

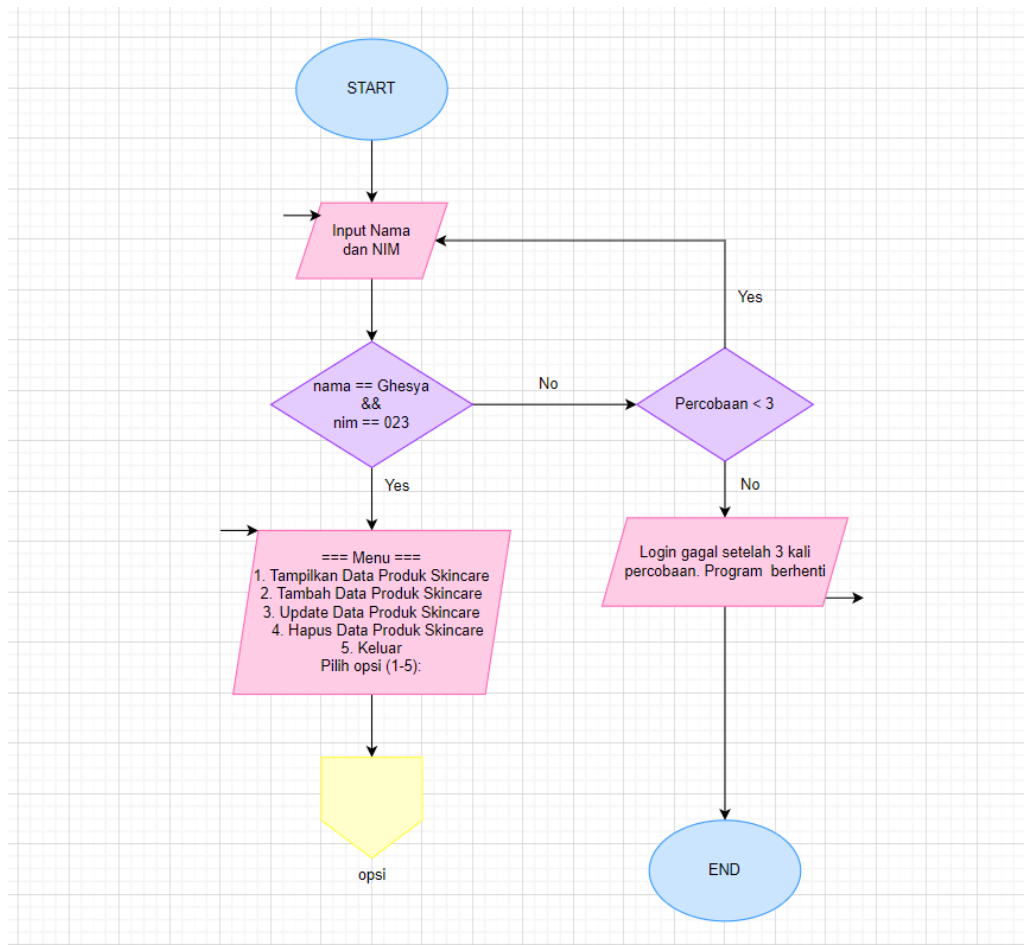
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



