LAPORAN PRAKTIKUM

POSTTEST 5

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Niky Jenita Putri (2409106019)

Kelas (A1 ‘24)

PROGRAM STUDI INFORMATIKA

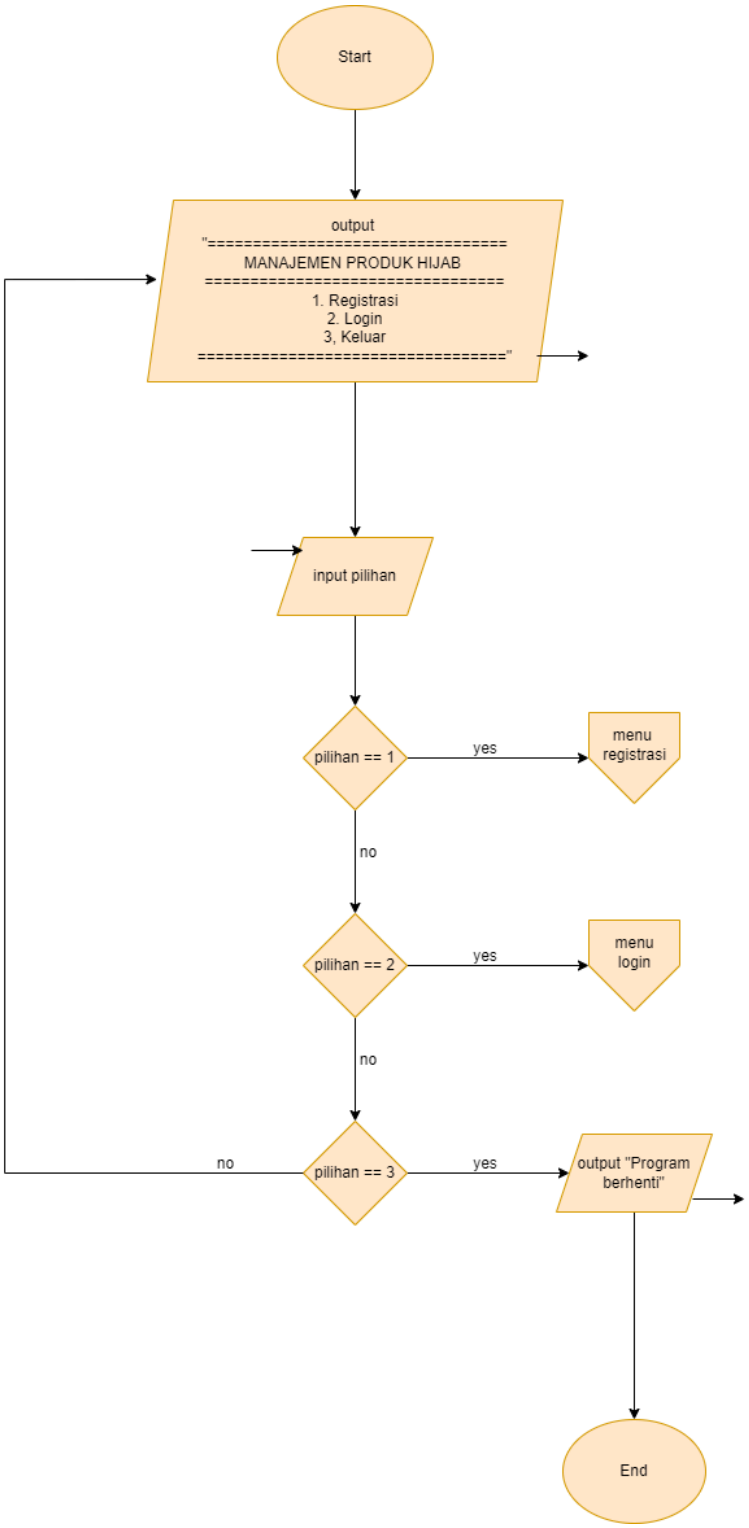
UNIVERSITAS MULAWARMAN

SAMARINDA

2025

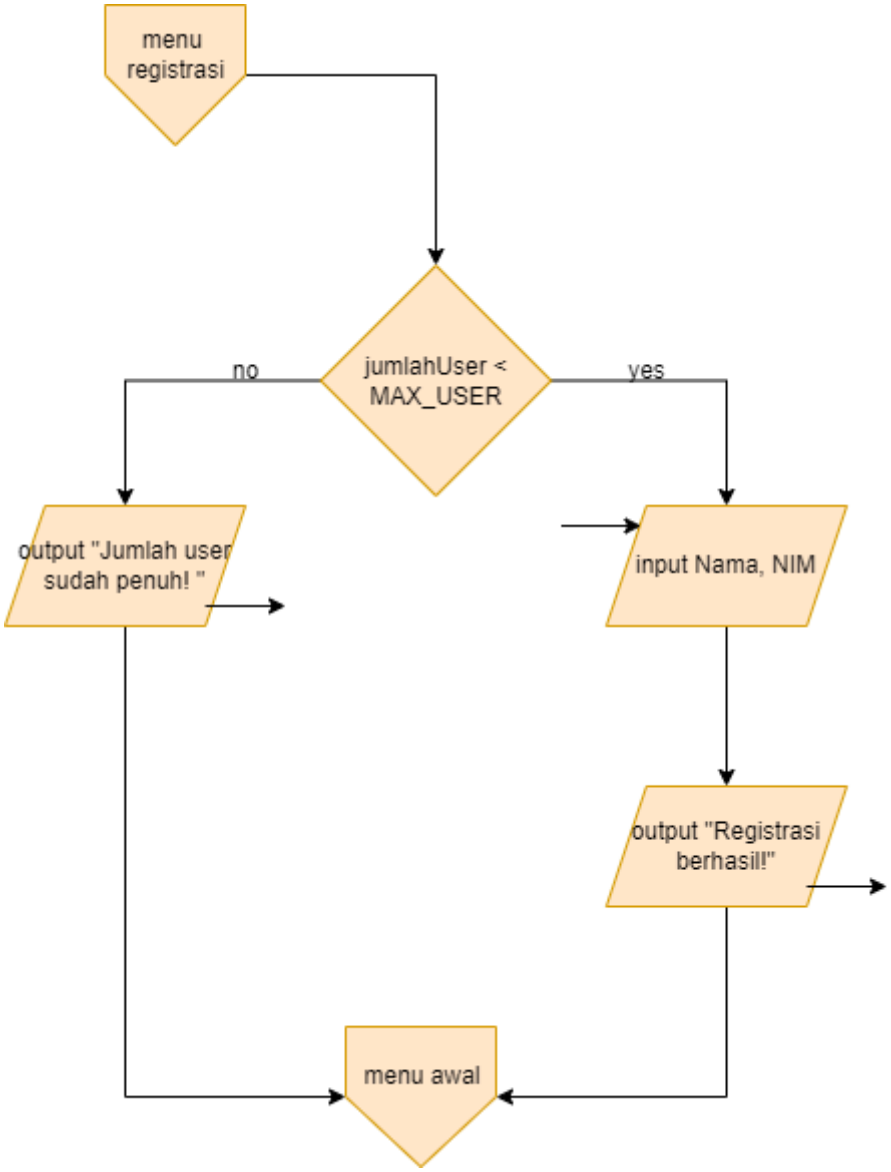
1. Flowchart

1.1 Menu Awal



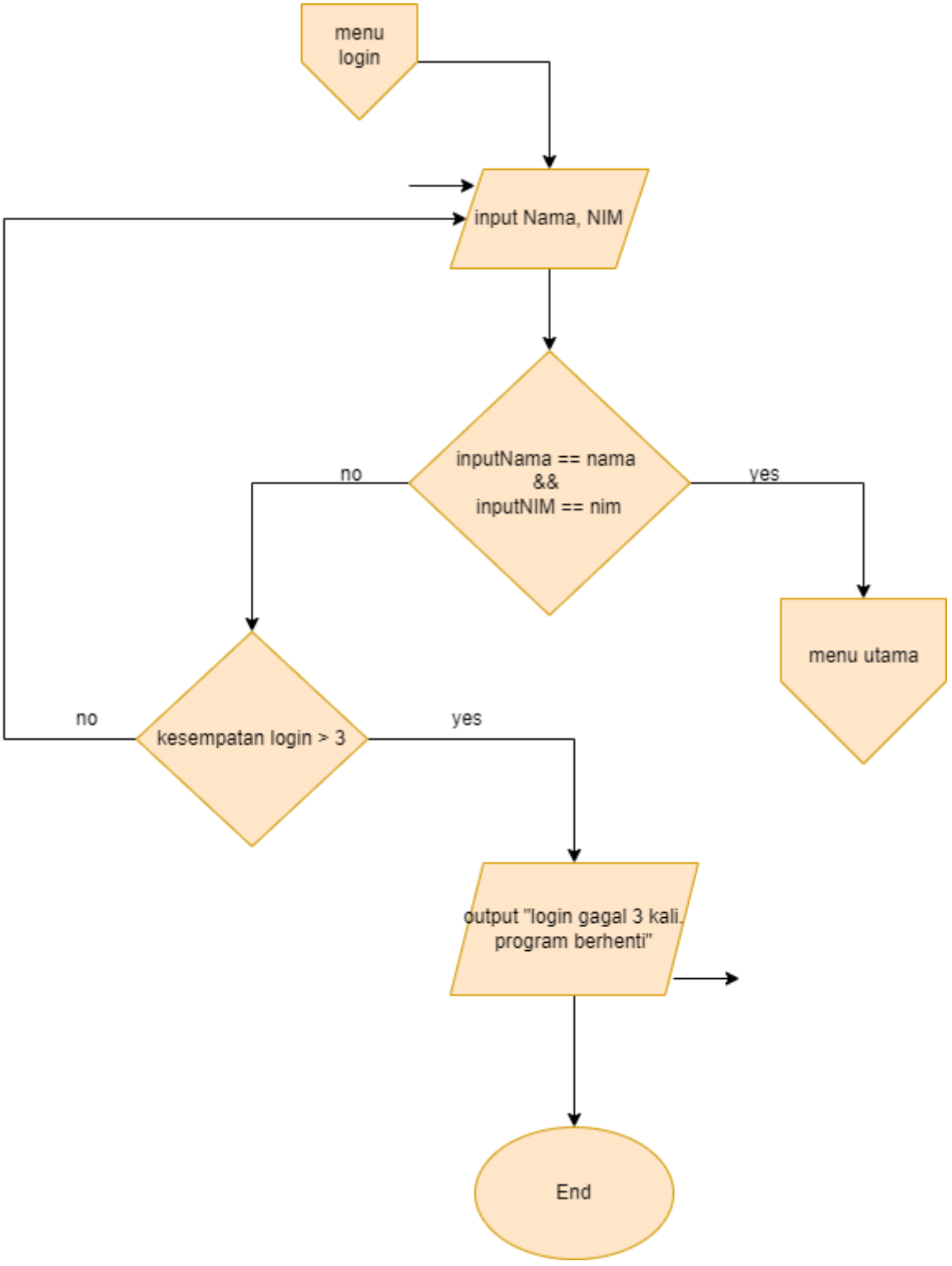
Gambar 1.1 Menu Awal

1.2 Menu Registrasi



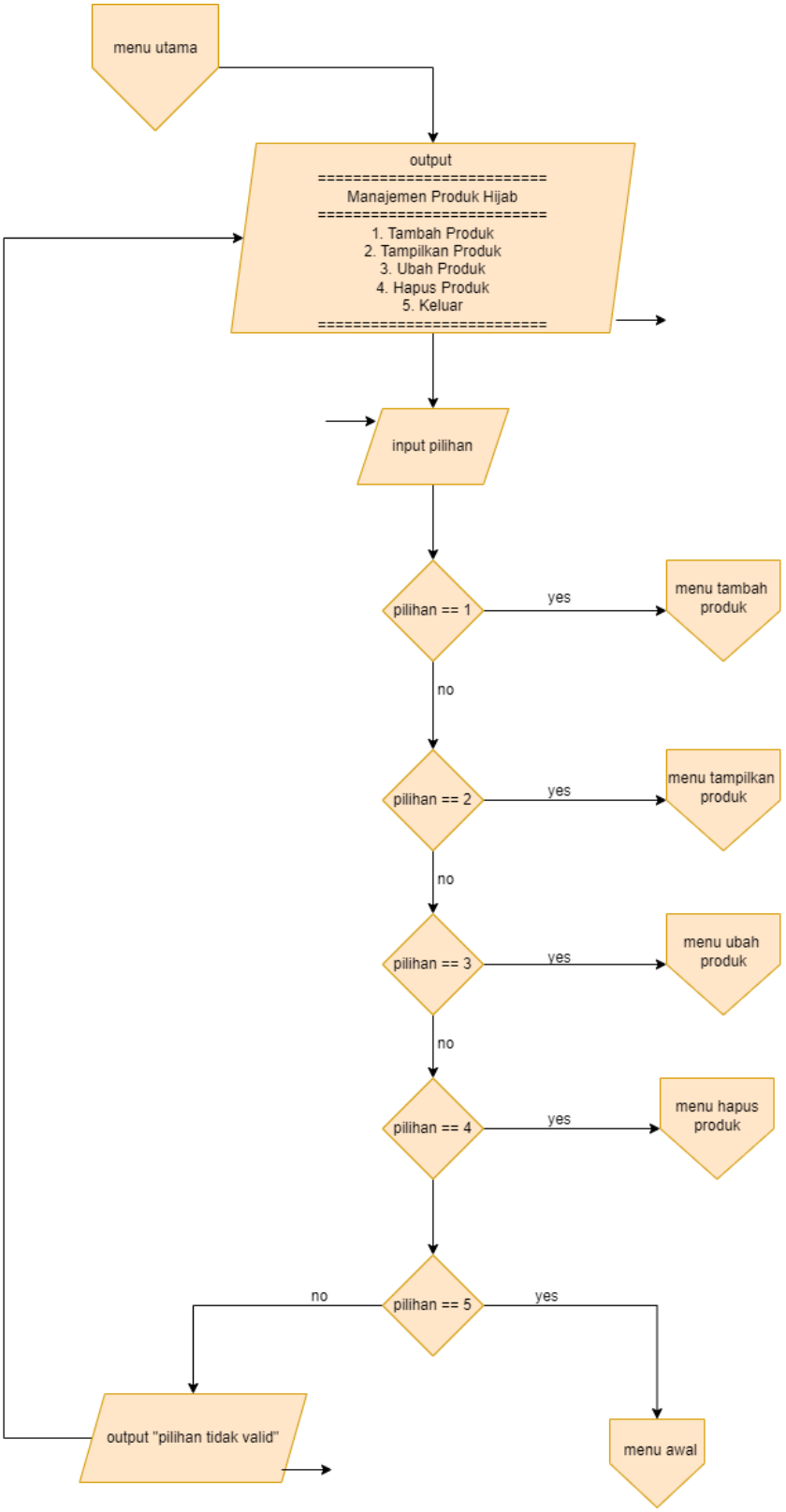
Gambar 1.2 Menu Registrasi

1.3 Menu Login



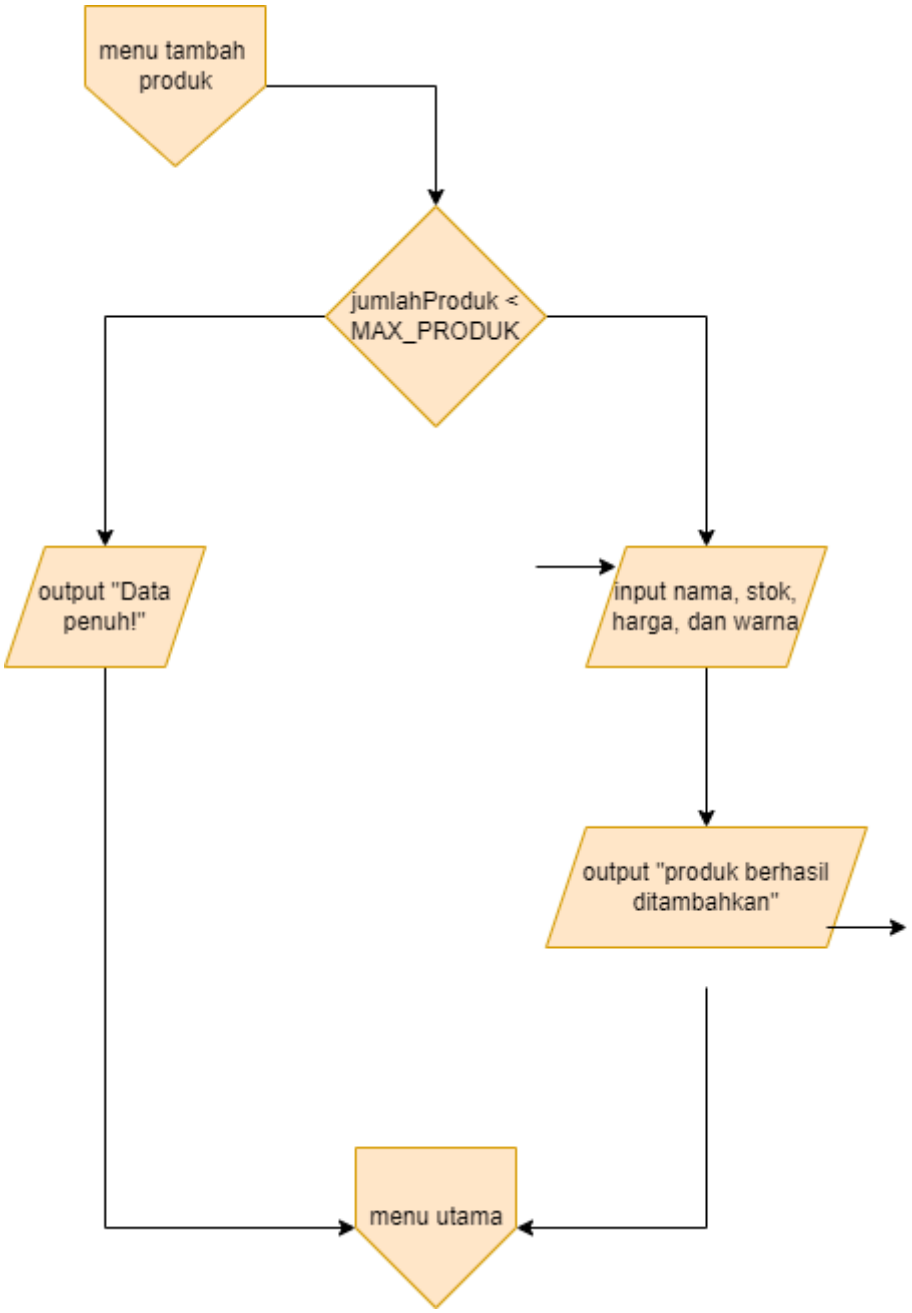
Gambar 1.3 Menu Login

1.4 Menu Utama



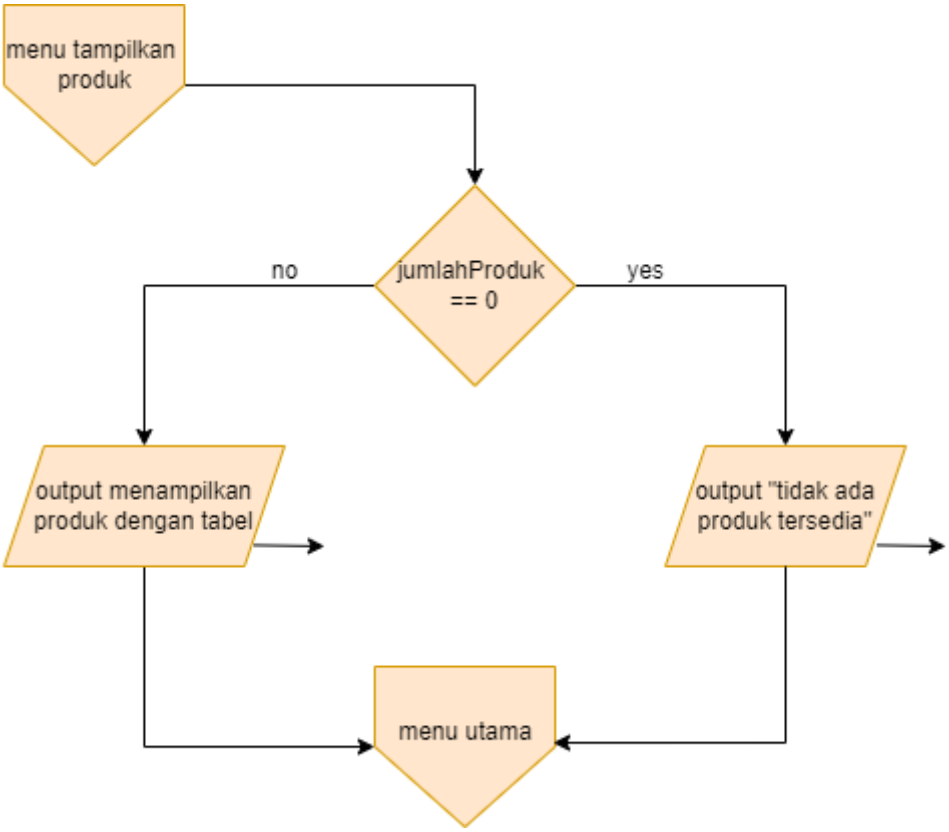
Gambar 1.4 Menu Utama

1.5 Menu Tambah Produk



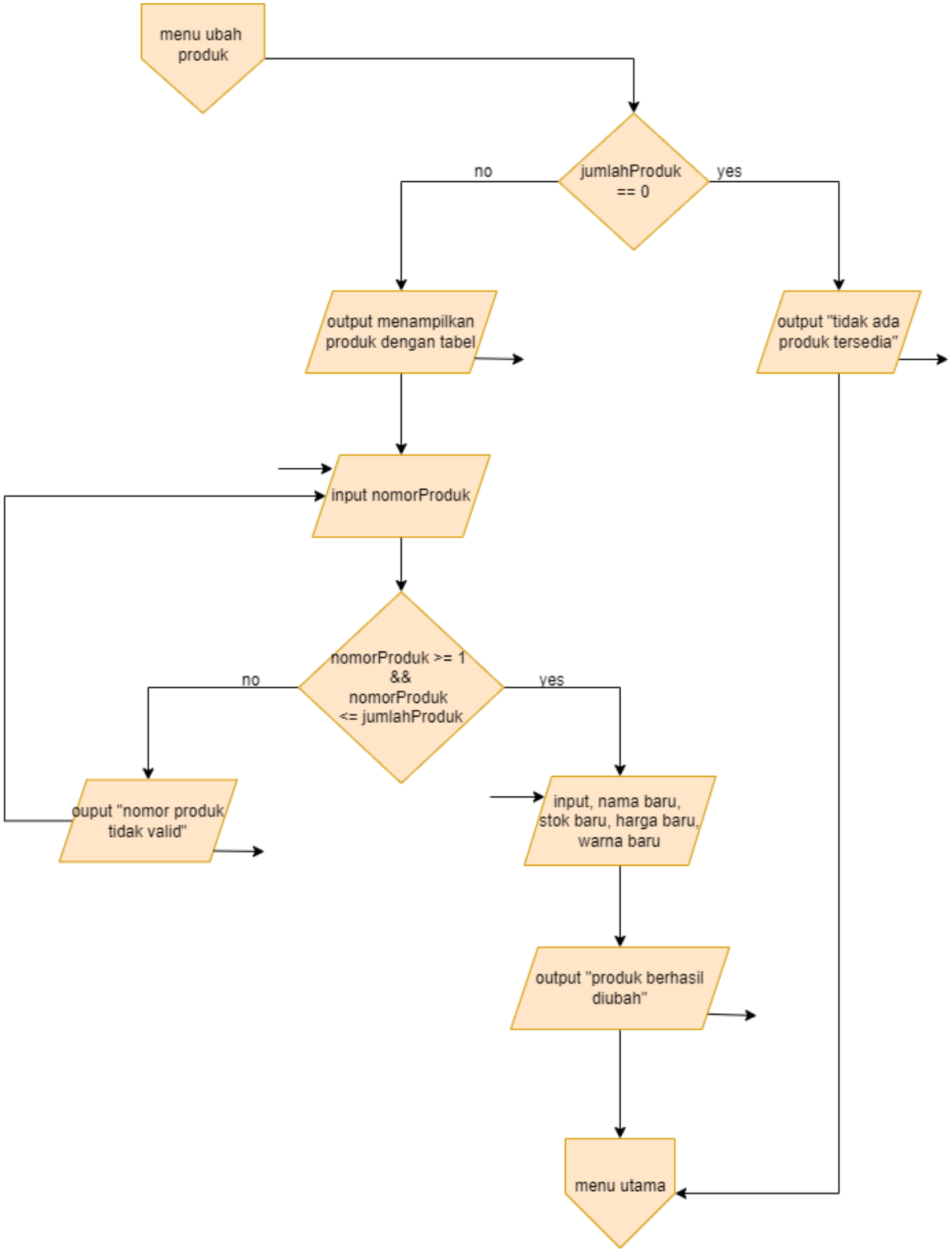
Gambar 1.5 Menu Tambah Produk

1.6 Menu Tampilkan Produk



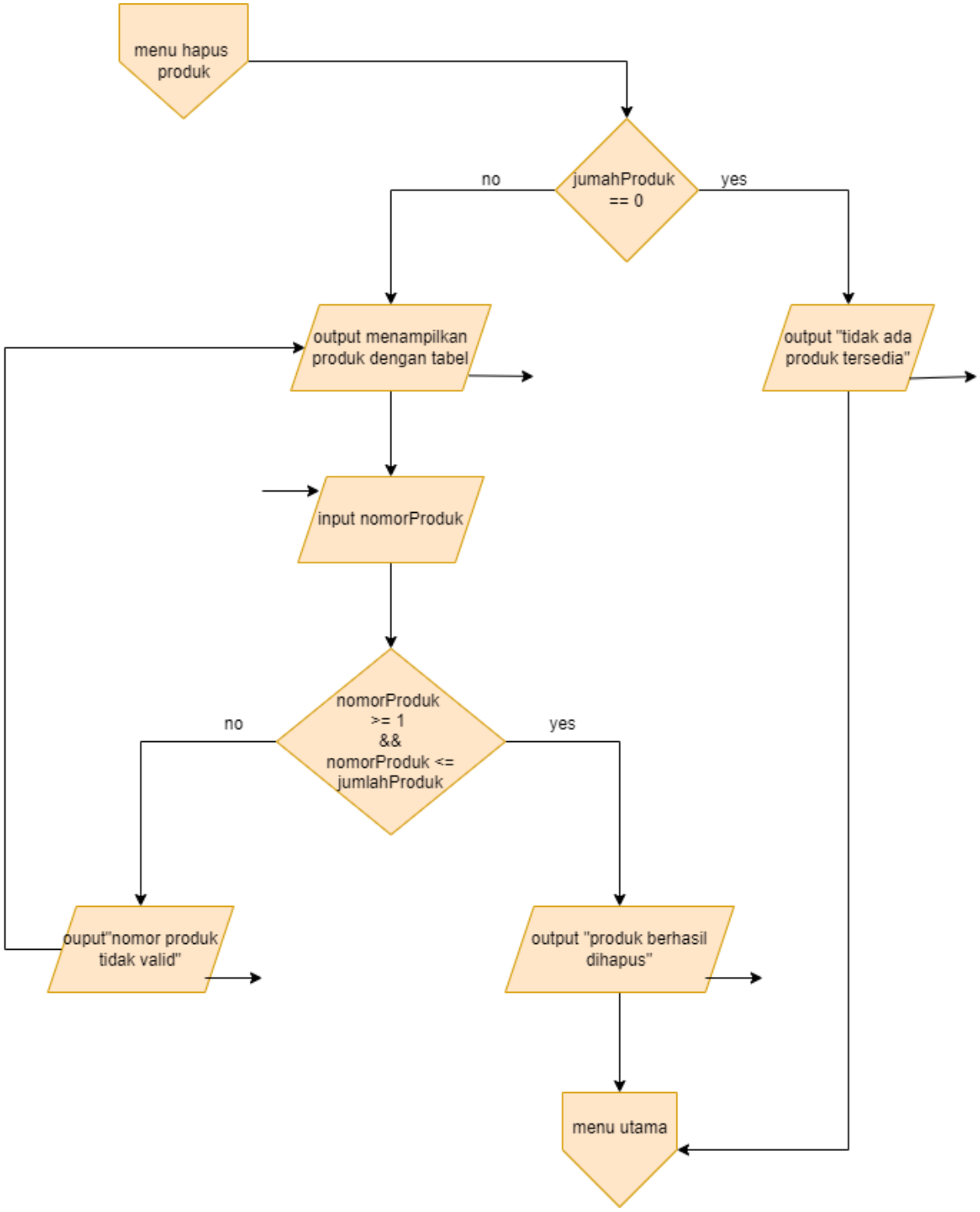
Gambar 1.6 Menu Tampilkan Produk

