

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT

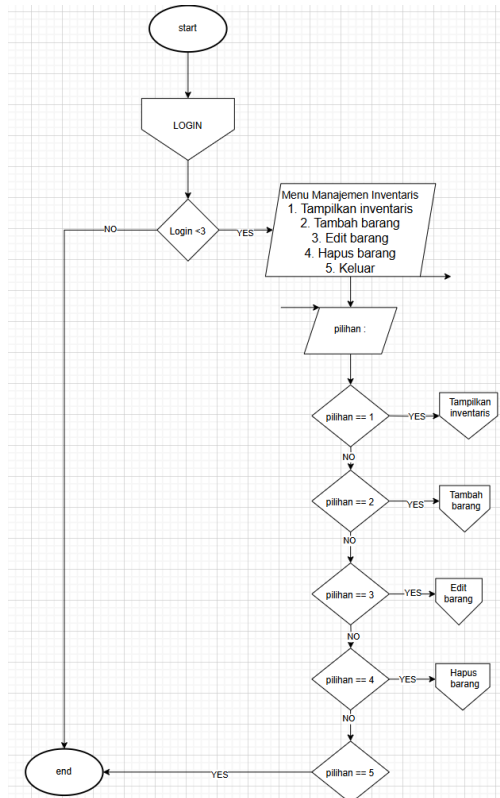


Disusun oleh:
Angelina Augustin Umami (2409106014)
Kelas (A1'24)

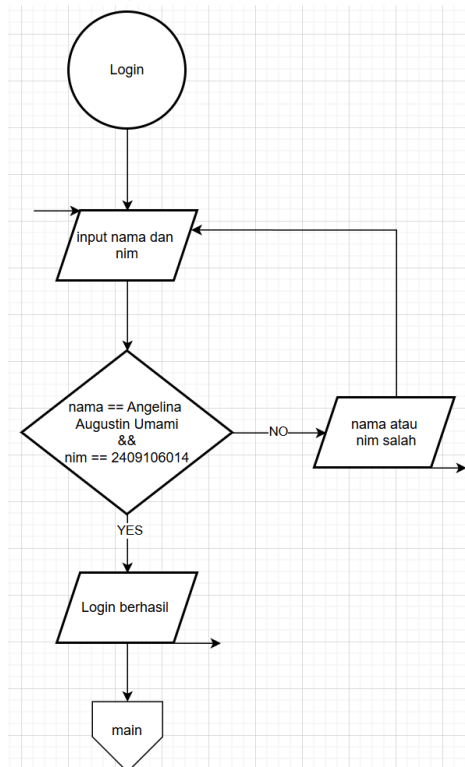
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

