

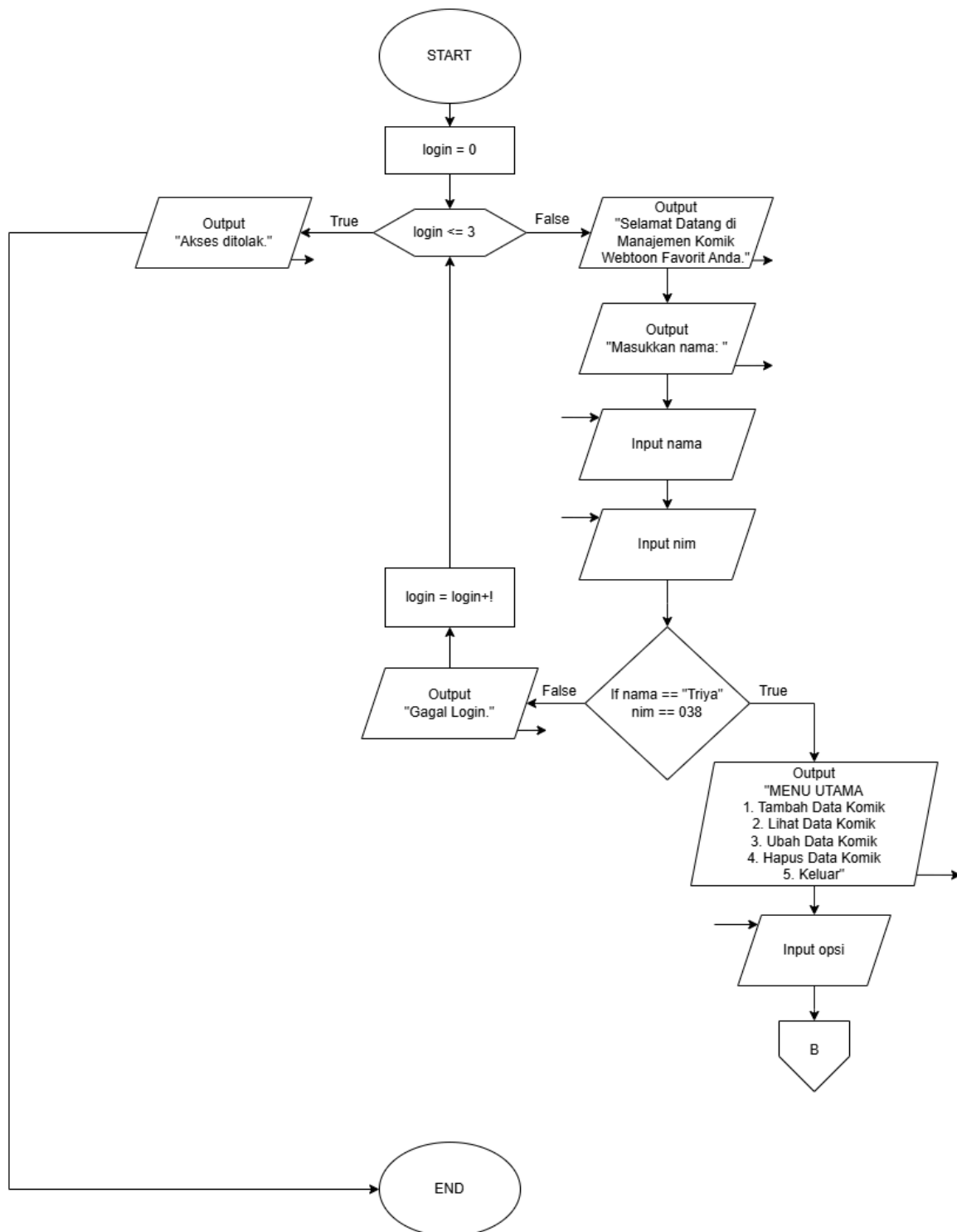
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



