

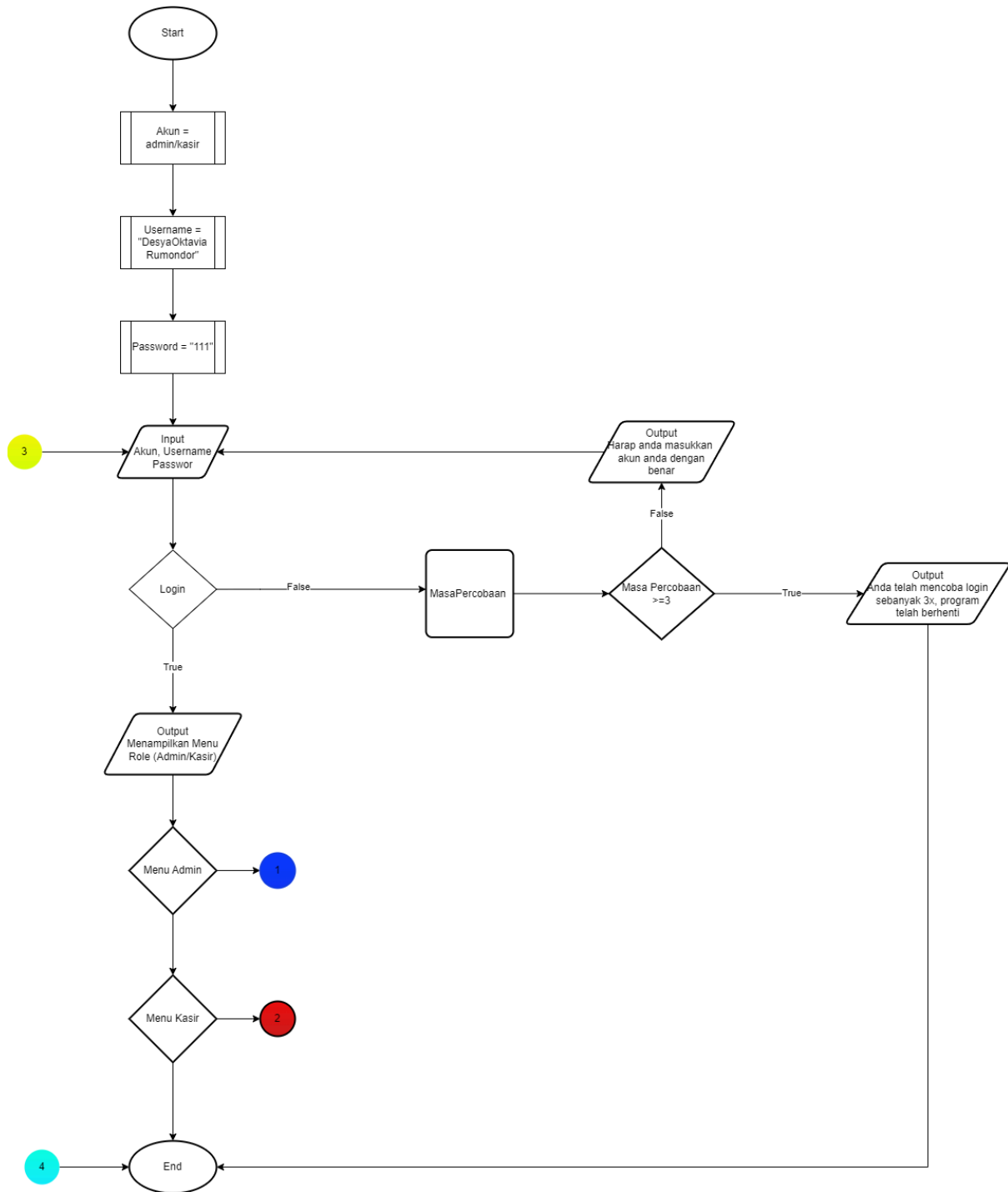
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT

