

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

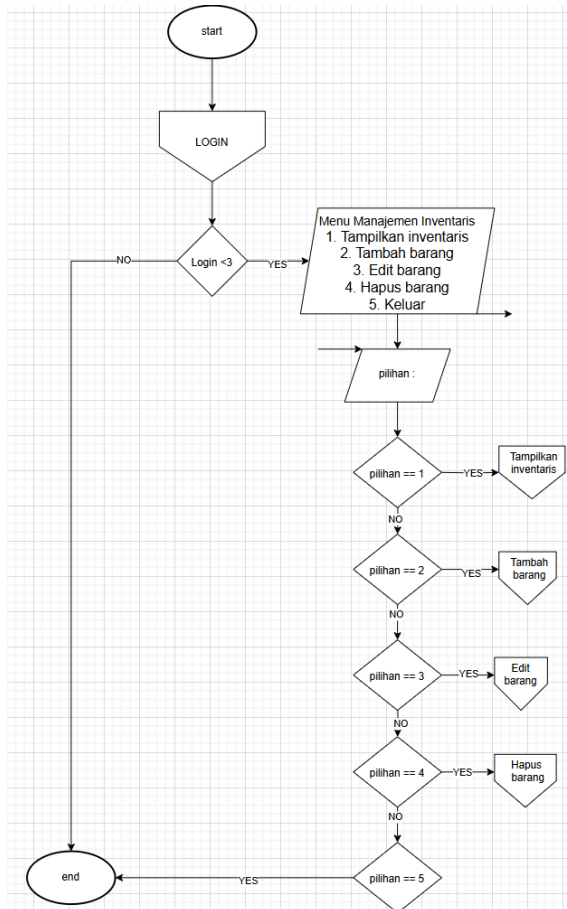


**Disusun oleh:**  
**Angelina Augustin Umami (2409106014)**  
**Kelas A1'24**

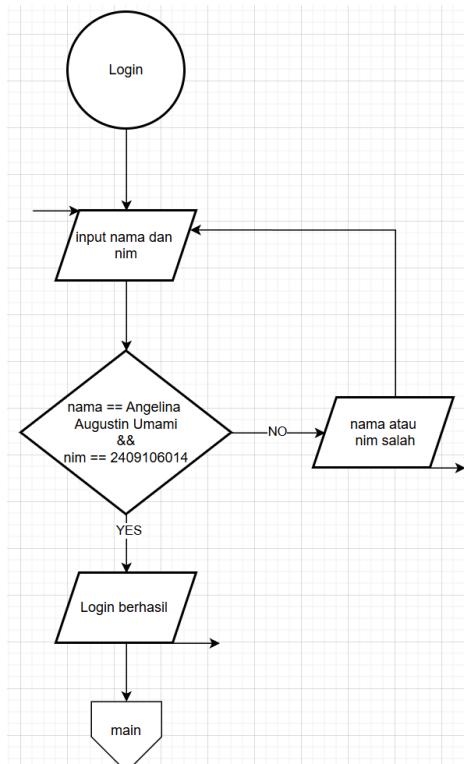
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart