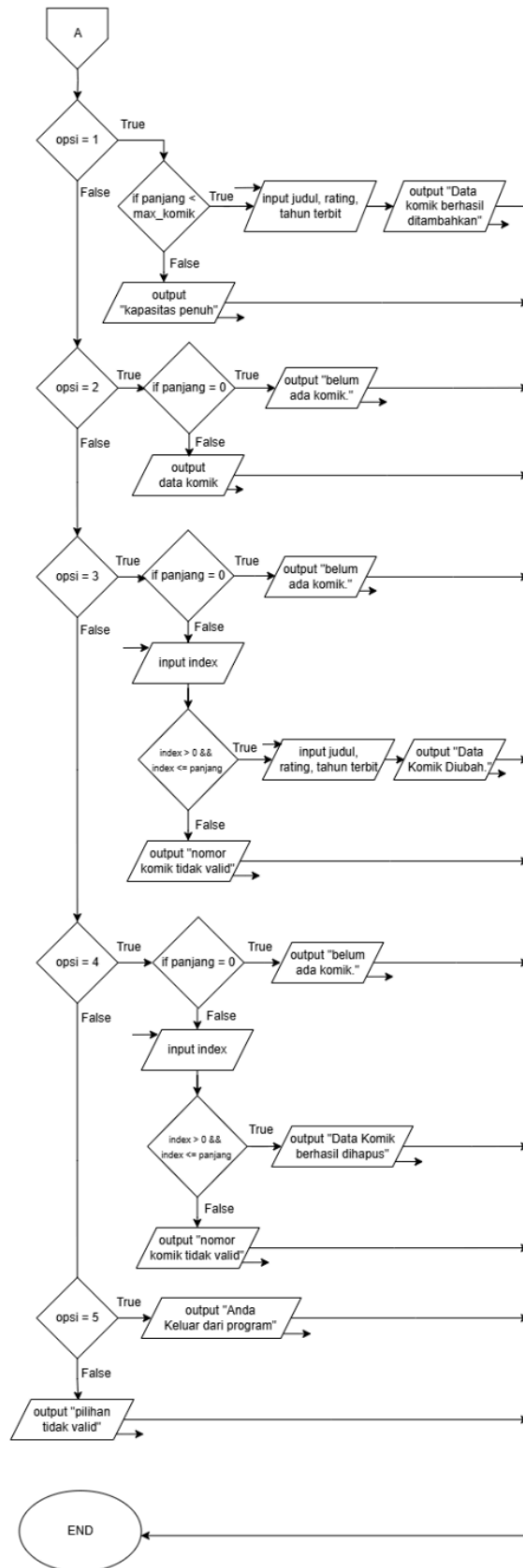
**Disusun oleh:**  
**Triya Khairun Nisa (2409106038)**  
**Kelas (A2 '24)**

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

## 1. Flowchart



Gambar 1. 1 Flowchart 1



Gambar 1. 2 Flowchart 2

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini bertujuan untuk membantu pengguna dalam manajemen data-data komik webtoon favorit yang ingin disimpan dengan menerapkan sistem CRUD, yakni konsep dasar dalam pengelolaan data yang mencakup *Create* (menambah data), *Read* (membaca data), *Update* (memperbarui data), dan *Delete* (menghapus data). Pengguna dapat memasukkan dan menyimpan data komik baru ke dalam program. Dengan program ini pengguna diharapkan dapat terbantu dalam menyimpan data komik favoritnya.

### 2.2 Penjelasan Alur & Algoritma

Pertama, program diawali dengan (fungsi `main()`) yang akan menampilkan pesan kepada pengguna yang berisikan “Selamat Datang di Manajemen Komik Webtoon Favorit”. Kemudian, pengguna akan diminta untuk mengisi ‘nama’ dan ‘nim’. Jika nama dan NIM cocok ("Triya" dan "038"), program akan menampilkan pesan sukses login dan masuk ke menu utama. Pengguna akan diberikan maksimal 3 kali kesempatan untuk memasukkan nama dan NIM. Jika pengguna gagal login setelah 3 kali percobaan, program akan menampilkan pesan akses ditolak dan berhenti.

