

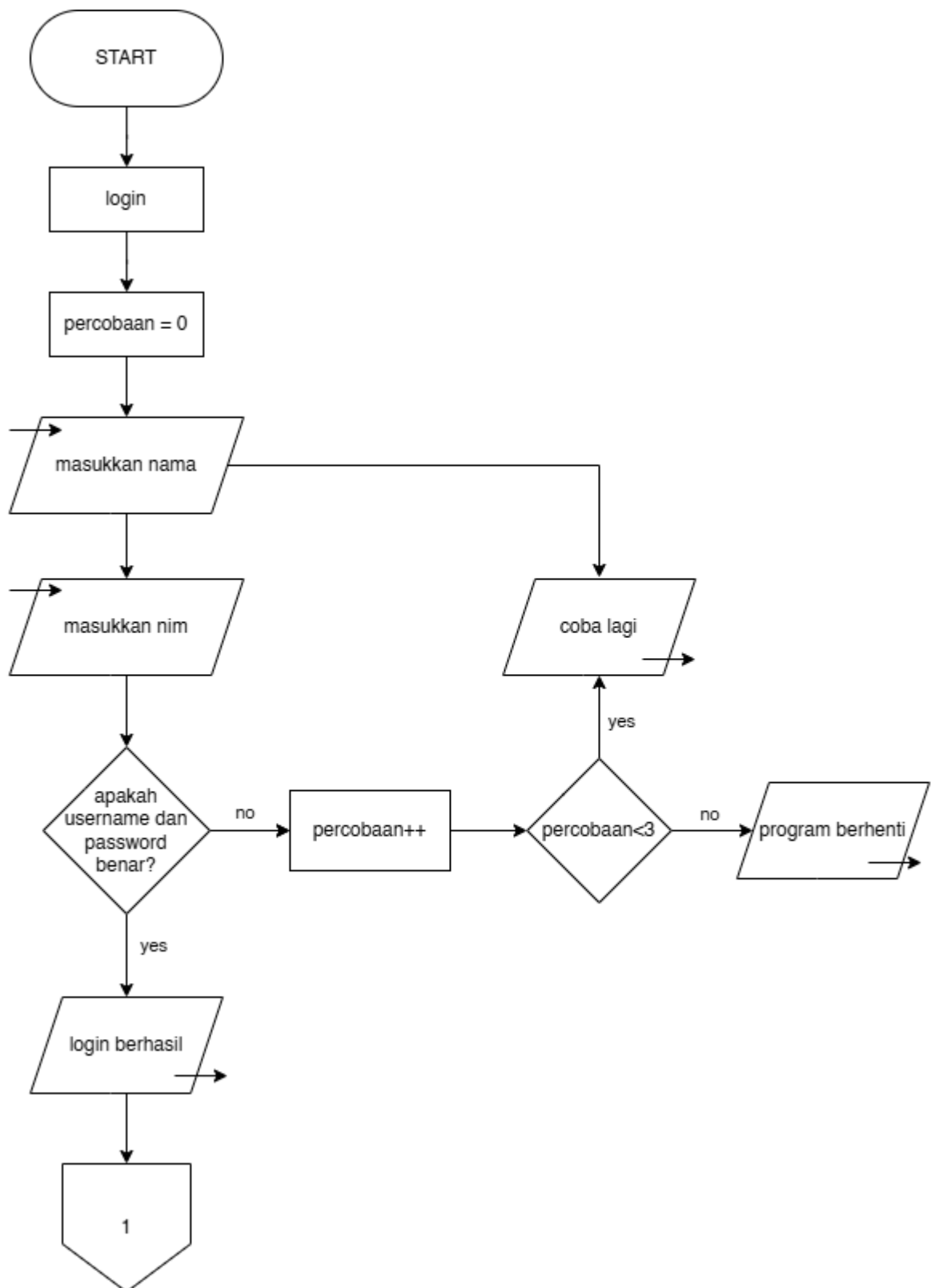
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**

