

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:

WIDYA AYU ANGGRAINI (2409106011)

Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA

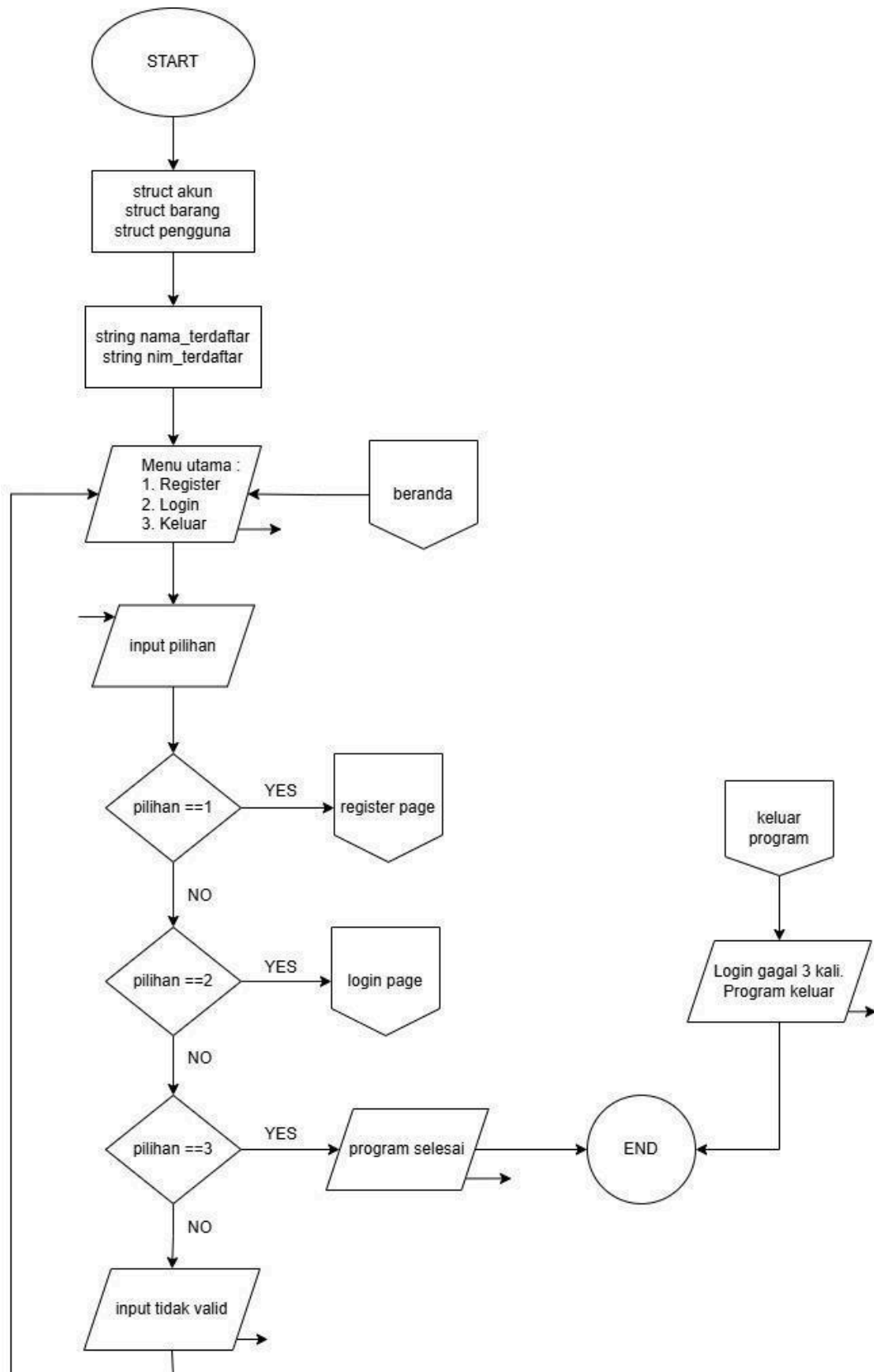
UNIVERSITAS MULAWARMAN

SAMARINDA

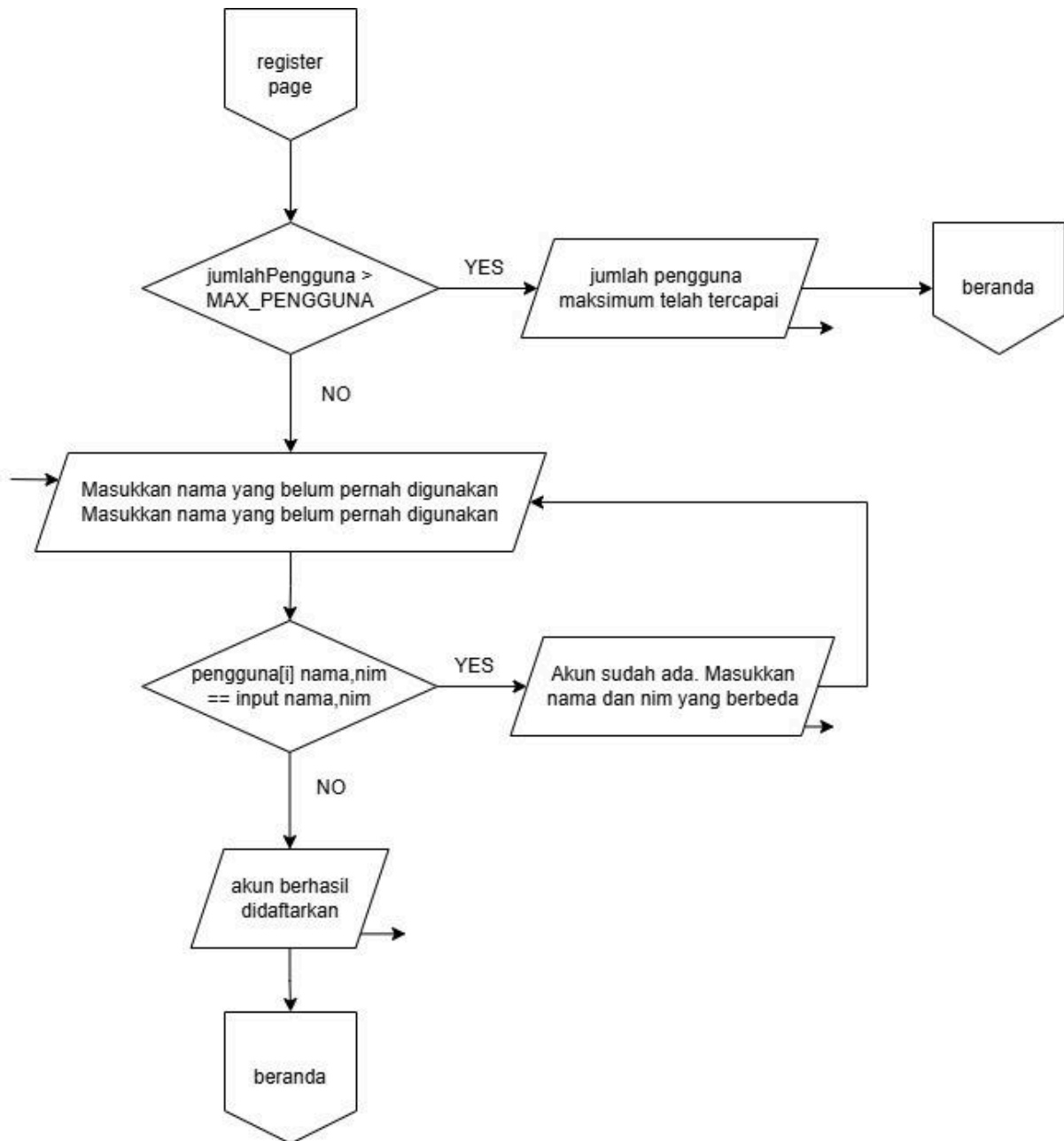
2025

1. Flowchart

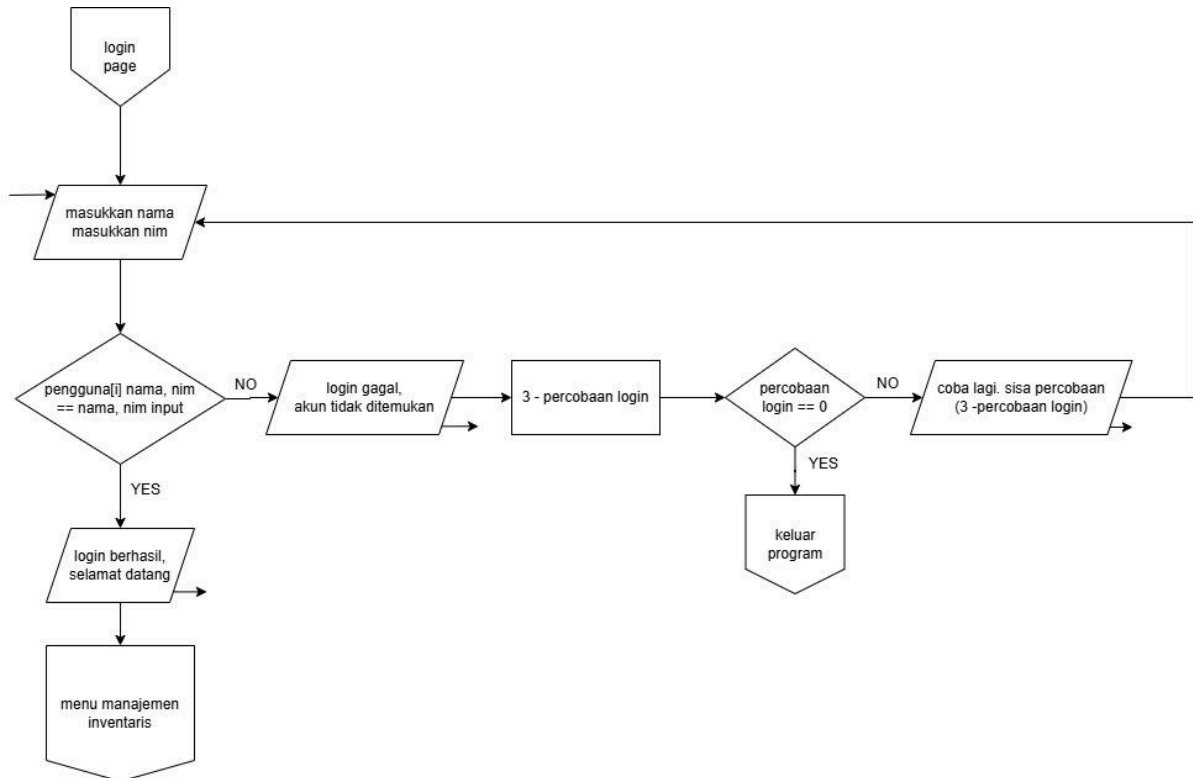
A. MENU UTAMA



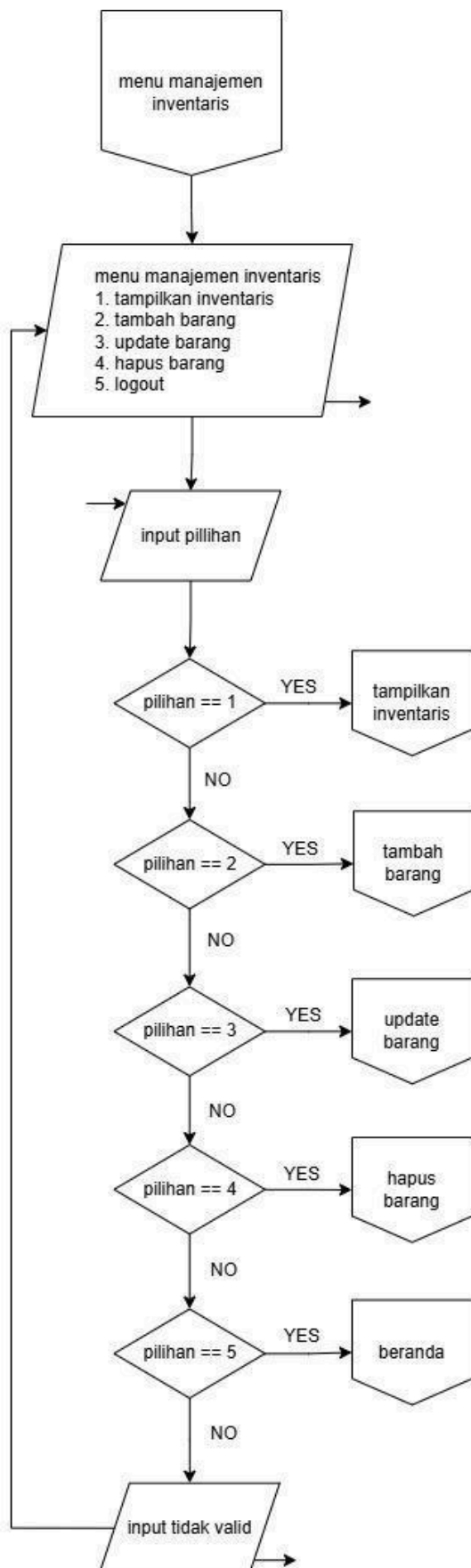
B. REGISTER



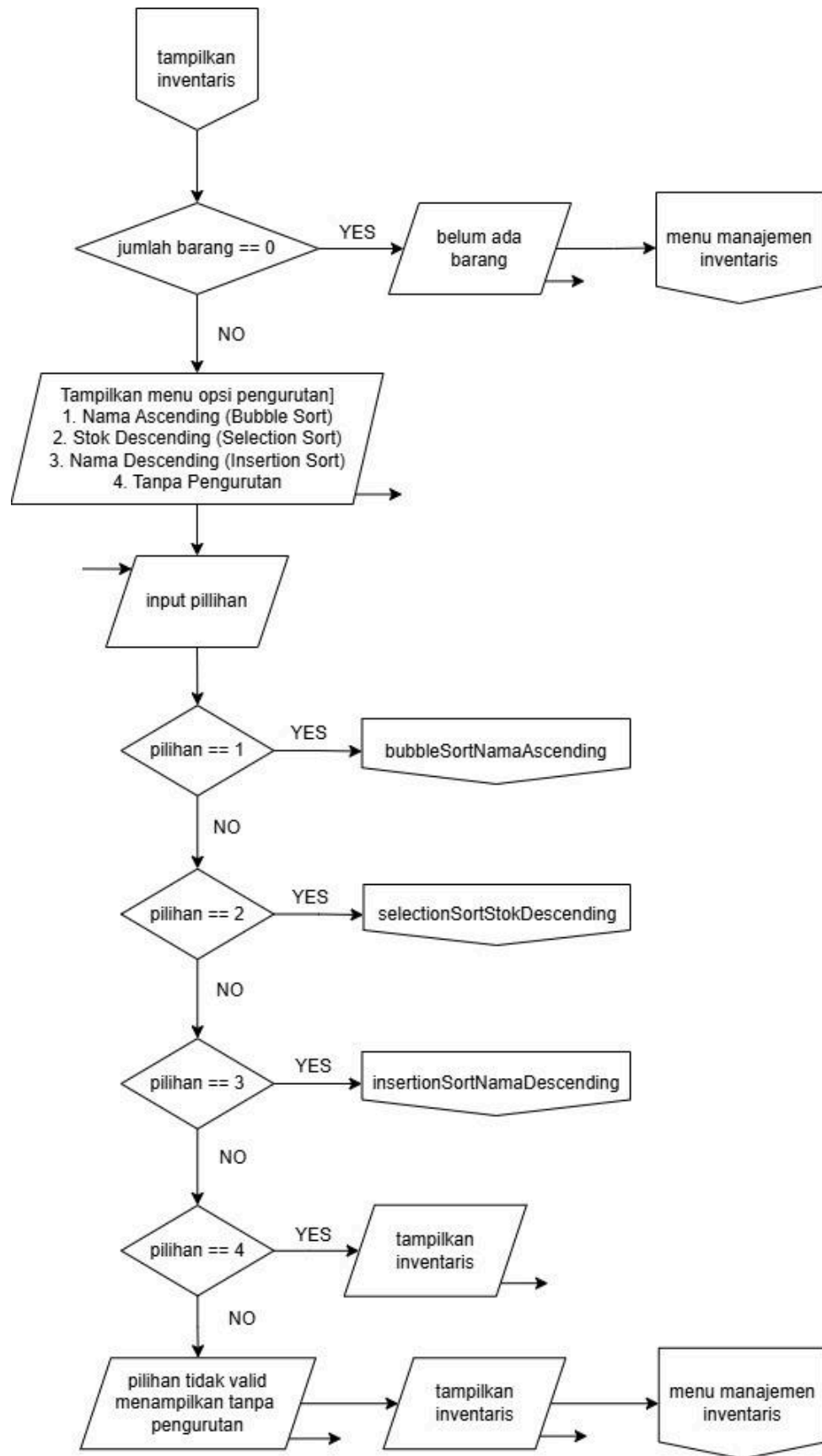
C. LOGIN



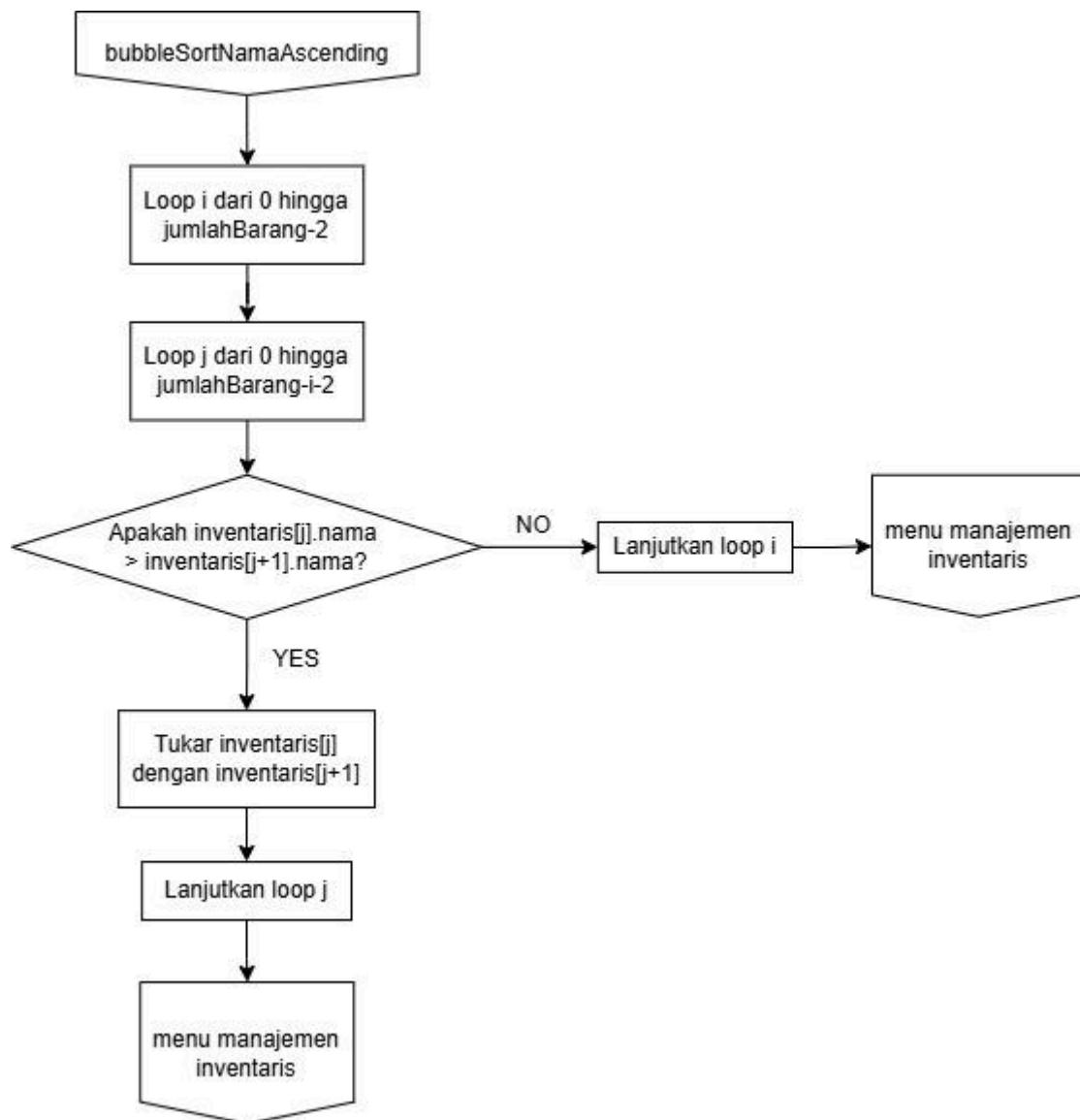
D. MENU MANAJEMEN INVENTARIS



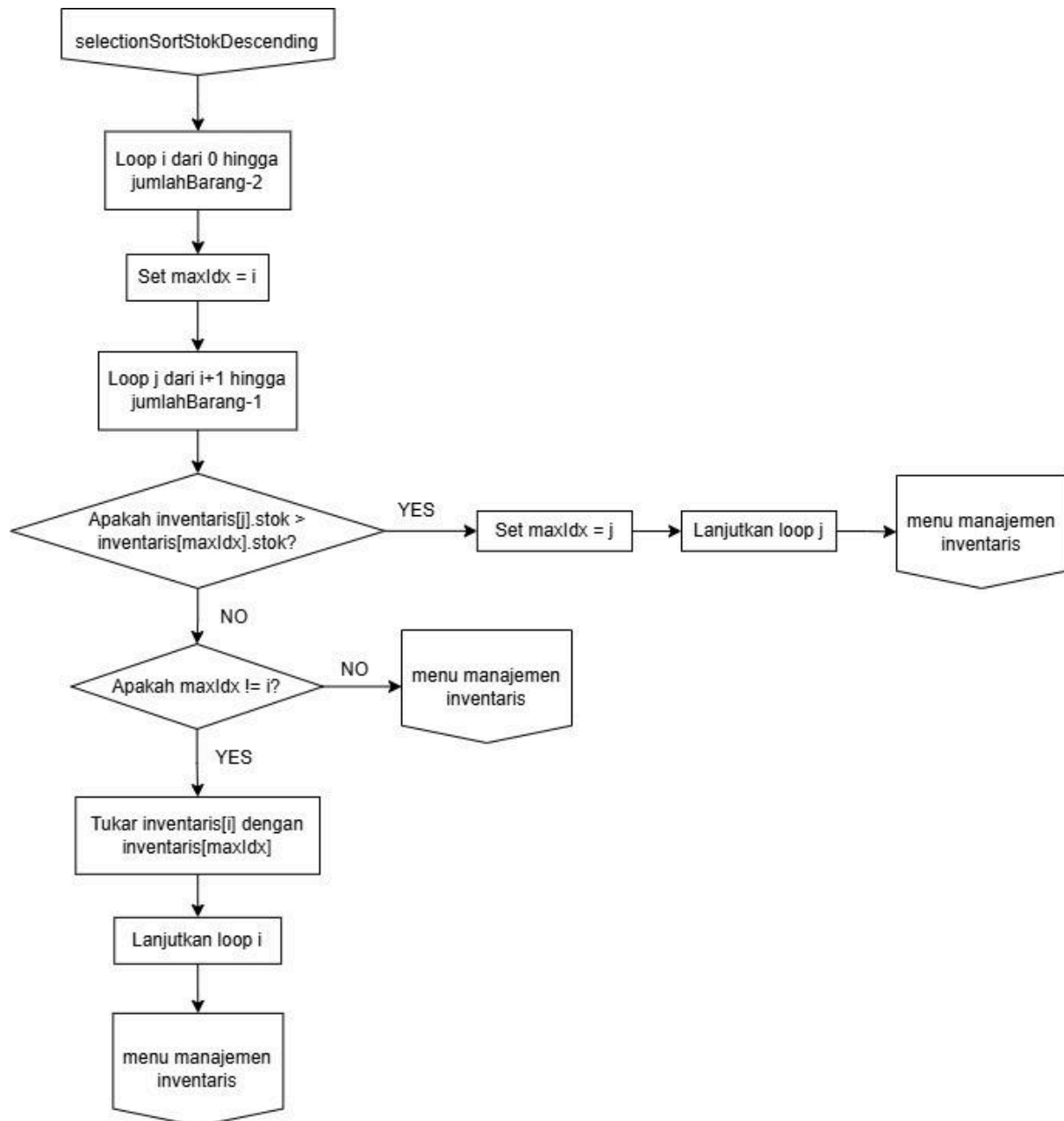
E. TAMPILKAN INVENTARIS



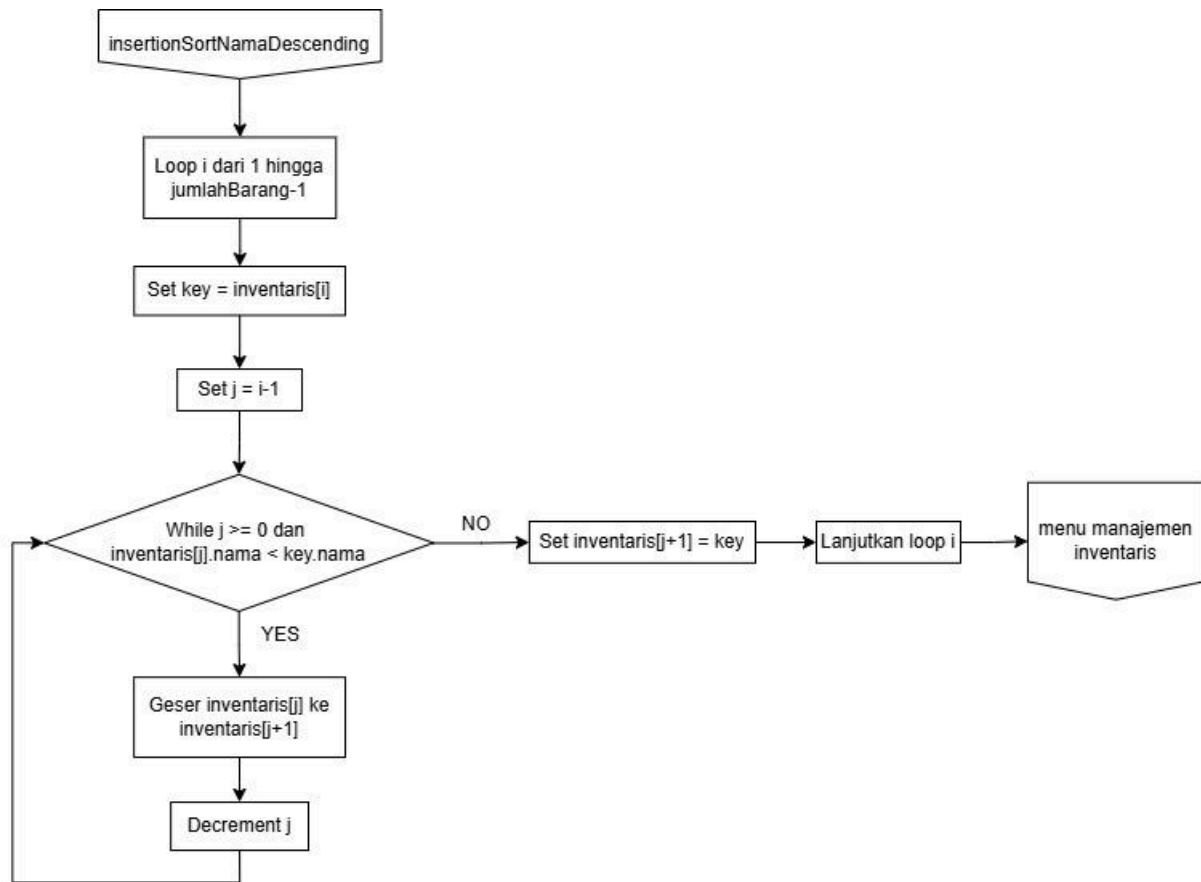
F. bubbleSortNamaAscending



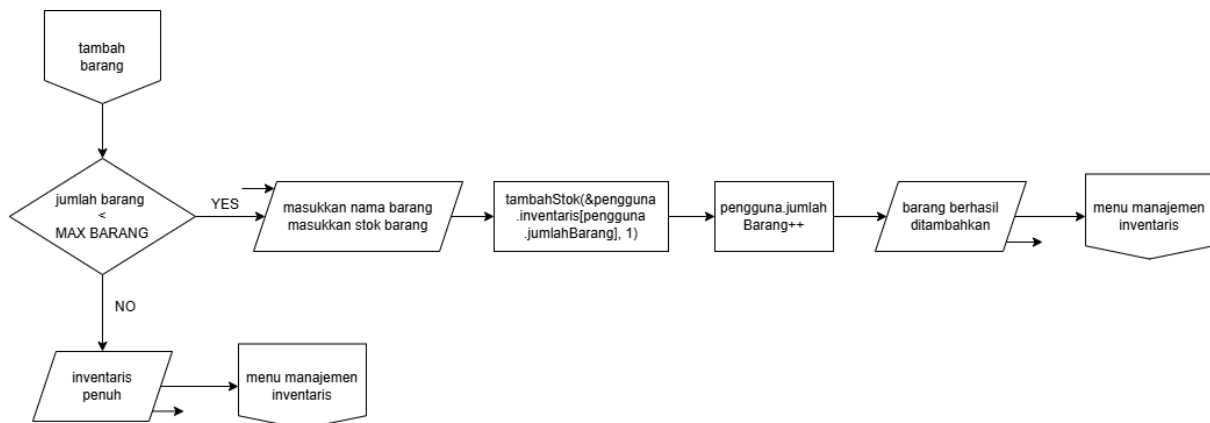
G. selectionSortStokDescending



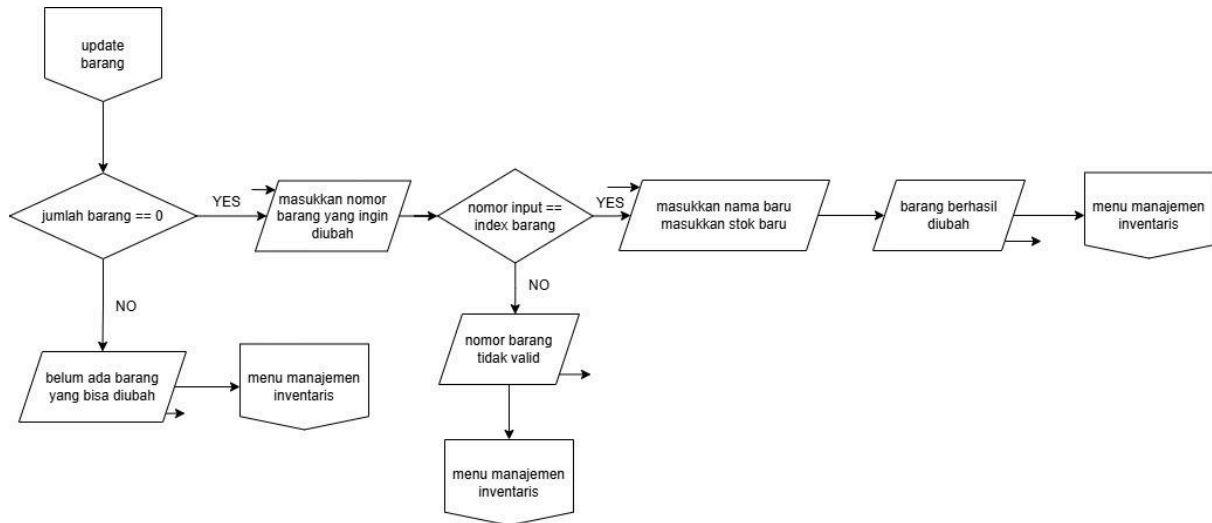
H. insertionSortNamaDescending



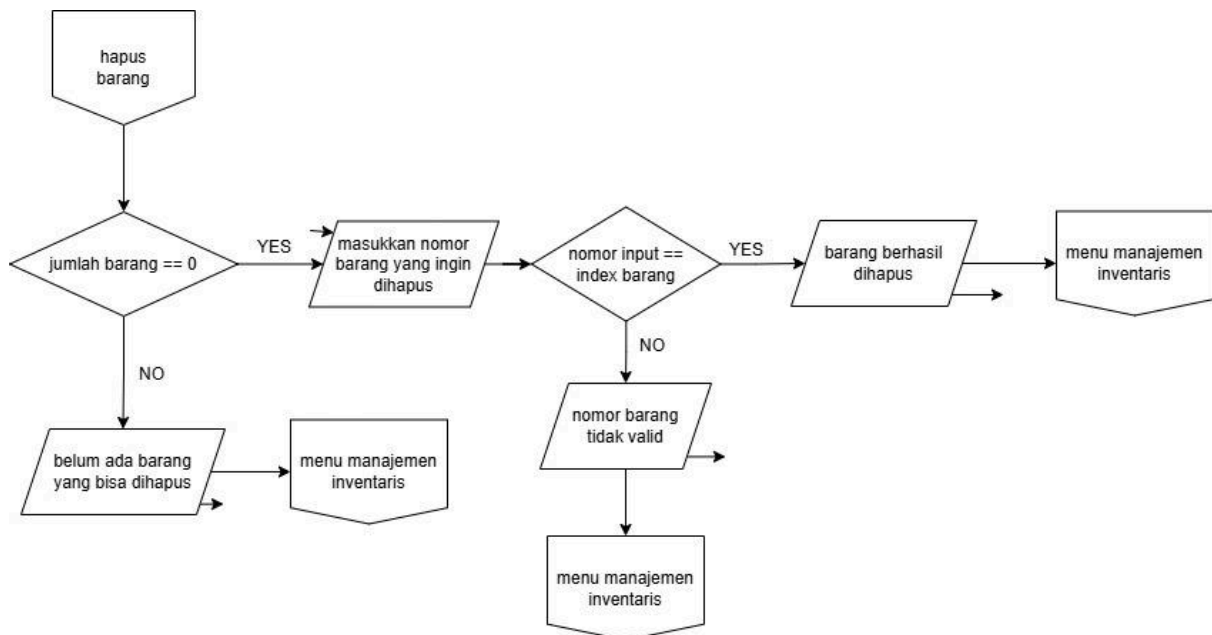
I. TAMBAH BARANG



J. UPDATE BARANG



K. HAPUS BARANG



2. Analisis Program

