# LAPORAN PRAKTIKUM

# POSTTEST 4

# ALGORITMA PEMROGRAMAN LANJUT


**Disusun oleh:**

**Niky Jenita Putri (2409106019)**

**Kelas (A1 ‘24)**

**PROGRAM STUDI INFORMATIKA**

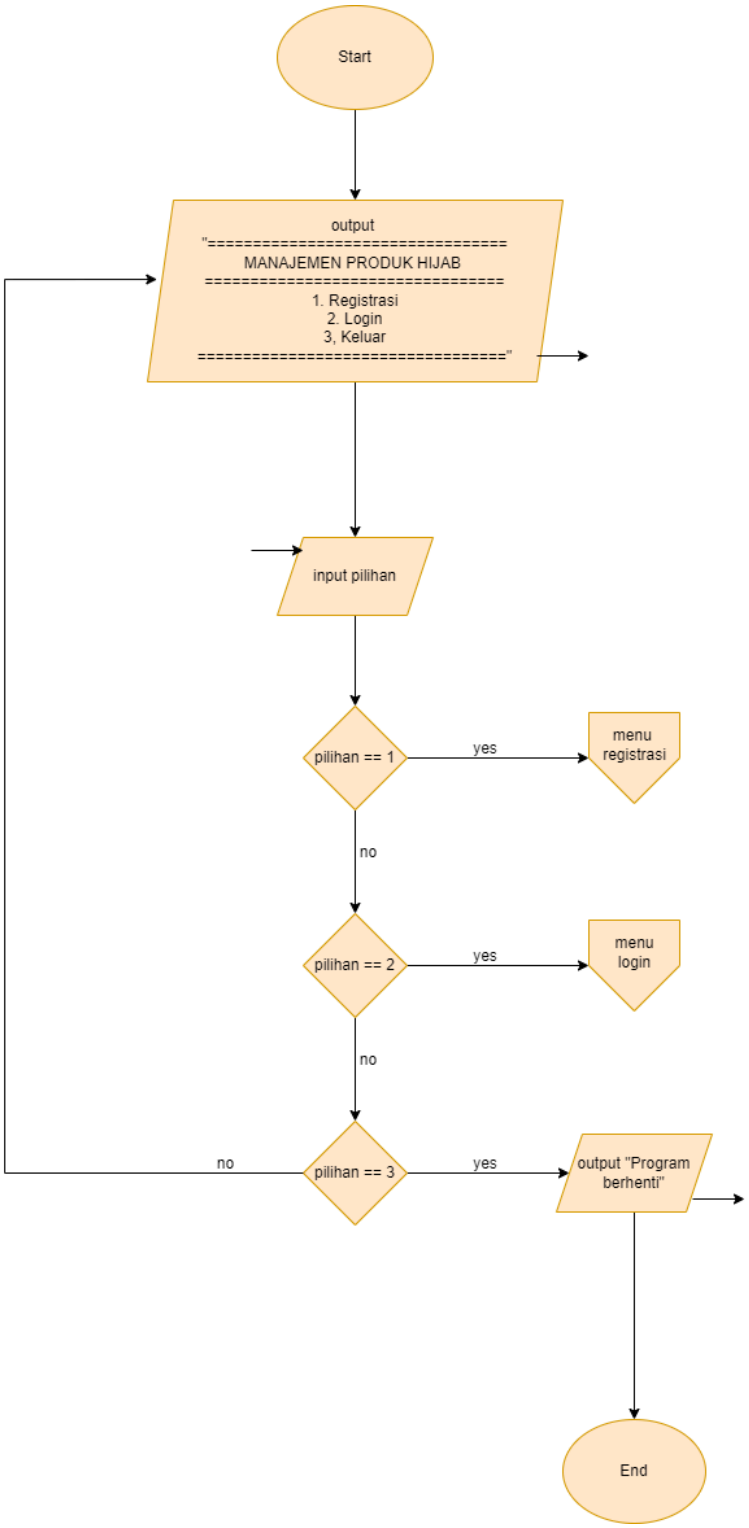
**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