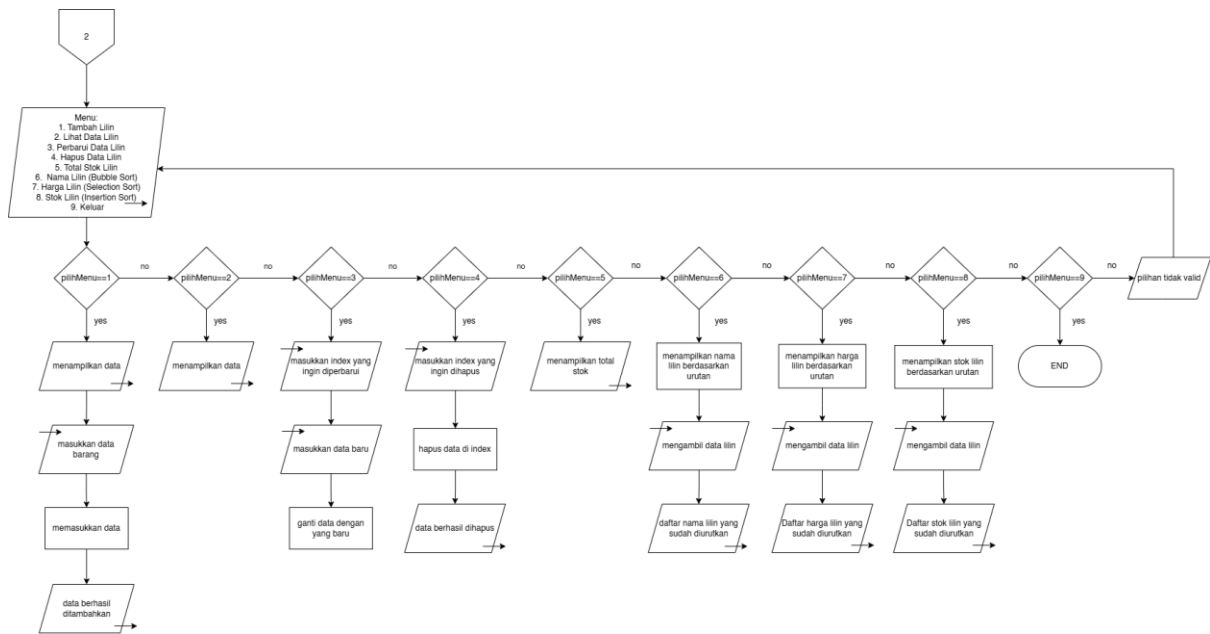


**Disusun oleh:**  
**Andi Nurfadillah Hasan (2409106087)**  
**Kelas (B2 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart





## 2. Analisis Program

Program ini merupakan sistem manajemen lilin aromaterapi yang dirancang untuk mengelola data produk lilin, termasuk menambah, melihat, memperbarui, dan menghapus data lilin. Selain itu, program ini juga memungkinkan untuk menghitung total stok lilin yang ada di sistem. Program ini dibangun dengan menggunakan struktur data **struct** untuk menyimpan informasi lilin, serta fitur login yang membatasi akses hanya untuk admin yang terdaftar.

### Fungsi Utama Program:

1. **Tambah Lilin:** Fitur untuk menambah data lilin baru ke dalam sistem. Admin dapat memasukkan informasi lilin seperti nama, harga, dan jumlah stok.
2. **Lihat Data Lilin:** Menampilkan seluruh data lilin dalam bentuk tabel yang memudahkan admin untuk memeriksa informasi lilin yang terdaftar.
3. **Perbarui Data Lilin:** Admin dapat memperbarui harga dan stok lilin menggunakan teknik pemrograman berbasis pointer. Fungsi **perbaruiHargaLilin** memanfaatkan referensi untuk memperbarui harga, sementara fungsi **ubahStokLilin** menggunakan pointer untuk memperbarui jumlah stok.
4. **Hapus Data Lilin:** Admin dapat menghapus data lilin yang tidak lagi diperlukan dengan menggeser elemen array setelah menghapus data tersebut.
5. **Total Stok Lilin:** Program menghitung dan menampilkan total jumlah stok lilin yang tersedia dalam sistem.
6. **Sorting Data Lilin:** Program mengimplementasikan tiga metode sorting, yaitu:
  - **Bubble Sort (Descending berdasarkan Nama Lilin):** Mengurutkan data lilin berdasarkan nama secara menurun.
  - **Selection Sort (Ascending berdasarkan Harga):** Mengurutkan data lilin berdasarkan harga secara menaik.