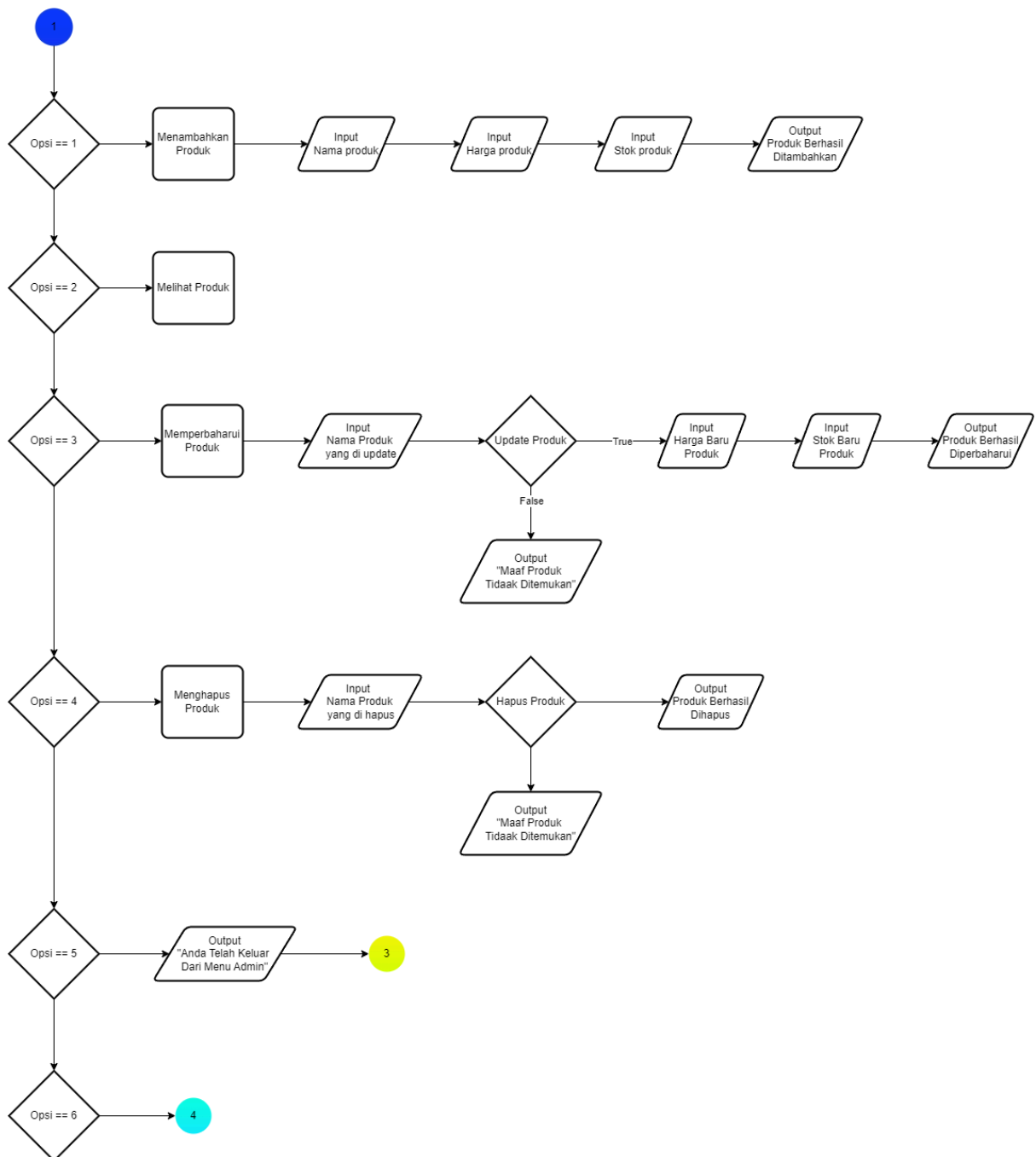


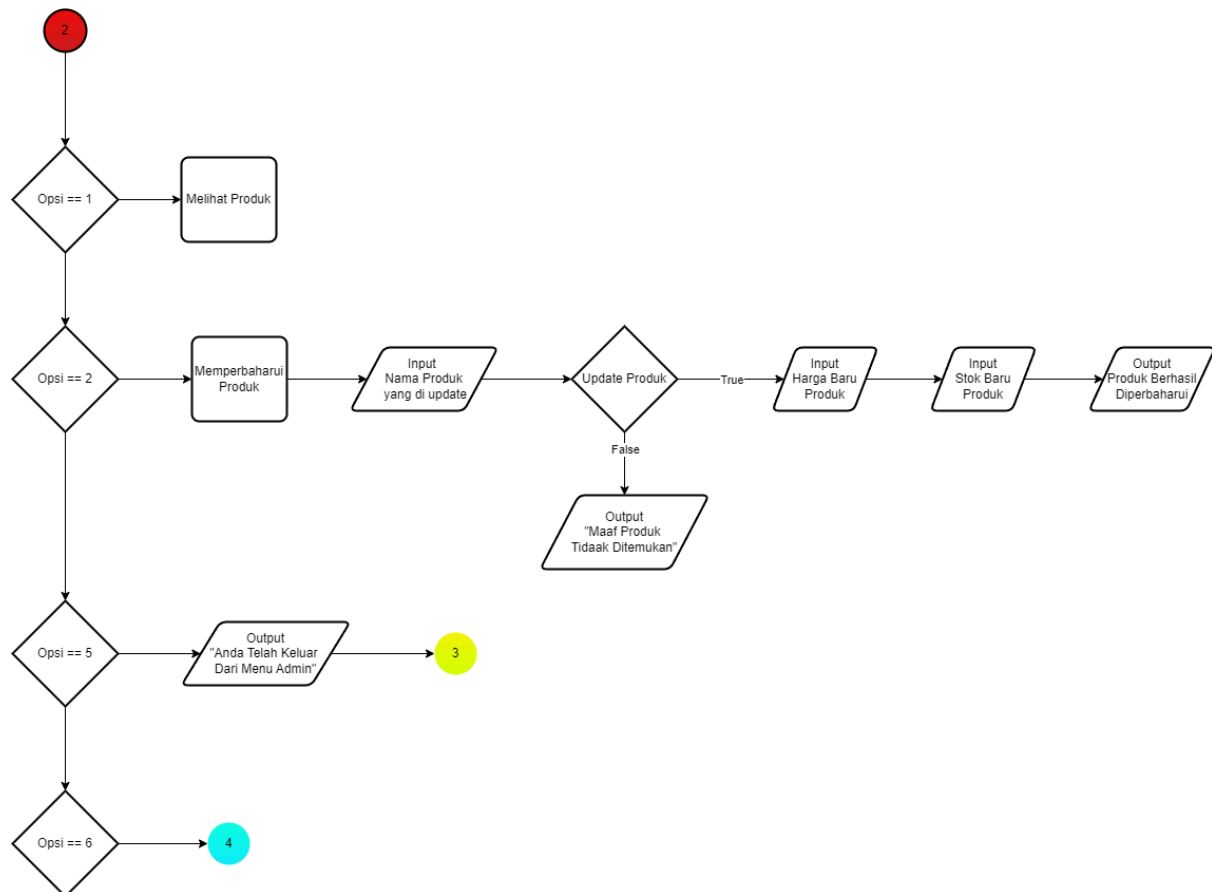
Disusun oleh:
Nama (2409106111)
Kelas (C2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart







2. Analisis Program

2.1 Deskripsi Singkat Program

Tujuan;

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.

Fungsi dan Manfaatnya;

1. Fitur Login

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.

2. Menampilkan Menu Program

Setelah melewati fitur login, nantinya pengguna dengan role admin/kasir akan ditampilkan menu program pada Admin memiliki CRUD (Creat, Read, Update, Delete) dan Kasir Tersedia Read dan Update.

3. Menginput Menu Program

Nantinya 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.

4. Keluar program

Jika pengguna dengan role admin/kasir telah selesai menjalankan program, bisa dapat keluar dari program.

2.2 Penjelasan Alur & Algoritma

Tahapan-tahapan utama pada program ini adalah adanya fitur **login role, validasi akun dan password, menampilkan menu program dan proses pada setiap opsi menu program**. Berikut adalah alur kerja program dari awal hingga akhir:

1. Fitur Login Role

Pada fitur ini nantinya pengguna akan diminta untuk login dengan memasukkan akun, nama/username dan password yang sudah tersimpan pada program dengan tipe data string, nantinya agar hanya pengguna/users yang terdata yang bisa dapat mengakses atau menjalankan program ini. Jika pengguna/users gagal melakukan login, nantinya program ini akan meminta kembali inputan username dan password hingga 3 kali percobaan.

```
string akun, username, password;  
bool Admin = false;
```

```

cout << "Login sebagai (admin/kasir): ";

cin >> akun;

cout << "Silahkan Masukkan Username Anda: ";

cin >> username;

cout << "Silahkan Masukkan Password Anda: ";

cin >> password;