Setelah pengguna berhasil masuk ke menu utama, pengguna akan diminta memilih opsi berdasarkan menu yang tersedia:

1. Jika pengguna memilih opsi 1, maka pengguna akan diminta untuk memasukkan judul, rating, dan tahun terbit jika jumlah data yang tersimpan masih di bawah jumlah maksimum data, yakni 100 data. Namun, apabila data yang tersimpan sudah melebihi batas maka program akan menampilkan “Kapasitas Penuh”.
2. Jika pengguna memilih opsi 2, maka program akan menampilkan data komik yang tersimpan dalam program. Namun, jika tidak ada data yang ditemukan di dalam program maka output akan menampilkan “Belum ada komik.”
3. Jika pengguna memilih opsi 3, maka program akan meminta pengguna untuk memasukkan nomor dari komik yang ingin diubah. Jika nomor yang dimasukkan tidak terdapat dalam data yang tersedia, maka program akan menampilkan “Nomor

komik tidak valid” tetapi jika nomor ditemukan maka program akan meminta pengguna untuk memasukkan judul, rating, dan tahun terbit baru. Jika data berhasil diubah, maka program akan menampilkan “Data komik berhasil diubah.”.

4. Jika pengguna memilih opsi 4, maka program akan meminta pengguna untuk memasukkan nomor dari data komik yang ingin diubah. Jika nomor ditemukan di dalam data program, maka program akan menghapus data komik yang terdapat pada nomor tersebut. Namun, apabila nomor yang dimasukkan tidak terdapat dalam program, maka program akan menampilkan “Nomor komik tidak valid.”
5. Jika pengguna memilih opsi 5, maka program akan ditutup.
6. Jika opsi yang dimasukkan selain dari 1, 2 , 3, 4, ataupun 5, maka program akan menampilkan “Pilihan tidak valid.”.

### 3. Source Code

#### A. Fitur Login

Fitur ini digunakan untuk masuk ke dalam menu utama dan mengidentifikasi pengguna.

**Source Code:**

```
int main()
{
    bool masuk = false;
    for (int login = 1; login <= 3; login++) {
        cout << CYAN << "= Selamat Datang di Manajemen Komik Webtoon Favorit\n" << RESET;
        cout << "Masukkan nama: ";
        cin >> nama;
        cout << "Masukkan NIM: ";
        cin >> nim;
        if (nama == "Triya" && nim == "038") {
            cout << "Berhasil Login, Hi " << nama << endl;
            masuk = true;
        }
    }
}
```

#### B. Fitur Menu Utama

Fitur ini digunakan untuk menampilkan opsi yang tersedia untuk pengguna.

Source code:

```
int panjang = 0;
string komik[max_komik][INFO];
int pilihan, index;
do {;
```

```
cout << CYAN << "===== Menu Utama =====" << endl;
cout << "1. Tambah Data Komik" << endl;
cout << "2. Lihat Data Komik" << endl;
cout << "3. Ubah Data Komik" << endl;
cout << "4. Hapus Data Komik" << endl;
cout << "5. Keluar" << endl;
cout << "Pilihan: " << RESET;
cin >> pilihan;
switch (pilihan) {
```

### C. Fitur Create (Menambahkan Data Komik)

Fitur ini digunakan untuk menambahkan data komik yang berisi judul, rating, dan genre.

Source code:

```
case 1:
if (panjang < max_komik) {
    cout << "Masukkan judul komik: ";
    cin.ignore();
    getline(cin, komik[panjang][0]);
    cout << "Masukkan rating komik: ";
    getline(cin, komik[panjang][1]);
    cout << "Masukkan tahun terbit komik: ";
    getline(cin, komik[panjang][2]);
    panjang++;
    cout << "Data Komik berhasil ditambahkan" << endl;
} else {
    cout << "Kapasitas penuh! Tidak bisa menambahkan komik." <<
endl;
}
break;
```

#### D. Fitur Read (Melihat Data Komik)

Fitur ini digunakan untuk melihat data komik yang tersimpan dalam program.

