

**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**

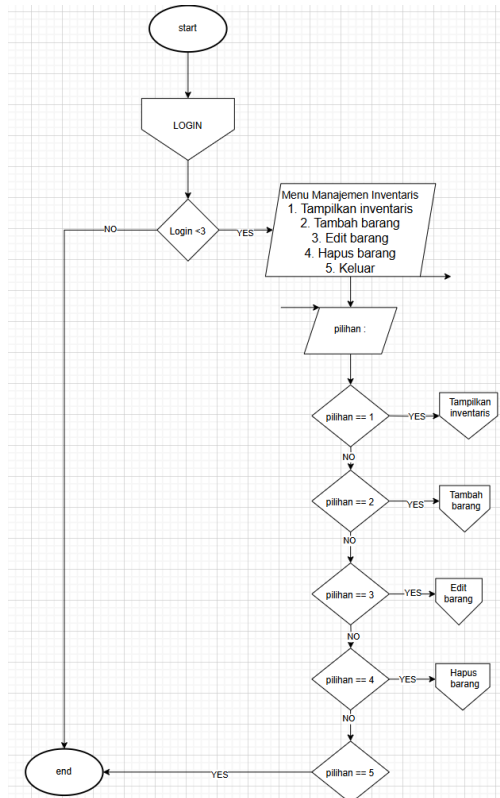


**Disusun oleh:**  
**Angelina Augustin Umami (2409106014)**  
**Kelas (A1'24)**

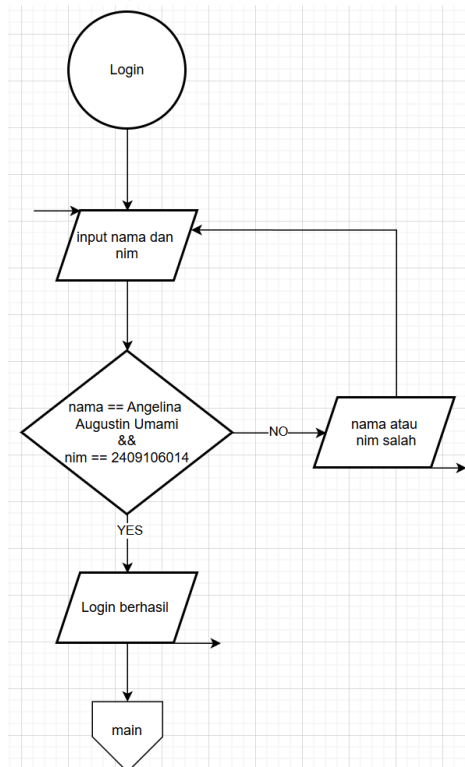
**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart

