

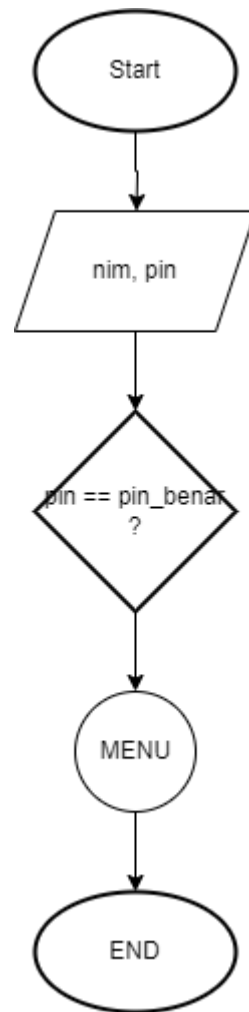
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT

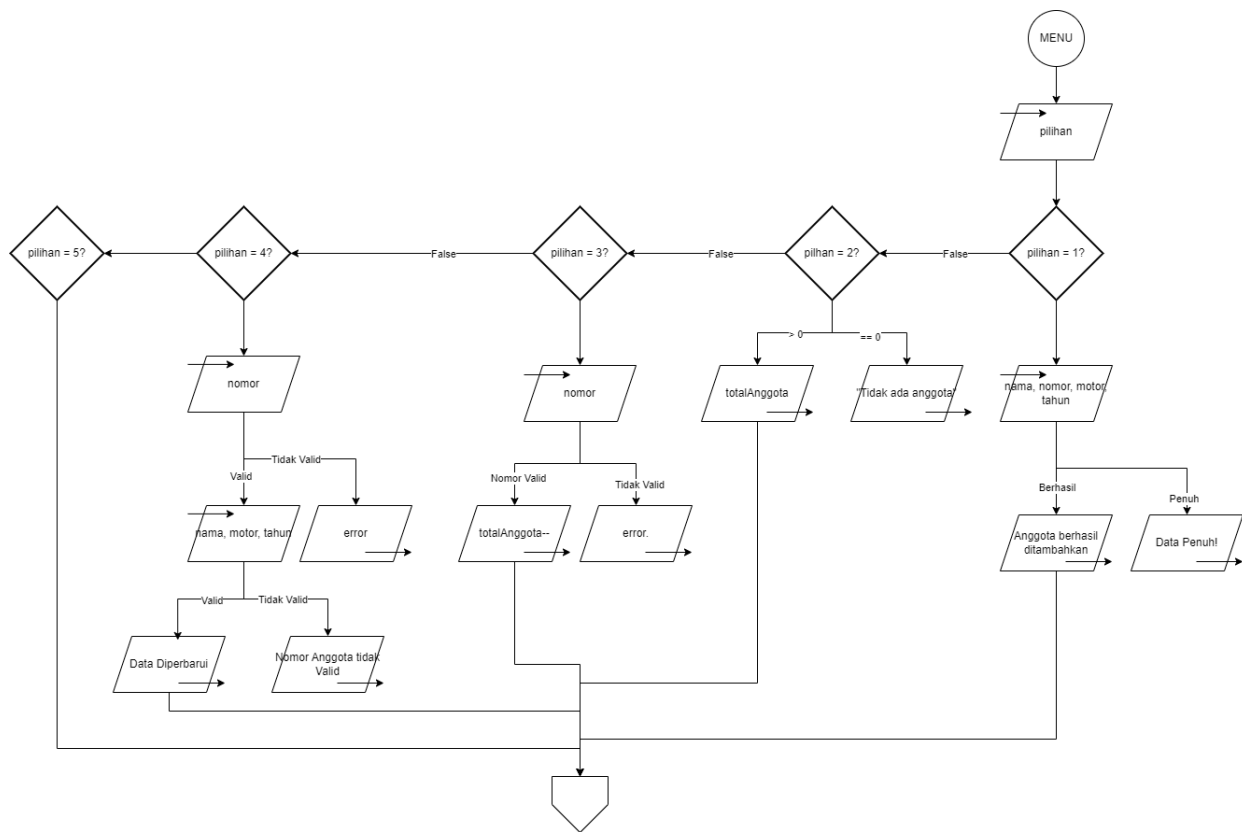


Disusun oleh:
Devon Falen Pasae
(2409106055)
Kelas (B1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart





2. Analisis Program

2.1 Deskripsi Singkat Program

Program "Manajemen Data Anggota Klub Motor" yang berbasis C++ yang digunakan untuk mengelola data anggota dalam sebuah klub motor. Program ini memungkinkan pengguna untuk melakukan operasi CRUD (Create, Read, Update, Delete) terhadap data anggota, yang mencakup informasi Nama, Nomor Anggota, Jenis Motor, dan Tahun Bergabung.

3. Source Code

- **Autentikasi**

Fitur ini dibuat sebagai bentuk autentikasi program dengan cara pengguna input password yang benar dan diberi kesempatan 3 kali. Jika pengguna gagal menginput password benar, maka program akan otomatis berakhir.

```
int autentikasi() {
    string username_benar = "Devon";
    string nim_benar = "6055";
    int max_percobaan = 3;
    bool beri_akses = false;

    for (int percobaan = 1; percobaan <= max_percobaan; percobaan++) {
        string input_username, input_nim;
        cout << "Masukkan Nama: ";
        getline(cin >> std::ws, input_username);
        cout << "Masukkan NIM: ";
        cin >> input_nim;

        if (input_username == username_benar && input_nim == nim_benar) {
            beri_akses = true;
            cout << "== Autentikasi berhasil! ==\n";
            break;
        } else {
            cout << "Nama atau NIM salah! Sisa percobaan: " << (max_percobaan
- percobaan) << endl;
        }
    }

    if (!beri_akses) {
        cout << "== Autentikasi gagal. Program berhenti. ==\n";
        return 0;
    }
    return 1;
}
```

- **Menu Utama**

Fitur ini berfungsi sebagai tampilan menu. Pengguna dapat memilih berbagai menu yang tersedia.

```
// Menu utama
do {
    cout << "\nMENU:" << endl;
    cout << "1. Tambah Anggota" << endl;
    cout << "2. Tampilkan Anggota" << endl;
    cout << "3. Hapus Anggota" << endl;
    cout << "4. Edit Anggota" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilih: ";
    cin >> pilihan;

    // sub-program berdasarkan pilihan
    switch (pilihan) {
        case 1:
            tambahAnggota(anggota, &totalAnggota);
            break;
        case 2:
            tampilkanAnggota(anggota, &totalAnggota);
            break;
        case 3:
            hapusAnggota(anggota, &totalAnggota);
            break;
        case 4:
            editAnggota(anggota, &totalAnggota);
            break;
        case 5:
            cout << "Program selesai." << endl;
            break;
        default:
            cout << "Pilihan tidak valid!" << endl;
    }
} while (pilihan != 5);

return 0;
}
```

- **Tambah Anggota**

Fitur ini berperan sebagai CREATE, yang bertugas untuk menambahkan data baru berupa informasi yang berkaitan dengan anggota klub motor.

```
// Fungsi penambahan anggota
void tambahAnggota(Anggota *data, int *total) {
    if (*total < MAX_ANGGOTA) {
        cout << "Masukkan Nama: ";
        getline(cin >> std::ws, (data + *total)->nama);
        cout << "Masukkan Nomor Anggota: ";
        cin >> (data + *total)->nomor;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        cout << "Masukkan Jenis Motor: ";
        getline(cin, (data + *total)->motor);
        cout << "Masukkan Tahun Bergabung: ";
        cin >> (data + *total)->tahun;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        (*total)++;
        cout << "Anggota berhasil ditambahkan!" << endl;
    } else {
        cout << "Data penuh!" << endl;
    }
}
```