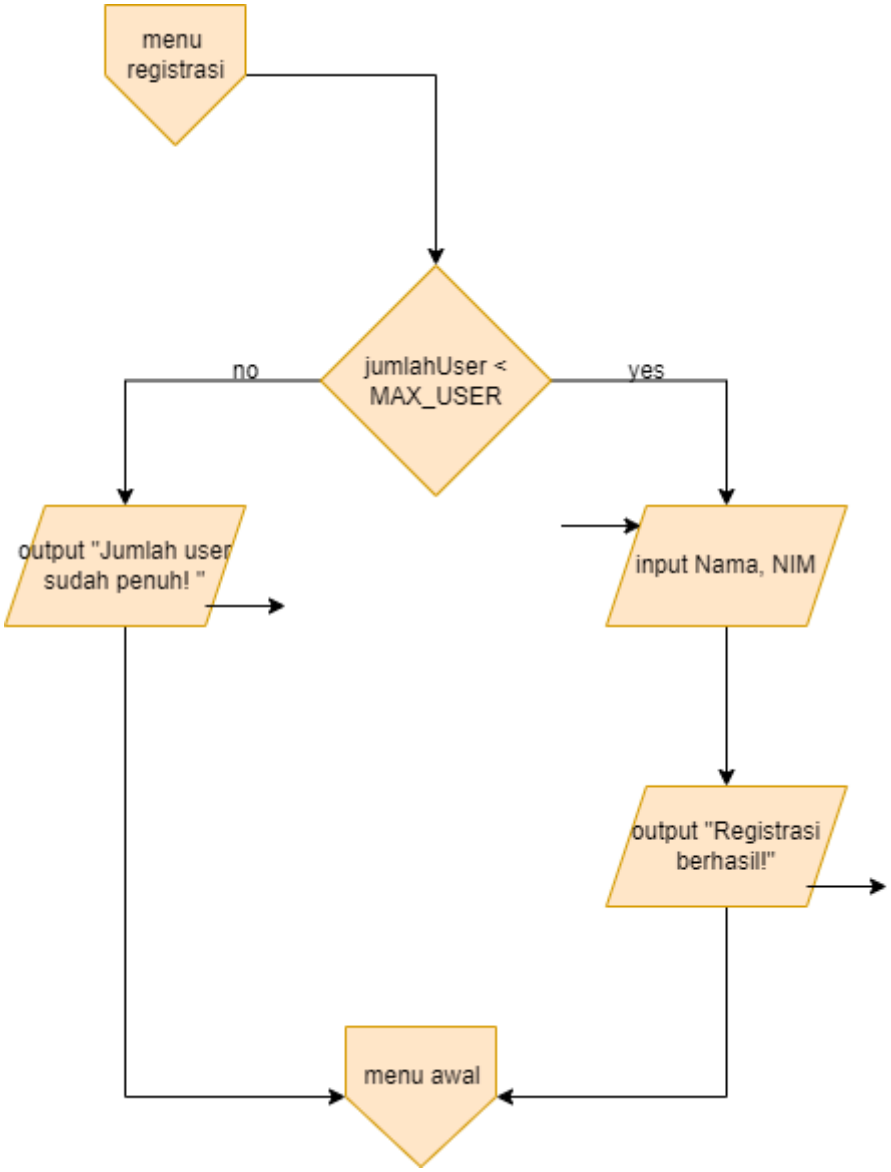
# 1. Flowchart

## 1.1 Menu Awal



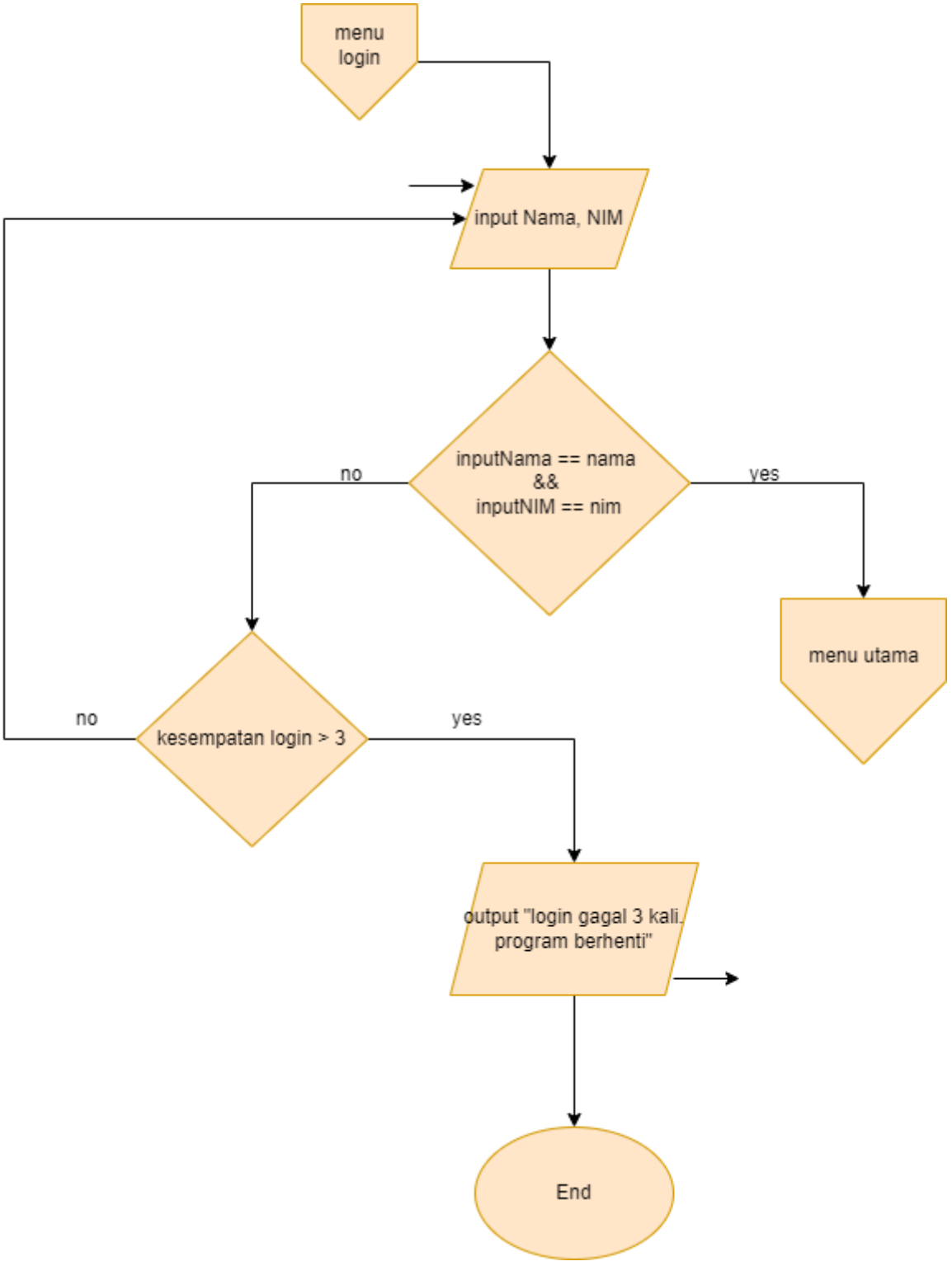

Gambar 1.1 Menu Awal

## 1.2 Menu Registrasi



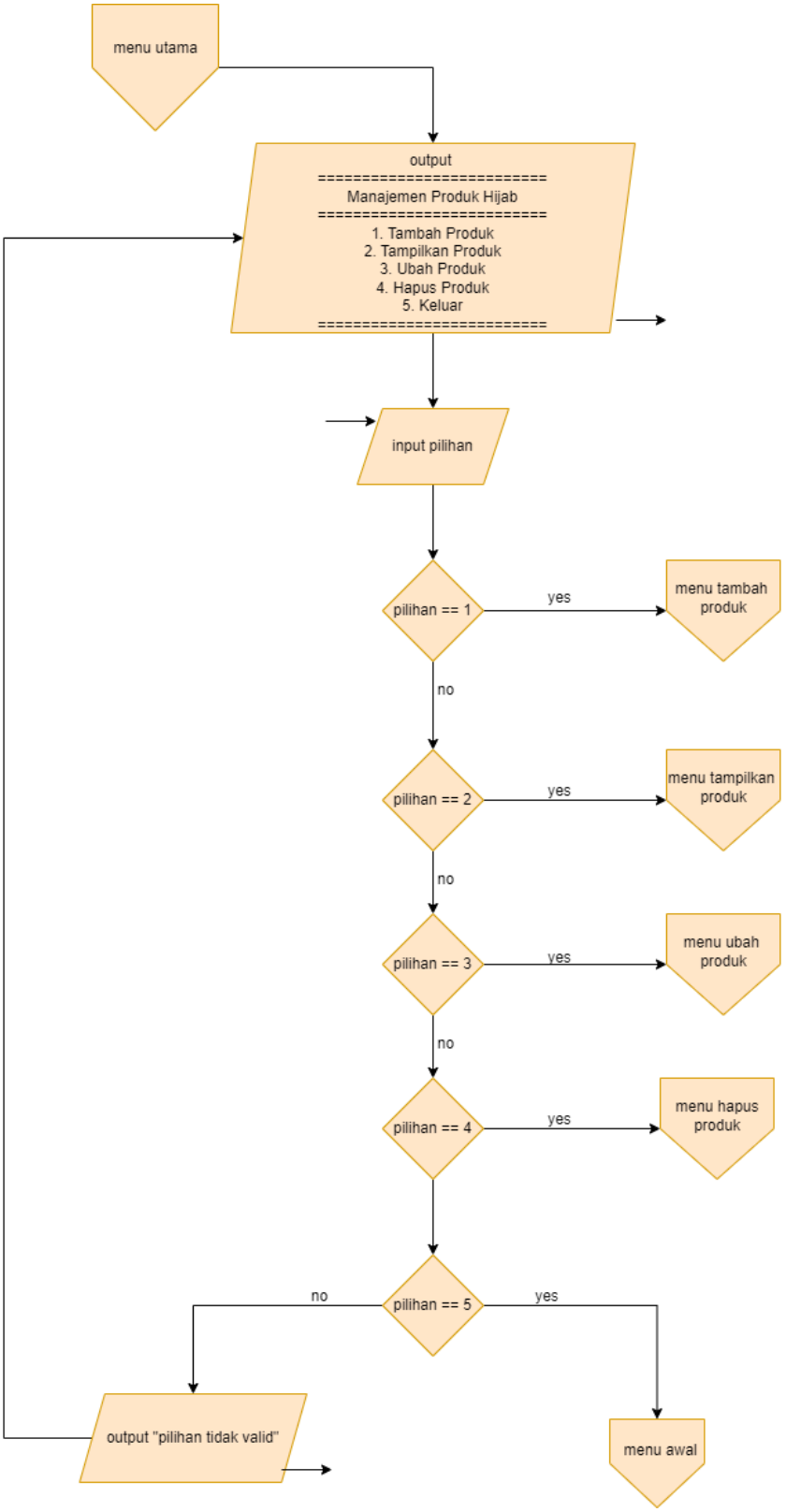

Gambar 1.2 Menu Registrasi

### 1.3 Menu Login



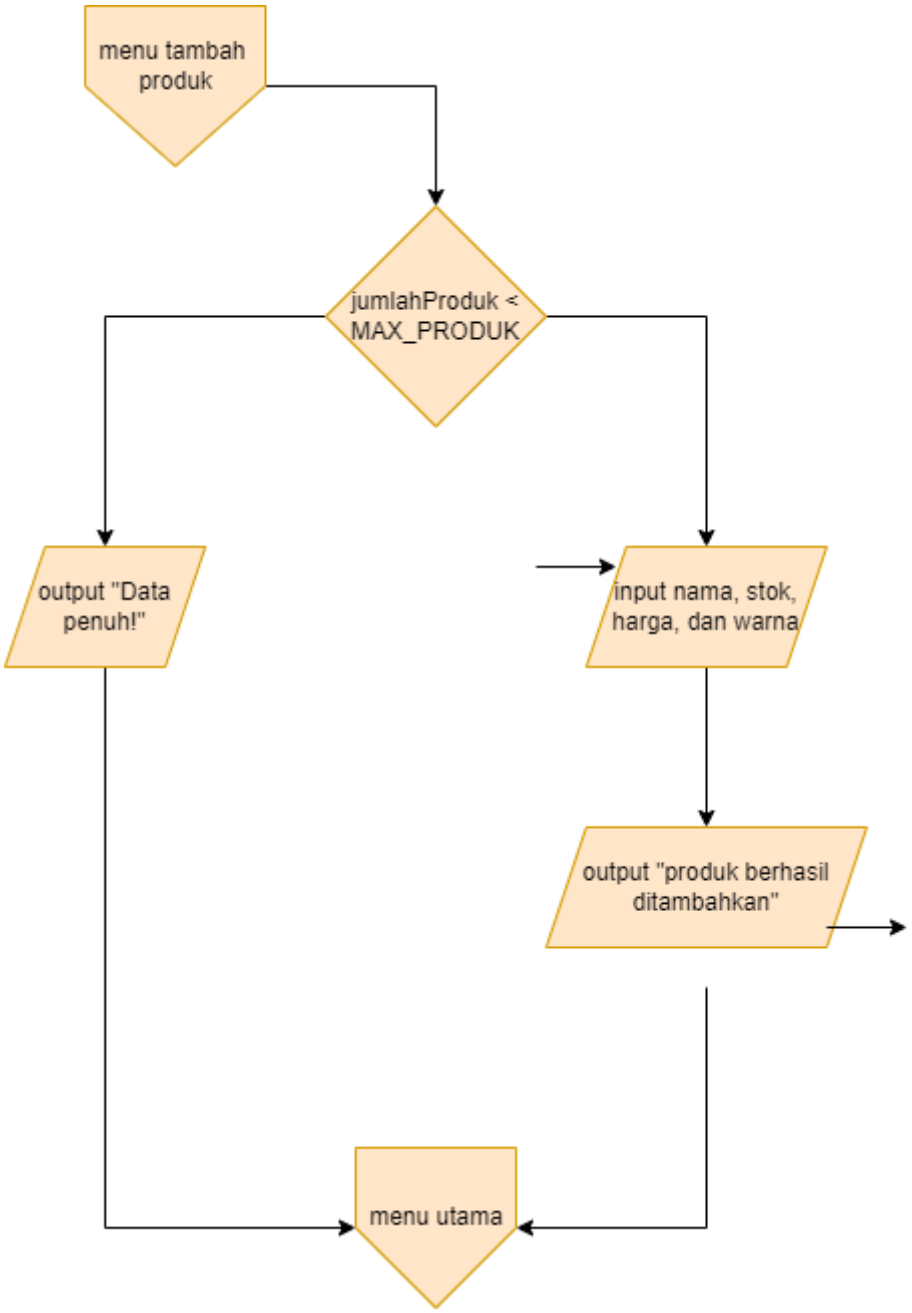

Gambar 1.3 Menu Login

## 1.4 Menu Utama



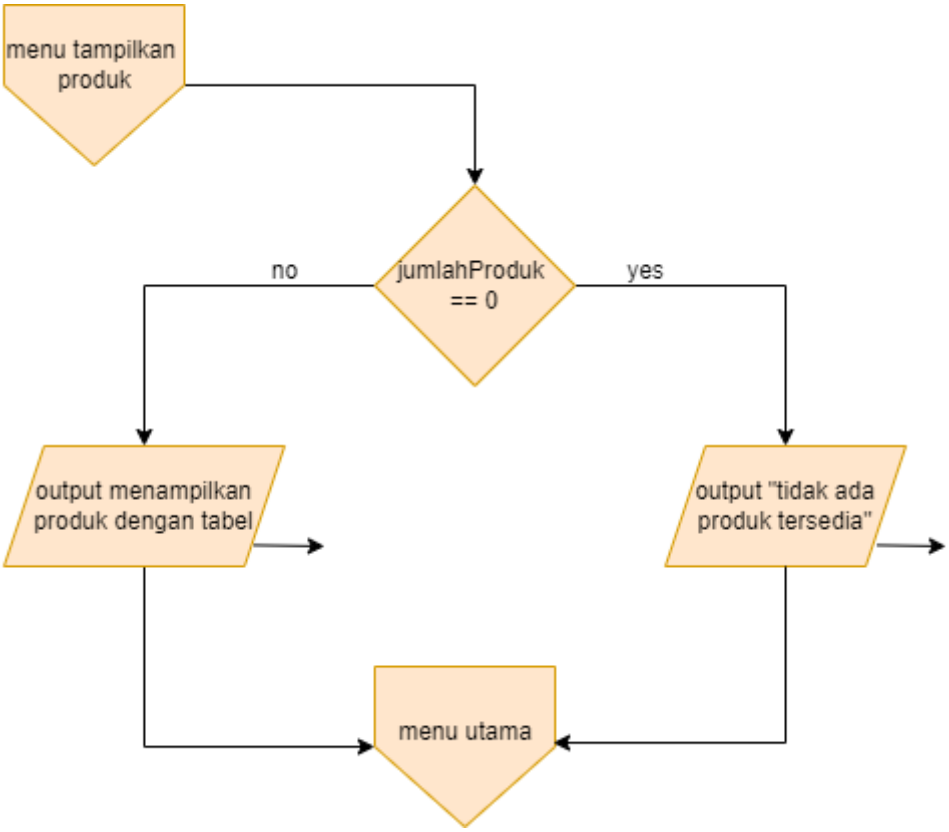

Gambar 1.4 Menu Utama

## 1.5 Menu Tambah Produk



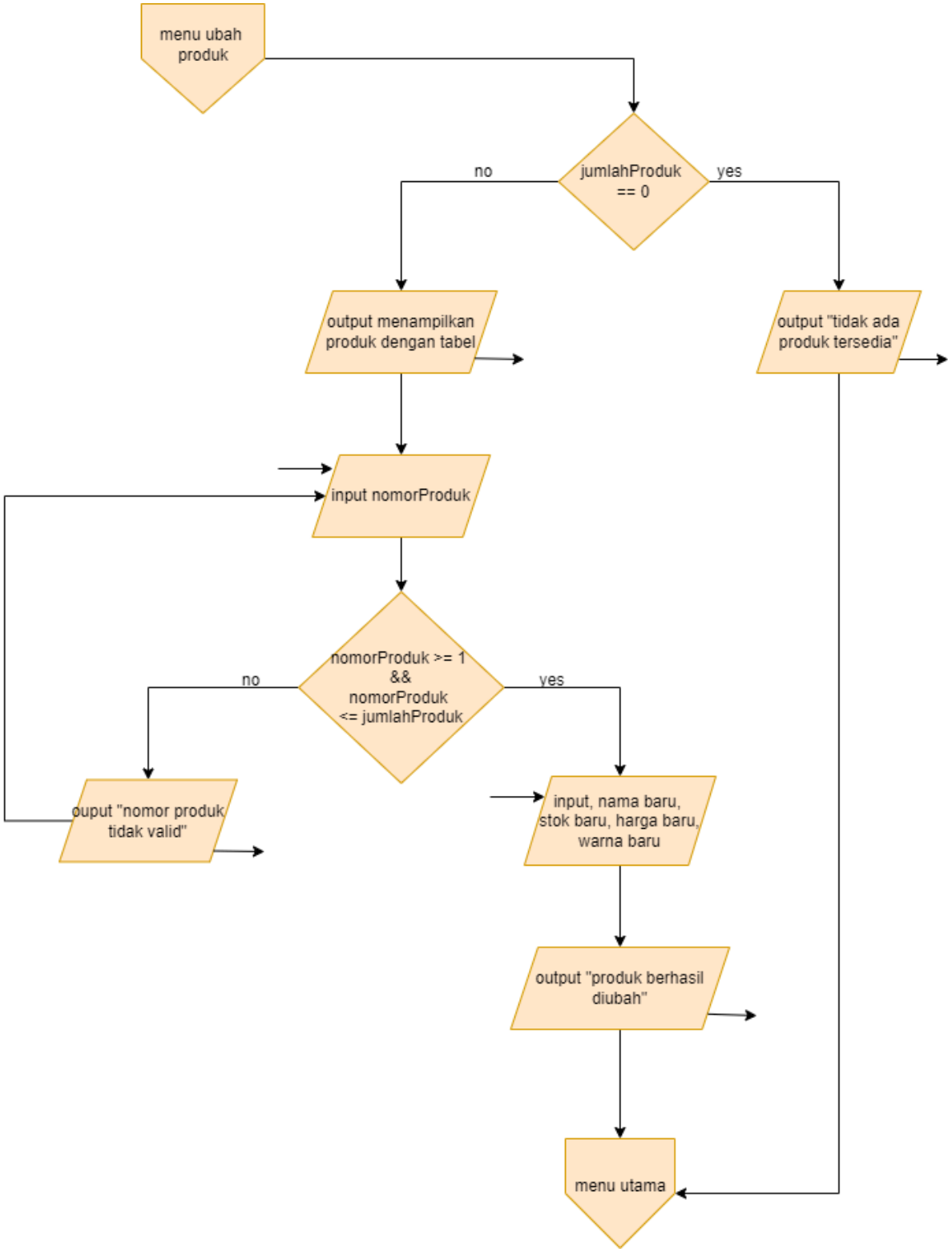

Gambar 1.5 Menu Tambah Produk

## 1.6 Menu Tampilkan Produk



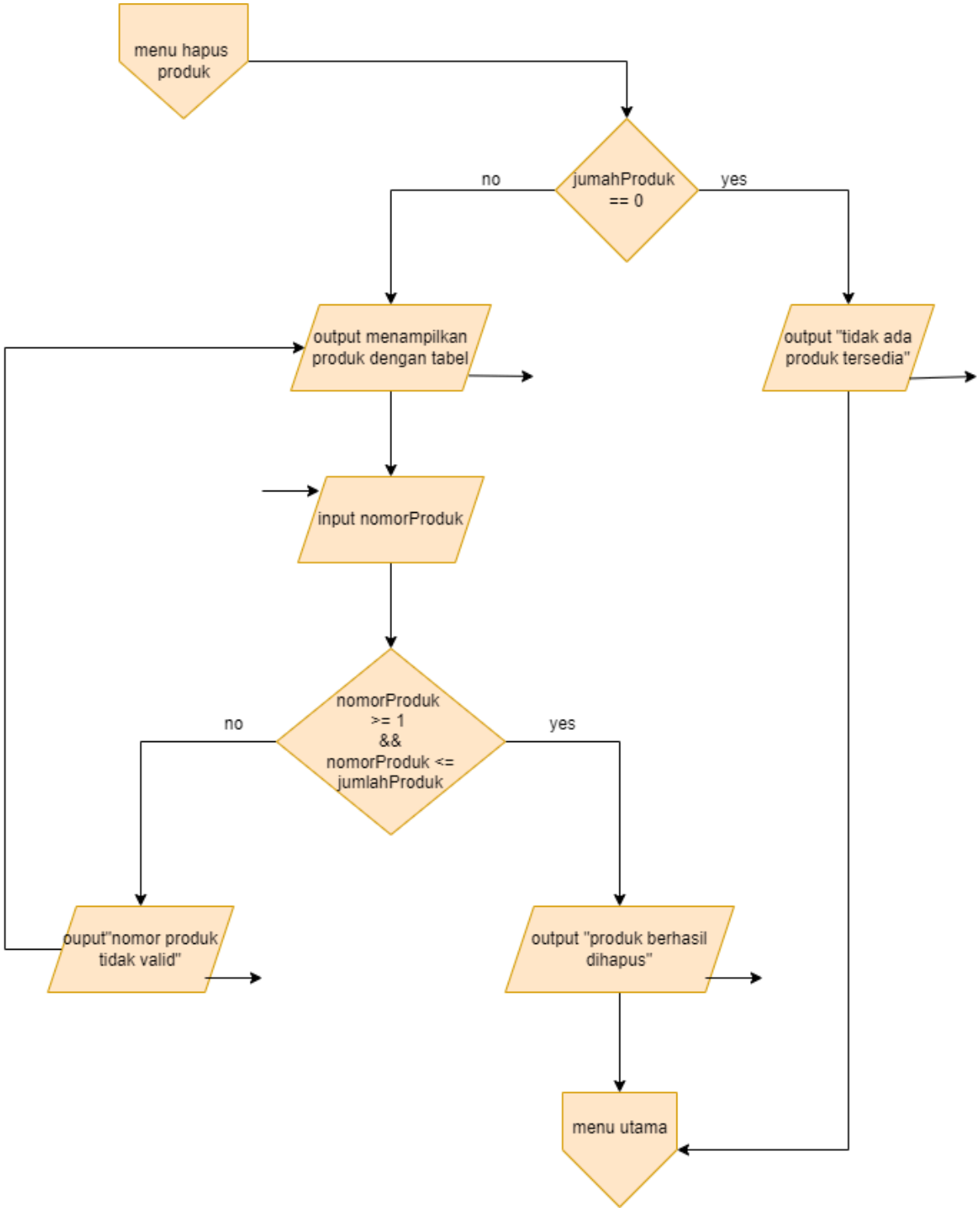

Gambar 1.6 Menu Tampilkan Produk

## 1.7 Menu Ubah Produk




Gambar 1.7 Menu Ubah Produk

## 1.8 Menu Hapus Produk




Gambar 1.8 Menu Hapus  Produk

## 2. Analisis Program

Program ini bertujuan untuk menyediakan sistem manajemen toko hijab yang memungkinkan pengguna untuk melakukan registrasi, login, serta mengelola produk-produk yang ada di toko. Pengguna yang telah terdaftar dapat masuk ke sistem dengan menggunakan nama dan NIM mereka. Setelah login berhasil, pengguna dapat menambah produk baru, melihat daftar produk yang tersedia, mengubah informasi produk (seperti nama, stok, harga, dan warna), serta menghapus produk yang sudah tidak diperlukan lagi. Sistem ini juga membatasi jumlah percobaan login hingga tiga kali untuk memastikan keamanan akses. Dengan demikian, program ini memberikan kemudahan bagi admin dalam mengelola produk dan mengontrol akses pengguna di toko hijab.

