

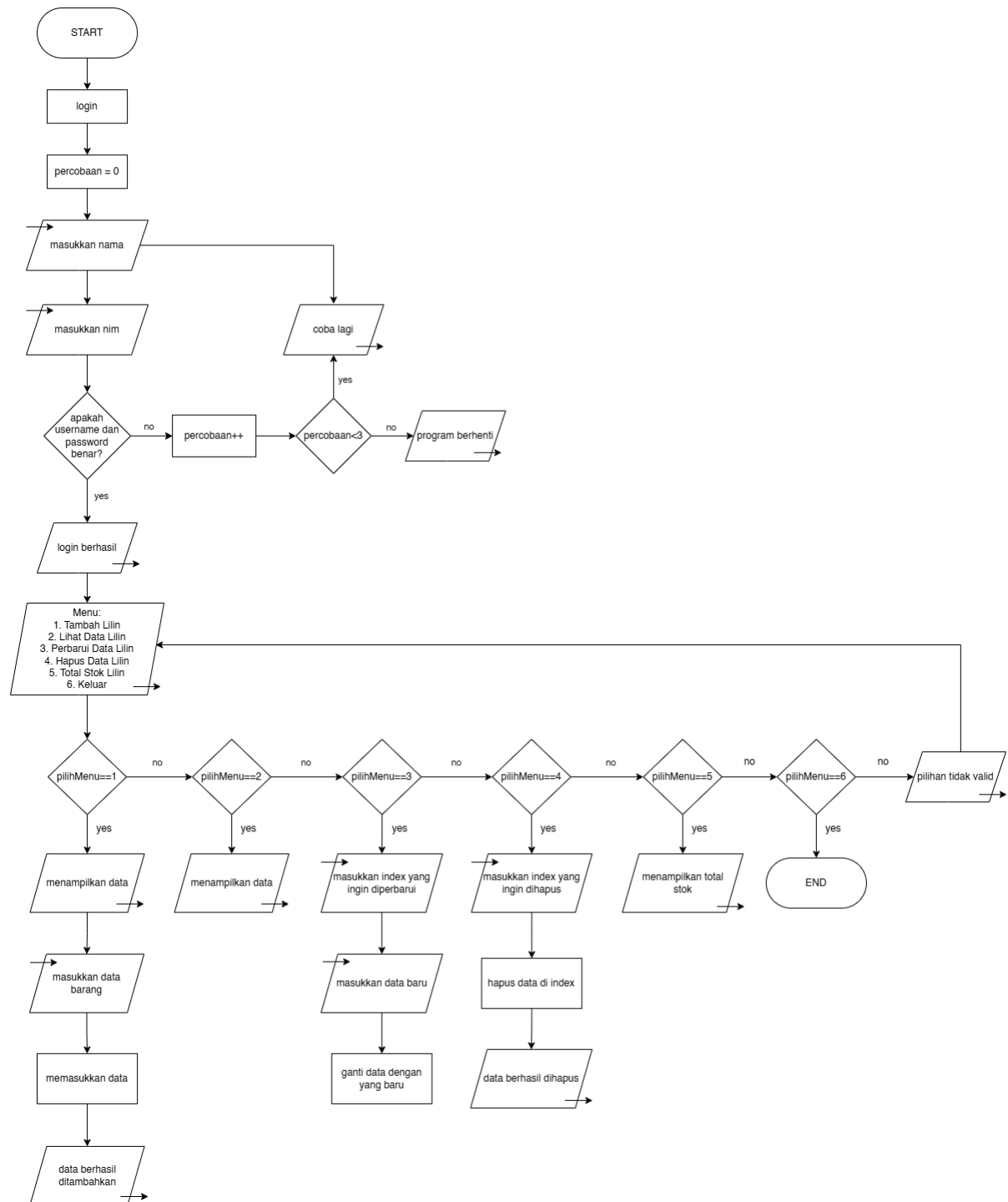
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



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 dibuat untuk membantu admin dalam melakukan pengelolaan data lilin secara efisien. Tujuan utama dari program ini adalah untuk memudahkan admin dalam **menambah, melihat, memperbarui, menghapus, dan menghitung total stok lilin** yang tersedia dalam bentuk antarmuka berbasis teks.