- **Tampilkan Anggota**

Fitur ini berperan sebagai READ, dimana pengguna dapat melihat data yang telah dibuat sebelumnya dan memeriksa data tersebut.

```
// Fungsi menampilkan anggota
void tampilkanAnggota(Anggota *data, int *total) {
    if (*total == 0) {
        cout << "Tidak ada anggota." << endl;
    } else {
        cout << "\n-----" << endl;
        cout << "| No   | Nama                               | No. Anggota | Jenis Motor" << endl;
        cout << "| Tahun |" << endl;
        cout << "-----" << endl;
        for (int i = 0; i < *total; i++) {
            cout << "| " << i + 1 << " | "
                << (data + i)->nama << string(18 - (data + i)-
>nama.length(), ' ') << "| "
                << (data + i)->nomor << "          | "
                << (data + i)->motor << string(17 - (data + i)-
>motor.length(), ' ') << "| "
                << (data + i)->tahun << "      |" << endl;
        }
        cout << "-----" << endl;
    }
}
```

- **Hapus Anggota**

Fitur ini berperan sebagai DELETE, dimana pengguna dapat menghapus data yang sekiranya ingin dihapus dari daftar anggota.

```
// Fungsi menghapus data anggota
void hapusAnggota(Anggota *data, int *total) {
    if (*total == 0) {
        cout << "Tidak ada anggota untuk dihapus." << endl;
    } else {
        int index;
        cout << "Masukkan nomor anggota yang dihapus: ";
        cin >> index;
        if (index > 0 && index <= *total) {
            for (int i = index - 1; i < *total - 1; i++) {
                *(data + i) = *(data + i + 1);
            }
            (*total)--;
            cout << "Anggota dihapus!" << endl;
        } else {
            cout << "Nomor anggota tidak valid!" << endl;
        }
    }
}
```


- **Edit Anggota**

Berfungsi sebagai EDIT, dimana pengguna dapat menyunting data anggota yang sebelumnya sudah terdapat dalam daftar.

```
// Fungsi mengedit data anggota
void editAnggota(Anggota *data, int *total) {
    if (*total == 0) {
        cout << "Tidak ada anggota untuk diedit." << endl;
    } else {
        int index;
        cout << "Masukkan nomor anggota yang diedit: ";
        cin >> index;
        cin.ignore(numeric_limits<streamsize>::max(), '\n');
        if (index > 0 && index <= *total) {
            cout << "Masukkan Nama Baru: ";
            getline(cin, (data + index - 1)->nama);
            cout << "Masukkan Jenis Motor Baru: ";
            getline(cin, (data + index - 1)->motor);
            cout << "Masukkan Tahun Bergabung Baru: ";
            cin >> (data + index - 1)->tahun;
            cin.ignore(numeric_limits<streamsize>::max(), '\n');
            cout << "Data diperbarui!" << endl;
        } else {
            cout << "Nomor anggota tidak valid!" << endl;
        }
    }
}
```

4. Uji Coba dan Hasil Output

1. **Skenario 1** : Jika pengguna salah menginput Nama dan NIM/PIN 3 kali berturut-turut.
2. **Skenario 2** : Jika pengguna memilih menu yang tidak terdapat dalam Menu Utama.
3. **Skenario 3** : Jika pengguna menginput nomor anggota yang tidak terdaftar.

4.1 Hasil Output

```
Masukkan NIM: 0  
Nama atau NIM salah! Sisa percobaan: 1  
Masukkan Nama: a  
Masukkan NIM: s  
Nama atau NIM salah! Sisa percobaan: 0  
== Autentikasi gagal. Program berhenti. ==
```

Gambar 4.1 Skenario 1

```
MENU:  
1. Tambah Anggota  
2. Tampilkan Anggota  
3. Hapus Anggota  
4. Edit Anggota  
5. Keluar  
Pilih: 6  
Pilihan tidak valid!
```

Gambar 4.2 Skenario 2

MENU:

1. Tambah Anggota
2. Tampilkan Anggota
3. Hapus Anggota
4. Edit Anggota
5. Keluar

Pilih: 4

Mengedit anggota...

Masukkan nomor anggota yang diedit: 002

Nomor anggota tidak valid!

Gambar 4.3 Skenario 3