A. Deskripsi Singkat Program

- **Penambahan Fungsi Sorting:**
 - **Bubble Sort (Nama Ascending):** Mengurutkan nama barang dari A-Z. Algoritma ini membandingkan dua elemen berdekatan dan menukar jika urutannya salah. Kompleksitas waktu: $O(n^2)$.
 - **Selection Sort (Stok Descending):** Mengurutkan stok barang dari besar ke kecil. Algoritma ini mencari elemen dengan stok terbesar di bagian yang belum diurutkan dan menempatkannya di awal. Kompleksitas waktu: $O(n^2)$.
 - **Insertion Sort (Nama Descending):** Mengurutkan nama barang dari Z-A sebagai pilihan bebas. Algoritma ini membangun daftar terurut satu per satu dengan menyisipkan elemen ke posisi yang tepat. Kompleksitas waktu: $O(n^2)$.
- **Modifikasi Fungsi tampilInventaris:**
 - Ditambahkan menu untuk memilih metode pengurutan sebelum menampilkan inventaris.
 - Pengguna dapat memilih antara:
 - Urutkan nama barang ascending (Bubble Sort).
 - Urutkan stok barang descending (Selection Sort).
 - Urutkan nama barang descending (Insertion Sort).
 - Tampilkan tanpa pengurutan.
 - Setelah memilih, inventaris diurutkan sesuai pilihan dan ditampilkan dalam format tabel.

Cara Kerja Program

- **Menu Utama:** Pengguna dapat mendaftar, login, atau keluar.
- **Setelah Login:** Pengguna masuk ke menu manajemen inventaris dengan opsi untuk menampilkan inventaris, menambah barang, mengupdate barang, menghapus barang, atau logout.
- **Tampilkan Inventaris:**
 - Pengguna diminta memilih metode pengurutan.
 - Inventaris diurutkan sesuai pilihan dan ditampilkan dalam tabel dengan kolom nomor, nama barang, dan stok.

- Jika tidak ada barang, pesan "Belum ada barang dalam inventaris" ditampilkan.
- **Sorting:** Pengurutan hanya dilakukan untuk tampilan dan tidak mengubah urutan asli kecuali pengguna memperbarui inventaris.

3. Source Code

1. STRUKTUR UMUM DAN HEADER