## 3. Source Code

### 3.1 Tampilkan Menu

Program void tampilkanMenu() adalah sebuah fungsi prosedur yang bertugas untuk menampilkan menu utama dari aplikasi manajemen toko hijab ke layar. Fungsi ini tidak menerima parameter dan tidak mengembalikan nilai (void).

```
//Prosedur
void tampilkanMenu() {
    cout << "======================================" << endl;
    cout << "|         MANAJEMEN TOKO HIJAB       |" << endl;
    cout << "======================================" << endl;
    cout << "|            1. Registrasi           |" << endl;
    cout << "|            2. Login                |" << endl;
    cout << "|            3. Keluar               |" << endl;
    cout << "======================================" << endl;
}
```

### 3.2 Menu Registrasi

Fungsi registrasiUser merupakan sebuah fungsi dalam program manajemen toko hijab yang digunakan untuk melakukan proses registrasi pengguna baru. Fungsi ini menerima dua parameter, yaitu array users[] yang berisi data semua pengguna dalam bentuk struct User, dan referensi jumlahUser yang menyatakan jumlah pengguna yang sudah terdaftar.

```
//Fungsi
int registrasiUser(User users[], int &jumlahUser) {
    if (jumlahUser < MAX_USER) {
        cout << "Masukkan Nama : ";
        getline(cin, users[jumlahUser].nama);
        cout << "Masukkan NIM : ";
        getline(cin, users[jumlahUser].nim);
