

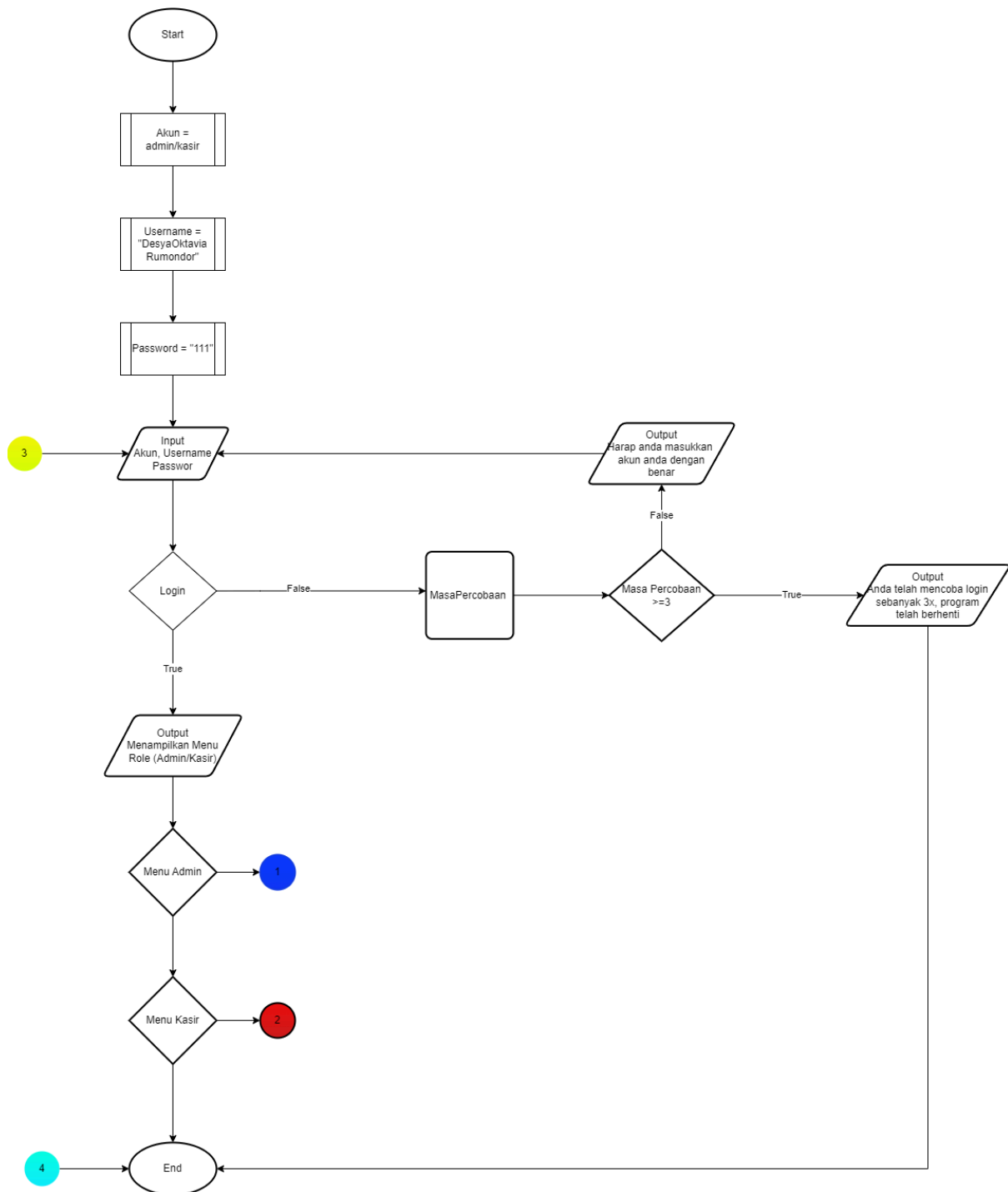
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT

