

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

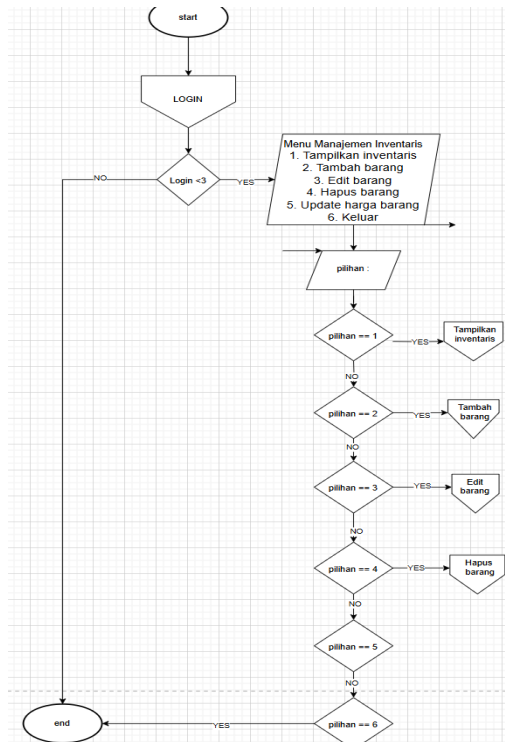


Disusun oleh:
Angelina Augustin Umami (2409106014)
Kelas (A1'24)

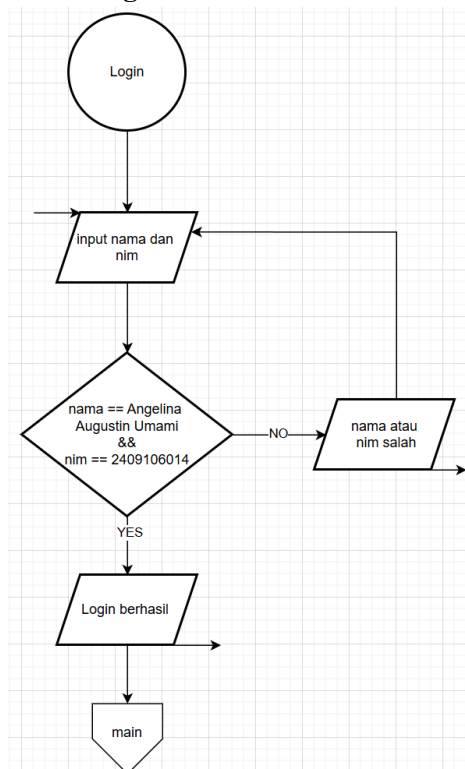
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

