

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT

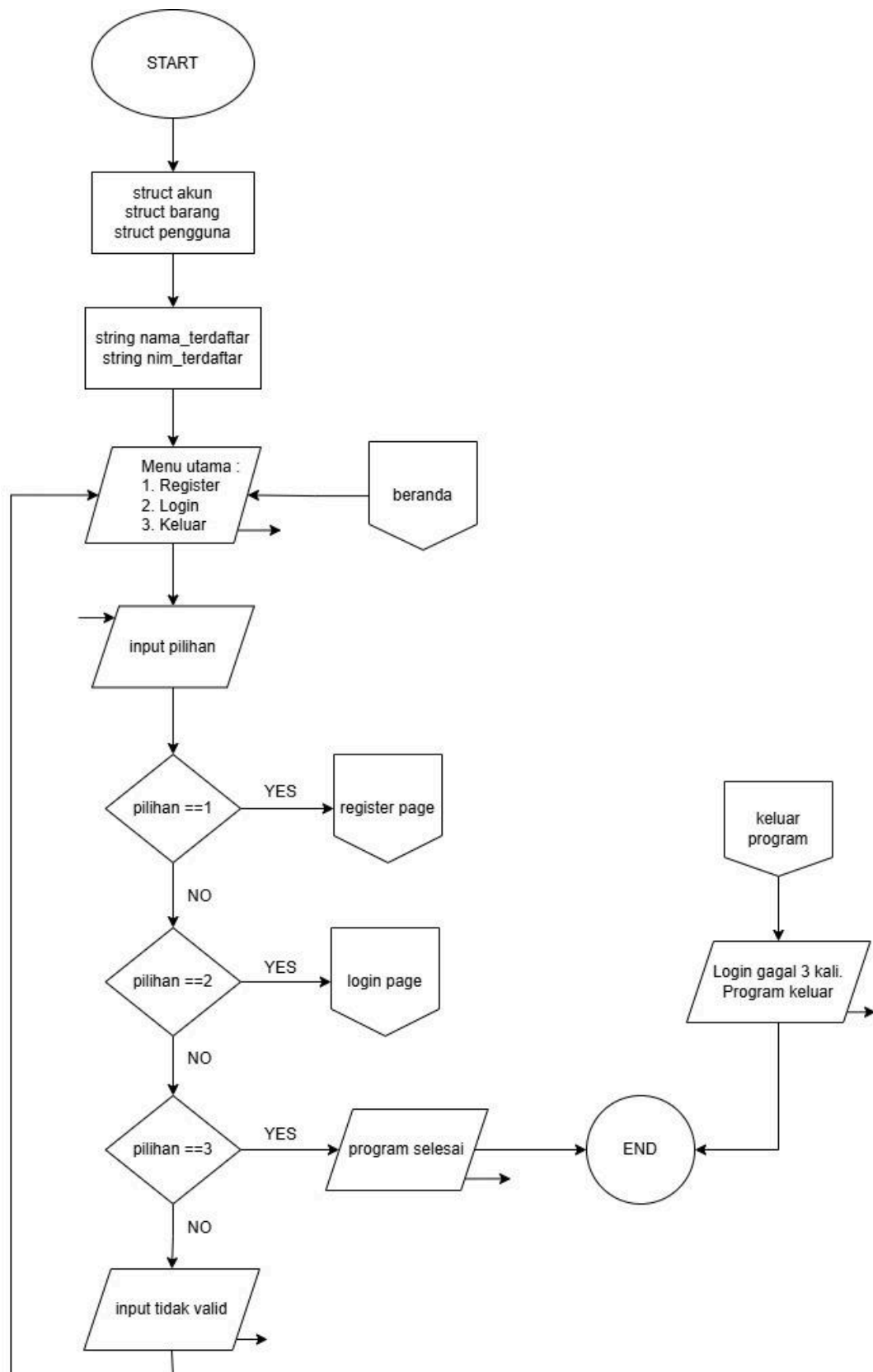


Disusun oleh:
WIDYA AYU ANGGRAINI (2409106011)
Kelas (A1 '24)