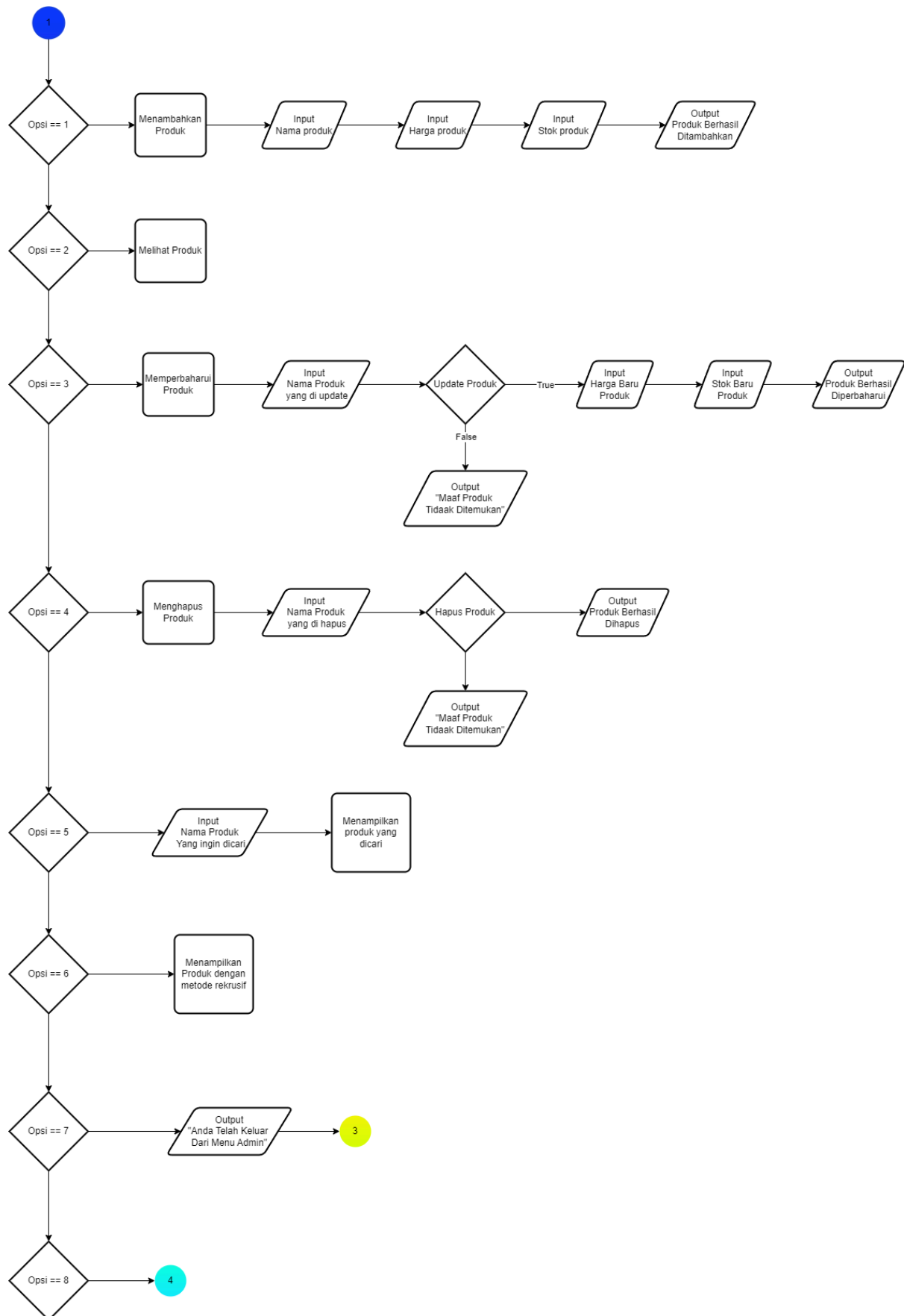


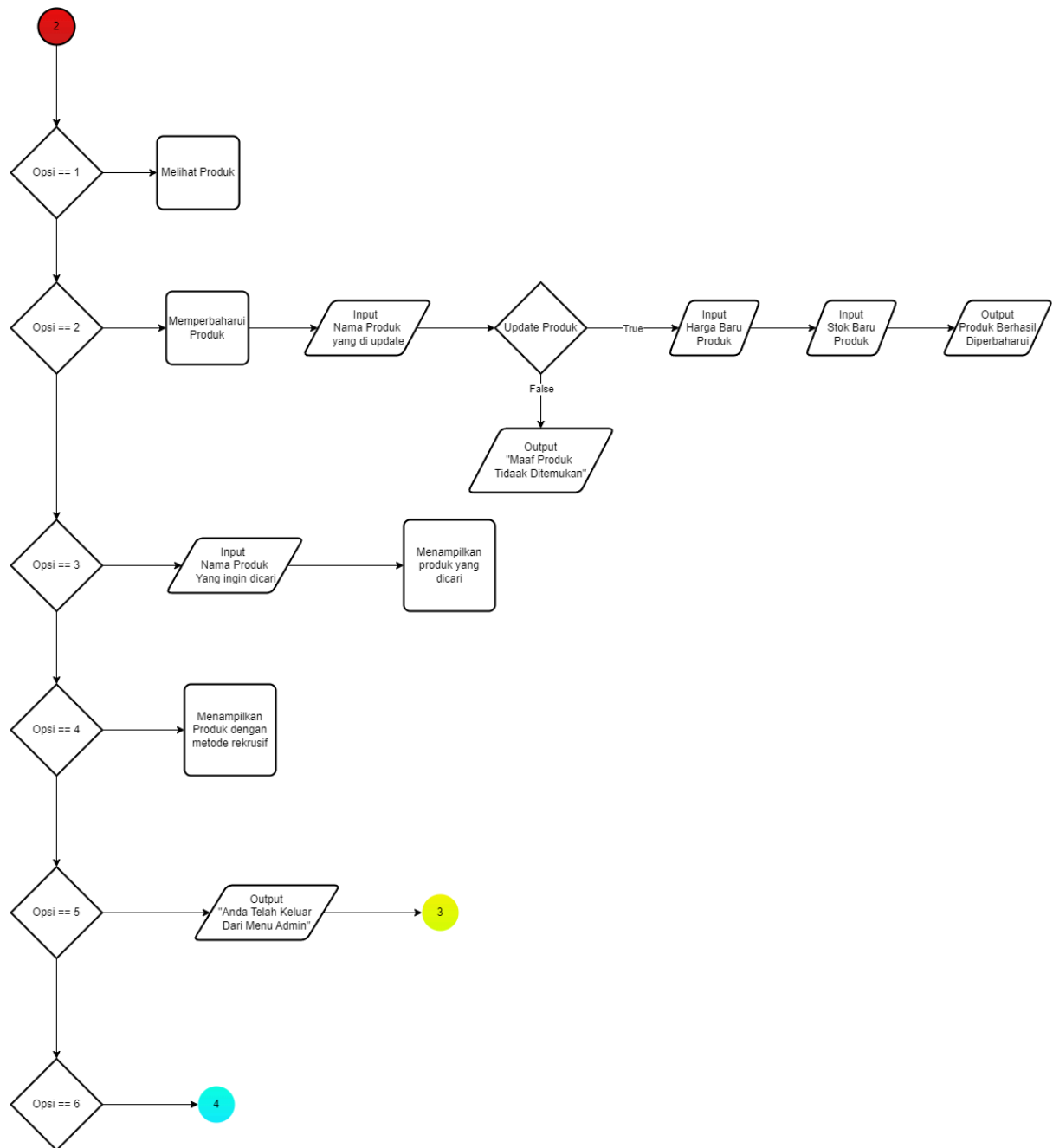
Disusun oleh:
Desya Oktavia Rumondor 2409106111
Kelas C2'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







