

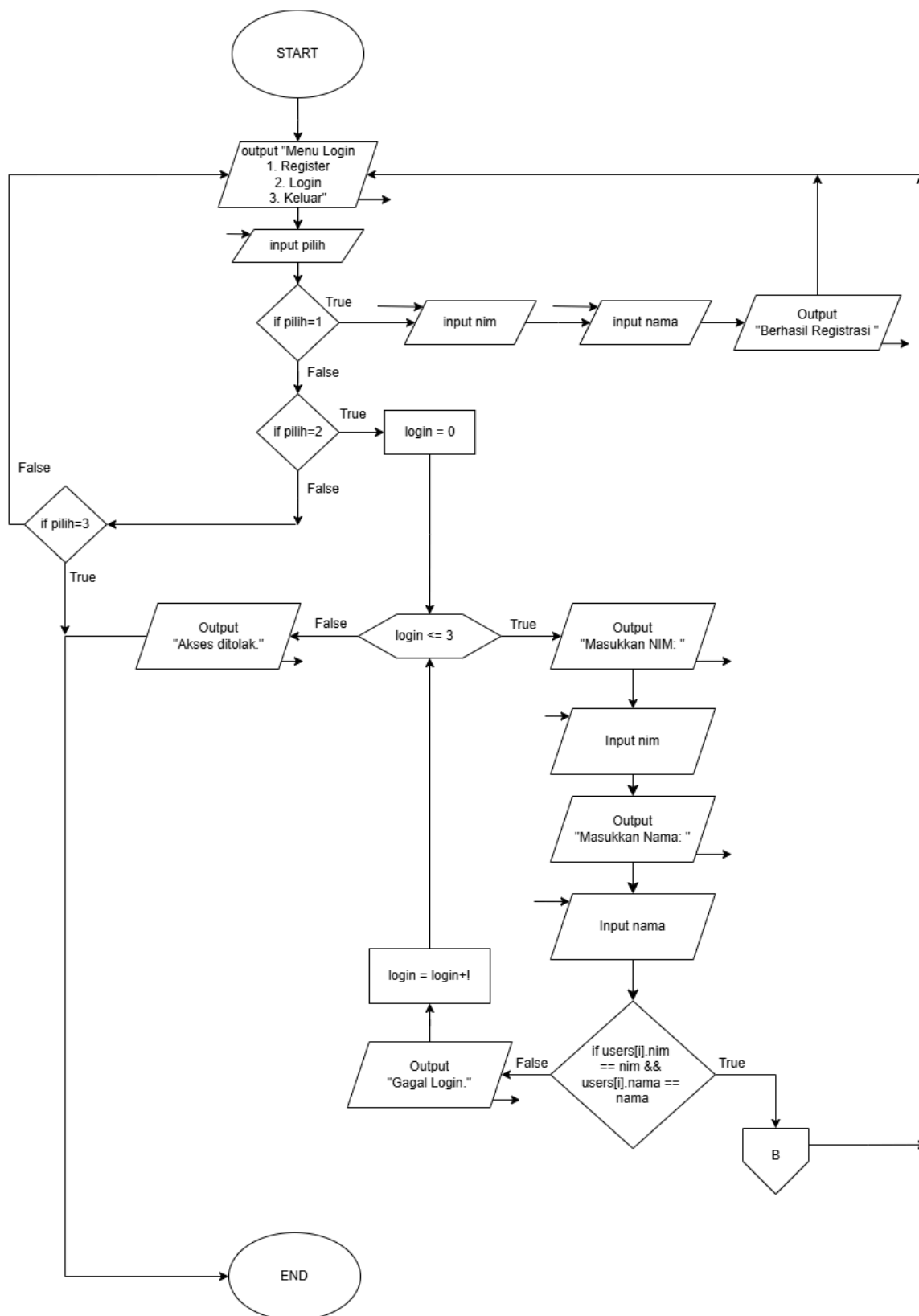
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