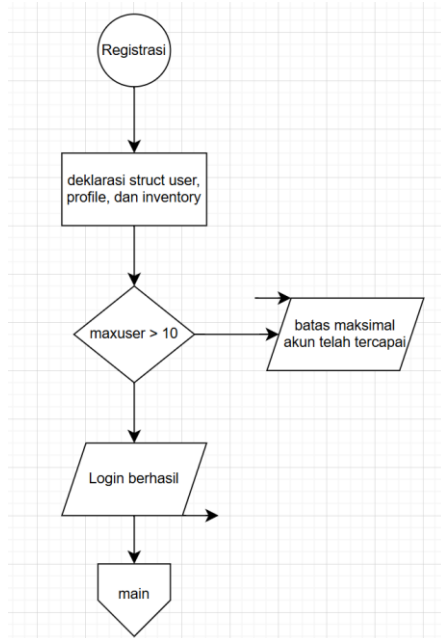
1. Menu Main



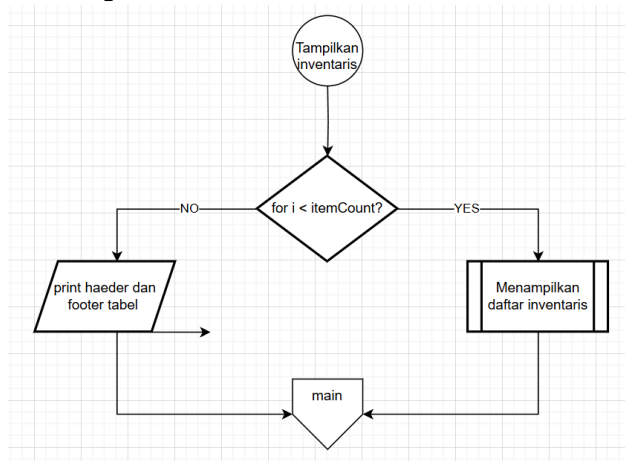
2. Menu Login



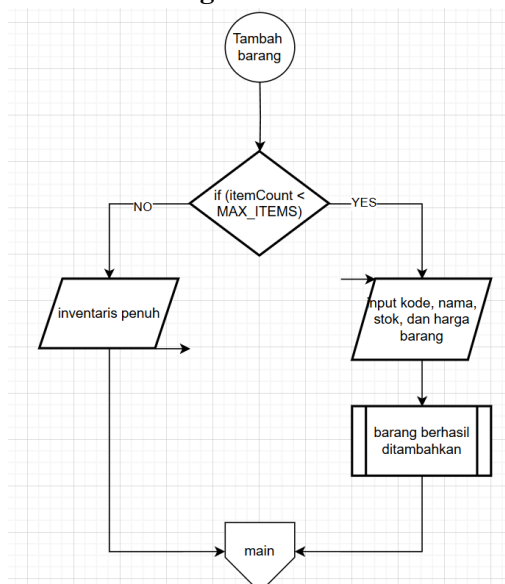
3. Registrasi



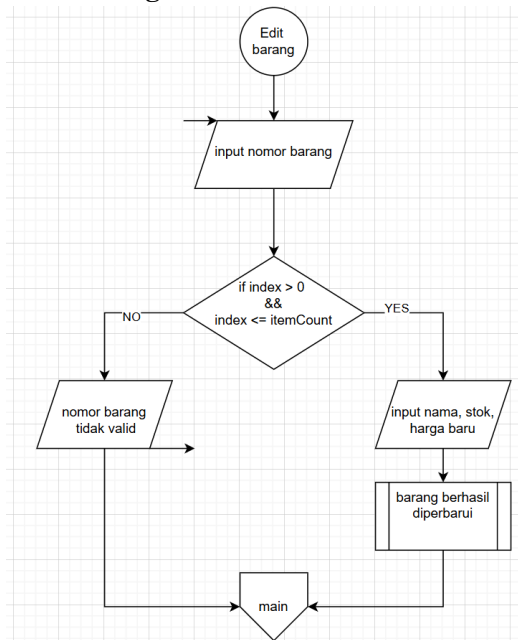
4. Menampilkan Inventaris



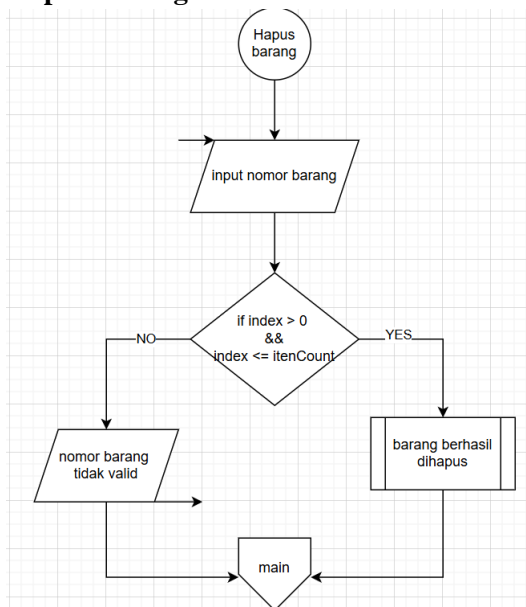
5. Tambah Barang



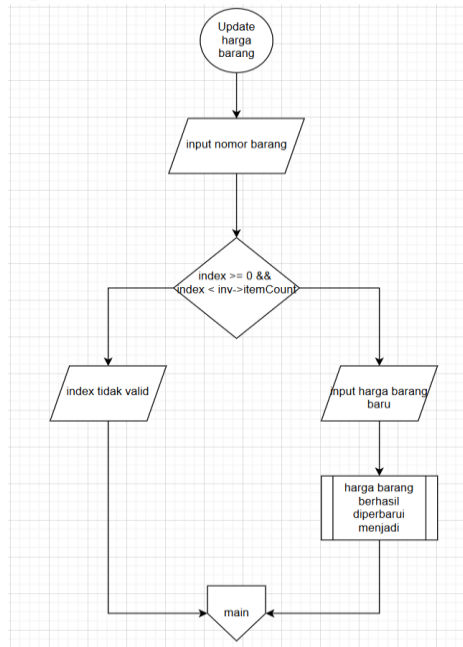
6. Edit Barang



7. Hapus Barang



8. Update Harga Barang



2. Analisis Program

Program ini bertujuan untuk mengelola inventaris toko sederhana yang menggunakan bahasa pemrograman C++. Program ini memudahkan pengguna yang dapat mendaftarkan akun, login kemudian melakukan pengecekan barang yang ada di dalam inventaris toko.