```

2. Validasi Akun dan Password

Pada fitur ini inputan pengguna akan di cek apakah benar dan salah pada data yang tersimpan dengan masa percobaan 3 kali, jika sudah lebih dari 3 kali maka program akan berhenti. Masa percobaan ialah banyaknya kesempatan pengguna dengan role admin/kasir untuk menginput akun, nama, dan password dengan benar. Dan jika benar maka program akan melanjutkan programnya dengan menampilkan menu program dan jika, salah maka program akan kembali meminta inputan pengguna hingga dapat login menjadi admin atau kasir dengan maks 3 kali.

```

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;

}

```

3. Menampilkan Menu Program

Pada fitur ini program akan menampilkan menu opsi (pilihan) dan setiap role memiliki menu yang berbeda, yaitu untuk admin dan untuk kasir. Jika pengguna menginput admin maka program akan menampilkan menu CRUD (Creat, Read, Update dan Delete) produk, logout biasa (keluar dari role), dan logout program, sedangkan jika pengguna menginput kasir maka program akan menampilkan menu Read, Update, dan logout biasa (keluar dari program), dan logout program.

```

if (Admin) {

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

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

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

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

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

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

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

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

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

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

} else {

```



```

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

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

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

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

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

cout << "| 3  | Keluar                 |" << endl;

cout << "| 4  | Keluar dari Program    |" << endl;

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

```

4. Proses Pada Setiap Opsi Menu Program

A. Admin

- Creat

Fitur Creat adalah untuk menambah produk, dengan adanya create data akan disimpan bila sudah menambahkan data ke dalam program.

```

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 Telah berhasil ditambahkan!" << endl;
}

```

```

} else {

    cout << "Maaf, Stok Penyimpanan Produk Sudah Penuh!" << endl;

}

```

- **Read**

Fitur Read ialah untuk membaca data yang sudah tersimpan setelah ditambahkan lewat create atau yang telah di update.

```

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;

```

- **Update**

Fitur Update ialah untuk mengupdate data atau memperbaharui data, biasa nya data diperbaharui karena memiliki stok baru atau barang baru nantinya barang lama atau stok lama akan diperbaharui, dan akan ditampilkan di dalam read.

```

string namasabun;

cout << "Silahkan Masukkan Nama Produk Yang Diupdate: ";

cin.ignore();

getline(cin, namasabun);

bool Update = false;

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

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

        cout << "Silahkan Masukkan Harga Baru: ";

        cin >> toko.DaftarProduk[i].Harga;

        cout << "Masukkan Jumlah Stok Baru: ";

        cin >> toko.DaftarProduk[i].Stok;

```

```

        cout << "Produk Telah Berhasil Diupdate!" << endl;

        Update = true;

        break;
    }
}

if (!Update) {

    cout << "Maaf, Produk Tidak Ditemukan!" << endl;
}

```

- **Delete**

Fitur Delete ialah menghapus data. Dimana jika ada data atau barang yang sudah tidak diproduksi atau stoknya habis maka bisa menggunakan fitur delete untuk menghapusnya, dan akan diperbaharui dalam fitur read untuk barang yang sudah dihapus tidak akan ditampilkan lagi.

```

string namasabun;

cout << "Masukkan nama produk yang ingin dihapus: ";

cin.ignore();

getline(cin, namasabun);

bool Update = false;

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

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

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

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

        }

        toko.JumlahSabun--;

        cout << "Produk berhasil dihapus!" << endl;

        Update = true;
    }
}

```

```

        break;

    }

}

if (!Update) {

    cout << "Produk tidak ditemukan!" << endl;

```

- **Keluar**

Untuk menu keluar pada admin ialah menu yang nantinya akan mengeluarkan admin dalam fitur admin, setelah keluar masih dapat login kembali sebagai admin atau kasir dengan tanpa benar-benar keluar dari program.

```

cout << "Keluar dari menu admin." << endl;

break;

```

- **Keluar Dari Program**

Untuk menu keluar program pada admin ialah menu membuat pengguna atau admin benar-benar bisa dapat keluar dari program tanpa bisa menginput kembali role atau login ulang.

```

    cout << "Anda telah keluar dari program, terimakasih" << endl;

    return 0;

} else {

    cout << "Pilihan tidak valid!" << endl;

```

B. Kasir

- **Read**

Jadi di dalam menu program kasir hanya tersedia beberapa saja, salah satunya read. yaitu membaca data yang sudah disimpan di data admin, sebagai kasir akan dapat membaca juga data yang telah tersimpan oleh admin.

```
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;
```

