

LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:

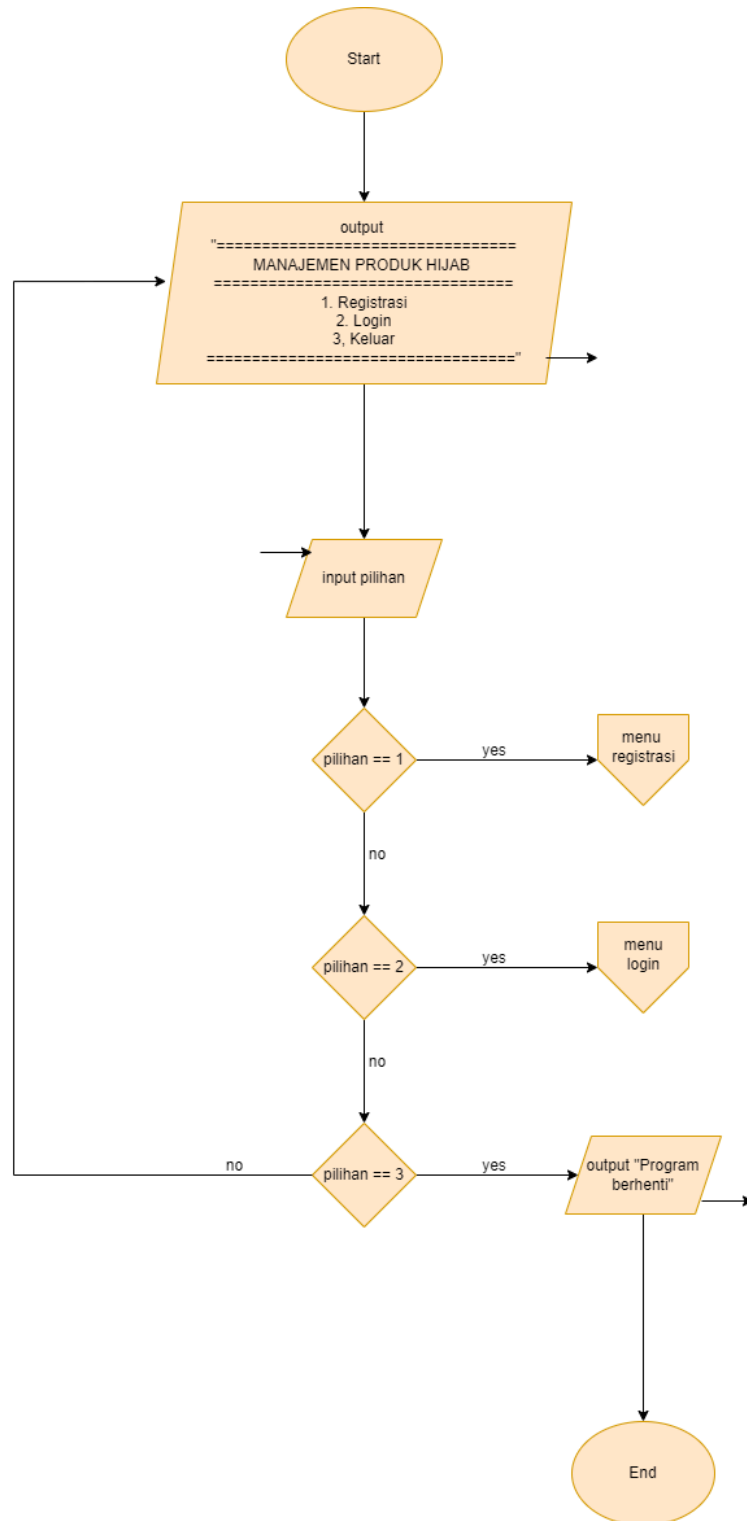
Nama (2409106019)

Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

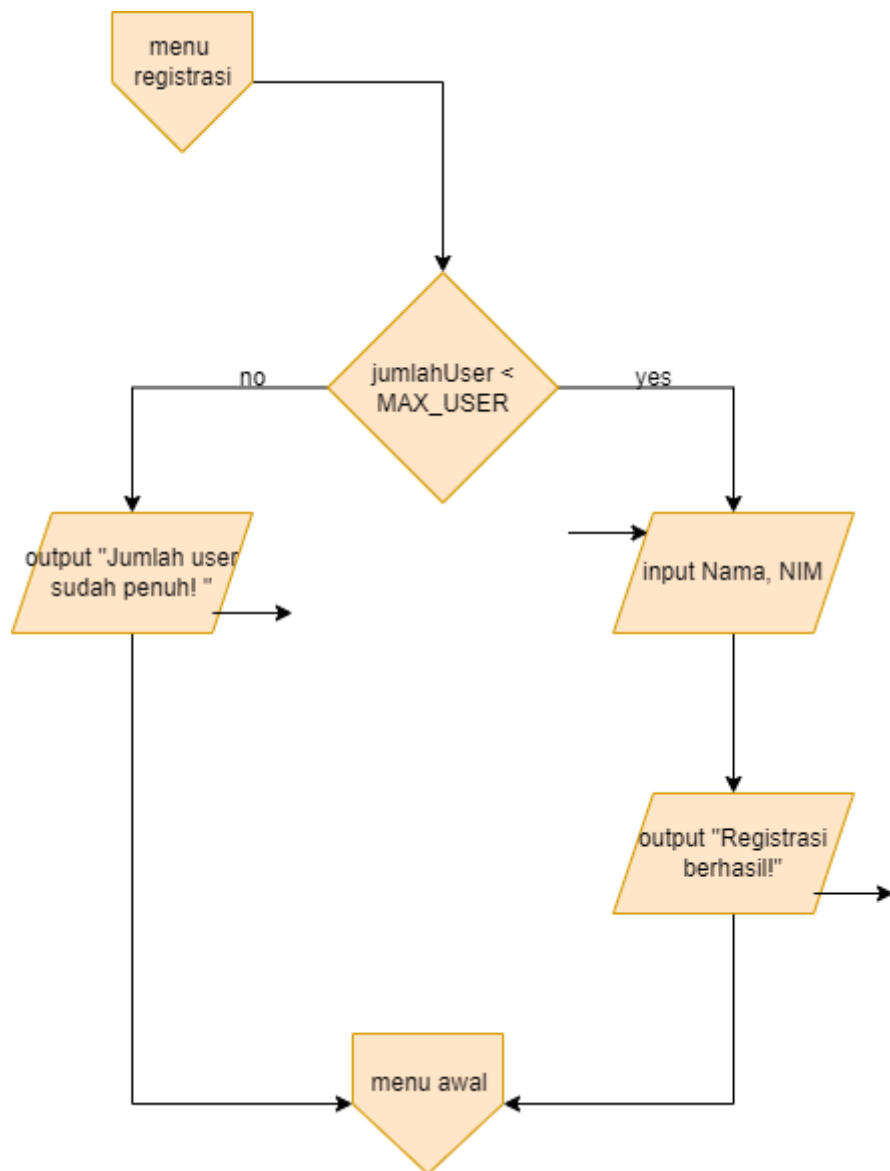
1. Flowchart

1.1 Menu Awal



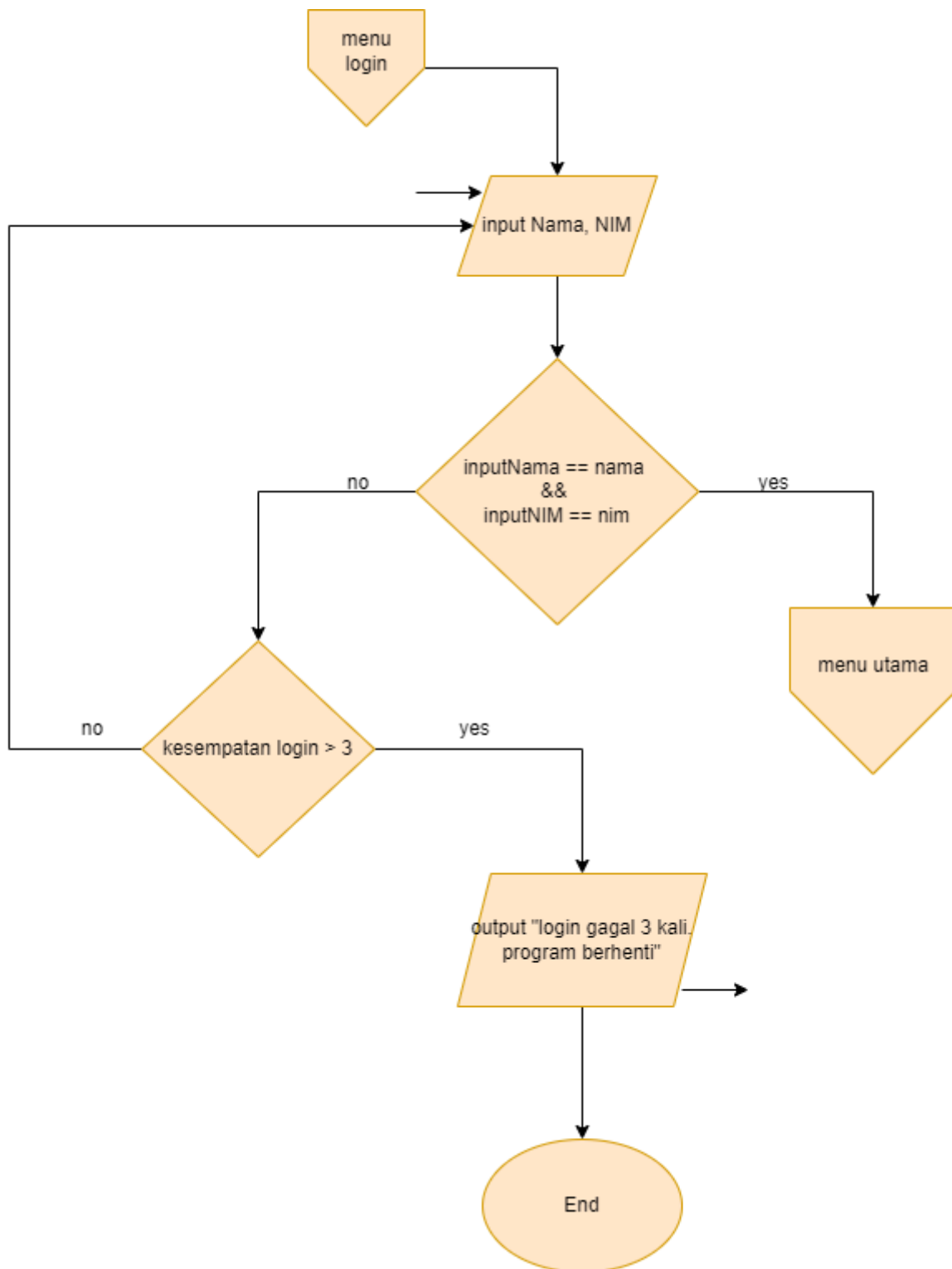
Gambar 1.1 Menu Awal

1.2 Menu Registrasi



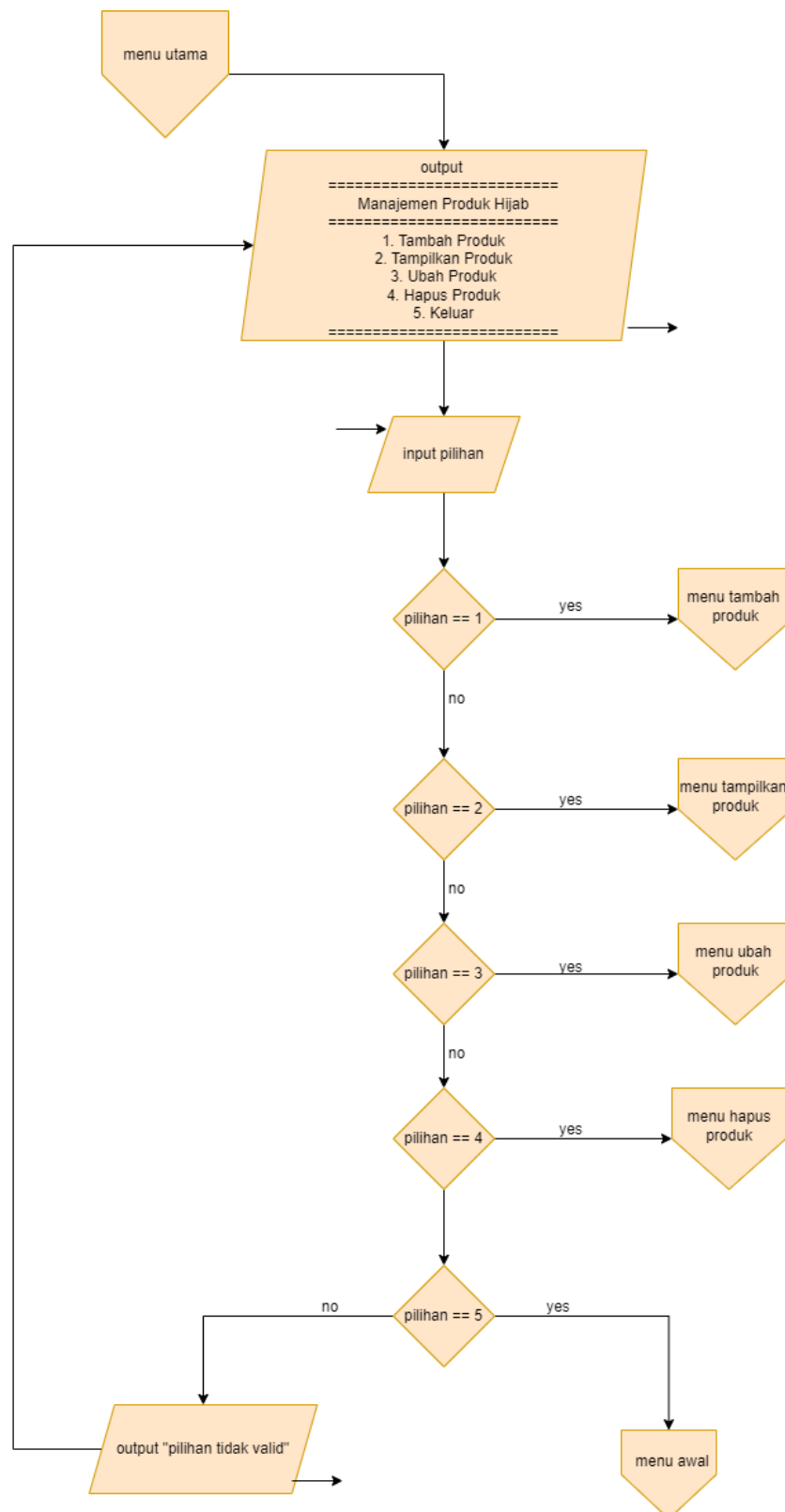
Gambar 1.2 Menu Registrasi

1.3 Menu Login



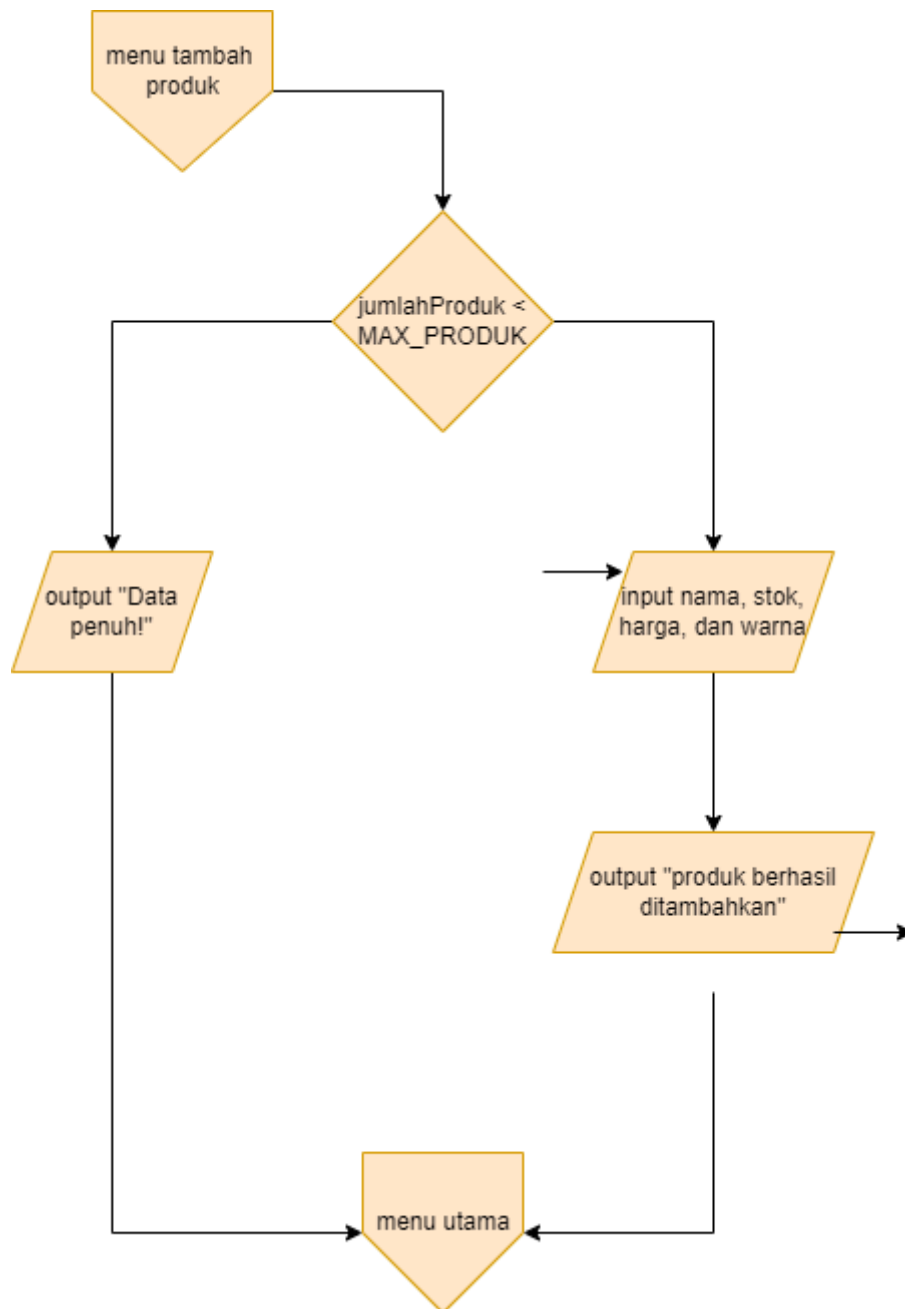
Gambar 1.3 Menu Login

1.4 Menu Utama



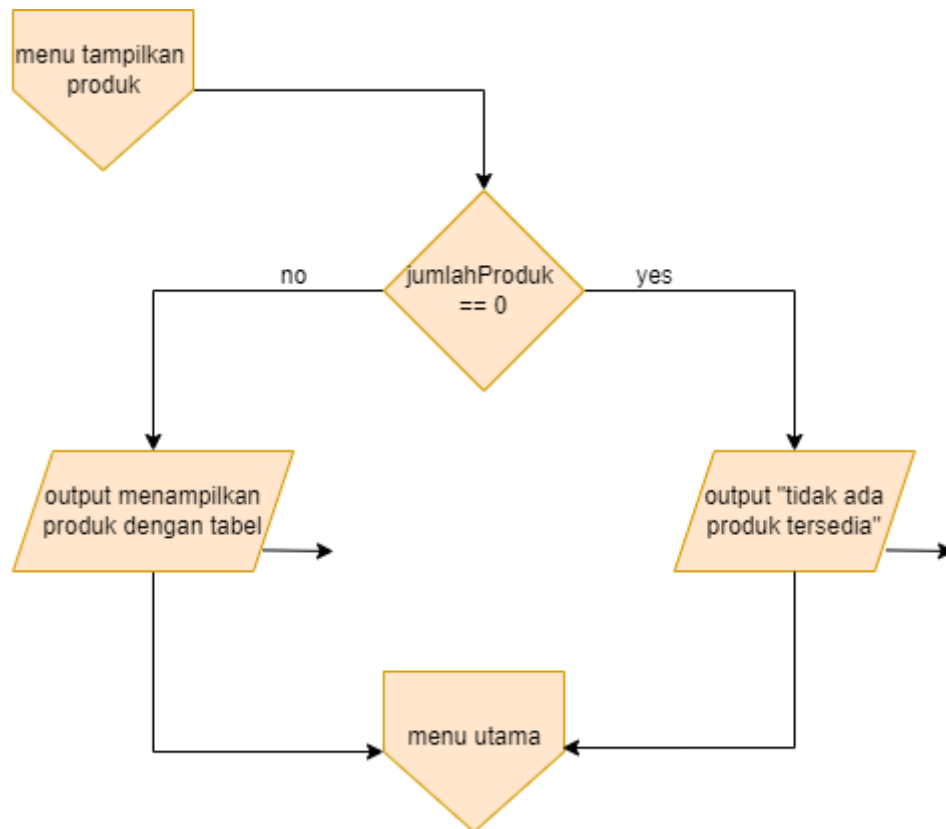
Gambar 1.4 Menu Utama

1.5 Menu Tambah Produk



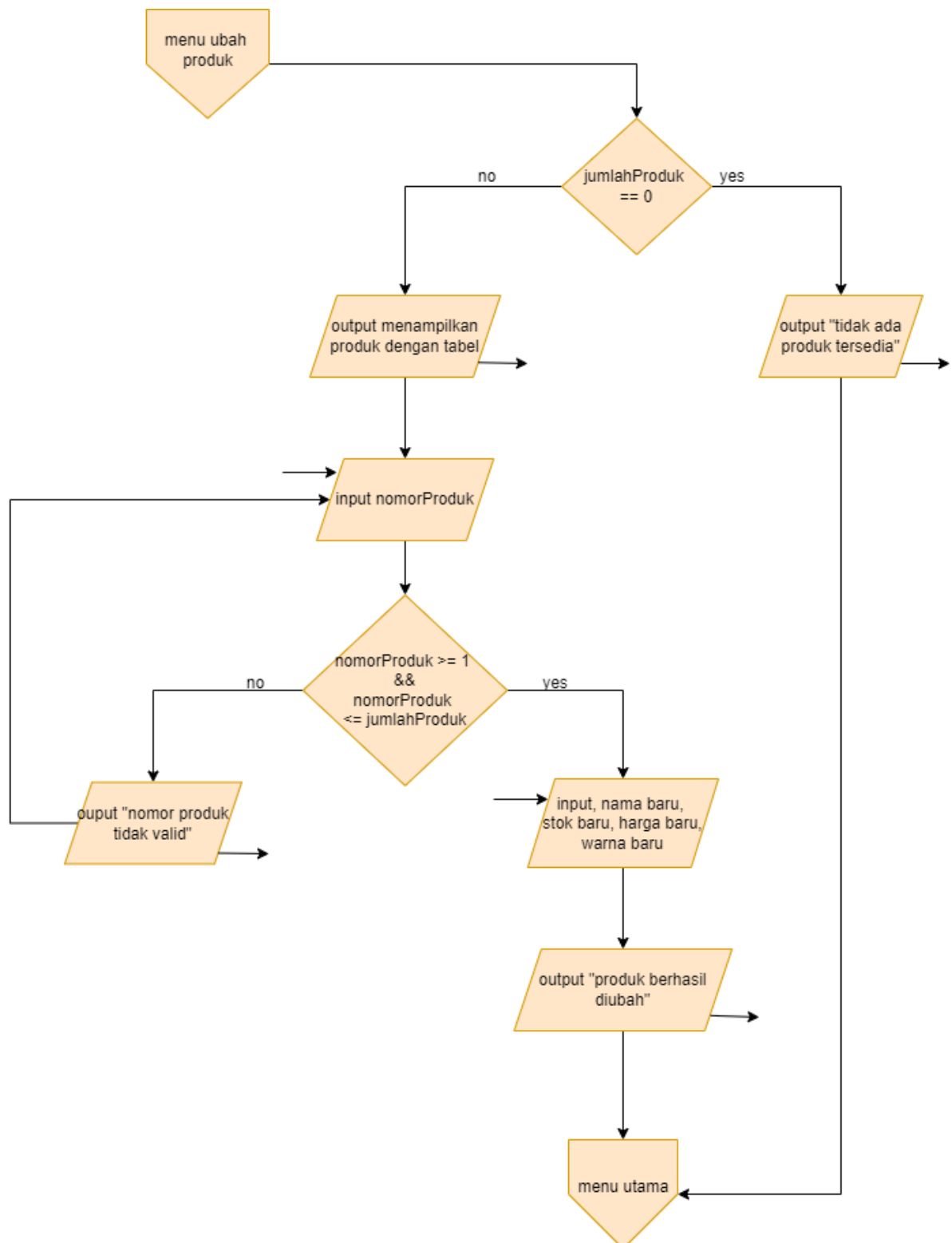