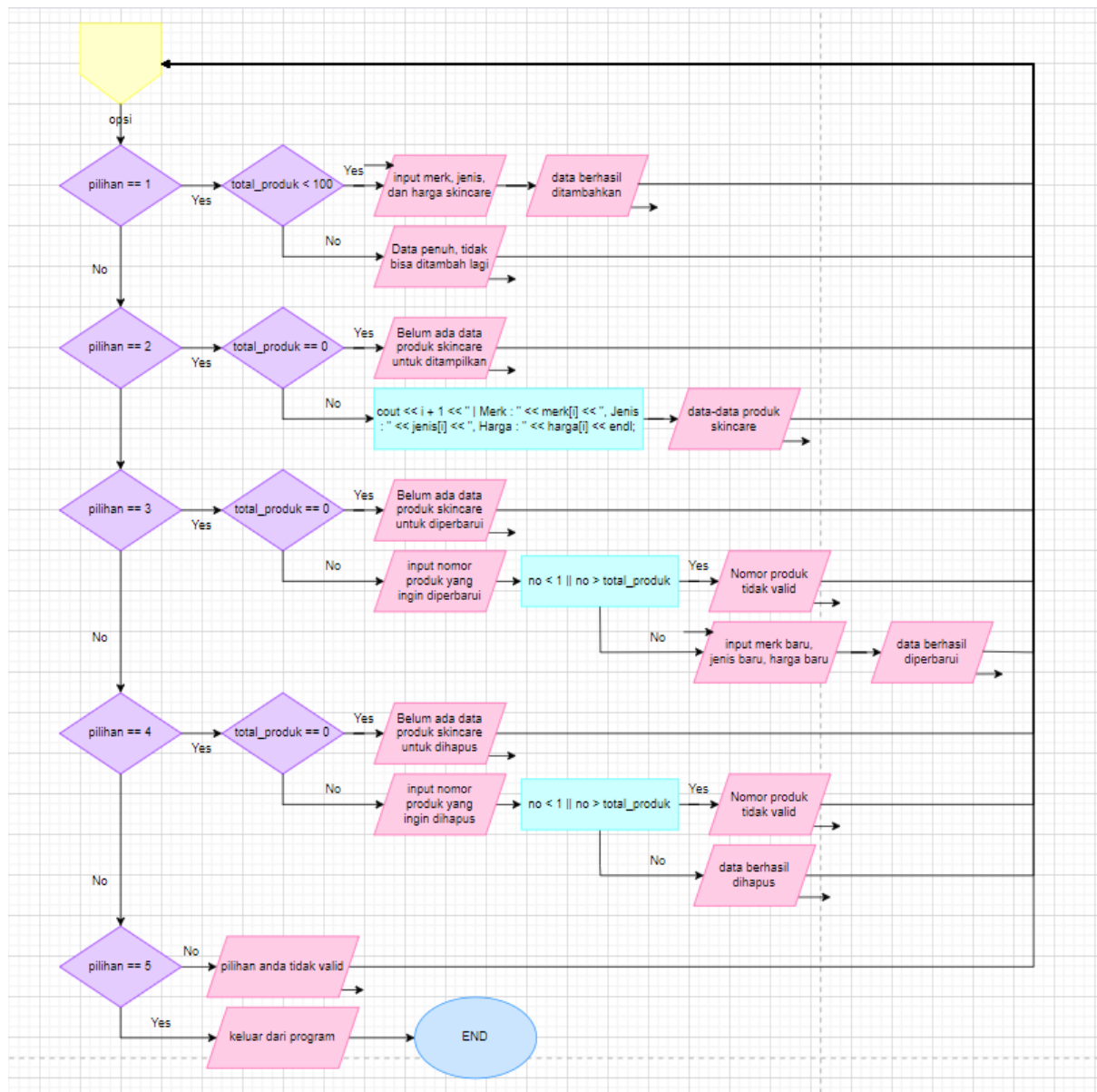
Disusun oleh:
Ghesya Rhegyta Al Rachman (2409106023)
Kelas (A2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart



Gambar 1.2 Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Tujuan :

Program ini bertujuan untuk membantu pengelolaan data produk skincare. Pengguna dapat menambah, melihat, memperbarui, serta menghapus data produk skincare. Program ini dapat membantu bisnis kecil untuk mencatat produk skincare.