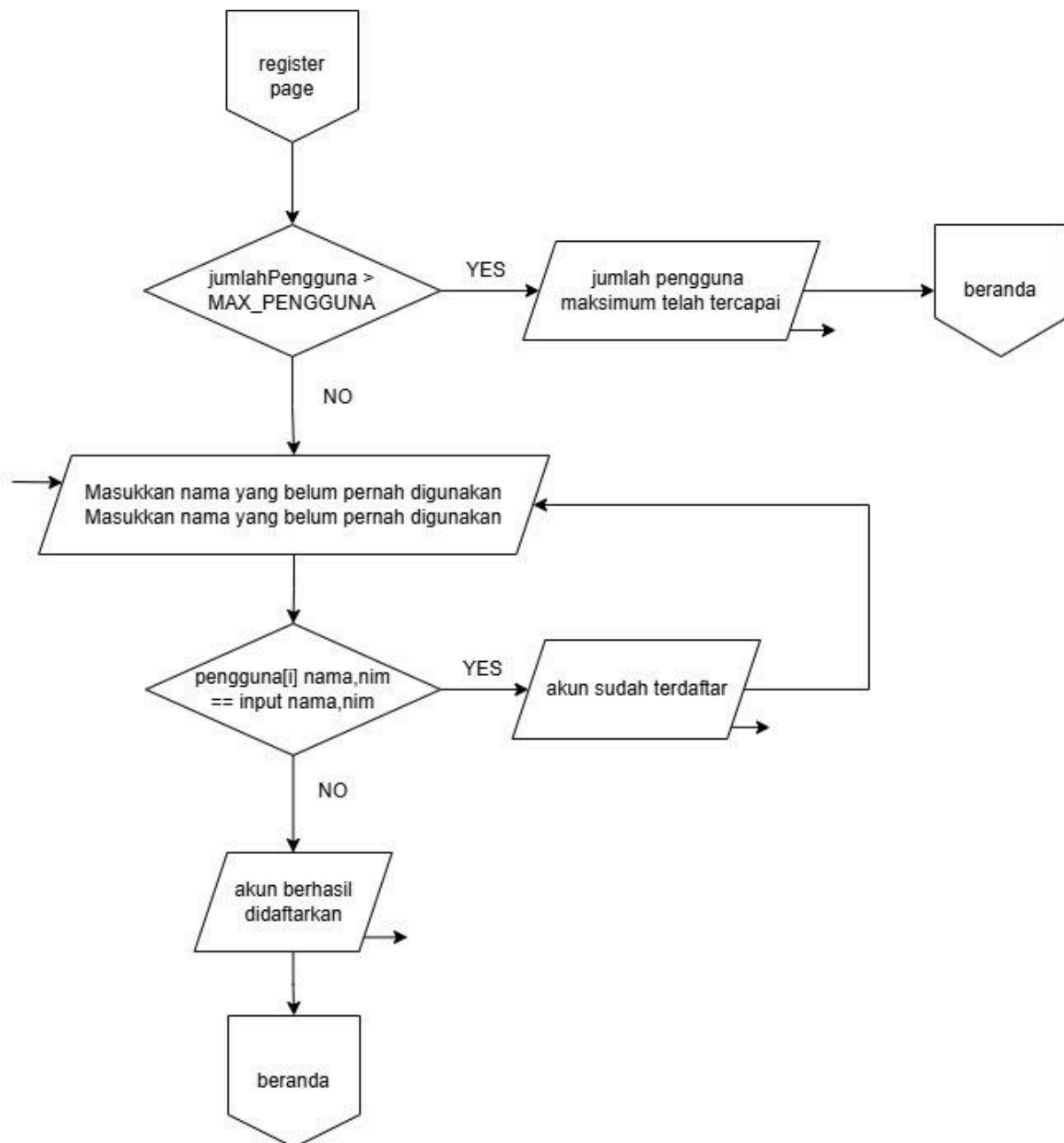
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

