

LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT

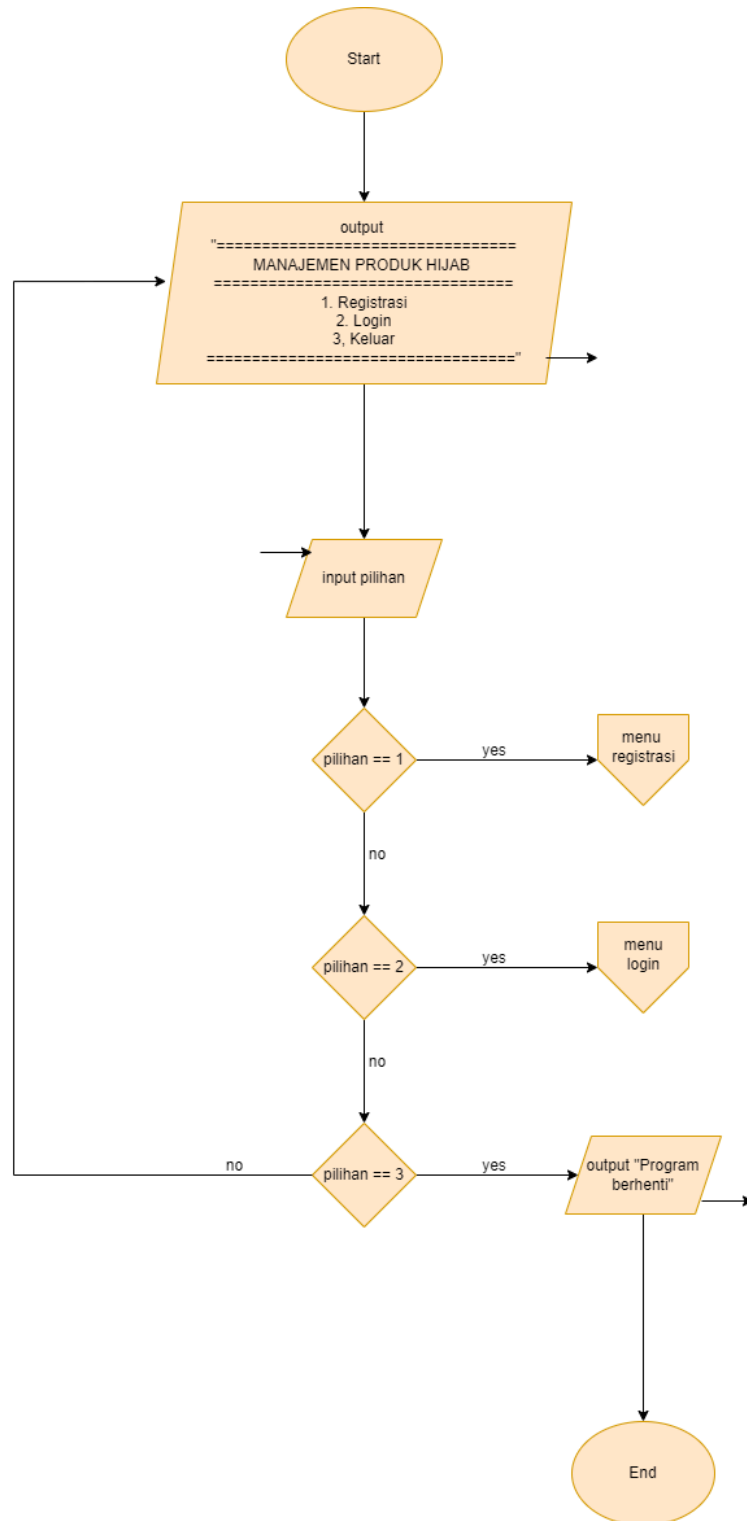


Disusun oleh:
Niky Jenita Putri (2409106019)
Kelas (A1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

