

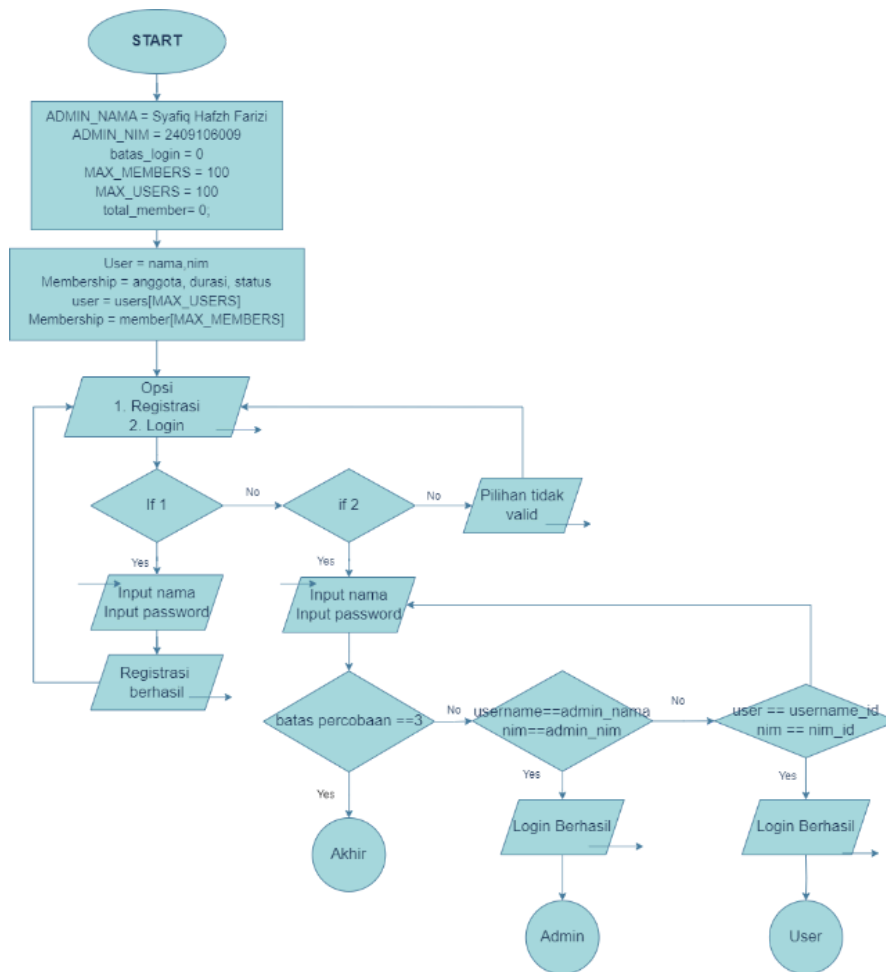
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



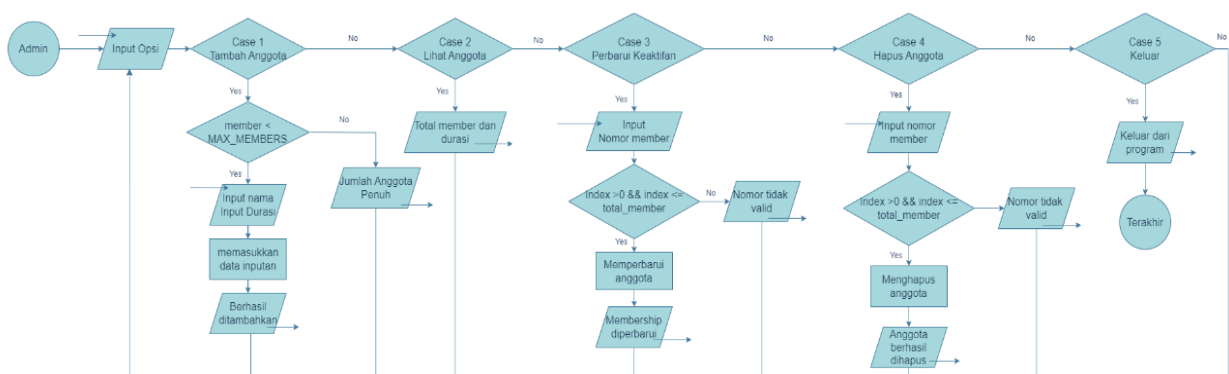
Disusun oleh:
Syafiq Hafizh Farizi (2409106009)
Kelas (A1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

