

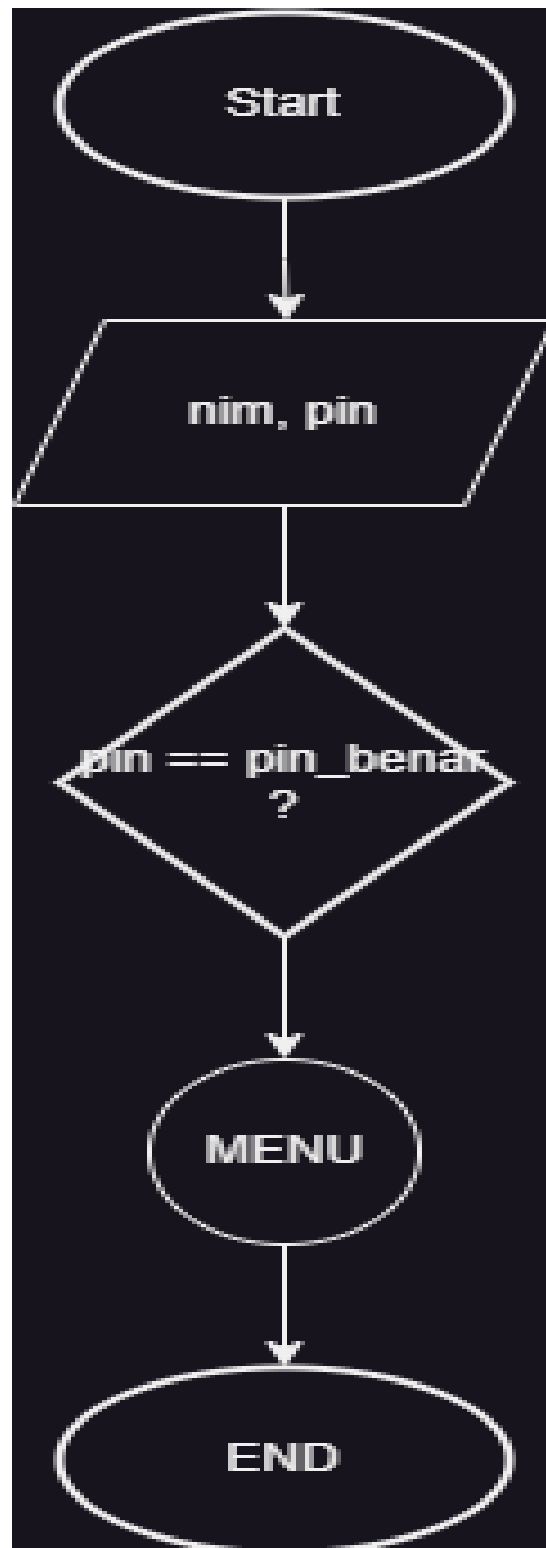
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**