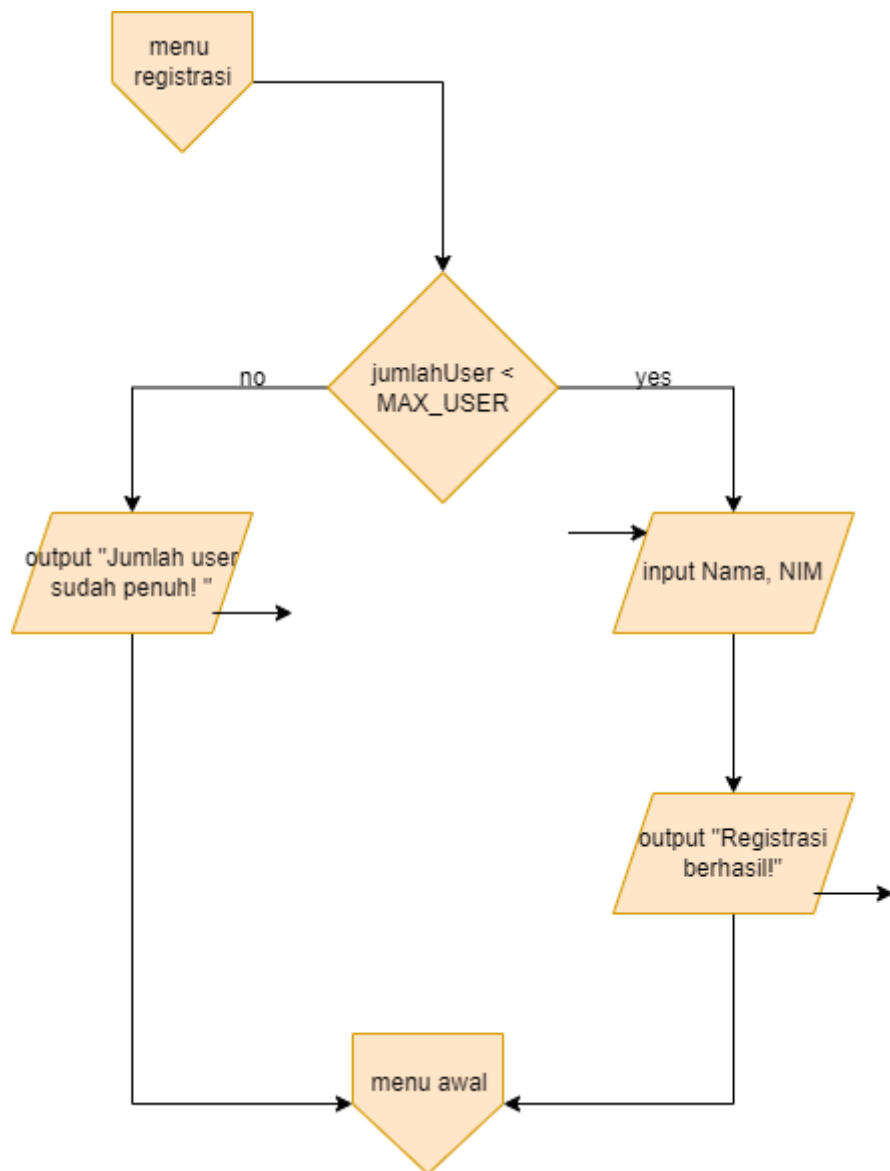
1. Flowchart

1.1 Menu Awal



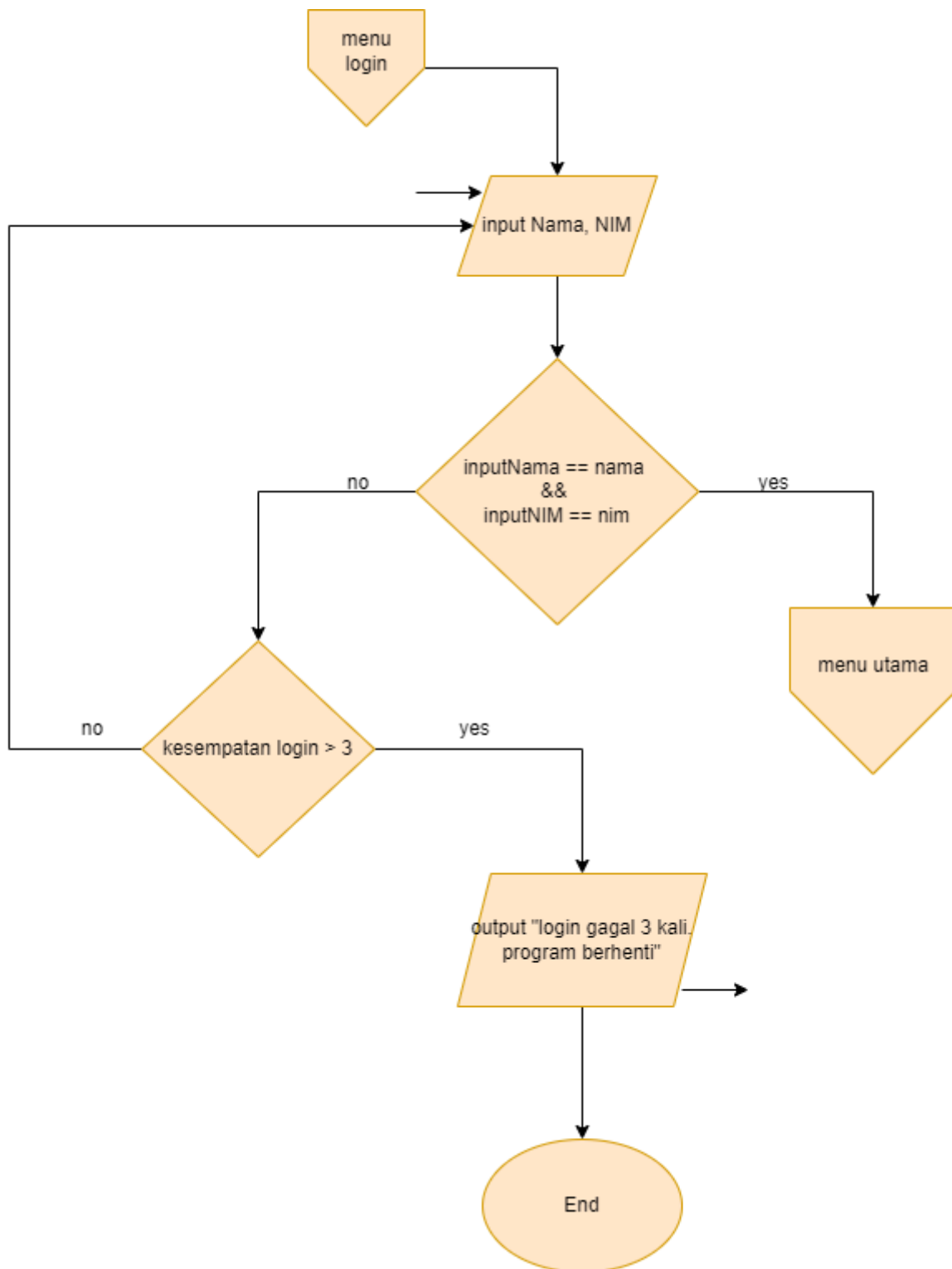
Gambar 1.1 Menu Awal

1.2 Menu Registrasi



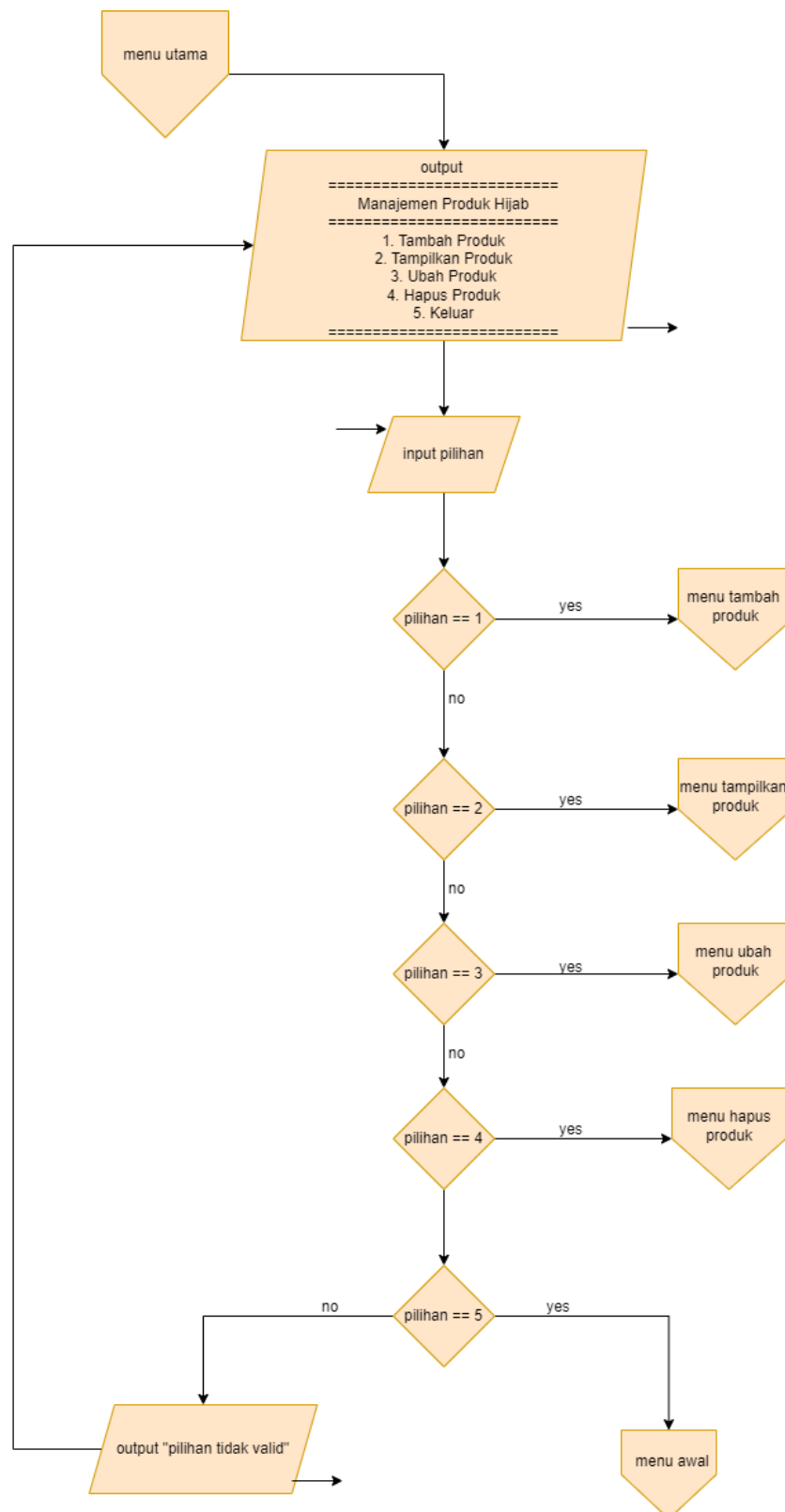
Gambar 1.2 Menu Registrasi

1.3 Menu Login



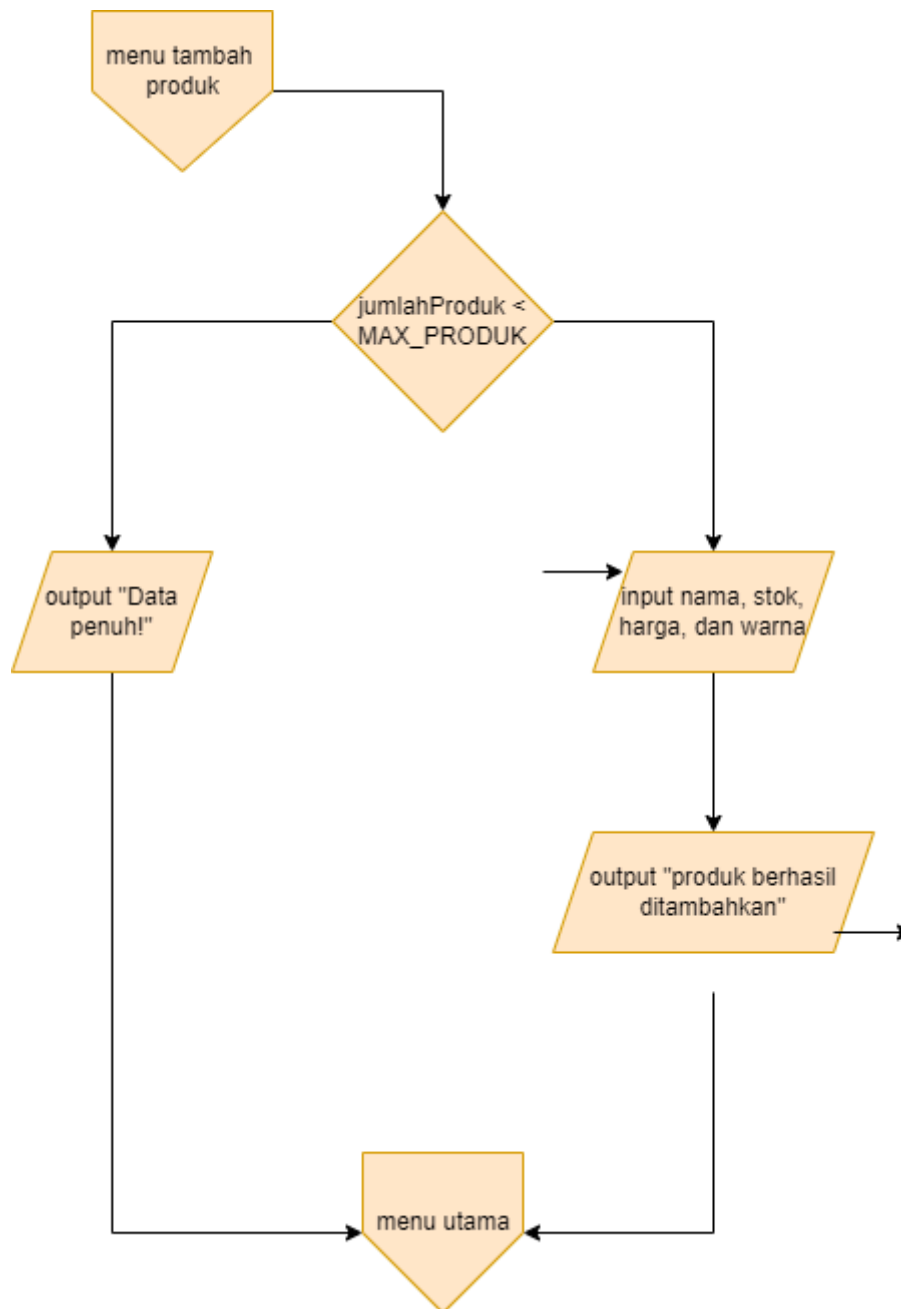
Gambar 1.3 Menu Login

1.4 Menu Utama



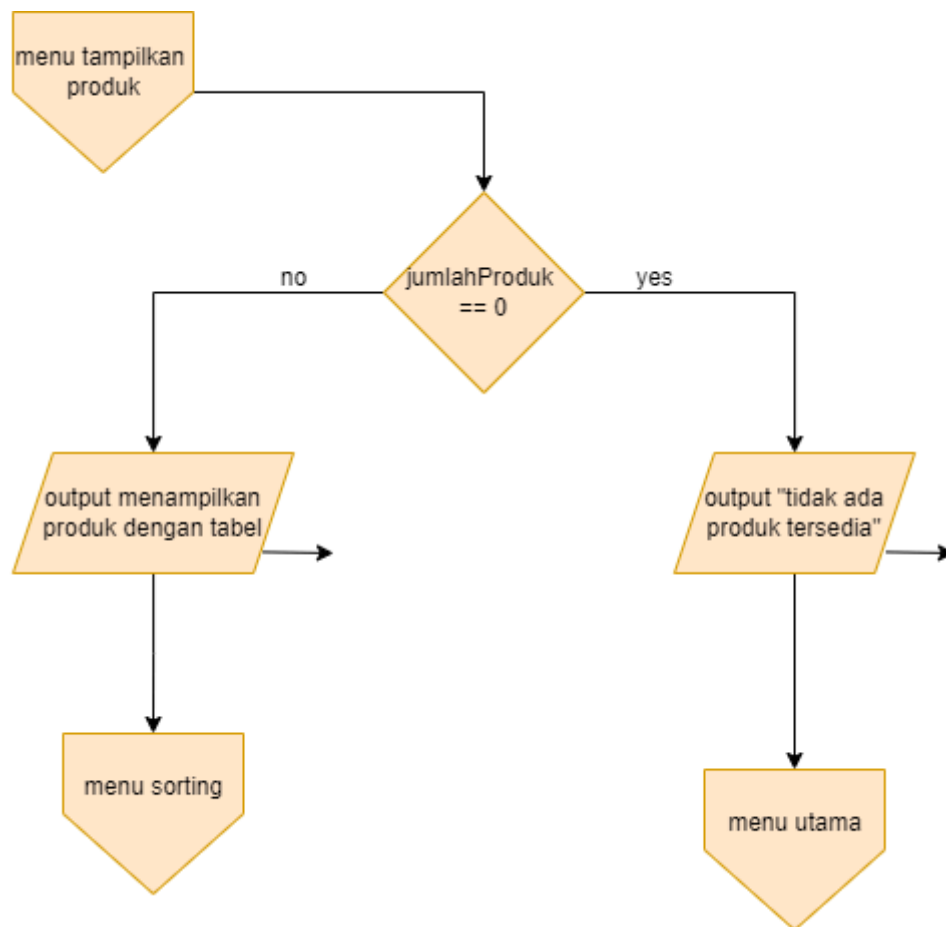
Gambar 1.4 Menu Utama

1.5 Menu Tambah Produk



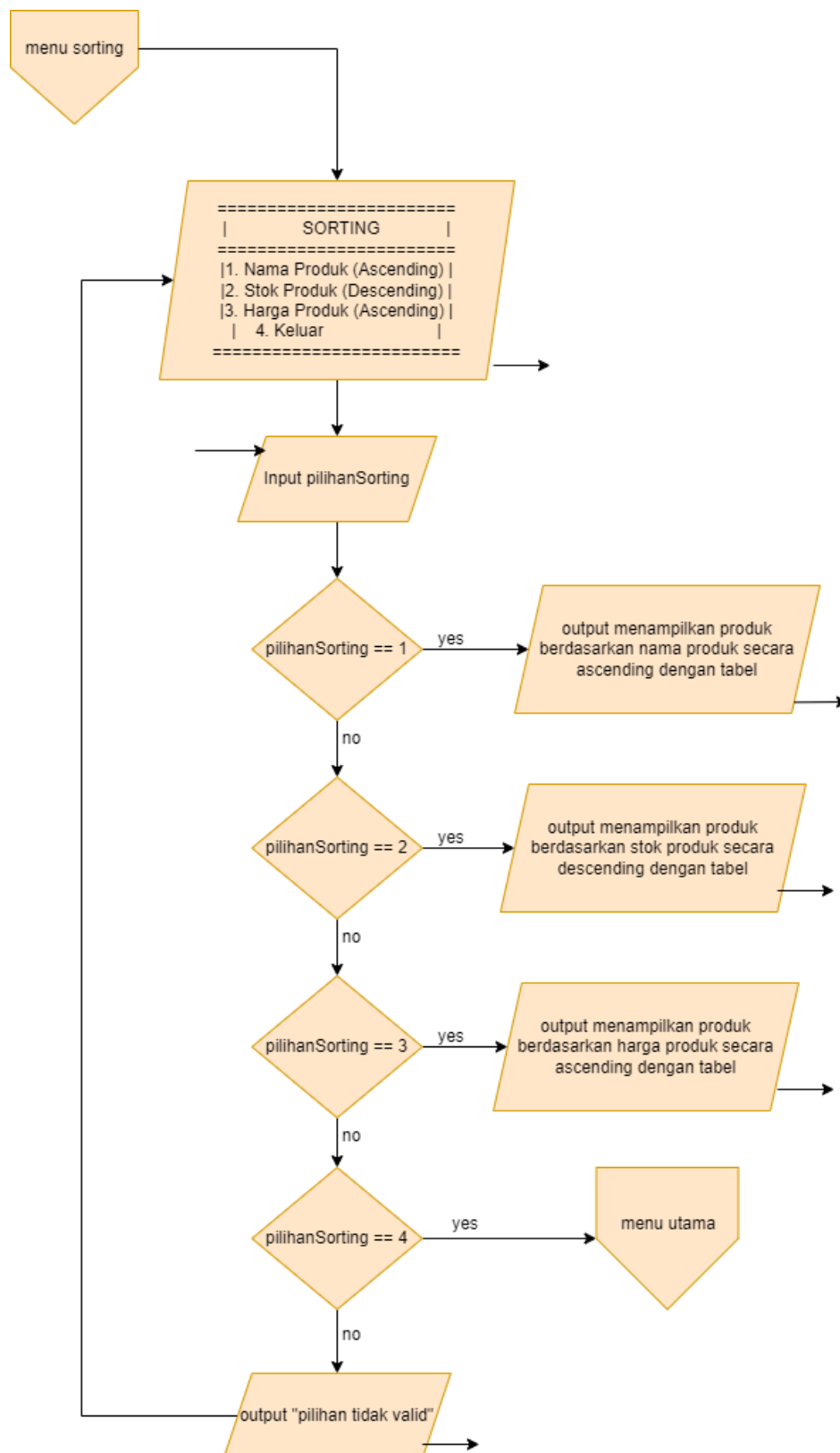
Gambar 1.5 Menu Tambah Produk

1.6 Menu Tampilkan Produk



