LAPORAN PRAKTIKUM

POSTTEST 5

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Andi Nurfadillah Hasan (2409106087)

Kelas (B2 ‘24)

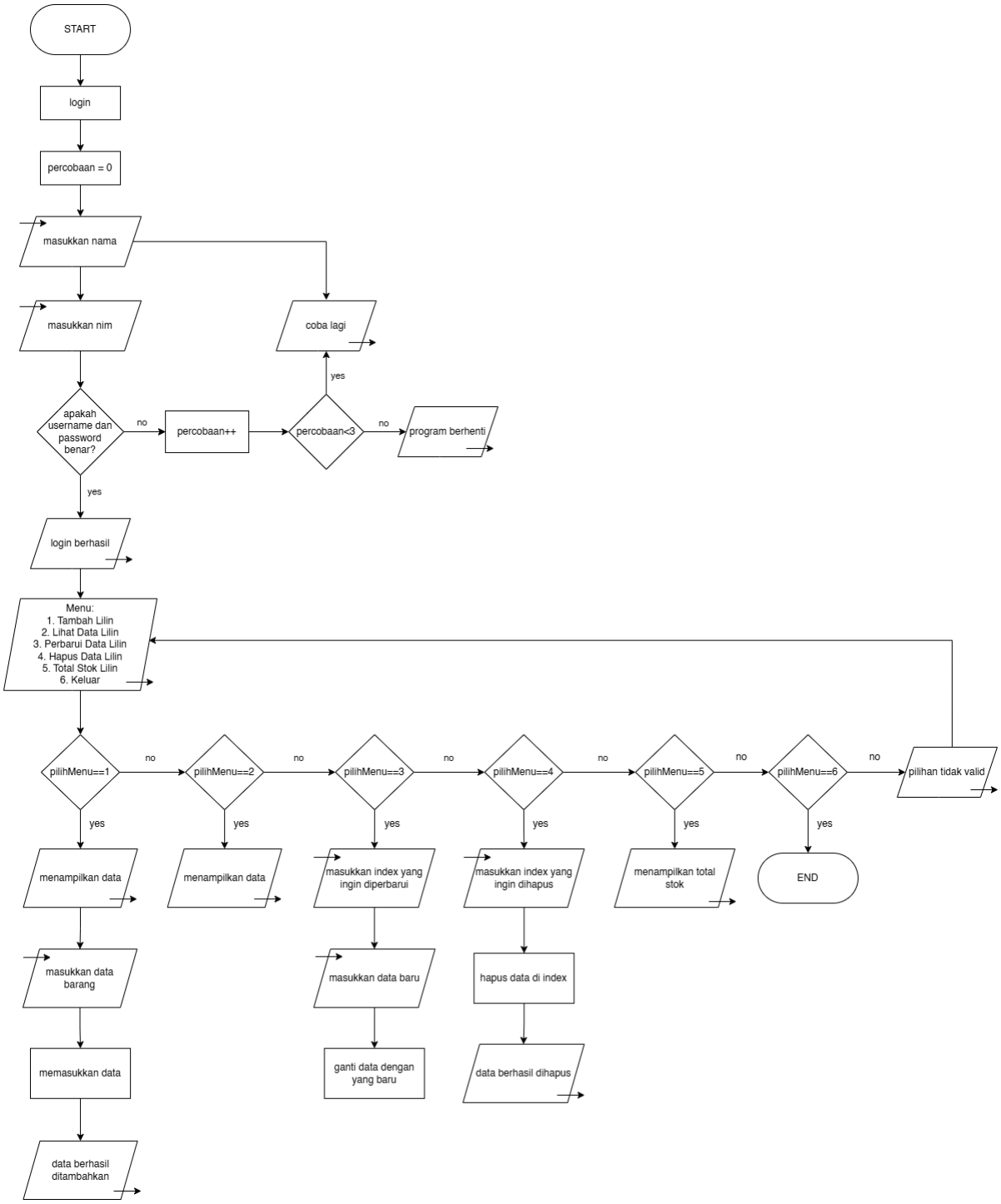
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