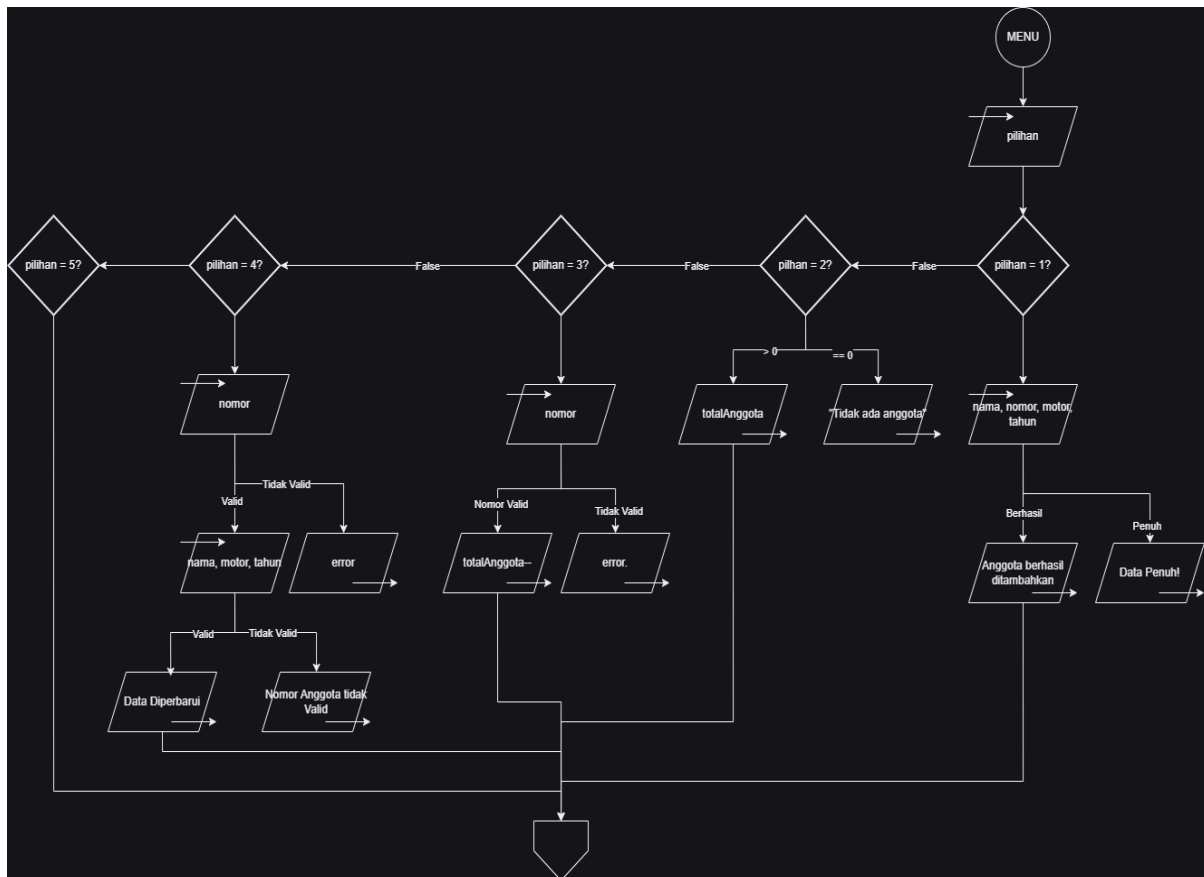


**Disusun oleh:**  
**DEVON FALEN PASAE**  
**2409106055**  
**Kelas B1'24**

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

## 1. Flowchart





## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

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

Program dilengkapi dengan fitur autentikasi menggunakan Nama "**Devon**" dan NIM/PIN "**6055**" sebelum pengguna dapat mengakses menu utama. Jika login gagal sebanyak 3 kali, program akan berhenti.

### 2.2 Penjelasan Alur & Algoritma

Program dimulai dengan meminta pengguna untuk melakukan autentikasi dengan memasukkan **Nama** dan **NIM**. Jika pengguna memasukkan **Nama = "Devon"** dan **NIM = "6055"**, autentikasi berhasil, dan program akan melanjutkan ke menu utama. Namun, jika pengguna salah memasukkan nama atau NIM sebanyak **3 kali**, program akan menampilkan pesan kegagalan dan berhenti.

Setelah berhasil login, pengguna akan disajikan **menu utama** yang berisi beberapa pilihan: **Tambah Anggota, Tampilkan Anggota, Hapus Anggota, Edit Anggota, dan Keluar**. Program akan terus berjalan dalam perulangan hingga pengguna memilih opsi **Keluar**.

Jika pengguna memilih **Tambah Anggota**, program akan meminta input berupa **Nama, Nomor Anggota, Jenis Motor, dan Tahun Bergabung**, kemudian menyimpannya ke dalam array. Jika data anggota sudah mencapai batas maksimal, program akan menampilkan pesan bahwa data penuh.

Ketika pengguna memilih **Tampilkan Anggota**, program akan menampilkan daftar anggota dalam bentuk **tabel** yang rapi, menampilkan **Nomor, Nama, Nomor Anggota, Jenis Motor, dan Tahun Bergabung**. Jika tidak ada anggota dalam daftar, program akan menampilkan pesan bahwa data masih kosong.

Jika pengguna ingin **menghapus anggota**, mereka akan diminta untuk memasukkan nomor anggota yang ingin dihapus. Program akan memeriksa apakah nomor tersebut valid, lalu menggeser data anggota berikutnya ke posisi sebelumnya untuk mengisi celah yang ditinggalkan. Setelah itu, jumlah total anggota akan berkurang. Jika nomor yang dimasukkan tidak valid, program akan menampilkan pesan error.

Ketika pengguna memilih **Edit Anggota**, mereka akan diminta untuk memasukkan nomor anggota yang ingin diedit. Jika nomor anggota valid, program akan meminta **Nama, Jenis Motor, dan Tahun Bergabung** yang baru untuk menggantikan data lama. Jika nomor tidak valid, program akan menampilkan pesan error.

Program akan terus kembali ke menu utama setelah setiap operasi, hingga pengguna memilih opsi **Keluar**, yang akan mengakhiri program dengan menampilkan pesan "**Program selesai**".

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

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

```
int main() {
    if (autentikasi() == 0) {
        return 0;
    }

    int pilihan;
    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;
        switch (pilihan) {
            case 1:
                cout << "Menambah anggota..." << endl;
                tambah();
                break;
            case 2:
                cout << "Menampilkan anggota..." << endl;
                tampilkan();
                break;
            case 3:
                cout << "Menghapus anggota..." << endl;
                hapus();
                break;
            case 4:
                cout << "Mengedit anggota..." << endl;
                edit();
                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.