2. Analisis Program

CRUD adalah Creat, Read, Update, dan Delete. Program ini dibuat agar bisa memudahkan manajemen penjualan pada suatu usaha dengan nantinya pendataan dalam suatu usaha yang dikelola dapat tersimpan secara aman. Program ini dapat digunakan untuk multiuser, dimana ada admin, kasir, user/pengguna, dengan sesuai kebutuhan dalam program masing-masing. Nantinya di setiap role memiliki fiturnya masing-masing.

Di program ini memiliki fitur login, dengan memasukkan akun, nama pengguna/user dan password pengguna yang sesuai dengan data yang tersimpan dalam program. Apabila login gagal, pengguna/users hanya memiliki 3 kali kesempatan untuk memasukkan inputan akun, nama, dan password dengan benar, jika sudah melewati batasnya maka program akan berhenti dan tidak berlanjut.

Setelah melewati fitur login, nantinya pengguna dengan role admin/kasir akan ditampilkan menu program, setelah itu pengguna dengan role admin/kasir diminta untuk menginput opsi pada menu program yang ditampilkan, setiap opsi nya masing-masing memiliki proses yang berbeda-beda pada tampilan CRUD yang tersedia di rolenya masing-masing. Dan pengguna dengan role admin/kasir telah selesai menjalankan program, bisa dapat keluar dari program.

3. Source Code

```
#include <iostream>

#include <string>

using namespace std;

#define MAX_PRODUK 250

struct Produk {

    string Nama;
```

```

    int Stok;

    double Harga;

};

struct Toko {

    Produk DaftarProduk[MAX_PRODUK];

    int JumlahSabun;

};

void TambahProdukPointer(Toko* toko);

void UpdateHarga(Produk* produk);

void LihatProduk(const Toko &toko);

void LihatProduk(const Toko &toko, const string &namaProduk);

bool UpdateProduk(Toko &toko, const string &nama);

bool HapusProduk(Toko &toko, const string &nama);

void CetakProdukRekursif(const Toko &toko, int index);

int main() {

    Toko toko;

    toko.JumlahSabun = 0;

    int MasaPercobaan = 0;

    while (true) {

```

```

string akun, username, password;

bool Admin = false;

cout << "\nLogin sebagai (admin/kasir): ";

cin >> akun;

cout << "Silahkan Masukkan Username Anda: ";

cin >> username;

cout << "Silahkan Masukkan Password Anda: ";

cin >> password;

if (akun == "admin" && username == "DesyaOktaviaRumondor" && password
== "111") {

    Admin = true;

    cout << "Anda Berhasil Login Sebagai Admin\n" << endl;

    MasaPercobaan = 0;

} else if (akun == "kasir" && password == "111") {

    Admin = false;

    cout << "Anda Berhasil Login Sebagai Kasir\n" << endl;

    MasaPercobaan = 0;

} else {

    cout << "Harap anda masukkan akun anda dengan benar!" << endl;

    MasaPercobaan++;

    if (MasaPercobaan >= 3) {

        cout << "Anda telah mencoba login sebanyak 3 kali. Program
telah berhenti!." << endl;

```

```

        return 0;

    }

    continue;

}

while (true) {

    if (Admin) {

        cout << "===== " << endl;

        cout << " |          Menu Admin          | " << endl;

        cout << "===== " << endl;

        cout << " | 1   | Tambah Produk           | " << endl;

        cout << " | 2   | Lihat Semua Produk        | " << endl;

        cout << " | 3   | Update Produk              | " << endl;

        cout << " | 4   | Hapus Produk               | " << endl;

        cout << " | 5   | Lihat Produk (Filter Nama) | " << endl;

        cout << " | 6   | Cetak Produk Rekursif      | " << endl;

        cout << " | 7   | Keluar dari Menu           | " << endl;

        cout << " | 8   | Keluar dari Program        | " << endl;

        cout << "===== " << endl;

    } else {

        cout << "===== " << endl;

        cout << " |          Menu Kasir          | " << endl;

        cout << "===== " << endl;

        cout << " | 1   | Lihat Semua Produk        | " << endl;