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;

// Struktur untuk menyimpan data akun
struct Akun {
    string nama;
    string nim;
};

// Struktur untuk menyimpan data barang
struct Barang {
    string nama;
    int stok;
};

// Struktur untuk menyimpan data pengguna
struct Pengguna {
    Akun akun;
    Barang inventaris[MAX_BARANG];
};
```

```
int jumlahBarang;  
};
```

2. FUNGSI TAMPILKAN TOTAL BARANG

Bagian ini menggunakan referensi untuk menghindari penyalinan objek Pengguna yang besar. const memastikan data tidak diubah. Fungsi ini hanya menampilkan jumlahBarang
SOURCE CODENYA :

```
void tampilkanTotalBarang(const Pengguna &pengguna) {  
    cout << "Total barang yang dimiliki: " << pengguna.jumlahBarang << endl;  
}
```

3. FUNGSI TAMBAH STOK

Bagian ini menggunakan pointer digunakan untuk langsung memodifikasi atribut stok dari objek Barang yang ditunjuk. Parameter tambahan menentukan jumlah stok yang ditambahkan.

SOURCE CODENYA :

```
void tambahStok(Barang *barang, int tambahan) {  
    barang->stok += tambahan;  
}
```

4. FUNGSI CLEAR SCREEN

Bagian ini berfungsi untuk menghilangkan riwayat di terminal agar program tersusun rapi
SOURCE CODENYA :

```
void clearScreen() {  
    system("cls");  
}
```

5. FUNGSI bubbleSortNamaAscending

Bagian ini berfungsi untuk mengurutkan nama menggunakan metode bubble sort secara ascending

SOURCE CODENYA :

```
void bubbleSortNamaAscending(Pengguna &pengguna) {
    for (int i = 0; i < pengguna.jumlahBarang - 1; i++) {
        for (int j = 0; j < pengguna.jumlahBarang - i - 1; j++) {
            if (pengguna.inventaris[j].nama > pengguna.inventaris[j + 1].nama) {
                // Tukar elemen
                Barang temp = pengguna.inventaris[j];
                pengguna.inventaris[j] = pengguna.inventaris[j + 1];
                pengguna.inventaris[j + 1] = temp;
            }
        }
    }
}
```

6. FUNGSI selectionSortStokDescending

Bagian ini berfungsi untuk mengurutkan stok menggunakan metode selectionsort secara descending

SOURCE CODENYA :

```
void selectionSortStokDescending(Pengguna &pengguna) {
    for (int i = 0; i < pengguna.jumlahBarang - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < pengguna.jumlahBarang; j++) {
            if (pengguna.inventaris[j].stok >
                pengguna.inventaris[maxIdx].stok) {
                maxIdx = j;
            }
        }
        if (maxIdx != i) {
            // Tukar elemen
        }
    }
}
```

```

        Barang temp = pengguna.inventaris[i];
        pengguna.inventaris[i] = pengguna.inventaris[maxIdx];
        pengguna.inventaris[maxIdx] = temp;
    }
}
}

```

7. FUNGSI insertionSortNamaDescending

Bagian ini berfungsi untuk mengurutkan nama menggunakan metode insertionsort secara descending

SOURCE CODENYA :

