

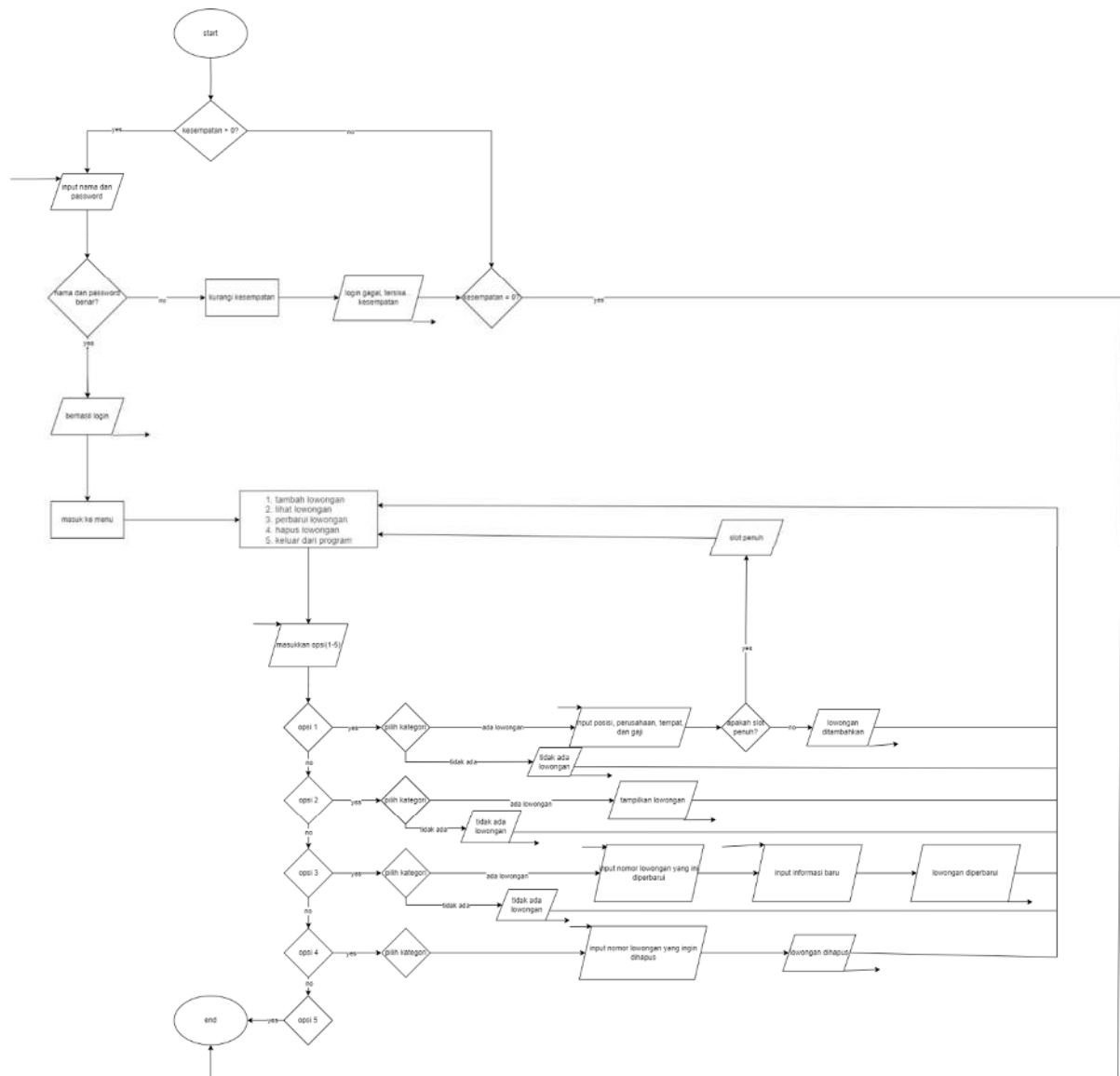
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Rangga Lawe (2409106043)**  
**Kelas (A2 '24)**

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

# 1. Flowchart



## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini adalah sistem manajemen lowongan pekerjaan yang memungkinkan pengguna untuk menambahkan, melihat, memperbarui, dan menghapus lowongan kerja berdasarkan kategori level pekerjaan (Entry-Level, Mid-Level, Senior-Level, dan Managerial). Program juga dilengkapi dengan sistem login yang membatasi tiga kali percobaan sebelum keluar dari program.

