

**LAPORAN PRAKTIKUM**  
**POSTTEST 3**  
**ALGORITMA PEMROGRAMAN LANJUT**

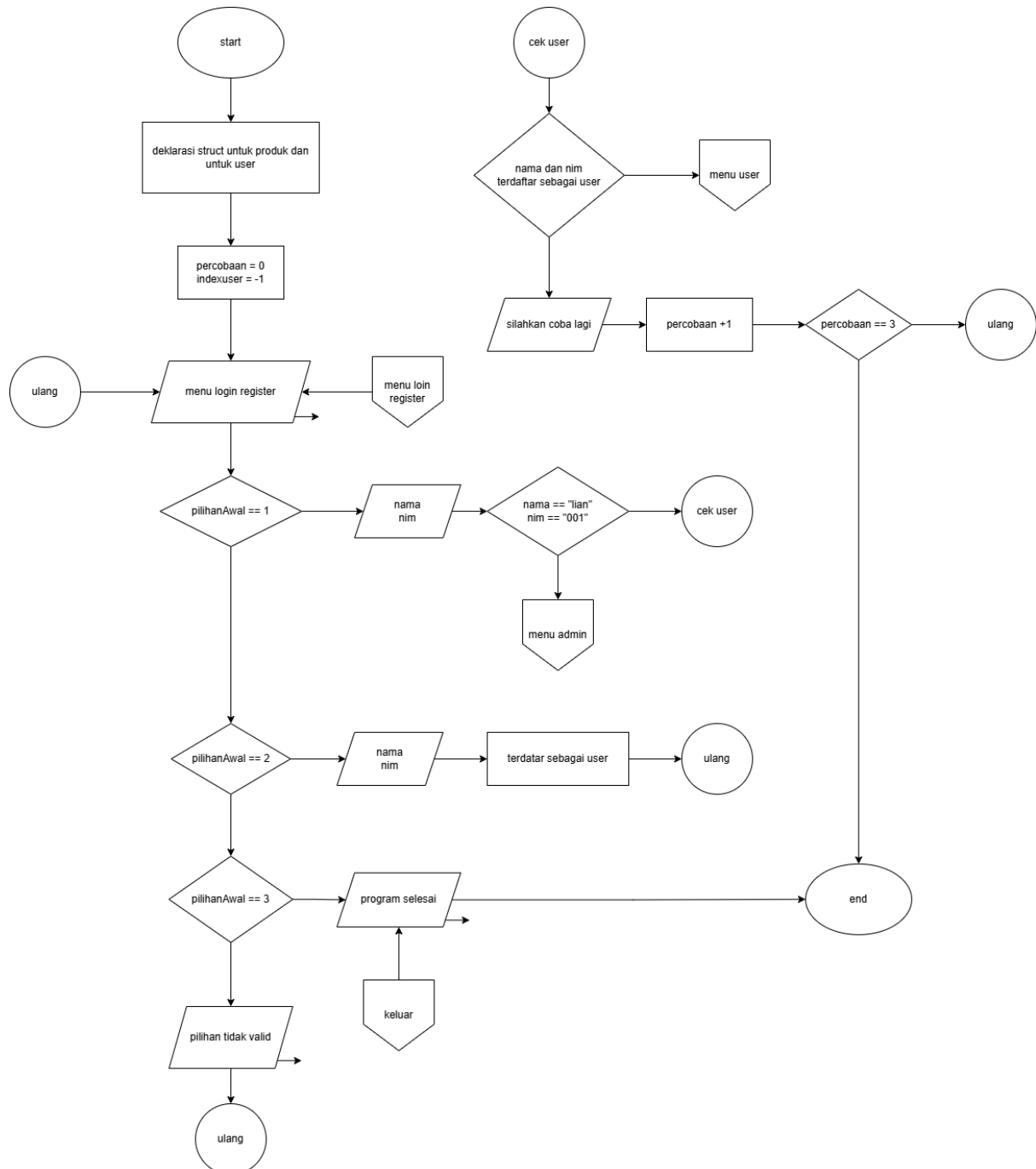


**Disusun oleh:**  
**Dwiki Aprilian Aryanda (2409106001)**  
**Kelas (A1 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