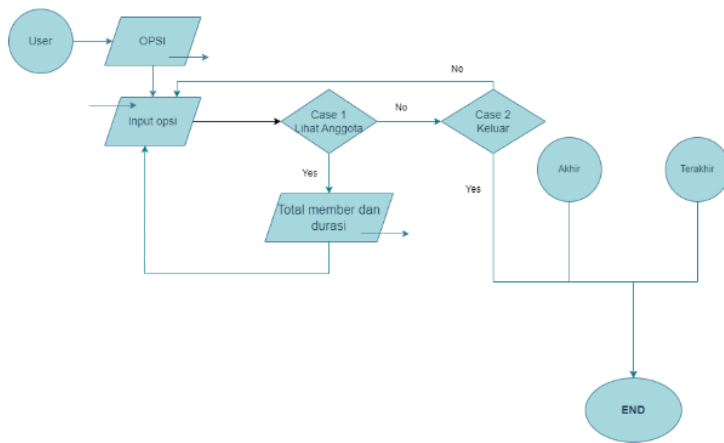
1. Flowchart



Gambar 1.1 Login



Gambar 1.2 Admin



Gambar 1.3 User

2. Analisis Program

Program manajemen sistem member gym ini dirancang untuk memudahkan pengelola gym dalam mengelola data anggota, memantau status keaktifan, serta melakukan berbagai operasi manajemen anggota secara efisien. Dengan sistem ini, pengelola gym dapat dengan mudah mencatat, mengupdate, dan menghapus data anggota, serta memantau aktivitas keanggotaan. dengan fitur yang ada sebagai berikut:

Fitur Admin

1. Create (Tambah Anggota): Menambahkan member baru dengan menyimpan nama, NIM, durasi membership, dan status aktif
2. Read (Lihat Anggota): Menampilkan daftar member dalam format tabel yang rapi dengan kolom No, Nama, NIM, Durasi, dan Status
3. Update (Perbarui Keaktifan): Memungkinkan admin mengubah durasi membership dan status aktif/tidak aktif
4. Delete (Hapus Anggota): Menghapus member dari sistem berdasarkan nomor urut

Fitur Member Biasa

Hanya dapat melihat daftar anggota (fitur Read) tanpa bisa melakukan modifikasi data dan memiliki tombol keluar

3. Source Code

A. Prosedur Register dan Fungsi Rekursif Login

Fitur Prosedur register ini digunakan untuk user baru yang akan membuat akun untuk masuk ke manajemen membership gym sementara Fitur Login yang menggunakan fungsi rekursif ini akan meminta user memilih opsi antara registrasi (Menggunakan parameter address of) dan login dan akan memanggil dirinya lagi jika login gagal, jika akan melakukan login ke akun admin maka inputan yang harus diisi saat login yaitu dengan “Nama= Syafiq Hafizh Farizi, Nim= 2409106009”

Source Code:

```
void registrasi(User users[], int &totalUsers) {
    if (totalUsers >= MAX_USERS) {
        cout << "Jumlah pengguna penuh!" << endl;
        return;
    }

    cout << "==== Registrasi =====< endl;
    cout << "Masukkan Nama: ";
    getline(cin, users[totalUsers].nama);
    cout << "Masukkan Password (NIM): ";
    getline(cin, users[totalUsers].nim);

    // Address-of
    totalUsers++;
    cout << "Registrasi berhasil!" << endl;
}

bool login(User users[], int totalUsers, string &namaLogin, string &nimLogin,
int batas = 0) {
    if (batas >= 3) {
        cout << "Terlalu banyak percobaan gagal.\n";
        return false;
    }

    cout << "==== Login =====< endl;
    cout << "Masukkan Nama: ";
    getline(cin, namaLogin);
    cout << "Masukkan NIM: ";
    getline(cin, nimLogin);

    if (namaLogin == ADMIN_NAMA && nimLogin == ADMIN_NIM)
        return true;

    for (int i = 0; i < totalUsers; i++) {
        if (users[i].nama == namaLogin && users[i].nim == nimLogin)
            return true;
    }
}
```

```

    }

    cout << "Login gagal! Coba lagi." << endl;
    return login(users, totalUsers, namaLogin, nimLogin, batas + 1);
}