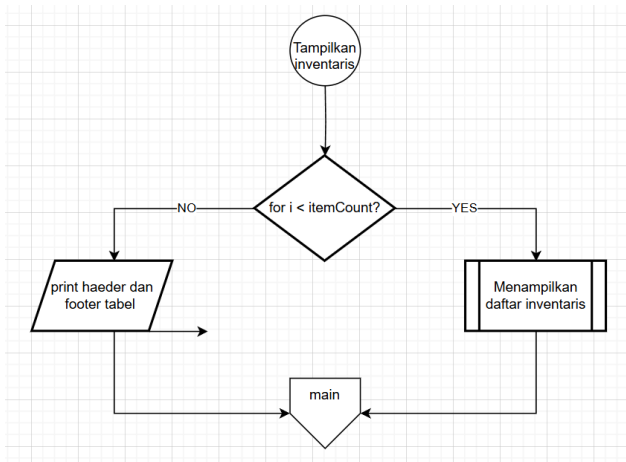
### 1. Menu Main



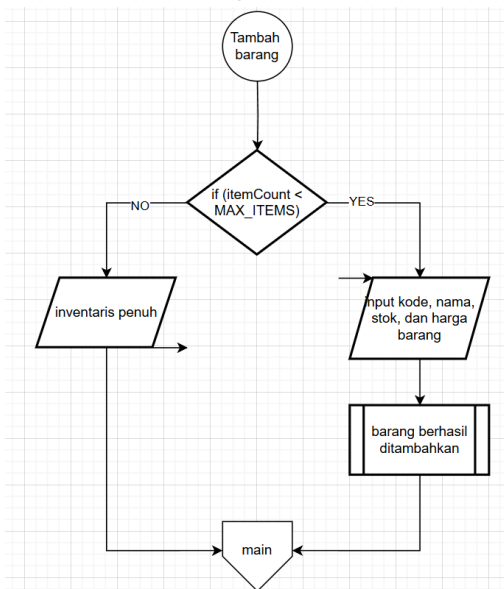
### 2. Menu Login



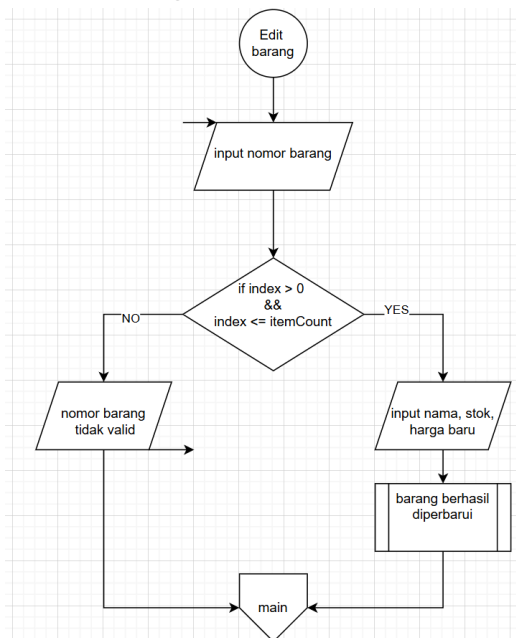
### 3. Menampilkan Inventaris



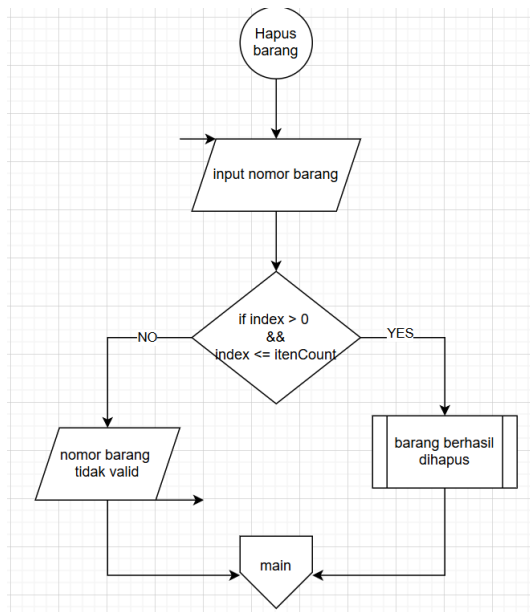
#### 4. Tambah Barang



#### 5. Edit Barang



#### 6. Hapus Barang



## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Tujuan :

Program ini dibuat untuk mengelola inventaris barang secara sederhana, sehingga memudahkan pengguna dalam mencatat, memperbarui, dan menghapus data barang dalam suatu sistem. Dengan adanya program ini, pengguna dapat mengontrol stok barang serta memastikan data barang selalu terupdate dengan mudah.

Selain itu, program ini dilengkapi dengan sistem login untuk memastikan hanya pengguna yang memiliki izin yang dapat mengakses dan mengelola inventaris. Hal ini bertujuan untuk menjaga keamanan dan keakuratan data.