```
int tambah() {
    if (totalAnggota < MAX_ANGGOTA) {
        cout << "Masukkan Nama: ";
        cin.ignore();
        getline(cin, nama[totalAnggota]);
        cout << "Masukkan Nomor Anggota: ";
        cin >> nomor[totalAnggota];
        cout << "Masukkan Jenis Motor: ";
        cin.ignore();
        getline(cin, motor[totalAnggota]);
        cout << "Masukkan Tahun Bergabung: ";
        cin >> tahun[totalAnggota];
        totalAnggota++;
        cout << "Anggota berhasil ditambahkan!" << endl;
    } else {
        cout << "Data penuh!" << endl;
    }
    return 0;
}
```

- Tampilkan Anggota

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

```
int tampilkan() {
    if (totalAnggota == 0) {
        cout << "Tidak ada anggota." << endl;
    } else {
        cout <<
        "\n-----" <<
        endl;
        cout << "| No | Nama | No. Anggota | Jenis Motor |" <<
        "Tahun |" << endl;
        cout <<
        "-----" <<
        endl;
        for (int i = 0; i < totalAnggota; i++) {
            cout << "| " << i + 1 << " | "
                << nama[i] << string(18 - nama[i].length(), ' ') << "| "
                << nomor[i] << " | "
                << motor[i] << string(17 - motor[i].length(), ' ') << "| "
                << tahun[i] << " |" << endl;
        }
        cout <<
        "-----" <<
        endl;
    }
}
```

```
}  
    return 0;  
}
```

- Hapus Anggota

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

```
int hapus() {  
    if (totalAnggota == 0) {  
        cout << "Tidak ada anggota untuk dihapus." << endl;  
    } else {  
        int index;  
        cout << "Masukkan nomor anggota yang dihapus: ";  
        cin >> index;  
        if (index > 0 && index <= totalAnggota) {  
            for (int i = index - 1; i < totalAnggota - 1; i++) {  
                nama[i] = nama[i + 1];  
                nomor[i] = nomor[i + 1];  
                motor[i] = motor[i + 1];  
                tahun[i] = tahun[i + 1];  
            }  
            totalAnggota--;  
            cout << "Anggota dihapus!" << endl;  
        } else {  
            cout << "Nomor anggota tidak valid!" << endl;  
        }  
    }  
    return 0;  
}
```



- Edit Anggota

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

```
int edit() {
    if (totalAnggota == 0) {
        cout << "Tidak ada anggota untuk diedit." << endl;
    } else {
        int index;
        cout << "Masukkan nomor anggota yang diedit: ";
        cin >> index;
        if (index > 0 && index <= totalAnggota) {
            cout << "Masukkan Nama Baru: ";
            cin.ignore();
            getline(cin, nama[index - 1]);
            cout << "Masukkan Jenis Motor Baru: ";
            getline(cin, motor[index - 1]);
            cout << "Masukkan Tahun Bergabung Baru: ";
            cin >> tahun[index - 1];
            cout << "Data diperbarui!" << endl;
        } else {
            cout << "Nomor anggota tidak valid!" << endl;
        }
    }
    return 0;
}
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

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.

## 4.2 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

## 5. Git

### 5.1 Git Add

```
PS D:\dump tugas\project 0\devon\praktikum-apl\praktikum-apl\post-test\post-test-apl-2> git add .
```

Git add berfungsi untuk menambahkan file/revisi ke dalam staging area sebelum di-commit.

### 5.2 Git Commit

```
PS D:\dump tugas\project 0\devon\praktikum-apl\praktikum-apl\post-test\post-test-apl-2> git commit -m "Finish Post Test 2"
[main a7e56bb] Finish Post Test 2
3 files changed, 166 insertions(+)
create mode 100644 praktikum-apl/post-test/post-test-apl-2/2409106055-DevonFalenPasae-PT-2.cpp
create mode 100644 praktikum-apl/post-test/post-test-apl-2/2409106055-DevonFalenPasae-PT-2.exe
create mode 100644 praktikum-apl/post-test/post-test-apl-2/2409106055-DevonFalenPasae-PT-2.pdf
```

Git commit berfungsi untuk menyimpan perubahan yang sudah ditambahkan ke staging area ke dalam repository lokal Git.

### 5.3 Git push -u main origin

```
PS D:\dump tugas\project 0\devon\praktikum-apl\praktikum-apl\post-test\post-test-apl-2> git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 20 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 1.07 MiB | 719.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/XZNVWS/praktikum-apl.git
cc1a26e..a7e56bb main -> main
branch 'main' set up to track 'origin/main'.
```

Git push origin main berfungsi untuk mengunggah (push) perubahan dari branch main di repository lokal ke repository remote yang telah dikonfigurasi sebelumnya.