

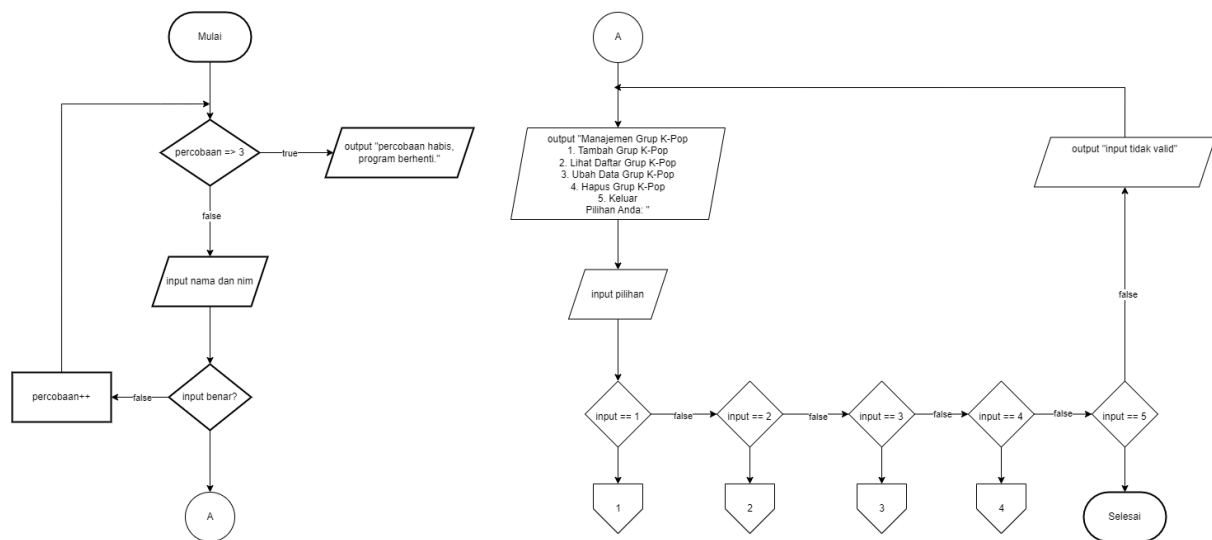
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



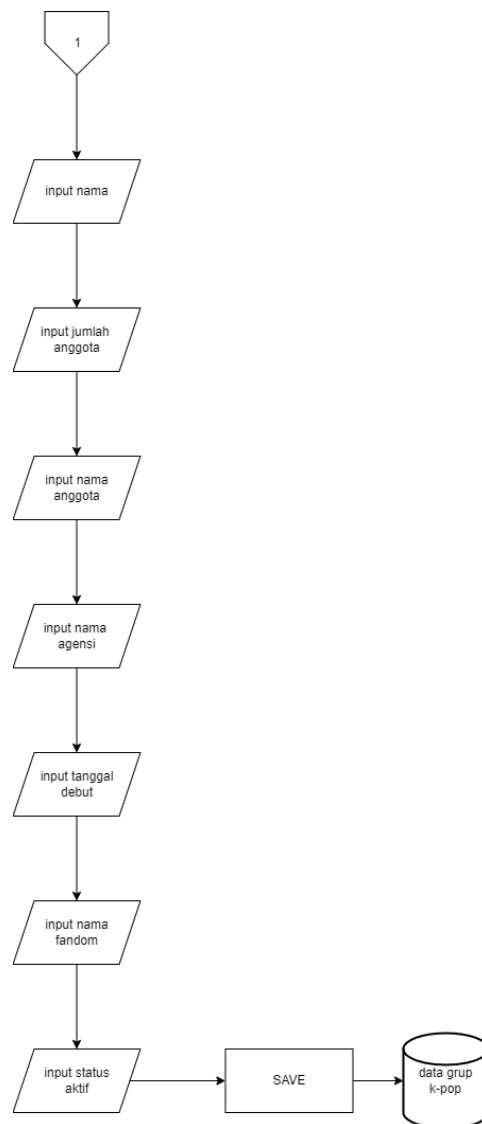
Disusun oleh:
Razib Ramadhan (2409106076)
Kelas (B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

