

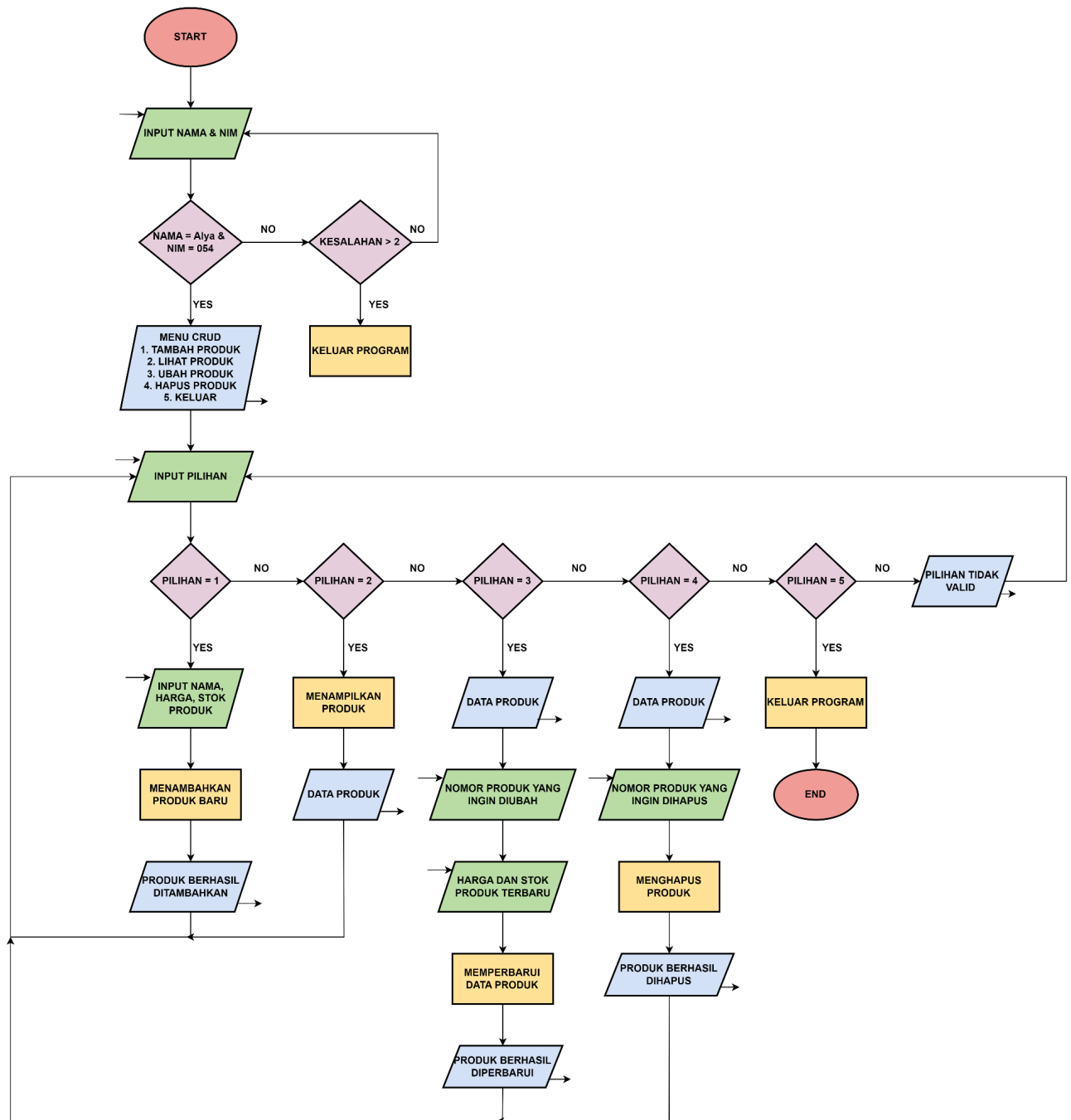
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Alya Mayasha (2409106054)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Program Manajemen Tas Rajut

