

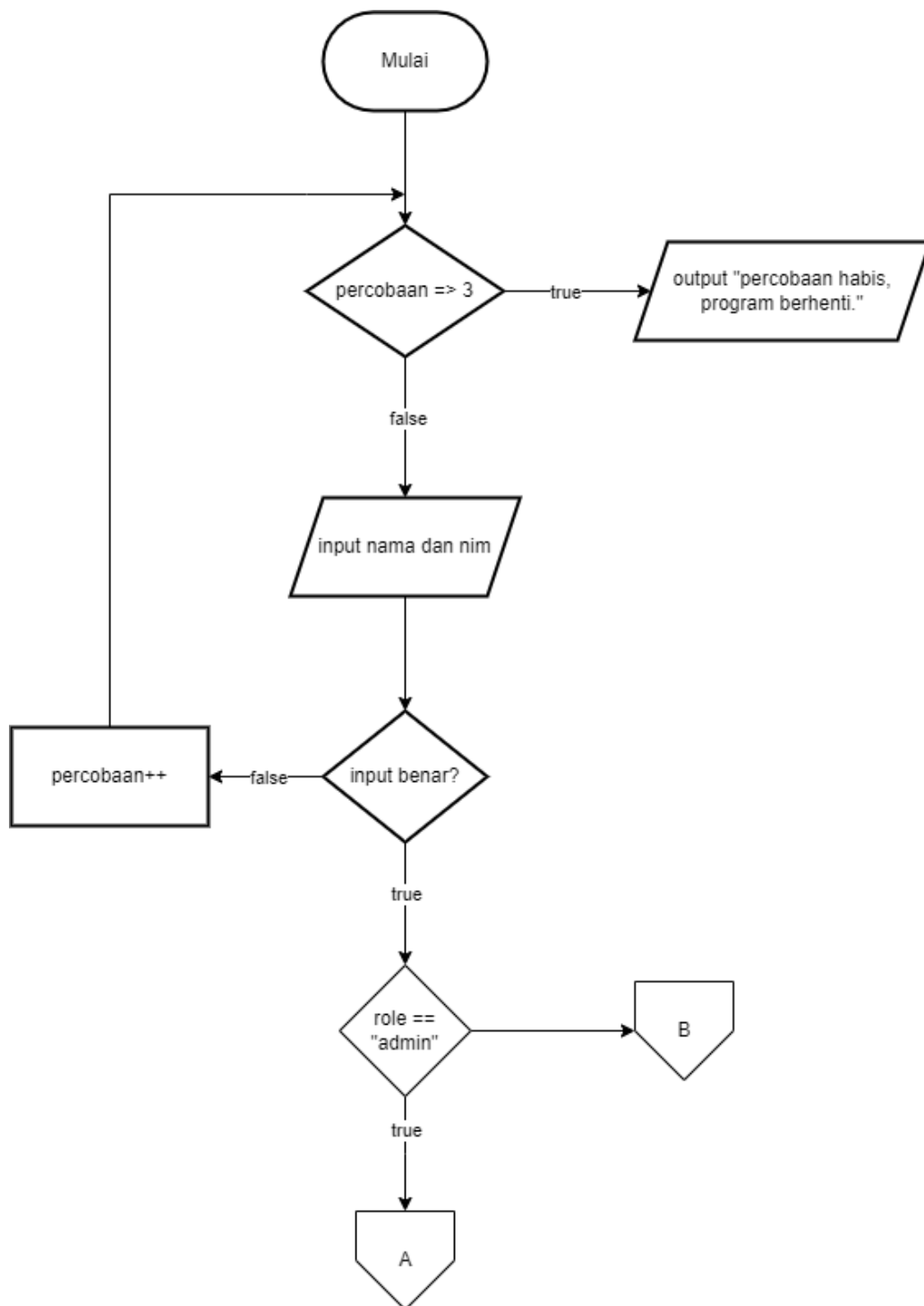
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**



