

LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT

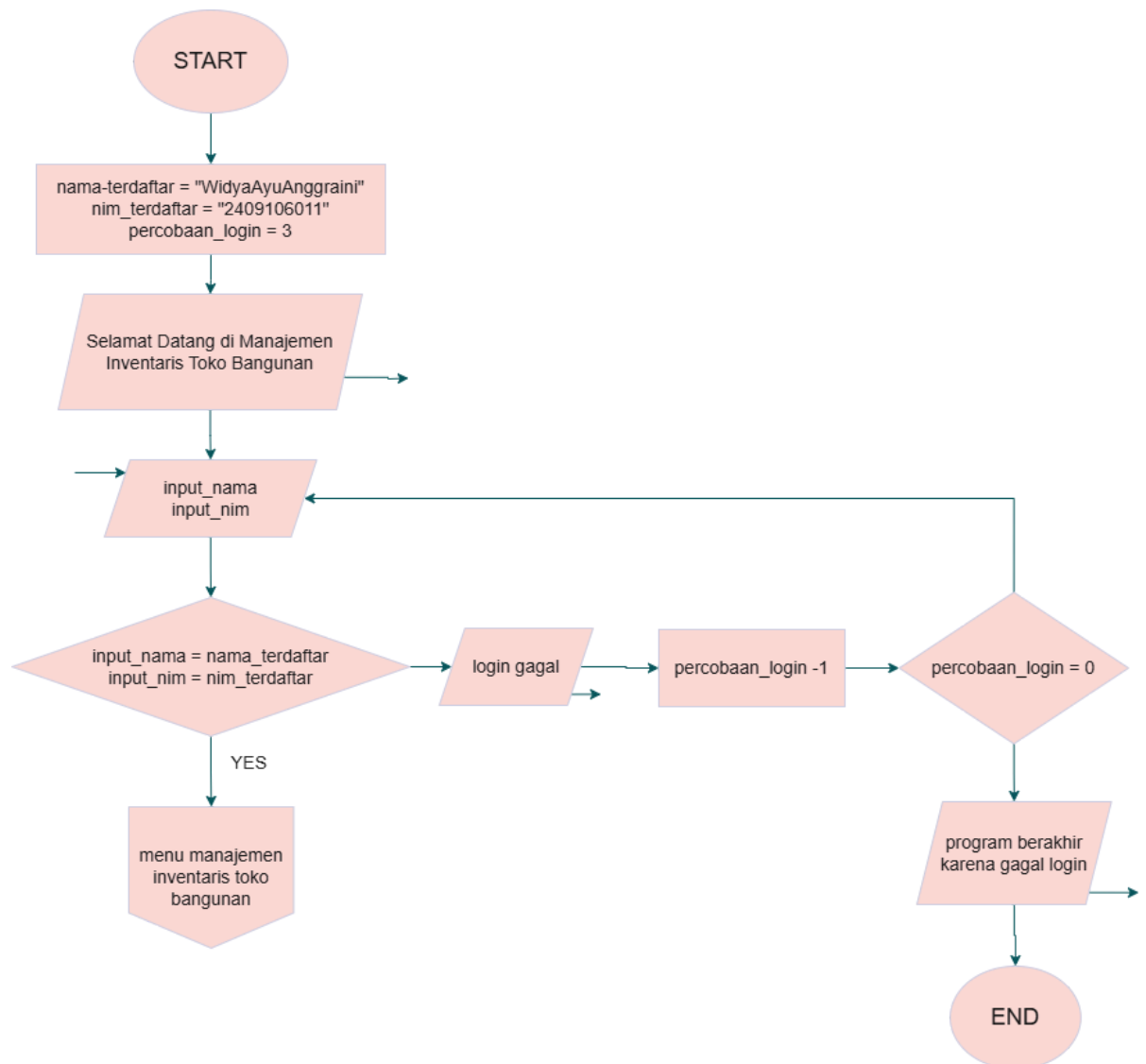


Disusun oleh:
Nama (2409106011)
Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