```

void insertionSortNamaDescending(Pengguna &pengguna) {
    for (int i = 1; i < pengguna.jumlahBarang; i++) {
        Barang key = pengguna.inventaris[i];
        int j = i - 1;
        while (j >= 0 && pengguna.inventaris[j].nama < key.nama) {
            pengguna.inventaris[j + 1] = pengguna.inventaris[j];
            j--;
        }
        pengguna.inventaris[j + 1] = key;
    }
}

```

8. FUNGSI REGISTER

Bagian ini berfungsi untuk melakukan registrasi

SOURCE CODENYA :

```

void registrasi(Pengguna pengguna[], int &jumlahPengguna) {
    if (jumlahPengguna < MAX_PENGGUNA) {
        bool akun_sudah_ada = false;
        string nama_input, nim_input;
        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;
    }
    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();
}

```

9. FUNGSI LOGIN

Bagian ini berfungsi untuk proses login

SOURCE CODENYA :

```

int login(Pengguna pengguna[], int jumlahPengguna) {

```



```

int percobaan_login = 0;
bool login_berhasil = false;
int index_pengguna = -1;
string nama_input, nim_input;

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 -2;
        }
    }
}

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

```

```

    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();
    return index_pengguna;
}

```

10. FUNGSI TAMPILKAN INVENTARIS

Bagian ini berfungsi untuk menampilkan inventaris sesuai dengan opsi sorting yang diinginkan

SOURCE CODENYA :

```

void tampilInventaris(Pengguna &pengguna) {
    if (pengguna.jumlahBarang == 0) {
        cout << "Belum ada barang dalam inventaris." << endl;
    } else {
        int sortChoice;
        cout << "Pilih metode pengurutan:" << endl;
        cout << "1. Nama Barang (Ascending - Bubble Sort)" << endl;
        cout << "2. Stok Barang (Descending - Selection Sort)" << endl;
        cout << "3. Nama Barang (Descending - Insertion Sort)" << endl;
        cout << "4. Tanpa Pengurutan" << endl;
        cout << "Pilihan: ";
        cin >> sortChoice;
        while (cin.get() != '\n');

        switch (sortChoice) {
            case 1:
                bubbleSortNamaAscending(pengguna);
                break;
            case 2:
                selectionSortStokDescending(pengguna);
                break;
            case 3:
                insertionSortNamaDescending(pengguna);
                break;
        }
    }
}

```

```

        case 4:
            // Tidak ada pengurutan
            break;
        default:
            cout << "Pilihan tidak valid. Menampilkan tanpa pengurutan."
<< endl;
    }

    cout << "No. | Nama Barang          | Stok" << endl;
    cout << "-----" << endl;
    for (int i = 0; i < pengguna.jumlahBarang; i++) {
        cout << i + 1 << ". | " << pengguna.inventaris[i].nama;
        for (int j = pengguna.inventaris[i].nama.length(); j < 22; j++)
        {
            cout << " ";
        }
        cout << "| " << pengguna.inventaris[i].stok << endl;
    }
}
cout << "\nTekan Enter untuk melanjutkan..." << endl;
cin.get();
clearScreen();
}

```

11. FUNGSI TAMBAH BARANG

Bagian ini berfungsi untuk menambah barang

SOURCE CODENYA :

```

void tambahBarang(Pengguna &pengguna) {
    if (pengguna.jumlahBarang < MAX_BARANG) {
        cout << "Masukkan nama barang: ";
        getline(cin, pengguna.inventaris[pengguna.jumlahBarang].nama);
        cout << "Masukkan stok barang: ";
        cin >> pengguna.inventaris[pengguna.jumlahBarang].stok;
        while (cin.get() != '\n');
    }
}

```

```

        // Tambahan: pakai pointer untuk menambah stok default 1
        tambahStok(&pengguna.inventaris[pengguna.jumlahBarang], 1); //
contoh penggunaan dereference
        pengguna.jumlahBarang++;

        cout << "Barang berhasil ditambahkan." << endl;

        // Tambahan: tampilkan total barang
        tampilkanTotalBarang(pengguna); // contoh penggunaan reference
    } else {
        cout << "Inventaris penuh." << endl;
    }
    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();
}

```

12. FUNGSI UPDATE BARANG

Bagian ini berfungsi untuk mengubah barang

SOURCE CODENYA :

```

void updateBarang(Pengguna &pengguna) {
    if (pengguna.jumlahBarang == 0) {
        cout << "Belum ada barang untuk diubah." << endl;
    } else {
        tampilInventaris(pengguna);
        int nomor_barang;
        cout << "\nMasukkan nomor barang yang ingin diubah: ";
        if (!(cin >> nomor_barang) || nomor_barang < 1 || nomor_barang >
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.inventaris[nomor_barang - 1].nama);
        cout << "Masukkan stok baru barang: ";
        cin >> pengguna.inventaris[nomor_barang - 1].stok;
        while (cin.get() != '\n');
        cout << "Barang berhasil diupdate." << endl;
    }
}
cout << "\nTekan Enter untuk melanjutkan..." << endl;
cin.get();
clearScreen();
}

```

13. FUNGSI HAPUS BARANG

Bagian ini berfungsi untuk menghapus barang

SOURCE CODENYA :

```

void hapusBarang(Pengguna &pengguna) {
    if (pengguna.jumlahBarang == 0) {
        cout << "Belum ada barang untuk dihapus." << endl;
    } else {
        tampilInventaris(pengguna);
        int nomor_barang;
        cout << "\nMasukkan nomor barang yang ingin dihapus: ";
        if (!(cin >> nomor_barang) || nomor_barang < 1 || nomor_barang >
pengguna.jumlahBarang) {
            cout << "Nomor barang tidak valid." << endl;
            cin.clear();
            while (cin.get() != '\n');
        } else {
            for (int i = nomor_barang - 1; i < pengguna.jumlahBarang - 1;
i++) {
                pengguna.inventaris[i] = pengguna.inventaris[i + 1];
            }
        }
    }
}

```

```

    }
    pengguna.jumlahBarang--;
    cout << "Barang berhasil dihapus." << endl;
    while (cin.get() != '\n');
}
}
cout << "\nTekan Enter untuk melanjutkan..." << endl;
cin.get();
clearScreen();
}

```

14. FUNGSI LOGOUT

Bagian ini berfungsi untuk logout dari program

SOURCE CODENYA :

```

void logout() {
    cout << "Logout berhasil." << endl;
    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();
}

```

15. FUNGSI KELUAR PROGRAM

Bagian ini berfungsi untuk keluar program

SOURCE CODENYA :