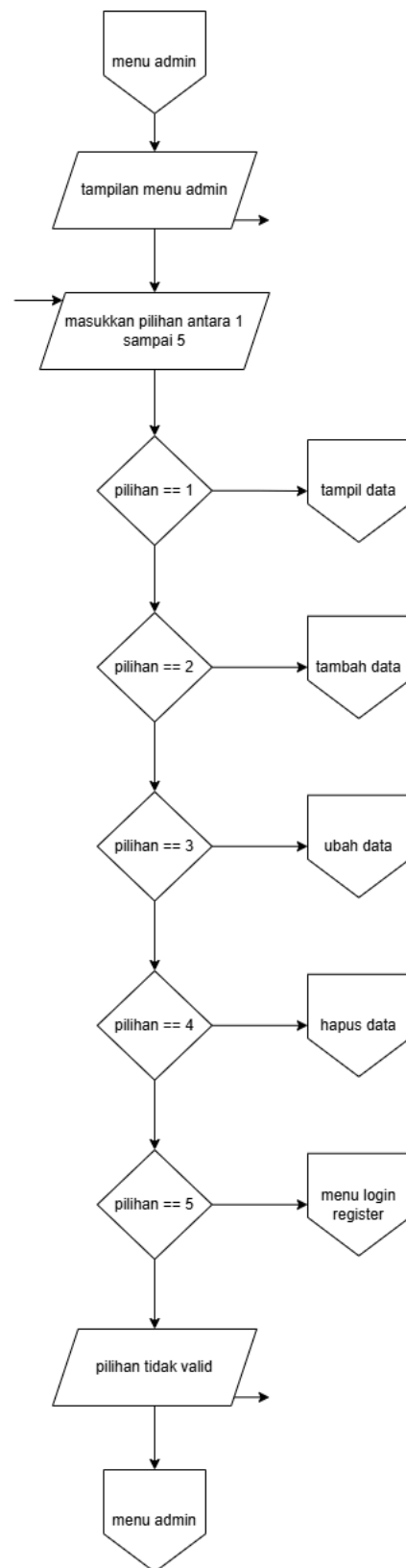
## 1. Flowchart

### 1.1 Menu Login Register



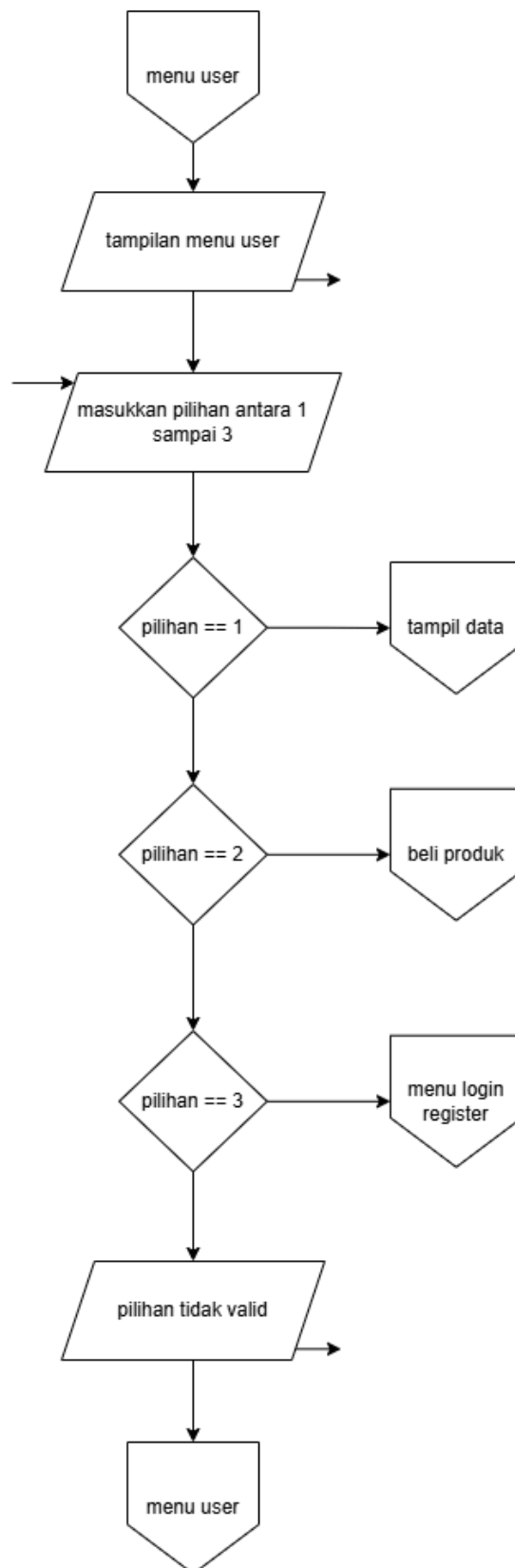
Gambar 1.1 Menu Login Register

## 1.2 Menu Admin



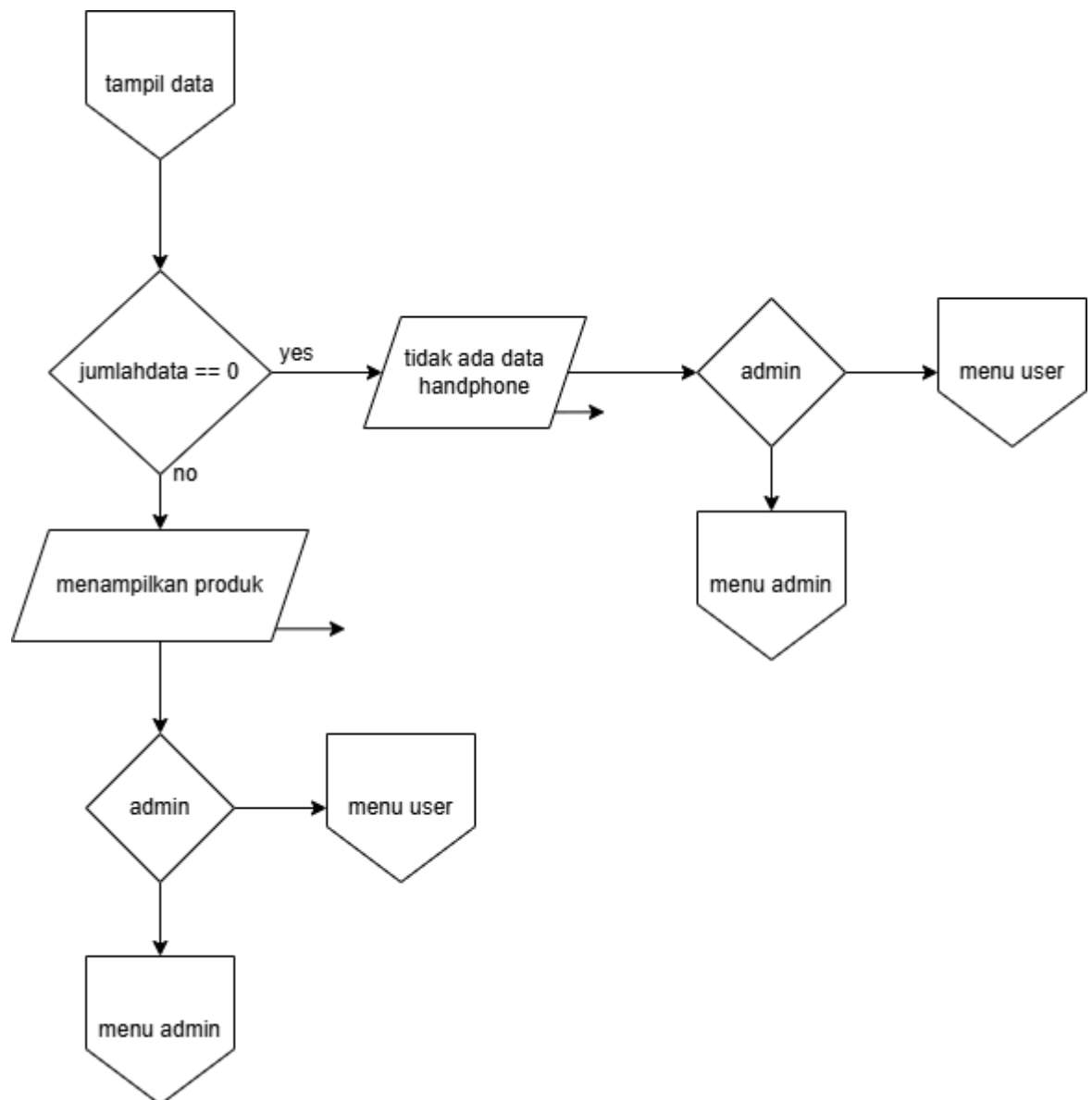
Gambar 1.2 Menu Admin

### 1.3 Menu User



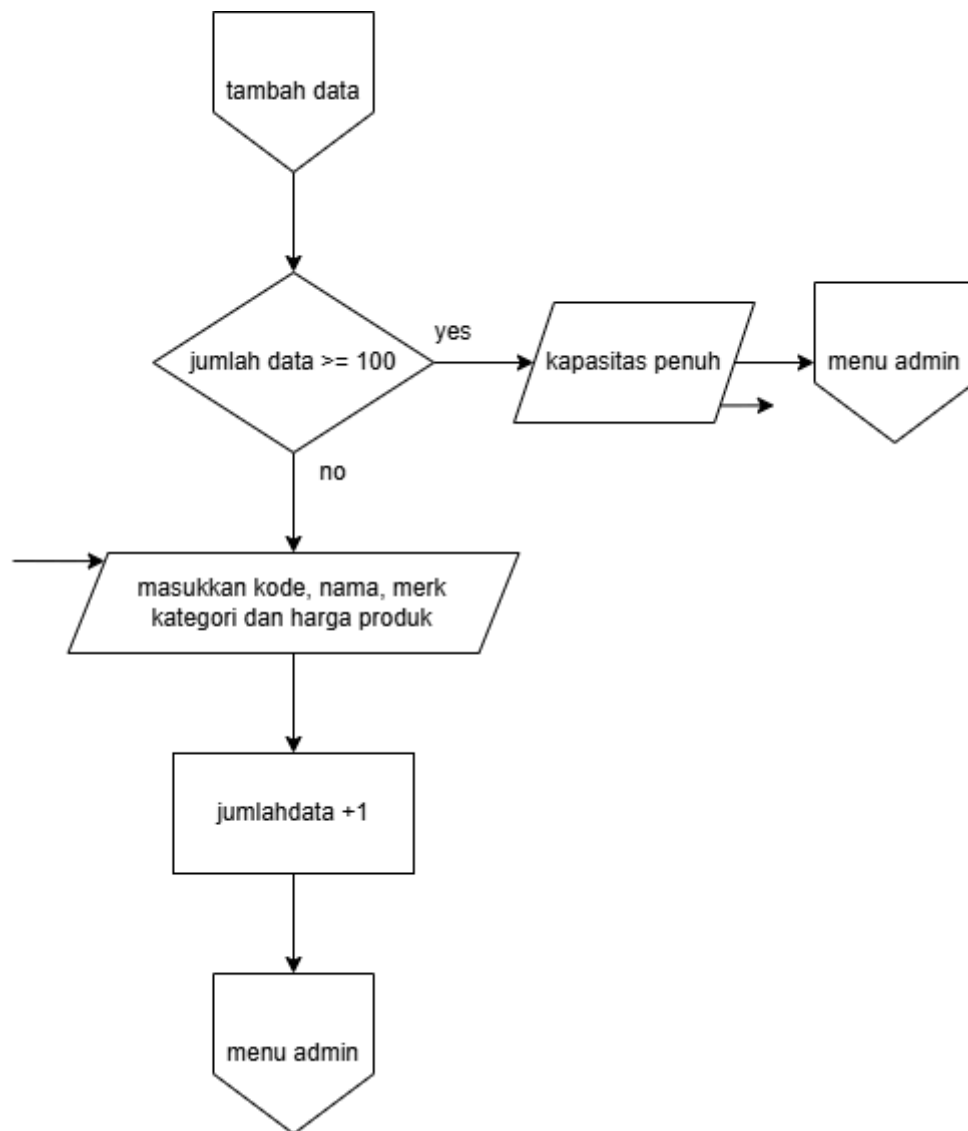
Gambar 1.3 Menu User

## 1.4 Tampil Data



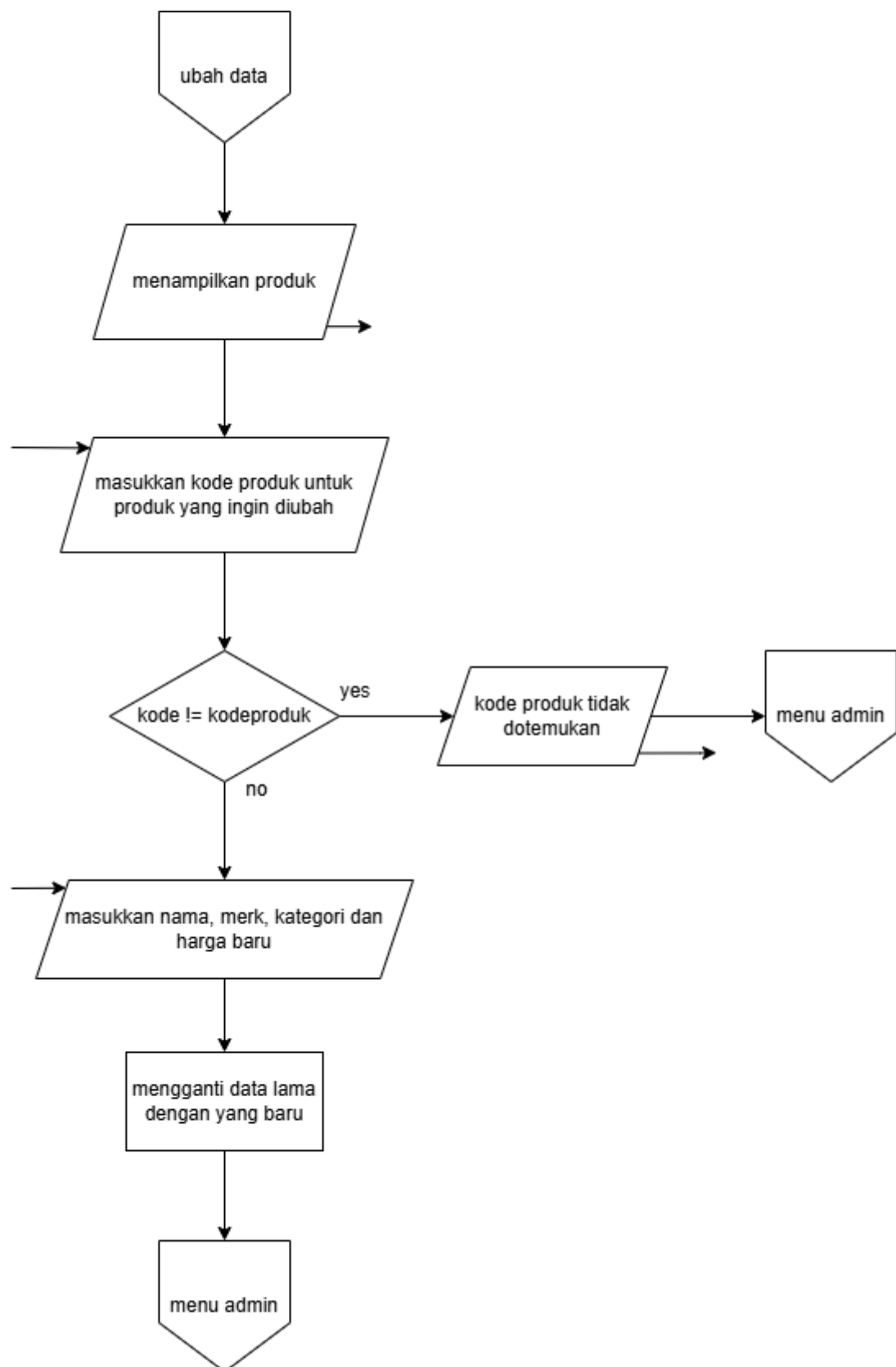
Gambar 1.4 Tampil Data

## 1.5 Tambah Data



Gambar 1.5 Tambah Data