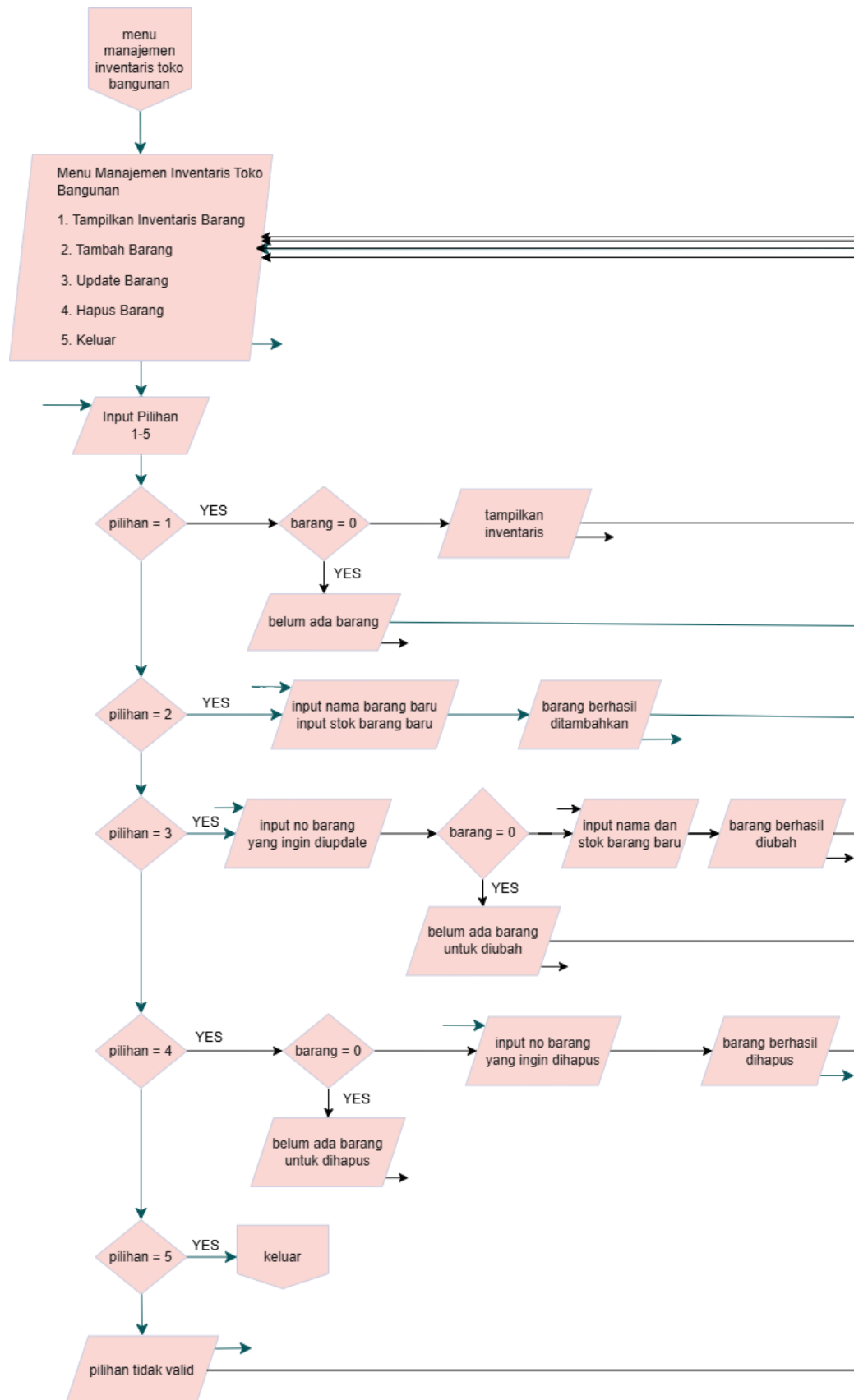
1. Flowchart

A. Fungsi Login



Gambar 1.1 Flowchart

B. Fungsi Menu Utama



Gambar 1.2 Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Program **Manajemen Inventaris Toko Bangunan** ini dibuat untuk membantu pengguna dalam mengelola data barang yang tersedia di toko bangunan. Program ini menggunakan bahasa C++ dan memiliki beberapa fitur utama, yaitu:

1. Login Sistem

- Pengguna harus memasukkan **nama** dan **NIM** yang sesuai untuk mengakses sistem.
- Jika gagal login setelah **3 kali percobaan**, program akan keluar secara otomatis.