Manfaat :

- Mudah digunakan – Dengan adanya menu yang jelas, pengguna dapat mengelola data skincare tanpa kesulitan
- Dapat dikembangkan lebih lanjut – Dapat ditambahkan fitur seperti pencarian produk
- Perulangan – Program terus berjalan hingga pengguna memilih untuk berhenti

2.2. Penjelasan Alur & Algoritma

Alur Program :

1. Login pengguna, batas maksimal 3 kali percobaan
 - Jika login berhasil = program lanjut ke menu utama
 - Jika login gagal = program memberikan kesempatan 3 kali percobaan. Setelah 3 kali percobaan namun gagal, program akan berhenti.
2. Menampilkan menu utama
 - Pilihan 1 = Tambah Data Produk Skincare
 - Pilihan 2 = Tampilkan Data Produk Skincare
 - Pilihan 3 = Perbarui Data Produk Skincare
 - Pilihan 4 = Hapus Data Produk Skincare
 - Pilihan 5 = Keluar dari program
3. Memproses sesuai opsi yang dipilih pengguna
4. Kembali ke menu pilihan atau jika pengguna memilih opsi 5, maka program akan berhenti

Algoritma Program :

1. Mulai
2. Inisialisasi variabel login
 - Nama == “Ghesya” dan nim == “023”
3. Proses login
 - Memeriksa apakah nama yang dimasukkan adalah “Ghesya” dengan NIM “023”
 - Jika login benar = menampilkan “Login berhasil!” lalu lanjut ke menu pilihan konversi suhu
 - Jika salah = terdapat 3 kali kesempatan percobaan login
 - Jika percobaan login habis = menampilkan pesan “Login gagal setelah 3 kali percobaan. Program berhenti.” dan program diakhiri
4. Menampilkan menu utama
 - Program menampilkan opsi yang dapat dipilih oleh pengguna
 - 1) Tambah Data Produk Skincare
 - 2) Tampilkan Data Produk Skincare
 - 3) Perbarui Data Produk Skincare
 - 4) Hapus Data Produk Skincare
 - 5) Keluar Dari Program
 - Pengguna memasukkan opsi yang dipilih, yang akan menentukan langkah selanjutnya
 - Jika memilih opsi 5, maka program berhenti
 - Jika pilihan tidak valid = menampilkan pesan “Pilihan anda tidak valid!” dan kembali ke menu utama
5. CRUD Data Produk
 - Tambah Data Produk Skincare : Pengguna menginput merk skincare, jenis skincare, dan harga skincare
 - Tampilkan Data Produk Skincare : Program menampilkan seluruh data produk skincare
 - Perbarui Data Produk Skincare : Pengguna memilih nomor produk yang ingin diperbarui, kemudian memasukkan data baru untuk mengganti data lama
 - Hapus Data Produk Skincare : Pengguna dapat memilih nomor produk yang ingin dihapus

3. Source Code

3.1. Fungsi Login

Memverifikasi nama dan 3 digit terakhir NIM dengan batas maksimal 3 kali percobaan. Jika gagal, maka program berhenti.

```
int main() {
    string nama, inputNama;
    string nim, inputNIM;
    int attempts = 0;

    cout << "=== LOGIN ===" << endl;
    cout << "Masukkan nama: ";
    cin >> nama;
    cout << "Masukkan 3 digit terakhir NIM: ";
    cin >> nim;

    if (nama == "Ghesya" && nim == "023") {
        cout << "Login berhasil!" << endl;
    } else {
        attempts++;
        while (attempts < 3) {
            cout << "\nLogin Ulang" << endl;
            cout << "Masukkan nama: ";
            cin >> inputNama;
            cout << "Masukkan 3 digit terakhir NIM: ";
            cin >> inputNIM;

            if (inputNama == "Ghesya" && inputNIM == "023") {
                cout << "Login berhasil!" << endl;
                break;
            } else {
                attempts++;
                cout << "Login gagal! Percobaan ke-" << attempts << " dari
3" << endl;
            }
        }

        if (attempts == 3) {
            cout << "Login gagal setelah 3 kali percobaan. Program berhenti." <<
endl;
            return 0;
        }
    }
}
```