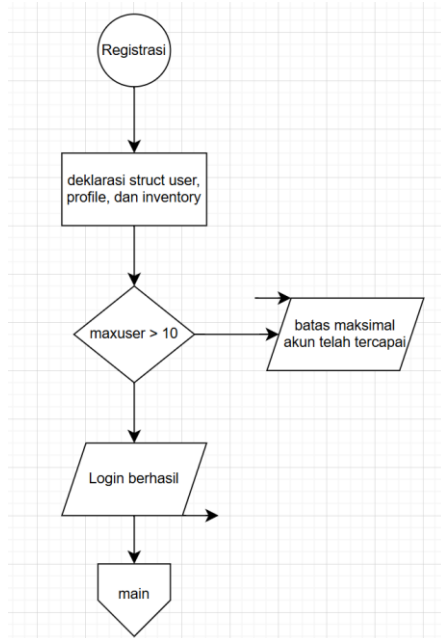
### 1. Menu Main



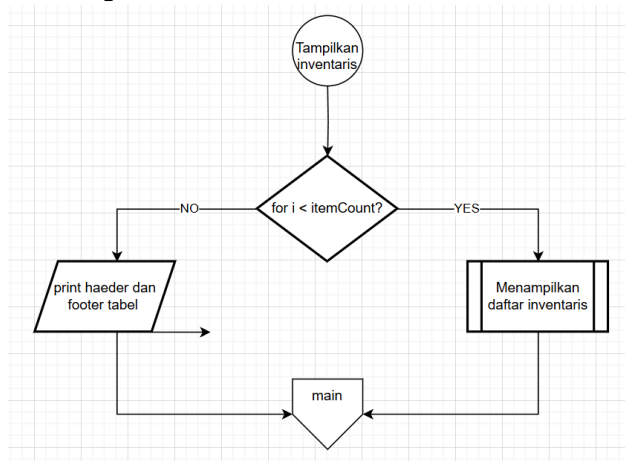
### 2. Menu Login



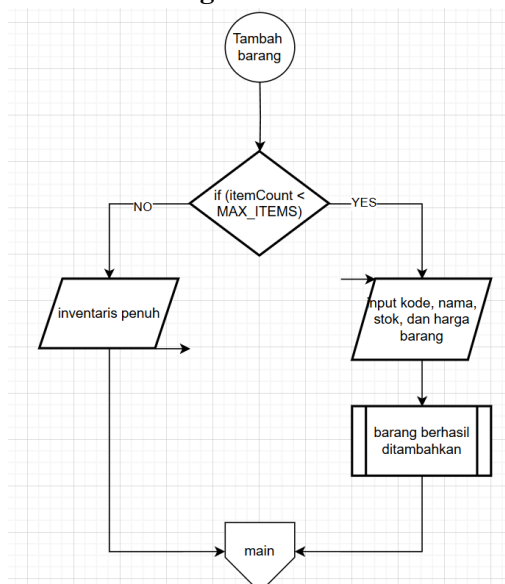
### 3. Registrasi



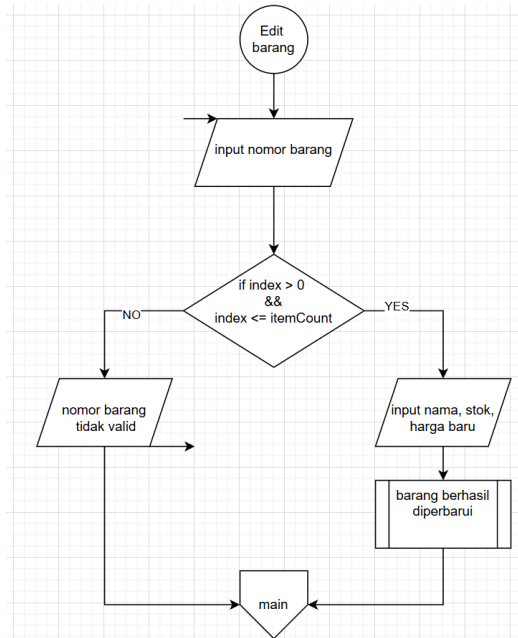
### 4. Menampilkan Inventaris



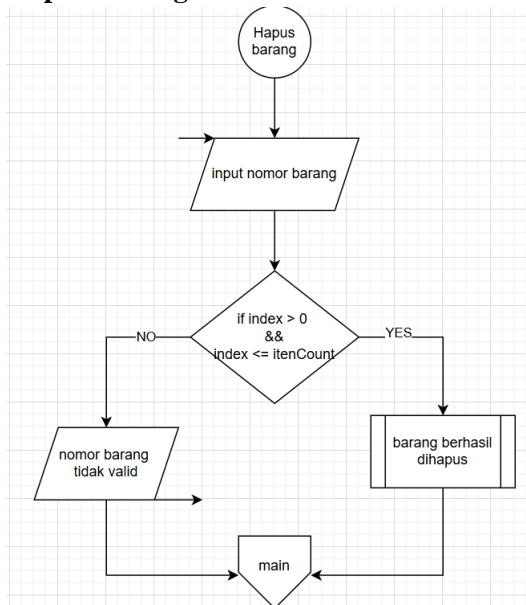
### 5. Tambah Barang



## 6. Edit Barang



## 7. Hapus Barang



## 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 invrntaris 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; // login sukses

            }

        }

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

    }

    cout << "Gagal login.\n";

    return 0; // login gagal

}

```

### 3.3 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);  
  
}
```

### 3.4 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";

    }

}

```

### 3.5 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";

}

}

```

### 3.6 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";
    }
}

```

```
}
```

### 3.7 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=====";  
  
        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;  
  
        }  
    }  
}
```

```

        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: Angel
Masukkan NIM: 2409106014
Akun berhasil didaftarkan!

```

### 4.2 pilih login

```

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

=== Login ===
Masukkan Nama: Angel
Masukkan NIM: 2409106014
Login berhasil!