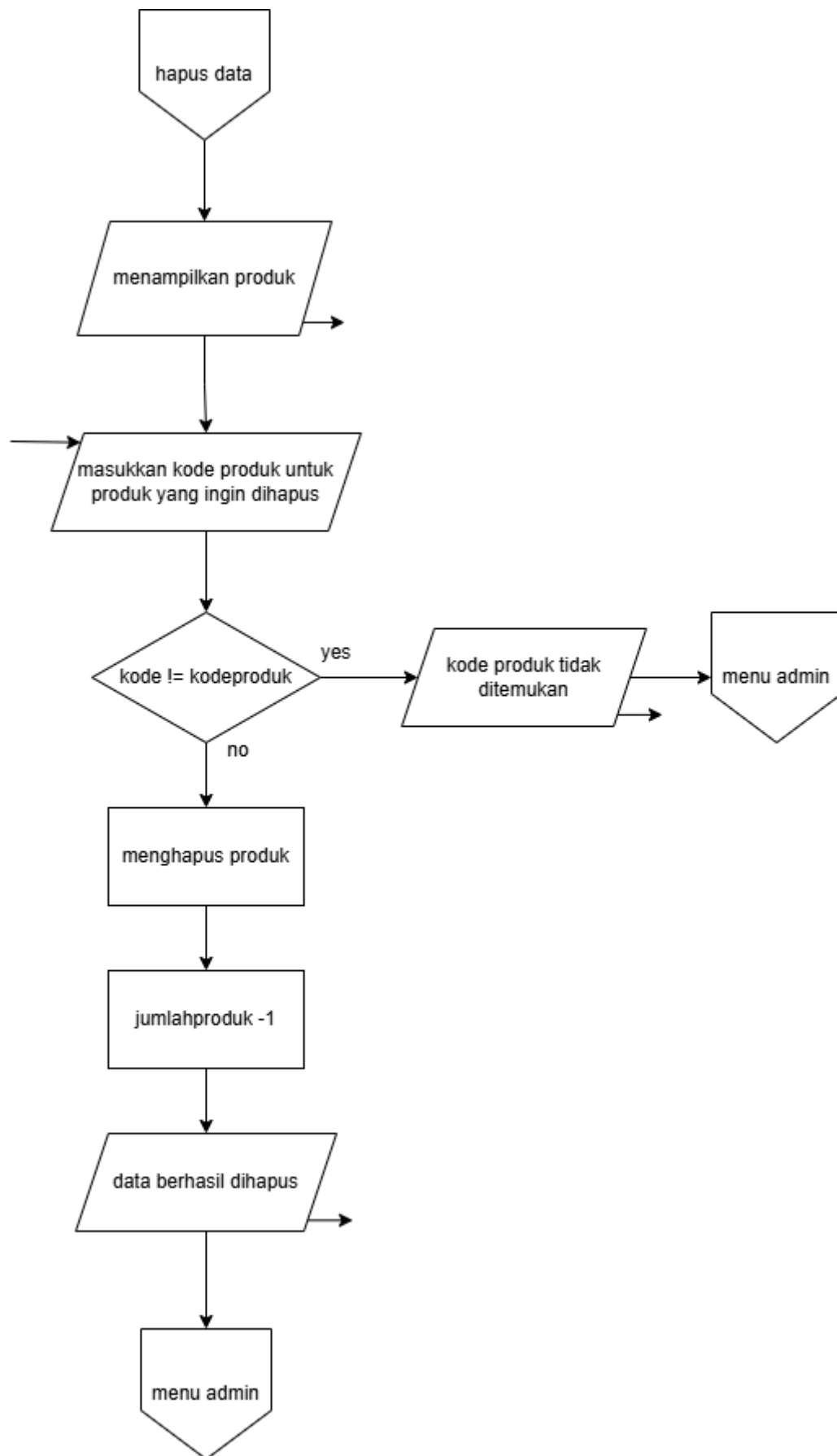
## 1.6 Ubah Data



Gambar 1.6 Ubah Data

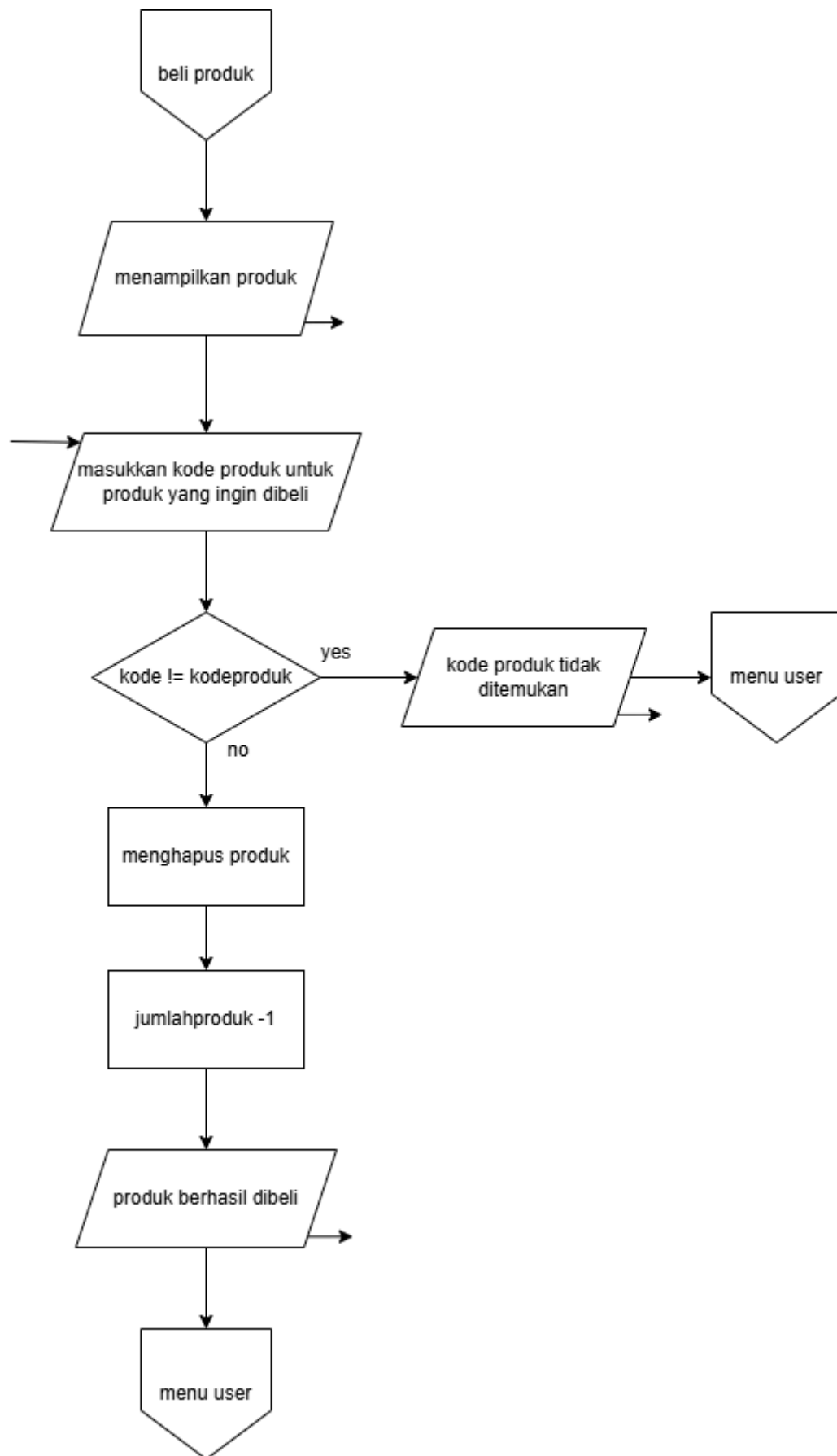
## **1.7 Hapus Data**





Gambar 1.7 Hapus Data

## **1.8 Beli Produk**



Gambar 1.8 Beli Produk

## **2. Analisis Program**

Program ini bertujuan untuk menyediakan sistem manajemen handphone sederhana yang memungkinkan admin ("lian", "001") mengelola data produk (tambah, ubah, hapus) dan user biasa untuk melihat serta membeli produk, dengan manfaat berupa efisiensi dalam pengelolaan inventaris, keamanan data melalui akses terbatas bagi user biasa, fleksibilitas untuk multiuser, dan simulasi realistis untuk pembelajaran sistem penjualan.

### 3. Source Code

#### A. Fitur Login dan Register

**Login:** Fitur ini memungkinkan pengguna (admin atau user biasa) untuk masuk ke sistem dengan memasukkan nama dan NIM, dengan batas tiga kali percobaan; jika berhasil, pengguna diarahkan ke menu utama sesuai peran mereka, memastikan hanya pengguna terdaftar yang bisa mengakses sistem.

**Register sebagai User Baru:** Fitur ini memungkinkan pengguna baru mendaftar dengan nama dan NIM, otomatis ditetapkan sebagai user biasa (bukan admin), sehingga sistem mendukung multiuser hingga kapasitas 50 pengguna, memberikan fleksibilitas untuk menambah pengguna tanpa mengubah kode.