- **Insertion Sort (Descending berdasarkan Stok):** Mengurutkan data lilin berdasarkan stok secara menurun.

### **Implementasi Konsep Pointer:**

- Program ini mengaplikasikan konsep pointer melalui dua fungsi: **perbaruiHargaLilin** dan **ubahStokLilin**. Fungsi pertama menggunakan parameter address-of (&) untuk memungkinkan perubahan langsung pada harga lilin tanpa membuat salinan data. Fungsi kedua menggunakan parameter dereference (\*) untuk mengakses dan mengubah stok lilin berdasarkan alamat memori yang diberikan.

### 3. Source Code

#### A. Fitur Login

Fitur login bertujuan membatasi akses ke sistem hanya untuk admin yang memiliki kredensial yang benar. Sistem akan meminta **nama** dan **NIM** sebagai username dan password. Jika keduanya cocok dengan data admin yang tersimpan, maka login berhasil.

Namun, jika tidak cocok, pengguna diberi kesempatan maksimal **3 kali** untuk mencoba kembali. Setelah 3 kali gagal, program akan berhenti secara otomatis.

#### Source Code:

```
int main() {
    string nama, nim;
    int percobaan = 3;

    while (percobaan > 0) {
        cout << "\n==== LOGIN ADMIN =====> endl;
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, nim);

        if (sistem.admin.username == nama && sistem.admin.password == nim) {
            cout << "Login berhasil!\n";
            break;
        } else {
            percobaan--;
            cout << "Login gagal! Sisa percobaan: " << percobaan << "\n";
        }
    }

    if (percobaan == 0) {
        cout << "Anda telah gagal login 3 kali. Program berhenti.\n";
        return 0;
    }
}
```

#### B. Menu Utama & Pengelolaan Lilin Aromaterapi

Setelah berhasil login, admin diarahkan ke menu utama yang berisi berbagai fitur manajemen data lilin. Berikut adalah fitur-fitur utama:

## 1. Tambah Lilin

Admin dapat menambahkan data lilin baru ke sistem, dengan memasukkan **nama lilin**, **harga**, dan **stok**. Namun, penambahan hanya bisa dilakukan jika jumlah data belum mencapai batas maksimal.

### Source Code:

```
void tambahLilin(Sistem &s) {
    if (s.jumlahLilin < MAX_LILIN) {
        cout << "Masukkan nama lilin: ";
        getline(cin, s.lilin[s.jumlahLilin].nama);
        cout << "Masukkan harga lilin (Rp): ";
        getline(cin, s.lilin[s.jumlahLilin].harga);
        cout << "Masukkan stok lilin (pcs): ";
        getline(cin, s.lilin[s.jumlahLilin].stok);
        s.jumlahLilin++;
        cout << "Data berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas penuh!\n";
    }
}
```

## 2. Lihat Data Lilin

Menampilkan seluruh data lilin dalam bentuk tabel yang rapi, termasuk nomor urut, nama lilin, harga, dan stok. Jika belum ada data, akan ditampilkan pesan khusus.

### Source Code:

```
void lihatLilin(const Sistem &s) {
    if (s.jumlahLilin == 0) {
        cout << "Belum ada data lilin.\n";
    } else {
        cout <<
        "\n+=====+\n";
        cout << "| No | Nama Lilin | Harga (Rp) | Stok (pcs) |"
        "\n";
        cout <<
        "+=====+\n";
        for (int i = 0; i < s.jumlahLilin; i++) {
            cout << "| " << setw(3) << left << i + 1 << " | "

```

```

        << setw(20) << left << s.lilin[i].nama << " | "
        << setw(12) << right << s.lilin[i].harga << " | "
        << setw(10) << right << s.lilin[i].stok << " |\n";
    }
    cout <<
    "+=====+\n";
}
}

