

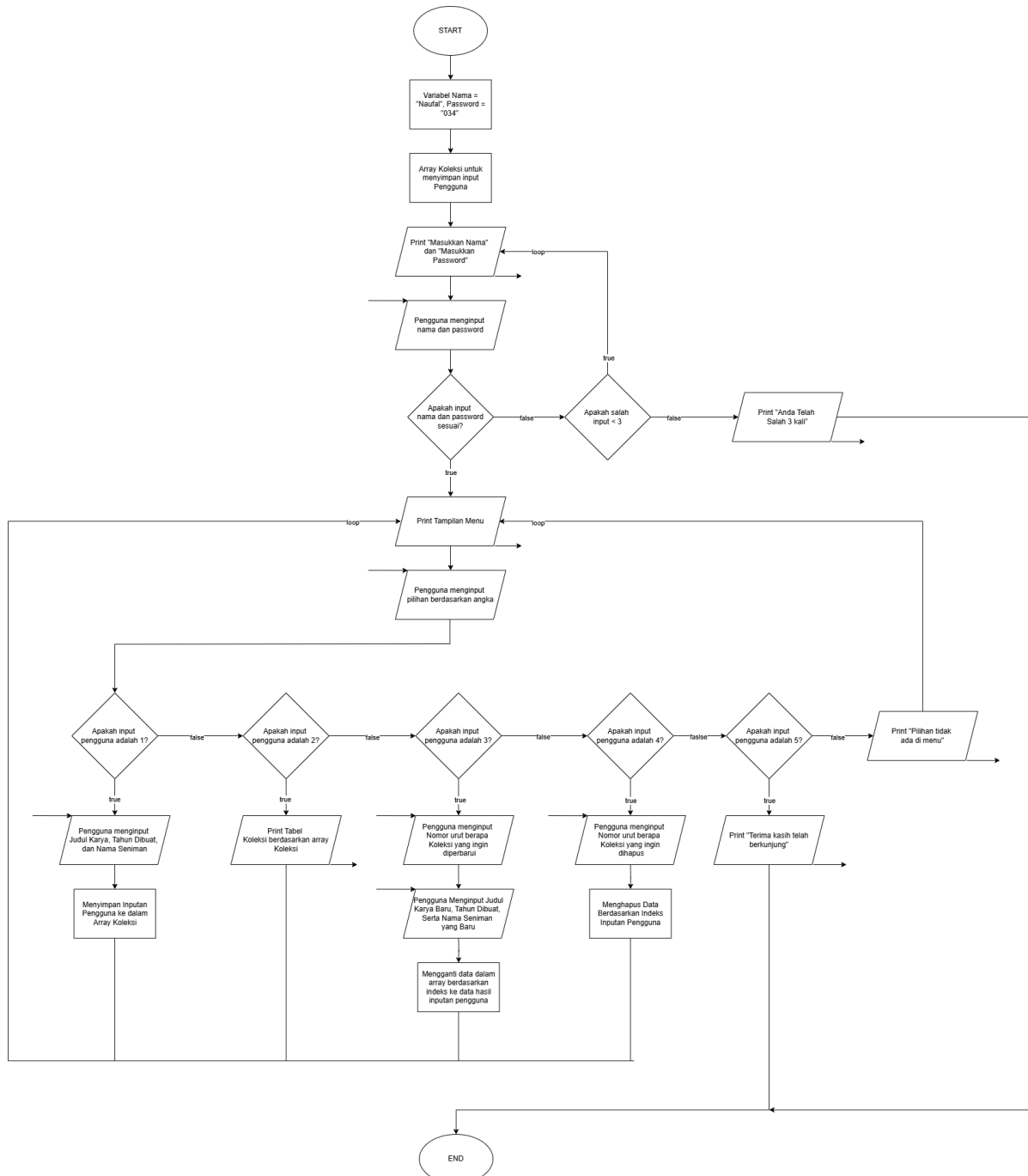
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Muhammad Naufal Rifyan Ilham (2409106034)**  
**Kelas (A2 '24)**