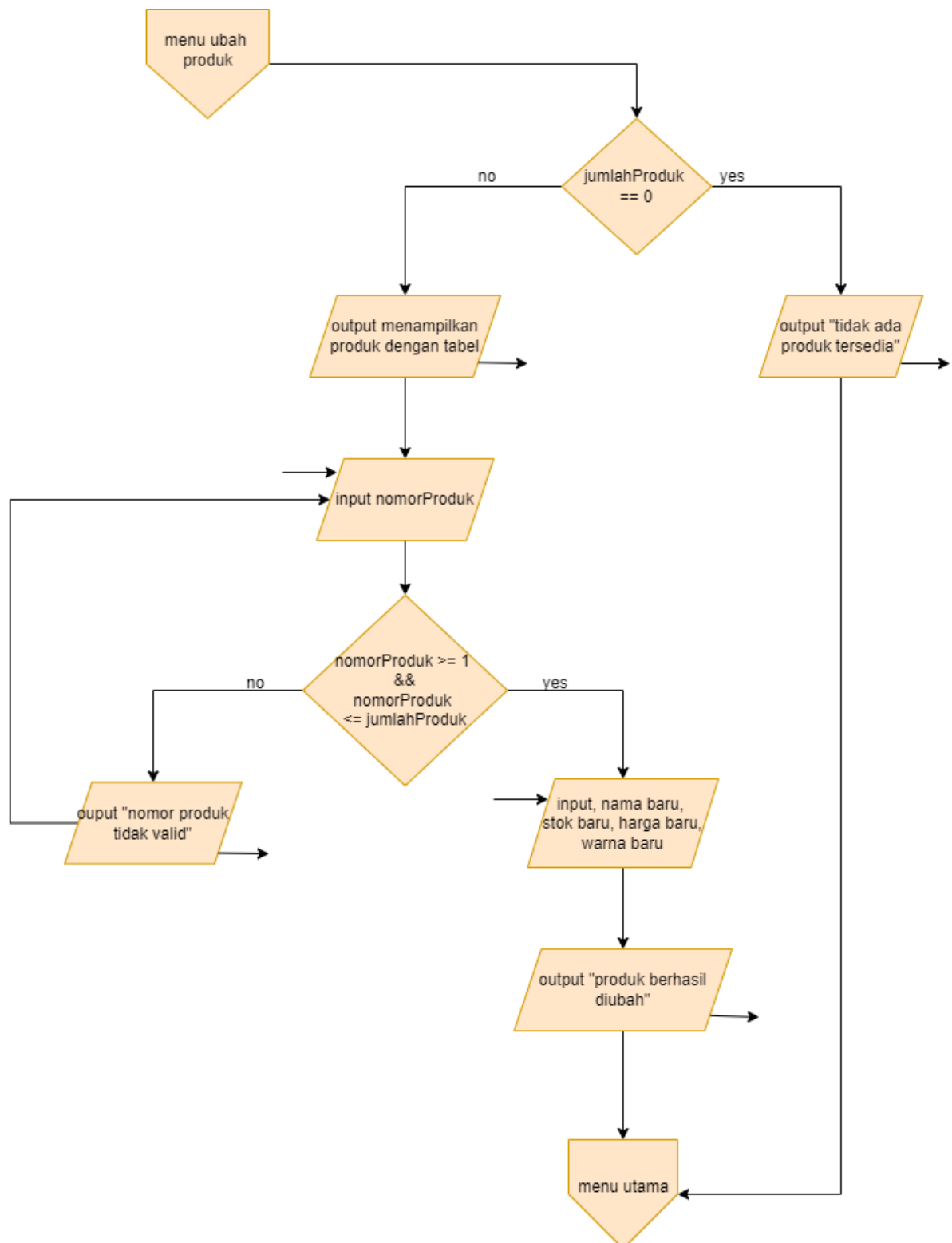
Gambar 1.6 Menu Tampilkan Produk

1.7 Menu Sorting



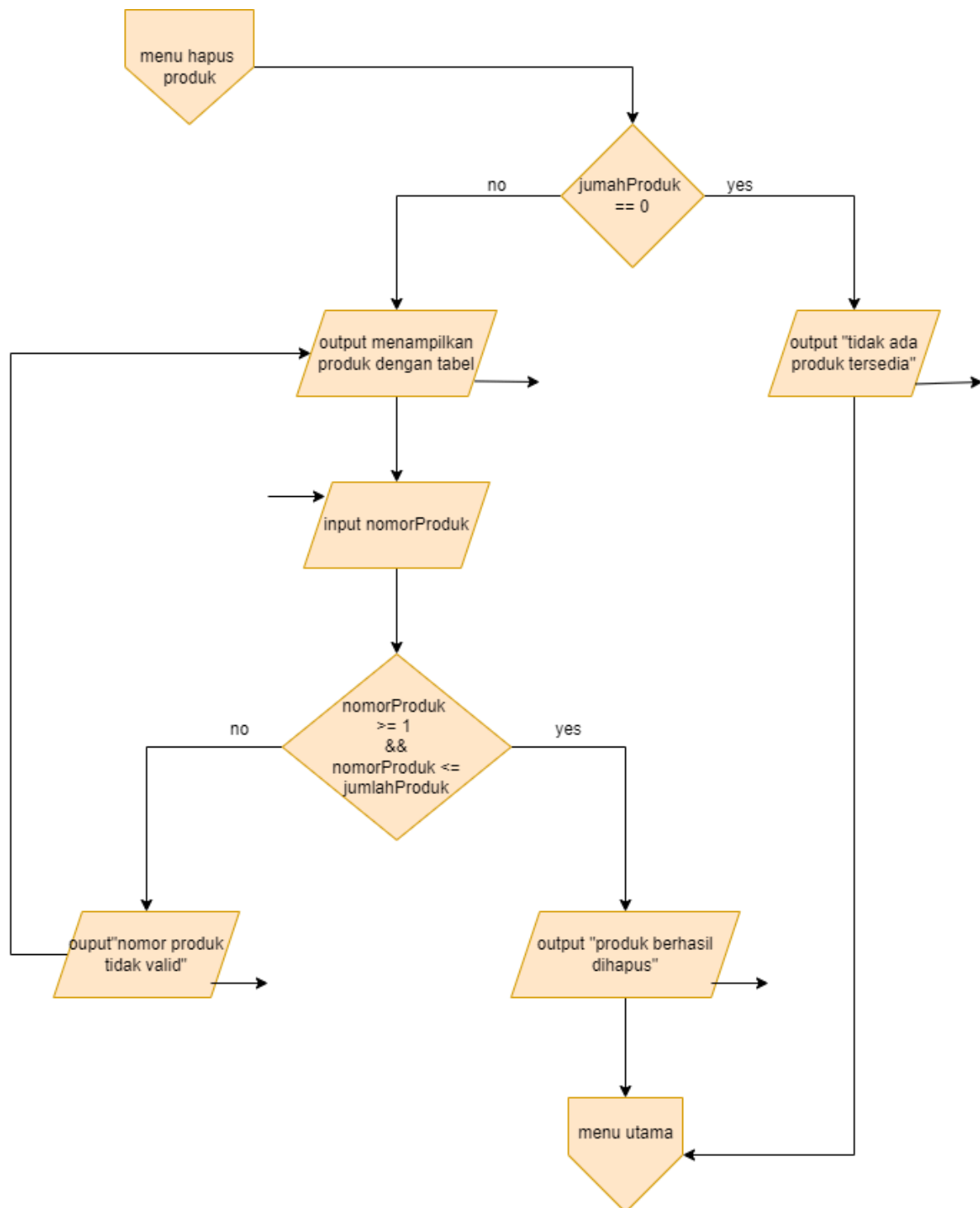
Gambar 1.7 Menu Sorting

1.8 Menu Ubah Produk



Gambar 1.8 Menu Ubah Produk

1.9 Menu Hapus Produk



Gambar 1.9 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. Fitur sorting juga disediakan untuk mempermudah pencarian produk berdasarkan kriteria tertentu seperti nama, stok, atau harga. Selain itu, program ini memungkinkan pemilik toko untuk memantau