Gambar 1.5 Menu Tambah Produk

1.6 Menu Tampilkan Produk



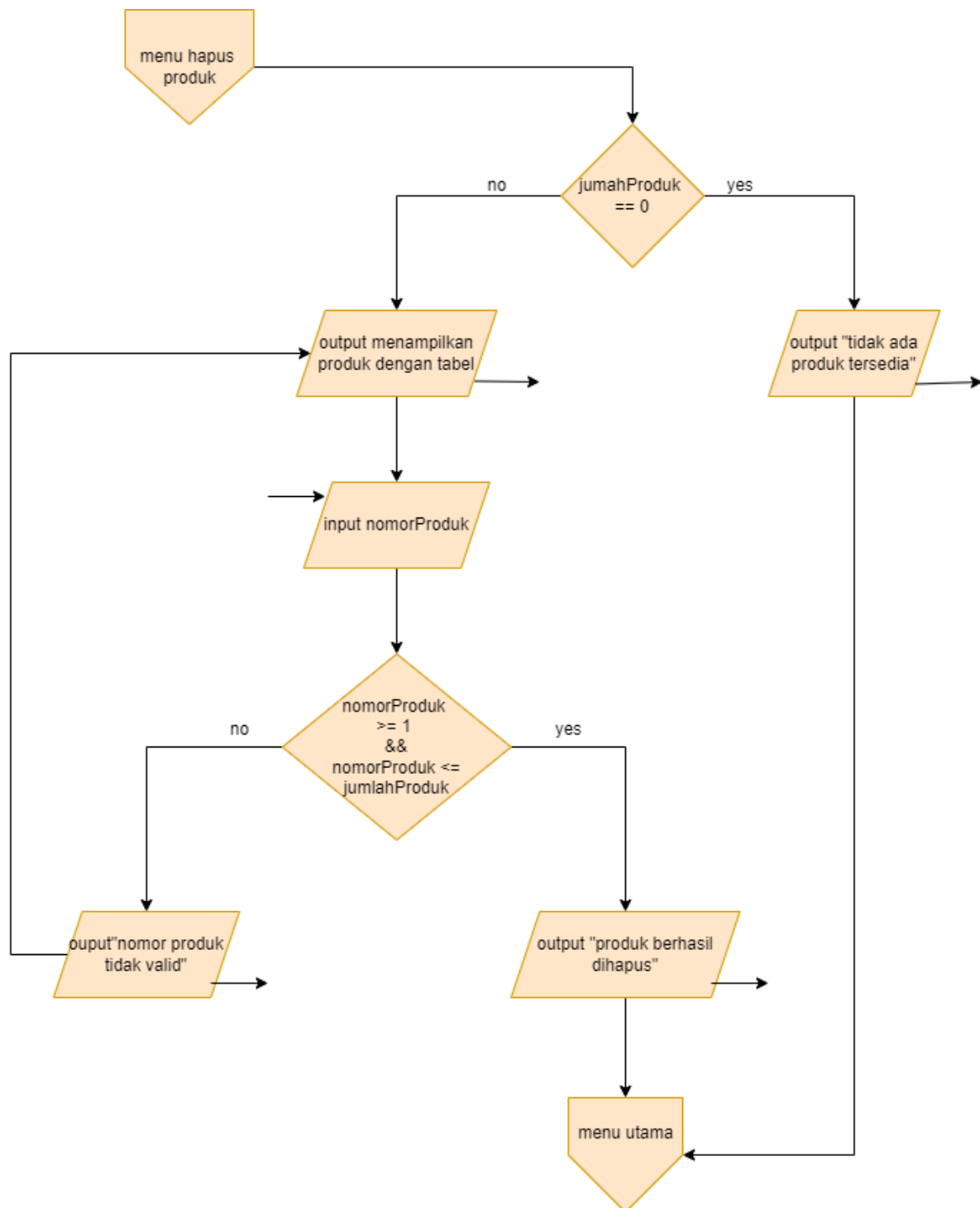
Gambar 1.6 Menu Tampilkan Produk

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 Menu Awal

Kode ini menampilkan menu utama yang memungkinkan pengguna memilih antara tiga opsi: Registrasi, Login, atau Keluar. Program akan terus menampilkan menu ini dalam loop selama variabel running bernilai true.

```
while (running) {
    cout << "===== " << endl;
    cout << "|          MANAJEMEN TOKO HIJAB          |" << endl;