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

# 1. Flowchart



Gambar 1 Flowchart

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini bertujuan untuk Memanajemen berbagai data yang berkaitan dengan Koleksi/Karya Seni yang ada di Museum Louvre, dimana pengguna dapat Menambahkan, Melihat, Mengupdate, Serta Menghapus berbagai data Koleksi yang ada di Museum Louvre

### **2.2 Penjelasan Alur & Algoritma**

1. Mulai
2. Program akan meminta pengguna untuk menginput nama dan password yang sesuai ( Nama : Naufal dan Password : 034) untuk melakukan login
3. Jika nama dan password yang di input sesuai, maka program akan dilanjutkan. Namun, jika input pengguna salah maka akan mengulang (loop) atau meminta ulang pengguna untuk menginput nama dan password lagi. Program akan terus mengulang hingga pengguna salah maksimal 3 kali, jika sudah mencapai batas maka program akan dihentikan dan menampilkan teks “Anda telah Salah 3 kali”
4. Setelah login selesai, program akan menampilkan menu berupa pilihan berdasarkan angka, dimana pilihan tersebut akan memberikan detail berikut :
  1. Menambah Koleksi di Museum Louvre
  2. Lihat Koleksi di Museum Louvre
  3. Memperbarui Koleksi di Museum Louvre
  4. Menghapus Koleksi di Museum Louvre
  5. Keluar dari program
5. Setelah pengguna menginput pilihan berdasarkan menu, maka program akan memproses berdasarkan pilihan pengguna.
  - Jika pengguna menginput angka 1, maka program akan meminta input berupa Judul Karya, Tahun Dibuat, dan Nama Seniman yang akan disimpan ke dalam Array Koleksi
  - Jika pengguna menginput angka 2. maka program akan menampilkan seluruh data yang ada didalam Array Koleksi dalam bentuk tabel ke terminal
  - Jika pengguna menginput angka 3, maka program akan meminta indeks ke berapa yang ingin diubah, setelah itu program akan meminta kembali Judul Karya Baru, Tahun Dibuat Baru, serta Nama Seniman yang baru. Sehingga data sebelumnya akan diubah menjadi data inputan pengguna yang baru
  - Jika pengguna menginput angka 4, maka program akan meminta indeks berapa yang ingin dihapus, lalu program akan menghapus semua elemen berdasarkan inputan pengguna
  - Jika pengguna menginput angka 5, maka program akan berhenti
6. Selesai

### 3. Source Code

#### A. Fitur Login

Fitur ini digunakan untuk melakukan login agar pengguna dapat menggunakan program. Fitur ini mengharuskan pengguna untuk menginput nama dan password yang sesuai

##### Source Code:

```
do {  
    cout << "Masukkan Nama anda : ";  
    cin >> input_nama;  
    cout << "Masukkan Password anda : ";  
    cin >> input_password;  
  
    if(input_nama == nama && input_password == password){  
        program = true;  
    }  
    else {  
        program = false;  
        salah += 1;  
    }  
  
} while (program == false && salah < 3);
```

Gambar 3A Fitur Login

#### B. Fitur Menu

Fitur ini akan menampilkan menu pada terminal

##### Source Code:

```
do {  
    cout << R"(  
=====   
    Selamat datang di Museum Louvre  
=====   
1. Menambah Koleksi di Museum Louvre  
2. Lihat Koleksi di Museum Louvre  
3. Memperbarui Koleksi di Museum Louvre
```

```

4. Menghapus Koleksi di Museum Louvre
5. Keluar dari program
=====
)" << endl;

    cout << "\nMasukkan Pilihan : ";
    cin >> pilihan;
    system("cls");

```

Gambar 3B Fitur Menu

### C. Fitur Menambah data

Fitur ini berfungsi untuk menambahkan data kedalam array Koleksi.