Fungsi dan Manfaat :

1. Menampilkan inventaris : Menampilkan daftar barang beserta kode, stok, dan harga.
2. Menambah barang : Pengguna dapat menambahkan barang baru ke inventaris.
3. Mengedit barang : Memungkinkan perubahan barang pada data barang yang

sudah ada.

4. Menghapus barang : Menghapus barang dari daftar inventaris.
5. Keamanan login : Hanya pengguna terdaftar yang bisa mengakses menu manajemen inventaris.

## **2.2 Penjelasan Alur & Algoritma**

### **2.2.1 Alur Program**

1. Login : Jika berhasil masuk ke menu utama
2. Menu utama : Pengguna memilih fitur untuk menampilkan, menambah, mengedit, atau menghapus barang
3. Proses sesuai pilihan :
  - Tampilkan data barang
  - Tambah barang ke dalam inventaris
  - Edit barang dalam inventaris
  - Hapus barang dari inventaris
4. Keluar dari menu dan ditanya apakah ingin mengulang atau berhenti
5. Program berhenti jika pengguna tidak ingin mengulang

### **2.2.2 Algoritma**

1. Mulai program
2. Login
  - Input nama dan nim
  - Jika benar, lanjut ke menu utama
  - Jika salah, ulang hingga 3 kali. Dan jika gagal maka akan keluar
3. Tampilkan menu utama
  - Pilihan :
    1. Tampilkan inventaris
    2. Tambah barang
    3. Edit barang
    4. Hapus barang
    5. Keluar
4. Eksekusi pilihan pengguna
  - Tampilkan inventaris : menampilkan daftar barang
  - Tambah barang : tambah data barang jika belum penuh
  - Edit barang : perbarui data barang berdasarkan nomor urut
  - Hapus barang : hapus barang dan geser data ke kiri

- Keluar : konfirmasi keluar atau ulangi program

5. Akhiri program

### 2.2.3 Blok Kode

#### 1. Library yang Digunakan

```
#include <iostream>
#include <iomanip>
#include <string>
```

- `#include <iostream>` → Digunakan untuk operasi input-output standar (seperti `cin` dan `cout`).
- `#include <iomanip>` → Digunakan untuk manipulasi format output, seperti `setw()` untuk pengaturan lebar kolom saat mencetak tabel.
- `#include <string>` → Digunakan untuk menangani tipe data string.

#### 2. Deklarasi Konstanta dan Struktur Data

```
using namespace std;

const int MAX_ITEMS = 100;

struct Item {
    string kode;
    string nama;
    int stok;
    double harga;
};
```

- `using namespace std;` → Agar bisa menggunakan fitur dari `std` tanpa perlu menuliskan `std::` di setiap penggunaan fungsi dari standard library.
- `const int MAX_ITEMS = 100;` → Konstanta untuk menentukan jumlah maksimal barang dalam inventaris.
- `struct Item` → Struktur untuk menyimpan data barang, terdiri dari:
  - `kode (string)` → Kode barang.
  - `nama (string)` → Nama barang.
  - `stok (int)` → Jumlah stok barang.
  - `harga (double)` → Harga barang.

#### 3. Deklarasi Variabel Global

```
Item inventory[MAX_ITEMS];
int itemCount = 0;
```

- `inventory[MAX_ITEMS]` → Array untuk menyimpan daftar barang dalam inventaris.
- `itemCount` → Variabel untuk menghitung jumlah barang yang ada di inventaris.

#### 4. Fungsi Menampilkan Inventaris

```

int tampilkanInventaris() {
    cout << "\n===== " << endl;
    cout << " | No | Kode | Nama Barang | Stok | Harga | " << endl;
    cout << "===== " << endl;
    for (int i = 0; i < itemCount; i++) {
        cout << " | " << setw(3) << i + 1 << " | "
            << setw(10) << inventory[i].kode << " | "
            << setw(18) << inventory[i].nama << " | "
            << setw(10) << inventory[i].stok << " | "
            << setw(12) << inventory[i].harga << " | " << endl;
    }
    cout << "===== \n";
    return 0;
}

```

- Menampilkan daftar barang yang ada dalam inventaris.
- Menggunakan setw() untuk mengatur lebar kolom agar tampilan lebih rapi.
- Perulangan for digunakan untuk mencetak setiap barang dalam daftar berdasarkan itemCount.