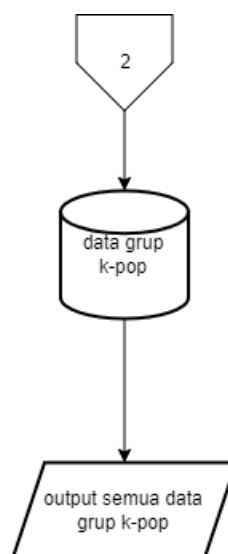
1. Flowchart



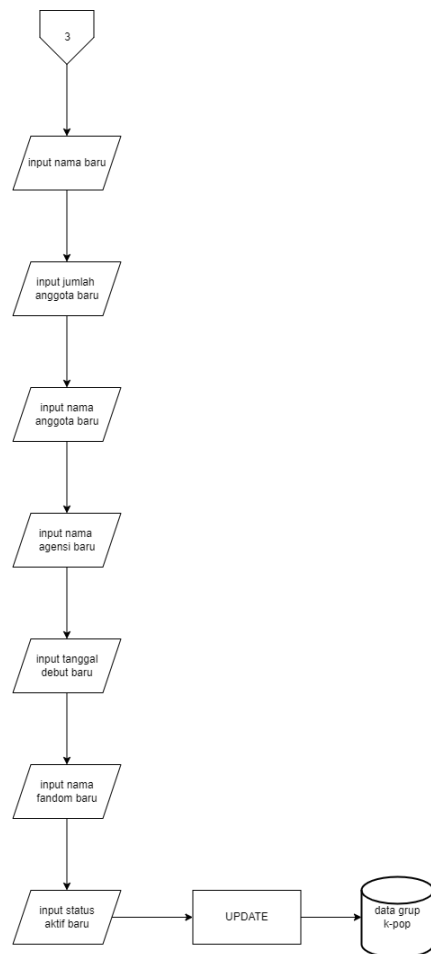
Gambar 1.1 *Main Flow* Bagian 1



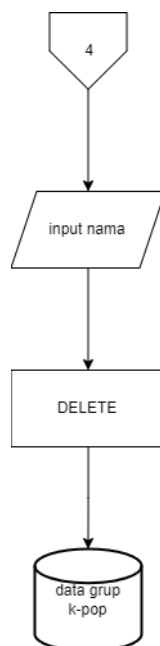
Gambar 1.2 *Main Flow* Bagian 2



Gambar 1.3 *Main Flow* Bagian 3



Gambar 1.4 *Main Flow* Bagian 4



Gambar 1.5 *Main Flow* Bagian 5

2. Analisis Program

2.1 Deskripsi Singkat 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.

2.2 Penjelasan Alur & Algoritma

1. Pengguna diberikan 3 kesempatan untuk memasukkan nama lengkap dan NIM mereka.
2. Jika pengguna salah memasukkan input, kesempatan berkurang 1. Jika kesempatan habis, program selesai. Jika pengguna benar memasukkan input, maka pengguna dapat mengakses menu utama.
3. Selama di menu utama, user dapat memilih menu menambah, melihat, mengedit, menghapus data grup K-Pop, atau keluar program.
4. Jika pengguna memilih menu menambah data grup K-Pop, maka pengguna akan diminta untuk memasukkan data baru seperti nama grup, jumlah anggota, nama anggota, nama agensi, tanggal debut, nama fandom, dan status aktif.
5. Jika pengguna memilih menu melihat data grup K-Pop, maka akan muncul semua data dari grup K-Pop yang sudah ditambahkan.
6. Jika pengguna memilih menu mengedit data grup K-Pop, user akan diminta untuk memasukkan nama grup yang ingin diedit. Jika grup ditemukan, maka akan muncul sub menu untuk mengedit salah satu data grup atau mengedit secara keseluruhan. Jika grup tidak ditemukan, maka tampilan akan kembali ke menu utama.
7. Jika pengguna memilih menu menghapus data grup K-Pop, user akan diminta untuk memasukkan nama grup yang ingin menghapus. Jika grup ditemukan, maka data dari grup tersebut akan dihapus. Jika grup tidak ditemukan, maka tampilan akan kembali ke menu utama.
8. Jika pengguna memilih keluar program, maka program akan berakhir.

3. Source Code

A. Sistem Login

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

Source Code:

```
string nama, nim;
int percobaan = 0;

while (percobaan < 3) {
    system("cls");
    cout << "Masukkan Nama: ";
    getline(cin, nama);
    cout << "Masukkan NIM: ";
    getline(cin, nim);

    if (nama == "Razib Ramadhan" && nim == "2409106076") {
        break;
    } else {
        percobaan++;
        cout << "Nama atau NIM 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;
}
```

B. Menu Utama

Bagian berikut digunakan untuk menampilkan menu utama yang tersedia dan meminta pengguna untuk memilih menu.