#### Source Code:

```

case 1:
    if (jumlah_elemen < MAX_JUMLAH) {

        cout << "Masukkan Judul Karya: ";
        cin.ignore();
        getline(cin, koleksi[jumlah_elemen][0]);

        cout << "Masukkan Tahun Dibuat : ";
        cin >> koleksi[jumlah_elemen][1];

        cout << "Masukkan Nama Seniman: ";
        cin.ignore();
        getline(cin, koleksi[jumlah_elemen][2]);

        jumlah_elemen++;
        cout << "Karya berhasil ditambahkan" << endl;
    }

    else {
        cout << "Koleksi Sudah Penuh." << endl;
    }
    break;

```

Gambar 3C Fitur Menambah data

### D. Fitur Menampilkan Data Berbentuk Tabel

Fitur ini berfungsi untuk menampilkan data didalam array Koleksi yang berbentuk tabel

### Source Code:

```
case 2:
    if (jumlah_elemen < MAX_JUMLAH) {
        cout <<
        "===== " <<
        endl;
        cout << " | No | " << left << setw(26) << "Judul Karya" << " | "
        << setw(8) << "Tahun" << " | " << setw(18) << "Seniman" << " | " << endl;
        cout <<
        "===== " <<
        endl;

        for (int i = 0; i < jumlah_elemen; i++){
            cout << " | " << setw(2) << i+1 << " | " << left << setw(26)
            << koleksi[i][0] << " | " << setw(8) << koleksi[i][1] << " | " << setw(18)
            << koleksi[i][2] << " | " << endl;
        }

        cout <<
        "===== " <<
        endl;
    }

    else {
        cout << "Tidak Ada Koleksi di Museum Louvre" << endl;
    }
    break;
```

Gambar 3D Fitur Menampilkan

### E. Fitur Mengupdate/Mengubah data

Fitur ini berfungsi untuk mengubah data yang ada didalam array

### Source Code:

```
case 3:
    for (int i = 0; i < jumlah_elemen; i++) {
        cout << "Daftar Koleksi\n" << i+1 << ". " << koleksi[i][0] <<
        endl;
    }

    cout << "Masukkan nomor koleksi yang ingin diubah/perbarui : ";
    cin >> index;
```

```

    if (index < 1 || index > jumlah_elemen) {
        cout << "Nomor tidak Valid" << endl;
    }

    else {
        index--;

        cout << "Masukkan Judul Karya Baru: ";
        cin.ignore();
        getline(cin, koleksi[index][0]);

        cout << "Masukkan Tahun Dibuat yang Baru : ";
        cin >> koleksi[index][1];

        cout << "Masukkan Nama Seniman Baru: ";
        cin.ignore();
        getline(cin, koleksi[index][2]);

        cout << "Koleksi berhasil diperbarui" << endl;

    }
    break;

```

Gambar 3E Fitur Mengubah data

## F. Fitur Menghapus data

Fitur ini berfungsi untuk menghapus data yang ada didalam array berdasarkan inputan pengguna

### Source Code:

```

case 4:
    for (int i = 0; i < jumlah_elemen; i++) {
        cout << "Daftar Koleksi\n" << i+1 << ". " << koleksi[i][0] <<
endl;
    }

    cout << "Masukkan nomor koleksi yang ingin dihapus : ";
    cin >> index;

    if (index < 1 || index > jumlah_elemen) {
        cout << "Nomor tidak Valid" << endl;
    }

    else {

```

```

        index--;

        for (int i = index; i < jumlah_element - 1; i++) {
            koleksi[i][0] = koleksi[i+1][0];
            koleksi[i][1] = koleksi[i+1][1];
            koleksi[i][2] = koleksi[i+1][2];
        }
        jumlah_element--;
        cout << "Koleksi Berhasil di Hapus" << endl;
    }

    break;

case 5:
    cout << "\nTerima Kasih telah berkunjung" << endl;
    break;

default:
    cout << "\nPilihan tidak ada di menu" << endl;
    break;

```

Gambar 3F Fitur Menghapus data



## **4. Uji Coba dan Hasil Output**

### **4.1 Uji Coba**

Beberapa skenario input yang bisa diterapkan

1. Skenario 1

Ketika login dan program meminta nama dan password, jika pengguna menginput Nama = Naufal dan Password = 034 maka program akan berlanjut.

2. Skenario 2

Ketika login dan program meminta nama dan password, jika pengguna menginput nama dan password yang salah hingga 3 kali maka program akan dihentikan.