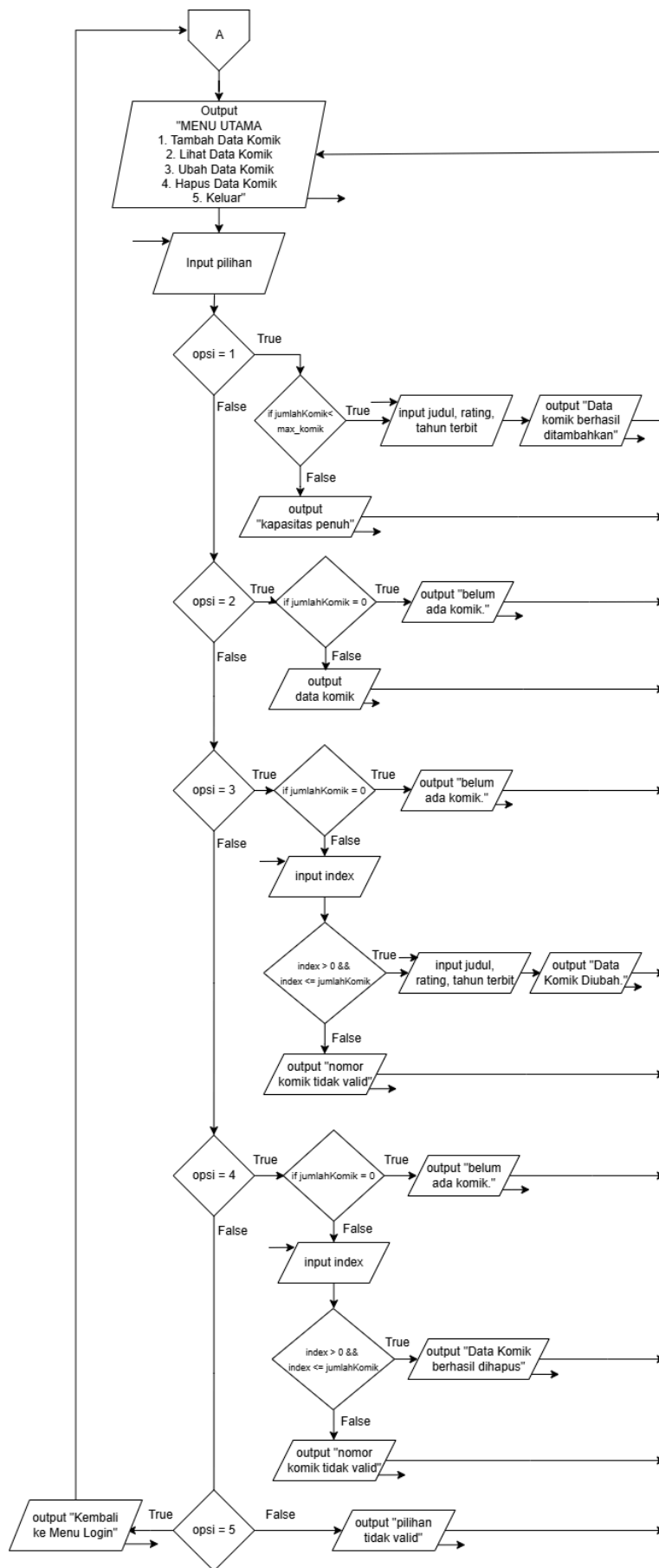
Disusun oleh:
Triya Khairun Nisa (2409106038)
Kelas (A2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart A



Gambar 1.2 Flowchart B

2. Analisis Program

Program ini bertujuan untuk membantu pengguna dalam memanajemen 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.

3. Source Code

A. Fitur Menu Login dan Keluar dari Program

Fitur ini digunakan bagi pengguna untuk melakukan dan menyimpan data registrasi serta *login* ke dalam program. Selain itu, terdapat fitur untuk mengeluarkan pengguna dari program.