Setiap data lilin disimpan dalam **array of struct** yang mencakup informasi:

- **Nama Lilin**
- **Harga (dalam Rupiah)**
- **Jumlah Stok (dalam pcs)**

Program ini menggunakan **subprogram berupa fungsi dan prosedur** untuk membagi tugas-tugas utama seperti menampilkan data, menambah data, dan menghitung total stok lilin. Untuk menambah fleksibilitas, juga digunakan **fungsi overloading** (dengan parameter dan tanpa parameter), serta **fungsi rekursif** untuk menghitung total stok lilin secara otomatis.

Agar lebih aman, sistem ini juga dilengkapi dengan **fitur login**. Pengguna hanya dapat melakukan akses jika berhasil memasukkan **username dan password admin yang benar**. Jika gagal login sebanyak **tiga kali berturut-turut**, program akan menghentikan proses secara otomatis sebagai bentuk pengamanan data.

Setelah berhasil login, admin akan dihadapkan pada menu utama yang terdiri dari enam pilihan, yaitu:

1. **Tambah Lilin** – Menambahkan data lilin baru ke dalam sistem.
2. **Lihat Data Lilin** – Menampilkan semua data lilin dalam bentuk tabel.
3. **Perbarui Data Lilin** – Mengubah informasi pada lilin tertentu.
4. **Hapus Data Lilin** – Menghapus data lilin yang sudah tidak diperlukan.
5. **Total Stok Lilin** – Menghitung total seluruh stok lilin menggunakan fungsi rekursif.
6. **Keluar** – Mengakhiri program dengan menampilkan pesan ucapan (menggunakan fungsi overloading).

3. Source Code

A. Fitur Login

Fitur ini memastikan bahwa hanya admin dengan nama dan NIM yang valid yang dapat mengakses sistem.

Penjelasan Cara Kerja:

- Program meminta admin untuk memasukkan nama dan NIM.
- Jika sesuai dengan data yang telah ditentukan (Andi Nurfadillah Hasan, 2409106087), maka login berhasil.
- Jika tidak cocok, program memberi kesempatan sebanyak **3 kali**.
- Jika gagal 3 kali berturut-turut, program menampilkan pesan kegagalan dan berhenti.

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";
        }
    }
}
```

B. Menu Utama

Setelah login, admin akan masuk ke **menu utama** yang terdiri dari beberapa pilihan penting, seperti:

1. Tambah Lilin

Digunakan untuk menambahkan data lilin baru ke dalam sistem.

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 semua data lilin yang sudah tersimpan dalam sistem.

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 untuk memperbarui harga dan stok lilin berdasarkan nomor indeks.

Source Code:

```

void perbaruililin(Sistem &s, int index) {
    if (index >= 0 && index < s.jumlahLilin) {
        cout << "Masukkan harga baru lilin (Rp): ";
        getline(cin, s.lilin[index].harga);
        cout << "Masukkan stok baru lilin (pcs): ";
        getline(cin, s.lilin[index].stok);
        cout << "Data berhasil diperbarui!\n";
    } else {
        cout << "Nomor lilin tidak valid.\n";
    }
}

```

4. Hapus Data Lilin

Menghapus data lilin berdasarkan indeks, lalu menggeser data berikutnya ke atas.

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

Menggunakan fungsi rekursif untuk menghitung total stok lilin secara keseluruhan.

Source Code:

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

6. Keluar

Menggunakan fungsi overloading untuk ucapan terima kasih.

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";  
}
```

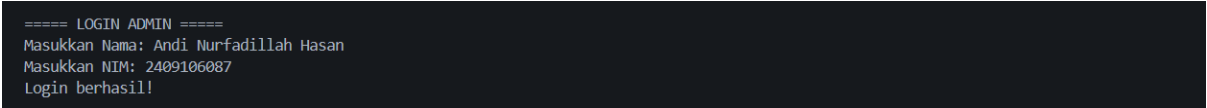
4. Uji Coba dan Hasil Output

A. Login Admin

Fitur login bertujuan membatasi akses hanya kepada admin yang memiliki username dan password yang sesuai. Program akan memberikan tiga kali kesempatan login. Jika semua kesempatan habis, program akan berhenti.