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();
```

C. 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_GRUP) {
    cout << "\n--- Tambah Grup K-Pop ---" << endl;
    string grupBaru;
    cout << "Nama Grup: ";
    getline(cin, grupBaru);

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

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

    namaGrup[jumlahGrup] = grupBaru;
```

```

        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;
        }
        jumlahAnggota[jumlahGrup] = 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 == ',') {
                namaAnggota[jumlahGrup][indexAnggota++] = anggota;
                anggota = "";
            } else {
                anggota += c;
            }
        }
        namaAnggota[jumlahGrup][indexAnggota] = anggota;

        cout << "Nama Agensi: ";
        getline(cin, namaAgensi[jumlahGrup]);

        cout << "Tanggal Debut (TTTT-BB-HH): ";
        getline(cin, tanggalDebut[jumlahGrup]);

        cout << "Nama Fandom: ";
        getline(cin, namaFandom[jumlahGrup]);

        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();
        break;
    }
    statusAktif[jumlahGrup] = statusAktifBaru;
    cin.ignore();

    cout << "Grup K-Pop berhasil ditambahkan!" << endl;
    jumlahGrup++;
} 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();

```

D. 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: " << namaGrup[i] << endl;
    cout << "Jumlah Anggota: " << jumlahAnggota[i] << endl;
    cout << "Nama Anggota: ";
    for (int j = 0; j < jumlahAnggota[i]; ++j) {
        cout << namaAnggota[i][j] << (j < jumlahAnggota[i] - 1 ?
", " : "");
    }
    cout << endl;
    cout << "Nama Agensi: " << namaAgensi[i] << endl;
    cout << "Tanggal Debut: " << tanggalDebut[i] << endl;
    cout << "Nama Fandom: " << namaFandom[i] << endl;
    cout << "Status: " << (statusAktif[i] ? "Aktif" : "Tidak
Aktif") << endl;
}

```

```

        cout << string(50, '-') << endl;
    }
    cout << "Tekan enter untuk melanjutkan..." << endl;
    cin.ignore();

```

E. 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 (namaGrup[i] == 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();
    }
}

```

F. 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 (namaGrup[i] == grupHapus) {
        for (int j = i; j < jumlahGrup - 1; ++j) {
            namaGrup[j] = namaGrup[j + 1];
            jumlahAnggota[j] = jumlahAnggota[j + 1];
            for(int k=0; k<25; k++){
                namaAnggota[j][k] = namaAnggota[j+1][k];
            }
            namaAgensi[j] = namaAgensi[j + 1];
            tanggalDebut[j] = tanggalDebut[j + 1];
            namaFandom[j] = namaFandom[j + 1];
            statusAktif[j] = statusAktif[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;
}
```

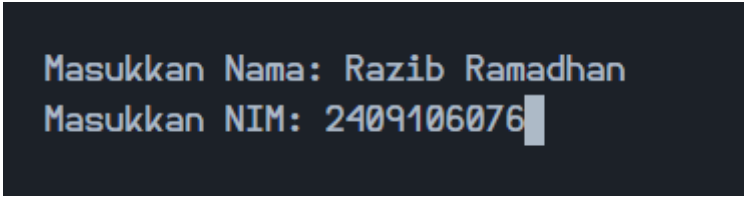
4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Ren ingin menggunakan program untuk mengubah status salah satu grup menjadi aktif kembali setelah tidak aktif selama empat tahun.
2. Ren ingin menggunakan program, namun ia lupa NIM-nya. Ia mencoba tiga kali dan mengakibatkan programnya terhenti.

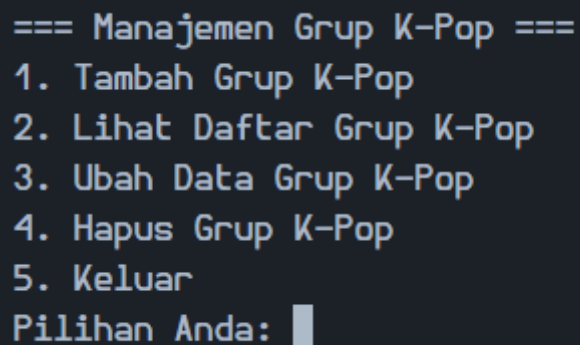
4.2 Hasil Output

A. Skenario 1



```
Masukkan Nama: Razib Ramadhan  
Masukkan NIM: 2409106076
```

Gambar 4.2.1 Proses Login (Skenario 1)



```
=== 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 4.2.2 Tampilan Menu Utama (Skenario 1)

```
--- Ubah Data Grup K-Pop ---  
Masukkan Nama Grup yang ingin diubah: GFRIEND  
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 4.2.3 Tampilan Menu Ubah Data (Skenario 1)

```
Pilih data yang ingin diubah: 7  
Status Aktif Baru (1: Aktif, 0: Tidak Aktif): 1  
Data grup K-Pop berhasil diubah!  
Tekan enter untuk melanjutkan...  
█
```

Gambar 4.2.4 Tampilan Ubah Data (Skenario 1)

```
--- 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 4.2.5 Tampilan Data Yang Sudah Diubah (Skenario 1)

B. Skenario 2

```
Masukkan Nama: Ren
Masukkan NIM: 000
Nama atau NIM salah. Percobaan ke-1
Tekan enter untuk melanjutkan...
█
```

