

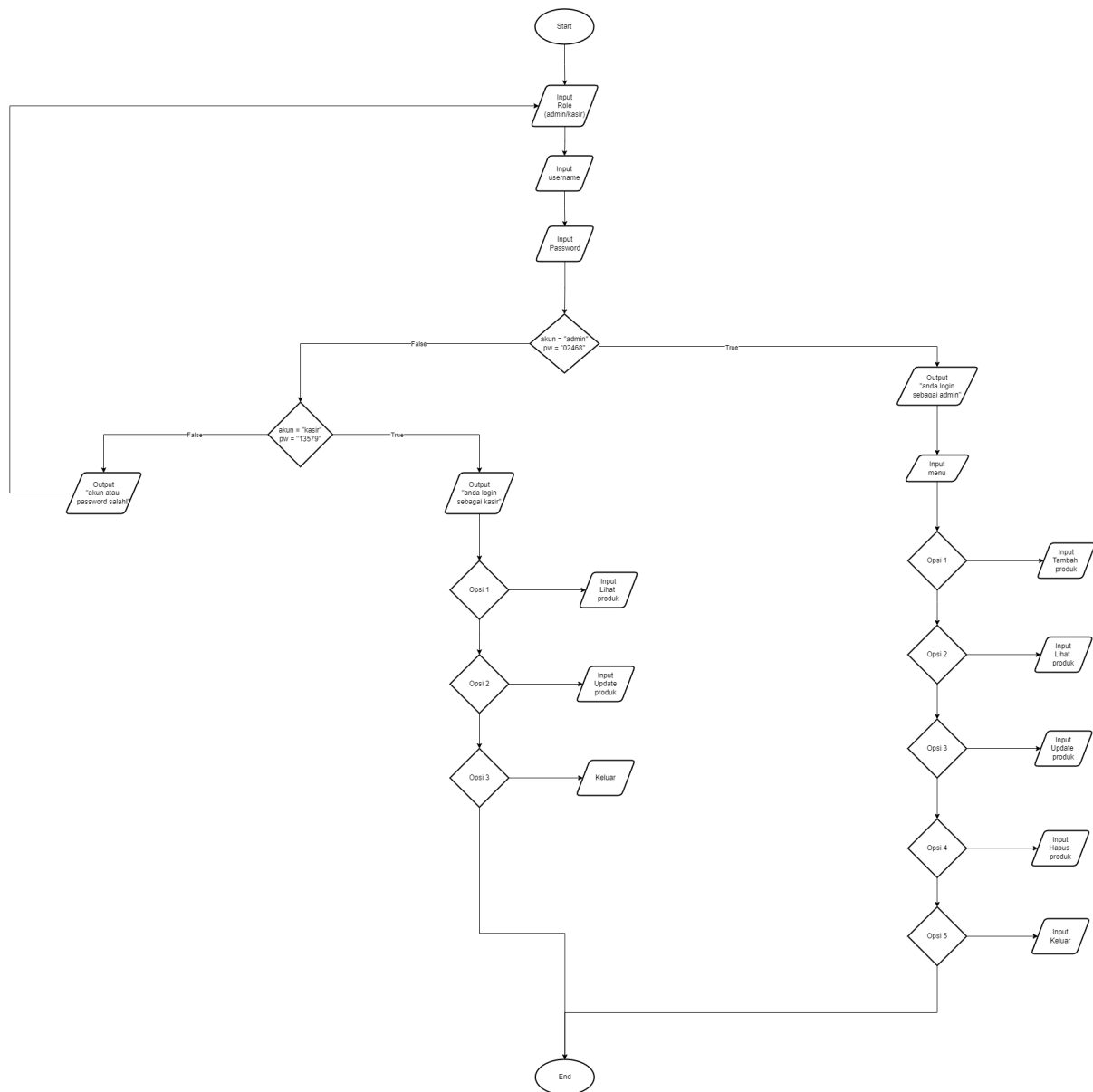
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



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 di kelola dapat tersimpan secara aman. Program ini dapat digunakan untuk multiuser, dimana ada admin, kasir, user/pengguna, dengan sesuai kebutuhan dalam program masing-masing. Nantinya disetiap 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 harus sesuai dengan data yang tersimpan dalam program. Apabila login gagal, pengguna/users akan diminta mengisi kembali data tersebut dengan benar hingga dapat login.

2. Menampilkan Menu Program

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.

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 memasukan akun, nama/username dan password yang sudah tersimpan pada program, 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 pengguna dapat login.

```
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, 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.

```
if (akun == "admin" && password == "02468") {

    Admin = true;

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

} else if (akun == "kasir" && password == "13579") {

    Admin = false;
```

```

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

    } else {

        cout << "akun atau password salah!" << endl;

        continue;

    }

```

3. Menampilkan Menu Program

Jadi pada fitur menampilkan program memiliki 2 menu, yang satu untuk admin dan yang satunya untuk kasir, admin memiliki CRUD (Creat, Read, Update dan Delete), sedangkan untuk kasir hanya memiliki Read dan Update.