```

### 3. Perbarui Data Lilin

Memungkinkan admin memperbarui data lilin tertentu berdasarkan indeks. Fitur ini menggunakan kombinasi **fungsi dengan referensi** dan **pointer**, yaitu:

- Harga diperbarui menggunakan **referensi** (&)
- Stok diperbarui menggunakan **pointer** (\*)

#### Source Code:

```

void perbaruiHargaLilin(string &hargaBaru) {
    cout << "Harga baru akan menjadi: ";
    getline(cin, hargaBaru);
}

void ubahStokLilin(string *stokPtr) {
    cout << "Stok baru akan menjadi: ";
    getline(cin, *stokPtr);
}

```

```

void perbaruiLilin(Sistem &s, int index) {
    if (index >= 0 && index < s.jumlahLilin) {
        perbaruiHargaLilin(s.lilin[index].harga);
        ubahStokLilin(&s.lilin[index].stok);
        cout << "Data berhasil diperbarui!\n";
    } else {
        cout << "Nomor lilin tidak valid.\n";
    }
}

```



## 4. Hapus Data Lilin

### Penjelasan:

Admin dapat menghapus lilin berdasarkan indeks. Setelah data dihapus, semua elemen setelahnya digeser ke atas untuk mengisi kekosongan.

### Source Code:

```
void hapusLilin(Sistem &s, int index) {
    if (index >= 0 && index < s.jumlahLilin) {
        for (int i = index; i < s.jumlahLilin - 1; i++) {
            s.lilin[i] = s.lilin[i + 1];
        }
        s.jumlahLilin--;
        cout << "Data berhasil dihapus!\n";
    } else {
        cout << "Nomor lilin tidak valid.\n";
    }
}
```

## 5. Total Stok Lilin

### Penjelasan:

Fitur ini menghitung **total seluruh stok lilin** yang tersedia menggunakan pendekatan **rekursif**, yaitu memanggil dirinya sendiri sampai mencapai indeks akhir.

### Source Code:

```
int totalStok(int index) {
    if (index == sistem.jumlahLilin) return 0;
    return stoi(sistem.lilin[index].stok) + totalStok(index + 1);
}
```

## C. Sorting Data Lilin

### 6. Sort Nama Lilin (Z → A)

Mengurutkan data lilin berdasarkan nama dari **huruf terakhir ke awal (Z ke A)** menggunakan **Bubble Sort**.

Source Code:

```
void sortNamaDescending(Sistem s) {
    for (int i = 0; i < s.jumlahLilin - 1; i++) {
        for (int j = 0; j < s.jumlahLilin - i - 1; j++) {
            if (s.lilin[j].nama < s.lilin[j + 1].nama) {
                swap(s.lilin[j], s.lilin[j + 1]);
            }
        }
    }
    lihatLilin(s);
}
```

### 7. Sort Harga Lilin (Termurah → Termahal)

Menyusun harga lilin dari yang **paling murah ke paling mahal** dengan **Selection Sort**.  
Dibandingkan dan ditukar hanya saat diperlukan.

Source Code:

```
void sortHargaAscending(Sistem s) {
    for (int i = 0; i < s.jumlahLilin - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < s.jumlahLilin; j++) {
            if (stoi(s.lilin[j].harga) < stoi(s.lilin[minIndex].harga)) {
                minIndex = j;
            }
        }
        swap(s.lilin[i], s.lilin[minIndex]);
    }
    lihatLilin(s);
}
```

## 8. Sort Stok Lilin (Terbanyak → Tersedikit)

Mengurutkan stok lilin dari yang **stok terbanyak ke paling sedikit** menggunakan **Insertion Sort**. Cocok untuk data yang hampir terurut.