Gambar 3.1.1 Fungsi Login

3.2. Menu Utama

Menampilkan pilihan menu utama produk skincare

```
while (true) {
    int pilihan;
    cout << "\n-----" << endl;
    cout << "||          === Menu ===          ||" << endl;
    cout << "-----" << endl;
    cout << "||  1. Tambah Data Produk Skincare  ||" << endl;
    cout << "||  2. Tampilkan Data Produk Skincare ||" << endl;
    cout << "||  3. Perbarui Data Produk Skincare ||" << endl;
    cout << "||  4. Hapus Data Produk Skincare   ||" << endl;
    cout << "||  5. Keluar dari program          ||" << endl;
    cout << "-----" << endl;
    cout << "Pilih opsi (1-5): ";
    cin >> pilihan;
```

Gambar 3.2.1 Menu Utama Produk Skincare

3.3. CRUD Manajemen Produk Skincare

Fitur CRUD dalam program ini yaitu Create (membuat atau menambahkan data), Read (menampilkan data yang tersimpan), Update (memperbarui data yang sudah ada), dan Delete (menghapus data yang sudah tidak diperlukan)

```
if (pilihan == 1) {
    if (total_produk < 100) {
        cout << "Masukkan merk skincare: ";
        cin >> merk[total_produk];
        cout << "Masukkan jenis skincare: ";
        cin >> jenis[total_produk];
        cout << "Masukkan harga skincare: ";
        cin >> harga[total_produk];

        total_produk++;
        cout << "Data berhasil ditambahkan!" << endl;
    } else {
        cout << "Jumlah data sudah penuh >_< huhuhuu tidak bisa ditambah lagi" << endl;
    }
}
else if (pilihan == 2) {
    if (total_produk == 0) {
        cout << "Belum ada data produk skincare untuk ditampilkan" << endl;
    } else {
        cout << "\n-----"
```

```

<< endl;

        cout << "=== Data Produk Skincare ===" << endl;
        cout << "-----"

<< endl;

        for (int i = 0; i < total_produk; ++i) {
            cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " <<
jenis[i] << ", Harga : " << harga[i] << endl;
        }
        cout << "-----"

<< endl;
    }
}
else if (pilihan == 3) {
    if (total_produk == 0) {
        cout << "Belum ada data produk skincare untuk diperbarui" <<
endl;
    } else {
        int no;
        cout << "\n=== Data Produk Skincare ===" << endl;
        cout << "-----"

<< endl;

        for (int i = 0; i < total_produk; ++i) {
            cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " <<
jenis[i] << ", Harga : " << harga[i] << endl;
        }
        cout << "-----"

<< endl;

        cout << "\nMasukkan nomor produk yang ingin diperbarui: ";
        cin >> no;

        if (no < 1 || no > total_produk) {
            cout << "Nomor produk tidak valid" << endl;
        } else {
            cout << "Masukkan merk baru: ";
            cin >> merk[no - 1];
            cout << "Masukkan jenis baru: ";
            cin >> jenis[no - 1];
            cout << "Masukkan harga baru: ";
            cin >> harga[no - 1];
            cout << "Data berhasil diperbarui!" << endl;
        }
    }
}
else if (pilihan == 4) {
    if (total_produk == 0) {
        cout << "Belum ada data produk skincare untuk dihapus" << endl;
    } else {
        int no;
        cout << "\n=== Data Produk Skincare ===" << endl;
        cout << "-----"

<< endl;

        for (int i = 0; i < total_produk; ++i) {
            cout << i + 1 << " | Merk : " << merk[i] << ", Jenis : " <<

```



```

jenis[i] << ", Harga : " << harga[i] << endl;
    }
    cout << "-----"
<< endl;

    cout << "\nMasukkan nomor produk yang ingin dihapus: ";
    cin >> no;

    if (no < 1 || no > total_produk) {
        cout << "Nomor produk tidak valid" << endl;
    } else {
        for (int i = no - 1; i < total_produk - 1; ++i) {
            merk[i] = merk[i + 1];
            jenis[i] = jenis[i + 1];
            harga[i] = harga[i + 1];
        }
        total_produk--;
        cout << "Data berhasil dihapus!" << endl;
    }
}
}
else if (pilihan == 5) {
    cout << "Timaaci telah menggunakan program ini!!! papayyy" << endl;
    break;
} else {
    cout << "Pilihan anda tidak valid!" << endl;
}
}

return 0;
}