```



```

        cout << "| 2 | Update Produk |" << endl;

        cout << "| 3 | Lihat Produk (Filter Nama) |" << endl;

        cout << "| 4 | Cetak Produk Rekursif |" << endl;

        cout << "| 5 | Keluar dari Menu |" << endl;

        cout << "| 6 | Keluar dari Program |" << endl;

        cout << "===== " << endl;

    }

    int Opsi;

    cout << "Pilih menu: ";

    cin >> Opsi;

    if (Admin) {

        switch (Opsi) {

            case 1: TambahProdukPointer(&toko); break;

            case 2: LihatProduk(toko); break;

            case 3: {

                string nama;

                cin.ignore();

                cout << "Masukkan Nama Produk yang Ingin Diupdate: ";

                getline(cin, nama);

                if (!UpdateProduk(toko, nama))

                    cout << "Produk tidak ditemukan.\n";

                break;
            }
        }
    }

```

```

    }

    case 4: {

        string nama;

        cin.ignore();

        cout << "Masukkan Nama Produk yang Ingin Dihapus: ";

        getline(cin, nama);

        if (!HapusProduk(toko, nama))

            cout << "Produk tidak ditemukan.\n";

        break;

    }

    case 5: {

        string nama;

        cin.ignore();

        cout << "Masukkan Nama Produk yang Ingin Dicari: ";

        getline(cin, nama);

        LihatProduk(toko, nama);

        break;

    }

    case 6: {

        cout << "\n=== Cetak Produk Rekursif ===\n";

        CetakProdukRekursif(toko, 0);

        break;

    }

    case 7: cout << "Keluar dari menu admin.\n"; break;

```

```

        case 8: cout << "Keluar dari program.\n"; return 0;

        default: cout << "Pilihan tidak valid!\n";

    }

    if (Opsi == 7) break;

} else {

    switch (Opsi) {

        case 1: LihatProduk(toko); break;

        case 2: {

            string nama;

            cin.ignore();

            cout << "Masukkan Nama Produk yang Ingin Diupdate: ";

            getline(cin, nama);

            if (!UpdateProduk(toko, nama))

                cout << "Produk tidak ditemukan.\n";

            break;

        }

        case 3: {

            string nama;

            cin.ignore();

            cout << "Masukkan Nama Produk yang Ingin Dicari: ";

            getline(cin, nama);

            LihatProduk(toko, nama);

```

```

        break;

    }

    case 4: {

        cout << "\n=== Cetak Produk Rekursif ===\n";

        CetakProdukRekursif(toko, 0);

        break;

    }

    case 5: cout << "Keluar dari menu kasir.\n"; break;

    case 6: cout << "Keluar dari program.\n"; return 0;

    default: cout << "Pilihan tidak valid!\n";

    }

    if (Opsi == 5) break;

    }

    }

    }

    return 0;
}

void TambahProdukPointer(Toko* toko) {

    if (toko->JumlahSabun < MAX_PRODUK) {

        cout << "Silahkan Masukkan Nama Sabun: ";

        cin.ignore();