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

#### **1. Login Pengguna:**

- Program meminta input nama dan NIM.
- Jika pengguna menginput nama dan NIM yang sesuai yaitu "Rangga" dan "2409106043", maka login berhasil.
- Jika tidak, pengguna diberi kesempatan hingga tiga kali untuk login. Jika gagal tiga kali, program berhenti.

#### **2. Menampilkan Menu Utama:**

- Program menampilkan pilihan menu:
  1. Tambah Lowongan
  2. Lihat Lowongan
  3. Perbarui Lowongan
  4. Hapus Lowongan
  5. Keluar
- Pengguna memilih salah satu opsi yang tersedia.

#### **3. Tambah Lowongan:**

- Pengguna memilih kategori lowongan (Entry-Level, Mid-Level, Senior-Level, Managerial).
- Jika kategori valid dan belum penuh, pengguna menginput detail lowongan (posisi, perusahaan, tempat, gaji).
- Data lowongan disimpan dalam array dua dimensi sesuai kategorinya.

#### **4. Lihat Lowongan:**

- Pengguna memilih kategori lowongan.

- Program menampilkan daftar lowongan dalam kategori yang dipilih.
- Jika tidak ada lowongan dalam kategori tersebut, program menampilkan pesan bahwa kategori kosong.

#### 5. **Perbarui Lowongan:**

- Pengguna memilih kategori lowongan.
- Jika kategori memiliki lowongan, pengguna memilih nomor lowongan yang ingin diperbarui.
- Pengguna memasukkan data baru untuk lowongan tersebut.

#### 6. **Hapus Lowongan:**

- Pengguna memilih kategori lowongan.
- Jika kategori memiliki lowongan, pengguna memilih nomor lowongan yang ingin dihapus.
- Data lowongan yang dipilih akan dihapus.

#### 7. **Keluar dari Program:**

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

#### **Struktur Data yang Digunakan:**

- **Struct Lowongan** untuk menyimpan informasi setiap lowongan kerja (posisi, perusahaan, tempat, gaji).
- **Array dua dimensi daftarLowongan[MAX\_KATEGORI][MAX\_LOWONGAN]** untuk menyimpan daftar lowongan berdasarkan kategori.
- **Array jumlahLowongan[MAX\_KATEGORI]** untuk menyimpan jumlah lowongan dalam masing-masing kategori.