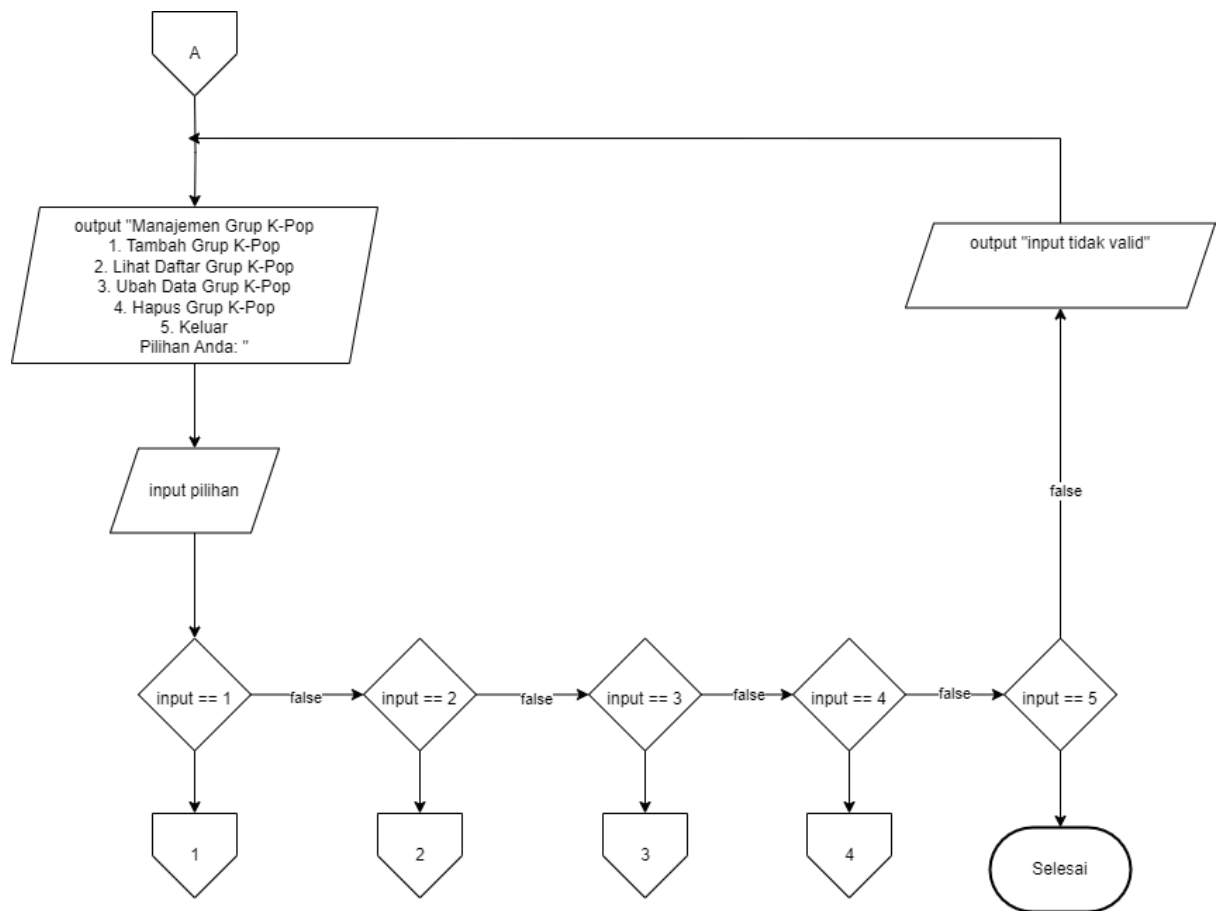
**Disusun oleh:**  
**Razib Ramadhan (2409106076)**  
**Kelas (B2 '24)**