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini adalah sebuah program manajemen produk tas rajut berbasis C++. Program memungkinkan pengguna untuk mengelola data produk dengan berbagai fitur, seperti *Create* pengguna dapat menambahkan produk baru ke dalam daftar dengan memasukkan nama, harga, stok, dan warna produk. Kemudian *Read* pengguna dapat melihat daftar produk dalam bentuk tabel dengan informasi lengkap. Selanjutnya *Update* pengguna dapat mengedit harga dan stok produk. Dan *Delete* pengguna dapat menghapus produk dari daftar berdasarkan nomor yang dipilih. Pada program ini pengguna diberikan kesempatan tiga kali untuk login dengan memasukkan nama (Alya) dan nim (054). Jika berhasil pengguna dapat mengakses menu CRUD yang tersedia. Program akan terus berjalan hingga pengguna memilih opsi keluar. Jika pengguna gagal login sebanyak tiga kali, maka program akan otomatis berhenti. Program ini menggunakan array untuk menyimpan data produk dan loop serta percabangan untuk navigasi menu.

2.2 Penjelasan Alur & Algoritma

1. Login Pengguna

- Program meminta pengguna memasukkan Nama dan NIM.
- Jika input sesuai dengan data yang tersimpan (Alya, 054), pengguna bisa masuk.
- Jika input salah, pengguna diberikan 3 kali percobaan sebelum program berhenti.

2. Menampilkan Menu Utama

Setelah login berhasil, program menampilkan menu manajemen produk tas rajut, yang berisi:

1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar dari Program

3. Memproses Pilihan Pengguna

Program meminta pengguna memilih salah satu opsi dari menu utama. Setiap pilihan akan menjalankan fungsi tertentu untuk mengelola data produk.

4. Fitur Manajemen Produk

- Tambah Produk → Pengguna bisa menambahkan produk baru jika kapasitas belum penuh. Untuk menambah produk baru ke dalam daftar, pengguna diminta memasukkan nama, harga, stok, dan warna produk.
- Tampilkan Produk → Program menampilkan daftar produk dalam bentuk tabel.
- Ubah Produk → Pengguna bisa mengedit harga dan stok produk yang sudah ada.
- Hapus Produk → Pengguna bisa menghapus produk berdasarkan nomor urutnya.
- Keluar → Program berhenti setelah menampilkan pesan perpisahan.

5. Program Berjalan dalam Loop

Selama pengguna belum memilih Keluar (5), menu utama akan terus muncul dan meminta input baru. Setelah memilih Keluar, program akan berhenti.

3. Source Code

A. Fitur Tambah Produk

Fitur ini memungkinkan pengguna untuk menambahkan produk baru ke dalam daftar jika kapasitas belum penuh.

Source Code:

```
if (jumlahproduk < max_produk) {
    cin.ignore();
    cout << "Masukkan nama produk: ";
    getline(cin, namaproduk[jumlahproduk]);
    cout << "Masukkan harga produk: ";
    cin >> hargaproduk[jumlahproduk];
    cout << "Masukkan stok produk: ";
    cin >> stokproduk[jumlahproduk];
    cout << "Masukkan warna produk: ";
    cin >> warnaproduk[jumlahproduk];
    jumlahproduk++;
    cout << "Produk berhasil ditambahkan!\n";
} else {
    cout << "Data penuh! Tidak bisa menambah produk lagi.\n";
}
```

B. Fitur Tampilkan Produk

Menampilkan daftar produk dalam bentuk tabel yang rapi.