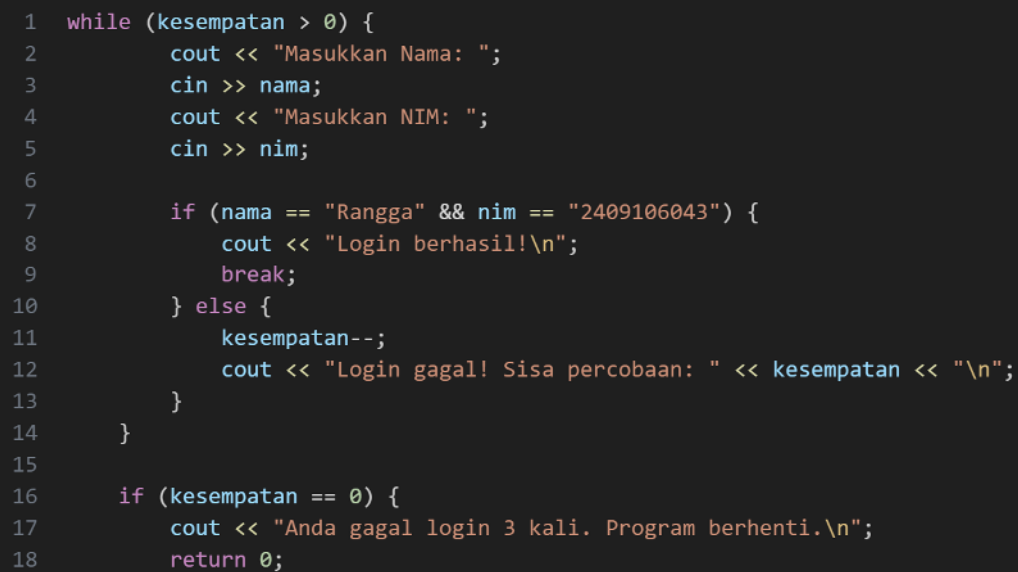
#### **Algoritma Utama:**

- Menggunakan perulangan do-while untuk terus menampilkan menu hingga pengguna memilih untuk keluar dari program.
- Menggunakan switch-case untuk menangani setiap pilihan menu.
- Menggunakan struktur data array untuk menyimpan dan mengelola data lowongan pekerjaan.
- Menggunakan perulangan for untuk menampilkan, memperbarui, dan menghapus data lowongan.

### 3. Source Code

#### A. Fitur login

Fitur ini memungkinkan pengguna untuk login dengan nama dan NIM yang sesuai, dan jika tidak atau pengguna salah menginputnya sebanyak 3 kali, maka program akan terhenti



```
1  while (kesempatan > 0) {
2      cout << "Masukkan Nama: ";
3      cin >> nama;
4      cout << "Masukkan NIM: ";
5      cin >> nim;
6
7      if (nama == "Rangga" && nim == "2409106043") {
8          cout << "Login berhasil!\n";
9          break;
10     } else {
11         kesempatan--;
12         cout << "Login gagal! Sisa percobaan: " << kesempatan << "\n";
13     }
14 }
15
16 if (kesempatan == 0) {
17     cout << "Anda gagal login 3 kali. Program berhenti.\n";
18     return 0;
19 }
```

#### B . Fitur tambah data

Pengguna memilih kategori pekerjaan (Entry-Level, Mid-Level, Senior-Level, atau Managerial), kemudian mengisi Posisi, Perusahaan, Tempat, dan Gaji, lalu lowongan ditambahkan jika kategori belum penuh.

```

1  case 1:
2      cout << R"(
3  0. Entry-Level
4  1. Mid-Level
5  2. Senior-Level
6  3. Managerial
7  Pilih kategori: )";
8      cin >> kategori;
9
10     if (kategori >= 0 && kategori < MAX_KATEGORI && jumlahLowongan[kategori] < MAX_LOWONGAN) {
11         index = jumlahLowongan[kategori];
12         cout << "Posisi: ";
13         cin >> daftarLowongan[kategori][index].posisi;
14         cout << "Perusahaan: ";
15         cin >> daftarLowongan[kategori][index].perusahaan;
16         cout << "Tempat: ";
17         cin >> daftarLowongan[kategori][index].tempat;
18         cout << "Gaji: ";
19         cin >> daftarLowongan[kategori][index].gaji;
20         jumlahLowongan[kategori]++;
21         cout << "Lowongan ditambahkan!\n";
22     } else {
23         cout << "Kategori tidak valid atau penuh!\n";
24     }
25     break;
26

```

### C. Fitur lihat data

Pengguna memilih kategori pekerjaan, lalu sistem menampilkan Nomor, Posisi, Perusahaan, Tempat, dan Gaji untuk setiap lowongan, namun jika tidak ada lowongan, sistem menampilkan pesan "Tidak ada lowongan di kategori ini!".

```

1  case 2:
2      cout << R"(
3  0. Entry-Level
4  1. Mid-Level
5  2. Senior-Level
6  3. Managerial
7  Pilih kategori: )";
8      cin >> kategori;
9
10     if (kategori >= 0 && kategori < MAX_KATEGORI && jumlahLowongan[kategori] > 0) {
11         for (int i = 0; i < jumlahLowongan[kategori]; i++) {
12             cout << i + 1 << ". " << daftarLowongan[kategori][i].posisi << " - "
13                 << daftarLowongan[kategori][i].perusahaan << " ("
14                 << daftarLowongan[kategori][i].tempat << ") Rp"
15                 << daftarLowongan[kategori][i].gaji << "\n";
16         }
17     } else {
18         cout << "Tidak ada lowongan di kategori ini!\n";
19     }
20     break;
21