Source code:

```

        case 2:
        if (panjang == 0) {
            cout << "Belum ada komik" << endl;
        } else {
            for (int i = 0; i < panjang; i++) {
                cout << MAGENTA <<
                "+-----+-----+-----+-----+\n";
                cout << i + 1 << " | Judul: " << komik[i][0] << " | Rating " <<
                komik[i][1] << " | Tahun terbit: " << komik[i][2] << " " << endl;
            }
        }
    }
}

```

```
cout <<
"+-----+-----+-----+\n" << RESET;
break;
```

### E. Fitur Update (Mengubah Data Komik)

Fitur ini digunakan untuk mengubah atau memperbaharui data komik yang tersimpan dalam program.

Source code:

```

        case 3:
        if (panjang == 0) {
            cout << "Belum ada komik untuk diubah." << endl;
        } else {
            for (int i = 0; i < panjang; i++) {
                cout << "komik ke-" << i + 1 << ": " << komik[i][0] << " (" <<
komik[i][1]
                << ")" << endl;
            }
            cout << "Masukkan nomor komik yang akan diubah: ";
            cin >> index;
            if (index > 0 && index <= panjang) {
                cout << "Masukkan nama komik baru: ";
                cin.ignore();
                getline(cin, komik[index - 1][0]);
                cout << "Masukkan rating baru: ";
                getline(cin, komik[index - 1][1]);
                cout << "Masukkan tahun terbit baru: ";
                getline(cin, komik[index - 1][2]);
                cout << "Data komik berhasil diubah" << endl;
            } else {
                cout << "Nomor komik tidak valid" << endl;
            }
        }
        break;
    }
}

```

## F. Fitur Delete (Menghapus Data Komik)

Fitur ini digunakan untuk menghapus data komik yang tersimpan dalam program.

Source code:

```
case 4:
    if (panjang == 0) {
        cout << "Belum ada komik untuk dihapus." << endl;
    } else {
        for (int i = 0; i < panjang; i++) {
```

```

        cout << "komik ke-" << i + 1 << ": " << komik[i][0] << " (" <<
komik[i][1]
        << ")" << endl;
    }
    cout << "Masukkan nomor komik yang akan dihapus: ";
    cin >> index;
    if (index > 0 && index <= panjang) {
        for (int i = index - 1; i < panjang - 1; i++) {
            komik[i][0] = komik[i + 1][0];
            komik[i][1] = komik[i + 1][1];
        }
        panjang--;
        cout << "Data Komik berhasil dihapus" << endl;
    } else {
        cout << "Nomor komik tidak valid" << endl;
    }
}

```

## G. Fitur Keluar dari Program

Fitur ini digunakan ketika pengguna telah selesai menggunakan dan ingin menutup program.

Source code:

```

case 5:
    cout << YELLOW << "Anda Keluar dari Program, bye bye :D\n";
    cout << " /\_/\ " << endl;
    cout << "( o.o ) " << endl;
    cout << " > ^ < " << endl << RESET;
    break;

```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

#### 1. Skenario 1

Pengguna memasukkan input selain “Triya” sebagai nama dan “038” sebagai NIM. Ketika pengguna memulai program, ia memasukkan “Nabila” sebagai nama dan “041” sebagai NIM. Lalu pada percobaan kedua, ia memasukkan “Triya” dan “001” sebagai NIM. Pada Percobaan ketiga, ia memasukkan “Niffari” sebagai nama dan “038” sebagai NIM.

#### 2. Skenario 2



Pengguna sebelumnya ingin memasukkan data komik favoritnya yang berjudul “Lore Olympus”. Ketika ia memulai program, ia memasukkan “Triya” sebagai nama dan “038” sebagai NIM. Kemudian ia memilih opsi 1 dan memasukkan judul komik yang ia ingin masukkan, yakni “Lore Olympus”. Tak hanya itu, ia juga memasukkan rating dan tahun terbit dari program. Setelah itu, ia ingin mengecek apakah data yang ia masukkan sudah masuk dengan memasukkan opsi 2. Kemudian ia ingin menambahkan data lagi, maka dari itu ia kembali memasukkan opsi 1 dan memasukkan data baru. Lalu, ia kembali mengecek datanya dengan memasukkan opsi 2. Setelah itu, ia ingin mengubah data sebelumnya sehingga ia memilih opsi 3 dan mengubah data komik sebelumnya menjadi data baru dengan memasukkan nomor dari komik tersebut dan memasukkan data baru. Kemudian, ia memasukkan opsi 4 dan menghapus data dari komik “WEE!!!” dengan memasukkan nomor komiknya. Sebelum menutup program, ia ingin mengecek datanya lagi dengan memasukkan opsi 2. Setelah ia merasa data yang ia masukkan sudah benar, ia menutup program dengan memasukkan opsi 5.