```

B. Opsi CRUD

Memiliki 5 pilihan untuk crud admin

Source Code:

```

while (aktif) {
    cout << "==== Membership Gym =====" << endl;
    cout << "1. Tambah Anggota" << endl;
    cout << "2. Lihat Anggota" << endl;
    cout << "3. Perbarui Keaktifan" << endl;
    cout << "4. Hapus Anggota" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilihan: ";
    cin >> pilihan;
    cin.ignore();

    switch (pilihan) {
        case 1: tambahAnggota(members, &totalMembers); break;
        case 2: tampilkanAnggota(members, totalMembers); break;
        case 3: perbaruiDurasi(members, totalMembers); break;
        case 4: hapusAnggota(members, totalMembers); break;
        case 5: aktif = false; break;
        default: cout << "Pilihan tidak valid!" << endl;
    }
}
}

```

C. Opsi Create

Menggunakan prosedur untuk bagian create (Menambah Anggota) dan terdapat fungsi dengan parameter dereference

Source Code:

```

void tambahAnggota(Membership* members, int* totalMembers) {
    if (*totalMembers >= MAX_MEMBERS) {
        cout << "Jumlah anggota penuh!" << endl;
    }
}

```

```

        return;
    }

    cout << "Masukkan Nama Anggota: ";
    getline(cin, members[*totalMembers].anggota.nama);
    cout << "Masukkan NIM Anggota: ";
    getline(cin, members[*totalMembers].anggota.nim);
    cout << "Masukkan Durasi Membership (1/2/3): ";
    cin >> members[*totalMembers].durasi;
    members[*totalMembers].status = true;
    cin.ignore();

    (*totalMembers)++; // dereference pointer untuk menambah jumlah
    cout << "Anggota berhasil ditambahkan!" << endl;
}

```

D. Opsi Read

Melihat anggota yang aktif pada GYM dengan menggunakan prosedur di bagian (Tampilkan anggota)

Source Code:

```

void tampilkanAnggota(const Membership members[], int totalMembers) {
    cout << "===== " << endl;
    cout << "| No |          Nama          |      NIM      | Durasi | Status |"
    << endl;
    cout << "===== " << endl;
    for (int i = 0; i < totalMembers; i++) {
        cout << "| " << setw(2) << i + 1 << " | " << setw(19) << left <<
        members[i].anggota.nama << " | "
        << setw(13) << members[i].anggota.nim << " | "
        << setw(6) << members[i].durasi << " | "
        << setw(8) << (members[i].status ? "Aktif" : "Tidak Aktif") << " | "
    << endl;
    }
    cout << "===== " << endl;
}

```

E. Opsi Update

Menggunakan prosedur perbarui data untuk Memperbarui member GYM yang masih aktif jika 0 maka otomatis tidak aktif

Source Code:

```

void perbaruiDurasi(Membership* members, int totalMembers) {
    tampilkanAnggota(members, totalMembers);
    int index;
    cout << "Masukkan nomor anggota yang ingin diperbarui: ";
    cin >> index;
    cin.ignore();

    if (index > 0 && index <= totalMembers) {
        Membership* anggotaPtr = &members[index - 1]; // pointer ke struct
        cout << "Perbarui Durasi (0 untuk nonaktif): ";
        cin >> anggotaPtr->durasi; // dereference struct
        anggotaPtr->status = (anggotaPtr->durasi > 0);
        cin.ignore();
        cout << "Durasi diperbarui!" << endl;
    } else {
        cout << "Nomor tidak valid!" << endl;
    }
}

```

F. Opsi Delete

Menghapus anggota yang sudah tidak membership dengan menggunakan prosedur hapusAnggota

Source Code:

```

void hapusAnggota(Membership members[], int &totalMembers) {
    tampilkanAnggota(members, totalMembers);
    int index;
    cout << "Masukkan nomor anggota yang ingin dihapus: ";
    cin >> index;
    cin.ignore();
    if (index > 0 && index <= totalMembers) {
        for (int i = index - 1; i < totalMembers - 1; i++) {
            members[i] = members[i + 1];
        }
        totalMembers--;
        cout << "Anggota dihapus." << endl;
    } else {
        cout << "Nomor tidak valid!" << endl;
    }
}

```


4. Uji Coba dan Hasil Output

```
===== MENU =====  
1. Registrasi  
2. Login  
Pilihan: 1  
===== Registrasi =====  
Masukkan Nama: tsssssssss  
Masukkan Password (NIM): 22222  
Registrasi berhasil!  
===== MENU =====  
1. Registrasi  
2. Login  
Pilihan: 2  
===== Login =====  
Masukkan Nama: tsssssssss  
Masukkan NIM: 22222  
===== Menu Pengguna =====  
1. Lihat Anggota  
2. Keluar  
Pilihan: 
```