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

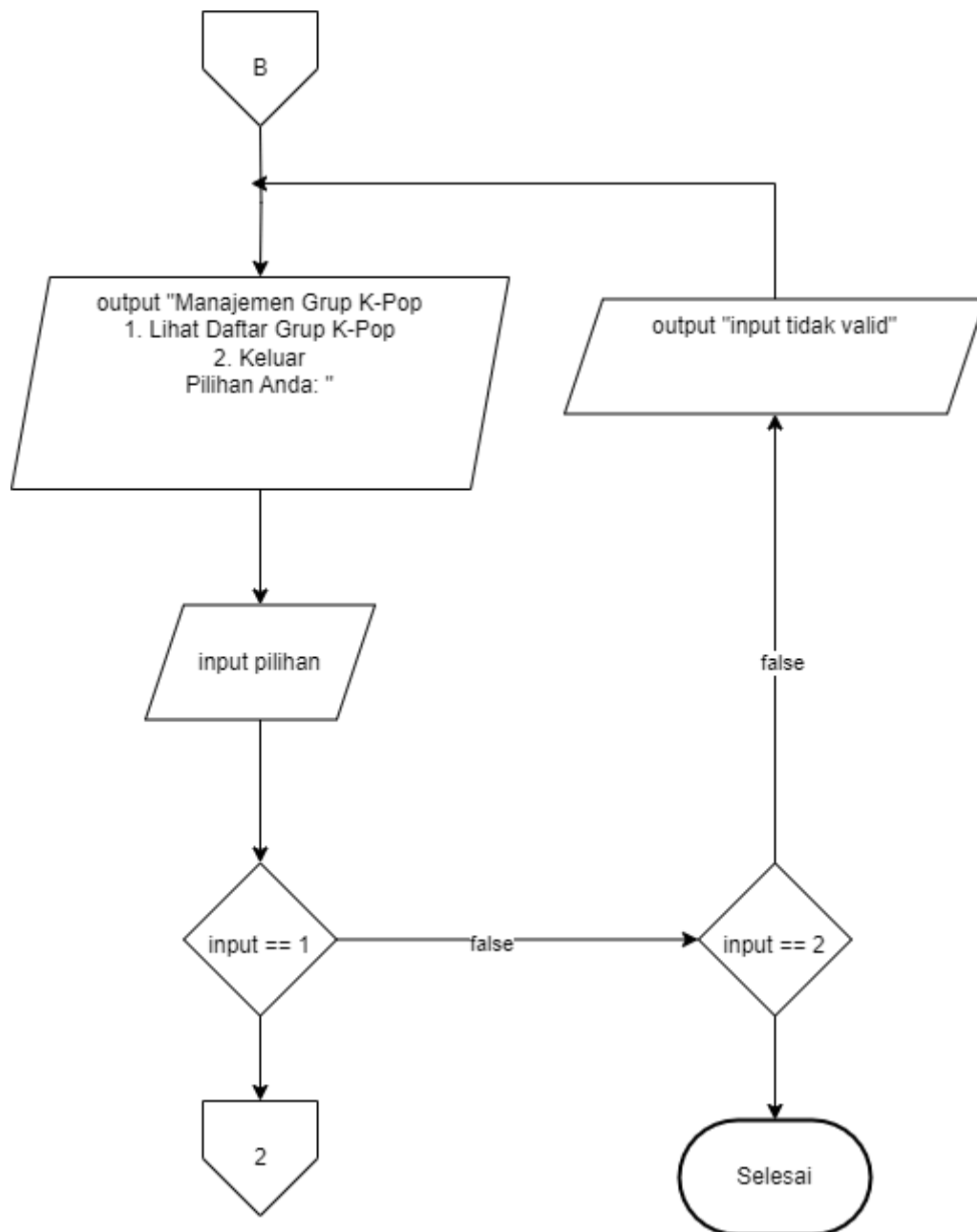
## 1. Flowchart



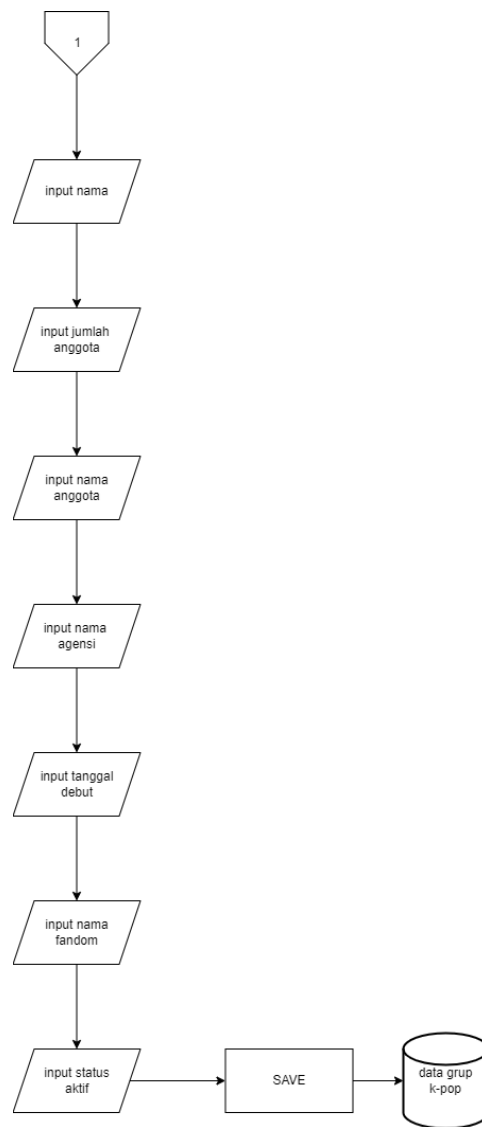
Gambar 1.1 *Main Flow* Bagian 1



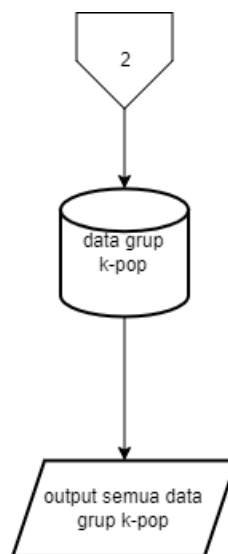
Gambar 1.2 *Main Flow* Bagian Admin



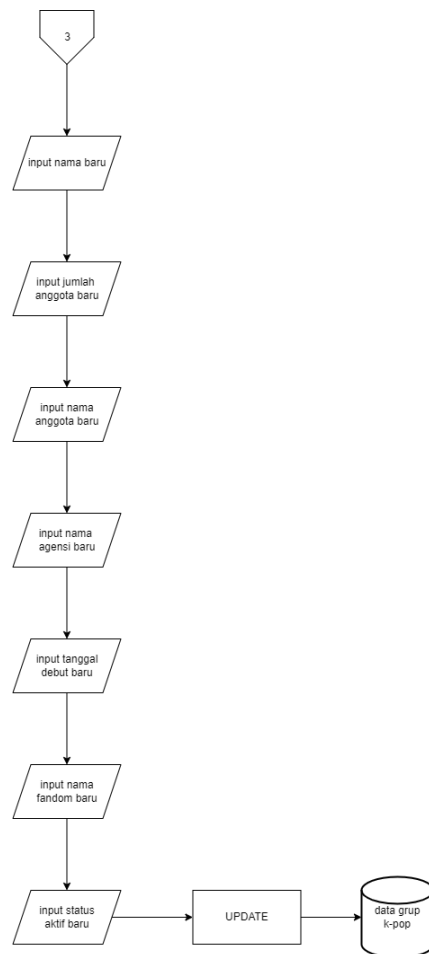
Gambar 1.3 *Main Flow* Bagian User



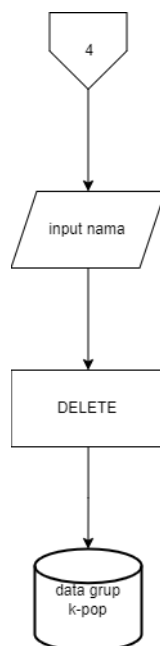
Gambar 1.4 *Main Flow* Bagian 2



Gambar 1.5 *Main Flow* Bagian 3



Gambar 1.6 *Main Flow* Bagian 4



Gambar 1.7 *Main Flow* Bagian 5

## 2. Analisis Program

Program ini adalah aplikasi manajemen data grup K-Pop yang dirancang untuk memudahkan pengguna dalam mengelola informasi terkait grup idola favorit mereka. Fitur utama program ini meliputi penambahan grup baru dengan detail lengkap seperti nama grup, jumlah dan nama anggota, agensi, tanggal debut, nama fandom, dan status aktif. Pengguna juga dapat melihat daftar grup yang telah ditambahkan, mengubah informasi grup yang sudah ada, atau menghapus grup dari daftar. Selain itu, program ini dilengkapi dengan fitur login sederhana yang membatasi percobaan akses sebanyak tiga kali. Validasi input diterapkan untuk memastikan data yang dimasukkan pengguna valid, seperti jumlah anggota yang tidak melebihi batas maksimum.

## 3. Source Code

### A. Halaman Awal

Pada bagian awal program, pengguna dapat memilih menu untuk login, regis, atau keluar program.

**Source Code:**

```
while (loginSuccess == false) {  
  
    system("cls");  
    cout << "=== Pilih menu ===" << endl;  
    cout << "1. Registrasi Pengguna" << endl;  
    cout << "2. Login Pengguna" << endl;  
    cout << "3. Keluar" << endl;  
    cout << "Pilihan: ";  
    int pilihan;  
    cin >> pilihan;  
    cin.ignore();  
}
```

### B. Menu Registrasi

Pada bagian ini, pengguna dapat mendaftarkan akun baru dengan memasukkan nama dan password.

## Source Code:

```
switch (pilihan) {
    case 1:
        if (jumlahPegguna >= MAX) {
            system("cls");
            cout << "Pegguna sudah penuh. Registrasi gagal." << endl;
            break;
        }
        system("cls");
        cout << "=== Registrasi Pegguna ===" << endl;
        cout << "Nama: ";
        getline(cin, nama);
        cout << "Password: ";
        getline(cin, password);
        peran = "user";

        if (jumlahPegguna < MAX) {
            for (int i = 0; i < jumlahPegguna; i++) {
                if (daftarPegguna[i].nama == nama) {