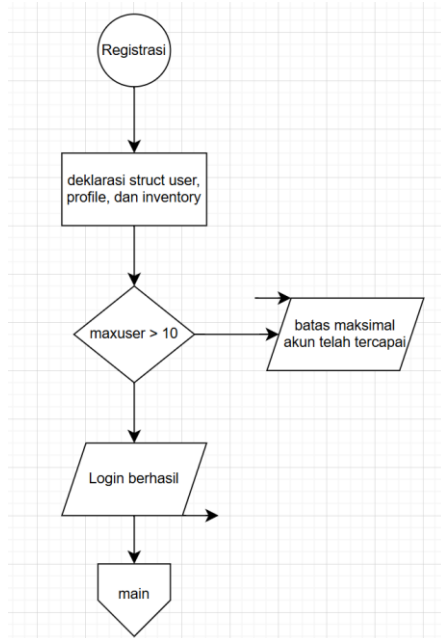
1. Menu Main



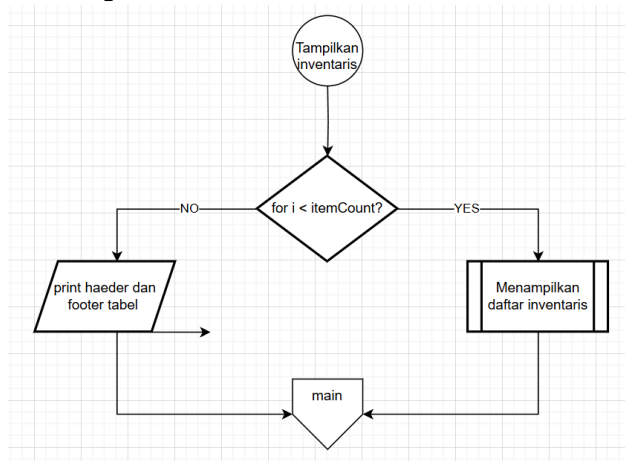
2. Menu Login



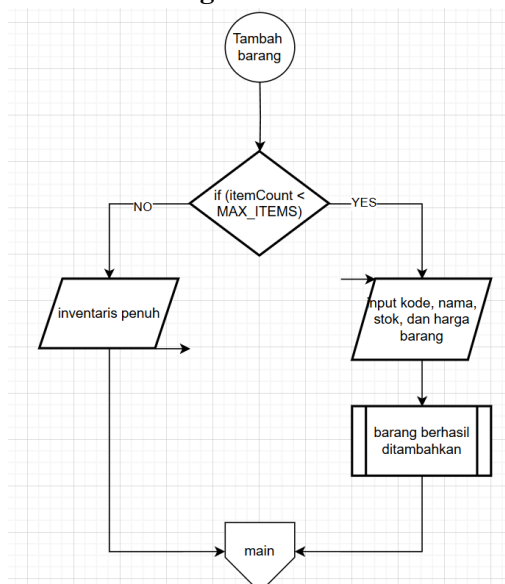
3. Registrasi



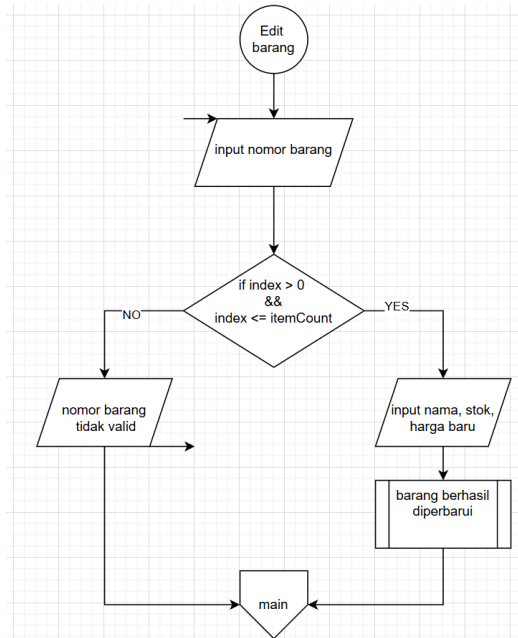
4. Menampilkan Inventaris



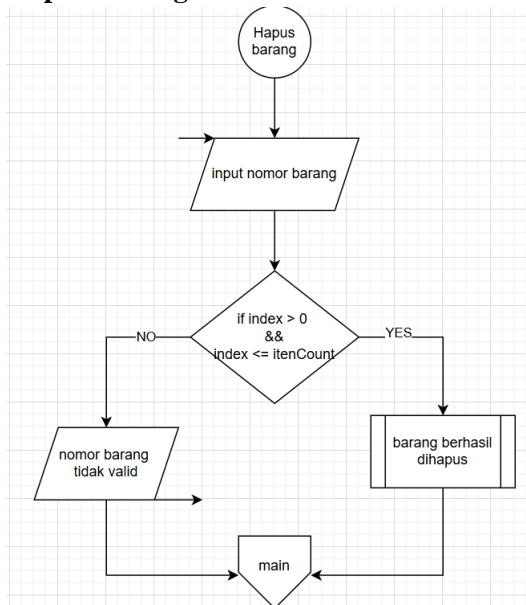
5. Tambah Barang



6. Edit Barang



7. Hapus Barang



2. Analisis Program

Program ini bertujuan untuk mengelola inventaris barang dengan beberapa fitur seperti fitur pendaftaran dan login pengguna.