Source Code:

```
void Register(User users[], int &jumlahUser) {
    if (jumlahUser >= MAX_USER) {
        cout << RED << "User sudah penuh!" << RESET << endl;
    } else {
        cout << "Masukkan NIM: ";
        cin >> users[jumlahUser].nim;
        cin.ignore();
        cout << "Masukkan Nama: ";
        getline(cin, users[jumlahUser].nama);
        cout << CYAN << "Berhasil Registrasi!" << RESET << endl;
        jumlahUser++;
    }
}

int Login(User users[], int jumlahUser) {
    for (int login = 1; login <= 3; login++) {
        int nim;
        string nama;
        cout << "Masukkan NIM: ";
        cin >> nim;
        cin.ignore();
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        for (int i = 0; i < jumlahUser; i++) {
            if (users[i].nim == nim && users[i].nama == nama) {
                cout << CYAN << "Login berhasil! Hai, " << users[i].nama <<
RESET << endl;
                return i;
            }
        }
    }
}
```

```

    }
}
cout << RED << "Login gagal! Coba lagi.\n" << RESET << endl;
if (login == 3) {
    cout << RED << "Akses ditolak!" << RESET << endl;
    cout << YELLOW << "Anda Keluar dari Program, bye bye :D";
    cout << " /\_/\ " << endl;
    cout << "( o.o ) " << endl;
    cout << " > ^ < " << endl << RESET;
    exit(0);
}

```

```

int main() {
    int pilih;
    while (true) {
        cout << CYAN << "===== " << endl;
        cout << "| Manajemen Komik Webtoon Favorit      |" << endl;
        cout << "| 1. Register                            |" << endl;
        cout << "| 2. Login                              |" << endl;
        cout << "| 3. Keluar                             |" << endl;
        cout << "===== " << RESET << endl;
        cout << "PILIH: ";
        cin >> pilih;
        cin.ignore();

        switch (pilih) {
            case 1:
                Register(users, jumlahUser);
                break;
            case 2: {
                int index = Login(users, jumlahUser);
                if (index != -1) menu_utama(users[index]);
                break;
            }
            case 3:
                cout << YELLOW << "Anda Keluar dari Program, bye bye :D\n";
                cout << " /\_/\ " << endl;
                cout << "( o.o ) " << endl;
                cout << " > ^ < " << endl << RESET;
                return 0;
            default:
                cout << RED << "Pilihan tidak valid!" << RESET << endl;
        }
    }
}

```

B. Fitur Menu Utama

Fitur ini digunakan untuk pengguna mengakses fitur-fitur yang tersedia di dalam program.

Source Code:

```
void menu_utama(User &user) {
    int pilihan;
    while (true) {
        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;
        cin.ignore();

        switch (pilihan) {
            case 1: tambahKomik(user); break;
            case 2: tampilkanKomik(user, 0); break;
            case 3: ubahKomik(user); break;
            case 4: hapusKomik(user); break;
            case 5:
                cout << YELLOW << "Kembali ke Menu Login." << RESET << endl;
                return;
            default:
                cout << RED << "Pilihan tidak valid!" << RESET << endl;
        }
    }
}
```

C. Fitur Create (Menambahkan Data Komik)

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

Source Code:

```
void tambahKomik(User &user) {
    if (user.jumlahKomik >= MAX_KOMIK) {
        cout << RED << "Koleksi penuh!" << RESET << endl;
        return;
    }
    Komik &k = user.koleksi[user.jumlahKomik];
    cout << "Masukkan judul komik: ";
    getline(cin, k.judul);
    cout << "Masukkan rating komik: ";
    cin >> k.rating;
    cout << "Masukkan tahun terbit komik: ";
    cin >> k.tahun_terbit;
    user.jumlahKomik++;
    cout << CYAN << "Data komik berhasil ditambahkan!" << RESET << endl;
}
```

D. Fitur Read (Melihat Data Komik)

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

Source code:

```
void tampilkanKomik(User &user, int index) {
    if (user.jumlahKomik == 0) {
        cout << YELLOW << "Belum ada komik." << RESET << endl;
        return;
    }
    if (index == user.jumlahKomik) return;
    Komik &k = user.koleksi[index];
    cout << MAGENTA <<
    "+-----+\n";
    cout << "| " << index + 1 << ". Judul: " << k.judul << " | Rating: " <<
    k.rating << " | Tahun terbit: " << k.tahun_terbit << " |\n";
    cout << "+-----+\n"
    << RESET;

    tampilkanKomik(user, index + 1);
}
```