3. Skenario 3

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka 1, maka program akan meminta input berupa Judul Karya, Tahun Dibuat, dan Nama Seniman.

4. Skenario 4

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka 2, maka program akan menampilkan seluruh data dalam bentuk tabel ke terminal

5. Skenario 5

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka 3, maka program akan meminta indeks ke berapa yang ingin diubah, setelah itu program akan meminta kembali Judul Karya Baru, Tahun Dibuat Baru

6. Skenario 6

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka 4, maka program akan meminta indeks berapa yang ingin dihapus

#### 7. Skenario 7

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka 5, maka program akan dihentikan.

#### 8. Skenario 8

Ketika program meminta inputan pilihan berdasarkan menu. Jika pengguna menginput angka selain 1 hingga 5, maka program akan menampilkan peringatan dan meminta ulang inputan pengguna.

#### 9. Skenario 9

Saat Pengguna Menginput angka 3 atau 4, lalu program meminta indeks berapa yang ingin diubah/hapus. Jika pengguna menginput angka yang tidak ada di daftar koleksi, maka program memberi peringatan dan akan kembali ke menu sebelumnya

### 4.2 Hasil Output

```
Masukkan Nama anda : Naufal
Masukkan Password anda : 034

=====
                Selamat datang di Museum Louvre
=====
1. Menambah Koleksi di Museum Louvre
2. Lihat Koleksi di Museum Louvre
3. Memperbarui Koleksi di Museum Louvre
4. Menghapus Koleksi di Museum Louvre
5. Keluar dari program
=====
```

Gambar 4.1 Skenario 1

```
Masukkan Nama anda : adsa
Masukkan Password anda : asd
Masukkan Nama anda : ad
Masukkan Password anda : fadf
Masukkan Nama anda : fafe
Masukkan Password anda : adf
Anda telah salah 3 kali
PS C:\praktikum-apl\post-test\post-test-2> █
```

Gambar 4.1 Skenario 2

```
Masukkan Judul Karya: Mona Lisa
Masukkan Tahun Dibuat : 1503
Masukkan Nama Seniman: Leonardo da Vinci
Karya berhasil ditambahkan
```

Gambar 4.1 Skenario 3

```
=====
| No | Judul Karya          | Tahun | Seniman          |
=====
| 1  | Mona Lisa           | 1503  | Leonardo da Vinci |
=====
```

Gambar 4.1 Skenario 4

```
Daftar Koleksi
1. Mona Lisa
Masukkan nomor koleksi yang ingin diubah/perbarui : 1
Masukkan Judul Karya Baru: The Starry Night
Masukkan Tahun Dibuat yang Baru : 1889
Masukkan Nama Seniman Baru: Vincent Van Gogh
Koleksi berhasil diperbarui
```

Gambar 4.1 Skenario 5

```
=====
| No | Judul Karya          | Tahun | Seniman          |
=====
| 1  | The Starry Night     | 1889  | Vincent Van Gogh |
=====
```

Gambar 4.1 Skenario 5 (Setelah diubah)

```
Daftar Koleksi
1. The Starry Night
Masukkan nomor koleksi yang ingin dihapus : 1
Koleksi Berhasil di Hapus
```

Gambar 4.1 Skenario 6

```
Terima Kasih telah berkunjung
PS C:\praktikum-apl\post-test\post-test-2> █
```

Gambar 4.1 Skenario 7

```
Pilihan tidak ada di menu

=====
                Selamat datang di Museum Louvre
=====

1. Menambah Koleksi di Museum Louvre
2. Lihat Koleksi di Museum Louvre
3. Memperbarui Koleksi di Museum Louvre
4. Menghapus Koleksi di Museum Louvre
5. Keluar dari program
=====

Masukkan Pilihan : █
```