```

Gambar 3.3.1 CRUD Manajemen Produk Skincare

4. Uji Coba dan Hasil Output

4.1. Uji Coba

1. Skenario 1 : Login dengan nama atau NIM yang salah
 - Input nama atau NIM yang salah sebanyak 3 kali
 - Jika pengguna menginputkan data yang benar sebelum 3 kali percobaan, maka pengguna dapat mengakses ke menu utama
 - Setelah 3 kali percobaan gagal, maka program akan berhenti
2. Skenario 2 : Login berhasil
 - Input nama, yaitu "Ghesya"
 - Input 3 digit terakhir NIM, yaitu "023"
 - Login berhasil dan program menampilkan menu utama

3. Skenario 3 : Menambahkan data produk skincare
 - Memilih opsi 1 yaitu “Tambah Data Produk Skincare”
 - Input data produk yaitu merk, jenis, dan harga skincare
 - Program akan menampilkan “Data berhasil ditambahkan!”
 - Jika data yang tersimpan sudah 100 (maksimal), maka program menampilkan pesan “Jumlah data sudah penuh >_< huhuhuu tidak bisa ditambah lagi”
4. Skenario 4 : Menampilkan data produk skincare
 - Memilih opsi 2 yaitu “Tampilkan Data Produk Skincare”
 - Program akan menampilkan seluruh data produk skincare
 - Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk ditampilkan”
5. Skenario 5 : Memperbarui data produk skincare
 - Memilih opsi 3 yaitu “Perbarui Data Produk Skincare”
 - Input nomor produk yang ingin diperbarui
 - Input data baru untuk menggantikan data lama produk skincare
 - Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk diperbarui”
6. Skenario 6 : Menghapus data produk skincare
 - Memilih opsi 4 yaitu “Hapus Data Produk Skincare”
 - Input nomor produk yang ingin dihapus
 - Program akan menampilkan “Data berhasil dihapus!”
 - Jika tidak ada data yang tersimpan, maka program menampilkan pesan “Belum ada data produk skincare untuk dihapus”
7. Skenario 7 : Keluar dari program
 - Memilih opsi 4 yaitu “Keluar dari program”
 - Program akan menampilkan “Timaaci telah menggunakan program ini!!! papayyy”
 - Program berhenti

4.2. Hasil Output

- Hasil Output Uji Coba Skenario 1

```
=== LOGIN ===
Masukkan nama: Ghesya
Masukkan 3 digit terakhir NIM: 196

Login Ulang
Masukkan nama: Ghes
Masukkan 3 digit terakhir NIM: 023
Login gagal! Percobaan ke-2 dari 3

Login Ulang
Masukkan nama: Sate
Masukkan 3 digit terakhir NIM: 111
Login gagal! Percobaan ke-3 dari 3
Login gagal setelah 3 kali percobaan. Program berhenti.
PS C:\praktikum-apl\post-test\post-test-apl-2> █
```

Gambar 4.2.1 Login gagal dan program berhenti

- Hasil Output Uji Coba Skenario 2

```

=== LOGIN ===
Masukkan nama: Ghesya
Masukkan 3 digit terakhir NIM: 023
Login berhasil!

-----
||          === Menu ===          ||
-----
||  1. Tambah Data Produk Skincare  ||
||  2. Tampilkan Data Produk Skincare ||
||  3. Perbarui Data Produk Skincare ||
||  4. Hapus Data Produk Skincare   ||
||  5. Keluar dari program          ||
-----
Pilih opsi (1-5): █

```

Gambar 4.2.2 Login berhasil dan program menampilkan menu utama

- Hasil Output Uji Coba Skenario 3

```

-----
||          === Menu ===          ||
-----
||  1. Tambah Data Produk Skincare  ||
||  2. Tampilkan Data Produk Skincare ||
||  3. Perbarui Data Produk Skincare ||
||  4. Hapus Data Produk Skincare   ||
||  5. Keluar dari program          ||
-----
Pilih opsi (1-5): 1
Masukkan merk skincare: Msglow
Masukkan jenis skincare: Toner
Masukkan harga skincare: 60000
Data berhasil ditambahkan!