                    statusRegis = true;
                    cout << "Nama pegguna sudah terdaftar. Registrasi
gagal." << endl;

                    cout << "Tekan enter untuk melanjutkan..." << endl;
                    cin.ignore();
                    break;
                }
            }
            if (statusRegis == false) {
                daftarPegguna[jumlahPegguna] = {nama, password,
peran};

                jumlahPegguna++;
                cout << "Registrasi berhasil!" << endl;
                cout << "Tekan enter untuk melanjutkan..." << endl;
                cin.ignore();
            }
            break;
        }
}
```

## C. Menu Login

Pada bagian ini, pengguna dapat login untuk masuk ke menu utama. Data yang harus diberikan dalam proses login adalah nama lengkap dan password. Maksimal percobaan hingga program berhenti adalah sebanyak tiga kali.



## Source Code:

```
while (percobaan < 3) {
    system("cls");
    cout << "=== Login Pengguna ===" << endl;
    cout << "Masukkan Nama: ";
    getline(cin, nama);
    cout << "Masukkan Password: ";
    getline(cin, password);

    for (int i = 0; i < jumlahPengguna; i++) {
        if (daftarPengguna[i].nama == nama &&
daftarPengguna[i].password == password) {
            level = daftarPengguna[i].peran;
            loginSuccess = true;
            break;
        }
    }

    if (loginSuccess) {
        cout << "Login berhasil!" << endl;
        cout << "Tekan enter untuk melanjutkan..." << endl;
        cin.ignore();
        system("cls");
        break;
    } else {
        percobaan++;
        cout << "Nama atau Password salah. Percobaan ke-" <<
percobaan << endl;
        cout << "Tekan enter untuk melanjutkan..." << endl;
        cin.ignore();
    }
}

if (percobaan == 3) {
    cout << "Anda telah gagal 3 kali. Program berakhir." << endl;
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();
    system("cls");
    return 0;
}

break;
```

## D. Menu Admin

Bagian berikut adalah kode dari halaman admin.

### Source Code:

```
system("cls");
cout << "\n=== Manajemen Grup K-Pop ===" << endl;
cout << "1. Tambah Grup K-Pop" << endl;
cout << "2. Lihat Daftar Grup K-Pop" << endl;
cout << "3. Ubah Data Grup K-Pop" << endl;
cout << "4. Hapus Grup K-Pop" << endl;
cout << "5. Keluar" << endl;
cout << "Pilihan Anda: ";
cin >> pilihan;
cin.ignore();
```

## E. Menu User

Bagian berikut adalah kode dari halaman user.