2. Menu Utama

Setelah berhasil login, pengguna dapat memilih beberapa opsi dalam **manajemen inventaris**, yaitu:

- **Tampilkan Inventaris Barang:** Menampilkan daftar barang beserta jumlah stoknya.
- **Tambah Barang:** Memungkinkan pengguna menambahkan barang baru ke dalam daftar inventaris.
- **Update Barang:** Mengubah nama atau stok barang yang sudah ada.
- **Hapus Barang:** Menghapus barang dari daftar inventaris.
- **Keluar:** Mengakhiri program.

3. Struktur Data

- Program menggunakan **array statis** dengan kapasitas maksimum **100 barang**.
- Setiap barang direpresentasikan dengan **struct** yang terdiri dari **nama barang** dan **stok**.

4. Pengolahan Data

- Penambahan barang dilakukan dengan menyimpan data ke dalam array.
- Perubahan barang dilakukan dengan mengganti nilai pada indeks tertentu.
- Penghapusan barang dilakukan dengan menggeser elemen dalam array untuk menghindari kekosongan data.

Program ini cocok digunakan untuk pengelolaan sederhana inventaris toko bangunan. Namun, karena menggunakan array statis, program memiliki keterbatasan dalam jumlah barang.

2.2 Penjelasan Alur & Algoritma

1. Alur Program

A. Menampilkan Pesan Selamat Datang

- Program menampilkan pesan pembuka untuk menyambut pengguna.

B. Proses Login

- Pengguna diminta memasukkan **Nama** dan **NIM**.
- Jika cocok dengan data yang terdaftar, login berhasil.
- Jika salah, pengguna diberikan **3 kesempatan** untuk mencoba login.
- Jika gagal setelah **3 kali percobaan**, program langsung **berhenti**.

C. Menampilkan Menu Utama

- Setelah berhasil login, program menampilkan **menu pilihan**:
 - **Tampilkan Inventaris Barang**
 - **Tambah Barang**
 - **Update Barang**
 - **Hapus Barang**
 - **Keluar**

D. Memproses Pilihan Pengguna

- **Pilihan 1: Menampilkan Inventaris Barang**
 - Jika belum ada barang, tampilkan pesan "Belum ada barang".
 - Jika ada barang, tampilkan daftar barang beserta stoknya.
- **Pilihan 2: Menambah Barang**
 - Pengguna memasukkan **nama barang** dan **stok barang**.
 - Data disimpan dalam array jika masih ada kapasitas.
- **Pilihan 3: Memperbarui Barang**

- Program menampilkan daftar barang.
- Pengguna memilih barang yang ingin diperbarui.
- Data barang dapat **diubah** dengan nama dan stok yang baru.
- **Pilihan 4: Menghapus Barang**
 - Program menampilkan daftar barang.
 - Pengguna memilih barang yang ingin **dihapus**.
 - Barang akan dihapus dan array akan digeser untuk menutup celah.
- **Pilihan 5: Keluar**
 - Program akan berhenti dan menampilkan pesan "Program selesai".

E. Mengulangi Menu

- Setelah setiap aksi, pengguna diminta **menekan Enter** untuk kembali ke menu.
- Program akan berjalan hingga pengguna memilih **Keluar (5)**.

2. Algoritma Program

a. Algoritma Login

1. Tampilkan pesan selamat datang.
2. Tetapkan nama dan NIM yang telah terdaftar.
3. Inisialisasi percobaan_login = 0 dan login_berhasil = false.
4. SELAMA (percobaan_login < 3 DAN login_berhasil == false) lakukan:
 - 4.1. Minta pengguna memasukkan nama dan NIM.
 - 4.2. JIKA nama dan NIM sesuai dengan yang terdaftar:
 - Set login_berhasil = true.
 - Tampilkan pesan "Login Berhasil!".
 - 4.3. JIKA salah:
 - Tambah percobaan_login.