## 5. Fungsi Menambah Barang

```

int tambahBarang() {
    if (itemCount < MAX_ITEMS) {
        cout << "\nMasukkan kode barang: ";
        cin >> inventory[itemCount].kode;
        cout << "Masukkan nama barang: ";
        cin.ignore();
        getline(cin, inventory[itemCount].nama);
        cout << "Masukkan stok barang: ";
        cin >> inventory[itemCount].stok;
        cout << "Masukkan harga barang: ";
        cin >> inventory[itemCount].harga;
        itemCount++;
        cout << "Barang berhasil ditambahkan!\n";
    } else {
        cout << "Inventaris penuh!\n";
    }
    return 0;
}

```

- Mengecek apakah masih ada slot kosong (itemCount < MAX\_ITEMS).
- Mengambil input pengguna untuk kode, nama, stok, dan harga barang.
- cin.ignore(); digunakan untuk menghindari kesalahan input akibat buffer cin.
- getline(cin, inventory[itemCount].nama); digunakan agar nama barang bisa mengandung spasi.
- itemCount++ menambah jumlah barang setelah berhasil ditambahkan.

## 6. Fungsi Mengedit Barang

```

int editBarang() {
    tampilkanInventaris();
    int index;
    cout << "\nMasukkan nomor barang yang ingin diedit: ";
    cin >> index;
    if (index > 0 && index <= itemCount) {
        index--;
        cout << "Masukkan nama baru: ";
        cin.ignore();
        getline(cin, inventory[index].nama);
        cout << "Masukkan stok baru: ";
        cin >> inventory[index].stok;
        cout << "Masukkan harga baru: ";
        cin >> inventory[index].harga;
        cout << "Barang berhasil diperbarui!\n";
    } else {
        cout << "Nomor barang tidak valid!\n";
    }
    return 0;
}

```

- Memanggil tampilkan Inventaris() agar pengguna dapat melihat daftar barang.



- Meminta nomor barang yang ingin diedit.
- Mengecek apakah nomor barang valid ( $\text{index} > 0 \ \&\& \ \text{index} \leq \text{itemCount}$ ).
- Jika valid, pengguna menginput data baru untuk barang tersebut.

## 7. Fungsi Menghapus Barang

```
int hapusBarang() {
    tampilkanInventaris();
    int index;
    cout << "\nMasukkan nomor barang yang ingin dihapus: ";
    cin >> index;
    if (index > 0 && index <= itemCount) {
        for (int i = index - 1; i < itemCount - 1; i++) {
            inventory[i] = inventory[i + 1];
        }
        itemCount--;
        cout << "Barang berhasil dihapus!\n";
    } else {
        cout << "Nomor barang tidak valid!\n";
    }
    return 0;
}
```

- Menampilkan inventaris sebelum meminta nomor barang yang akan dihapus.
- Mengecek apakah nomor barang valid.
- Jika valid, barang dihapus dengan cara menggeser semua barang setelahnya ke posisi sebelumnya.
- $\text{itemCount--}$  untuk mengurangi jumlah barang.

## 8. Fungsi Login

```
int login() {
    string nama, nim;
    int login_attempts = 3;
    while (login_attempts > 0) {
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        cin >> nim;
        cin.ignore();

        if (nama == "Angelina Augustin Umami" && nim == "2409106014") {
            return 1;
        } else {
            cout << "Nama atau NIM salah! Percobaan tersisa: " << --login_attempts << endl;
        }
    }
    return 0;
}
```

- Meminta input nama dan NIM pengguna.
- Jika nama dan NIM sesuai dengan data yang ditentukan, login berhasil (return 1).
- Jika salah, pengguna memiliki 3 kesempatan sebelum program berhenti.