### Source Code:

```
int pilihan;
system("cls");
cout << "=== Menu User ===" << endl;
cout << "1. Lihat Daftar Grup K-Pop" << endl;
cout << "2. Keluar" << endl;
cout << "Pilihan: ";
cin >> pilihan;
```

## F. Menu Tambah Grup K-Pop

Bagian berikut digunakan untuk menampilkan menu tambah grup K-Pop dan meminta pengguna untuk memasukkan data baru.

### Source Code:

```
system("cls");
if (jumlahGrup < MAX) {
    GrupKPop grupBaru;
    cout << "\n--- Tambah Grup K-Pop ---" << endl;
    cout << "Nama Grup: ";
    getline(cin, grupBaru.nama);
}
```

```

bool namaSudahAda = false;
for (int i = 0; i < jumlahGrup; ++i) {
    if (daftarGrup[i].nama == grupBaru.nama) {
        namaSudahAda = true;
        break;
    }
}

if (namaSudahAda) {
    cout << "Grup dengan nama tersebut sudah ada." << endl;
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();
    break;
}

int anggotaBaru;
cout << "Jumlah Anggota: ";
cin >> anggotaBaru;

if (cin.fail() || anggotaBaru <= 0 || anggotaBaru > 25) {
    cout << "Input tidak valid." << endl;
    cin.ignore();
    cin.clear();
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();
    cin.ignore();
    cin.ignore();
    break;
}

grupBaru.jumlahAnggota = anggotaBaru;
cin.ignore();

cout << "Nama Anggota (pisahkan dengan koma): ";
string anggotaInput;
getline(cin, anggotaInput);
string anggota;
int indexAnggota = 0;
for (char c : anggotaInput) {
    if (c == ',') {
        grupBaru.namaAnggota[indexAnggota++] = anggota;
        anggota = "";
    } else {
        anggota += c;
    }
}

grupBaru.namaAnggota[indexAnggota] = anggota;

cout << "Nama Agensi: ";
getline(cin, grupBaru.namaAgensi);

```

```

        cout << "Tanggal Debut (TTTT-BB-HH): ";
        getline(cin, grupBaru.tanggalDebut);

        cout << "Nama Fandom: ";
        getline(cin, grupBaru.namaFandom);

        bool statusAktifBaru;
        cout << "Status Aktif (1: Aktif, 0: Tidak Aktif): ";
        cin >> statusAktifBaru;
        if (cin.fail() || (statusAktifBaru != 1 && statusAktifBaru
!= 0)) {
            cout << "Input tidak valid." << endl;
            cin.clear();
            cin.ignore();
            cout << "Tekan enter untuk melanjutkan..." << endl;
            cin.ignore();
            cin.ignore();
            cin.ignore();
            break;
        }
        grupBaru.statusAktif = statusAktifBaru;
        cin.ignore();

        daftarGrup[jumlahGrup++] = grupBaru;
        cout << "Grup K-Pop berhasil ditambahkan!" << endl;
    } else {
        cout << "Jumlah grup maksimum telah tercapai." << endl;
        cout << "Tekan enter untuk melanjutkan..." << endl;
        cin.ignore();
        cin.ignore();
    }
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();
    cin.ignore();

```

## G. Menu Lihat Daftar Grup K-Pop

Bagian berikut digunakan untuk menampilkan daftar grup K-Pop yang sudah ditambahkan.

### Source Code:

```

system("cls");

    cout << "\n--- Daftar Grup K-Pop ---\n" << endl;
    cout << string(50, '-') << endl;
    if (jumlahGrup == 0) {
        cout << "Belum ada grup K-Pop yang ditambahkan." << endl;
        cout << string(50, '-') << endl;
        cout << "Tekan enter untuk melanjutkan..." << endl;
    }

```

```

        cin.ignore();
        break;
    }

    for (int i = 0; i < jumlahGrup; ++i) {
        cout << "Nama Grup: ";
        cout << daftarGrup[i].nama << endl;
        cout << "Jumlah Anggota: " << daftarGrup[i].jumlahAnggota <<
endl;

        cout << "Nama Anggota: ";
        for (int j = 0; j < daftarGrup[i].jumlahAnggota; ++j) {
            cout << daftarGrup[i].namaAnggota[j] << (j <
daftarGrup[i].jumlahAnggota - 1 ? ", " : "");
        }
        cout << endl;
        cout << "Nama Agensi: " << daftarGrup[i].namaAgensi << endl;
        cout << "Tanggal Debut: " << daftarGrup[i].tanggalDebut <<
endl;

        cout << "Nama Fandom: " << daftarGrup[i].namaFandom << endl;
        cout << "Status: " << (daftarGrup[i].statusAktif ? "Aktif" :
"Tidak Aktif") << endl;
        cout << string(50, '-') << endl;
    }
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();

```

## H. Menu Edit Grup K-Pop

Bagian berikut digunakan untuk menampilkan menu edit data grup K-Pop dan meminta pengguna untuk memasukkan nama grup yang ingin diedit. Jika grup ditemukan, maka akan menampilkan sub menu dan meminta pengguna untuk memilih sub menu tersebut.