#### Source Code:

```
int main() {
    // Inisialisasi admin default
    if (jumlahUser == 0) {
        daftarUser[0].nama = "lian";
        daftarUser[0].nim = "001";
        daftarUser[0].isAdmin = true;
        daftarUser[0].jumlahProduk = 0;
        jumlahUser = 1;
    }

    while (true) { // Loop utama program
        string nama, nim;
        int percobaan = 0;
        int indexUser = -1;
        int pilihanAwal;

        // Menu Awal
        cout << "\n===== " << endl;
        cout << "=== Sistem Manajemen Handphone ===" << endl;
        cout << "===== " << endl;
        cout << "|1. Login                      |" << endl;
        cout << "|2. Register sebagai User Baru |" << endl;
        cout << "|3. Keluar                      |" << endl;
        cout << "===== " << endl;
        cout << "Pilih opsi (1-3): ";
        cin >> pilihanAwal;

        if (pilihanAwal == 2) { // Register User Baru
            if (jumlahUser >= 50) {
```

```

        cout << "Kapasitas user penuh!" << endl;
    } else {
        cout << "\n=== Registrasi User Baru ===" << endl;
        cout << "Masukkan Nama: ";
        cin.ignore();
        getline(cin, daftarUser[jumlahUser].nama);
        cout << "Masukkan NIM: ";
        cin >> daftarUser[jumlahUser].nim;
        daftarUser[jumlahUser].isAdmin = false;
        daftarUser[jumlahUser].jumlahProduk = 0;
        jumlahUser++;
        cout << "Registrasi berhasil! Silakan login." << endl;
    }
    continue;
} else if (pilihanAwal == 3) { // Keluar
    cout << "Program selesai!" << endl;
    return 0;
} else if (pilihanAwal != 1) {
    cout << "Pilihan tidak valid!" << endl;
    continue;
}

// Login
while (percobaan < 3) {
    cout << "\n=== Login Sistem ===" << endl;
    cout << "Masukkan Nama: ";
    cin.ignore();
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    cin >> nim;

    // Verifikasi Login
    bool loginBerhasil = false;
    for (int i = 0; i < jumlahUser; i++) {
        if (daftarUser[i].nama == nama && daftarUser[i].nim == nim) {
            indexUser = i;
            loginBerhasil = true;
            break;
        }
    }

    if (loginBerhasil) {
        cout << "Login berhasil!" << endl;
        break;
    } else {
        percobaan++;
        if (jumlahUser == 0) {
            cout << "Belum ada user terdaftar. Silakan register terlebih dahulu." << endl;
            break;
        } else {

```

```

        cout << "Login gagal. Sisa percobaan: " << (3 - percobaan)
    << endl;
    }
}

if (percobaan >= 3) {
    cout << "Percobaan login habis!" << endl;
    continue;
} else if (indexUser == -1) {
    continue;
}

```

## B. Fitur Tampilkan Data

Fitur ini tersedia untuk admin dan user biasa, menampilkan daftar handphone (kode, nama, merk, kategori, harga) yang dikelola admin dalam format tabel, memungkinkan semua pengguna melihat inventaris terkini atau pesan "Belum ada data" jika kosong

### Source Code:

```

case 1: // Tampilkan Data
    if (daftarUser[0].jumlahProduk == 0) {
        cout << "Belum ada data handphone!" << endl;
    } else {
        cout << "\n===== Data
Handphone =====" << endl;
        cout << setw(10) << "Kode" << setw(20) << "Nama" <<
        setw(15) << "Merk"
        << setw(20) << "Kategori" << setw(15) <<
        "Harga" << endl;
        cout << string(80, '-') << endl;
        for (int i = 0; i < daftarUser[0].jumlahProduk; i++)
        {
            cout << setw(10) <<
            daftarUser[0].daftarProduk[i].kode
            << setw(20) <<
            daftarUser[0].daftarProduk[i].nama
            << setw(15) <<
            daftarUser[0].daftarProduk[i].merk
            << setw(20) <<
            daftarUser[0].daftarProduk[i].kategori
            << setw(15) <<
            daftarUser[0].daftarProduk[i].harga << endl;
        }
    }
}

```

```

    }
}
break;

```

### C. Fitur Tambah Data

Fitur eksklusif admin ini memungkinkan penambahan data handphone baru ke dalam daftar admin (maksimum 100 produk), dengan input kode, nama, merk, kategori, dan harga

#### Source Code:

```

case 2: // Tambah Data
    if (daftarUser[0].jumlahProduk >= 100) {
        cout << "Kapasitas produk penuh!" << endl;
    } else {
        cout << "\n=== Tambah Data Handphone ===" << endl;
        int idx = daftarUser[0].jumlahProduk;
        cout << "Masukkan Kode Produk: ";
        cin >> daftarUser[0].daftarProduk[idx].kode;
        cout << "Masukkan Nama Produk: ";
        cin.ignore();
        getline(cin, daftarUser[0].daftarProduk[idx].nama);
        cout << "Masukkan Merk Produk: ";
        getline(cin, daftarUser[0].daftarProduk[idx].merk);
        cout << "Masukkan Kategori Produk: ";
        getline(cin,
daftarUser[0].daftarProduk[idx].kategori);
        cout << "Masukkan Harga: ";
        cin >> daftarUser[0].daftarProduk[idx].harga;
        daftarUser[0].jumlahProduk++;
        cout << "Data berhasil ditambahkan!" << endl;
    }
    break;

```

### D. Fitur Ubah Data

Fitur ini hanya untuk admin, memungkinkan perubahan detail handphone berdasarkan kode produk, menampilkan daftar saat ini sebelum meminta input baru untuk nama, merk, kategori, dan harga, memastikan data tetap akurat dan terbaru