        jumlahUser++;
        return jumlahUser - 1;
    } else {
        cout << "Jumlah user sudah penuh!\n";
        return -1;
    }
}
```

### 3.3 Menu Login

Fungsi loginUser digunakan untuk proses login pengguna pada program manajemen toko hijab. Fungsi ini meminta input nama dan NIM, lalu mencocokkannya dengan data yang sudah terdaftar dalam array users[]. Jika cocok, login berhasil dan fungsi mengembalikan indeks user tersebut. Jika tidak cocok, fungsi akan memanggil dirinya sendiri (rekursif) untuk

memberikan kesempatan login ulang, hingga maksimal jumlah percobaan yaitu 3 kali. Jika gagal login sebanyak tiga kali, program akan dihentikan.

```
//Fungsi rekursif
int loginUser(User users[], int jumlahUser, int loginAttempts) {
    string inputNama, inputNIM;
    int userIndex = -1;

    if (loginAttempts == MAX_LOGIN_ATTEMPTS) {
        cout << "Login gagal 3 kali. Program berhenti.\n";
        exit(0);
    }

    cout << "Masukkan Nama : ";
    getline(cin, inputNama);
    cout << "Masukkan NIM : ";
    getline(cin, inputNIM);

    for (int i = 0; i < jumlahUser; i++) {
        if(users[i].nama == inputNama && users[i].nim == inputNIM) {
            userIndex = i;
            break;
        }
    }

    if (userIndex != -1) {
        cout << "Login berhasil\n";
        return userIndex;
    } else {
        cout << "Login gagal. Coba Lagi,\n";
        return loginUser(users, jumlahUser, loginAttempts + 1); //