Gambar 4.1 Registrasi

```

===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 1
Masukkan Nama Anggota: oyoy
Masukkan NIM Anggota: 20
Masukkan Durasi Membership (1/2/3): 2
Anggota berhasil ditambahkan!
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 2
=====
| No | Nama | NIM | Durasi | Status |
=====
| 1 | oyoy | 20 | 2 | Aktif |
=====
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: █

```

Gambar 4.2 Login & Tambah & Lihat

```

===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 3
=====
| No | Nama | NIM | Durasi | Status |
=====
| 1 | oyoy | 20 | 2 | Aktif |
=====
Masukkan nomor anggota yang ingin diperbarui: 1
Perbarui Durasi (0 untuk nonaktif): 0
Durasi diperbarui!
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 2
=====
| No | Nama | NIM | Durasi | Status |
=====
| 1 | oyoy | 20 | 0 | Tidak Aktif |
=====
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: █

```

Gambar 4.3 Update

```

===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 4
=====
| No | Nama | NIM | Durasi | Status |
=====
| 1 | oyoy | 20 | 0 | Tidak Aktif |
=====
Masukkan nomor anggota yang ingin dihapus: 1
Anggota dihapus.
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 2
=====
| No | Nama | NIM | Durasi | Status |
=====
===== Membership Gym =====
1. Tambah Anggota
2. Lihat Anggota
3. Perbarui Keaktifan
4. Hapus Anggota
5. Keluar
Pilihan: 

```

Gambar 4.4 Delete

```
PS D:\praktikum-apl> cd "d:\praktikum-apl"
===== MENU =====
1. Registrasi
2. Login
Pilihan: 1
===== Registrasi =====
Masukkan Nama: Tes
Masukkan Password (NIM): 12
Registrasi berhasil!
===== MENU =====
1. Registrasi
2. Login
Pilihan: 2
===== Login =====
Masukkan Nama: Tes
Masukkan NIM: 12
===== Menu Pengguna =====
1. Lihat Anggota
2. Keluar
Pilihan: █
```

Gambar 4.5 User

5. Langkah-Langkah Git pada VSCode

1. Git add & commit

melakukan git add untuk menambahkan file yang akan kita commit, dan melakukan git commit untuk memasukkan ke dalam repo

```
PS D:\praktikum-apl> git add .
PS D:\praktikum-apl> git commit -m "up posttest 5"
[main e1d953e] up posttest 5
2 files changed, 215 insertions(+)
create mode 100644 post-test/post-test-5/2409106009-SyafiqHafizhFarizi-PT-5.cpp
create mode 100644 post-test/post-test-5/2409106009-SyafiqHafizhFarizi-PT-5.exe
```

Gambar 5.1 add & commit

2. Git push

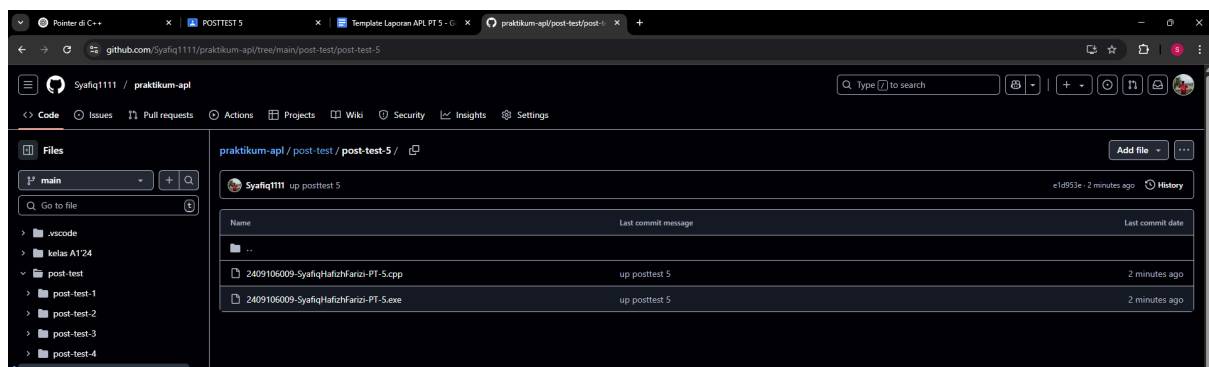
Mengupload semua yang ada perubahan direpository lokal kita

```
PS D:\praktikum-apl> git push origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 679.33 KiB | 7.38 MiB/s, done.
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/Syafiq1111/praktikum-apl.git
9e2a7a3..e1d953e main -> main
PS D:\praktikum-apl>
```

Gambar 5.2 Git Push

3. Reload Tab

reload tab github pada browser yang digunakan



Gambar 5.3 Reload