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 int registrasiUser(Data *data) adalah fungsi yang menggunakan parameter dereference, karena menerima parameter berupa pointer (Data *data) dan mengakses nilainya dengan operator ->. Fungsi ini bertugas untuk menambahkan user baru ke dalam array users jika jumlah user saat ini (data->jumlahUser) masih kurang dari MAX_USER. Di dalamnya, nama dan NIM pengguna diminta melalui input, lalu disimpan ke array users menggunakan dereferensi pointer. Setelah data disimpan, nilai jumlahUser akan ditambah satu dan fungsi mengembalikan indeks user yang baru ditambahkan. Fungsi ini menunjukkan penggunaan prinsip pointer dengan jelas melalui akses data menggunakan ->.

```
int registrasiUser(Data *data) {
    if (data ->jumlahUser < MAX_USER) {
        cout << "Masukkan Nama : ";
        getline(cin, data ->users[data ->jumlahUser].nama);
        cout << "Masukkan NIM : ";
        getline(cin, data ->users[data ->jumlahUser].nim);
        data ->jumlahUser++;
        return data ->jumlahUser - 1;
        cout << "Registrasi berhasil" << endl;
    } else {
        cout << "Jumlah user sudah penuh!\n";
        return -1;
    }
}
```

3.3 Menu Login

Fungsi loginUser(Data *data, int loginAttempts) menggunakan parameter dereference karena menerima pointer data dan mengaksesnya dengan ->. Fungsi ini juga merupakan fungsi rekursif karena memanggil dirinya sendiri saat login gagal.

```
int loginUser(Data *data, int loginAttempts) {
    if (loginAttempts == MAX_LOGIN_ATTEMPTS) {
        cout << "Login gagal 3 kali. Program berhenti.\n";
        exit(0);
    }

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

    for (int i = 0; i < data ->jumlahUser; i++) {
        if(data ->users[i].nama == inputNama && data ->users[i].nim == inputNIM)
{