Fungsi program dan manfaat program :

1. Manajemen pengguna:
 - Registrasi akun baru dengan menggunakan nama dan NIM
 - Login menggunakan nama dan NIM yang terdaftar, setelah itu baru bisa mengakses inventaris.
2. Manajemen inventaris barang:
 - Menampilkan daftar barang beserta kode, nama, stok dan harganya
 - Menambahkan barang baru ke dalam inventaris
 - Mengedit barang yang ingin diubah
 - Menghapus barang untuk menghapus barang yang ingin dihapus
 - Keluar, jika ingin keluar dari program
3. Rekursif untuk tampilan data:
 - Menampilkan daftar inventaris menggunakan fungsi rekursif
4. Pembatasan maksimal data:

- Terdapat batasan maksimal untuk jumlah user dan item barang, sesuai dengan kapasitas array statis

3. Source Code

3.1 Register Akun

```
void registerUser(User* users, int* userCount) {
    if (*userCount < MAX_USERS) {
        cout << "\n=== Register Akun ===\n";
        cout << "Masukkan Nama: ";
        cin.ignore();
        cin.getline(users[*userCount].profil.nama, MAX_NAME);
        cout << "Masukkan NIM: ";
        cin.getline(users[*userCount].profil.nim, MAX_NIM);
        cout << "Akun berhasil didaftarkan!\n";
        (*userCount)++;
    } else {
        cout << "\nBatas maksimal akun telah tercapai!\n";
    }
}
```

3.2 Login

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

```
int loginUser(User* users, int* userCount) {

    char nama[MAX_NAME];

    char nim[MAX_NIM];

    int attempts = 3;

    while (attempts-- > 0) {

        cout << "\n=== Login ===\n";

        cout << "Masukkan Nama: ";

        cin.ignore();
```

```

        cin.getline(nama, MAX_NAME);

        cout << "Masukkan NIM: ";

        cin.getline(nim, MAX_NIM);

        for (int i = 0; i < *userCount; i++) {

            if (strcmp(users[i].profil.nama, nama) == 0 &&
                strcmp(users[i].profil.nim, nim) == 0) {

                cout << "Login berhasil!\n";

                return 1;

            }

        }

        cout << "Nama atau NIM salah! Percobaan tersisa: " << attempts << "\n";

    }

    cout << "Gagal login.\n";

    return 0;

}

```

3.3 Tampilkan Inventaris Bubble Sort Descending

```

void tampilkanInventaris_descending( Inventory &inv) {

    bubbleSort_descending(inv);

    cout << "\n=== Daftar Inventaris ===\n";

    for (int i = 0; i < inv.itemCount; i++) {

        cout << i + 1 << ". " << inv.nama[i]

        << " - kode: " << inv.kode[i]

        << " - Stok: " << inv.stok[i]
    }
}

```

```

        << " - Harga: " << inv.harga[i]

        << endl;

    }

}

```

3.4 Tampilkan Inventaris Insertion Sort Ascending

```

void tampilkanInventaris_ascending( Inventory &inv) {

    insertionSort_ascending(inv);

    cout << "\n=== Daftar Inventaris ===\n";

    for (int i = 0; i < inv.itemCount; i++) {

        cout << i + 1 << ". " << inv.nama[i]

        << " - kode: " << inv.kode[i]

        << " - Stok: " << inv.stok[i]

        << " - Harga: " << inv.harga[i]

        << endl;

    }

}

```

3.5 Tampilkan Inventaris Selection Sort

```

void tampilkanInventaris( Inventory &inv) {

    selectionSort(inv);

    cout << "\n=== Daftar Inventaris ===\n";

    for (int i = 0; i < inv.itemCount; i++) {

        cout << i + 1 << ". " << inv.nama[i]

```



```

        << " - kode: " << inv.kode[i]

        << " - Stok: " << inv.stok[i]

        << " - Harga: " << inv.harga[i]

        << endl;

    }

}

