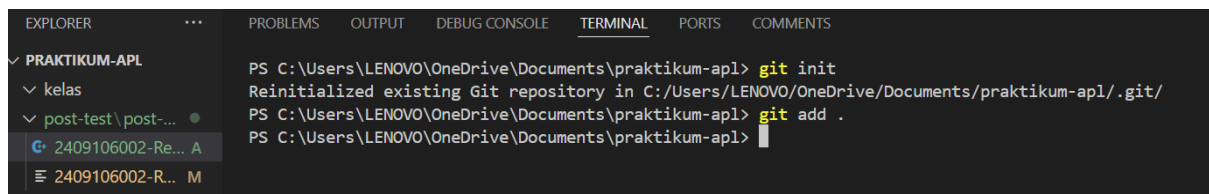
Gambar 5.3 Penyesuaian Path

5.4 Git Init



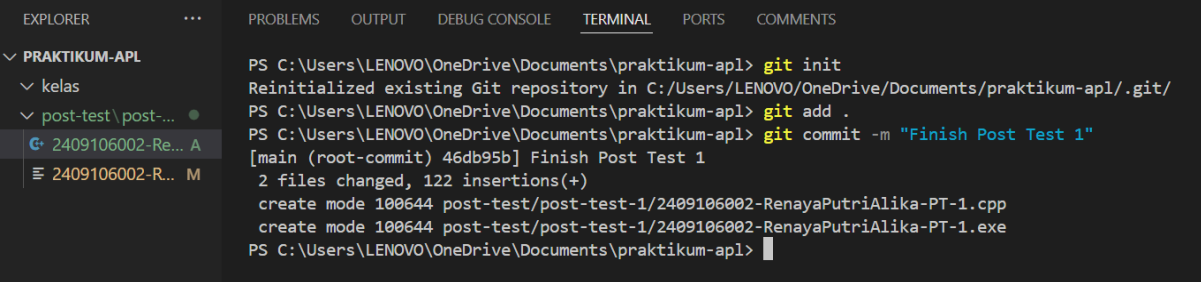
Gambar 5.4 Git Init

5.5 Git Add



Gambar 5.5 Git Add

5.6 Git Commit (Check Point)

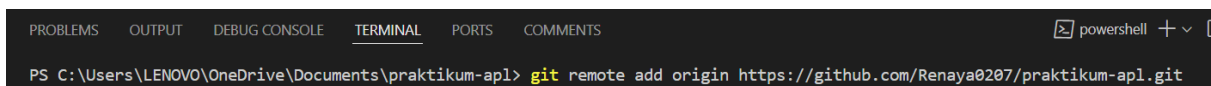


The screenshot shows the VS Code interface with the Explorer pane on the left displaying a file tree for 'PRAKTIKUM-APL' containing 'kelas' and 'post-test\post-...'. The Terminal pane on the right shows the following commands and output:

```
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/LENOVO/OneDrive/Documents/praktikum-apl/.git/
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl> git add .
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl> git commit -m "Finish Post Test 1"
[main (root-commit) 46db95b] Finish Post Test 1
2 files changed, 122 insertions(+)
create mode 100644 post-test/post-test-1/2409106002-RenayaPutriAlika-PT-1.cpp
create mode 100644 post-test/post-test-1/2409106002-RenayaPutriAlika-PT-1.exe
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl>
```

Gambar 5.6 Git Commit

5.7 Git Remote (Menghubungkan Repository Lokal dengan GitHub)

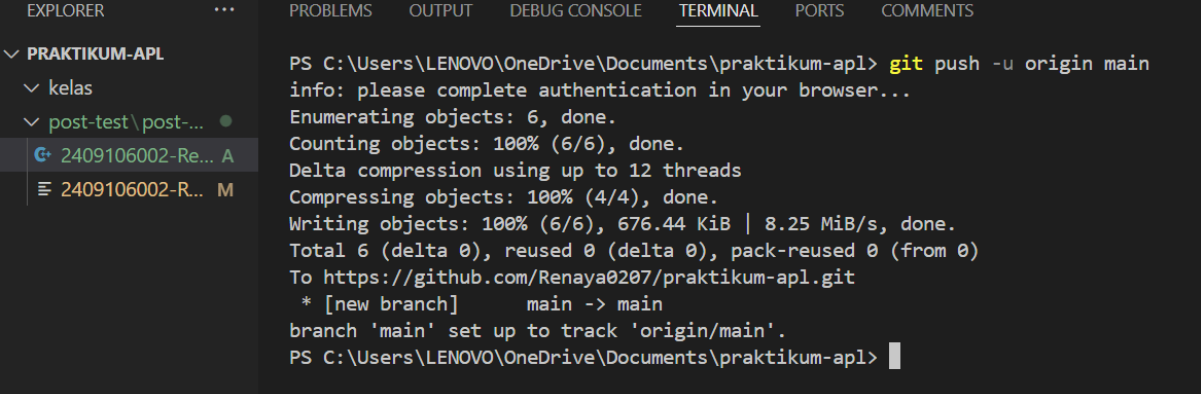


The screenshot shows the VS Code interface with the Terminal pane displaying the following command and output:

```
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl> git remote add origin https://github.com/Renaya0207/praktikum-apl.git
```

Gambar 5.7 Git Remote

5.8 Git Push (Update Semua yang Ada di Repository Lokal)

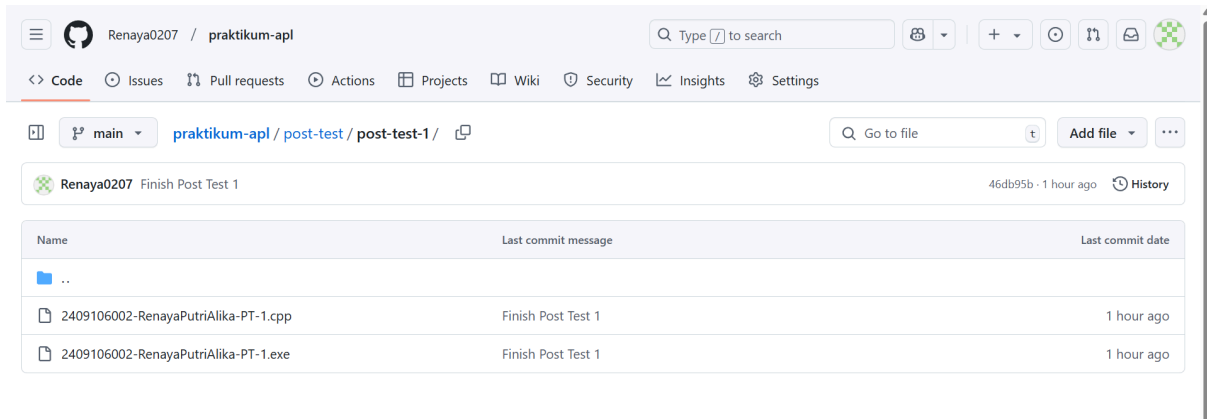


The screenshot shows the VS Code interface with the Explorer pane on the left displaying a file tree for 'PRAKTIKUM-APL' containing 'kelas' and 'post-test\post-...'. The Terminal pane on the right shows the following commands and output:

```
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 676.44 KiB | 8.25 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Renaya0207/praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\LENOVO\OneDrive\Documents\praktikum-apl>
```

Gambar 5.8 Git Push

5.9 Reload GitHub.



Gambar 5.9 Reload GitHub