Source Code:

```
if (jumlahproduk == 0) {
    cout << "Belum ada produk yang tersedia.\n";
} else {
    cout << "\n+---+-----+-----+-----+-----+\n";
    cout << "| No | Nama Produk          | Harga      | Stok | Warna          |\n";
    cout << "+---+-----+-----+-----+-----+\n";
    for (int i = 0; i < jumlahproduk; i++) {
        cout << "| " << setw(2) << i + 1
            << " | " << left << setw(18) << namaproduk[i]
            << " | " << right << setw(10) << hargaproduk[i]
            << " | " << setw(4) << stokproduk[i]
            << " | " << setw(10) << left << warnaproduk[i] << " |\n";
    }
    cout << "+---+-----+-----+-----+-----+\n";
}
```

C. Fitur Ubah Produk

Fitur ini memungkinkan pengguna untuk mengedit harga dan stok produk berdasarkan nomor urutnya.

Source Code:

```
if (jumlahproduk == 0) {
    cout << "Tidak ada produk untuk diubah.\n";
} else {
    int index;
    cout << "Masukkan nomor produk yang ingin diubah: ";
    cin >> index;
    if (index > 0 && index <= jumlahproduk) {
        index--;
        cout << "Masukkan harga baru: ";
        cin >> hargaproduk[index];
        cout << "Masukkan stok baru: ";
        cin >> stokproduk[index];
        cout << "Produk berhasil diperbarui!\n";
    } else {
        cout << "Nomor produk tidak valid!\n";
    }
}
```

D. Fitur Hapus Produk

Fitur ini untuk menghapus produk dari daftar dengan menggeser elemen array ke kiri.

Source Code:

```
if (jumlahproduk == 0) {
    cout << "Tidak ada produk yang bisa dihapus.\n";
} else {
    int index;
    cout << "Masukkan nomor produk yang ingin dihapus: ";
    cin >> index;
    if (index > 0 && index <= jumlahproduk) {
        index--;
        for (int i = index; i < jumlahproduk - 1; i++) {
            namaproduk[i] = namaproduk[i + 1];
            hargaproduk[i] = hargaproduk[i + 1];
            stokproduk[i] = stokproduk[i + 1];
            warnaproduk[i] = warnaproduk[i + 1];
        }
    }
}
```

```
        jumlahproduk--;  
        cout << "Produk berhasil dihapus!\n";  
    } else {  
        cout << "Nomor produk tidak valid!\n";  
    }  
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1
Ketika pengguna ingin melihat daftar produk dan memilih pilihan 2.
2. Skenario 2
Ketika pengguna ingin menambah produk baru ke dalam daftar produk, yaitu Dompot Rajut dengan harga Rp. 95.000,- , stok sebanyak 50, dan warna navy.
3. Skenario 3
Ketika pengguna ingin mengubah harga dan stok produk Tote Bag Rajut menjadi Rp. 140.000,- dan stok sebanyak 20.
4. Skenario 4
Ketika pengguna ingin menghapus produk Pouch Rajut dari daftar produk.
5. Skenario 5
Ketika pengguna melakukan kesalahan input nama atau nim sebanyak tiga kali.

4.2 Hasil Output

```
=== Manajemen Produk Tas Rajut ===
1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 2
```

No	Nama Produk	Harga	Stok	Warna
1	Backpack Rajut	300000	20	Pink
2	Tote Bag Rajut	125000	45	Baby Blue
3	Sling Bag Rajut	95000	30	Hitam
4	Hand Bag Rajut	80000	15	Cream
5	Pouch Rajut	65000	50	Maroon

Gambar 4.1 Output Skenario 1


```

=== Manajemen Produk Tas Rajut ===
1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 1
Masukkan nama produk: Dompot Rajut
Masukkan harga produk: 95000
Masukkan stok produk: 50
Masukkan warna produk: Navy
Produk berhasil ditambahkan!