```

3.6 Menu Sorting

```

switch (pilihan) {

    case 1:

        int pilih;

        cout << " Menu Sorting " <<endl;

        cout << " 1. Sorting Huruf Berdasarkan Descending" <<endl;

        cout << " 2. Sorting Angka Berdasarkan Ascending" <<endl;

        cout << " 3. Sorting " <<endl;

        cout << " Pilih menu: ";

        cin >> pilih ;

```

3.7 Menu Tampilkan Inventaris

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

```

void tampilkanInventaris(const Inventory &inv, int index = 0) {

```

```

    if (index >= inv.itemCount) return;

    cout << index + 1 << ". " << inv.nama[index] << " - Stok: " << inv.stok[index]
<< " - Harga: " << inv.harga[index] << endl;

    tampilkanInventaris(inv, index + 1);
}

```

3.8 Menu Tambah Barang

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

```

void tambahBarang(Inventory &inv) {

    if (inv.itemCount < MAX_ITEMS) {

        cout << "Masukkan kode barang: ";

        cin.ignore();

        cin.getline(inv.kode[inv.itemCount], MAX_KODE);

        cout << "Masukkan nama barang: ";

        cin.getline(inv.nama[inv.itemCount], MAX_NAMA_BARANG);

        cout << "Masukkan stok barang: ";

        cin >> inv.stok[inv.itemCount];

        cout << "Masukkan harga barang: ";

        cin >> inv.harga[inv.itemCount];

        inv.itemCount++;

        cout << "Barang berhasil ditambahkan!\n";

    } else {

        cout << "Inventaris penuh!\n";

    }
}

```

```
}
```

3.9 Menu Edit Barang

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

```
void editBarang(Inventory &inv) {  
  
    if (inv.itemCount == 0) {  
  
        cout << "Inventaris kosong.\n";  
  
        return;  
  
    }  
  
    cout << "\nDaftar Barang:\n";  
    tampilkanInventaris(inv);  
  
    int nomor;  
  
    cout << "Masukkan nomor barang yang akan diedit: ";  
  
    cin >> nomor;  
  
    cin.ignore();  
  
    if (nomor > 0 && nomor <= inv.itemCount) {  
  
        int index = nomor - 1;  
  
        cout << "Masukkan kode barang baru: ";  
  
        cin.getline(inv.kode[index], MAX_KODE);  
  
        cout << "Masukkan nama barang baru: ";  
  
        cin.getline(inv.nama[index], MAX_NAMA_BARANG);  
    }  
}
```

```

        cout << "Masukkan stok barang baru: ";

        cin >> inv.stok[index];

        cout << "Masukkan harga barang baru: ";

        cin >> inv.harga[index];

        cin.ignore();

        cout << "Barang berhasil diedit!\n";

    } else {

        cout << "Nomor barang tidak valid!\n";

    }

}

```

3.10 Menu Hapus Barang

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

```

void hapusBarang(Inventory &inv) {

    if (inv.itemCount == 0) {

        cout << "Inventaris kosong.\n";

        return;

    }

    cout << "\nDaftar Barang:\n";

    tampilkanInventaris(inv);

    int nomor;

    cout << "Masukkan nomor barang yang akan dihapus: ";

```

```

cin >> nomor;

cin.ignore();

if (nomor > 0 && nomor <= inv.itemCount) {

    int index = nomor - 1;

    for (int i = index; i < inv.itemCount - 1; i++) {

        strcpy(inv.kode[i], inv.kode[i+1]);

        strcpy(inv.nama[i], inv.nama[i+1]);

        inv.stok[i] = inv.stok[i+1];

        inv.harga[i] = inv.harga[i+1];

    }

    inv.itemCount--;

    cout << "Barang berhasil dihapus!\n";

} else {

    cout << "Nomor barang tidak valid!\n";

}

}

```

3.11 Menu Update Harga Barang

```

void updateHargaBarang(Inventory* inv, int index, int* hargaBaru) {

    if (index >= 0 && index < inv->itemCount) {

        inv->harga[index] = *hargaBaru;

        cout << "Harga barang berhasil diperbarui menjadi " << inv->harga[index] <<
endl;

    } else {