```

#### D. Fitur perbarui data

Pengguna memilih kategori pekerjaan, kemudian memilih nomor lowongan yang ingin diperbarui, lalu mengganti Posisi, Perusahaan, Tempat, dan Gaji, dan data lowongan diperbarui setelah perubahan disimpan.



```
1  case 3:
2      cout << R"(
3  0. Entry-Level
4  1. Mid-Level
5  2. Senior-Level
6  3. Managerial
7  Pilih kategori: )";
8      cin >> kategori;
9
10     if (kategori >= 0 && kategori < MAX_KATEGORI && jumlahLowongan[kategori] > 0) {
11         cout << "Nomor lowongan: ";
12         cin >> index;
13         if (index > 0 && index <= jumlahLowongan[kategori]) {
14             index--;
15             cout << "Posisi baru: ";
16             cin >> daftarLowongan[kategori][index].posisi;
17             cout << "Perusahaan baru: ";
18             cin >> daftarLowongan[kategori][index].perusahaan;
19             cout << "Tempat baru: ";
20             cin >> daftarLowongan[kategori][index].tempat;
21             cout << "Gaji baru: ";
22             cin >> daftarLowongan[kategori][index].gaji;
23             cout << "Lowongan diperbarui!\n";
24         } else {
25             cout << "Nomor tidak valid!\n";
26         }
27     } else {
28         cout << "Kategori tidak valid atau kosong!\n";
29     }
30     break;
```

#### E. Fitur hapus data

Pengguna memilih kategori pekerjaan, kemudian memilih nomor lowongan yang ingin dihapus, lalu lowongan dihapus dari daftar, dan sistem memperbarui daftar lowongan setelah penghapusan.

```

1  case 4:
2      cout << R"(
3  0. Entry-Level
4  1. Mid-Level
5  2. Senior-Level
6  3. Managerial
7  Pilih kategori: )";
8      cin >> kategori;
9
10     if (kategori >= 0 && kategori < MAX_KATEGORI && jumlahLowongan[kategori] > 0) {
11         cout << "Nomor lowongan: ";
12         cin >> index;
13         if (index > 0 && index <= jumlahLowongan[kategori]) {
14             for (int i = index - 1; i < jumlahLowongan[kategori] - 1; i++) {
15                 daftarLowongan[kategori][i] = daftarLowongan[kategori][i + 1];
16             }
17             jumlahLowongan[kategori]--;
18             cout << "Lowongan dihapus!\n";
19         } else {
20             cout << "Nomor tidak valid!\n";
21         }
22     } else {
23         cout << "Kategori tidak valid atau kosong!\n";
24     }
25     break;

```

#### F. Fitur keluar dari program

Pengguna memilih opsi untuk keluar dari program dan program akan berhenti.

```

1  case 5:
2      cout << "Keluar dari program.\n";
3      break;
4
5      default:
6          cout << "Pilihan tidak valid!\n";
7          break;

```



## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Skenario 1: Login Berhasil

- Input: Nama = Rangga, NIM = 2409106043
- Ekspektasi: Program menampilkan "Login berhasil!" dan masuk ke menu utama.

Skenario 2: Login Gagal 3 Kali

- Input: Nama dan NIM yang salah tiga kali berturut-turut.
- Ekspektasi: Program menampilkan "Login gagal!" hingga kesempatan habis, lalu menampilkan "Anda gagal login 3 kali. Program berhenti." dan program berhenti.