E. Fitur Update (Mengubah Data Komik)

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

Source code:

```
void ubahKomik(User &user) {
    tampilkanKomik(user, 0);
    int index;
    cout << "Masukkan nomor komik yang akan diubah: ";
    cin >> index;
    cin.ignore();
    if (index < 1 || index > user.jumlahKomik) {
        cout << RED << "Nomor komik tidak valid!" << RESET << endl;
        return;
    }
    Komik &k = user.koleksi[index - 1];
    cout << "Masukkan judul komik baru: ";
    getline(cin, k.judul);
    cout << "Masukkan rating komik baru: ";
```

```

    cin >> k.rating;
    cout << "Masukkan tahun tahun komik baru: ";
    cin >> k.tahun_terbit;
    cout << CYAN << "Data komik berhasil diubah!" << RESET << endl;
}

```

F. Fitur Delete (Menghapus Data Komik)

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

Source code:

```

void hapusKomik(User &user) {
    tampilkanKomik(user, 0);
    int index;
    cout << "Masukkan nomor komik yang akan dihapus: ";
    cin >> index;
    if (index < 1 || index > user.jumlahKomik) {
        cout << RED << "Nomor komik tidak valid!" << RESET << endl;
        return;
    }
    for (int i = index - 1; i < user.jumlahKomik - 1; i++) {
        user.koleksi[i] = user.koleksi[i + 1];
    }
    user.jumlahKomik--;
    cout << CYAN << "Data komik berhasil dihapus!" << RESET << endl;
}

```


4. Uji Coba dan Hasil Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> cd "d:\TUGAS KULIAH\praktikum-apl\praktikum-apl\post-test\post-test-apl-4"; if ($?) { g++ 2409106038-TriyaKhairunNisa-PT-4.cpp -o 2409106038-TriyaKhairunNisa-PT-4 } ; if ($?) { .\2409106038-TriyaKhairunNisa-PT-4 }

=====
| Manajemen Komik Webtoon Favorit |
| 1. Register                      |
| 2. Login                        |
| 3. Keluar                       |
|                                 |
=====

PILIH: 1
Masukkan NIM: 038
Masukkan Nama: Triya
Berhasil Registrasi!

=====
| Manajemen Komik Webtoon Favorit |
| 1. Register                      |
| 2. Login                        |
| 3. Keluar                       |
|                                 |
=====

PILIH: 2
Masukkan NIM: 038
Masukkan Nama: Triya
Login berhasil! Hai, 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: WEE!!!
Masukkan rating komik: 9.2
Masukkan tahun terbit komik: 2014
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

```

Gambar 4.1 Output 1A

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
Masukkan judul komik: Solo Leveling
Masukkan rating komik: 8.1
Masukkan tahun terbit komik: 2015
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: 2
+-----+
| 1. Judul: WEE!!! | Rating: 9.2 | Tahun terbit: 2014 |
+-----+
+-----+
| 2. Judul: Solo Leveling | Rating: 8.1 | Tahun terbit: 2015 |
+-----+
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 3
+-----+
| 1. Judul: WEE!!! | Rating: 9.2 | Tahun terbit: 2014 |
+-----+
+-----+
| 2. Judul: Solo Leveling | Rating: 8.1 | Tahun terbit: 2015 |
+-----+
Masukkan nomor komik yang akan diubah: 1
Masukkan judul komik baru: WEE!!!
Masukkan rating komik baru: 10
Masukkan tahun terbit komik baru: 2016
Data komik berhasil diubah!
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik

```

Gambar 4.2 Output 2A

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
5. Keluar
Pilihan: 2
+-----+
| 1. Judul: WEE!!! | Rating: 10 | Tahun terbit: 2016 |
+-----+
+-----+
| 2. Judul: Solo Leveling | Rating: 8.1 | Tahun terbit: 2015 |
+-----+
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 4
+-----+
| 1. Judul: WEE!!! | Rating: 10 | Tahun terbit: 2016 |
+-----+
+-----+
| 2. Judul: Solo Leveling | Rating: 8.1 | Tahun terbit: 2015 |
+-----+
Masukkan nomor komik yang akan dihapus: 1
Data komik berhasil dihapus!
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 2
+-----+
| 1. Judul: Solo Leveling | Rating: 8.1 | Tahun terbit: 2015 |
+-----+
===== Menu Utama =====
1. Tambah Data Komik
2. Lihat Data Komik
3. Ubah Data Komik
4. Hapus Data Komik
5. Keluar
Pilihan: 5
```

Gambar 4.3 Output 3A