```

```

        cout << "Index tidak valid!\n";

    }

}

```

3.12 Menu Inventaris

```

void menuInventaris(Inventory &inv) {

    int pilihan;

    while (true) {

        cout << "\n===== ";

        cout << "\n|      Menu Manajemen Inventaris Toko      |";

        cout << "\n===== ";

        cout << "\n| 1 | Tampilkan Inventaris                |";

        cout << "\n| 2 | Tambah Barang                        |";

        cout << "\n| 3 | Edit Barang                          |";

        cout << "\n| 4 | Hapus Barang                         |";

        cout << "\n| 5 | Logout                               |";

        cout << "\n| 6 | Update Harga Barang (Pointer)        |";

        cout << "\n===== ";

        cout << "\nPilih menu: ";

        cin >> pilihan;

        switch (pilihan) {

            case 1: tampilkanInventaris(inv); break;

            case 2: tambahBarang(inv); break;

            case 3: editBarang(inv); break;

```

```

        case 4: hapusBarang(inv); break;

        case 5: cout << "Logout berhasil.\n"; return;

        case 6: {

            int nomor, hargaBaru;

            cout << "\nDaftar Barang:\n";

            tampilkanInventaris(inv);

            cout << "Masukkan nomor barang: ";

            cin >> nomor;

            if (nomor > 0 && nomor <= inv.itemCount) {

                cout << "Masukkan harga baru: ";

                cin >> hargaBaru;

                updateHargaBarang(&inv, nomor - 1, &hargaBaru);

            } else {

                cout << "Nomor tidak valid!\n";

            }

            break;

        }

        default: cout << "Pilihan tidak valid!\n";

    }

}

```

4. Hasil Output

4.1 pilih register akun dengan memasukkan nama dan nim

```
=====
|      Menu User      |
=====
| 1 | Register Akun  |
| 2 | Login          |
| 3 | Keluar         |
=====
Pilih menu: 1

=== Register Akun ===
Masukkan Nama: angelina
Masukkan NIM: 014
Akun berhasil didaftarkan!
```

4.2 pilih login

```
=====
|      Menu User      |
=====
| 1 | Register Akun  |
| 2 | Login          |
| 3 | Keluar         |
=====
Pilih menu: 2

=== Login ===
Masukkan Nama: angelina
Masukkan NIM: 014
Login berhasil!
```


4.3 Tambah Barang

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                   |
| 4 | Hapus Barang                  |
| 5 | Logout                        |
| 6 | Update Harga Barang (Pointer) |
=====
Pilih menu: 2
Masukkan kode barang: 5
Masukkan nama barang: levis
Masukkan stok barang: 23
Masukkan harga barang: 100000
Barang berhasil ditambahkan!
```

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                   |
| 4 | Hapus Barang                  |
| 5 | Logout                        |
| 6 | Update Harga Barang (Pointer) |
=====
Pilih menu: 2
Masukkan kode barang: 1
Masukkan nama barang: cardigan
Masukkan stok barang: 40
Masukkan harga barang: 80000
Barang berhasil ditambahkan!
```

4.4 Tampilkan Inventaris Sorting huruf berdasarkan descending

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                   |
| 4 | Hapus Barang                  |
| 5 | Logout                        |
| 6 | Update Harga Barang (Pointer) |
=====
Pilih menu: 1
Menu Sorting
1. Sorting Huruf Berdasarkan Descending
2. Sorting Angka Berdasarkan Ascending
3. Sorting
Pilih menu: 1

=== Daftar Inventaris ===
1. levis - kode: 5 - Stok: 23 - Harga: 100000
2. cardigan - kode: 1 - Stok: 40 - Harga: 80000
```

4.5 Tampilkan Inventaris Sorting huruf berdasarkan Ascending

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer)  |
=====
Pilih menu: 1
Menu Sorting
1. Sorting Huruf Berdasarkan Descending
2. Sorting Angka Berdasarkan Ascending
3. Sorting
Pilih menu: 2

=== Daftar Inventaris ===
1. cardigan - kode: 1 - Stok: 40 - Harga: 80000
2. levis - kode: 5 - Stok: 23 - Harga: 100000
```

4.6 Tampilkan Inventaris Sorting

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer)  |
=====
Pilih menu: 1
Menu Sorting
1. Sorting Huruf Berdasarkan Descending
2. Sorting Angka Berdasarkan Ascending
3. Sorting
Pilih menu: 3
Tahap ke-1: cardigan levis