```

Gambar 4.2.3 Tambah data produk

- Hasil Output Uji Coba Skenario 4

```

-----
||          === Menu ===          ||
-----
||  1. Tambah Data Produk Skincare  ||
||  2. Tampilkan Data Produk Skincare ||
||  3. Perbarui Data Produk Skincare ||
||  4. Hapus Data Produk Skincare   ||
||  5. Keluar dari program          ||
-----
Pilih opsi (1-5): 2

-----
=== Data Produk Skincare ===
-----
1 | Merk : Azarine, Jenis : Sunscreen, Harga : 60000
2 | Merk : Npure, Jenis : Toner, Harga : 90000
3 | Merk : Senka, Jenis : Sabun, Harga : 50000
4 | Merk : Innisfree, Jenis : Serum, Harga : 350000
5 | Merk : Skintific, Jenis : Moisturizer, Harga : 100000
6 | Merk : Msglow, Jenis : Toner, Harga : 60000
-----

```

Gambar 4.2.4 Tampilkan data produk

- Hasil Output Uji Coba Skenario 5

```

=====
||           === Menu ===           ||
=====
|| 1. Tambah Data Produk Skincare  ||
|| 2. Tampilkan Data Produk Skincare ||
|| 3. Perbarui Data Produk Skincare ||
|| 4. Hapus Data Produk Skincare   ||
|| 5. Keluar dari program          ||
=====
Pilih opsi (1-5): 3

=== Data Produk Skincare ===
-----
1 | Merk : Azarine, Jenis : Sunscreen, Harga : 60000
2 | Merk : Npure, Jenis : Toner, Harga : 90000
3 | Merk : Senka, Jenis : Sabun, Harga : 50000
4 | Merk : Innisfree, Jenis : Serum, Harga : 350000
5 | Merk : Skintific, Jenis : Moisturizer, Harga : 100000
6 | Merk : Msglow, Jenis : Toner, Harga : 60000
-----

Masukkan nomor produk yang ingin diperbarui: 3
Masukkan merk baru: Senka
Masukkan jenis baru: Sabun
Masukkan harga baru: 80000
Data berhasil diperbarui!

```

Gambar 4.2.5 Perbarui data produk

```

=====
||           === Menu ===           ||
=====
|| 1. Tambah Data Produk Skincare  ||
|| 2. Tampilkan Data Produk Skincare ||
|| 3. Perbarui Data Produk Skincare ||
|| 4. Hapus Data Produk Skincare   ||
|| 5. Keluar dari program          ||
=====
Pilih opsi (1-5): 2

=== Data Produk Skincare ===
-----
1 | Merk : Azarine, Jenis : Sunscreen, Harga : 60000
2 | Merk : Npure, Jenis : Toner, Harga : 90000
3 | Merk : Senka, Jenis : Sabun, Harga : 80000
4 | Merk : Innisfree, Jenis : Serum, Harga : 350000
5 | Merk : Skintific, Jenis : Moisturizer, Harga : 100000
6 | Merk : Msglow, Jenis : Toner, Harga : 60000
-----