        cout << "Login berhasil\n";
        return i;
        }
    }

    cout << "Login gagal. Coba Lagi,\n";
    return loginUser(data, loginAttempts + 1);
}
```

3.4 Menu Utama

Fungsi menuUser(Data *data, int userIndex) menggunakan parameter dereference pada data untuk mengakses struct Data dengan menggunakan operator *data. Fungsi ini menampilkan menu untuk user dan memungkinkan mereka memilih opsi seperti menambah, menampilkan, mengubah, atau menghapus produk. Setiap pilihan akan memanggil fungsi lain yang menggunakan dereference pointer seperti tambahProduk(*data) dan tampilkanProduk(*data).

```
void menuUser (Data *data, int userIndex) {
    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(*data);
        } else if (pilihan == 2) {
            tampilkanProduk(*data);
        } else if (pilihan == 3) {
            ubahProduk(data);
        } else if (pilihan == 4) {
            hapusProduk(data);
        } else if (pilihan == 5) {
            userIndex = -1;
        }
    }
}
```

3.5 Menu Tambah Produk

Fungsi tambahProduk(Data &data) dan tambahProduk(Data &data, string nama, int stok, int harga, Warna warna) menerapkan prinsip pointer dengan menggunakan parameter address-of (referensi &data). Hal ini memungkinkan fungsi untuk memodifikasi data asli tanpa membuat salinan. Fungsi pertama menggunakan referensi untuk mengakses dan memodifikasi produk, sedangkan fungsi kedua juga menggunakan referensi untuk menambah produk baru dengan parameter tambahan.

```
bool tambahProduk(Data &data) {
    if (data.jumlahProduk < MAX_PRODUK) {
        Produk &p = data.produk[data.jumlahProduk];
        cout << "Masukkan nama produk : ";
        getline(cin, p.nama);
        cout << "Masukkan stok : ";
        cin >> p.stok;
        cout << "Masukkan harga : ";
        cin >> p.harga;
        cin.ignore();
        for (int i = 0; i < 3; i++) {
            cout << "Warna standar ke-" << i+1 << ": ";
            getline(cin, p.warna.standar[i]);
            cout << "Warna premium ke-" << i+1 << ": ";
            getline(cin, p.warna.premium[i]);
        }
        data.jumlahProduk++;
        cout << "Produk berhasil ditambahkan.\n";
```

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

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

}
```

3.6 Menu Tampilkan Produk

Fungsi tampilkanProduk(Data &data) menampilkan semua produk yang ada, sedangkan tampilkanProduk(Data &data, int minHarga) menampilkan produk dengan harga di atas nilai minHarga.Kedua fungsi ini menggunakan parameter address-of pada data yang memungkinkan mereka untuk mengubah dan mengakses data produk tanpa membuat salinan.