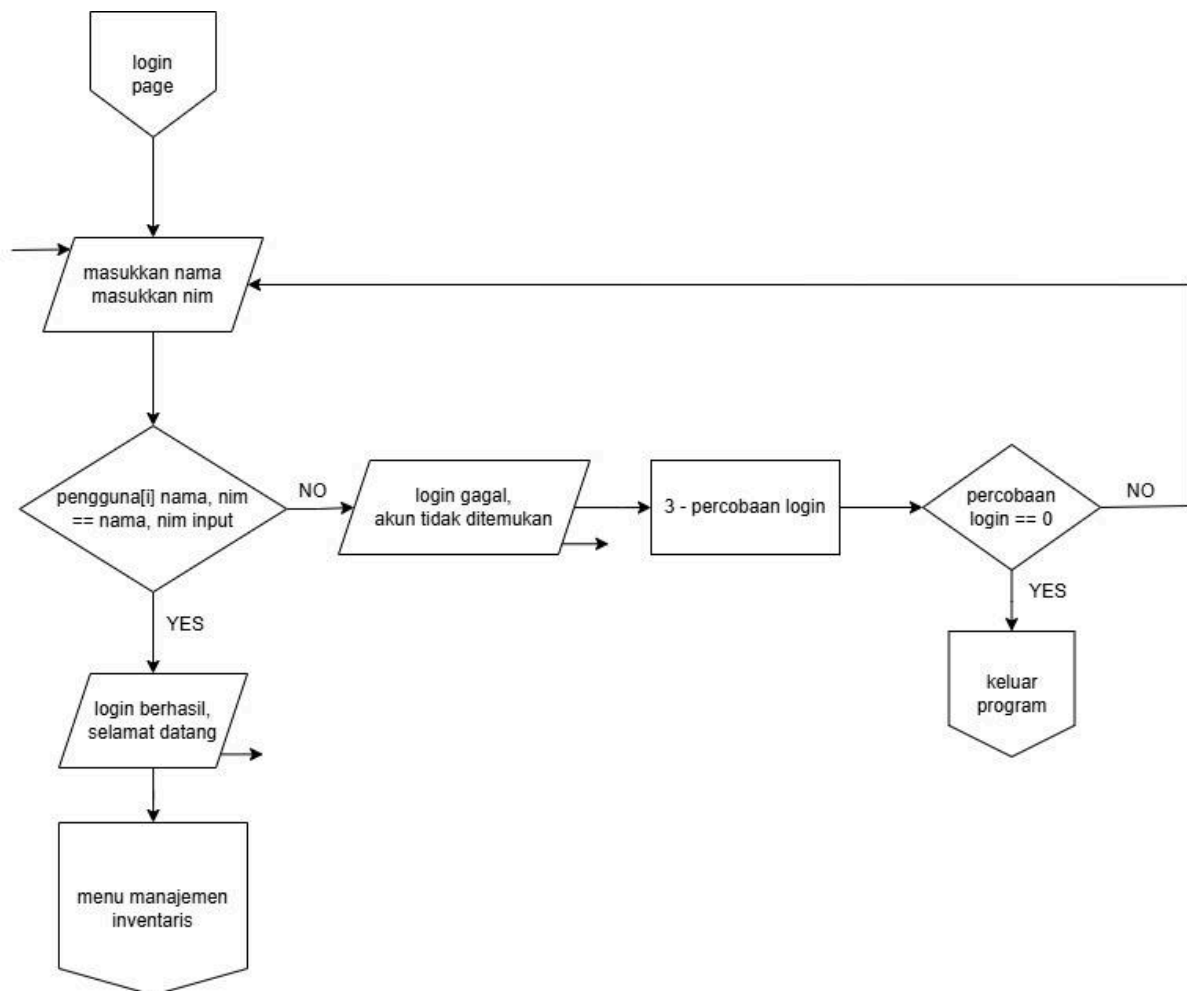
A. MENU UTAMA



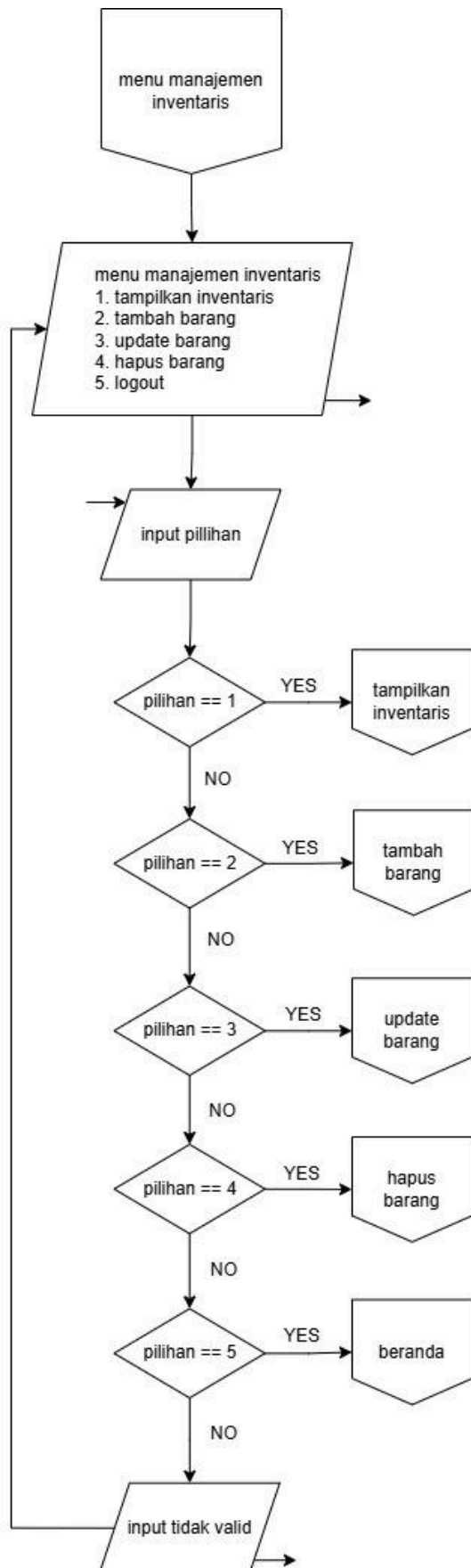
B. REGISTER



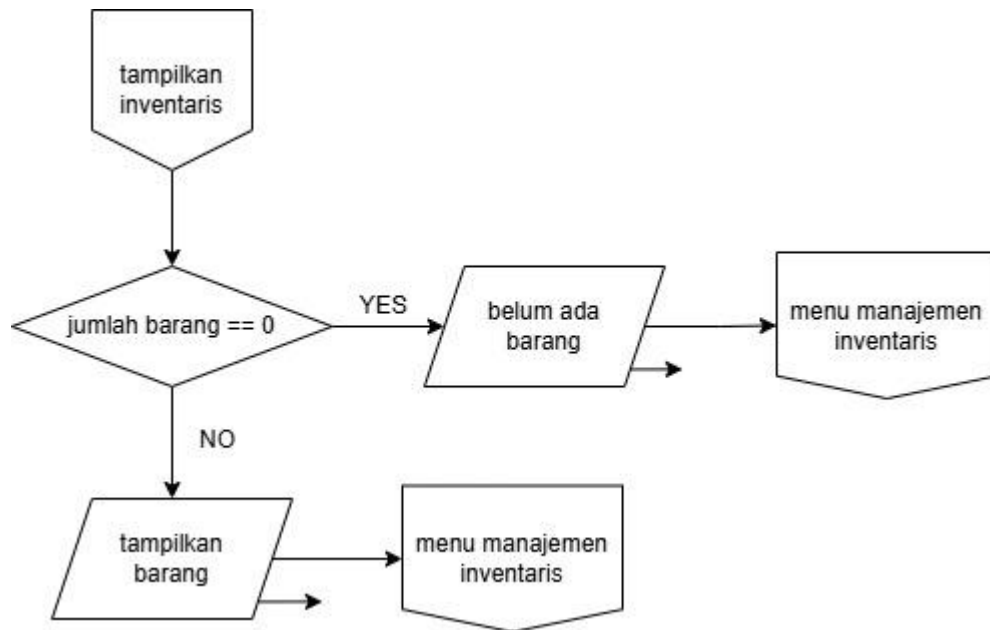
C. LOGIN



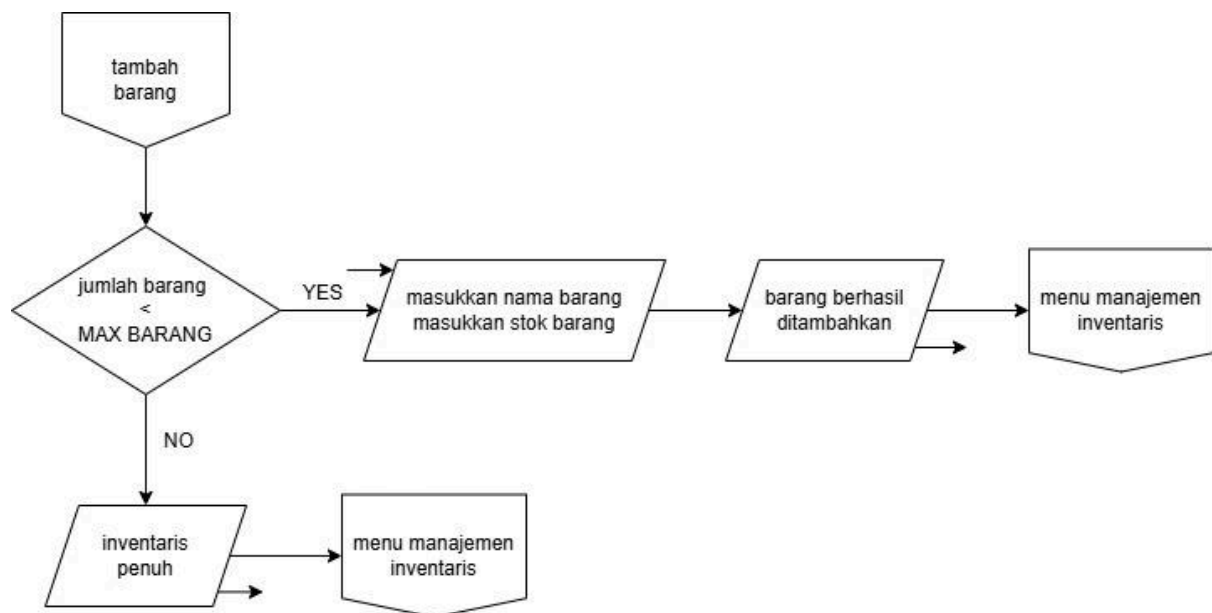
D. MENU MANAJEMEN INVENTARIS



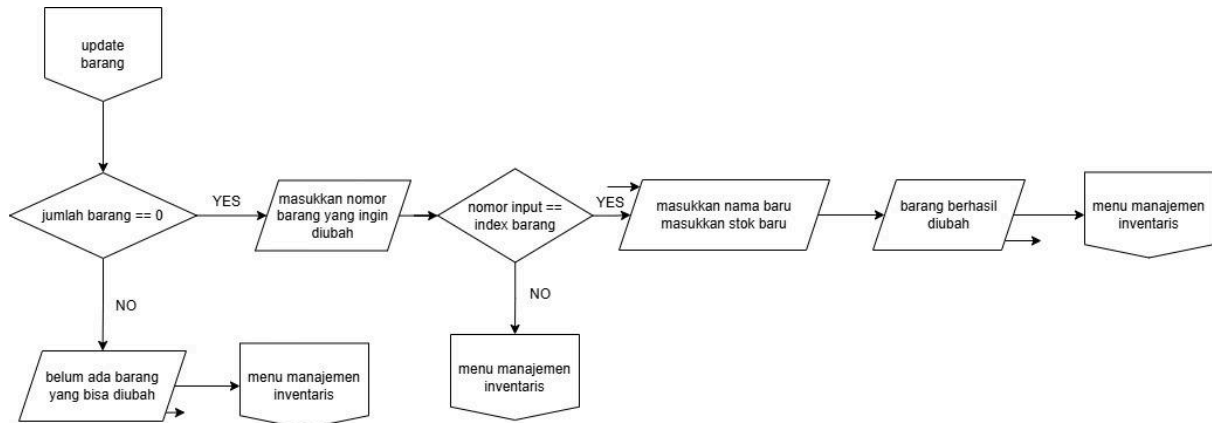
E. TAMPILKAN INVENTARIS



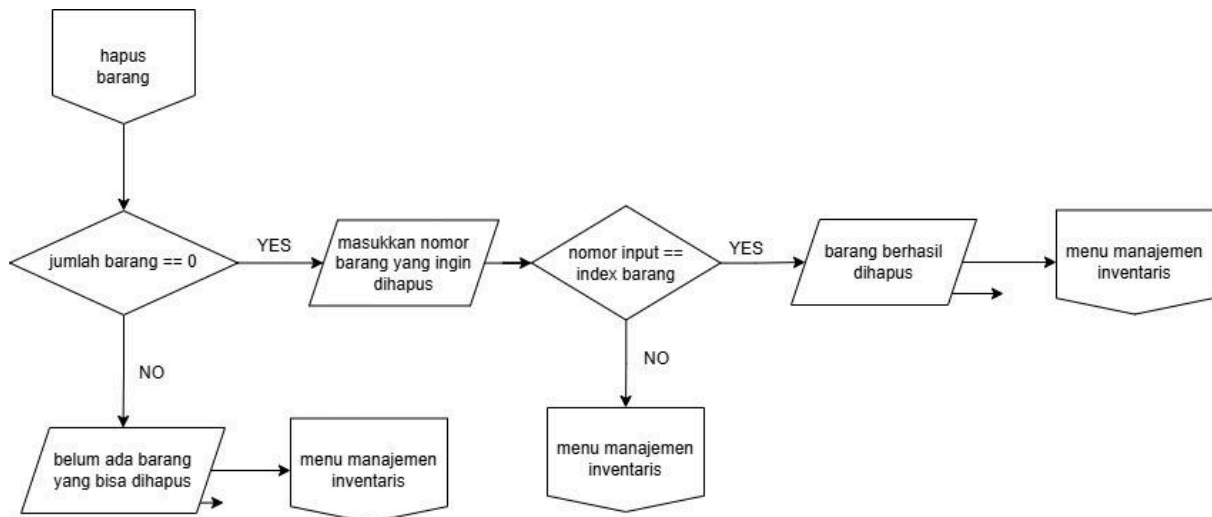
F. TAMBAH BARANG



G. UPDATE BARANG



H. HAPUS BARANG



2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini mensimulasikan sistem manajemen inventaris untuk pengguna. Pengguna dapat mendaftar dengan nama dan NIM unik, masuk dengan kredensial tersebut, dan mengelola inventaris barang mereka. Fitur-fitur yang tersedia meliputi:

- **Registrasi:** Pengguna baru dapat membuat akun dengan nama dan NIM unik.
- **Login:** Pengguna yang terdaftar dapat masuk dengan nama dan NIM mereka.
- **Manajemen Inventaris:** Setelah login, pengguna dapat:
 - Melihat daftar inventaris barang mereka.
 - Menambah barang baru ke inventaris.
 - Memperbarui informasi barang yang sudah ada.
 - Menghapus barang dari inventaris.
 - Logout dari akun yang sedang digunakan.
- **Validasi Input:** Program menangani input yang tidak valid dan memberikan pesan kesalahan yang sesuai.
- **Batas Pengguna dan Barang:** Program memiliki batasan jumlah pengguna dan barang yang dapat disimpan.

Fitur Utama:

- Penggunaan struktur data untuk menyimpan informasi pengguna dan barang.
- Implementasi fungsi untuk menampilkan, menambah, memperbarui, dan menghapus barang.
- Penanganan kesalahan input pengguna.
- Sistem login dengan percobaan login terbatas.
- Menu utama untuk registrasi, login, dan keluar program.