Penjelasan Cara Kerja:

1. Program meminta input **Nama** dan **NIM**.
2. Jika cocok dengan data admin (Andi Nurfadillah Hasan, 2409106087), maka login berhasil dan lanjut ke menu utama.
3. Jika salah, program menampilkan “Login gagal! Sisa percobaan: x”.
4. Jika gagal 3 kali, program akan menampilkan “Anda telah gagal login 3 kali. Program berhenti.”



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

Gambar 1. Login Admin

B. Menu Utama

Setelah login berhasil, sistem menampilkan menu utama. Pengguna bisa memilih berbagai aksi sesuai kebutuhan pengelolaan data lilin.

1. Menu Tambah Lilin

Penjelasan:

Menu ini digunakan untuk menambahkan data lilin baru ke dalam sistem. Pengguna akan diminta memasukkan:

- Nama lilin
- Harga lilin
- Stok lilin

Jika data berhasil ditambahkan, sistem akan memberikan pesan konfirmasi.


```
Pilih menu: 1
Masukkan nama lilin: Vanilla Relax
Masukkan harga lilin (Rp): 25000
Masukkan stok lilin (pcs): 15
Data berhasil ditambahkan!
```

Gambar 2. Menu Tambah Lilin

2. Menu Lihat Data Lilin

Penjelasan:

Menu ini akan menampilkan daftar semua lilin yang sudah ditambahkan sebelumnya dalam bentuk tabel. Informasi yang ditampilkan meliputi:

- Nomor urut
- Nama lilin
- Harga
- Stok

Jika belum ada data, maka sistem akan memberi informasi bahwa data masih kosong.

```
Pilih menu: 2

+=====+
| No | Nama Lilin      | Harga (Rp) | Stok (pcs) |
+=====+
| 1  | Vanilla Relax  | 25000     | 15         |
+=====+
```

Gambar 3. Menu Lihat Data Lilin

3. Menu Perbarui Data Lilin

Penjelasan:

Digunakan untuk memperbarui harga dan stok dari lilin yang sudah ada. Pengguna harus memasukkan nomor urut lilin yang ingin diperbarui.

```
Pilih menu: 3
Masukkan nomor lilin: 1
Masukkan harga baru lilin (Rp): 27000
Masukkan stok baru lilin (pcs): 25
Data berhasil diperbarui!
```

Gambar 4. Menu Perbarui Data Lilin

4. Menu Hapus Data Lilin

Penjelasan:

Menu ini digunakan untuk menghapus data lilin berdasarkan nomor urut. Setelah penghapusan berhasil, data akan bergeser untuk mengisi posisi yang kosong.

```
Pilih menu: 4
Masukkan nomor lilin: 1
Data berhasil dihapus!
```

Gambar 5. Menu Hapus Data Lilin

5. Menu Total Stok Lilin

Penjelasan:

Menggunakan **fungsi rekursif** untuk menghitung total seluruh stok lilin dari awal sampai akhir data. Ini menunjukkan penggunaan teknik rekursi dalam program.

```
+=====+
| No | Nama Lilin      | Harga (Rp) | Stok (pcs) |
+=====+
| 1  | Lily            | 25000      | 25         |
| 2  | Vanilla Relax   | 27000      | 15         |
+=====+
```

Gambar 6.1 Menu 2

```
Pilih menu: 5
Total seluruh stok lilin: 40 pcs
```

Gambar 6.2 Menu Total Stok Lilin

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 1f1a22c] upload code
2 files changed, 157 insertions(+)
create mode 100644 post-test/post-test-apl-4/2409106087-AndiNurfadillahHasan-PT-4.cpp
create mode 100644 post-test/post-test-apl-4/2409106087-AndiNurfadillahHasan-PT-4.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), 678.88 KiB | 6.01 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/
 dd881da..1f1a22c  main -> main
```