Gambar 4.2.6 Proses Login Gagal Satu Kali (Skenario 2)

```
Masukkan Nama: Ren
Masukkan NIM: 2409106076
Nama atau NIM salah. Percobaan ke-2
Tekan enter untuk melanjutkan...
█
```

Gambar 4.2.7 Proses Login Gagal Dua Kali (Skenario 2)

```
Masukkan Nama: Ren
Masukkan NIM: 12345
Nama atau NIM salah. Percobaan ke-3
Tekan enter untuk melanjutkan...
█
```

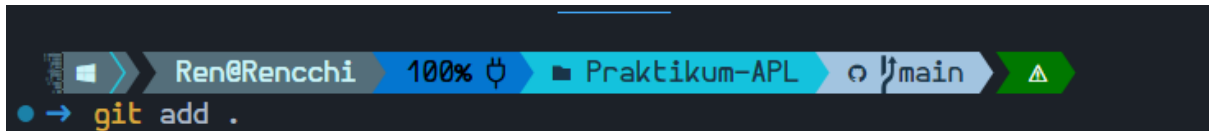
Gambar 4.2.8 Proses Login Gagal Tiga Kali (Skenario 2)

```
Anda telah gagal 3 kali. Program berakhir.
Tekan enter untuk melanjutkan...
█
```

Gambar 4.2.9 Tampilan Program Berakhir (Skenario 2)

5. Langkah-langkah Git

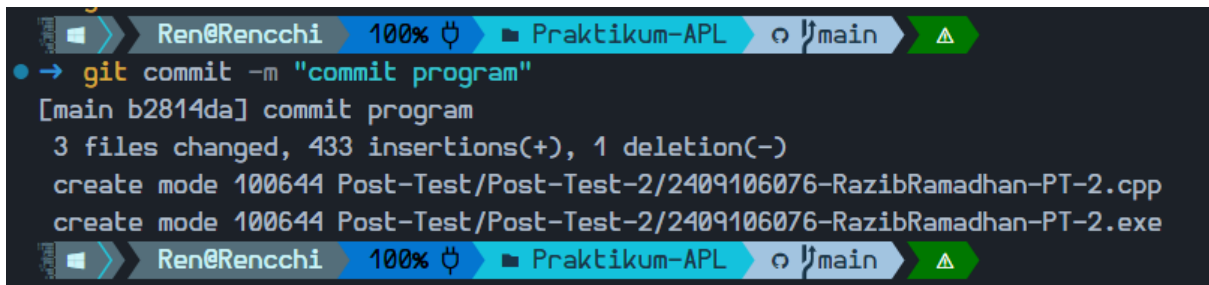
A. Git Add



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 memasukan semua perubahan kedalam *commit* berikutnya, dapat juga melakukan *git add <namaFile>* agar hanya file tersebut yang berubah.

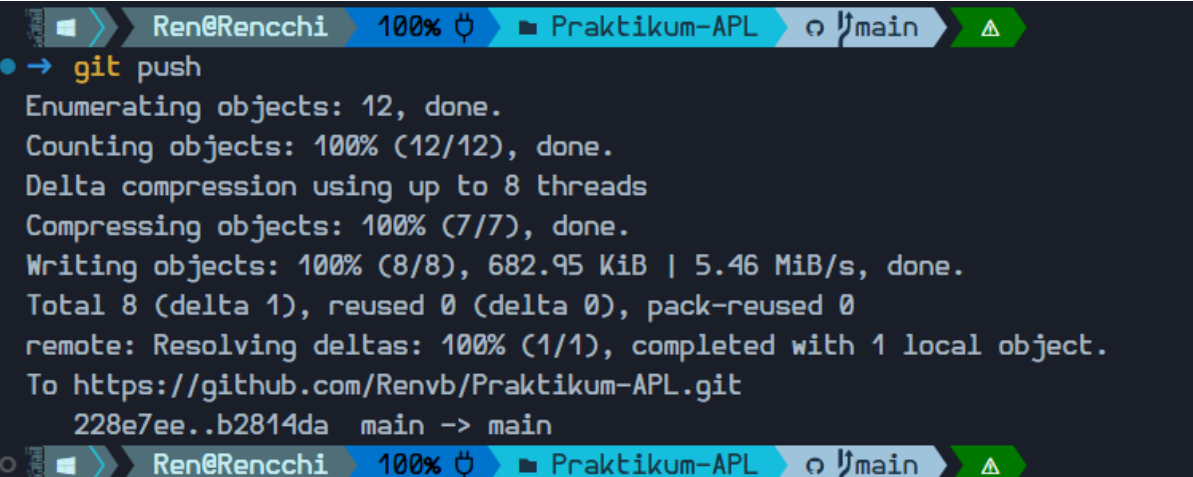
B. Git Commit






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.