```

void keluarProgram() {
    cout << "Terima kasih telah menggunakan program ini." << endl;
    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();
}

```

16. FUNGSI MAIN

Bagian ini berfungsi untuk menjalankan program utama

SOURCE CODENYA :

```
int main() {
    Pengguna pengguna[MAX_PENGGUNA];
    int jumlahPengguna = 0;
    int pilihan_menu;

    // 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');
            cout << "\nTekan Enter untuk melanjutkan..." << endl;
            cin.get();
            clearScreen();
            continue;
        }
        while (cin.get() != '\n');

        switch (pilihan_menu) {
            case 1:
                registrasi(pengguna, jumlahPengguna);
```

```

        break;
    case 2: {
        int index_pengguna = login(pengguna, jumlahPengguna);
        if (index_pengguna == -2) {
            return 0;
        }
        if (index_pengguna != -1) {
            int pilihan_login;
            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');
                    cout << "\nTekan Enter untuk melanjutkan..." <<
endl;

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

                switch (pilihan_login) {
                    case 1:
tampilInventaris(pengguna[index_pengguna]); break;
                    case 2: tambahBarang(pengguna[index_pengguna]);
break;
                    case 3: updateBarang(pengguna[index_pengguna]);
break;

```



```

        case 4: hapusBarang(pengguna[index_pengguna]);
break;

        case 5: logout(); break;
        default:
            cout << "Pilihan tidak valid." << endl;
            cout << "\nTekan Enter untuk melanjutkan..."
<< endl;

            cin.get();
            clearScreen();

        }
    } while (pilihan_login != 5);
}
break;
}
case 3:
    keluarProgram();
    return 0;
default:
    cout << "Pilihan tidak valid." << endl;
    cout << "\nTekan Enter untuk melanjutkan..." << endl;
    cin.get();
    clearScreen();

}
} while (pilihan_menu != 3);

return 0;
}

```

4. Output

Hasil Output

```
PS C:\Users\silly\Downloads\Praktikum-apl> cd "c:\Users\silly\Downloads\Praktikum-apl\post-test\post-test-6\" ; if ($?) { g++ 2409106011-WidyaAyuAnggraini-PT-6.cpp -o 2409106011-WidyaAyuAnggraini-PT-6 } ; if ($?) { .\2409106011-WidyaAyuAnggraini-PT-6 }  
Menu Utama  
1. Register  
2. Login  
3. Keluar  
Pilihan: 1  
Masukkan Nama Yang Belum Pernah Digunakan: widya  
Masukkan NIM Yang Belum Pernah Digunakan: 11  
Akun berhasil didaftarkan.  
  
Tekan Enter untuk melanjutkan...
```

Gambar 1 Registrasi

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

Gambar 2 Login

```
Menu Manajemen Inventaris  
1. Tampilkan Inventaris  
2. Tambah Barang  
3. Update Barang  
4. Hapus Barang  
5. Logout  
Pilihan: 2  
Masukkan nama barang: semen  
Masukkan stok barang: 12  
Barang berhasil ditambahkan.  
Total barang yang dimiliki: 1  
  
Tekan Enter untuk melanjutkan...
```

Gambar 3.1 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: 63
Barang berhasil ditambahkan.
Total barang yang dimiliki: 2

Tekan Enter untuk melanjutkan...

```

Gambar 3.2 Tambah Barang

```

Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 1
Pilih metode pengurutan:
1. Nama Barang (Ascending - Bubble Sort)
2. Stok Barang (Descending - Selection Sort)
3. Nama Barang (Descending - Insertion Sort)
4. Tanpa Pengurutan
Pilihan: 1

```

No.	Nama Barang	Stok
1.	batu	64
2.	semen	13

```

Tekan Enter untuk melanjutkan...

```

Gambar 4.1 Tampilkan Inventaris sorting nama ascending bubblesort

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 1
Pilih metode pengurutan:
1. Nama Barang (Ascending - Bubble Sort)
2. Stok Barang (Descending - Selection Sort)
3. Nama Barang (Descending - Insertion Sort)
4. Tanpa Pengurutan
Pilihan: 2
No. | Nama Barang          | Stok
-----
1.  | batu                  | 64
2.  | semen                 | 13

Tekan Enter untuk melanjutkan...
█
```

Gambar 4.2 Tampilkan Inventaris sorting stok descending selectionsort

```
Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 1
Pilih metode pengurutan:
1. Nama Barang (Ascending - Bubble Sort)
2. Stok Barang (Descending - Selection Sort)
3. Nama Barang (Descending - Insertion Sort)
4. Tanpa Pengurutan
Pilihan: 3
No. | Nama Barang          | Stok
-----
1.  | semen                 | 13
2.  | batu                  | 64

Tekan Enter untuk melanjutkan...
█
```

Gambar 4.3 Tampilkan Inventaris sorting nama descending insertionsort

```

Menu Manajemen Inventaris
1. Tampilkan Inventaris
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Logout
Pilihan: 1
Pilih metode pengurutan:
1. Nama Barang (Ascending - Bubble Sort)
2. Stok Barang (Descending - Selection Sort)
3. Nama Barang (Descending - Insertion Sort)
4. Tanpa Pengurutan
Pilihan: 4
No. | Nama Barang          | Stok
-----
1.  | semen                  | 13
2.  | batu                   | 64

Tekan Enter untuk melanjutkan...

```

Gambar 4.4 Tampilkan Inventaris tanpa sorting

```

Masukkan nomor barang yang ingin diubah: 1
Masukkan nama baru barang: kayu
Masukkan stok baru barang: 1
Barang berhasil diupdate.

Tekan Enter untuk melanjutkan...

```

Gambar 5 Update Barang

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

Tekan Enter untuk melanjutkan...



Gambar 6 Hapus Barang

Menu Manajemen Inventaris

1. Tampilkan Inventaris

2. Tambah Barang

3. Update Barang

4. Hapus Barang

5. Logout

Pilihan: 1

Pilih metode pengurutan:

1. Nama Barang (Ascending - Bubble Sort)

2. Stok Barang (Descending - Selection Sort)

3. Nama Barang (Descending - Insertion Sort)

4. Tanpa Pengurutan

Pilihan: 4

No.	Nama Barang	Stok
-----	-------------	------

1.	semen	13
----	-------	----

Tekan Enter untuk melanjutkan...



Gambar 7 Hasil Inventaris Akhir

5. GIT

5.1 Git Add (Menambah File yang Ingin Dicommit)

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

Gambar 5.1 Git Add

5.2 Git Commit (CheckPoint)

```
PS C:\Users\silly\Downloads\Praktikum-apl> git commit -m "kelarr"  
On branch main  
nothing to commit, working tree clean  
PS C:\Users\silly\Downloads\Praktikum-apl> 
```

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
```

Gambar 5.3 Git Push