```

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

    }

```

4. Proses Pada Setiap Opsi Menu Program

A. Admin

- Creat

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

```

if (JumlahSabun < MAX_PRODUK) {

    cout << "Silahkan Masukkan Nama Sabun: ";

    cin.ignore();

    getline(cin, NamaSabun[JumlahSabun]);

    cout << "Tetapkan untuk harga produk: ";

    cin >> Harga[JumlahSabun];

    cout << "Masukkan Jumlah Stok Sabun: ";

    cin >> Stok[JumlahSabun];

    JumlahSabun++;

    cout << "Produk Telah berhasil ditambahkan!" << endl;

} else {

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

}

```

- **Read**

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 < JumlahSabun; i++) {

        cout << "Nama: " << NamaSabun[i] << ", Harga: " <<
Harga[i] << ", Stok: " << Stok[i] << endl;

    }
```

- **Update**

Update ialah untuk mengupdate data atau memperbaharui data, biasanya data diperbaharui karena memiliki stok baru atau barang baru nantinya barang lama atau stok lama akan di perbaharui, 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 < JumlahSabun; i++) {

    if (NamaSabun[i] == namasabun) {

        cout << "Silahkan Masukkan Harga Baru: ";

        cin >> Harga[i];

        cout << "Masukkan Jumlah Stok Baru: ";
```

```

        cin >> Stok[i];

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

        Update = true;

        break;
    }
}

if (!Update) {
    cout << "Maaf, Produk Tidak Ditemukan!" << endl;
}

```

- **Delete**

Delete ialah menghapus data. Dimana jika ada data atau barang yang sudah tidak diproduksi atau stoknya habis maka menggunakan delete untuk menghapusnya, dan di read 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 < JumlahSabun; i++) {
    if (NamaSabun[i] == namasabun) {
        for (int j = i; j < JumlahSabun - 1; j++) {
            NamaSabun[j] = NamaSabun[j + 1];

```



```

        Harga[j] = Harga[j + 1];

        Stok[j] = Stok[j + 1];

    }

    JumlahSabun--;

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

    Update = true;

    break;

}

}

if (!Update) {

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

}

```

- **Keluar**

Terakhir pada dalam menu program admin terdapat keluar, yang nantinya program ini akan mengeluarkan admin dalam fitur admin, dan nantinya dapat login kembali sebagai admin atau kasir tanpa benar-benar keluar dari program.

```

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

    break;

} 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 < JumlahSabun; i++) {

    cout << "Nama: " << NamaSabun[i] << ", Harga: " <<
Harga[i] << ", Stok: " << Stok[i] << endl;

}
```

- Update

Dan fitur yang 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 < JumlahSabun; i++) {

    if (NamaSabun[i] == namesabun) {
```

```

        cout << "Silahkan Masukkan Harga Baru: ";

        cin >> Harga[i];

        cout << "Masukkan Jumlah Stok Baru: ";

        cin >> Stok[i];

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

        found = true;

        break;
    }
}

if (!found) {

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

}

```

- **Keluar**

Terakhir 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.

3. Source Code

```
#include <iostream>

#include <string>

using namespace std;

#define MAX_PRODUK 250

int main() {

    string NamaSabun[MAX_PRODUK];

    int Stok[MAX_PRODUK];

    double Harga[MAX_PRODUK];

    int JumlahSabun = 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" && password == "02468") {

    Admin = true;

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

} else if (akun == "kasir" && password == "13579") {

    Admin = false;

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

} else {

    cout << "akun atau password salah!" << endl;

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

    }

    int Opsi;

    cout << "Pilih menu: ";

    cin >> Opsi;

    if (Admin) {

        if (Opsi == 1) {

            if (JumlahSabun < MAX_PRODUK) {

                cout << "Silahkan Masukkan Nama Sabun: ";

                cin.ignore();

                getline(cin, NamaSabun[JumlahSabun]);

                cout << "Tetapkan untuk harga produk: ";

                cin >> Harga[JumlahSabun];

                cout << "Masukkan Jumlah Stok Sabun: ";

                cin >> Stok[JumlahSabun];

                JumlahSabun++;
            }
        }
    }

```

```

        cout << "Produk Telah berhasil ditambahkan!" << endl;

    } else {

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

    }

} else if (Ops == 2) {

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

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

        cout << "Nama: " << NamaSabun[i] << ", Harga: " <<
Harga[i] << ", Stok: " << Stok[i] << endl;

    }

} else if (Ops == 3) {

    string namasabun;

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

    cin.ignore();

    getline(cin, namasabun);

    bool Update = false;

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

        if (NamaSabun[i] == namasabun) {

            cout << "Silahkan Masukkan Harga Baru: ";

            cin >> Harga[i];

            cout << "Masukkan Jumlah Stok Baru: ";

```

```

        cin >> Stok[i];

        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 < JumlahSabun; i++) {
        if (NamaSabun[i] == namasabun) {
            for (int j = i; j < JumlahSabun - 1; j++) {
                NamaSabun[j] = NamaSabun[j + 1];
                Harga[j] = Harga[j + 1];
                Stok[j] = Stok[j + 1];
            }

```



```

        JumlahSabun--;

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

        Update = true;

        break;
    }
}

if (!Update) {
    cout << "Produk tidak ditemukan!" << endl;
}

} else if (Opsi == 5) {
    cout << "Keluar dari menu admin/kasir." << endl;

    break;
} else {
    cout << "Pilihan tidak valid!" << endl;
}

} else {
    if (Opsi == 1) {
        cout << "\n=== Daftar Produk ===" << endl;

        for (int i = 0; i < JumlahSabun; i++) {
            cout << "Nama: " << NamaSabun[i] << ", Harga: " <<
            Harga[i] << ", Stok: " << Stok[i] << 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 < JumlahSabun; i++) {

            if (NamaSabun[i] == namasabun) {

                cout << "Silahkan Masukkan Harga Baru: ";

                cin >> Harga[i];

                cout << "Masukkan Jumlah Stok Baru: ";

                cin >> Stok[i];

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

                found = true;

                break;

            }

        }

        if (!found) {

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

        }

    } else if (Opsii == 3) {

        cout << "Keluar dari menu admin/kasir." << endl;
    }
}

```

```
        break;

    } else {

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

    }

}

}

}

return 0;

}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

4.2 Hasil Output