Cara Kerja:

1. **Registrasi:** Pengguna memasukkan nama dan NIM. Program memeriksa apakah kombinasi tersebut sudah ada. Jika tidak, akun baru dibuat.

2. **Login:** Pengguna memasukkan nama dan NIM. Program mencari akun yang cocok. Jika ditemukan, pengguna masuk. Jika tidak, pengguna diberi kesempatan untuk mencoba lagi hingga tiga kali.
3. **Manajemen Inventaris:** Setelah login, pengguna dapat memilih dari menu untuk melihat, menambah, memperbarui, atau menghapus barang.
4. **Validasi:** Program memvalidasi input pengguna untuk memastikan data yang benar dimasukkan.

3. Source Code

1. HEADER DAN STRUKTUR DATA

Bagian ini berisi include, define, dan definisi struct yang digunakan secara global.

SOURCE CODENYA :

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

#define MAX_PENGGUNA 100
#define MAX_BARANG 100

using namespace std;

struct Akun {
    string nama;
    string nim;
};

struct Barang {
    string nama;
    int stok;
};

struct Pengguna {
    Akun akun;
    Barang inventaris[MAX_BARANG];
    int jumlahBarang;
};
```

2. MAIN

Bagian ini berfungsi untuk proses masuk program
SOURCE CODENYA :

```
int main() {
    Pengguna pengguna[MAX_PENGGUNA];
    int jumlahPengguna = 0;
    int pilihan_menu, pilihan_login;
    string nama_input, nim_input;

    // Tambahkan akun default
    pengguna[jumlahPengguna].akun.nama = "WidyaAyuAnggraini";
    pengguna[jumlahPengguna].akun.nim = "2409106011";
    pengguna[jumlahPengguna].jumlahBarang = 0;
    jumlahPengguna++;

    do {
        cout << "Menu Utama" << endl;
        cout << "1. Register" << endl;
        cout << "2. Login" << endl;
        cout << "3. Keluar" << endl;
        cout << "Pilihan: ";

        if (!(cin >> pilihan_menu)) {
            cout << "Input tidak valid. Masukkan angka." << endl;
            cin.clear();
            while (cin.get() != '\n');
            continue;
        }
        while (cin.get() != '\n');
```

3. REGISTER

Bagian ini berfungsi untuk proses registrasi
SOURCE CODENYA :

```
case 1: { // Register
    if (jumlahPengguna < MAX_PENGGUNA) {
        bool akun_sudah_ada = false;
        do {
            cout << "Masukkan Nama Yang Belum Pernah Digunakan:
";

            getline(cin, nama_input);
            cout << "Masukkan NIM Yang Belum Pernah Digunakan:
";

            getline(cin, nim_input);
```

```

        akun_sudah_ada = false;
        for (int i = 0; i < jumlahPengguna; i++) {
            if (pengguna[i].akun.nama == nama_input &&
                pengguna[i].akun.nim == nim_input) {
                akun_sudah_ada = true;
                cout << "Akun sudah ada. Masukkan nama dan
                NIM yang berbeda." << endl;
                break;
            }
        }
    } while (akun_sudah_ada);

    pengguna[jumlahPengguna].akun.nama = nama_input;
    pengguna[jumlahPengguna].akun.nim = nim_input;
    pengguna[jumlahPengguna].jumlahBarang = 0;
    jumlahPengguna++;
    cout << "Akun berhasil didaftarkan." << endl;
} else {
    cout << "Jumlah pengguna maksimum tercapai." << endl;
}
break;
}

```

4. LOGIN

Bagian ini berfungsi untuk proses login

SOURCE CODENYA :

```

case 2: { // Login
    int percobaan_login = 0;
    bool login_berhasil = false;
    int index_pengguna = -1;

    while (percobaan_login < 3 && !login_berhasil) {
        cout << "Masukkan Nama: ";
        getline(cin, nama_input);
        cout << "Masukkan NIM: ";
        getline(cin, nim_input);

        for (int i = 0; i < jumlahPengguna; i++) {
            if (pengguna[i].akun.nama == nama_input &&
                pengguna[i].akun.nim == nim_input) {
                login_berhasil = true;
                index_pengguna = i;
                break;
            }
        }
    }
}

```

```

        }
    }

    if (!login_berhasil) {
        percobaan_login++;
        cout << "Login Gagal! Akun tidak ditemukan." <<
endl;

        if (percobaan_login < 3) {
            cout << "Coba lagi. Sisa percobaan: " << 3 -
percobaan_login << endl;
        } else {
            cout << "Login gagal 3 kali. Program keluar." <<
endl;

            return 0;
        }
    }
}

if (login_berhasil) {
    cout << "Login Berhasil! Selamat datang, " <<
pengguna[index_pengguna].akun.nama << "!" << endl;
}

```

5. MENU UTAMA

Bagian ini berfungsi untuk masuk ke menu utama

SOURCE CODENYA :