+---+-----+-----+-----+-----+
| No | Nama Produk | Harga | Stok | Warna |
+---+-----+-----+-----+-----+
| 1 | Backpack Rajut | 300000 | 20 | Pink |
| 2 | Tote Bag Rajut | 125000 | 45 | Baby Blue |
| 3 | Sling Bag Rajut | 95000 | 30 | Hitam |
| 4 | Hand Bag Rajut | 80000 | 15 | Cream |
| 5 | Pouch Rajut | 65000 | 50 | Maroon |
| 6 | Dompot Rajut | 95000 | 50 | Navy |
+---+-----+-----+-----+-----+

```

Gambar 4.2 Output Skenario 2

```

=== Manajemen Produk Tas Rajut ===
1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 3

+---+-----+-----+-----+-----+
| No | Nama Produk | Harga | Stok | Warna |
+---+-----+-----+-----+-----+
| 1 | Backpack Rajut | 300000 | 20 | Pink |
| 2 | Tote Bag Rajut | 125000 | 45 | Baby Blue |
| 3 | Sling Bag Rajut | 95000 | 30 | Hitam |
| 4 | Hand Bag Rajut | 80000 | 15 | Cream |
| 5 | Pouch Rajut | 65000 | 50 | Maroon |
| 6 | Dompot Rajut | 95000 | 50 | Navy |
+---+-----+-----+-----+-----+

Masukkan nomor produk yang ingin diubah: 2
Masukkan harga baru: 140000
Masukkan stok baru: 20
Produk berhasil diperbarui!

```

Gambar 4.3 Output Skenario 3

```

=== Manajemen Produk Tas Rajut ===
1. Tambah Produk
2. Tampilkan Produk
3. Ubah Produk
4. Hapus Produk
5. Keluar
Pilihan: 4

+---+-----+-----+-----+-----+
| No | Nama Produk | Harga | Stok | Warna |
+---+-----+-----+-----+-----+
| 1 | Backpack Rajut | 300000 | 20 | Pink |
| 2 | Tote Bag Rajut | 140000 | 20 | Baby Blue |
| 3 | Sling Bag Rajut | 95000 | 30 | Hitam |
| 4 | Hand Bag Rajut | 80000 | 15 | Cream |
| 5 | Pouch Rajut | 65000 | 50 | Maroon |
| 6 | Dompot Rajut | 95000 | 50 | Navy |
+---+-----+-----+-----+-----+
Masukkan nomor produk yang ingin dihapus: 5
Produk berhasil dihapus!

```

Gambar 4.4 Output Skenario 4

```

Masukkan Nama: Alya
Masukkan NIM: 041
Login gagal! Sisa percobaan: 2
Masukkan Nama: Newwie
Masukkan NIM: 054
Login gagal! Sisa percobaan: 1
Masukkan Nama: Tawan
Masukkan NIM: 019
Login gagal! Sisa percobaan: 0
Mohon maaf percobaan login gagal! Program berhenti.
PS C:\Users\alyam\coding>

```

Gambar 4.5 Output Skenario 5

5. GIT

5.1 Git Add

Git Add berfungsi untuk menambahkan file tertentu ke staging area agar siap untuk dikommit.

```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/desktop/praktikum-apl/post-test/post-test-2 (main)
$ git add .
```

Gambar 5.1 Git Add

5.2 Git Commit

Git Commit berfungsi untuk menyimpan perubahan yang ada di staging area ke dalam repository disertai dengan pesan deskriptif.

```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/desktop/praktikum-apl/post-test/post-test-2 (main)
$ git commit -m "file cpp dan exe"
[main fcf0e6c] file cpp dan exe
2 files changed, 176 insertions(+)
create mode 100644 post-test/post-test-2/2409106054-AlyaMayasha-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106054-AlyaMayasha-PT-2.exe
```

Gambar 5.2 Git Commit

5.3 Git Push Origin Main

Git Push Origin Main berfungsi untuk mengirim (mengunggah) perubahan dari branch main di repository lokal ke repository remote yang telah dikonfigurasi.

```
HP-GK@DESKTOP-N8AD402 MINGW64 ~/desktop/praktikum-apl/post-test/post-test-2 (main)
$ git push 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), 678.63 KiB | 2.71 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/AlyaMayasha/praktikum-apl.git
0d479ec..fcf0e6c main -> main
```

Gambar 5.3 Git Push Origin Main