```

### 4.3 Tambah Barang

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris          |
| 2 | Tambah Barang                 |
| 3 | Edit Barang                   |
| 4 | Hapus Barang                  |
| 5 | Logout                        |
=====
Pilih menu: 2
Masukkan kode barang: 1
Masukkan nama barang: kaos
Masukkan stok barang: 45
Masukkan harga barang: 60000
Barang berhasil ditambahkan!
```

### 4.4 Tampilkan Inventaris

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris          |
| 2 | Tambah Barang                 |
| 3 | Edit Barang                   |
| 4 | Hapus Barang                  |
| 5 | Logout                        |
=====
Pilih menu: 1
1. kaos - Stok: 45 - Harga: 60000
```

### 4.5 Edit Barang

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

Daftar Barang:
1. kaos - Stok: 45 - Harga: 60000
Masukkan nomor barang yang akan diedit: 1
Masukkan kode barang baru: 2
Masukkan nama barang baru: rajut
Masukkan stok barang baru: 50
Masukkan harga barang baru: 100000
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              |
=====
Pilih menu: 4

Daftar Barang:
1. rajut - Stok: 50 - Harga: 100000
Masukkan nomor barang yang akan dihapus: 1
Barang berhasil dihapus!
```

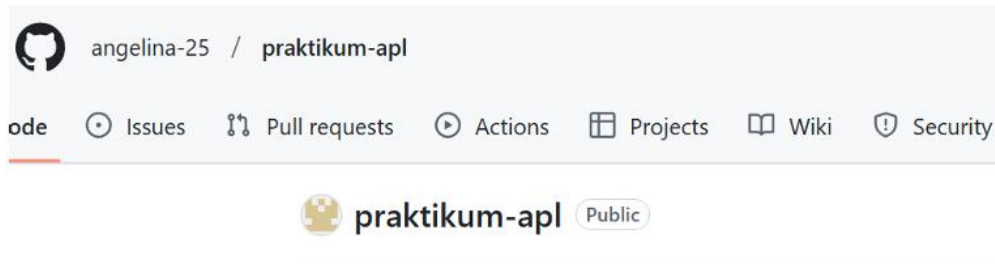
## 4.7 Logout dari Program

```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris |
| 2 | Tambah Barang       |
| 3 | Edit Barang         |
| 4 | Hapus Barang        |
| 5 | Logout              |
=====
Pilih menu: 5
Logout berhasil.

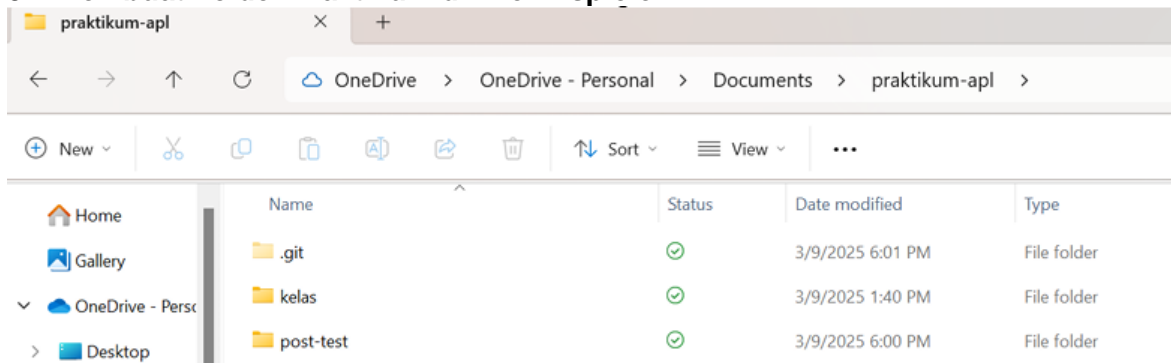
=====
|       Menu User       |
=====
| 1 | Register Akun    |
| 2 | Login            |
| 3 | Keluar           |
=====
Pilih menu: 3
Program berhenti.
```

## 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)

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git commit -m "Finish Post Test 4"
[main (root-commit) 09e9fa0] Finish Post Test 4
```

### 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.git
```

### 5.7 Git Push

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git push origin main
>>
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 16 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (29/29), 7.32 MiB | 820.00 KiB/s, done.
Total 29 (delta 5), reused 9 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), done.
To https://github.com/angelina-25/praktikum-apl.git
 8a85761..87a2c1a  main -> main
```