#### Source Code:

```

case 3: // Ubah Data

```



```

        if (daftarUser[0].jumlahProduk == 0) {
            cout << "Belum ada data handphone!" << endl;
        } else {
            cout << "\n===== Data
Handphone =====" << endl;
            cout << setw(10) << "Kode" << setw(20) << "Nama" <<
setw(15) << "Merk"
<< setw(20) << "Kategori" << setw(15) <<
"Harga" << endl;
            cout << string(80, '-') << endl;
            for (int i = 0; i < daftarUser[0].jumlahProduk; i++)
            {
                cout << setw(10) <<
daftarUser[0].daftarProduk[i].kode
<< setw(20) <<
daftarUser[0].daftarProduk[i].nama
<< setw(15) <<
daftarUser[0].daftarProduk[i].merk
<< setw(20) <<
daftarUser[0].daftarProduk[i].kategori
<< setw(15) <<
daftarUser[0].daftarProduk[i].harga << endl;
            }
            string kode;
            cout << "\nMasukkan Kode Produk yang akan diubah: ";
            cin >> kode;
            bool ditemukan = false;
            for (int i = 0; i < daftarUser[0].jumlahProduk; i++)
            {
                if (daftarUser[0].daftarProduk[i].kode == kode)
                {
                    cout << "Masukkan Nama Produk baru: ";
                    cin.ignore();
                    getline(cin,
daftarUser[0].daftarProduk[i].nama);
                    cout << "Masukkan Merk Produk baru: ";
                    getline(cin,
daftarUser[0].daftarProduk[i].merk);
                    cout << "Masukkan Kategori Produk baru: ";
                    getline(cin,
daftarUser[0].daftarProduk[i].kategori);
                    cout << "Masukkan Harga baru: ";
                    cin >> daftarUser[0].daftarProduk[i].harga;
                    cout << "Data berhasil diubah!" << endl;
                    ditemukan = true;
                    break;
                }
            }
            if (!ditemukan) cout << "Kode produk tidak
ditemukan!" << endl;
        }
    }

```

```
break;
```

## E. Fitur Hapus Data

Fitur admin ini memungkinkan penghapusan handphone dari daftar berdasarkan kode produk, menggeser data untuk mengisi kekosongan, sehingga inventaris dapat dikurangi dengan mudah sesuai kebutuhan

### Source Code:

```
case 4: // Hapus Data
    if (daftarUser[0].jumlahProduk == 0) {
        cout << "Belum ada data handphone!" << endl;
    } else {
        cout << "\n===== Data
Handphone =====" << endl;
        cout << setw(10) << "Kode" << setw(20) << "Nama" <<
setw(15) << "Merk"
        << setw(20) << "Kategori" << setw(15) << "Harga" <<
endl;
        cout << string(80, '-') << endl;
        for (int i = 0; i < daftarUser[0].jumlahProduk; i++) {
            cout << setw(10) <<
daftarUser[0].daftarProduk[i].kode
            << setw(20) <<
daftarUser[0].daftarProduk[i].nama
            << setw(15) <<
daftarUser[0].daftarProduk[i].merk
            << setw(20) <<
daftarUser[0].daftarProduk[i].kategori
            << setw(15) <<
daftarUser[0].daftarProduk[i].harga << endl;
        }
        string kode;
        cout << "\nMasukkan Kode Produk yang akan dihapus: ";
        cin >> kode;
        bool ditemukan = false;
        for (int i = 0; i < daftarUser[0].jumlahProduk; i++) {
            if (daftarUser[0].daftarProduk[i].kode == kode) {
                for (int j = i; j < daftarUser[0].jumlahProduk -
1; j++) {
                    daftarUser[0].daftarProduk[j] =
daftarUser[0].daftarProduk[j + 1];
                }
            }
        }
    }
}
```

```

        daftarUser[0].jumlahProduk--;
        cout << "Produk berhasil dihapus!" << endl;
        ditemukan = true;
        break;
    }
}
if (!ditemukan) cout << "Kode produk tidak ditemukan!"
<< endl;
}
break;

```

## F. Fitur Beli Produk

Fitur untuk user biasa ini memungkinkan pembelian handphone dengan menghapus produk dari daftar admin berdasarkan kode, menampilkan daftar sebelum meminta input kode, mensimulasikan transaksi sederhana dengan pesan konfirmasi

### Source Code:

```

case 2: // Beli Produk
    if (daftarUser[0].jumlahProduk == 0) {
        cout << "Belum ada data handphone!" << endl;
    } else {
        cout << "\n===== Data
Handphone =====" << endl;
        cout << setw(10) << "Kode" << setw(20) << "Nama" <<
setw(15) << "Merk"
        << setw(20) << "Kategori" << setw(15) <<
"Harga" << endl;
        cout << string(80, '-') << endl;
        for (int i = 0; i < daftarUser[0].jumlahProduk; i++)
        {
            cout << setw(10) <<
daftarUser[0].daftarProduk[i].kode
            << setw(20) <<
daftarUser[0].daftarProduk[i].nama
            << setw(15) <<
daftarUser[0].daftarProduk[i].merk
            << setw(20) <<
daftarUser[0].daftarProduk[i].kategori
            << setw(15) <<
daftarUser[0].daftarProduk[i].harga << endl;
        }
        string kode;
        cout << "\nMasukkan Kode Produk yang ingin dibeli:

```

```

";
        cin >> kode;
        bool ditemukan = false;
        for (int i = 0; i < daftarUser[0].jumlahProduk; i++)
        {
            if (daftarUser[0].daftarProduk[i].kode == kode)
            {
                for (int j = i; j <
daftarUser[0].jumlahProduk - 1; j++) {
                    daftarUser[0].daftarProduk[j] =
daftarUser[0].daftarProduk[j + 1];
                }
                daftarUser[0].jumlahProduk--;
                cout << "Produk berhasil dibeli!" << endl;
                ditemukan = true;
                break;
            }
        }
        if (!ditemukan) cout << "Kode produk tidak
ditemukan!" << endl;
    }
    break;
}

```

#### 4. Uji Coba dan Hasil Output

```
=====
=== Sistem Manajemen Handphone ===
=====
|1. Login |
|2. Register sebagai User Baru |
|3. Keluar |
=====
Pilih opsi (1-3): 1

=== Login Sistem ===
Masukkan Nama: lian
Masukkan NIM: 001
Login berhasil!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data |
|2. Tambah Data |
|3. Ubah Data |
|4. Hapus Data |
|5. Logout |
=====
Pilih menu (1-5): █
```

Gambar 4.1 Login Admin

```

=====
=== Sistem Manajemen Handphone ===
=====
|1. Login |
|2. Register sebagai User Baru |
|3. Keluar |
=====
Pilih opsi (1-3): 2

=== Registrasi User Baru ===
Masukkan Nama: dwiki ihir
Masukkan NIM: 000
Registrasi berhasil! Silakan login.

=====
=== Sistem Manajemen Handphone ===
=====
|1. Login |
|2. Register sebagai User Baru |
|3. Keluar |
=====
Pilih opsi (1-3): 1

=== Login Sistem ===
Masukkan Nama: dwiki ihir
Masukkan NIM: 000
Login berhasil!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data |
|2. Beli Produk |
|3. Logout |
=====
Pilih menu (1-3): █

```

Gambar 4.2 Login User Baru

```

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data |
|2. Tambah Data |
|3. Ubah Data |
|4. Hapus Data |
|5. Logout |
=====
Pilih menu (1-5): 1
Belum ada data handphone!