```
bool tampilkanProduk(Data &data) {
    if (data.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 < data.jumlahProduk; i++) {
            Produk &p = data.produk[i];
            cout << "| " << setw(2) << i+1 << " |"
                 << setw(12) << p.nama << " |"
                 << setw(6) << p.stok << " |"
                 << setw(7) << p.harga << " |"
                 << setw(20) << p.warna.standar[0] +", " + p.warna.standar[1] +
", " + p.warna.standar[2] << " |"
                 << setw(25) << p.warna.premium[0] +", " + p.warna.premium[1] +
", " + p.warna.premium[2] << " |\n";
        }
```

```
        cout <<
"\n=================================================================================
=========\n";
        return true;
    }
}

bool tampilkanProduk(Data &data, int minHarga) {
    bool found = false;
    if (data.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 < data.jumlahProduk; i++) {
            Produk &p = data.produk[i];
            if (p.harga >= minHarga) {
                cout << "| " << setw(2) << i+1 << " |"
                     << setw(12) << p.nama << " |"
                     << setw(6) << p.stok << " |"
                     << setw(7) << p.harga << " |"
                     << setw(20) << p.warna.standar[0] +", " +
p.warna.standar[1] + ", " + p.warna.standar[2] << " |"
                     << setw(25) << p.warna.premium[0] +", " +
p.warna.premium[1] + ", " + p.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

Fungsi ubahProduk digunakan untuk mengubah data produk pada program manajemen toko. Fungsi ini menerima parameter pointer ke Data, yang berisi daftar produk. Jika tidak ada produk, tampil pesan kesalahan. Jika ada, tampilkan daftar dan minta pengguna memilih produk yang valid. Setelah itu, pengguna memasukkan nama, stok, harga, serta 3 warna standar dan premium yang baru. Terakhir, tampilkan pesan bahwa produk berhasil diubah.

```
void ubahProduk(Data *data) {
    if (data->jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        return;
    } else {
        tampilkanProduk(*data);
        int nomorProduk;

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

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

        Produk &p = data->produk[nomorProduk - 1];
        cout << "Masukkan nama baru : ";
        getline(cin, p.nama);
        cout << "Masukkan stok baru : ";
        cin >> p.stok;
        cout << "Masukkan harga baru : ";
        cin >> p.harga;
        cin.ignore();

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

3.8 Menu Hapus Produk

Fungsi hapusProduk memakai pointer Data *data agar bisa langsung mengubah data asli. Produk dihapus dengan menggeser elemen array, lalu jumlahProduk dikurangi. Karena pakai pointer, perubahan langsung tersimpan di data utama tanpa perlu return.

```
void hapusProduk(Data *data) {
    if (data->jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        return;
    } else {
        tampilkanProduk(*data);
        int nomorProduk;

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

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

        for (int i = nomorProduk - 1; i < data->jumlahProduk - 1; i++) {
            data->produk[i] = data->produk[i + 1];
        }
        data->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.
```

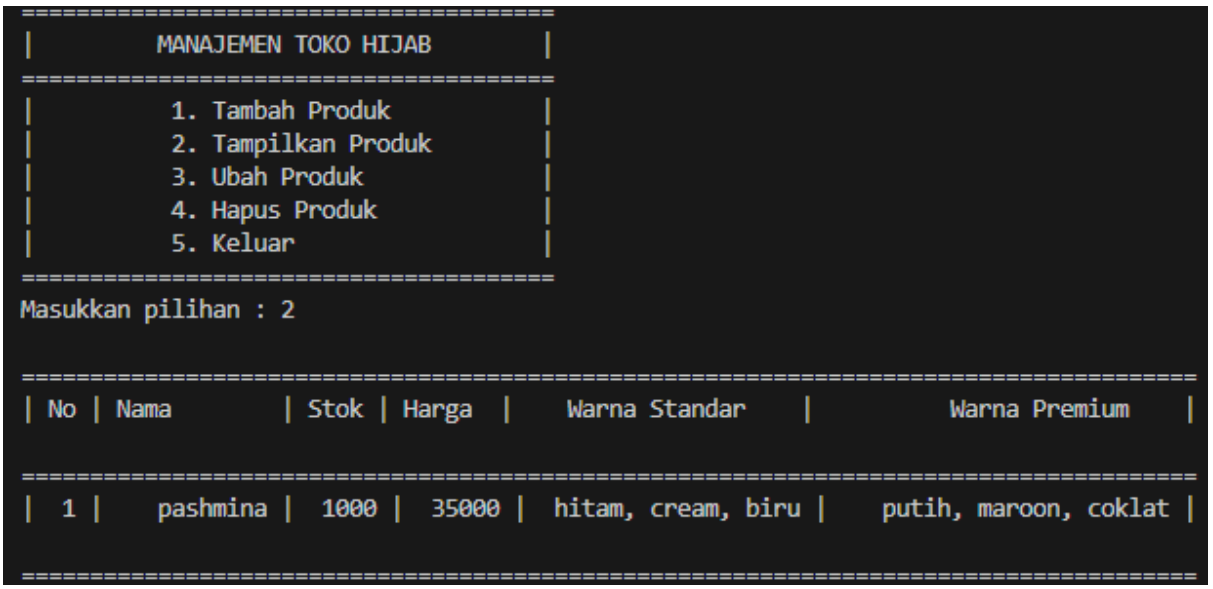
Gambar 4.3 Tambah Produk