Fungsi program :

1. Manajemen pengguna:

- Registrasi akun baru dengan menggunakan nama dan NIM

- Login menggunakan nama dan NIM yang terdaftar.
2. Manajemen inventaris:
- 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

Manfaat program :

- Meningkatkan efisiensi dalam pencatatan dan pengelolaan barang
- Mencegah akses yang tidak terdaftar dengan menggunakan sistem login
- Memudahkan pengguna untuk pemantauan stok barang dengan menampilkan inventaris yang rapi.

3. Source Code

3.1 Register Akun

```
int main() {

    User users[MAX_USERS];

    int userCount = 0;

    Inventory inventory;

    inventory.itemCount = 0;

    int pilihan = 0;

    while (true) {

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

        cout << "|      Menu User      |" << endl;

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

        cout << "| 1 | Register Akun   |" << endl;

        cout << "| 2 | Login           |" << endl;
```

```

cout << "| 3 | Keluar          |" << endl;

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

cout << "Pilih menu: ";

cin >> pilihan;

cin.ignore();

if (pilihan == 1) {

    if (userCount < MAX_USERS) {

        cout << "\n== Register Akun ==\n";

        cout << "Masukkan Nama: ";

        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.

```

} else if (pilihan == 2) {

    if (userCount == 0) {

        cout << "\nBelum ada akun yang terdaftar. Silakan register terlebih dahulu!\n";

    }

}

```

```

        continue;

    }

    char nama[MAX_NAME];

    char nim[MAX_NIM];

    int attempts = 3;

    bool loggedIn = false;

    while (attempts > 0) {

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

        cout << "Masukkan Nama: ";

        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";

                loggedIn = true;

                break;

            }

        }

        if (loggedIn) break;

        attempts--;
    }

```

```

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

    }

    if (!loggedIn) {

        cout << "Gagal login. Program berhenti.\n";

        break;

    }

```

3.3 Menu Tampilkan Inventaris

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

```

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;

    cin.ignore();

```



```

        if (pilihan == 1) {

            cout << "\nInventaris:" << endl;

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

                cout << (i+1) << ". " << inventory.nama[i] << " - Stok: "
<< inventory.stok[i] << " - Harga: " << inventory.harga[i] << endl;

            }

```

3.4 Menu Tambah Barang

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

```

} else if (pilihan == 2) {

    if (inventory.itemCount < MAX_ITEMS) {

        cout << "Masukkan kode barang: ";

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

        cout << "Masukkan nama barang: ";

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

        cout << "Masukkan stok barang: ";

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

        cout << "Masukkan harga barang: ";

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

        cin.clear();

        cin.ignore();

        inventory.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

```

} else if (pilihan == 3){

    int nomor;

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

    cin >> nomor;

    cin.ignore();

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

        int index = nomor - 1;

        cout << "Masukkan kode barang baru: ";

        cin.getline(inventory.kode[index], MAX_KODE);

        cout << "Masukkan nama barang baru: ";

        cin.getline(inventory.nama[index], MAX_NAMA_BARANG);

        cout << "Masukkan stok barang baru: ";

        cin >> inventory.stok[index];

        cout << "Masukkan harga barang baru: ";

        cin >> inventory.harga[index];

        cin.clear();
    }
}

```

```

        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

```

}else if (pilihan == 4){

    int nomor;

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

    cin >> nomor;

    cin.ignore();

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

        int index = nomor - 1;

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

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

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

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

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

        }

        inventory.itemCount--;

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

```

```

    } else {

        cout << "Nomor barang 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: 1234
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: 1234
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: 22
Masukkan nama barang: kaos
Masukkan stok barang: 44
Masukkan harga barang: 50000
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

Inventaris:
1. kaos - Stok: 44 - Harga: 50000

```

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
Masukkan nomor barang yang akan diedit: 1
Masukkan kode barang baru: 22
Masukkan nama barang baru: rajut
Masukkan stok barang baru: 23
Masukkan harga barang baru: 60000
Barang berhasil diedit!

