

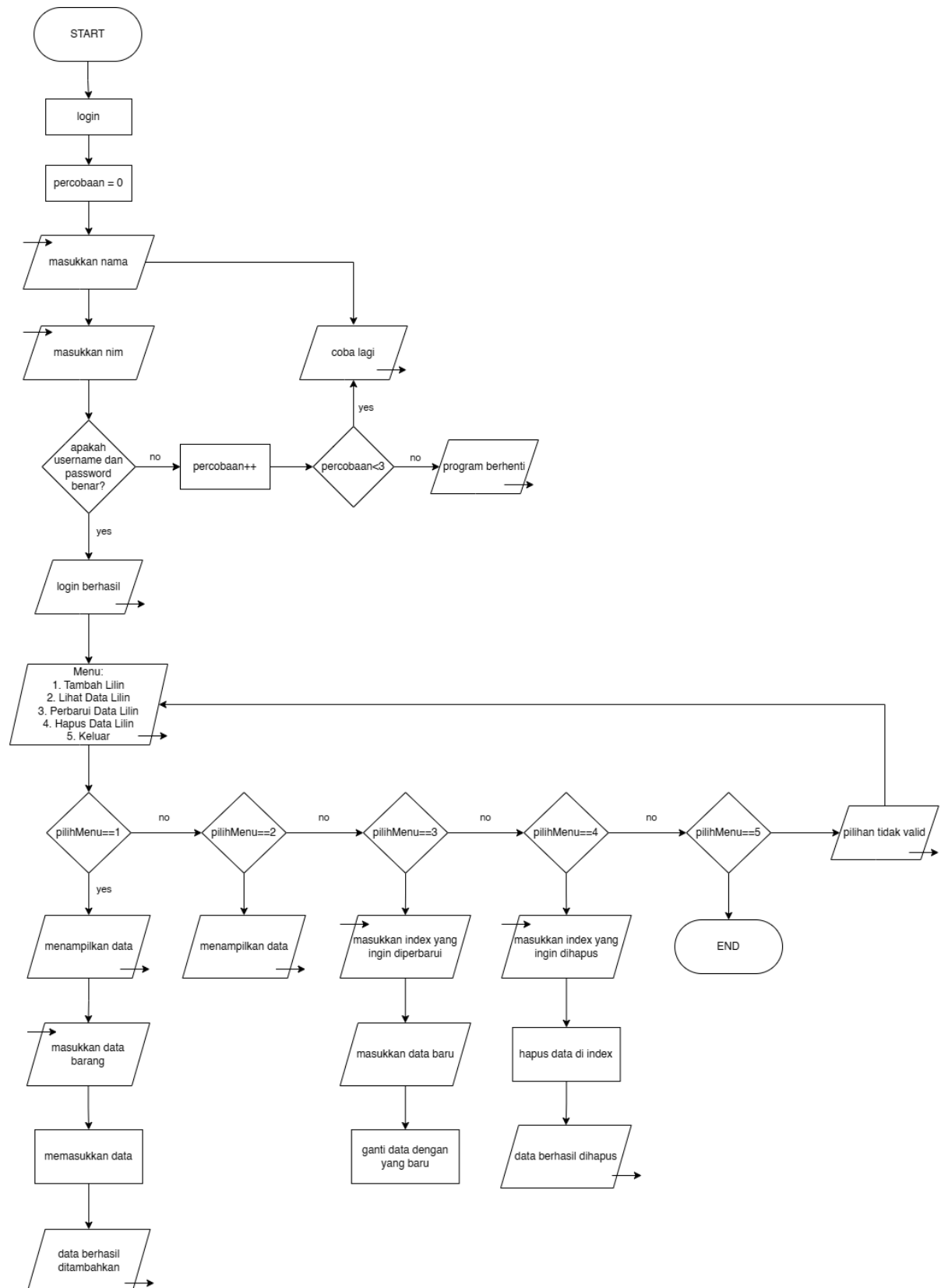
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



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 data lilin aromaterapi** yang bertujuan untuk mempermudah admin dalam mengelola data lilin, termasuk **menambah, melihat, memperbarui, dan menghapus** data lilin yang tersedia.

Setiap data lilin akan disimpan dalam **array of struct** yang berisi **Nama Lilin, Harga (Rp), dan Stok (pcs)**. Dengan tampilan berbentuk tabel, admin dapat lebih mudah membaca serta mengelola data yang telah dimasukkan.

Program juga dilengkapi dengan **sistem login** untuk memastikan hanya admin yang memiliki kredensial yang benar yang dapat mengakses fitur-fitur di dalamnya. Jika admin salah memasukkan username atau password sebanyak **tiga kali**, program akan otomatis berhenti untuk menjaga keamanan data.