```

do {
    cout << "\nMenu Manajemen Inventaris" << endl;
    cout << "1. Tampilkan Inventaris" << endl;
    cout << "2. Tambah Barang" << endl;
    cout << "3. Update Barang" << endl;
    cout << "4. Hapus Barang" << endl;
    cout << "5. Logout" << endl;
    cout << "Pilihan: ";

    if (!(cin >> pilihan_login)) {
        cout << "Input tidak valid. Masukkan angka." <<
endl;

        cin.clear();
        while (cin.get() != '\n');
        continue;
    }
    while (cin.get() != '\n');
}

```

6. TAMPILKAN INVENTARIS

Bagian ini berfungsi untuk memanggil inventaris

SOURCE CODENYA :

```
case 1: { // Tampilkan Inventaris
    if (pengguna[index_pengguna].jumlahBarang ==
0) {
        cout << "Belum ada barang dalam
inventaris." << endl;
    } else {
        cout << "No. | Nama Barang          |
Stok" << endl;
        cout <<
"-----" << endl;
        for (int i = 0; i <
pengguna[index_pengguna].jumlahBarang; i++) {
            cout << i + 1 << ". | " <<
pengguna[index_pengguna].inventaris[i].nama;
            for (int j =
pengguna[index_pengguna].inventaris[i].nama.length(); j < 22; j++) {
                cout << " ";
            }
            cout << "| " <<
pengguna[index_pengguna].inventaris[i].stok << endl;
        }
    }
    break;
}
```

7. TAMBAH BARANG

Bagian ini berfungsi untuk menambah barang

SOURCE CODENYA :

```
case 2: { // Tambah Barang
    if (pengguna[index_pengguna].jumlahBarang <
MAX_BARANG) {
        cout << "Masukkan nama barang: ";
        getline(cin,
pengguna[index_pengguna].inventaris[pengguna[index_pengguna].jumlahBarang].n
ama);
        cout << "Masukkan stok barang: ";
        cin >>
pengguna[index_pengguna].inventaris[pengguna[index_pengguna].jumlahBarang].s
tok;
        while (cin.get() != '\n');
        pengguna[index_pengguna].jumlahBarang++;
    }
}
```

```

        cout << "Barang berhasil ditambahkan."
<< endl;
    } else {
        cout << "Inventaris penuh." << endl;
    }
    break;
}

```

8. UPDATE BARANG

Bagian ini berfungsi untuk mengubah barang

SOURCE CODENYA :

```

        case 3: { // Update Barang
            if (pengguna[index_pengguna].jumlahBarang ==
0) {
                cout << "Belum ada barang untuk diubah."
<< endl;
            } else {
                cout << "No. | Nama Barang          |
Stok" << endl;
                cout <<
"-----" << endl;
                for (int i = 0; i <
pengguna[index_pengguna].jumlahBarang; i++) {
                    cout << i + 1 << ". | " <<
pengguna[index_pengguna].inventaris[i].nama;
                    for (int j =
pengguna[index_pengguna].inventaris[i].nama.length(); j < 22; j++) {
                        cout << " ";
                    }
                    cout << "| " <<
pengguna[index_pengguna].inventaris[i].stok << endl;
                }
                int nomor_barang;
                cout << "\nMasukkan nomor barang yang
ingin diubah: ";
                if (!(cin >> nomor_barang) ||
nomor_barang < 1 || nomor_barang > pengguna[index_pengguna].jumlahBarang) {
                    cout << "Nomor barang tidak valid."
<< endl;
                    cin.clear();
                    while (cin.get() != '\n');
                } else {
                    while (cin.get() != '\n');
                    cout << "Masukkan nama baru barang:
";

```

```

                                getline(cin,
pengguna[index_pengguna].inventaris[nomor_barang - 1].nama);
                                cout << "Masukkan stok baru barang:
";

                                cin >>
pengguna[index_pengguna].inventaris[nomor_barang - 1].stok;
                                while (cin.get() != '\n');
                                cout << "Barang berhasil diupdate."
<< endl;

                                }
                                }
                                break;
                                }

```

9. HAPUS BARANG

Bagian ini berfungsi untuk menghapus barang

SOURCE CODENYA :

```

                                case 4: { // Hapus Barang
                                        if (pengguna[index_pengguna].jumlahBarang ==
0) {

                                                cout << "Belum ada barang untuk
dihapus." << endl;

                                        } else {
                                                cout << "No. | Nama Barang          |
Stok" << endl;

                                                cout <<
"-----" << endl;
                                                for (int i = 0; i <
pengguna[index_pengguna].jumlahBarang; i++) {
                                                        cout << i + 1 << ". | " <<
pengguna[index_pengguna].inventaris[i].nama;
                                                        for (int j =
pengguna[index_pengguna].inventaris[i].nama.length(); j < 22; j++) {
                                                                cout << " ";
                                                        }
                                                        cout << "| " <<
pengguna[index_pengguna].inventaris[i].stok << endl;
                                                        }
                                                int nomor_barang;
                                                cout << "\nMasukkan nomor barang yang
ingin dihapus: ";