## 9. Fungsi Utama (main)

```

int main() {
    while (true) {
        if (!login()) {
            cout << "Anda telah mencoba login 3 kali. Program berhenti." << endl;
            break;
        }

        int pilihan;
        do {
            cout << "\n===== " << endl;
            cout << "|          Menu Manajemen Inventaris          |" << endl;
            cout << "===== " << endl;
            cout << "| 1 | Tampilkan Inventaris          |" << endl;
            cout << "| 2 | Tambah Barang                  |" << endl;
            cout << "| 3 | Edit Barang                   |" << endl;
            cout << "| 4 | Hapus Barang                  |" << endl;
            cout << "| 5 | Keluar                        |" << endl;
            cout << "===== " << endl;
            cout << "Pilih menu: ";
            cin >> pilihan;
            cin.ignore();

            switch (pilihan) {
                case 1:
                    tampilkanInventaris();
                    break;
                case 2:
                    tambahBarang();
                    break;
                case 3:
                    editBarang();
                    break;
                case 4:
                    hapusBarang();
                    break;
                case 5:
                    cout << "Keluar dari menu manajemen inventaris..." << endl;
                    break;
                default:
                    cout << "Pilihan tidak valid!" << endl;
            }
        } while (pilihan != 5);

        char ulang;
        cout << "Apakah anda ingin mengulang program? (y/n): ";
        cin >> ulang;
        cin.ignore();
        if (ulang != 'y' && ulang != 'Y') {
            cout << "Program berhenti." << endl;
            break;
        }
    }
    return 0;
}

```

- Memanggil login.
- Menampilkan menu utama.
- Memproses pilihan pengguna menggunakan switch-case.
- Jika pengguna memilih keluar (5), program berhenti.

### 3. Source Code

#### 3.1 Login

Fitur ini penting karena memberikan kontrol akses untuk masuk ke dalam sistem inventaris.

Source Code :

```
int login() {
    string nama, nim;
    int login_attempts = 3;
    while (login_attempts > 0) {
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        cin >> nim;
        cin.ignore();

        if (nama == "Angelina Augustin Umami" && nim == "2409106014") {
            return 1;
        } else {
            cout << "Nama atau NIM salah! Percobaan tersisa: " << --
login_attempts << endl;
        }
    }
    return 0;
}
```

#### 3.2 Menampilkan Inventaris

Fitur ini penting karena memungkinkan pengguna melihat daftar barang yang ada di dalam inventaris

```
int tampilkanInventaris() {
    cout <<
"\n=====
=====" << endl;
    cout << " | No   | Kode       | Nama Barang       | Stok   | Harga
|" << endl;
    cout <<
"=====
=====" << endl;
    for (int i = 0; i < itemCount; i++) {
        cout << " | " << setw(3) << i + 1 << " | "
            << setw(10) << inventory[i].kode << " | "
            << setw(18) << inventory[i].nama << " | "
            << setw(10) << inventory[i].stok << " | "
            << setw(12) << inventory[i].harga << " |" << endl;
    }
    cout <<
"=====
```

```

===\n";
    return 0;
}

```

### 3.3 Menambah Barang

Fitur ini penting karena untuk memperbarui stok dan menambahkan barang baru ke dalam sistem

```

int tambahBarang() {
    if (itemCount < MAX_ITEMS) {
        cout << "\nMasukkan kode barang: ";
        cin >> inventory[itemCount].kode;
        cout << "Masukkan nama barang: ";
        cin.ignore();
        getline(cin, inventory[itemCount].nama);
        cout << "Masukkan stok barang: ";
        cin >> inventory[itemCount].stok;
        cout << "Masukkan harga barang: ";
        cin >> inventory[itemCount].harga;
        itemCount++;
        cout << "Barang berhasil ditambahkan!\n";
    } else {
        cout << "Inventaris penuh!\n";
    }
    return 0;
}