- **Update**

Fitur kedua pada menu program kasir ialah update, dimana kasir nantinya dapat mengambil peran untuk memperbaharui data, dengan bisa melihat apakah stok yang tersedia sudah habis atau stok yang tersedia menambah, serta kasir juga bisa memperbaharui produk, jika ada produk baru .

```
string namesabun;

cout << "Silahkan Masukkan Nama Produk Yang Diupdate: ";

cin.ignore();

getline(cin, namesabun);

bool found = false;

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

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

        cout << "Silahkan Masukkan Harga Baru: ";

        cin >> toko.DaftarProduk[i].Harga;

        cout << "Masukkan Jumlah Stok Baru: ";

        cin >> toko.DaftarProduk[i].Stok;

        cout << "Produk Telah Berhasil Diupdate!" << endl;
```

```

        found = true;

        break;
    }
}

if (!found) {

    cout << "Maaf, Produk Tidak Ditemukan!" << endl;
}

```

- **Keluar**

Pada kasir memiliki fitur keluar atau log out dimana nantinya setelah keluar kasir tetap bisa login kembali menjadi kasir tanpa benar-benar keluar dari program, dengan keuntungan data yang sudah tersimpan akan tetap tersimpan.

```

cout << "Keluar dari menu kasir." << endl;

break

```

- **Keluar Dari Program**

Untuk menu keluar program pada admin ialah menu membuat pengguna atau admin benar-benar bisa dapat keluar dari program tanpa bisa menginput kembali role atau login ulang.

```

    cout << "Anda telah keluar dari program, terimakasih" << endl;

    return 0;
} else {

    cout << "Pilihan tidak valid!" << endl;
}

```

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;

};

int main() {

    Toko toko;

    toko.JumlahSabun = 0;

    int MasaPercobaan = 0;

    while (true) {

        string akun, username, password;

        bool Admin = false;
    }

```

```

    cout << "Login 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 Produk           |" << endl;

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

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

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

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

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

    } else {

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

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

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

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

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

        cout << "| 3  |  Keluar                 |" << endl;

        cout << "| 4  |  Keluar dari Program     |" << endl;

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

    }

}

```

```

    int Opsi;

    cout << "Pilih menu: ";

    cin >> Opsi;

    if (Admin) {

        if (Opsi == 1) {

            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 Telah berhasil ditambahkan!" << endl;

            } else {

                cout << "Maaf, Stok Penyimpanan Produk Sudah Penuh!" <<
endl;

            }

        } else if (Opsi == 2) {

            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;

```

```

    }

    } else if (Opsii == 3) {

        string namasabun;

        cout << "Silahkan Masukkan Nama Produk Yang Diupdate: ";

        cin.ignore();

        getline(cin, namasabun);

        bool Update = false;

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

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

                cout << "Silahkan Masukkan Harga Baru: ";

                cin >> toko.DaftarProduk[i].Harga;

                cout << "Masukkan Jumlah Stok Baru: ";

                cin >> toko.DaftarProduk[i].Stok;

                cout << "Produk Telah Berhasil Diupdate!" << endl;

                Update = true;

                break;

            }

        }

        if (!Update) {

            cout << "Maaf, Produk Tidak Ditemukan!" << endl;

        }

    } else if (Opsii == 4) {

```

```

    string namasabun;

    cout << "Masukkan nama produk yang ingin dihapus: ";

    cin.ignore();

    getline(cin, namasabun);

    bool Update = false;

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

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

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

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

            }

            toko.JumlahSabun--;

            cout << "Produk berhasil dihapus!" << endl;

            Update = true;

            break;

        }

    }

    if (!Update) {

        cout << "Produk tidak ditemukan!" << endl;

    }

} else if (Opsii == 5) {

    cout << "Keluar dari menu admin." << endl;

    break;

} else if (Opsii == 6) {

```

```

        cout << "Anda telah keluar dari program, terimakasih" << endl;

        return 0;

    } else {

        cout << "Pilihan tidak valid!" << endl;

    }

} else {

    if (Opsii == 1) {

        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;

        }

    }

    } else if (Opsii == 2) {

        string namasabun;

        cout << "Silahkan Masukkan Nama Produk Yang Diupdate: ";

        cin.ignore();

        getline(cin, namasabun);

        bool found = false;