```
=====================================
|         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 |
=============================================================================================
```

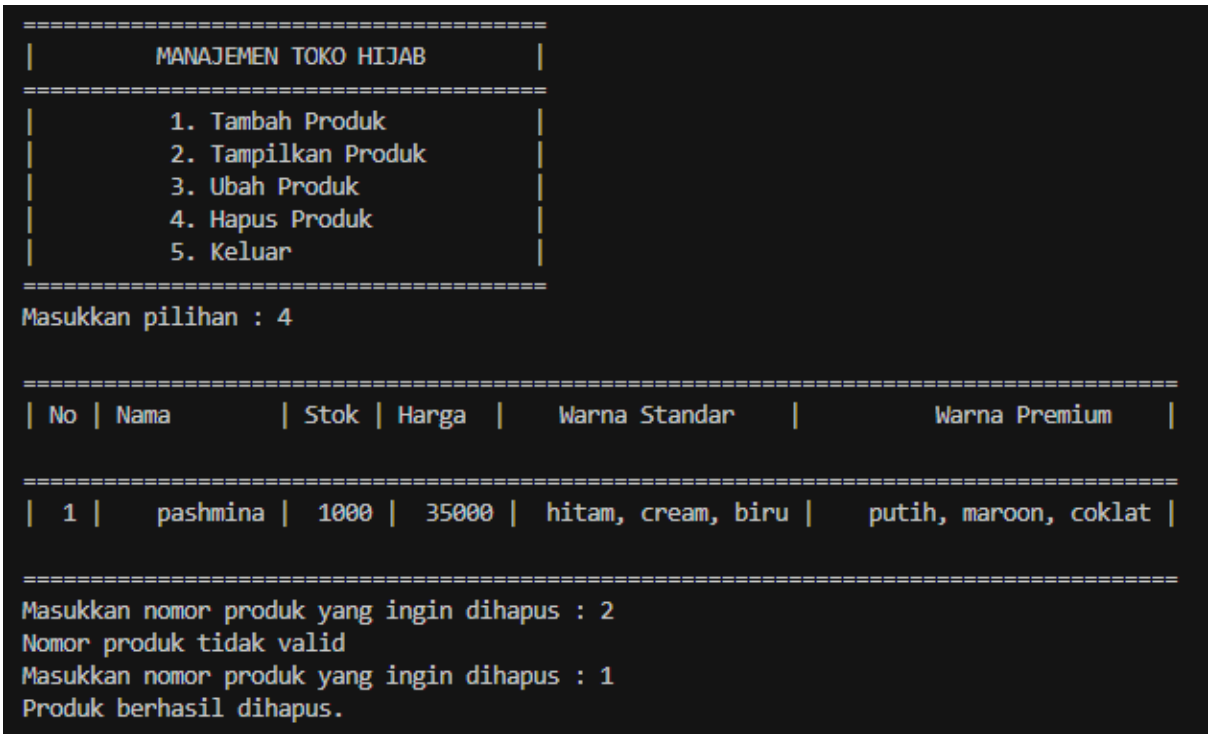
Gambar 4.4 Tampilkan Produk

```
=====================================
|         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 : 3
Nomor produk tidak valid
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 "program selesai"
[main fb9103e] program selesai
 2 files changed, 319 insertions(+)
 create mode 100644 post-test/post-test-apl-5/2409106019-NikyJenitaPutri-PT-5.cpp
 create mode 100644 post-test/post-test-apl-5/2409106019-NikyJenitaPutri-PT-5.exe
PS D:\praktikum-apl>
```

Gambar 5.3 Git Commit

5.4 Git Remote

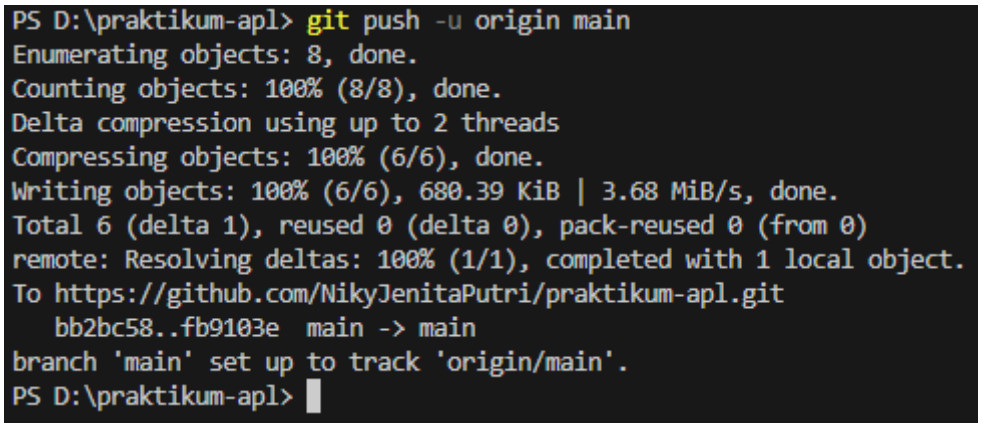
Git remote add origin digunakan untuk menghubungkan repository lokal dengan repository di GitHub.

```
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