    }
}
```

### 3.4 Menu Utama

Fungsi menuUser menampilkan menu setelah login, berisi pilihan untuk menambah, menampilkan, mengubah, dan menghapus produk.

```
//Prosedur
void menuUser (int userIndex, Produk produk[], int &jumlahProduk) {
    int pilihan;
    while (userIndex != -1) {
        cout << "=======================================" << endl;
        cout << "|         MANAJEMEN TOKO HIJAB        |" << endl;
        cout << "=======================================" << endl;
        cout << "|          1. Tambah Produk           |" << endl;
        cout << "|          2. Tampilkan Produk        |" << endl;
        cout << "|          3. Ubah Produk             |" << endl;
        cout << "|          4. Hapus Produk            |" << endl;
```

```
        cout << "|          5. Keluar                  |" << endl;
        cout << "=======================================" << endl;
        cout << "Masukkan pilihan : ";
        cin >> pilihan;
        cin.ignore();

        if (pilihan == 1) {
            tambahProduk(produk, jumlahProduk);
        } else if (pilihan == 2) {
            tampilkanProduk(produk, jumlahProduk);
        } else if (pilihan == 3) {
            ubahProduk(produk, jumlahProduk);
        } else if (pilihan == 4) {
            hapusProduk(produk, jumlahProduk);
        } else if (pilihan == 5) {
            userIndex = -1;
        }
    }
}
```