dan mengontrol stok dan harga produk dengan mudah. Dengan demikian, program ini bertujuan untuk mempermudah pengelolaan toko hijab, meningkatkan efisiensi operasional, dan membantu pemilik toko dalam mengatur produk serta melakukan transaksi dengan lebih terorganisir.

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);
            menusorting(*data);
        } else if (pilihan == 3) {
            ubahProduk(data);
        } else if (pilihan == 4) {
            hapusProduk(data);
        } else if (pilihan == 5) {
            userIndex = -1;
        } else {
            cout << "Pilihan tidak valid" << endl;
        }
    }
}
```

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";
    }
}

```

3.9 Menu Sorting Nama Produk

Fungsi `sortingNamaProduk` digunakan untuk mengurutkan produk berdasarkan nama secara ascending (dari A ke Z) menggunakan algoritma bubble sort. Fungsi ini membandingkan nama produk satu per satu, dan jika nama produk di posisi sebelumnya lebih besar dari nama produk di posisi berikutnya, kedua produk tersebut ditukar. Proses ini diulang hingga semua produk terurutkan berdasarkan nama.

```

void sortingNamaProduk(Data& data) {
    for (int i = 0; i < data.jumlahProduk - 1; i++) {
        for (int j = 0; j < data.jumlahProduk - i - 1; j++) {
            if (data.produk[j].nama > data.produk[j+1].nama) {
                Produk temp = data.produk[j];
                data.produk[j] = data.produk[j+1];
                data.produk[j+1] = temp;
            }
        }
    }
}