```

```

        getline(cin, toko->DaftarProduk[toko->JumlahSabun].Nama);

        cout << "Tetapkan untuk harga produk: ";

        cin >> toko->DaftarProduk[toko->JumlahSabun].Harga;

        cout << "Masukkan Jumlah Stok Sabun: ";

        cin >> toko->DaftarProduk[toko->JumlahSabun].Stok;

        toko->JumlahSabun++;

        cout << "Produk berhasil ditambahkan!\n";
    } else {

        cout << "Stok penuh!\n";

    }
}

void UpdateHarga(Produk* produk) {

    cout << "Masukkan Harga Baru: ";

    cin >> produk->Harga;

    cout << "Masukkan Stok Baru: ";

    cin >> produk->Stok;

    cout << "Produk berhasil diupdate!\n";

}

void LihatProduk(const Toko &toko) {

    cout << "\n=== Daftar Produk ===" << endl;

    for (int i = 0; i < toko.JumlahSabun; i++) {

        cout << "Nama: " << toko.DaftarProduk[i].Nama

```

```

        << ", Harga: " << toko.DaftarProduk[i].Harga
        << ", Stok: " << toko.DaftarProduk[i].Stok << endl;
    }
}

void LihatProduk(const Toko &toko, const string &namaProduk) {
    bool found = false;

    for (int i = 0; i < toko.JumlahSabun; i++) {
        if (toko.DaftarProduk[i].Nama == namaProduk) {
            cout << "Nama: " << toko.DaftarProduk[i].Nama
                << ", Harga: " << toko.DaftarProduk[i].Harga
                << ", Stok: " << toko.DaftarProduk[i].Stok << endl;

            found = true;
        }
    }

    if (!found) cout << "Produk tidak ditemukan.\n";
}

bool UpdateProduk(Toko &toko, const string &nama) {
    for (int i = 0; i < toko.JumlahSabun; i++) {
        if (toko.DaftarProduk[i].Nama == nama) {
            UpdateHarga(&toko.DaftarProduk[i]);

            return true;
        }
    }
}

```

```

    }

    return false;
}

bool HapusProduk(Toko & toko, const string & nama) {

    for (int i = 0; i < toko.JumlahSabun; i++) {

        if (toko.DaftarProduk[i].Nama == nama) {

            for (int j = i; j < toko.JumlahSabun - 1; j++) {

                toko.DaftarProduk[j] = toko.DaftarProduk[j + 1];

            }

            toko.JumlahSabun--;

            cout << "Produk berhasil dihapus!\n";

            return true;

        }

    }

    return false;
}

void CetakProdukRekursif(const Toko & toko, int index) {

    if (index >= toko.JumlahSabun) return;

    cout << "Nama: " << toko.DaftarProduk[index].Nama

        << ", Harga: " << toko.DaftarProduk[index].Harga

        << ", Stok: " << toko.DaftarProduk[index].Stok << endl;

    CetakProdukRekursif(toko, index + 1);
}

```

```
}
```

4. Uji Coba dan Hasil Output

4.1 Admin

- Login

```
Login sebagai (admin/kasir): admin
Silahkan Masukkan Username Anda: DesyaOktaviaRumondor
Silahkan Masukkan Password Anda: 111
Anda Berhasil Login Sebagai Admin
```

```
Login sebagai (admin/kasir): kasir
Silahkan Masukkan Username Anda: memet
Silahkan Masukkan Password Anda: 111
Anda Berhasil Login Sebagai Kasir
```

- Menu

```
=====
|                               Menu Admin                               |
=====
| 1 | | Tambah Produk |
| 2 | | Lihat Semua Produk |
| 3 | | Update Produk |
| 4 | | Hapus Produk |
| 5 | | Lihat Produk (Filter Nama) |
| 6 | | Cetak Produk Rekursif |
| 7 | | Keluar dari Menu |
| 8 | | Keluar dari Program |
=====
Pilih menu: |
```

- Tambah Produk


```

=====
|           Menu Admin           |
=====
| 1 | Tambah Produk              |
| 2 | Lihat Semua Produk         |
| 3 | Update Produk              |
| 4 | Hapus Produk               |
| 5 | Lihat Produk (Filter Nama) |
| 6 | Cetak Produk Rekursif      |
| 7 | Keluar dari Menu           |
| 8 | Keluar dari Program        |
=====
Pilih menu: 3
Masukkan Nama Produk yang Ingin Diupdate: Coffe
Masukkan Harga Baru: 55.000
Masukkan Stok Baru: 20
Produk berhasil diupdate!