### 3.5 Menu Tambah Produk

Kedua fungsi tambahProduk di bawah adalah fungsi overloading, yaitu dua fungsi dengan nama yang sama tetapi parameter berbeda. Fungsi pertama meminta input langsung dari pengguna untuk nama produk, stok, harga, dan warna, lalu menyimpannya ke dalam array produk[], jika masih ada ruang. Fungsi kedua melakukan hal yang sama, tetapi menerima semua data produk sebagai parameter, sehingga cocok digunakan jika data produk sudah tersedia sebelumnya. Keduanya akan menambahkan data ke array produk dan menambah jumlahProduk jika belum mencapai batas maksimum.

```
//Fungsi overloading
bool tambahProduk(Produk produk[], int &jumlahProduk) {
    if (jumlahProduk < MAX_PRODUK) {
        cout << "Masukkan nama produk : ";
        getline(cin, produk[jumlahProduk].nama);
        cout << "Masukkan stok : ";
        cin >> produk[jumlahProduk].stok;
        cout << "Masukkan harga : ";
        cin >> produk[jumlahProduk].harga;
        cin.ignore();
        for (int k = 0; k < 3; k++) {
            cout << "Warna standar ke-" << k+1 << ": ";
            getline(cin, produk[jumlahProduk].warna.standar[k]);
            cout << "Warna premium ke-" << k+1 << ": ";
            getline(cin, produk[jumlahProduk].warna.premium[k]);
        }
        jumlahProduk++;
```

```
        cout << "Produk berhasil ditambahkan.\n";
        return true;
    } else {
        cout << "Data penuh!\n";
        return false;
    }
}

//Fungsi overloading
bool tambahProduk(Produk produk[], int &jumlahProduk, string nama, int stok, int
harga, Warna warna) {
    if (jumlahProduk < MAX_PRODUK) {
        produk[jumlahProduk].nama = nama;
        produk[jumlahProduk].stok = stok;
        produk[jumlahProduk].harga = harga;
        produk[jumlahProduk].warna = warna;
        jumlahProduk++;
        cout << "Produk berhasil ditambahkan.\n";
        return true;
    } else {
        cout << "Data penuh!\n";
        return false;
    }
}
```

### 3.6 Menu Tampilkan Produk

Kedua fungsi tampilkanProduk merupakan fungsi overloading yang menampilkan data produk, namun dengan cara berbeda. Fungsi pertama menampilkan semua produk yang tersimpan jika jumlahProduk tidak nol. Fungsi kedua menampilkan hanya produk yang memiliki harga lebih besar atau sama dengan minHarga. .

```
//Fungsi overloading
bool tampilkanProduk(Produk produk[], int &jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        return false;
    } else {
        cout <<
"\n=================================================================================
=========\n";
        cout << "| No | Nama        | Stok | Harga  |    Warna Standar    |
Warna Premium    |\n";
        cout <<
"\n=================================================================================
=========\n";
        for (int i = 0; i < jumlahProduk; i++) {
            cout << "| " << setw(2) << i+1 << " |"
                 << setw(12) << produk[i].nama << " |"
```

```
                << setw(6) << produk[i].stok << " |"
                << setw(7) << produk[i].harga << " |"
                << setw(20) << produk[i].warna.standar[0] +", " +
produk[i].warna.standar[1] + ", " + produk[i].warna.standar[2] << " |"
                << setw(25) << produk[i].warna.premium[0] +", " +
produk[i].warna.premium[1] + ", " + produk[i].warna.premium[2] << " |\n";
        }
        cout <<
"\n=============================================================================
=========\n";
        return true;
    }
}

//Fungsi overloading
bool tampilkanProduk(Produk produk[], int &jumlahProduk, int minHarga) {
    bool found = false;
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        return false;
    } else {
        cout <<
"\n=============================================================================
=========\n";
        cout << "| No | Nama        | Stok | Harga  |    Warna Standar    |
Warna Premium    |\n";
        cout <<
"\n=============================================================================
=========\n";
        for (int i = 0; i < jumlahProduk; i++) {
            if (produk[i].harga >= minHarga) {
                cout << "| " << setw(2) << i+1 << " |"
                     << setw(12) << produk[i].nama << " |"
                     << setw(6) << produk[i].stok << " |"
                     << setw(7) << produk[i].harga << " |"
                     << setw(20) << produk[i].warna.standar[0] +", " +
produk[i].warna.standar[1] + ", " + produk[i].warna.standar[2] << " |"
                     << setw(25) << produk[i].warna.premium[0] +", " +
produk[i].warna.premium[1] + ", " + produk[i].warna.premium[2] << " |\n";
                found = true;
            }
        }
        cout <<
"\n=============================================================================
=========\n";
        if (!found) cout << "\nTidak ada produk dengan harga lebih dari " <<
minHarga << endl;
        return found;
    }
}
```

### 3.7 Menu Ubah Produk

Prosedur ubahProduk berfungsi untuk mengedit informasi produk yang telah terdaftar. Pertama-tama, sistem akan menampilkan seluruh daftar produk yang tersedia, kemudian meminta pengguna untuk memilih nomor produk yang ingin diubah. Setelah itu, pengguna akan diminta memasukkan data baru. Data yang diubah akan langsung disimpan, dan sistem akan menampilkan pesan bahwa produk berhasil diubah.

```
//Prosedur
void ubahProduk(Produk produk[], int &jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
    } else {
        tampilkanProduk(produk, jumlahProduk);
        int nomorProduk;

        //Meminta input nomor produk yang valid
        do {
            cout << "Masukkan nomor produk yang ingin diubah : ";
            cin >> nomorProduk;
            cin.ignore();

            if (nomorProduk < 1 || nomorProduk > jumlahProduk) {
                cout << "Nomor produk tidak valid\n";
            }
        } while (nomorProduk < 1 || nomorProduk > jumlahProduk);

        nomorProduk--;
        cout << "Masukkan nama baru : ";
        getline(cin, produk[nomorProduk].nama);
        cout << "Masukkan stok baru : ";
        cin >> produk[nomorProduk].stok;
        cout << "Masukkan harga baru : ";
        cin >> produk[nomorProduk].harga;
        cin.ignore();

        for (int k = 0; k < 3; k++) {
            cout << "Warna standar ke-" << k+1 << ": ";
            getline(cin, produk[nomorProduk].warna.standar[k]);
            cout << "Warna premium ke-" << k+1 << ": ";
            getline(cin, produk[nomorProduk].warna.premium[k]);
        }
        cout << "Produk berhasil diubah.\n";
    }
}
```