### Source Code:

```

system("cls");

string grupUbah;
cout << "\n--- Ubah Data Grup K-Pop ---" << endl;
cout << "Masukkan Nama Grup yang ingin diubah: ";
getline(cin, grupUbah);

bool ditemukan = false;
for (int i = 0; i < jumlahGrup; ++i) {
    if (daftarGrup[i].nama == grupUbah) {
        string anggotaInput;
        string anggota;
        int indexAnggota = 0;
        cout << "1. Nama Grup" << endl;
    }
}

```

```

        cout << "2. Jumlah Anggota" << endl;
        cout << "3. Nama Anggota" << endl;
        cout << "4. Nama Agensi" << endl;
        cout << "5. Tanggal Debut" << endl;
        cout << "6. Nama Fandom" << endl;
        cout << "7. Status Aktif" << endl;
        cout << "8. Ubah Semua Data" << endl;
        cout << "Pilih data yang ingin diubah: ";
        int pilihanUbah;
        cin >> pilihanUbah;
        cin.ignore();

        switch (pilihanUbah) {
            case 1:
                cout << "Nama Grup Baru: ";
                getline(cin, daftarGrup[i].nama);
                cout << "Data grup K-Pop berhasil diubah!" <<
endl;

                cout << "Tekan enter untuk melanjutkan..." <<
endl;

                cin.ignore();
                break;
            case 2: {
                int anggotaBaru;
                cout << "Jumlah Anggota Baru: ";
                cin >> anggotaBaru;

                if (cin.fail() || anggotaBaru <= 0 ||
anggotaBaru > 25) {

                    cout << "Input tidak valid." << endl;
                    cin.ignore();
                    cin.clear();
                    cout << "Tekan enter untuk melanjutkan..."
<< endl;

                    cin.ignore();
                    cin.ignore();
                    cin.ignore();
                    break;
                }
                daftarGrup[i].jumlahAnggota = anggotaBaru;
                cout << "Data grup K-Pop berhasil diubah!" <<
endl;

                cout << "Tekan enter untuk melanjutkan..." <<
endl;

                cin.ignore();
                cin.ignore();
                break;
            }
            case 3: {
                cout << "Nama Anggota Baru (pisahkan dengan
koma): ";

```

```

        getline(cin, anggotaInput);
        int indexAnggota = 0;
        for (char c : anggotaInput) {
            if (c == ',') {
                daftarGrup[i].namaAnggota[indexAnggota++] = anggota;
                anggota = "";
            } else {
                anggota += c;
            }
        }
        daftarGrup[i].namaAnggota[indexAnggota] =
anggota;

        cout << "Data grup K-Pop berhasil diubah!" <<
endl;

        cout << "Tekan enter untuk melanjutkan..." <<
endl;

        cin.ignore();
        break;
    }
    case 4:
        cout << "Nama Agensi Baru: ";
        getline(cin, daftarGrup[i].namaAgensi);
        cout << "Data grup K-Pop berhasil diubah!" <<
endl;

        cout << "Tekan enter untuk melanjutkan..." <<
endl;

        cin.ignore();
        break;
    case 5:
        cout << "Tanggal Debut Baru: ";
        getline(cin, daftarGrup[i].tanggalDebut);
        cout << "Data grup K-Pop berhasil diubah!" <<
endl;

        cout << "Tekan enter untuk melanjutkan..." <<
endl;

        cin.ignore();
        break;
    case 6:
        cout << "Nama Fandom Baru: ";
        getline(cin, daftarGrup[i].namaFandom);
        cout << "Data grup K-Pop berhasil diubah!" <<
endl;

        cout << "Tekan enter untuk melanjutkan..." <<
endl;

        cin.ignore();
        break;
    case 7:
        bool statusAktifBaru;
        cout << "Status Aktif Baru (1: Aktif, 0: Tidak
Aktif): ";

```

```

        cin >> statusAktifBaru;
        if (cin.fail() || (statusAktifBaru != 1 &&
statusAktifBaru != 0)) {
            cout << "Input tidak valid." << endl;
            cin.clear();
            cin.ignore();
            cout << "Tekan enter untuk melanjutkan..."
<< endl;

            cin.ignore();
            cin.ignore();
            cin.ignore();
            break;
        }
        daftarGrup[i].statusAktif = statusAktifBaru;
        cout << "Data grup K-Pop berhasil diubah!" <<
endl;

        cout << "Tekan enter untuk melanjutkan..." <<
endl;

        cin.ignore();
        cin.ignore();
        break;
    case 8:
        cout << "Nama Grup Baru: ";
        getline(cin, daftarGrup[i].nama);

        int anggotaBaru;
        cout << "Jumlah Anggota Baru: ";
        cin >> anggotaBaru;

        if (cin.fail() || anggotaBaru <= 0 ||
anggotaBaru > 25) {
            cout << "Input tidak valid." << endl;
            cin.ignore();
            cin.clear();
            cout << "Tekan enter untuk melanjutkan..."
<< endl;

            cin.ignore();
            cin.ignore();
            cin.ignore();
            break;
        }
        daftarGrup[i].jumlahAnggota = anggotaBaru;
        cin.ignore();

        cout << "Nama Anggota Baru (pisahkan dengan
koma): ";

        getline(cin, anggotaInput);
        indexAnggota = 0;
        for (char c : anggotaInput) {
            if (c == ',') {