1. Flowchart


START
login
percobaan = 0
masukkan nama
masukkan nim
coba lagi
yes
apakah username dan password benar?
no
percobaan++
percobaan<3
no
program berhenti
yes
login berhasil
Menu:
1. Tambah Lilin
2. Lihat Data Lilin
3. Perbarui Data Lilin
4. Hapus Data Lilin
5. Total Stok Lilin
6. Keluar
pilihMenu==1
no
pilihMenu==2
no
pilihMenu==3
no
pilihMenu==4
no
pilihMenu==5
no
pilihMenu==6
no
pilihan tidak valid
yes
menampilkan data
masukkan data barang
memasukkan data
data berhasil ditambahkan
menampilkan data
masukkan index yang ingin diperbarui
masukkan data baru
ganti data dengan yang baru
masukkan index yang ingin dihapus
hapus data di index
data berhasil dihapus
menampilkan total stok
END

2. Analisis Program

Program ini merupakan sistem manajemen lilin aromaterapi yang dikembangkan sebagai lanjutan dari program pada Posttest 4. Pengembangan ini dilakukan untuk menerapkan konsep **pointer** dalam pengelolaan data, sesuai dengan materi yang dibahas pada Modul 5. Tujuan dari program ini adalah untuk memberikan fleksibilitas yang lebih tinggi dalam memanipulasi data secara langsung melalui alamat memori, khususnya saat memperbarui informasi lilin.

Setiap data lilin tetap disimpan dalam array of struct yang berisi informasi:

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

Pada pengembangan ini, program menggunakan subprogram dalam bentuk fungsi yang mengimplementasikan **pointer**. Terdapat dua fungsi utama yang menjadi fokus:

- Fungsi perbaruiHargaLilin(string &hargaBaru) menggunakan **parameter address-of (&)** agar nilai harga dapat dimodifikasi langsung melalui referensi tanpa membuat salinan data.
- Fungsi ubahStokLilin(string *stokPtr) menggunakan **parameter dereference (*)** untuk mengakses dan mengubah stok lilin berdasarkan alamat memori yang diterima sebagai parameter.

Selain penambahan fungsi pointer, program tetap mempertahankan struktur awal, termasuk fitur login untuk keamanan akses. Pengguna hanya dapat masuk jika memasukkan nama dan NIM admin yang sesuai. Apabila gagal melakukan login sebanyak tiga kali, maka program akan berhenti secara otomatis untuk mencegah akses tidak sah.

Setelah berhasil login, admin akan diarahkan ke menu utama dengan 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 harga dan stok lilin menggunakan konsep pointer
4. **Hapus Data Lilin** – Menghapus data lilin dari sistem
5. **Total Stok Lilin** – Menghitung total stok lilin
6. **Keluar** – Mengakhiri program dan menampilkan pesan ucapan menggunakan fungsi overloading

Dengan penerapan pointer dalam bentuk **address-of** dan **dereference**, program ini berhasil memenuhi instruksi pengembangan lanjutan dari tugas sebelumnya, sekaligus mengaplikasikan prinsip penting dalam pemrograman tingkat lanjut.

3. Source Code

A. Fitur Login

Fitur login digunakan untuk memverifikasi identitas admin sebelum mengakses sistem.

Penjelasan Cara Kerja:

- Program meminta admin untuk memasukkan **nama** dan **NIM**.
- Sistem memeriksa apakah nama dan NIM yang dimasukkan sesuai dengan data yang telah ditentukan (Andi Nurfadillah Hasan, 2409106087).
- Jika login berhasil, program melanjutkan ke menu utama.
- Jika login gagal, admin diberi kesempatan sebanyak 3 kali untuk mencoba.
- Jika admin gagal login sebanyak 3 kali, program berhenti dan menampilkan pesan kegagalan.

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

Menu utama memberikan beberapa pilihan yang dapat dipilih oleh admin setelah login berhasil.

Penjelasan Cara Kerja:

- Setelah login berhasil, admin akan masuk ke dalam menu utama.
- Menu utama memiliki beberapa pilihan untuk mengelola data lilin:

1. Tambah Lilin

Fitur Tambah Lilin memungkinkan admin untuk menambahkan data lilin baru ke dalam sistem.