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

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

                cout << "Silahkan Masukkan Harga Baru: ";

                cin >> toko.DaftarProduk[i].Harga;

```

```

        cout << "Masukkan Jumlah Stok Baru: ";

        cin >> toko.DaftarProduk[i].Stok;

        cout << "Produk Telah Berhasil Diupdate!" << endl;

        found = true;

        break;
    }

}

if (!found) {

    cout << "Maaf, Produk Tidak Ditemukan!" << endl;

}

} else if (Opsi == 3) {

    cout << "Keluar dari menu kasir." << endl;

    break;

} else if (Opsi == 4) {

    cout << "Anda telah keluar dari program, terimakasih" << endl;

    return 0;

} else {

    cout << "Pilihan tidak valid!" << endl;

}

}

}

}

return 0;

}

```

4. Uji Coba dan Hasil Output

1. ADMIN

1) Login Admin

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

Gambar 4.1.1

2) Menu Admin

```
=====
|                               |
|           Menu Admin        |
|                               |
=====
| 1   | Tambah Produk          |
| 2   | Lihat Produk            |
| 3   | Update Produk           |
| 4   | Hapus Produk             |
| 5   | Keluar                    |
| 6   | Keluar dari Program      |
|     |                           |
=====
Pilih menu:
```

Gambar 4.1.2

3) Tambah Produk

```
=====
|                               |
|           Menu Admin         |
|                               |
=====
| 1 | Tambah Produk            |
| 2 | Lihat Produk             |
| 3 | Update Produk            |
| 4 | Hapus Produk             |
| 5 | Keluar                   |
| 6 | Keluar dari Program      |
|                               |
=====
Pilih menu: 1
Silahkan Masukkan Nama Sabun: Seven Cactus Coffe and Cream Soap
Tetapkan untuk harga produk: 60.000
Masukkan Jumlah Stok Sabun: 30
Produk Telah berhasil ditambahkan!
```

Gambar 4.1.3

4) Lihat Produk

```
=====
|                               Menu Admin                               |
=====
| 1 | | Tambah Produk |
| 2 | | Lihat Produk  |
| 3 | | Update Produk |
| 4 | | Hapus Produk  |
| 5 | | Keluar        |
| 6 | | Keluar dari Program |
=====
Pilih menu: 2

=== Daftar Produk ===
Nama: Seven Cactus Coffe and Cream Soap, Harga: 60, Stok: 30
=====
```

Gambar 4.1.4

5) Update Produk

```
=====
|                               Menu Admin                               |
=====
| 1 | | Tambah Produk |
| 2 | | Lihat Produk  |
| 3 | | Update Produk |
| 4 | | Hapus Produk  |
| 5 | | Keluar        |
| 6 | | Keluar dari Program |
=====
Pilih menu: 3
Silahkan Masukkan Nama Produk Yang Diupdate: Seven Cactus Coffe and Cream Soap
Silahkan Masukkan Harga Baru: 55.000
Masukkan Jumlah Stok Baru: 50
Produk Telah Berhasil Diupdate!
```

Gambar 4.1.5

6) Delete Produk


```
=====
|           Menu Admin           |
=====
| 1 | Tambah Produk |
| 2 | Lihat Produk  |
| 3 | Update Produk |
| 4 | Hapus Produk  |
| 5 | Keluar        |
| 6 | Keluar dari Program |
=====
Pilih menu: 4
Masukkan nama produk yang ingin dihapus: Seven Cactus Coffe and Cream Soap
Produk berhasil dihapus!
```

Gambar 4.1.6

7) Keluar

```
=====
|           Menu Admin           |
=====
| 1 | Tambah Produk |
| 2 | Lihat Produk  |
| 3 | Update Produk |
| 4 | Hapus Produk  |
| 5 | Keluar        |
| 6 | Keluar dari Program |
=====
Pilih menu: 5
Keluar dari menu admin.
```

Gambar 4.1.7

8) Keluar Dari Program

```
=====
|           Menu Admin           |
=====
| 1 | Tambah Produk |
| 2 | Lihat Produk  |
| 3 | Update Produk |
| 4 | Hapus Produk  |
| 5 | Keluar        |
| 6 | Keluar dari Program |
=====
Pilih menu: 6
Anda telah keluar dari program, terimakasih
```

Gambar 4.1.8

2. KASIR

1) Login Kasir

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

Gambar 4.2.1

2) Menu Kasir