```

3.10 Menu Sorting Stok Produk

Fungsi `sortingStokProduk` digunakan untuk mengurutkan produk berdasarkan stok secara descending (dari yang terbanyak ke yang sedikit) menggunakan algoritma bubble sort. Fungsi ini membandingkan stok produk satu per satu, dan jika stok produk di posisi sebelumnya lebih sedikit daripada stok produk di posisi berikutnya, kedua produk tersebut ditukar. Proses ini diulang hingga semua produk terurutkan berdasarkan stok secara menurun.

```

void sortingStokProduk(Data& data) {
    for (int i = 0; i < data.jumlahProduk - 1; i++) {
        for (int j = 0; j < data.jumlahProduk - i - 1; j++) {
            if (data.produk[j].stok < data.produk[j+1].stok) {
                Produk temp = data.produk[j];
                data.produk[j] = data.produk[j+1];
                data.produk[j+1] = temp;
            }
        }
    }
}

```

```

    }
}

```

3.11 Menu Sorting Harga Produk

Fungsi `sortingHargaProduk` digunakan untuk mengurutkan produk berdasarkan harga secara ascending (dari yang terendah ke yang tertinggi) menggunakan algoritma bubble sort. Fungsi ini membandingkan harga produk satu per satu, dan jika harga produk di posisi sebelumnya lebih tinggi daripada harga produk di posisi berikutnya, kedua produk tersebut ditukar. Proses ini diulang hingga semua produk terurutkan berdasarkan harga secara menaik.

```

void sortingHargaProduk(Data& data) {
    for (int i = 0; i < data.jumlahProduk - 1; i++) {
        for (int j = 0; j < data.jumlahProduk - i - 1; j++) {
            if (data.produk[j].harga > data.produk[j+1].harga) {
                Produk temp = data.produk[j];
                data.produk[j] = data.produk[j+1];
                data.produk[j+1] = temp;
            }
        }
    }
}

```

3.12 Menu Sorting

Fungsi `menusorting` menampilkan menu untuk mengurutkan produk berdasarkan nama (ascending), stok (descending), atau harga (ascending). Jika produk tersedia, pengguna dapat memilih kriteria sorting, dan produk akan diurutkan serta ditampilkan. Jika memilih keluar, program kembali ke menu utama.

```

void menusorting(Data &data) {
    int pilihanSorting;
    if (data.jumlahProduk == 0) {
        cout << "\nTidak ada produk tersedia.\n";
        return ;
    } else {
        do{
            cout << "===== " << endl;
            cout << "|                SORTING                |" << endl;
            cout << "===== " << endl;
            cout << "|    1. Nama Produk (Ascending)          |" << endl;
            cout << "|    2. Stok Produk (Descending)         |" << endl;
            cout << "|    3. Harga Produk (Ascending)         |" << endl;
            cout << "|    4. Keluar                           |" << endl;
            cout << "===== " << endl;
            cout << "Masukkan pilihan : ";

```

```

    cin >> pilihanSorting;
    cin.ignore();

    switch (pilihanSorting) {
        case 1:
            sortingNamaProduk(data);
            cout << "Produk berhasil diurutkan berdasarkan Nama Produk
(Ascending)" << endl;
            tampilkanProduk(data);
            break;
        case 2:
            sortingStokProduk(data);
            cout << "Produk berhasil diurutkan berdasarkan Stok Produk
(descending)" << endl;
            tampilkanProduk(data);
            break;
        case 3:
            sortingHargaProduk(data);
            cout << "Produk berhasil diurutkan berdasarkan Harga Produk
(Ascending)" << endl;
            tampilkanProduk(data);
            break;
        case 4:
            cout << "Kembali ke menu" << endl;
            break;
        default:
            cout << "Pilihan tidak valid" << endl;
    }
} while (pilihanSorting!=4 );
}
}

