

LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT

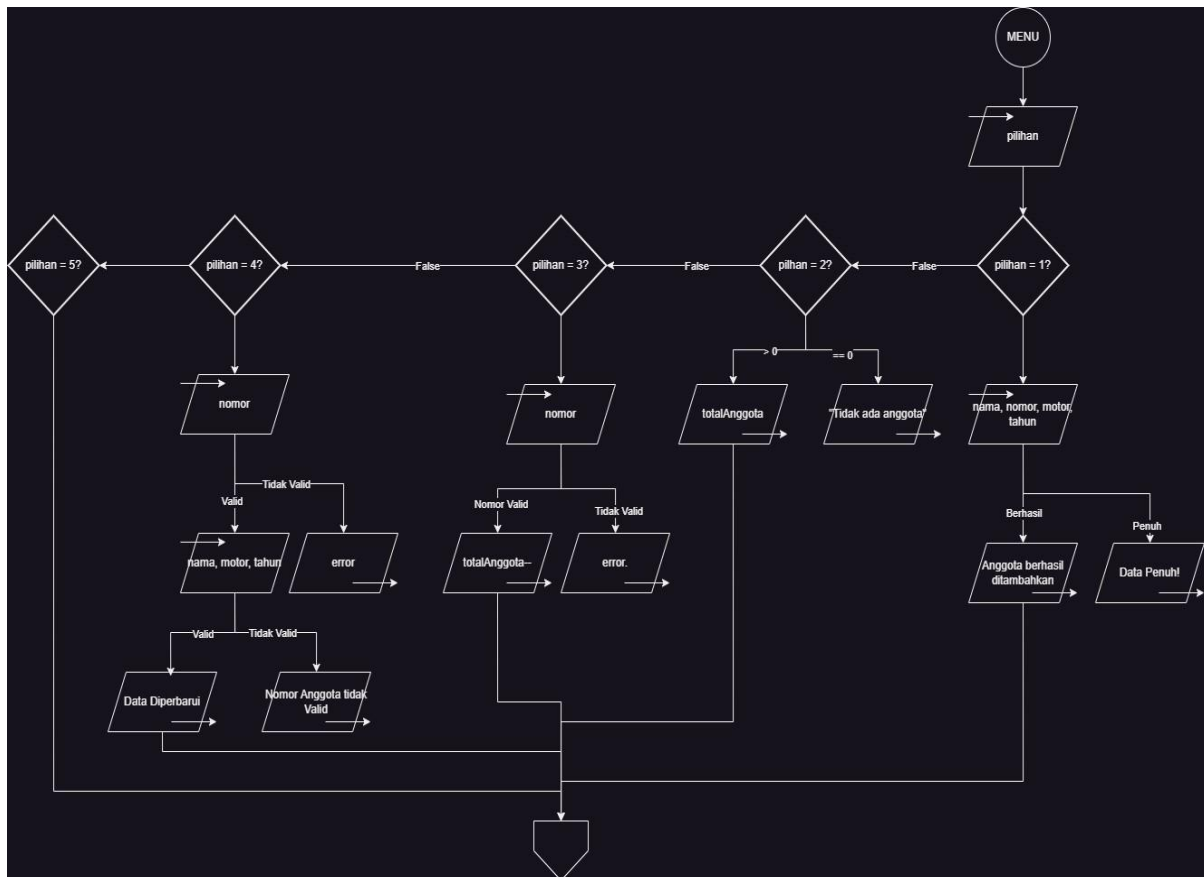


Disusun oleh:
DEVON FALEN PASAE
2409106055
Kelas B1'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart





2. Analisis Program

2.1 Deskripsi Singkat Program

Program "**Manajemen Data Anggota Klub Motor**" adalah 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**. Untuk memastikan akses yang aman, program dilengkapi dengan fitur autentikasi menggunakan Nama "**Devon**" dan NIM/PIN "**6055**" sebelum pengguna dapat mengakses menu utama. Jika pengguna gagal melakukan login sebanyak **tiga kali**, program akan otomatis berhenti.

Dalam pengelolaan data, program ini menggunakan **struct** untuk mengelompokkan variabel yang berkaitan dalam satu kesatuan, sehingga lebih terorganisir. Data anggota disimpan dalam **array of struct**, di mana setiap elemen struct menyimpan informasi anggota seperti **nama, nomor anggota, jenis motor, dan tahun bergabung**. Dengan pendekatan ini, program menjadi lebih efisien dalam menyimpan dan mengelola data dibandingkan menggunakan beberapa array terpisah.

Setelah berhasil masuk, pengguna akan disajikan **menu utama** yang berisi beberapa opsi.

Jika pengguna memilih **Tambah Anggota**, program akan meminta input berupa **Nama, Nomor Anggota, Jenis Motor, dan Tahun Bergabung**, lalu menyimpannya ke dalam array. Jika jumlah anggota telah mencapai batas maksimal, program akan menampilkan peringatan bahwa data sudah penuh. Jika pengguna memilih **Tampilkan Anggota**, program akan menampilkan daftar anggota dalam bentuk tabel yang rapi. Jika belum ada anggota yang terdaftar, program akan memberi tahu bahwa data masih kosong.

Selain itu, program juga mendukung fitur **Hapus Anggota**, di mana pengguna dapat menghapus anggota berdasarkan **nomor anggota** yang dimasukkan. Jika nomor tersebut valid, program akan menggeser data berikutnya untuk mengisi celah yang ditinggalkan dan mengurangi total jumlah anggota. Jika nomor yang dimasukkan tidak valid, program akan menampilkan pesan error. Begitu pula dengan fitur **Edit Anggota**, pengguna dapat mengubah informasi anggota dengan memasukkan **nomor anggota yang valid**, lalu mengganti data yang lama dengan yang baru.

Program akan terus berulang, mengembalikan pengguna ke menu utama setelah setiap operasi, hingga pengguna memilih opsi **Keluar**, yang akan mengakhiri program dengan menampilkan pesan "**Program selesai.**" Dengan penggunaan **struct** dan **array of struct**, program menjadi lebih terstruktur, mudah dipahami, serta efisien dalam pengelolaan data anggota klub motor.

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,
anggota[totalAnggota].nama);        cout <<
"Masukkan Nomor Anggota: ";        cin >>
anggota[totalAnggota].nomor;        cout <<
"Masukkan Jenis Motor: ";        cin.ignore();
        getline(cin, anggota[totalAnggota].motor);
cout << "Masukkan Tahun Bergabung: ";        cin >>
anggota[totalAnggota].tahun;        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 << "   | "
        << anggota[i].nama << string(18 - anggota[i].nama.length(), ' ')
    << " | "
        << anggota[i].nomor << "           | "
        << anggota[i].motor << string(17 - anggota[i].motor.length(), '
    ') << " | "
        << anggota[i].tahun << "   |" << 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++) {                anggota[i] =
anggota[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, anggota[index -
1].nama);
            cout << "Masukkan Jenis Motor
Baru: ";
            getline(cin, anggota[index -
1].motor);
            cout << "Masukkan Tahun
Bergabung Baru: ";
            cin >> anggota[index -
1].tahun;
            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