- Hitung dan tampilkan sisa percobaan login.
- Jika percobaan_login == 3, tampilkan pesan gagal dan keluar.

b. Algoritma Menampilkan Barang

1. JIKA banyak == 0:

- Tampilkan "Belum ada barang".

2. JIKA ada barang dalam array:

- UNTUK setiap barang dari 0 hingga banyak - 1:
- Tampilkan nomor barang, nama, dan stok.

c. Algoritma Menambah Barang

1. JIKA banyak < MAX_BARANG:

- Minta pengguna memasukkan nama barang.
- Minta pengguna memasukkan stok barang.
- Simpan data barang ke dalam array pada indeks banyak.
- Tambah nilai banyak (jumlah barang bertambah).
- Tampilkan pesan "Barang berhasil ditambahkan".

2. JIKA banyak == MAX_BARANG:

- Tampilkan pesan "Kapasitas penuh! Tidak bisa menambah barang lagi."

d. Algoritma Memperbarui Barang

1. JIKA banyak == 0:

- Tampilkan "Belum ada barang untuk diubah."

2. JIKA ada barang:

- Tampilkan daftar barang.
- Minta pengguna memilih nomor barang yang akan diubah.
- JIKA nomor valid ($1 \leq \text{index} \leq \text{banyak}$):
- Minta nama baru dan stok baru.
- Perbarui barang di array dengan data baru.
- Tampilkan pesan "Barang berhasil diubah".
- JIKA tidak valid:
- Tampilkan pesan "Nomor barang tidak valid".

e. Algoritma Menghapus Barang

1. JIKA banyak == 0:

- Tampilkan "Belum ada barang untuk dihapus."

2. JIKA ada barang:

- Tampilkan daftar barang.
- Minta pengguna memilih nomor barang yang akan dihapus.
- JIKA nomor valid ($1 \leq \text{index} \leq \text{banyak}$):
- Geser semua elemen setelah index ke kiri (menimpa yang dihapus).
- Kurangi nilai banyak.
- Tampilkan pesan "Barang berhasil dihapus".
- JIKA tidak valid:
- Tampilkan pesan "Nomor barang tidak valid".

f. Algoritma Keluar Program

1. Tampilkan pesan "Program selesai".
2. Keluar dari loop utama.

3. Source Code

1. Header dan Definisi Konstanta

SOURCE CODE :

```
#include <iostream>
#include <string>
#include <limits> // Untuk numeric_limits

#define MAX_BARANG 100
#define KOLOM_BARANG 2 // Kolom untuk nama dan stok

using namespace std;
```

- Bagian ini menyertakan pustaka yang diperlukan dan mendefinisikan konstanta untuk mengatur ukuran array barang dan jumlah kolom.

2. Struktur Data **Barang**

SOURCE CODE :

```
struct Barang {
    string data[KOLOM_BARANG];
};
```

- Struktur **Barang** digunakan untuk menyimpan informasi tentang setiap barang, yaitu nama dan stoknya.

3. Fungsi **tampilkanInventaris**

SOURCE CODE :

```
// Fungsi untuk menampilkan tabel inventaris
int tampilkanInventaris(Barang barang[], int banyak) {
    if (banyak == 0) {
        cout << "Belum ada barang" << endl;
        return 0; // Mengembalikan 0 jika tidak ada barang
    }

    cout << "No. | Nama Barang          | Stok" << endl;
    cout << "-----" << endl;

    for (int i = 0; i < banyak; i++) {
        cout << i + 1 << ". | ";
        cout << barang[i].data[0];
        for (int j = barang[i].data[0].length(); j < 22; j++) {
            cout << " ";
        }
        cout << "| ";
        cout << barang[i].data[1] << endl;
    }
    return 1; // Mengembalikan 1 jika barang ditampilkan
}
```

- Fungsi ini menampilkan inventaris barang dalam bentuk tabel.
- Jika tidak ada barang, fungsi akan menampilkan pesan "Belum ada barang".