```

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 : 102
Masukkan harga : 35900
Warna standar ke-1: hitam
Warna premium ke-1: putih
Warna standar ke-2: navy
Warna premium ke-2: coklat
Warna standar ke-3: sage
Warna premium ke-3: cream
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     | 102  | 35900 | hitam, navy, sage | putih, coklat, cream |
| 2 | bergo        | 99   | 28000 | abu, maroon, hitam | dusty, caramel, coral |
| 3 | segi empat   | 0    | 19900 | army, peach, coklat | olive, lilac, rosepink |
| 4 | instan       | 112  | 40000 | biru, mustard, navy | stabilo, mint, hitam |
| 5 | khimar       | 98   | 69900 | putih, coklat, coral | abu, mocca, cream |
=====

```

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     | 102  | 35900 | hitam, navy, sage | putih, coklat, cream |
| 2 | bergo        | 99   | 28000 | abu, maroon, hitam | dusty, caramel, coral |
| 3 | segi empat   | 0    | 19900 | army, peach, coklat | olive, lilac, rosepink |
| 4 | instan       | 112  | 40000 | biru, mustard, navy | stabilo, mint, hitam |
| 5 | khimar       | 98   | 69900 | putih, coklat, coral | abu, mocca, cream |
=====

Masukkan nomor produk yang ingin diubah : 6
Nomor produk tidak valid
Masukkan nomor produk yang ingin diubah : 3
Masukkan nama baru : segi empat
Masukkan stok baru : 123
Masukkan harga baru : 21900
Warna standar ke-1: hitam
Warna premium ke-1: hitam
Warna standar ke-2: army
Warna premium ke-2: peach
Warna standar ke-3: coklat
Warna premium ke-3: olive
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	pashmina	102	35900	hitam, navy, sage	putih, coklat, cream	
2	bergo	99	28000	abu, maroon, hitam	dusty, caramel, coral	
3	segi empat	123	21900	hitam, army, coklat	hitam, peach, olive	
4	instan	112	40000	biru, mustard, navy	stabilo, mint, hitam	
5	khimar	98	69900	putih, coklat, coral	abu, mocca, cream	

Gambar 4.6 Tampilkan produk setelah diubah

SORTING						
1. Nama Produk (Ascending)						
2. Stok Produk (Descending)						
3. Harga Produk (Ascending)						
4. Keluar						
Masukkan pilihan : 1						
Produk berhasil diurutkan berdasarkan Nama Produk (Ascending)						
No	Nama	Stok	Harga	Warna Standar	Warna Premium	
1	bergo	99	28000	abu, maroon, hitam	dusty, caramel, coral	
2	instan	112	40000	biru, mustard, navy	stabilo, mint, hitam	
3	khimar	98	69900	putih, coklat, coral	abu, mocca, cream	
4	pashmina	102	35900	hitam, navy, sage	putih, coklat, cream	
5	segi empat	123	21900	hitam, army, coklat	hitam, peach, olive	

Gambar 4.7 Tampilkan produk dengan mengurutkan nama produk secara ascending

SORTING						
1. Nama Produk (Ascending)						
2. Stok Produk (Descending)						
3. Harga Produk (Ascending)						
4. Keluar						
Masukkan pilihan : 2						
Produk berhasil diurutkan berdasarkan Stok Produk (descending)						
No	Nama	Stok	Harga	Warna Standar	Warna Premium	
1	segi empat	123	21900	hitam, army, coklat	hitam, peach, olive	
2	instan	112	40000	biru, mustard, navy	stabilo, mint, hitam	
3	pashmina	102	35900	hitam, navy, sage	putih, coklat, cream	
4	bergo	99	28000	abu, maroon, hitam	dusty, caramel, coral	
5	khimar	98	69900	putih, coklat, coral	abu, mocca, cream	

Gambar 4.8 Tampilkan produk dengan mengurutkan stok produk secara descending

SORTING						
1. Nama Produk (Ascending)						
2. Stok Produk (Descending)						
3. Harga Produk (Ascending)						
4. Keluar						
Masukkan pilihan : 3						
Produk berhasil diurutkan berdasarkan Harga Produk (Ascending)						
No	Nama	Stok	Harga	Warna Standar	Warna Premium	
1	segi empat	123	21900	hitam, army, coklat	hitam, peach, olive	
2	bergo	99	28000	abu, maroon, hitam	dusty, caramel, coral	
3	pashmina	102	35900	hitam, navy, sage	putih, coklat, cream	
4	instan	112	40000	biru, mustard, navy	stabilo, mint, hitam	
5	khimar	98	69900	putih, coklat, coral	abu, mocca, cream	

Gambar 4.9 Tampilkan produk dengan mengurutkan harga produk secara ascending

```
=====
|          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 | segi empat | 123 | 21900 | hitam, army, coklat | hitam, peach, olive |
| 2 | bergo      | 99  | 28000 | abu, maroon, hitam | dusty, caramel, coral |
| 3 | pashmina   | 102 | 35900 | hitam, navy, sage | putih, coklat, cream |
| 4 | instan     | 112 | 40000 | biru, mustard, navy | stabilo, mint, hitam |
| 5 | khimar     | 98  | 69900 | putih, coklat, coral | abu, mocca, cream |
=====
Masukkan nomor produk yang ingin dihapus : 10
Nomor produk tidak valid
Masukkan nomor produk yang ingin dihapus : 1
Produk berhasil dihapus.
```

Gambar 4.10 Menghapus 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 | bergo      | 99  | 28000 | abu, maroon, hitam | dusty, caramel, coral |
| 2 | pashmina   | 102 | 35900 | hitam, navy, sage | putih, coklat, cream |
| 3 | instan     | 112 | 40000 | biru, mustard, navy | stabilo, mint, hitam |
| 4 | khimar     | 98  | 69900 | putih, coklat, coral | abu, mocca, cream |
=====
```

Gambar 4.11 Tampilkan produk setelah 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 4.12 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 d5daa96] program selesai
2 files changed, 416 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106019-NikyJenitaPutri-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106019-NikyJenitaPutri-PT-6.exe
PS D:\praktikum-apl>
```

Gambar 5.3 Git Commit

5.4 Git Remote

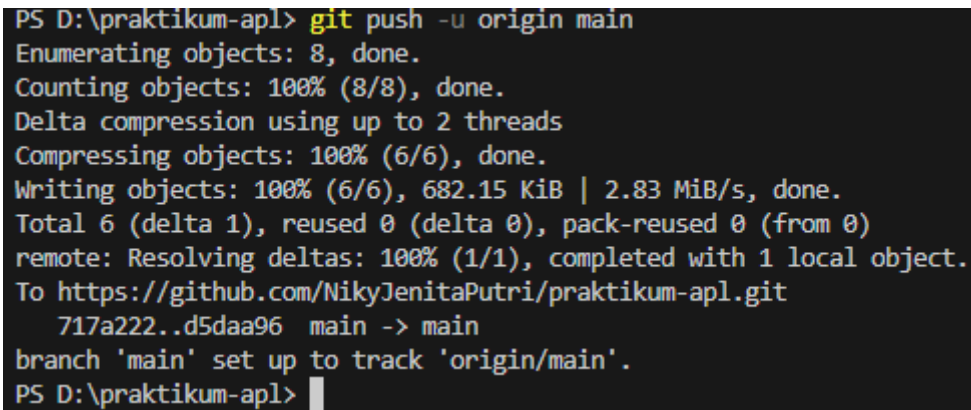
Git remote add origin digunakan untuk menghubungkan repository lokal dengan repository di GitHub.

```
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



```
PS D:\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 2 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 682.15 KiB | 2.83 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/NikyJenitaPutri/praktikum-apl.git
  717a222..d5daa96  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\praktikum-apl>
```

Gambar 5.5 Git Push