```
Logout berhasil!!
=====
| Manajemen Komik Webtoon Favorit |
| 1. Register                      |
| 2. Login                        |
| 3. Keluar                       |
=====
PILIH: 3
Anda Keluar dari Program, bye bye :D
/\_/\
( o.o )
> ^ <
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl\post-test\post-test-apl-4>
```

Gambar 4.4 Output 4A

5. Langkah-Langkah Git pada VSCode

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 D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git init
Reinitialized existing Git repository in D:/TUGAS KULIAH/praktikum-apl/praktikum-apl/.git/
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git add .
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git commit -m "Finish Post Test 4"
[main 3451965] Finish Post Test 4
7 files changed, 230 insertions(+), 74 deletions(-)
delete mode 100644 .vscode/settings.json
create mode 100644 kelas/pertemuan-4.cpp
create mode 100644 kelas/pertemuan-4.exe
delete mode 100644 post-test/post-test-apl-2/tempCodeRunnerFile.cpp
create mode 100644 post-test/post-test-apl-4/2409106038-TriyaKhairunNisa-PT-4.cpp
create mode 100644 post-test/post-test-apl-4/2409106038-TriyaKhairunNisa-PT-4.exe
delete mode 100644 tempCodeRunnerFile.cpp
```

Gambar 5.1 inisiasi dan 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.

c) **git commit -m**

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

```
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git remote add origin https://github.com/3yakhansa/praktikum-apl.git
error: remote origin already exists.
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git branch -M main
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl> git push -u origin main
To https://github.com/3yakhansa/praktikum-apl.git
```

Gambar 5.2 remote and push Git

d) **git remote**

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

e) **git branch**

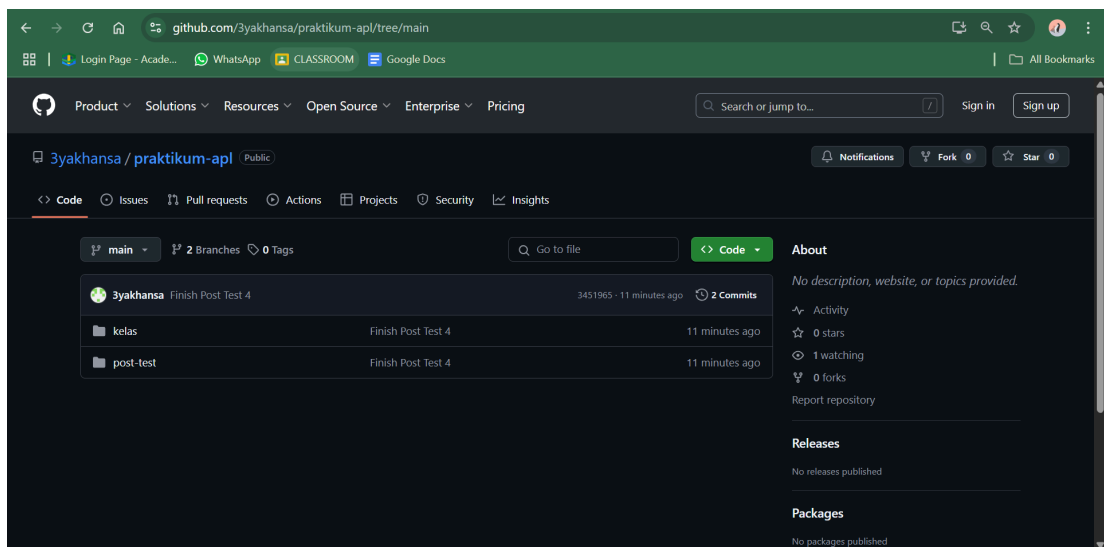
- ❖ git branch -M main adalah untuk mengganti nama branch saat ini menjadi main

f) **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.

```
Enumerating objects: 38, done.
Counting objects: 100% (38/38), done.
Delta compression using up to 8 threads
Compressing objects: 100% (37/37), done.
Writing objects: 100% (38/38), 2.95 MiB | 1.50 MiB/s, done.
Total 38 (delta 13), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (13/13), done.
To https://github.com/3yakhansa/praktikum-apl.git
+ e2add49...3451965 main -> main (forced update)
PS D:\TUGAS KULIAH\praktikum-apl\praktikum-apl>
```

Gambar 5.3 Hasil Git



Gambar 5.5 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.