4. Fungsi **main** (Program Utama)

A. Blok Kode Inisialisasi dan Login

SOURCE CODE :

```
int main() {
    // Informasi Login
    string nama_terdaftar = "WidyaAyuAnggraini";
    string nim_terdaftar = "2409106011";

    cout << "Selamat Datang di Manajemen Inventaris Toko Bangunan" << endl;

    // Proses Login
    int percobaan_login = 0;
    bool login_berhasil = false;
}
```

```

while (percobaan_login < 3 && !login_berhasil) {
    string nama_input, nim_input;

    cout << "\n==== Login ==== " << endl;
    cout << "Masukkan Nama: ";
    cin >> nama_input;
    cout << "Masukkan NIM: ";
    cin >> nim_input;

    if (nama_input == nama_terdaftar && nim_input == nim_terdaftar) {
        login_berhasil = true;
        cout << "Login Berhasil! Selamat datang, " << nama_terdaftar <<
        "!" << endl;
    } else {
        percobaan_login++;
        int sisa_percobaan = 3 - percobaan_login;
        cout << "Login Gagal! Sisa percobaan: " << sisa_percobaan <<
        endl;

        if (percobaan_login == 3) {
            cout << "Anda telah melebihi batas percobaan login." <<
            endl;

            return 0;
        }
    }
}
if (!login_berhasil) {
    cout << "Program berakhir karena gagal login." << endl;
    return 0;
}

```

- Tujuan: Blok kode ini menangani proses inisialisasi variabel dan login pengguna.
- Penjelasan:
 - Variabel `nama_terdaftar` dan `nim_terdaftar` menyimpan informasi login yang benar.
 - Program menyambut pengguna dengan pesan selamat datang.
 - Loop `while` digunakan untuk memberikan 3 kali kesempatan login.
 - Pengguna diminta memasukkan nama dan NIM.
 - Jika informasi login benar, `login_berhasil` diatur menjadi `true`, dan pesan selamat datang ditampilkan.
 - Jika login gagal, `percobaan_login` bertambah, dan sisa percobaan ditampilkan.

- Jika 3 kali login gagal, program berakhir.
- Apabila login tidak berhasil maka program akan dihentikan.

B. Blok Kode Menu dan Pemrosesan Inventaris

```
int banyak = 0; // Jumlah elemen saat ini
Barang barang[MAX_BARANG]; // Array 2 dimensi
int pilihan, index;

do {
    system("cls");
    cout << "Menu Manajemen Inventaris Toko Bangunan" << endl;
    cout << "1. Tampilkan Inventaris Barang" << endl;
    cout << "2. Tambah Barang" << endl;
    cout << "3. Update Barang" << endl;
    cout << "4. Hapus Barang" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";

    // Validasi input pilihan
    if (!(cin >> pilihan)) {
        cout << "Input tidak valid. Masukkan angka." << endl;
        cin.clear();
        cin.ignore(); // Bersihkan buffer input
        pilihan = 0; // Set pilihan ke nilai default
        continue; // Kembali ke awal loop
    }

    cin.ignore(); // Membersihkan newline

    switch (pilihan) {
        case 1:
            tampilkanInventaris(barang, banyak);
            break;

        case 2:
            if (banyak < MAX_BARANG) {
                cout << "Masukkan nama barang: ";
                getline(cin, barang[banyak].data[0]);
                cout << "Masukkan stok barang: ";
            }
    }
}
```

```

        getline(cin, barang[banyak].data[1]);
        banyak++;
        cout << "Barang berhasil ditambahkan" << endl;
    } else {
        cout << "Kapasitas penuh! Tidak bisa menambah
barang lagi." << endl;
    }
    break;

case 3:
    if (banyak == 0) {
        cout << "Belum ada barang untuk diubah." <<
endl;
    } else {
        tampilkanInventaris(barang, banyak);
        cout << "Masukkan nomor barang yang akan
diubah: ";

        cin >> index;
        cin.ignore();

        if (index > 0 && index <= banyak) {
            cout << "Masukkan nama barang baru: ";
            getline(cin, barang[index - 1].data[0]);
            cout << "Masukkan stok barang baru: ";
            getline(cin, barang[index - 1].data[1]);
            cout << "Barang berhasil diubah" << endl;
        } else {
            cout << "Nomor barang tidak valid" <<
endl;
        }
    }
    break;

case 4:
    if (banyak == 0) {
        cout << "Belum ada barang untuk dihapus." <<
endl;
    } else {
        tampilkanInventaris(barang, banyak);
        cout << "Masukkan nomor barang yang akan
dihapus: ";

        cin >> index;