Penjelasan:

- **Fungsi tambahLilin** digunakan untuk menambahkan lilin baru ke dalam sistem.
- Program memeriksa apakah jumlah lilin yang ada masih kurang dari kapasitas maksimum (MAX_LILIN).
- Jika masih ada ruang, admin diminta untuk memasukkan nama, harga, dan stok lilin baru.
- Setelah data dimasukkan, jumlah lilin bertambah (s.jumlahLilin++), dan pesan "Data berhasil ditambahkan!" ditampilkan.
- Jika kapasitas sistem penuh, akan muncul pesan "Kapasitas penuh!".

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

Fitur **Lihat Data Lilin** digunakan untuk menampilkan semua data lilin yang sudah tersimpan dalam sistem.

Penjelasan:

- **Fungsi lihatLilin** digunakan untuk menampilkan semua data lilin yang ada.

- Jika tidak ada lilin dalam sistem (s.jumlahLilin == 0), program akan menampilkan pesan "Belum ada data lilin".

- Jika ada data lilin, program akan mencetak tabel yang berisi nomor, nama lilin, harga, dan stok lilin untuk setiap lilin yang terdaftar 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

Fitur Perbarui Data Lilin memungkinkan admin untuk memperbarui harga dan stok lilin berdasarkan nomor indeks yang dipilih.

Penjelasan:

- **Fungsi perbaruiLilin** digunakan untuk memperbarui data lilin berdasarkan indeks yang diberikan.

- Jika indeks valid (di antara 0 dan jumlah lilin yang ada), program akan memanggil dua fungsi lainnya:

    - **perbaruiHargaLilin** untuk mengubah harga lilin.

    - **ubahStokLilin** untuk mengubah stok lilin.

- Setelah perubahan selesai, program akan menampilkan pesan "Data berhasil diperbarui!".

- Jika indeks tidak valid, program akan menampilkan pesan "Nomor lilin tidak valid."