```

Gambar 4.2.6 Data setelah diperbarui

- Hasil Output Uji Coba Skenario 6

```

=====
||           === Menu ===           ||
=====
|| 1. Tambah Data Produk Skincare  ||
|| 2. Tampilkan Data Produk Skincare ||
|| 3. Perbarui Data Produk Skincare ||
|| 4. Hapus Data Produk Skincare   ||
|| 5. Keluar dari program          ||
=====
Pilih opsi (1-5): 4

=== Data Produk Skincare ===
-----
1 | Merk : Azarine, Jenis : Sunscreen, Harga : 60000
2 | Merk : Npure, Jenis : Toner, Harga : 90000
3 | Merk : Senka, Jenis : Sabun, Harga : 80000
4 | Merk : Innisfree, Jenis : Serum, Harga : 350000
5 | Merk : Skintific, Jenis : Moisturizer, Harga : 100000
6 | Merk : Msglow, Jenis : Toner, Harga : 60000
-----

Masukkan nomor produk yang ingin dihapus: 5
Data berhasil dihapus!

```

Gambar 4.2.7 Hapus data produk

```

===== Menu =====
1. Tambah Data Produk Skincare
2. Tampilkan Data Produk Skincare
3. Perbarui Data Produk Skincare
4. Hapus Data Produk Skincare
5. Keluar dari program

Pilih opsi (1-5): 2

===== Data Produk Skincare =====
1 | Merk : Azarine, Jenis : Sunscreen, Harga : 60000
2 | Merk : Npure, Jenis : Toner, Harga : 90000
3 | Merk : Senka, Jenis : Sabun, Harga : 80000
4 | Merk : Innisfree, Jenis : Serum, Harga : 350000
5 | Merk : Mglow, Jenis : Toner, Harga : 60000

```

Gambar 4.2.8 Data setelah dihapus

- Hasil Output Uji Coba Skenario 7

```

===== Menu =====
1. Tambah Data Produk Skincare
2. Tampilkan Data Produk Skincare
3. Perbarui Data Produk Skincare
4. Hapus Data Produk Skincare
5. Keluar dari program

Pilih opsi (1-5): 5
Timaaci telah menggunakan program ini!!! papayyy
PS C:\praktikum-apl\post-test\post-test-apl-2>

```

Gambar 4.2.9 Keluar dari program

5. Langkah-Langkah GIT

1. Git Add .

Menambahkan semua perubahan seperti file baru, file yang dimodifikasi, maupun file yang dihapus ke dalam staging area Git sebelum dikomit. Tanda titik (.) menunjukkan semua file dalam folder proyek akan ditambahkan ke staging area

```

PS C:\praktikum-apl\post-test\post-test-apl-2> git init
Initialized empty Git repository in C:/praktikum-apl/post-test/post-test-apl-2/.git/
PS C:\praktikum-apl\post-test\post-test-apl-2> git add .
PS C:\praktikum-apl\post-test\post-test-apl-2>

```

Gambar 5.1 Git Add .

2. Git Commit

Menyimpan perubahan yang ada di staging area ke dalam commit history. Setiap commit mencatat perubahan apa yang dilakukan pada kode. (-m) digunakan untuk menyertakan pesan commit secara langsung

```

PS C:\praktikum-apl\post-test\post-test-apl-2> git commit -m "Finish Post Test 2"
[main b052421] Finish Post Test 2
3 files changed, 171 insertions(+)
create mode 100644 post-test/post-test-apl-2/2409106023-GhesyaRhegytaAlRachman-PT-2.cpp
create mode 100644 post-test/post-test-apl-2/2409106023-GhesyaRhegytaAlRachman-PT-2.docx
create mode 100644 post-test/post-test-apl-2/2409106023-GhesyaRhegytaAlRachman-PT-2.exe
PS C:\praktikum-apl\post-test\post-test-apl-2>

```

Gambar 5.2 Git Commit

3. Git Push

Mengirim commit yang ada di lokal ke server seperti GitHub. Digunakan agar dapat diakses oleh orang lain atau untuk kolaborasi

```
PS C:\praktikum-apl> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 5.71 MiB | 563.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/ghesyarhegyta/praktikum-apl.git
 6bccf39..11c90e0  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apl>
```

Gambar 5.3 Git Push