```

```

        if (index > 0 && index <= banyak) {
            for (int i = index - 1; i < banyak - 1;
i++) {
                barang[i] = barang[i + 1];
            }
            banyak--;
            cout << "Barang berhasil dihapus" << endl;
        } else {
            cout << "Nomor barang tidak valid" <<
endl;
        }
    }
    break;

    case 5:
        cout << "Program selesai" << endl;
        break;

    default:
        cout << "Pilihan tidak valid" << endl;
        break;
    }
    cout << "\nTekan Enter untuk melanjutkan...";
    cin.ignore();
    cin.get();
} while (pilihan != 5);

return 0;
}

```

- Tujuan: Blok kode ini menampilkan menu utama dan memproses pilihan pengguna untuk mengelola inventaris.
- Penjelasan:
 - Variabel `banyak` menyimpan jumlah barang dalam inventaris.
 - Array `barang` menyimpan data inventaris.
 - Loop `do-while` menampilkan menu dan memproses pilihan sampai pengguna memilih keluar.
 - Validasi input memastikan pengguna memasukkan angka untuk pilihan menu.

- `switch` statement memanggil fungsi yang sesuai berdasarkan pilihan:
 - `case 1`: Menampilkan inventaris.
 - `case 2`: Menambah barang.
 - `case 3`: Memperbarui barang.
 - `case 4`: Menghapus barang.
 - `case 5`: Keluar dari program.
- Setelah setiap pilihan, program menunggu pengguna menekan Enter sebelum menampilkan menu lagi.

Penjelasan Tambahan:

- Program ini menggunakan array statis `barang` untuk menyimpan data inventaris.
- Validasi input dilakukan untuk mencegah kesalahan input pengguna.
- `system("cls")` digunakan untuk membersihkan layar, yang mungkin hanya berfungsi di sistem operasi Windows.
- Program menggunakan struktur perulangan `do while`, yang dimana program akan terus berjalan sampai kondisi pada `while` terpenuhi.
- Program juga menggunakan percabangan `switch case`, yang berguna untuk memilih bagian kode mana yang akan di eksekusi, sesuai dengan input dari user.

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

- Pengguna login dengan benar
- Memilih menu 1
- Memilih menu 2 dengan menambahkan; nama barang : semen; stok barang : 20
- Memilih menu 1 kembali untuk memastikan keberhasilan menu 2

- Memilih menu 3 mengupdate barang 1 dengan data; nama barang baru : batu; stok barang baru : 15
- Memilih menu 4 dengan menghapus barang 1 yang sudah diupdate
- Memilih menu 1 untuk memastikan keberhasilan menu 4
- Memilih menu 5 untuk keluar program

2. Skenario 2

- Pengguna melakukan login dengan 2 kesalahan, lalu login dengan benar ketiga kalinya
- Memilih menu 2 dengan menambahkan; nama barang : semen; stok barang : 20
- Memilih menu 1 untuk memastikan keberhasilan menu 2
- Memilih menu 3 untuk mengupdate barang dengan inputan tidak valid, dimana nomor barang yang valid adalah 1 tetapi pengguna menginput 2
- Memilih menu 4 untuk menghapus barang dengan inputan tidak valid, dimana nomor barang yang valid adalah 1 tetapi pengguna menginput 5
- Memilih menu 1 untuk memastikan ketidakberhasilan pemilihan menu 3 dan 4
- Memilih menu 5 untuk keluar program

4.2 Hasil Output

1. Skenario 1

```
PS C:\Users\silly\Downloads\praktikum-apl> cd "c:\Users\silly\Downloads\praktikum-apl\post-test\post-test-2\" ; if ($?) { g++ 2409106011-WidyaAyuAnggraini-PT-2.cpp -o 2409106011-WidyaAyuAnggraini-PT-2 } ; if ($?) { .\2409106011-WidyaAyuAnggraini-PT-2 }
Selamat Datang di Manajemen Inventaris Toko Bangunan

===== Login =====
Masukkan Nama: WidyaAyuAnggraini
Masukkan NIM: 2409106011
```