```
=====
|                               |
|           Menu Kasir        |
|                               |
| 1 | Lihat Produk            |
| 2 | Update Produk           |
| 3 | Keluar                  |
| 4 | Keluar dari Program     |
|                               |
|                               |
Pilih menu: |
```

Gambar 4.2.2

3) Lihat Produk

```
=====
|                               |
|           Menu Kasir        |
|                               |
| 1 | Lihat Produk            |
| 2 | Update Produk           |
| 3 | Keluar                  |
| 4 | Keluar dari Program     |
|                               |
|                               |
Pilih menu: 1

=== Daftar Produk ===
Nama: Seven Cactus Coffe and Cream Soap, Harga: 60, Stok: 30
```

Gambar 4.2.3

4) Update Produk

```
=====
|           Menu Kasir           |
=====
| 1 | Lihat Produk |
| 2 | Update Produk |
| 3 | Keluar       |
| 4 | Keluar dari Program |
=====
Pilih menu: 2
Silahkan Masukkan Nama Produk Yang Diupdate: Seven Cactus Coffe and Cream Soap
Silahkan Masukkan Harga Baru: 55.000
Masukkan Jumlah Stok Baru: 60
Produk Telah Berhasil Diupdate!
```

Gambar 4.2.4

5) Keluar

```
=====
|           Menu Kasir           |
=====
| 1 | Lihat Produk |
| 2 | Update Produk |
| 3 | Keluar       |
| 4 | Keluar dari Program |
=====
Pilih menu: 3
Keluar dari menu kasir.
```

Gambar 4.2.5

6) Keluar Program

```
=====
|           Menu Kasir           |
=====
| 1 | Lihat Produk |
| 2 | Update Produk |
| 3 | Keluar       |
| 4 | Keluar dari Program |
=====
Pilih menu: 4
Anda telah keluar dari program, terimakasih
```

Gambar 4.2.6

5. Langkah-Langkah Git pada VSCode

```
ASUS@LAPTOP-1QEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL (main)
$ git add .

ASUS@LAPTOP-1QEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL (main)
$ git commit -m "Post-test-3"
[main 0a8699d] Post-test-3
12 files changed, 508 insertions(+), 460 deletions(-)
create mode 100644 Post-test/Post-test-1/.vscode/tasks.json
delete mode 100644 Post-test/Post-test-1/2409106111-DesyaOktaviaRumondor-PT-1.D
rawio
rename {Post-test-1 => Post-test/Post-test-1}/2409106111-DesyaOktaviaRumondor-P
T-1.cpp.cpp (96%)
rename {Post-test-1 => Post-test/Post-test-1}/2409106111-DesyaOktaviaRumondor-P
T-1.exe (100%)
rename {Post-test-1 => Post-test/Post-test-1}/2409106111-DesyaOktaviaRumondor-P
T-1.pdf.pdf (100%)
create mode 100644 Post-test/Post-test-2/.vscode/settings.json
rename {Post-test-2 => Post-test/Post-test-2}/2409106111-DesyaOktaviaRumondor-P
T-2.pdf (100%)
rename {Post-test-2 => Post-test/Post-test-2}/2409106111_DesyaOktaviaRumondor_P
T_2.cpp (95%)
rename {Post-test-2 => Post-test/Post-test-2}/2409106111_DesyaOktaviaRumondor_P
T_2.exe (100%)
create mode 100644 Post-test/Post-test-3/.vscode/tasks.json
create mode 100644 Post-test/Post-test-3/2409106111_DesyaOktaviaRumondor_PT_3.c
pp
create mode 100644 Post-test/Post-test-3/2409106111_DesyaOktaviaRumondor_PT_3.e
xe

ASUS@LAPTOP-1QEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL (main)
$ git push
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (18/18), 1.98 MiB | 890.00 KiB/s, done.
Total 18 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/DesyaOktavia/Praktikum-APL.git
e3b80a5..0a8699d main -> main

ASUS@LAPTOP-1QEB2S9C MINGW64 /d/GIT HUB/Praktikum-APL (main)
$
```

- **Git add**

Digunakan untuk menambahkan semua perubahan (file baru, perubahan, dan penghapusan) ke dalam staging area. Ini mempersiapkan perubahan sebelum di-commit.

- **Git Commit -m "Post-test-3"**

Menyimpan perubahan ke repository dengan commit dan pesan update

- **Git Push**

Digunakan untuk mengunggah (upload) perubahan yang telah Anda commit di repositori lokal (local repository) ke repositori remote.