Source Code:

```
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

Fitur **Hapus Data Lilin** digunakan untuk menghapus data lilin berdasarkan nomor indeks, lalu menggeser data berikutnya ke atas.

Penjelasan:

- **Fungsi hapusLilin** digunakan untuk menghapus data lilin berdasarkan indeks yang diberikan.

- Jika indeks valid, program akan menggeser semua data setelahnya ke atas untuk mengisi posisi yang kosong.

- Jumlah lilin akan berkurang (s.jumlahLilin--), dan pesan "Data berhasil dihapus!" akan ditampilkan.

- Jika indeks tidak valid, program akan menampilkan pesan "Nomor lilin tidak valid."

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

Fitur **Total Stok Lilin** menggunakan fungsi rekursif untuk menghitung total stok lilin secara keseluruhan.

Penjelasan:

- **Fungsi totalStok** menghitung total stok lilin dalam sistem dengan cara **rekursif**.
- Fungsi ini menambahkan jumlah stok lilin dari setiap indeks dan memanggil dirinya sendiri dengan indeks yang bertambah (index + 1).
- Rekursi berhenti ketika indeks mencapai jumlah lilin yang tersimpan (index == sistem.jumlahLilin), dan total stok akan dihitung.

Source Code:

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

6. Keluar

Fitur **Keluar** digunakan untuk keluar dari sistem dan menampilkan ucapan terima kasih menggunakan fungsi overloading.

Penjelasan:

- **Fungsi ucapan** digunakan untuk memberikan ucapan terima kasih setelah admin memilih keluar dari sistem.

- Fungsi ini memiliki versi **overloaded** yang menerima parameter nama untuk memberikan ucapan khusus yang menyebutkan nama admin yang telah login.

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

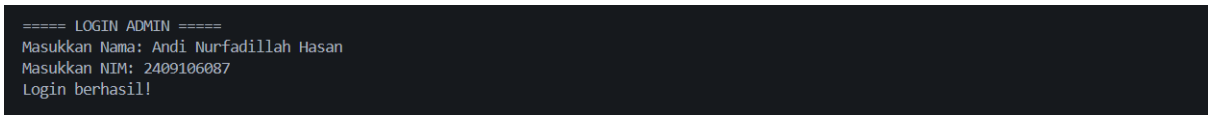
Penjelasan Cara Kerja:

1. Program meminta admin untuk memasukkan **Nama** dan **NIM**.

2. Jika data yang dimasukkan cocok dengan data admin yang telah ditentukan (Andi Nurfadillah Hasan, 2409106087), maka login berhasil dan admin masuk ke menu utama.

3. Jika salah, program akan menampilkan pesan "Login gagal! Sisa percobaan: x", dengan x mengacu pada sisa percobaan yang tersedia.

4. Jika admin gagal login hingga 3 kali berturut-turut, maka program menampilkan pesan "Anda telah gagal login 3 kali. Program berhenti." dan menghentikan proses lebih lanjut.



Gambar 1. Login Admin Berhasil

Gambar 2. Login Admin Gagal

B. Menu Utama

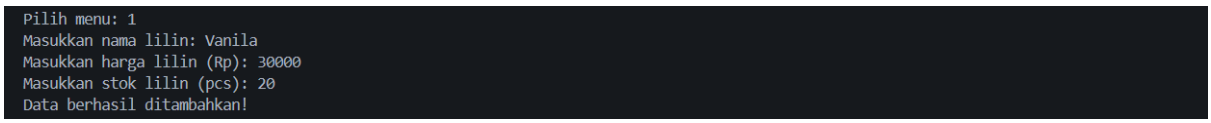
Setelah login berhasil, admin akan dibawa ke **Menu Utama** yang berisi beberapa pilihan untuk mengelola data lilin.

1. Menu Tambah Lilin

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

- Nama lilin
- Harga lilin
- Stok lilin

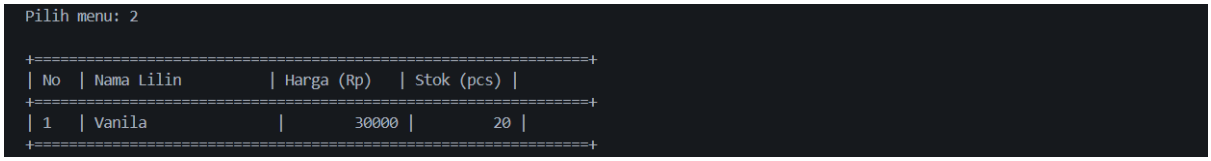
Jika data berhasil ditambahkan, sistem akan memberikan pesan konfirmasi.



Gambar 1.1 Menu Tambah Lilin

2. Menu Lihat Data Lilin

Menu ini akan menampilkan daftar semua lilin yang sudah ditambahkan sebelumnya dalam bentuk tabel. Jika belum ada data, sistem akan memberi informasi bahwa data masih kosong.



Gambar 2.1 Menu Lihat Data Lilin

3. Menu Perbarui Data Lilin

Menu ini digunakan untuk memperbarui harga dan stok lilin yang ada berdasarkan nomor urut lilin yang dipilih.

```
Pilih menu: 3

+============================================================+
| No  | Nama Lilin          | Harga (Rp)   | Stok (pcs) |
+============================================================+
| 1   | Vanila              |        30000 |         20 |
+============================================================+
Masukkan nomor lilin yang ingin diperbarui: 1
Harga baru akan menjadi: 35000
Stok baru akan menjadi: 25
Data berhasil diperbarui!
```

Gambar 3.1 Menu Perbarui Data Lilin

4. Menu Hapus Data Lilin

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

```
Pilih menu: 4

+============================================================+
| No  | Nama Lilin          | Harga (Rp)   | Stok (pcs) |
+============================================================+
| 1   | Vanila              |        35000 |         25 |
+============================================================+
Masukkan nomor lilin yang ingin dihapus: 1
Data berhasil dihapus!
```

Gambar 4.1 Menu Hapus Data Lilin

5. Menu Total Stok Lilin

Menu ini menghitung total stok lilin menggunakan teknik rekursif.

```
Pilih menu: 5
Total seluruh stok lilin: 0 pcs
```

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 29dd9a9] upload code
 2 files changed, 167 insertions(+)
 create mode 100644 post-test/post-test-apl-5/2409106087-AndiNurfadillahHasan-PT-5.cpp
 create mode 100644 post-test/post-test-apl-5/2409106087-AndiNurfadillahHasan-PT-5.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), 679.14 KiB | 4.78 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/
   a0f8c39..29dd9a9  main -> main
```