```

- **Hapus Produk**

```

=====
|           Menu Admin           |
=====
| 1 | Tambah Produk              |
| 2 | Lihat Semua Produk         |
| 3 | Update Produk              |
| 4 | Hapus Produk               |
| 5 | Lihat Produk (Filter Nama) |
| 6 | Cetak Produk Rekursif      |
| 7 | Keluar dari Menu           |
| 8 | Keluar dari Program        |
=====
Pilih menu: 4
Masukkan Nama Produk yang Ingin Dihapus: Coffe
Produk berhasil dihapus!

```

- **Lihat Produk (Filter nama)**

```

=====
|           Menu Admin           |
=====
| 1 | Tambah Produk |
| 2 | Lihat Semua Produk |
| 3 | Update Produk |
| 4 | Hapus Produk |
| 5 | Lihat Produk (Filter Nama) |
| 6 | Cetak Produk Rekursif |
| 7 | Keluar dari Menu |
| 8 | Keluar dari Program |
=====
Pilih menu: 5
Masukkan Nama Produk yang Ingin Dicari: Honey
Nama: Honey, Harga: 40, Stok: 15

```

- **Cetak Produk Rekrusif**

```

=====
|           Menu Admin           |
=====
| 1 | Tambah Produk |
| 2 | Lihat Semua Produk |
| 3 | Update Produk |
| 4 | Hapus Produk |
| 5 | Lihat Produk (Filter Nama) |
| 6 | Cetak Produk Rekursif |
| 7 | Keluar dari Menu |
| 8 | Keluar dari Program |
=====
Pilih menu: 6

=== Cetak Produk Rekursif ===
Nama: Lavender, Harga: 65, Stok: 12
Nama: Sunflower, Harga: 65, Stok: 7
Nama: Honey, Harga: 40, Stok: 15

```

- **Keluar Dari Menu**

```
=====
|                               |
|           Menu Admin        |
|                               |
|-----|-----|
| 1      | Tambah Produk          |
| 2      | Lihat Semua Produk     |
| 3      | Update Produk          |
| 4      | Hapus Produk           |
| 5      | Lihat Produk (Filter Nama) |
| 6      | Cetak Produk Rekursif  |
| 7      | Keluar dari Menu       |
| 8      | Keluar dari Program    |
|-----|-----|
Pilih menu: 7
Keluar dari menu admin.
```

- **Keluar Program**

```
=====
|                               |
|           Menu Admin         |
|                               |
|=====|=====|=====|
| 1      | Tambah Produk              |
| 2      | Lihat Semua Produk        |
| 3      | Update Produk             |
| 4      | Hapus Produk              |
| 5      | Lihat Produk (Filter Nama)|
| 6      | Cetak Produk Rekursif     |
| 7      | Keluar dari Menu          |
| 8      | Keluar dari Program       |
|=====|=====|=====|

Pilih menu: 8
Keluar dari program.
PS C:\Users\ASUS>
```

5. Langkah-Langkah Git pada VSCode

```
MINGW64:/d/GIT HUB/Praktikum-APL/Post-test/Post-test-5

ASUS@LAPTOP-IQEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL/Post-test/Post-test-5 (mai
n)
$ git add .

ASUS@LAPTOP-IQEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL/Post-test/Post-test-5 (mai
n)
$ git commit -m "post-test-5"
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

ASUS@LAPTOP-IQEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL/Post-test/Post-test-5 (mai
n)
$ git push
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 12 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (18/18), 1.79 MiB | 826.00 KiB/s, done.
Total 18 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/DesyaOktavia/Praktikum-APL.git
   0d476d1..4cb1f4f  main -> main

ASUS@LAPTOP-IQEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL/Post-test/Post-test-5 (mai
n)
$
```