Gambar 4.1 Skenario 8

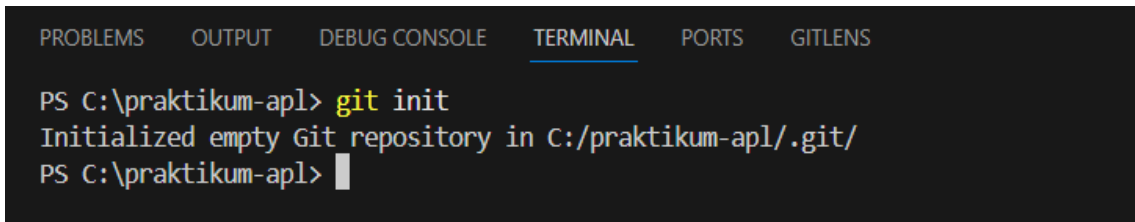
```
Daftar Koleksi
1. Mona Lisa
Masukkan nomor koleksi yang ingin diubah/perbarui : 3
Nomor tidak valid
```

Gambar 4.1 Skenario 9

## 5. Langkah Langkah GIT

### 1. Git Init

Git init merupakan langkah pertama untuk memulai git, Perintah git init digunakan untuk membuat sebuah direktori bernama .git di dalam proyek kita.



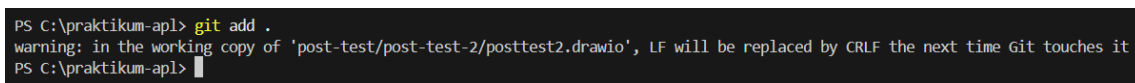
```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS

PS C:\praktikum-apl> git init
Initialized empty Git repository in C:/praktikum-apl/.git/
PS C:\praktikum-apl> █
```

Gambar 5.1 Git Init

### 2. Git Add

Selanjutnya adalah Git Add, Perintah ini digunakan untuk menambahkan file baru pada repository yang akan dipilih. Gambar dibawah ini menggunakan perintah “Git Add .” untuk menambah semua file ke dalam repository. Selain itu, kita juga dapat menggunakan “Git Add <nama file>” untuk menambahkan file tertentu saja.

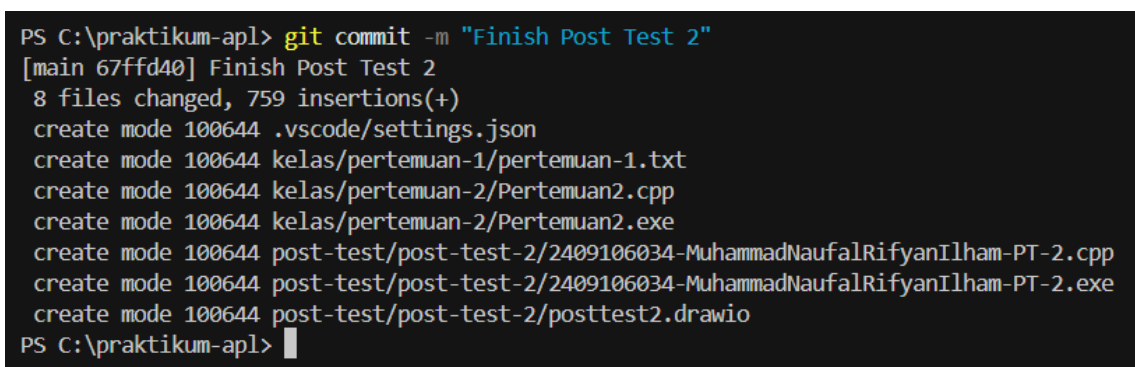


```
PS C:\praktikum-apl> git add .
warning: in the working copy of 'post-test/post-test-2/posttest2.drawio', LF will be replaced by CRLF the next time Git touches it
PS C:\praktikum-apl> █
```

Gambar 5.2 Git Add

### 3. Git Commit

Setelah menambah file pada repository, maka langkah selanjutnya adalah Git Commit. Git Commit adalah perintah Git untuk menyimpan perubahan versi revisi pada repository Git. Jadi, setiap kita melakukan “commit”, Git akan membuat dan menyimpan history revisi pada repository proyek kita.