### 3.8 Menu Hapus Produk

Prosedur hapusProduk digunakan untuk menghapus produk dari daftar. Prosesnya diawali dengan menampilkan semua produk, lalu pengguna diminta memilih nomor produk yang akan dihapus. Setelah nomor produk dipastikan valid, sistem akan menggeser data produk berikutnya ke posisi sebelumnya untuk menutupi celah data yang dihapus, kemudian mengurangi jumlah total produk. Jika berhasil, sistem menampilkan pesan bahwa produk telah dihapus.

```
//Prosedur
void hapusProduk(Produk produk[], int &jumlahProduk) {
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
    } else {
        tampilkanProduk(produk, jumlahProduk);
        int nomorProduk;

        //Meminta input nomor produk yang valid
        do {
            cout << "Masukkan nomor produk yang ingin diubah : ";
            cin >> nomorProduk;
            cin.ignore();

            if (nomorProduk < 1 || nomorProduk > jumlahProduk) {
                cout << "Nomor produk tidak valid\n";
            }
        } while (nomorProduk < 1 || nomorProduk > jumlahProduk); //Loop jika
nomor tidak valid

        for (int i = nomorProduk - 1; i < jumlahProduk - 1; i++) {
            produk[i] = produk[i + 1];
        }
        jumlahProduk--;
        cout << "Produk berhasil dihapus.\n";
    }
}
```

## 4. Uji Coba dan Hasil Output

```
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|            1. Registrasi            |
|            2. Login                 |
|            3. Keluar                |
=======================================
Masukkan pilihan : 1
Masukkan Nama : NIKY JENITA PUTRI
Masukkan NIM : 2409106019
Registrasi berhasil
```

Gambar 4.1 Registrasi

```
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|            1. Registrasi            |
|            2. Login                 |
|            3. Keluar                |
=======================================
Masukkan pilihan : 2
Masukkan Nama : NIKY JENITA PUTRI
Masukkan NIM : 2409106019
Login berhasil
```

Gambar 4.2 Login

```
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|           1. Tambah Produk          |
|           2. Tampilkan Produk       |
|           3. Ubah Produk            |
|           4. Hapus Produk           |
|           5. Keluar                 |
=======================================
Masukkan pilihan : 1
Masukkan nama produk : pashmina
Masukkan stok : 1000
Masukkan harga : 35900
Warna standar ke-1: hijau
Warna premium ke-1: merah
Warna standar ke-2: putih
Warna premium ke-2: biru
Warna standar ke-3: cream
Warna premium ke-3: kuning
Produk berhasil ditambahkan.
```

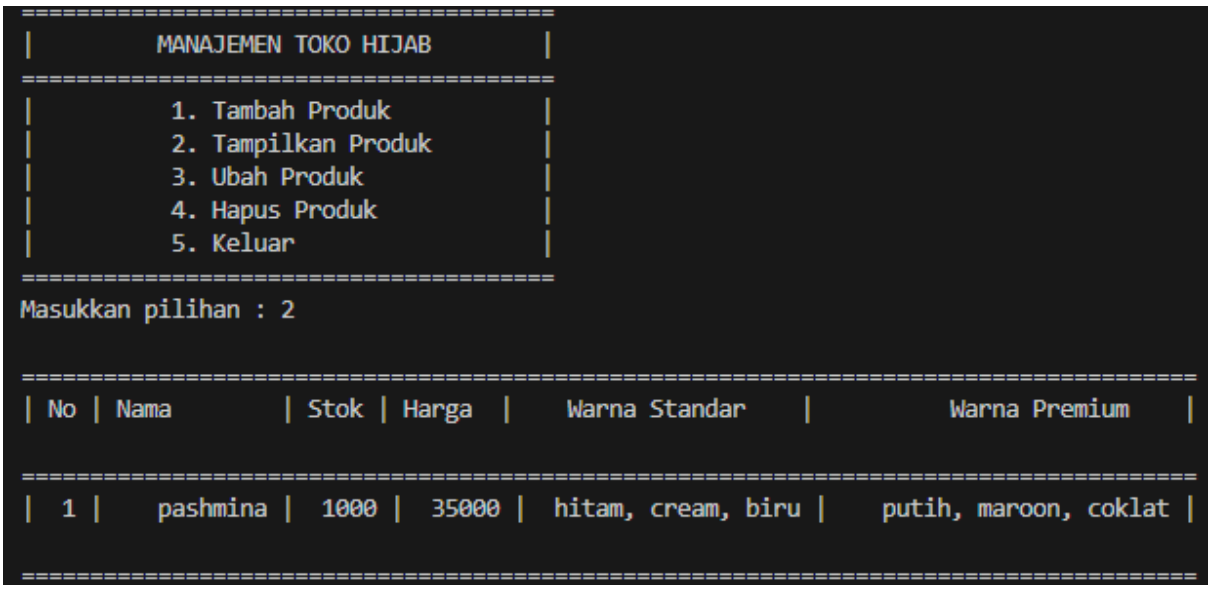
Gambar 4.3 Tambah Produk

```
=====================================
|         MANAJEMEN TOKO HIJAB       |
=====================================
|          1. Tambah Produk          |
|          2. Tampilkan Produk       |
|          3. Ubah Produk            |
|          4. Hapus Produk           |
|          5. Keluar                 |
=====================================
Masukkan pilihan : 2

=============================================================================================
| No | Nama        | Stok | Harga  |    Warna Standar    |          Warna Premium    |
=============================================================================================
|  1 |    pashmina |  1000 |  35900 | hijau, putih, cream |      merah, biru, kuning |
=============================================================================================
```