```

### 3.4 Mengedit Barang

Fungsi ini penting karena memungkinkan pengguna untuk memperbarui informasi barang jika terjadi perubahan harga dan stok

```

int editBarang() {
    tampilkanInventaris();
    int index;
    cout << "\nMasukkan nomor barang yang ingin diedit: ";
    cin >> index;
    if (index > 0 && index <= itemCount) {
        index--;
        cout << "Masukkan nama baru: ";
        cin.ignore();
        getline(cin, inventory[index].nama);
        cout << "Masukkan stok baru: ";
        cin >> inventory[index].stok;
        cout << "Masukkan harga baru: ";
        cin >> inventory[index].harga;
        cout << "Barang berhasil diperbarui!\n";
    } else {
        cout << "Nomor barang tidak valid!\n";
    }
}

```

```

    return 0;
}

```

### 3.5 Menghapus Barang

Fitur ini penting karena untuk menghapus barang yang sudah tidak tersedia dan tidak dibutuhkan lagi

```

int hapusBarang() {
    tampilkanInventaris();
    int index;
    cout << "\nMasukkan nomor barang yang ingin dihapus: ";
    cin >> index;
    if (index > 0 && index <= itemCount) {
        for (int i = index - 1; i < itemCount - 1; i++) {
            inventory[i] = inventory[i + 1];
        }
        itemCount--;
        cout << "Barang berhasil dihapus!\n";
    } else {
        cout << "Nomor barang tidak valid!\n";
    }
    return 0;
}

```

### 3.6 Menu Navigasi Interaktif

Fitur ini penting karena memudahkan pengguna untuk mengelola inventaris

```

int main() {
    while (true) {
        if (!login()) {
            cout << "Anda telah mencoba login 3 kali. Program berhenti." << endl;
            break;
        }

        int pilihan;
        do {
            cout << "\n===== " << endl;
            cout << "|          Menu Manajemen Inventaris          |" << endl;
            cout << "===== " << endl;
            cout << "| 1 | Tampilkan Inventaris                    |" << endl;
            cout << "| 2 | Tambah Barang                          |" << endl;
            cout << "| 3 | Edit Barang                            |" << endl;
            cout << "| 4 | Hapus Barang                           |" << endl;
            cout << "| 5 | Keluar                                 |" << endl;
            cout << "===== " << endl;
            cout << "Pilih menu: ";
            cin >> pilihan;

```

```
        cin.ignore();

        switch (pilihan) {
            case 1:
                tampilkanInventaris();
                break;
            case 2:
                tambahBarang();
                break;
            case 3:
                editBarang();
                break;
            case 4:
                hapusBarang();
                break;
            case 5:
                cout << "Keluar dari menu manajemen inventaris..." << endl;
                break;
            default:
                cout << "Pilihan tidak valid!" << endl;
        }
    } while (pilihan != 5);

    char ulang;
    cout << "Apakah anda ingin mengulang program? (y/n): ";
    cin >> ulang;
    cin.ignore();
    if (ulang != 'y' && ulang != 'Y') {
        cout << "Program berhenti." << endl;
        break;
    }
}

return 0;
```

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

### **4.1 Uji Coba**

#### **1. Skenario 1**

- Percobaan login pertama input nama dan nim yang salah
- Login kedua benar
- Memilih menu 2
- Tambah barang yg ingin dimasukkan
- Memilih menu 1
- Lihat barang di dalam baru dimasukkan ke inventaris
- Memilih menu 5

#### **2. Skenario 2**

- Percobaan login pertama input nama dan nim benar
- Memilih menu 3
- Edit barang yang ingin diubah
- Memilih menu 4
- Hapus barang
- Memilih menu 5