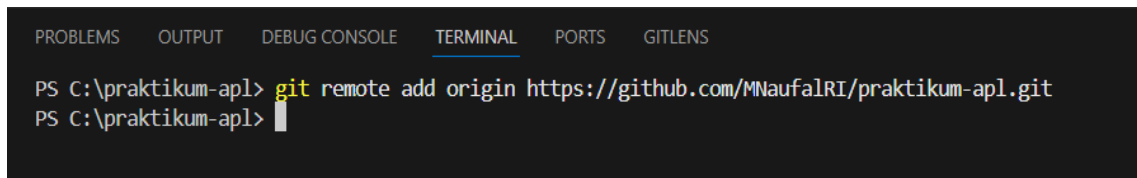


```
PS C:\praktikum-apl> git commit -m "Finish Post Test 2"
[main 67ffd40] Finish Post Test 2
8 files changed, 759 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-1/pertemuan-1.txt
create mode 100644 kelas/pertemuan-2/Pertemuan2.cpp
create mode 100644 kelas/pertemuan-2/Pertemuan2.exe
create mode 100644 post-test/post-test-2/2409106034-MuhammadNaufalRifyanIlham-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106034-MuhammadNaufalRifyanIlham-PT-2.exe
create mode 100644 post-test/post-test-2/posttest2.drawio
PS C:\praktikum-apl> █
```

Gambar 5.3 Git Commit

#### 4. Git Remote

Langkah selanjutnya adalah Git Remote, git remote adalah perintah dalam Git yang digunakan untuk mengelola daftar repositori jarak jauh (remote repositories) yang terhubung dengan repositori lokal.

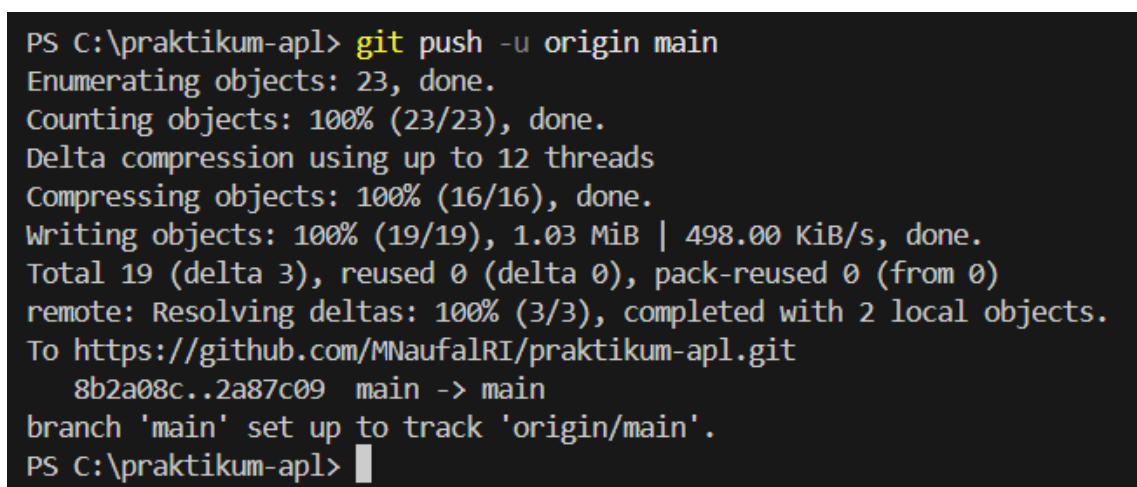


```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
PS C:\praktikum-apl> git remote add origin https://github.com/MNaufalRI/praktikum-apl.git
PS C:\praktikum-apl>
```

Gambar 5.4 Git Remote

#### 5. Git Push

Langkah yang terakhir adalah Git Push, dimana perintah ini akan berfungsi untuk mengirimkan perubahan ke master branch dari remote repository yang berhubungan dengan direktori kerja.



```
PS C:\praktikum-apl> git push -u origin main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (19/19), 1.03 MiB | 498.00 KiB/s, done.
Total 19 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 2 local objects.
To https://github.com/MNaufalRI/praktikum-apl.git
    8b2a08c..2a87c09  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apl>
```

Gambar 5.5 Git Push