```



```

daftarGrup[i].namaAnggota[indexAnggota++] = anggota;
        anggota = "";
    } else {
        anggota += c;
    }
}
daftarGrup[i].namaAnggota[indexAnggota] =
anggota;

    cout << "Nama Agensi Baru: ";
    getline(cin, daftarGrup[i].namaAgensi);

    cout << "Tanggal Debut Baru: ";
    getline(cin, daftarGrup[i].tanggalDebut);

    cout << "Nama Fandom Baru: ";
    getline(cin, daftarGrup[i].namaFandom);

    cout << "Status Aktif Baru (1: Aktif, 0: Tidak
Aktif): ";

    cin >> statusAktifBaru;
    if (cin.fail() || (statusAktifBaru != 1 &&
statusAktifBaru != 0)) {

        cout << "Input tidak valid." << endl;
        cin.clear();
        cin.ignore();
        cout << "Tekan enter untuk melanjutkan..."
<< endl;

        cin.ignore();
        cin.ignore();
        cin.ignore();
        break;
    }
    daftarGrup[i].statusAktif = statusAktifBaru;
    cout << "Data grup K-Pop berhasil diubah!" <<
endl;

    cout << "Tekan enter untuk melanjutkan..." <<
endl;

    cin.ignore();
    cin.ignore();
    break;
default:
    cout << "Pilihan tidak valid." << endl;
    cout << "Tekan enter untuk melanjutkan..." <<
endl;

    cin.ignore();
}
ditemukan = true;
break;
}
}

```

```

        if (!ditemukan) {
            cout << "Grup K-Pop tidak ditemukan." << endl;
            cout << "Tekan enter untuk melanjutkan..." << endl;
            cin.ignore();
            break;
        }
    }

```

## I. Menu Hapus Data Grup K-Pop

Bagian berikut digunakan untuk menampilkan menu hapus data grup K-Pop dan meminta pengguna untuk memasukkan nama grup yang ingin dihapus. Jika grup ditemukan, maka data dari grup tersebut akan dihapus.

### Source Code:

```

system("cls");
string grupHapus;
bool ditemukan = false;
cout << "\n--- Hapus Grup K-Pop ---" << endl;
cout << "Masukkan Nama Grup yang ingin dihapus: ";
getline(cin, grupHapus);

for (int i = 0; i < jumlahGrup; ++i) {
    if (daftarGrup[i].nama == grupHapus) {
        for (int j = i; j < jumlahGrup - 1; ++j) {
            daftarGrup[j] = daftarGrup[j + 1];
        }
        jumlahGrup--;
        ditemukan = true;
        cout << "Grup K-Pop berhasil dihapus!" << endl;
        cout << "Tekan enter untuk melanjutkan..." << endl;
        cin.ignore();
        break;
    }
}
if (!ditemukan) {
    cout << "Grup tidak ditemukan" << endl;
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();
    break;
}

```

## 4. Hasil Output

### A. Halaman Awal

```
o === Pilih menu ===  
  1. Registrasi Pengguna  
  2. Login Pengguna  
  3. Keluar  
Pilihan: █
```

Gambar 4.1 Tampilan Halaman Awal

### B. Menu Registrasi

```
o === Registrasi Pengguna ===  
  Nama: ren  
  Password: ren  
  Registrasi berhasil!  
  Tekan enter untuk melanjutkan...  
  █
```

Gambar 4.2 Tampilan Menu Registrasi

### C. Menu Login

```
o === Login Pengguna ===  
  Masukkan Nama: ren  
  Masukkan Password: ren  
  Login berhasil!  
  Tekan enter untuk melanjutkan...  
  █
```

Gambar 4.3 Tampilan Menu Login

#### D. Menu Admin

```
=== Manajemen Grup K-Pop ===  
1. Tambah Grup K-Pop  
2. Lihat Daftar Grup K-Pop  
3. Ubah Data Grup K-Pop  
4. Hapus Grup K-Pop  
5. Keluar  
Pilihan Anda: █
```

Gambar 4.4 Tampilan Menu Admin

#### E. Menu User

```
o === Menu User ===  
1. Lihat Daftar Grup K-Pop  
2. Keluar  
Pilihan: █
```

Gambar 4.5 Tampilan Menu User

#### F. Menu Tambah Grup K-Pop

```
--- Tambah Grup K-Pop ---  
Nama Grup: dummy  
Jumlah Anggota: 3  
Nama Anggota (pisahkan dengan koma): dummy 1, dummy 2, dummy 3  
Nama Agensi: dummy ent  
Tanggal Debut (TTTT-BB-HH): 0001-01-01  
Nama Fandom: dummies  
Status Aktif (1: Aktif, 0: Tidak Aktif): 1  
Grup K-Pop berhasil ditambahkan!  
Tekan enter untuk melanjutkan...  
█
```