## 4.2 Hasil Output

```
= Selamat Datang di Manajemen Komik Webtoon Favorit =  
Masukkan nama: Nabila  
Masukkan NIM: 041  
Login gagal. Silakan coba lagi.  
= Selamat Datang di Manajemen Komik Webtoon Favorit =  
Masukkan nama: Triya  
Masukkan NIM: 001  
Login gagal. Silakan coba lagi.  
= Selamat Datang di Manajemen Komik Webtoon Favorit =  
Masukkan nama: Niffari  
Masukkan NIM: 038  
Login gagal. Silakan coba lagi.  
Kesempatan login habis. Akses ditolak!  
PS C:\Users\triya\praktikum-apl\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.1 Skenario 1

```
= Selamat Datang di Manajemen Komik Webtoon Favorit =
Masukkan nama: Triya
Masukkan NIM: 038
Berhasil Login, Hi Triya
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 1
Masukkan judul komik: Lore Olympus
Masukkan rating komik: 7/10
Masukkan tahun terbit komik: 2010
Data Komik berhasil ditambahkan
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 1
Masukkan judul komik: WEE!!!
Masukkan rating komik: 9/10
Masukkan tahun terbit komik: 2019
Data Komik berhasil ditambahkan
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
```

Gambar 4.2 Skenario 2A

```
Pilihan: 2
+---+-----+-----+-----+-----+
1 | Judul: Lore Olympus | Rating 7/10 | Tahun terbit: 2010
+---+-----+-----+-----+-----+
2 | Judul: WEE!!! | Rating 9/10 | Tahun terbit: 2019
+---+-----+-----+-----+-----+
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 3
komik ke-1: Lore Olympus (7/10)
komik ke-2: WEE!!! (9/10)
Masukkan nomor komik yang akan diubah: 1
Masukkan nama komik baru: Tower Of God
Masukkan rating baru: 7/10
Masukkan tahun terbit baru: 2011
Data komik berhasil diubah
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 4
komik ke-1: Tower Of God (7/10)
komik ke-2: WEE!!! (9/10)
Masukkan nomor komik yang akan dihapus: 2
Data Komik berhasil dihapus
===== Menu Utama =====
1. Tambah Data Komik
```

Gambar 4.3 Skenario 2B

```
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 2
+---+-----+-----+-----+
1 | Judul: Tower Of God | Rating 7/10 | Tahun terbit: 2011
+---+-----+-----+-----+
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 5
Anda Keluar dari Program, bye bye :D
/\_/\
( o.o )
> ^ <
PS C:\Users\triva\praktikum-apl\praktikum-apl\post-test\post-test-apl-2>
```

Gambar 4.4 Skenario 2C

## 5. Git

Git berfungsi agar banyak pengembang dapat bekerja secara bersamaan, melacak revisi, dan mengelola riwayat perubahan proyek dengan efisien dibanding mengirim program secara berulang. Dengan demikian, kita dapat mengurangi penggunaan memori perangkat dan file.

```
PS C:\Users\triva\praktikum-apl\praktikum-apl> git init
Reinitialized existing Git repository in C:/Users/triva/praktikum-apl/praktikum-apl/.git/
PS C:\Users\triva\praktikum-apl\praktikum-apl> git add .
```

Gambar 5.1 inisiasi & tambah Git

### a) git init

- ❖ Perintah ini digunakan untuk menginisialisasi atau membuat repository Git di dalam folder proyek.

### b) git add .

- ❖ Perintah ini digunakan untuk menambahkan semua perubahan (file baru, perubahan pada file yang ada, dan file yang dihapus) ke dalam staging area sebelum dilakukan commit.