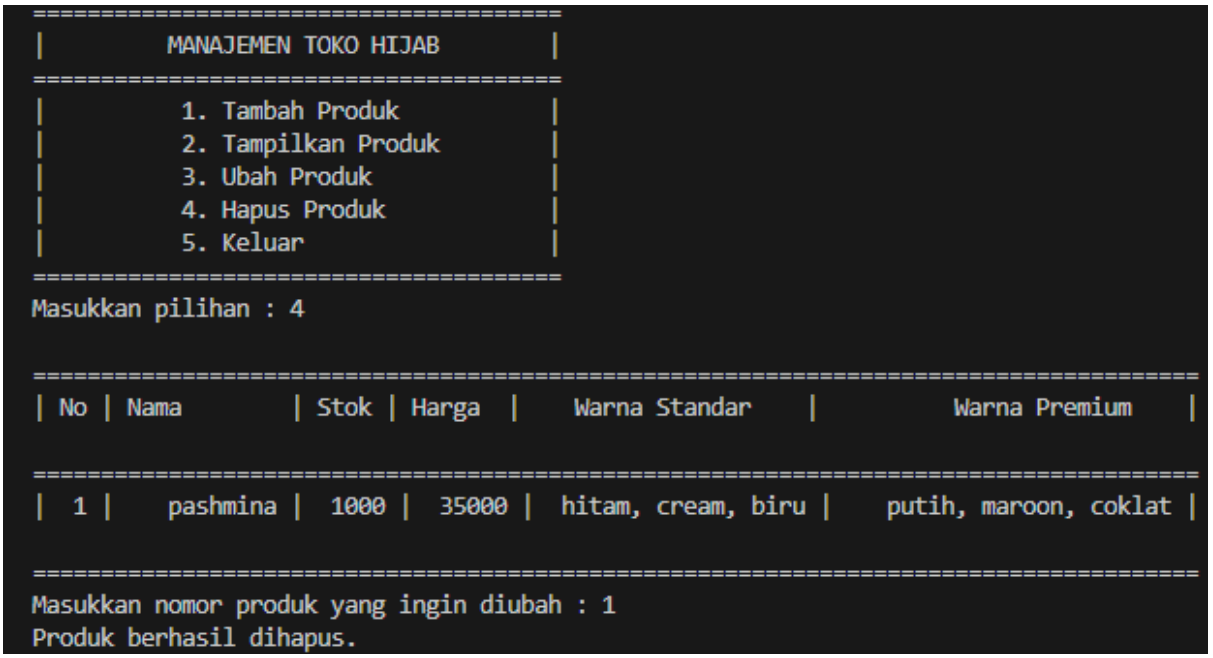
Gambar 4.4 Tampilkan Produk

```
=====================================
|         MANAJEMEN TOKO HIJAB       |
=====================================
|          1. Tambah Produk          |
|          2. Tampilkan Produk       |
|          3. Ubah Produk            |
|          4. Hapus Produk           |
|          5. Keluar                 |
=====================================
Masukkan pilihan : 3

=============================================================================================
| No | Nama        | Stok | Harga  |    Warna Standar    |          Warna Premium    |
=============================================================================================
|  1 |    pashmina |  1000 |  35900 | hijau, putih, cream |      merah, biru, kuning |
=============================================================================================
Masukkan nomor produk yang ingin diubah : 1
Masukkan nama baru : pashmina
Masukkan stok baru : 1000
Masukkan harga baru : 35000
Warna standar ke-1: hitam
Warna premium ke-1: putih
Warna standar ke-2: cream
Warna premium ke-2: maroon
Warna standar ke-3: biru
Warna premium ke-3: coklat
Produk berhasil diubah.
```

Gambar 4.5 Ubah Produk




Gambar 4.6 Tampilkan Produk Setelah Diubah




Gambar 4.7 Hapus Produk

```
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|           1. Tambah Produk          |
|           2. Tampilkan Produk       |
|           3. Ubah Produk            |
|           4. Hapus Produk           |
|           5. Keluar                 |
=======================================
Masukkan pilihan : 2

Tidak ada produk tersedia.
```

Gambar 4.8 Tampilkan Produk Setelah Produk Dihapus

```
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|           1. Tambah Produk          |
|           2. Tampilkan Produk       |
|           3. Ubah Produk            |
|           4. Hapus Produk           |
|           5. Keluar                 |
=======================================
Masukkan pilihan : 5
=======================================
|         MANAJEMEN TOKO HIJAB        |
=======================================
|           1. Registrasi             |
|           2. Login                  |
|           3. Keluar                 |
=======================================
Masukkan pilihan : 3
====================
Program berhenti.
====================
PS D:\praktikum-apl\post-test\post-test-apl-4>
```

Gambar 4.9 Keluar dari Program

## 5. Langkah-Langkah Git pada VSCode

### 5.1 Git Init

Git init adalah perintah Git yang digunakan untuk menginisialisasi repository Git baru dalam sebuah direktori. Perintah ini akan membuat subdirektori tersembunyi bernama .git yang berisi semua file dan metadata yang diperlukan untuk melacak perubahan dalam proyek tersebut.

```
PS D:\praktikum-apl> git init
Initialized empty Git repository in D:/praktikum-apl/.git/
```

Gambar 5.1 Git Init

### 5.2 Git Add

Git add . adalah perintah Git yang digunakan untuk menambahkan semua perubahan dalam direktori kerja (working directory) ke dalam staging area. Dengan kata lain, perintah ini akan menandai semua file yang telah dibuat, diubah, atau dihapus agar siap untuk di-commit.

```
PS D:\praktikum-apl> git add .
```

Gambar 5.2 Git Add

### 5.3 Git Commit -m

Git commit -m adalah perintah Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository dengan pesan commit yang diberikan.

```
PS D:\praktikum-apl> git commit -m "fitur selesai"
[main 1440d55] fitur selesai
 2 files changed, 319 insertions(+)
 create mode 100644 post-test/post-test-apl-4/2409106019-NikyJenitaPutri-PT-4.cpp
 create mode 100644 post-test/post-test-apl-4/2409106019-NikyJenitaPutri-PT-4.exe
PS D:\praktikum-apl>
```

Gambar 5.3 Git Commit

### 5.4 Git Remote

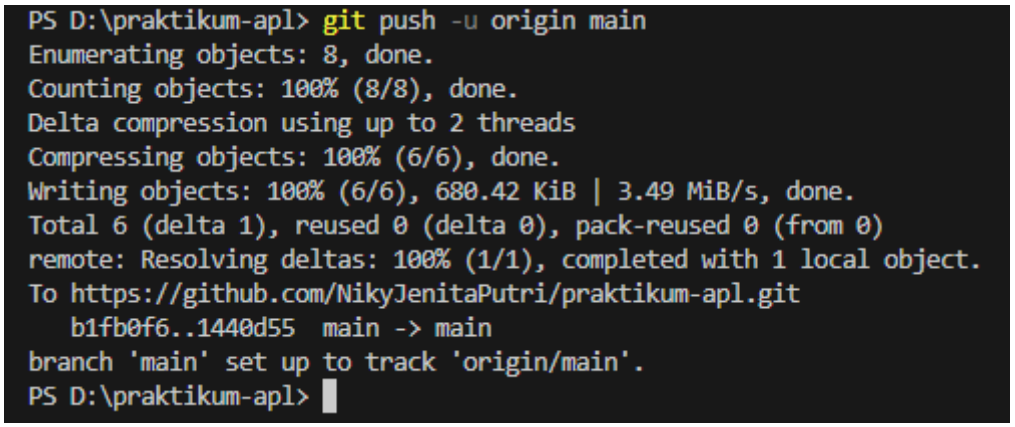
Git remote add origin digunakan untuk menghubungkan repository lokal dengan repository di GitHub.

```
PS D:\praktikum-apl> git remote add origin https://github.com/NikyJenitaPutri/praktikum-apl.git
PS D:\praktikum-apl>
```

Gambar 5.4 Git Remote

### 5.5 Git Push

Git push adalah perintah Git yang digunakan untuk mengunggah (upload) perubahan dari repository lokal ke repository di GitHub. -u origin main berarti mengatur branch main sebagai default untuk push ke origin




Gambar 5.5 Git Push