**Source Code:**

```
void sortStokDescending(Sistem s) {
    for (int i = 1; i < s.jumlahLilin; i++) {
        Lilin key = s.lilin[i];
        int j = i - 1;
        while (j >= 0 && stoi(s.lilin[j].stok) < stoi(key.stok)) {
            s.lilin[j + 1] = s.lilin[j];
            j--;
        }
        s.lilin[j + 1] = key;
    }
    lihatLilin(s);
}
```

## 9. Keluar

Fitur ini memungkinkan admin keluar dari sistem dan menghentikan program. Ketika admin memilih menu ini, program akan menampilkan pesan keluar dan tidak mengulang ke menu utama lagi. Ini penting sebagai penutup aktivitas administrasi dalam sistem.

**Source Code:**

```
void ucapan() {
    cout << "Terima kasih telah menggunakan sistem ini!\n";
}

void ucapan(string nama) {
    cout << "Terima kasih, " << nama << ", telah menggunakan sistem ini!\n";
}
```

## 4. Uji Coba dan Hasil Output

### A. Login Admin

Setelah program dijalankan, pengguna diminta untuk memasukkan **nama** dan **NIM** yang sesuai dengan data admin ("**Andi Nurfadillah Hasan**" dan "**2409106087**"). Jika data benar, maka akan muncul pesan "**Login berhasil!**" dan pengguna akan diarahkan ke Menu Utama. Jika salah, akan muncul pesan gagal dan kesempatan login berkurang hingga maksimal 3 kali.

```
===== LOGIN ADMIN =====  
Masukkan Nama: Andi Nurfadillah Hasan  
Masukkan NIM: 2409106087  
Login berhasil!
```

Gambar 1. Login Admin Berhasil

```
===== LOGIN ADMIN =====  
Masukkan Nama: Andi Nurfadillah Hasan  
Masukkan NIM: 2409106080  
Login gagal! Sisa percobaan: 2  
  
===== LOGIN ADMIN =====  
Masukkan Nama: Andi Nurfadillah Hasan  
Masukkan NIM: 2409106085  
Login gagal! Sisa percobaan: 1  
  
===== LOGIN ADMIN =====  
Masukkan Nama: Andi Nurfadillah Hasan  
Masukkan NIM: 2409106086  
Login gagal! Sisa percobaan: 0  
Anda telah gagal login 3 kali. Program berhenti.
```

Gambar 2. Login Admin Gagal

### B. Menu Utama

Setelah berhasil login, program menampilkan daftar menu utama dengan berbagai fitur yang bisa dipilih oleh admin, yaitu:

1. Tambah Lilin
2. Lihat Data Lilin
3. Perbarui Data Lilin
4. Hapus Data Lilin

5. Total Stok Lilin
6. Nama Lilin (Bubble Sort)
7. Harga Lilin (Selection Sort)
8. Stok Lilin (Insertion Sort)
9. Keluar

## 1. Menu Tambah Lilin

Pengguna diminta memasukkan data lilin baru berupa: nama, harga, dan stok. Setelah data diisi, program menyimpan data tersebut dan menampilkan pesan berhasil.

```
Pilih menu: 1
Masukkan nama lilin: Apple Cinnamon
Masukkan harga lilin (Rp): 37000
Masukkan stok lilin (pcs): 8
Data berhasil ditambahkan!
```

Gambar 1.1 Menu Tambah Lilin

## 2. Menu Lihat Data Lilin

Menampilkan seluruh data lilin yang tersimpan dalam bentuk tabel dengan kolom: nomor, nama lilin, harga, dan stok.

```
Pilih menu: 2
+=====+
| No | Nama Lilin | Harga (Rp) | Stok (pcs) |
+=====+
| 1 | Lavender Bliss | 35000 | 12 |
| 2 | Ocean Breeze | 40000 | 7 |
| 3 | Vanilla Dream | 32000 | 15 |
| 4 | Rose Garden | 45000 | 9 |
| 5 | Lemon Zest | 30000 | 18 |
| 6 | Cinnamon Spice | 38000 | 5 |
| 7 | Mint Fresh | 29000 | 10 |
| 8 | Jasmine Bloom | 41000 | 13 |
| 9 | Coffee Scent | 36000 | 11 |
| 10 | Woodland Pine | 47000 | 6 |
| 11 | Apple Cinnamon | 37000 | 8 |
+=====+
```