## 4.2 Hasil Output

### Skenario 1

```
PS C:\Users\ASUS> cd "c:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl\post-test\post-test-2\" ; if ($?) { g++ 2409106014-AngelinaAugustinUmami-PT-2.cpp -o 2409106014-AngelinaAugustinUmami-PT-2 } ; if ($?) { .\2409106014-AngelinaAugustinUmami-PT-2 }
Masukkan Nama: angel
Masukkan NIM: 234
Nama atau NIM salah! Percobaan tersisa: 2
Masukkan Nama: Angelina Augustin Umami
Masukkan NIM: 2409106014

=====
|           Menu Manajemen Inventaris           |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 2

Masukkan kode barang: rajut
Masukkan nama barang: cardigan
Masukkan stok barang: 40
Masukkan harga barang: 65000
Barang berhasil ditambahkan!

=====
|           Menu Manajemen Inventaris           |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 1

=====
| No | Kode | Nama Barang | Stok | Harga |
=====
| 1  | rajut | cardigan   | 40   | 65000 |
=====
```

```
=====
|           Menu Manajemen Inventaris           |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 5
Keluar dari menu manajemen inventaris...
Apakah anda ingin mengulang program? (y/n): y
```

Gambar 4.1 Skenario 1



## Skenario 2

```
Masukkan Nama: Angelina Augustin Umami
Masukkan NIM: 2409106014

=====
|      Menu Manajemen Inventaris      |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 3

=====
| No | Kode | Nama Barang | Stok | Harga |
=====
| 1  | rajut | cardigan   | 40   | 65000 |
=====

Masukkan nomor barang yang ingin diedit: 1
Masukkan nama baru: sweater
Masukkan stok baru: 55
Masukkan harga baru: 80000
Barang berhasil diperbarui!

=====
|      Menu Manajemen Inventaris      |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 4

=====
| No | Kode | Nama Barang | Stok | Harga |
=====
| 1  | rajut | sweater    | 55   | 80000 |
=====

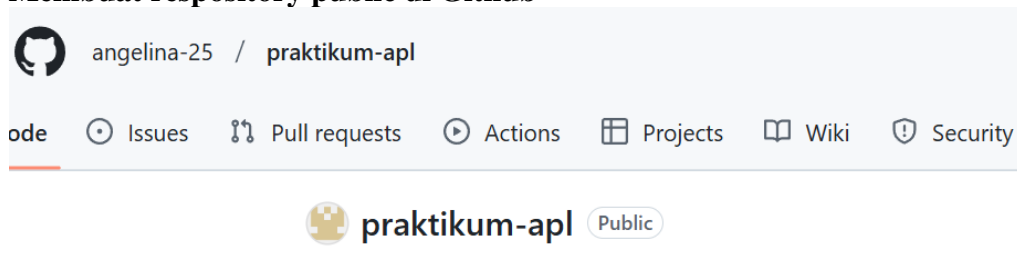
Masukkan nomor barang yang ingin dihapus: 1
Barang berhasil dihapus!

=====
|      Menu Manajemen Inventaris      |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Keluar              |
=====
Pilih menu: 5
Keluar dari menu manajemen inventaris...
Apakah anda ingin mengulang program? (y/n): n
Program berhenti.
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl\post-test\post-test-2> █
```