=== Daftar Inventaris ===
1. cardigan - kode: 1 - Stok: 40 - Harga: 80000
2. levis - kode: 5 - Stok: 23 - Harga: 100000
```

4.5 Edit Barang

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer) |
=====
Pilih menu: 3

Daftar Barang:
Tahap ke-1: cardigan levis

=== Daftar Inventaris ===
1. cardigan - kode: 1 - Stok: 40 - Harga: 80000
2. levis - kode: 5 - Stok: 23 - Harga: 100000
Masukkan nomor barang yang akan diedit: 1
Masukkan kode barang baru: 3
Masukkan nama barang baru: kaos
Masukkan stok barang baru: 50
Masukkan harga barang baru: 60000
Barang berhasil diedit!
```

4.6 Hapus Barang

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer) |
=====
Pilih menu: 4

Daftar Barang:
Tahap ke-1: kaos levis

=== Daftar Inventaris ===
1. kaos - kode: 3 - Stok: 50 - Harga: 60000
2. levis - kode: 5 - Stok: 23 - Harga: 100000
Masukkan nomor barang yang akan dihapus: 2
Barang berhasil dihapus!
```

4.7 Logout dari Program

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer)  |
=====
Pilih menu: 5
Logout berhasil.
```

4.8 Update Harga Barang

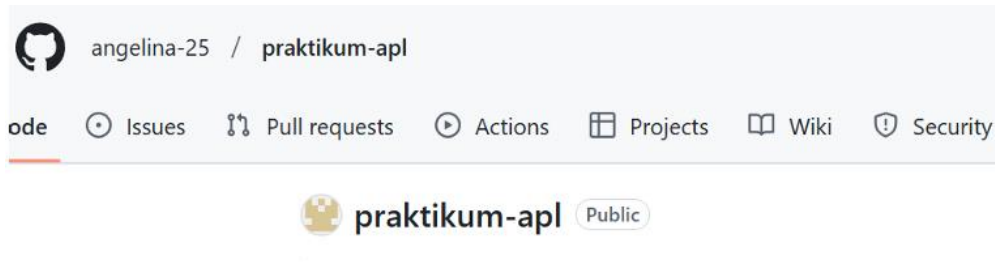
```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
| 6 | Update Harga Barang (Pointer)  |
=====
Pilih menu: 6

Daftar Barang:

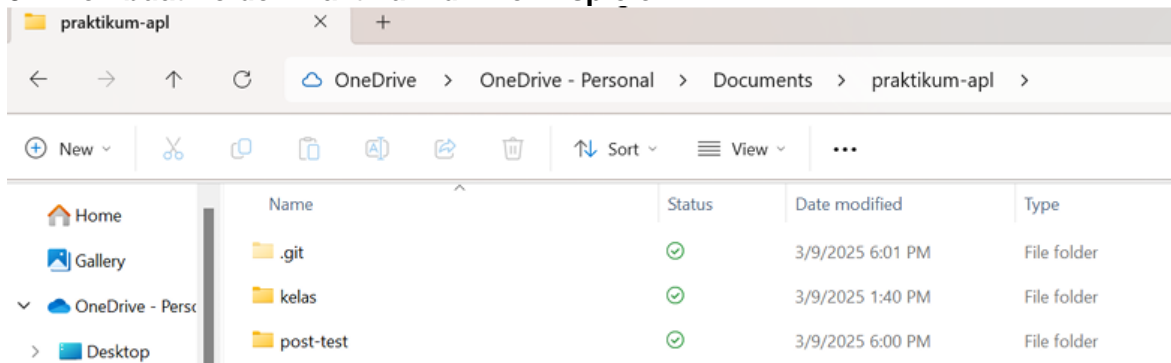
=== Daftar Inventaris ===
1. kaos - kode: 3 - Stok: 50 - Harga: 60000
Masukkan nomor barang: 1
Masukkan harga baru:
70000
Harga barang berhasil diperbarui menjadi 70000
```

5. Langkah-Langkah Git pada VSCode

5.1 Membuat repository public di Github



5.2 Membuat Folder Praktikum di File Eksplorer



5.3 Git Init (Inisiasi Repository Git)

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git init
```

5.4 Git Add (Menambah File yang ingin dicommit)

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git add .
```

5.5 Git Commit (CheckPoint)

```
git commit -m "Finish Post Test 5"
```

5.6 Git Remote

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

5.7 Git Push

```
git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 16 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 45.74 KiB | 5.08 MiB/s, done.
Total 6 (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/angelina-25/praktikum-apl.git
 9385dbe..a99f967 main -> main
PS C:\Users\ASUS\Downloads\praktikum-apl\post-test\post-test 5>
```