```

Gambar 4.3 Tampilkan Data Jika Tidak Ada Data

```
=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Tambah Data        |
|3. Ubah Data          |
|4. Hapus Data         |
|5. Logout             |
=====
Pilih menu (1-5): 1

===== Data Handphone =====
      Kode      Nama      Merk      Kategori      Harga
-----
      00      ipon      apel      ipong      1500000

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Tambah Data        |
|3. Ubah Data          |
|4. Hapus Data         |
|5. Logout             |
=====
Pilih menu (1-5): █
```

Gambar 4.4 Tampilkan Data

```
=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Tambah Data         |
|3. Ubah Data           |
|4. Hapus Data          |
|5. Logout              |
=====
Pilih menu (1-5): 2

=== Tambah Data Handphone ===
Masukkan Kode Produk: 01
Masukkan Nama Produk: poko x100 promex
Masukkan Merk Produk: xiami
Masukkan Kategori Produk: hape
Masukkan Harga: 40000000
Data berhasil ditambahkan!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Tambah Data         |
|3. Ubah Data           |
|4. Hapus Data          |
|5. Logout              |
=====
Pilih menu (1-5): 1

===== Data Handphone =====
      Kode      Nama      Merk      Kategori      Harga
-----
      00      ipon      apel      ipong      1500000
      01      poko x100 promex      xiami      hape      40000000
```

Gambar 4.5 Tambah Data



```

|3. Ubah Data          |
|4. Hapus Data         |
|5. Logout             |
=====
Pilih menu (1-5): 3

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00             ipon          apel          ipong          1500000
      01      poko x100 promex      xiami          hape          40000000

Masukkan Kode Produk yang akan diubah: 00
Masukkan Nama Produk baru: ipong 18 pro
Masukkan Merk Produk baru: mangga
Masukkan Kategori Produk baru: ipon
Masukkan Harga baru: 2000000
Data berhasil diubah!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data     |
|2. Tambah Data        |
|3. Ubah Data          |
|4. Hapus Data         |
|5. Logout             |
=====
Pilih menu (1-5): 1

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00             ipong 18 pro      mangga          ipon          2000000
      01      poko x100 promex      xiami          hape          40000000

```

Gambar 4.6 Ubah Data

```
=====
|1. Tampilkan Data      |
|2. Tambah Data         |
|3. Ubah Data           |
|4. Hapus Data          |
|5. Logout              |
=====
Pilih menu (1-5): 4

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00      ipong 18 pro      mangga      ipon      2000000
      01      poko x100 promex      xiami      hape      40000000
      02      redmi kamera boba      xiami      hape      3000000

Masukkan Kode Produk yang akan dihapus: 02
Produk berhasil dihapus!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Tambah Data         |
|3. Ubah Data           |
|4. Hapus Data          |
|5. Logout              |
=====
Pilih menu (1-5): 1

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00      ipong 18 pro      mangga      ipon      2000000
      01      poko x100 promex      xiami      hape      40000000
```

Gambar 4.7 Hapus Data

```
=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Beli Produk        |
|3. Logout             |
=====
Pilih menu (1-3): 2

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00      ipong 18 pro      mangga      ipon      2000000
      01      poko x100 promex      xiami      hape      40000000

Masukkan Kode Produk yang ingin dibeli: 01
Produk berhasil dibeli!

=====
=== Manajemen Produk Handphone ===
=====
|1. Tampilkan Data      |
|2. Beli Produk        |
|3. Logout             |
=====
Pilih menu (1-3): 1

===== Data Handphone =====
      Kode          Nama          Merk          Kategori          Harga
-----
      00      ipong 18 pro      mangga      ipon      2000000
```

Gambar 4.8 Beli Produk

## 5. Langkah-Langkah Git pada VSCode

### 5.1 Git Init

Git init merupakan command yang berfungsi untuk menginisiasi repository yang ada pada file lokal dan berlokasi di folder .git

```
ASUS-GK@DESKTOP-M11TQOT MINGW64 ~/documents/github/praktikum-apl (main)
$ git init
Reinitialized existing Git repository in C:/Users/ASUS-GK/Documents/GitHub/praktikum-apl/.git/
```

Gambar 5.1 Git Init

### 5.2 Git Add

Menambahkan file yang ingin di commit.

```
ASUS-GK@DESKTOP-M11TQOT MINGW64 ~/documents/github/praktikum-apl/post-test/post-test-apl-3 (main)
$ git add 2409106001-DwikiAprilianAryanda-PT-3.cpp
```

Gambar 5.2 Git Add

### 5.3 Git Commit

Menyimpan perubahan yang disimpan pada folder lokal dengan sebuah pesan.

```
ASUS-GK@DESKTOP-M11TQOT MINGW64 ~/documents/github/praktikum-apl/post-test/post-test-apl-3 (main)
$ git commit -m "Finish Post Test 3"
[main 03de1dd] Finish Post Test 3
1 file changed, 331 insertions(+)
create mode 100644 post-test/post-test-apl-3/2409106001-DwikiAprilianAryanda-PT-3.cpp
```

Gambar 5.3 Git Commit

### 5.4 Git Push

Git push berfungsi untuk mengupload semua yang ada di lokal ke github.

```
ASUS-GK@DESKTOP-M11TQOT MINGW64 ~/documents/github/praktikum-apl/post-test/post-test-apl-3 (main)
$ git push origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.29 KiB | 783.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/DwikiAprilianAryanda/praktikum-apl.git
d9ddf3..03de1dd main -> main
```

Gambar 5.4 Git Push