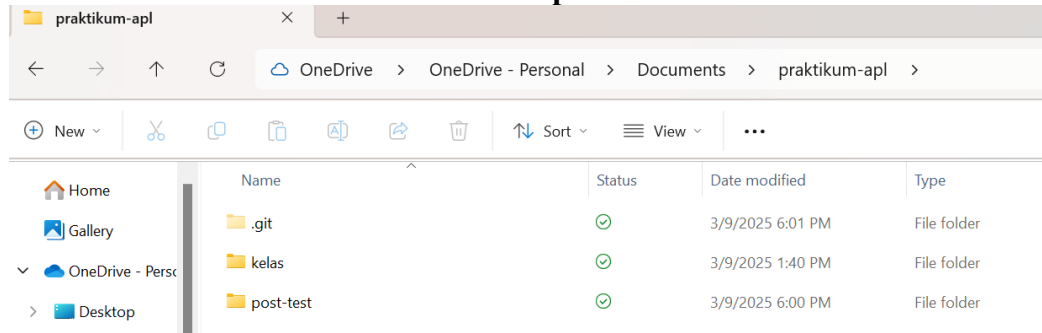
Gambar 4.1 Skenario 2

## 5. Git

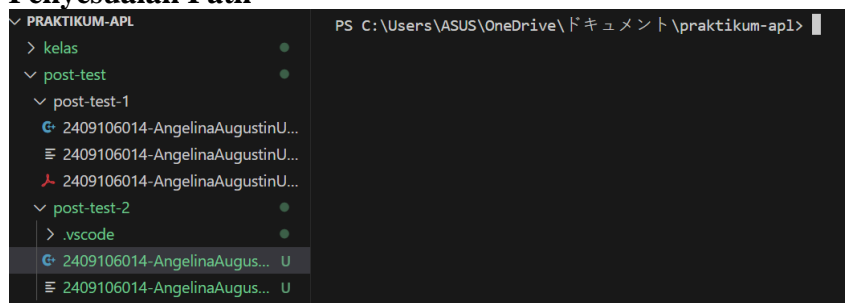
### 5.1 Membuat repository public di Github



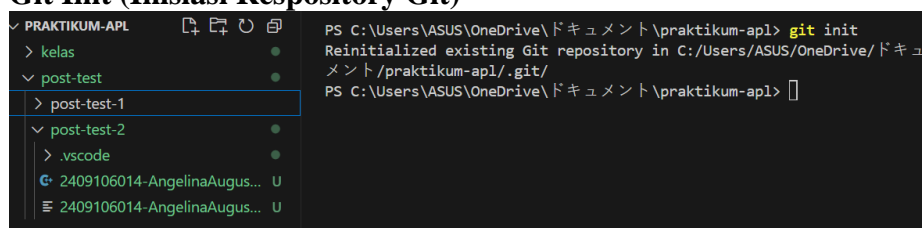
### 5.2 Membuat Folder Praktikum di File Eksplorer



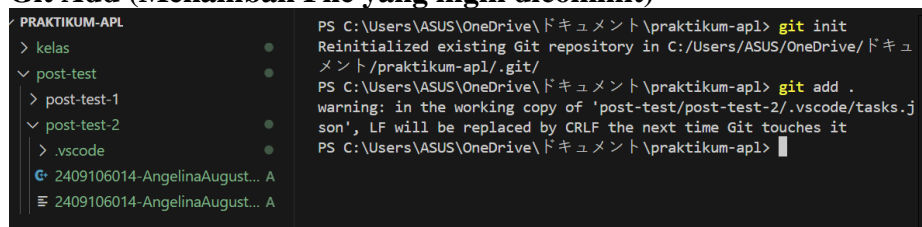
### 5.3 Penyesuaian Path



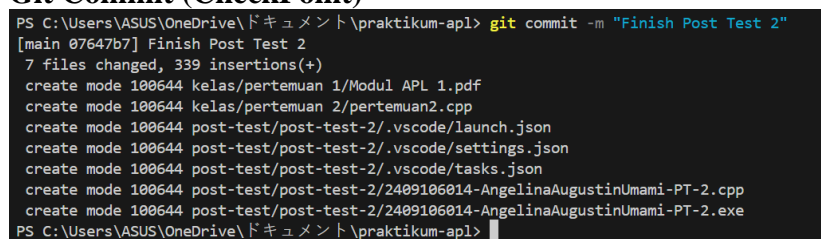
### 5.4 Git Init (Inisiasi Repository Git)



### 5.5 Git Add (Menambah File yang ingin dicommit)



### 5.6 Git Commit (CheckPoint)



## 5.7 Git Remote

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git remote set-url origin https://github.com/angelina-25/praktikum-apl.git
```

## 5.8 Git Push

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git push origin main
>>
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 16 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (13/13), 48.58 KiB | 2.86 MiB/s, done.
Total 13 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/angelina-25/praktikum-apl.git
   501da8c..7c0621e  main -> main
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl>
```