Gambar 4.6 Tampilan Menu Tambah Grup K-Pop

### G. Menu Lihat Daftar Grup K-Pop

```
--- Daftar Grup K-Pop ---  
  
-----  
Nama Grup: GFRIEND  
Jumlah Anggota: 6  
Nama Anggota: Sowon, Yerin, Eunha, Yuju, SinB, Umji  
Nama Agensi: Source Music  
Tanggal Debut: 2015-01-16  
Nama Fandom: BUDDY  
Status: Aktif
```

Gambar 4.7 Tampilan Menu Lihat Daftar Grup K-Pop

### H. Menu Edit Grup K-Pop

```
--- Ubah Data Grup K-Pop ---  
Masukkan Nama Grup yang ingin diubah: dummy  
1. Nama Grup  
2. Jumlah Anggota  
3. Nama Anggota  
4. Nama Agensi  
5. Tanggal Debut  
6. Nama Fandom  
7. Status Aktif  
8. Ubah Semua Data  
Pilih data yang ingin diubah: █
```

Gambar 4.8 Tampilan Menu Edit Grup K-Pop

## I. Menu Hapus Data Grup K-Pop

```
--- Hapus Grup K-Pop ---
Masukkan Nama Grup yang ingin dihapus: dummy
Grup K-Pop berhasil dihapus!
Tekan enter untuk melanjutkan...
█
```

Gambar 4.9 Tampilan Menu Hapus Data Grup K-Pop

## 5. Langkah-langkah Git

### A. Git Add

```
Ren@Rencchi 100% ▬ Praktikum-APL ○ main
→ git add .
```

Gambar 5.1 *git add*

*git add* bertujuan untuk menambah file atau perubahan apapun yang dilakukan kedalam *commit* berikutnya. Perintah “.” digunakan karena bertujuan untuk langsung memasukkan semua perubahan kedalam *commit* berikutnya, dapat juga melakukan *git add <namaFile>* agar hanya file tersebut yang berubah.

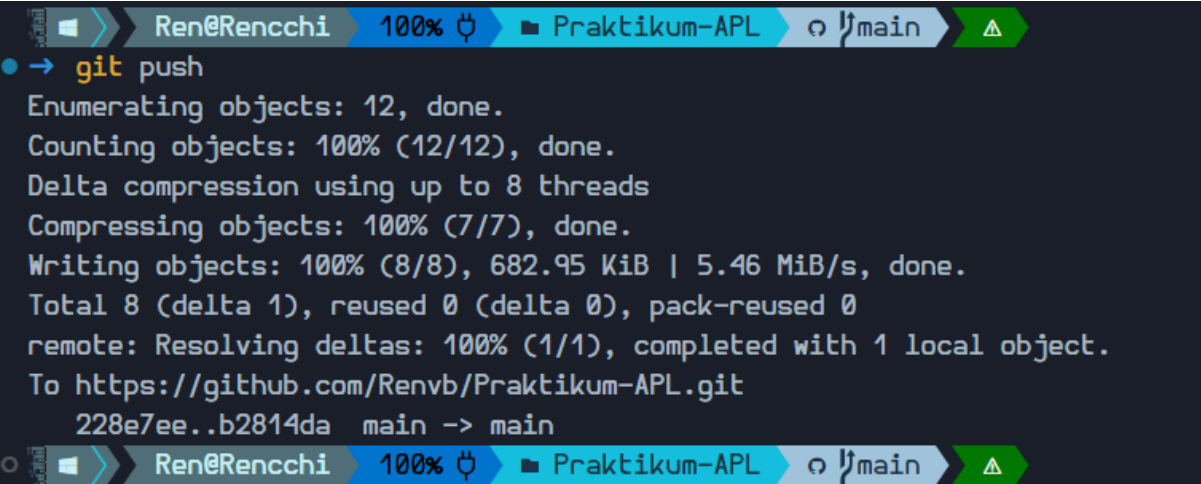
### B. Git Commit




```
Ren@Rencchi 100% ▬ Praktikum-APL ○ main
→ git commit -m "commit program"
[main b2814da] commit program
3 files changed, 433 insertions(+), 1 deletion(-)
create mode 100644 Post-Test/Post-Test-2/2409106076-RazibRamadhan-PT-2.cpp
create mode 100644 Post-Test/Post-Test-2/2409106076-RazibRamadhan-PT-2.exe
Ren@Rencchi 100% ▬ Praktikum-APL ○ main
```

Gambar 5.2 *git commit*

*git commit* digunakan untuk membuat *checkpoint* file yang telah ditambahkan pada *git add*, sehingga kondisi file saat ini dapat dilihat kembali di masa depan.

### C. Git Push



```
Ren@Rencchi 100%  Praktikum-APL  main 
→ git push
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 8 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 682.95 KiB | 5.46 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Renvb/Praktikum-APL.git
228e7ee..b2814da main -> main
```

Gambar 5.6 Git Push

Setelah repository lokal telah terhubung dengan repository server seperti GitHub, perintah *git push* akan mengunggah seluruh *commit* yang terdapat pada repository lokal dalam satu branch yang sama. Sehingga progres dari repository di GitHub akan menampilkan *commit* terakhir dari repository lokal.