Skenario 3: Menambahkan Lowongan

- Input:
  - Pilihan: 1 (Tambah Lowongan)
  - Kategori: 0 (Entry-Level)
  - Posisi: Helper
  - Perusahaan: KPC
  - Tempat: Sangatta
  - Gaji: 900000
- Ekspektasi: Program menampilkan "Lowongan ditambahkan!" dan data tersimpan.

Skenario 4: Melihat Lowongan yang Telah Ditambahkan

- Input:
  - Pilihan: 2 (Lihat Lowongan)
  - Kategori: 0 (Entry-Level)

- Ekspektasi: Program menampilkan daftar lowongan yang sudah ditambahkan.

#### Skenario 5: Memperbarui Lowongan

- Input:
  - Pilihan: 3 (Perbarui Lowongan)
  - Kategori: 0 (Entry-Level)
  - Nomor lowongan: 1
  - Posisi baru: Operator
  - Perusahaan baru: KPC
  - Tempat baru: Sangatta
  - Gaji baru: 950000
- Ekspektasi: Program menampilkan "Lowongan diperbarui!" dan data lowongan diperbarui.

#### Skenario 6: Menghapus Lowongan

- Input:
  - Pilihan: 4 (Hapus Lowongan)
  - Kategori: 0 (Entry-Level)
  - Nomor lowongan: 1
- Ekspektasi: Program menampilkan "Lowongan dihapus!" dan lowongan dihapus dari daftar.

#### Skenario 7: Keluar dari Program

- Input: Pilihan: 5 (Keluar)
- Ekspektasi: Program menampilkan "Keluar dari program." dan berhenti.

## 4.2 Hasil Output

Skenario 1(Login berhasil)

```
Masukkan Nama: Rangga
Masukkan NIM: 2409106043
Login berhasil!
```

Skenario 2(Login gagal)

```
Masukkan Nama: rangga
Masukkan NIM: 12
Login gagal! Sisa percobaan: 2
Masukkan Nama: rangga
Masukkan NIM: 21
Login gagal! Sisa percobaan: 1
Masukkan Nama: rangga
Masukkan NIM: 11
Login gagal! Sisa percobaan: 0
Anda gagal login 3 kali. Program berhenti.
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl\Post-test\posttest 2>
```

Skenario 3(Menambah data lowongan)

```
0. Entry-Level
1. Mid-Level
2. Senior-Level
3. Managerial
Pilih kategori: 0
Posisi: Helper
Perusahaan: KPC
Tempat: Sangatta
Gaji: 900000
Lowongan ditambahkan!
```

Skenario 4(Menampilkan data lowongan)

```
1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Pilih: 2

0. Entry-Level
1. Mid-Level
2. Senior-Level
3. Managerial
Pilih kategori: 0
1. Helper - KPC (Sangatta) Rp900000
```

Skenario 5 (Memperbarui data lowongan)

```
1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Pilih: 3
```

```
0. Entry-Level
1. Mid-Level
2. Senior-Level
3. Managerial
Pilih kategori: 0
Nomor lowongan: 1
Posisi baru: Operator
Perusahaan baru: KPC
Tempat baru: Sangatta
Gaji baru: 950000
Lowongan diperbarui!
```

Skenario 6(Menghapus data lowongan)

```
1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Pilih: 4
```