                                                if (!(cin >> nomor_barang) ||
nomor_barang < 1 || nomor_barang > pengguna[index_pengguna].jumlahBarang) {
                                                        cout << "Nomor barang tidak valid."

```

```

<< endl;

                                cin.clear();
                                while (cin.get() != '\n');
                                } else {
                                    for (int i = nomor_barang - 1; i <
pengguna[index_pengguna].jumlahBarang - 1; i++) {

pengguna[index_pengguna].inventaris[i] =
pengguna[index_pengguna].inventaris[i + 1];
                                }

pengguna[index_pengguna].jumlahBarang--;
                                cout << "Barang berhasil dihapus."
<< endl;

                                while (cin.get() != '\n');

                                }
                                }
                                break;
                                }

```

10. LOGOUT

Bagian ini berfungsi untuk keluar akun

SOURCE CODENYA :

```

                                case 5:
                                    cout << "Logout berhasil." << endl;
                                    break;
                                default:
                                    cout << "Pilihan tidak valid." << endl;
                                    break;

```

11. MENGHENTIKAN PROGRAM

Bagian ini berfungsi untuk berhenti

SOURCE CODENYA :

```

                                }
                                if (pilihan_menu != 3 && pilihan_menu != 2) {
                                    cout << "\nTekan Enter untuk melanjutkan..." << endl;
                                    cin.get();
                                    system("cls");
                                }
                                } while (pilihan_menu != 3);

                                return 0;

```



```
}
```

4. Output Program

4.1 Menu Utama

```
Menu Utama  
1. Register  
2. Login  
3. Keluar  
Pilihan:
```

Gambar 4.1 Menu Utama

4.2 Register

```
Menu Utama  
1. Register  
2. Login  
3. Keluar  
Pilihan: 1  
Masukkan Nama Yang Belum Pernah Digunakan: ayang  
Masukkan NIM Yang Belum Pernah Digunakan: 11  
Akun berhasil didaftarkan.  
  
Tekan Enter untuk melanjutkan...
```

Gambar 4.2 Register

4.3 Login

```
Menu Utama  
1. Register  
2. Login  
3. Keluar  
Pilihan: 2  
Masukkan Nama: ayang  
Masukkan NIM: 11  
Login Berhasil! Selamat datang, ayang!  
  
Tekan Enter untuk melanjutkan...
```

Gambar 4.3 Login

4.4 Tampilkan Inventaris

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 1
Belum ada barang

Tekan Enter untuk melanjutkan...
```

Gambar 4.4 Tampilkan Inventaris

4.5 Tambah Barang

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 2
Masukkan nama barang: batu
Masukkan stok barang: 12
Barang berhasil ditambahkan.

Tekan Enter untuk melanjutkan...
```

Gambar 4.5 Tambah Barang

4.6 Update Barang

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 3
No. | Nama Barang          | Stok
-----
1.  | batu                    | 12
Masukkan nomor barang yang akan diubah: 1
Masukkan nama barang baru: kayu
Masukkan stok barang baru: 10
Barang berhasil diubah.

Tekan Enter untuk melanjutkan...
```

Gambar 4.6 Update Barang

4.7 Hapus Barang

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 4
No. | Nama Barang          | Stok
-----
1.  | kayu                 | 10
Masukkan nomor barang yang akan dihapus: 1
Barang berhasil dihapus.

Tekan Enter untuk melanjutkan...
```

Gambar 4.7 Hapus Barang

4.8 Keluar Program

```
Menu Utama  
1. Register  
2. Login  
3. Keluar  
Pilihan: 3  
Program selesai.  
PS C:\Users\silly\Downloads\praktikum-apl\post-test\post-test-3>
```

Gambar 4.8 Keluar Program

5. GIT

5.1 Git Add (Menambah File yang Ingin Dicommit)

```
PS C:\Users\silly\Downloads\praktikum-apl> git add .
```

Gambar 5.1 Git Add

5.2 Git Commit (CheckPoint)

```
PS C:\Users\silly\Downloads\praktikum-apl> git commit -m "kelar"  
[main 98c7063] kelar  
1 file changed, 0 insertions(+), 0 deletions(-)  
delete mode 100644 post-test/post-test-3/tempCodeRunnerFile.cpp
```

Gambar 5.2 Git Commit

5.3 Git Push (Mengupdate Semua yang Ada di Repository Lokal)

```
PS C:\Users\silly\Downloads\praktikum-apl> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 352 bytes | 352.00 KiB/s, done.
Total 4 (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/WidyaRanyyy/praktikum-apl
  71127d9..98c7063  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\silly\Downloads\praktikum-apl> █
```

Gambar 5.3 Git Push