Gambar 2.1 Menu Lihat Data Lilin

### 3. Menu Perbarui Data Lilin

Pengguna memilih nomor lilin yang ingin diperbarui. Setelah itu, diminta memasukkan harga dan stok yang baru. Program lalu memperbarui data tersebut.

```
Pilih menu: 3

+-----+
| No | Nama Lilin | Harga (Rp) | Stok (pcs) |
+-----+
| 1 | Lavender Bliss | 35000 | 12 |
| 2 | Ocean Breeze | 40000 | 7 |
| 3 | Vanilla Dream | 32000 | 15 |
| 4 | Rose Garden | 45000 | 9 |
| 5 | Lemon Zest | 30000 | 18 |
| 6 | Cinnamon Spice | 38000 | 5 |
| 7 | Mint Fresh | 29000 | 10 |
| 8 | Jasmine Bloom | 41000 | 13 |
| 9 | Coffee Scent | 36000 | 11 |
| 10 | Woodland Pine | 47000 | 6 |
| 11 | Apple Cinnamon | 37000 | 8 |
+-----+

Masukkan nomor lilin yang ingin diperbarui: 11
Harga baru akan menjadi: 36000
Stok baru akan menjadi: 14
Data berhasil diperbarui!
```

Gambar 3.1 Menu Perbarui Data Lilin

```
+-----+
| No | Nama Lilin | Harga (Rp) | Stok (pcs) |
+-----+
| 1 | Lavender Bliss | 35000 | 12 |
| 2 | Ocean Breeze | 40000 | 7 |
| 3 | Vanilla Dream | 32000 | 15 |
| 4 | Rose Garden | 45000 | 9 |
| 5 | Lemon Zest | 30000 | 18 |
| 6 | Cinnamon Spice | 38000 | 5 |
| 7 | Mint Fresh | 29000 | 10 |
| 8 | Jasmine Bloom | 41000 | 13 |
| 9 | Coffee Scent | 36000 | 11 |
| 10 | Woodland Pine | 47000 | 6 |
| 11 | Apple Cinnamon | 36000 | 14 |
+-----+
```

Gambar 3.2 Data Lilin berhasil diperbarui

#### 4. Menu Hapus Data Lilin

Pengguna memilih nomor lilin yang ingin dihapus. Program akan menghapus data lilin tersebut dan menampilkan pesan berhasil.

```
Pilih menu: 4
+-----+
| No | Nama Lilin      | Harga (Rp) | Stok (pcs) |
+-----+
| 1  | Lavender Bliss  | 35000      | 12         |
| 2  | Ocean Breeze    | 40000      | 7          |
| 3  | Vanilla Dream   | 32000      | 15         |
| 4  | Rose Garden     | 45000      | 9          |
| 5  | Lemon Zest      | 30000      | 18         |
| 6  | Cinnamon Spice  | 38000      | 5          |
| 7  | Mint Fresh      | 29000      | 10         |
| 8  | Jasmine Bloom   | 41000      | 13         |
| 9  | Coffee Scent    | 36000      | 11         |
| 10 | Woodland Pine   | 47000      | 6          |
| 11 | Apple Cinnamon  | 36000      | 14         |
+-----+
Masukkan nomor lilin yang ingin dihapus: 11
Data berhasil dihapus!
```

Gambar 4.1 Menu Hapus Data Lilin

```
+-----+
| No | Nama Lilin      | Harga (Rp) | Stok (pcs) |
+-----+
| 1  | Lavender Bliss  | 35000      | 12         |
| 2  | Ocean Breeze    | 40000      | 7          |
| 3  | Vanilla Dream   | 32000      | 15         |
| 4  | Rose Garden     | 45000      | 9          |
| 5  | Lemon Zest      | 30000      | 18         |
| 6  | Cinnamon Spice  | 38000      | 5          |
| 7  | Mint Fresh      | 29000      | 10         |
| 8  | Jasmine Bloom   | 41000      | 13         |
| 9  | Coffee Scent    | 36000      | 11         |
| 10 | Woodland Pine   | 47000      | 6          |
+-----+
```

Gambar 4.2 Data Lilin berhasil dihapus

#### 5. Menu Total Stok Lilin

Program menghitung total stok semua lilin menggunakan **fungsi rekursif** `totalStok()` dan menampilkannya.

```
Pilih menu: 5
Total seluruh stok lilin: 106 pcs
```

Gambar 5.1 Menu Total Stok Lilin

## 6. Nama Lilin (Bubble Sort)

Menyortir nama lilin secara **descending (Z → A)** menggunakan metode **Bubble Sort** dan menampilkannya kembali.