```

4.6 Hapus Barang

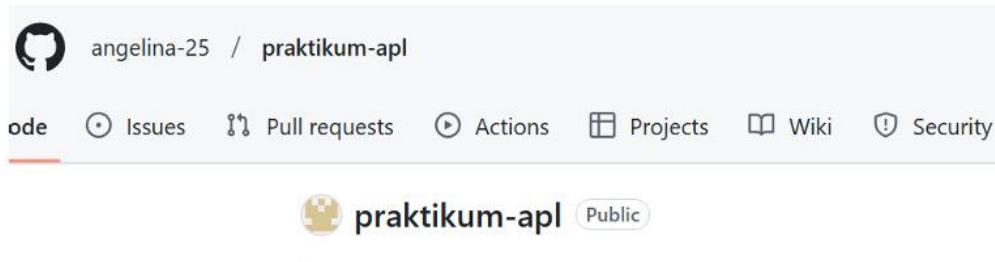
```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
=====
Pilih menu: 4
Masukkan nomor barang yang akan dihapus: 1
Barang berhasil dihapus!
```

4.7 Logout dari Program

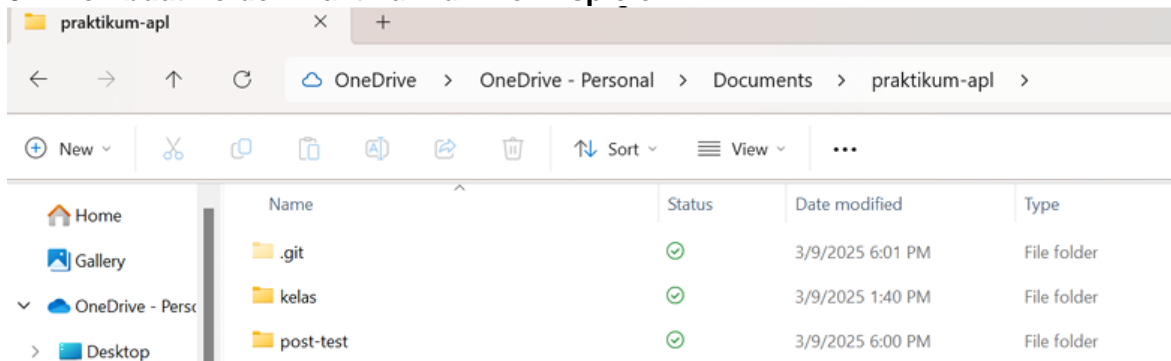
```
=====
|   Menu Manajemen Inventaris Toko   |
=====
| 1 | Tampilkan Inventaris           |
| 2 | Tambah Barang                  |
| 3 | Edit Barang                    |
| 4 | Hapus Barang                   |
| 5 | Logout                         |
=====
Pilih menu: 5
Logout berhasil.
=====
|      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



5.3 Git Init (Inisiasi Respository Git)

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/ASUS/OneDrive/ドキュメント/praktikum-apl/.git/
```

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 3"
[main d13f8ec] Finish Post Test 3
6 files changed, 357 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan 3/pertemuan3.cpp
create mode 100644 post-test/post-test-3/2409106014-AngelinaAugustinUmami-PT-3.cpp
create mode 100644 post-test/post-test-3/2409106014-AngelinaAugustinUmami-PT-3.exe
create mode 100644 post-test/post-test-3/tempCodeRunnerFile.cpp
```

5.6 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
```

5.7 Git Push

```
PS C:\Users\ASUS\OneDrive\ドキュメント\praktikum-apl> git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 16 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (14/14), 46.60 KiB | 4.66 MiB/s, done.
Total 14 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/angelina-25/praktikum-apl
4dda263..d13f8ec main -> main
```