Gambar 4.2.1.1 Login


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

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.2 Menu 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 2
Masukkan nama barang: semen
Masukkan stok barang: 20
Barang berhasil ditambahkan

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.3 Menu 2

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 1
No. | Nama Barang          | Stok
-----
1.  | semen                  | 20

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.4 Menu 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 3
No. | Nama Barang          | Stok
-----
1.  | semen                  | 20
Masukkan nomor barang yang akan diubah: 1
Masukkan nama barang baru: batu
Masukkan stok barang baru: 15
Barang berhasil diubah

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.5 Menu 3

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 4
No. | Nama Barang          | Stok
-----
1.  | batu                  | 15
Masukkan nomor barang yang akan dihapus: 1
Barang berhasil dihapus

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.6 Menu 4

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 1
Belum ada barang

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.7 Menu 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 5
Program selesai

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.1.8 Menu 5

2. Skenario 2

```
PS C:\Users\silly\Downloads\praktikum-apl> cd "c:\Users\silly\Downloads\praktikum-apl\post-test\post-test-2\" ; if ($?) { g++ 2409106011-WidyaAyuAnggraini-PT-2.cpp -o 2409106011-WidyaAyuAnggraini-PT-2 } ; if ($?) { .\2409106011-WidyaAyuAnggraini-PT-2 }
Selamat Datang di Manajemen Inventaris Toko Bangunan

===== Login =====
Masukkan Nama: Widya Ayu Anggraini
Masukkan NIM: Login Gagal! Sisa percobaan: 2

===== Login =====
Masukkan Nama: Masukkan NIM: WidyaAyuAnggraini
Login Gagal! Sisa percobaan: 1

===== Login =====
Masukkan Nama: WidyaAyuAnggraini
Masukkan NIM: 2409106011
```

Gambar 4.2.2.1 Login

```
Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 2
Masukkan nama barang: semen
Masukkan stok barang: 20
Barang berhasil ditambahkan

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.2.2 Menu 2

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 3
No. | Nama Barang          | Stok
-----
1.  | semen                  | 20
Masukkan nomor barang yang akan diubah: 2
Nomor barang tidak valid

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.2.3 Menu 3

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 4
No. | Nama Barang          | Stok
-----
1.  | semen                  | 20
Masukkan nomor barang yang akan dihapus: 5
Nomor barang tidak valid

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.2.4 Menu 4

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 1
No. | Nama Barang          | Stok
-----
1.  | semen                  | 20

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.2.5 Menu 1

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu Manajemen Inventaris Toko Bangunan
1. Tampilkan Inventaris Barang
2. Tambah Barang
3. Update Barang
4. Hapus Barang
5. Keluar
Pilihan: 5
Program selesai

Tekan Enter untuk melanjutkan...
```

Gambar 4.2.2.6 Menu 5

5. GIT

5.1 Git Init (Menginisiasi Repository Git)

```
PS C:\Users\silly\Downloads\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/silly/Downloads/praktikum-apl/.git/
PS C:\Users\silly\Downloads\praktikum-apl> █
```

Gambar 5.1 Git Init

5.2 Git Add (Menambah File yang Ingin Dicommit)

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

Gambar 5.2 Git Add

5.3 Git Commit (Checkpoint)

```
PS C:\Users\silly\Downloads\praktikum-apl> git commit -m "Finish post test 2"
[main 314883e] Finish post test 2
 2 files changed, 1 insertion(+), 1 deletion(-)
PS C:\Users\silly\Downloads\praktikum-apl> █
```

Gambar 5.3 Git Commit

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

```
PS C:\Users\silly\Downloads\praktikum-apl> git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 718 bytes | 718.00 KiB/s, done.
Total 8 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/WidyaRanyyy/praktikum-apl
 36c8450..314883e  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\silly\Downloads\praktikum-apl> █
```

Gambar 5.4 Git Push