Setelah berhasil login, admin dapat memilih beberapa fitur utama melalui menu yang disediakan, yaitu:

1. **Tambah Lilin** – Menambahkan data lilin baru ke dalam sistem.
2. **Lihat Data Lilin** – Menampilkan daftar lilin dalam bentuk tabel yang rapi.
3. **Perbarui Data Lilin** – Memperbarui informasi lilin yang sudah ada.
4. **Hapus Data Lilin** – Menghapus data lilin yang tidak diperlukan.
5. **Keluar** – Menghentikan program.

3. Source Code

A. Fitur Login

Fitur login dalam program ini berfungsi untuk memastikan bahwa hanya admin yang memiliki kredensial yang benar yang dapat mengakses sistem.

Penjelasan Cara Kerja:

- Program meminta pengguna memasukkan nama dan NIM sebagai kredensial login.
- Jika nama dan NIM sesuai dengan data admin yang telah ditentukan, program akan menampilkan pesan "Login berhasil!", dan pengguna dapat mengakses menu utama untuk melakukan pengelolaan data lilin.
- Jika pengguna salah memasukkan nama atau NIM, program akan memberikan tiga kali kesempatan untuk mencoba kembali.
- Jika pengguna gagal login sebanyak tiga kali berturut-turut, program akan menampilkan pesan "Anda telah gagal login 3 kali. Program berhenti." dan langsung menghentikan proses eksekusi program.
- Setelah berhasil login, admin dapat mengakses menu yang mencakup fitur CRUD untuk mengelola data lilin, termasuk menambah, melihat, memperbarui, dan menghapus data lilin yang ada dalam sistem.

```
int main() {
    sistem.jumlahLilin = 0;
    sistem.admin = {"Andi Nurfadillah Hasan", "2409106087"};

    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

Setelah berhasil login, pengguna akan masuk ke menu utama yang memiliki 5 opsi utama:

Pilihan 1: Tambah Lilin

Fungsi ini digunakan untuk menambahkan data lilin baru ke dalam sistem. Pengguna akan diminta untuk memasukkan nama lilin, harga, dan jumlah stoknya.

Source Code:

```

if (pilihan == 1) {
    if (sistem.jumlahLilin < MAX_LILIN) {
        cout << "Masukkan nama lilin: ";
        getline(cin, sistem.lilin[sistem.jumlahLilin].nama);
        cout << "Masukkan harga lilin (Rp): ";
        getline(cin, sistem.lilin[sistem.jumlahLilin].harga);
        cout << "Masukkan stok lilin (pcs): ";
        getline(cin, sistem.lilin[sistem.jumlahLilin].stok);
        sistem.jumlahLilin++;
        cout << "Data berhasil ditambahkan!\n";
    } else {
        cout << "Kapasitas penuh! Tidak bisa menambah lilin
lagi.\n";
    }
}

```

Pilihan 2: Lihat Data Lilin

Opsi ini digunakan untuk melihat semua data lilin yang telah tersimpan dalam sistem. Jika tidak ada data, akan ditampilkan pesan bahwa data lilin masih kosong.

Source Code:

```
    } else if (pilihan == 2) {
        if (sistem.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 < sistem.jumlahLilin; i++) {
                cout << "| " << setw(3) << left << i + 1 << " | "
                    << setw(20) << left << sistem.lilin[i].nama << " |
"
                    << setw(12) << right << sistem.lilin[i].harga << "
| "
                    << setw(10) << right << sistem.lilin[i].stok << "
|\n";
            }
            cout <<
"+=====+\n";
        }
    }
```

Pilihan 3: Perbarui Data Lilin

Fitur ini memungkinkan pengguna untuk memperbarui harga dan stok lilin yang telah ada dalam daftar berdasarkan nomor indeks lilin.

Source Code:

```
    } else if (pilihan == 3) {
        if (sistem.jumlahLilin == 0) {
            cout << "Belum ada data lilin.\n";
            continue;
        }
    }
```

```

    cout << "Masukkan nomor lilin yang ingin diperbarui: ";
    int indeks;
    cin >> indeks;
    cin.ignore();

    if (indeks > 0 && indeks <= sistem.jumlahLilin) {
        cout << "Masukkan harga baru lilin (Rp): ";
        getline(cin, sistem.lilin[indeks - 1].harga);
        cout << "Masukkan stok baru lilin (pcs): ";
        getline(cin, sistem.lilin[indeks - 1].stok);
        cout << "Data berhasil diperbarui!\n";
    } else {
        cout << "Nomor lilin tidak valid.\n";
    }
}

```

Pilihan 4: Hapus Data Lilin

Opsi ini digunakan untuk menghapus lilin berdasarkan nomor indeksnya. Data yang dihapus akan dihapus dari array dan data berikutnya akan digeser ke atas.

Source Code:

```

    } else if (pilihan == 4) {
        if (sistem.jumlahLilin == 0) {
            cout << "Belum ada data lilin.\n";
            continue;
        }
        cout << "Masukkan nomor lilin yang ingin dihapus: ";
        int indeks;
        cin >> indeks;

        if (indeks > 0 && indeks <= sistem.jumlahLilin) {
            for (int i = indeks - 1; i < sistem.jumlahLilin - 1; i++) {
                sistem.lilin[i] = sistem.lilin[i + 1];
            }
            sistem.jumlahLilin--;
            cout << "Data berhasil dihapus!\n";
        } else {
            cout << "Nomor lilin tidak valid.\n";
        }
    }
}

```

Pilihan 5: Keluar dari Program

Jika pengguna memilih opsi ini, program akan menampilkan pesan terima kasih dan keluar dari sistem.

Source Code:

```
    } else if (pilihan == 5) {  
        cout << "Terima kasih! Program berhenti.\n";  
        break;  
    } else {  
        cout << "Pilihan tidak valid!\n";  
    }  
}  
return 0;  
}
```

C. Struct dan Nested Struct

Penjelasan

Dalam kode yang kamu kirim sebelumnya, terdapat nested struct yang digunakan untuk menyimpan informasi mengenai admin dan lilin.

- Struct **Lilin** menyimpan informasi tentang nama, harga, dan stok lilin.
- Struct **Admin** menyimpan username dan password admin.
- Struct **Sistem** yang berisi array lilin dan informasi admin.
 - **Sistem** memiliki array **Lilin** yang menyimpan daftar lilin.
 - **Sistem** juga memiliki nested struct **Admin** yang menyimpan informasi login admin.

Source Code:

```
struct Lilin {
    string nama;
    string harga;
    string stok;
};

struct Admin {
    string username;
    string password;
};

struct Sistem {
    Lilin lilin[MAX_LILIN];
    Admin admin;
    int jumlahLilin;
};
```

4. Uji Coba dan Hasil Output

A. Kode Login Admin

Kode login ini memastikan bahwa hanya admin yang memiliki username dan password yang benar yang bisa mengakses sistem. Jika salah, program akan memberi kesempatan login sebanyak 3 kali, dan jika tetap gagal, program akan berhenti.

Penjelasan Cara Kerja

1. **Program meminta input Nama dan NIM.**
2. **Jika Nama & NIM benar**, program menampilkan "**Login berhasil!**" dan masuk ke menu utama.
3. **Jika Nama atau NIM salah**, program menampilkan pesan "**Login gagal! Sisa percobaan: x**".
4. **Jika gagal sebanyak 3 kali**, program menampilkan "**Anda telah gagal login 3 kali. Program berhenti.**", lalu program berhenti.

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

B. Menu Utama

Penjelasan Cara Kerja Menu Utama

1. **Menampilkan pilihan menu** setelah login berhasil.
2. **Menunggu input pengguna** untuk memilih menu yang diinginkan.
3. **Jika memilih:**
 - **1 (Tambah Lilin)** → Memasukkan nama, harga, dan stok lilin baru.
 - **2 (Lihat Data Lilin)** → Menampilkan seluruh daftar lilin yang tersedia.

- **3 (Perbarui Data Lilin)** → Mengubah harga dan stok lilin berdasarkan nomor indeks.
- **4 (Hapus Data Lilin)** → Menghapus lilin berdasarkan nomor indeks.
- **5 (Keluar)** → Menghentikan program.

```
Pilih menu: 1
Masukkan nama lilin: Lavender
Masukkan harga lilin (Rp): 25000
Masukkan stok lilin (pcs): 10
Data berhasil ditambahkan!
```

Gambar 1.1 Pilihan 1

```
Pilih menu: 2

+-----+
| No  | Nama Lilin      | Harga (Rp)  | Stok (pcs) |
+-----+
| 1   | Lavender        | 25000       | 10         |
+-----+
```

Gambar 1.2 Pilihan 2

```
Pilih menu: 3
Masukkan nomor lilin yang ingin diperbarui: 1
Masukkan harga baru lilin (Rp): 30000
Masukkan stok baru lilin (pcs): 15
Data berhasil diperbarui!
```

Gambar 1.3 Pilihan 3

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

Gambar 1.4 Pilihan 4

```
Pilih menu: 5
Terima kasih! Program berhenti.
```

Gambar 1.5 Pilihan 5

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/post-test (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/post-test (main)
$ git commit -m "upload code"
[main 0e8dd56] upload code
 2 files changed, 142 insertions(+)
 create mode 100644 post-test/post-test-apl-3/2409106087-AndiNurfadillahHasan-PT-3.cpp
 create mode 100644 post-test/post-test-apl-3/2409106087-AndiNurfadillahHasan-PT-3.exe
```

5.5 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/post-test (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), 678.09 KiB | 4.95 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/
 0b530c4..0e8dd56  main -> main
```