    cout << "===== " << endl;
    cout << "|          1. Registrasi          |" << endl;
    cout << "|          2. Login              |" << endl;
    cout << "|          3. Keluar              |" << endl;
    cout << "===== " << endl;
    cout << "Masukkan pilihan : ";
    cin >> pilihan;
    cin.ignore();
}
```

3.1 Menu Registrasi

Bagian kode ini menangani registrasi pengguna baru. Jika jumlah pengguna belum mencapai batas ('MAX_USER'), program akan meminta nama dan NIM pengguna, kemudian menambahkannya ke dalam array 'users'. Jika jumlah pengguna sudah penuh, program menampilkan pesan bahwa jumlah pengguna sudah penuh.

```
if (pilihan == 1) {
    //Registrasi user baru
    if (jumlahUser < MAX_USER) {
        cout << "Masukkan Nama : ";
        getline(cin, users[jumlahUser].nama);
        cout << "Masukkan NIM : ";
        getline(cin, users[jumlahUser].nim);
        jumlahUser++;
        cout << "Registrasi berhasil!\n";
    }else {
        cout << "Jumlah user sudah penuh!\n";
    }
}
```

3.1 Menu Login

Bagian kode ini menangani login pengguna. Pengguna diminta memasukkan Nama dan NIM, lalu program memeriksa apakah data cocok dengan yang terdaftar. Jika login gagal 3 kali, program berhenti. Jika berhasil, login diterima dan program melanjutkan.

```
} else if (pilihan == 2) {
    loginAttempts = 0;
    userIndex = -1;
    while (loginAttempts < MAX_LOGIN_ATTEMPTS) {
        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";
            break;
        } else {
            cout << "Login gagal. Coba Lagi,\n";
            loginAttempts++;
        }
    }
    if (loginAttempts == MAX_LOGIN_ATTEMPTS) {
        cout << "Login gagal 3 kali. Program berhenti.\n";
        return 0;
    }
}
```

3.1 Menu Utama

Bagian kode ini menampilkan menu manajemen produk setelah login, dengan opsi untuk menambah, menampilkan, mengubah, atau menghapus produk, serta keluar dari menu. Program terus menampilkan menu ini selama pengguna berhasil login.

```
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();