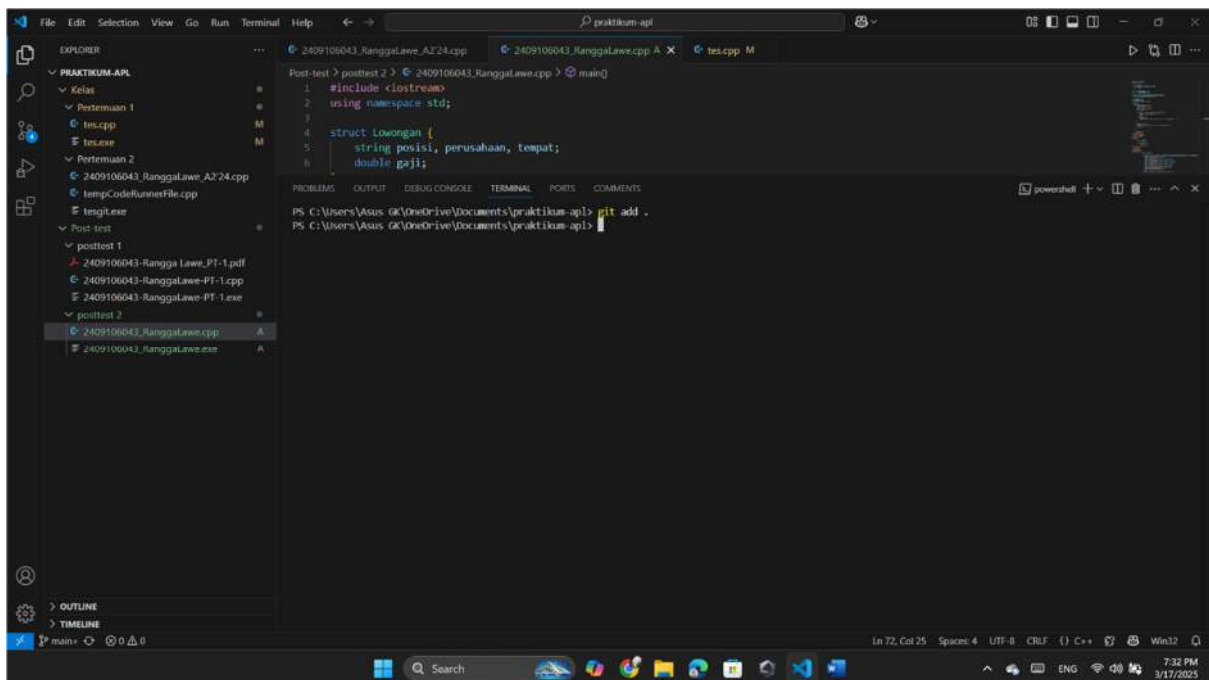
```
0. Entry-Level
1. Mid-Level
2. Senior-Level
3. Managerial
Pilih kategori: 0
Nomor lowongan: 1
Lowongan dihapus!
```

Skenario 7(keluar dari program)

```
0. Entry-Level
1. Mid-Level
2. Senior-Level
3. Managerial
Pilih kategori: 0
Nomor lowongan: 1
Lowongan dihapus!

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Pilih: 5
Keluar dari program.
```

## 5. Langkah-langkah Git



Git add . ini untuk menyiapkan semua perubahan di file agar bisa disimpan.

```
Post-test > posttest 2 > 2409106043_RanggaLawe.cpp > main()
1 #include <iostream>
2 using namespace std;
3
4 struct lowongan {
5     string posisi, perusahaan, tempat;
6     double gaji;
7 }
```

```
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl> git add .
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl> git commit -m "Finish posttest 2"
[main 012d3ae] finish posttest 2
4 files changed, 193 insertions(+), 23 deletions(-)
create mode 100644 Post-test/posttest 2/2409106043_RanggaLawe.cpp
create mode 100644 Post-test/posttest 2/2409106043_RanggaLawe.exe
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl>
```

Git commit ini untuk menyimpan perubahan ke dalam repository lokal.

```
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl> git add .
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl> git commit -m "Finish posttest 2"
[main 012d3ae] finish posttest 2
4 files changed, 193 insertions(+), 23 deletions(-)
create mode 100644 Post-test/posttest 2/2409106043_RanggaLawe.cpp
create mode 100644 Post-test/posttest 2/2409106043_RanggaLawe.exe
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl> git push
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 16 threads
compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 495.41 KiB | 3.96 MiB/s, done.
Total 10 (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/Ridkun/praktikum-apl.git
5e2a55e..012d3ae main -> main
PS C:\Users\Asus G\OneDrive\Documents\praktikum-apl>
```

Git push ini untuk mengirim perubahan ke GitHub atau repository online supaya orang lain bisa melihatnya.