Pilih menu: 6

No	Nama Lilin	Harga (Rp)	Stok (pcs)
1	Woodland Pine	47000	6
2	Vanilla Dream	32000	15
3	Rose Garden	45000	9
4	Ocean Breeze	40000	7
5	Mint Fresh	29000	10
6	Lemon Zest	30000	18
7	Lavender Bliss	35000	12
8	Jasmine Bloom	41000	13
9	Coffee Scent	36000	11
10	Cinnamon Spice	38000	5

Gambar 6.1 Nama Lilin (Bubble Sort)

## 7. Harga Lilin (Selection Sort)

Menyortir data berdasarkan harga lilin secara **ascending (termurah ke termahal)** menggunakan **Selection Sort**.

Pilih menu: 7

No	Nama Lilin	Harga (Rp)	Stok (pcs)
1	Mint Fresh	29000	10
2	Lemon Zest	30000	18
3	Vanilla Dream	32000	15
4	Lavender Bliss	35000	12
5	Coffee Scent	36000	11
6	Cinnamon Spice	38000	5
7	Ocean Breeze	40000	7
8	Jasmine Bloom	41000	13
9	Rose Garden	45000	9
10	Woodland Pine	47000	6

Gambar 7.1 Harga Lilin (Selection Sort)



## 8. Stok Lilin (Insertion Sort)

Menyortir lilin berdasarkan **stok terbanyak ke yang paling sedikit** menggunakan **Insertion Sort**.

```
Pilih menu: 8
```

No	Nama Lilin	Harga (Rp)	Stok (pcs)
1	Lemon Zest	30000	18
2	Vanilla Dream	32000	15
3	Jasmine Bloom	41000	13
4	Lavender Bliss	35000	12
5	Coffee Scent	36000	11
6	Mint Fresh	29000	10
7	Rose Garden	45000	9
8	Ocean Breeze	40000	7
9	Woodland Pine	47000	6
10	Cinnamon Spice	38000	5

Gambar 8.1 Stok Lilin (Insertion Sort)

## 9. Keluar

Program menampilkan pesan ucapan terima kasih (menggunakan **overload prosedur ucapan()**) dan kemudian keluar dari sistem.

```
Pilih menu: 9
Terima kasih, Andi Nurfadillah Hasan, telah menggunakan sistem ini!
```

Gambar 9.1 Keluar

## 5. Langkah-Langkah Git pada VSCode

### 5.1 Git Add

Git init adalah perintah untuk menginisialisasi repository Git dalam suatu folder, sehingga memungkinkan pelacakan perubahan dalam proyek.

```
USER@LAPTOP-FE0F865G MINGW64 /d/praktikum-apl (main)
$ git add .
```

### 5.2 Git Commit

Git commit adalah perintah untuk menyimpan perubahan yang sudah ada di staging area ke dalam repository Git. Commit ini seperti "checkpoint" dalam proyek, yang mencatat perubahan dengan pesan deskriptif.

```
USER@LAPTOP-FE0F865G MINGW64 /d/praktikum-apl (main)
$ git commit -m "upload code"
[main d1cc295] upload code
2 files changed, 234 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106087-AndiNurfadillahHasan-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106087-AndiNurfadillahHasan-PT-6.exe
```

### 5.3 Git Push

Git Push adalah perintah dalam Git yang digunakan untuk mengupload commit dari repository lokal ke repository remote.

```
USER@LAPTOP-FE0F865G MINGW64 /d/praktikum-apl (main)
$ git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 682.22 KiB | 4.84 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/fadydv/praktikum-apl/
 c7fd7c2..d1cc295  main -> main
```