```
PS C:\Users\triya\praktikum-apl\praktikum-apl> git commit -m "Finish Coding"
[main d57436a] Finish Coding
8 files changed, 257 insertions(+)
create mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-2.cpp
rename post-test/post-test-apl-1/2409106038-TriyaKhairunNisa-PT-1.exe => kelas/pertemuan-2.exe (52%)
create mode 100644 kelas/tempCodeRunnerFile.cpp
create mode 100644 post-test/post-test-apl-2/2409106038-TriyaKhairunNisa-PT-2.cpp
create mode 100644 post-test/post-test-apl-2/2409106038-TriyaKhairunNisa-PT-2.exe
create mode 100644 post-test/post-test-apl-2/tempCodeRunnerFile.cpp
create mode 100644 tempCodeRunnerFile.cpp
```

Gambar 5.2 git commit

### c) git commit -m

- ❖ Perintah ini digunakan untuk menyimpan perubahan yang telah ditambahkan ke dalam staging area dengan pesan commit.

```
PS C:\Users\triya\praktikum-apl\praktikum-apl> git remote add origin https://github.com/3yakhansa/praktikum-apl.git
error: remote origin already exists.
PS C:\Users\triya\praktikum-apl\praktikum-apl> git push -u origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 8 threads
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 57.91 KiB | 2.52 MiB/s, done.
Total 15 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/3yakhansa/praktikum-apl.git
27b5665..d57436a main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Users\triya\praktikum-apl\praktikum-apl>
```

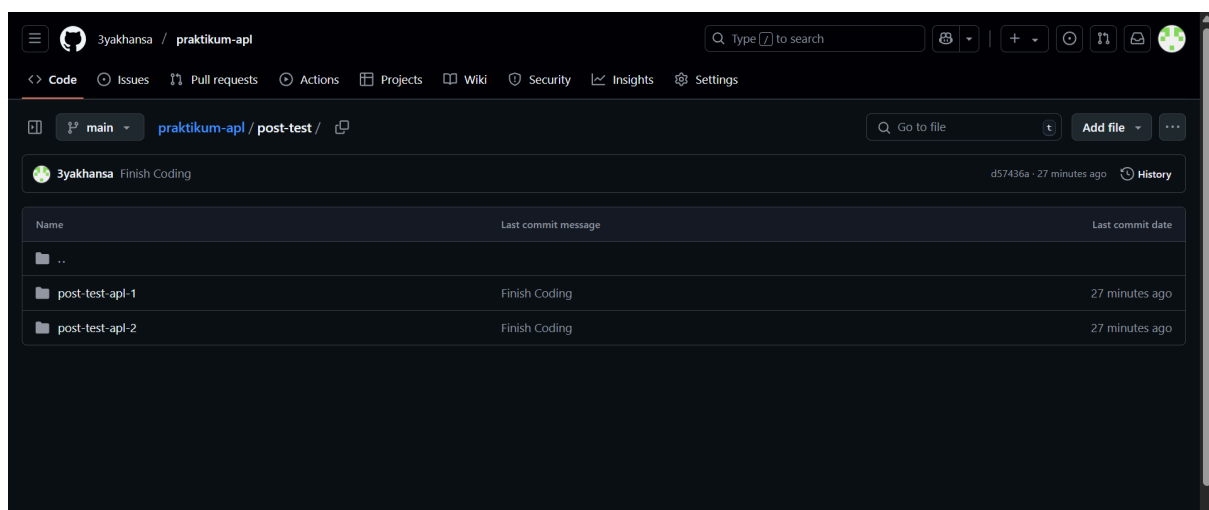
Gambar 5.3 git remote & git push

### d) git remote

- ❖ git remote adalah perintah dalam Git yang digunakan untuk mengelola repositori jarak jauh (*remote repository*).

### e) git push

- ❖ Git push adalah perintah yang berfungsi untuk mengunggah perubahan file dari repositori lokal ke repositori jarak jauh. Perintah ini digunakan untuk memperbarui cabang jarak jauh dengan komitmen lokal.



#### Gambar 5.4 Tampilan Github

Jika *push* berhasil, maka tampilan pada github akan berubah. Dapat terlihat setiap file yang sebelumnya terdapat di lokal perangkat sudah di-*publish* di Github.