```

3.1 Menu Tambah Produk

Bagian kode ini menangani penambahan produk. Jika jumlah produk belum penuh, pengguna diminta memasukkan nama, stok, harga, dan warna produk. Setelah itu, produk ditambahkan ke dalam daftar dan jumlah produk bertambah. Jika data penuh, program menampilkan pesan "Data penuh!".

```

if (pilihan == 1) {
    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";
    } else {
        cout << "Data penuh!\n";
    }
}

```

3.1 Menu Tampilkan Produk

Bagian kode ini menampilkan daftar produk. Jika produk tersedia, akan ditampilkan dalam format tabel. Jika tidak ada produk, program menampilkan pesan "Tidak ada produk tersedia."

```

} else if (pilihan == 2) {
    if (jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
    } 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";
        }

```

3.1 Menu Ubah Produk

Bagian kode ini memungkinkan pengguna untuk mengubah produk. Setelah menampilkan daftar produk, pengguna memilih produk yang ingin diubah, kemudian memasukkan nama, stok, harga, dan warna baru untuk produk tersebut.

```

} else if (pilihan == 3) {
    if (jumlahProduk == 0) {
        cout << "\\nTidak ada produk tersedia.\\n";
    } 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";
    int nomorProduk;
    while (true) {
        cout << "Masukkan nomor produk yang ingin diubah : ";
        cin >> nomorProduk;
        cin.ignore();
        if (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";
            break;
        } else {
            cout << "Nomor produk tidak valid\\n";
        }
    }
}

```

3.1 Menu Hapus Produk

Bagian kode ini memungkinkan pengguna untuk menghapus produk. Setelah menampilkan daftar produk, pengguna memilih produk yang ingin dihapus dengan memasukkan nomor produk. Jika nomor valid, produk dihapus dan pesan "Produk berhasil dihapus" ditampilkan.

```

} else if (pilihan == 4) {
    if (jumlahProduk == 0) {
        cout << "\\nTidak ada produk tersedia.\\n";
    } 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";
        int nomorProduk;
        while (true) {
            cout << "Masukkan nomor produk yang ingin diubah : ";
            cin >> nomorProduk;
            cin.ignore();
            if (nomorProduk >= 1 && nomorProduk <= jumlahProduk) {
                for (int i = nomorProduk - 1; i < jumlahProduk - 1; i++) {
                    produk[i] = produk[i + 1];
                }
                jumlahProduk--;
                cout << "Produk berhasil dihapus.\\n";
                break;
            } else {
                cout << "Nomor produk tidak valid\\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 : segitiga
Masukkan stok : 23
Masukkan harga : 19500
Warna standar ke-1: hitam
Warna premium ke-1: putih
Warna standar ke-2: biru
Warna premium ke-2: abu
Warna standar ke-3: pink
Warna premium ke-3: coklat
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 | segitiga      |  23 | 19500 | hitam, biru, pink | putih, abu, coklat |
=====
```

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 | segitiga      |  23 | 19500 | hitam, biru, pink | putih, abu, coklat |
=====

Masukkan nomor produk yang ingin diubah : 1
Masukkan nama baru : Segitiga
Masukkan stok baru : 100
Masukkan harga baru : 20500
Warna standar ke-1: hitam
Warna premium ke-1: putih
Warna standar ke-2: navy
Warna premium ke-2: abu
Warna standar ke-3: pink
Warna premium ke-3: coklat
Produk berhasil diubah.
```

Gambar 4.5 Ubah 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 | Segitiga     |  100 | 20500 | hitam, navy, pink | putih, abu, coklat |
=====
```

Gambar 4.6 Tampilkan Produk Setelah Diubah

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

=====
| No | Nama          | Stok | Harga | Warna Standar | Warna Premium |
=====
|  1 | Segitiga     |  100 | 20500 | hitam, navy, pink | putih, abu, coklat |
=====

Masukkan nomor produk yang ingin diubah : 1
Produk berhasil dihapus.
```

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-3> █
```

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 03924aa] Program selesai
2 files changed, 231 insertions(+)
create mode 100644 post-test/post-test-apl-3/2409106019-NikyJenitaPutri-PT-3.cpp
create mode 100644 post-test/post-test-apl-3/2409106019-NikyJenitaPutri-PT-3.exe
```

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

```
PS D:\praktikum-apl> git push -u origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 152.66 KiB | 454.00 KiB/s, done.
Total 6 (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/NikyJenitaPutri/praktikum-apl.git
    07a7b9c..8f8a18f  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\praktikum-apl> █
```

Gambar 5.5 Git Push