

**POSTTEST 1**

**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**

**Nama Gilbert Finsen Zakaria Lian (2409106110)**

**Kelas C'2**

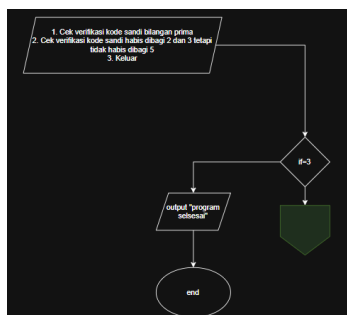
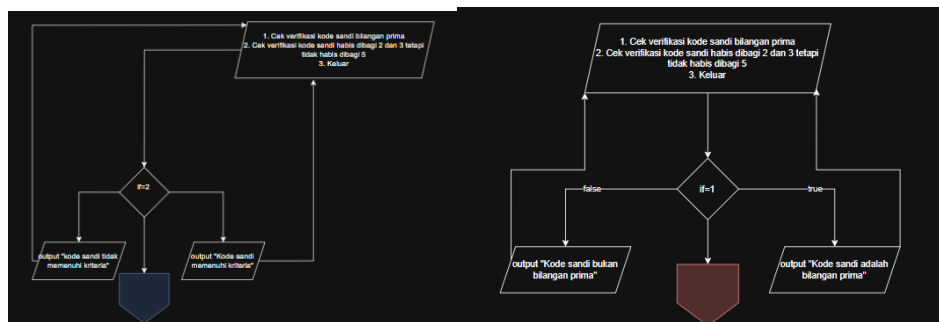
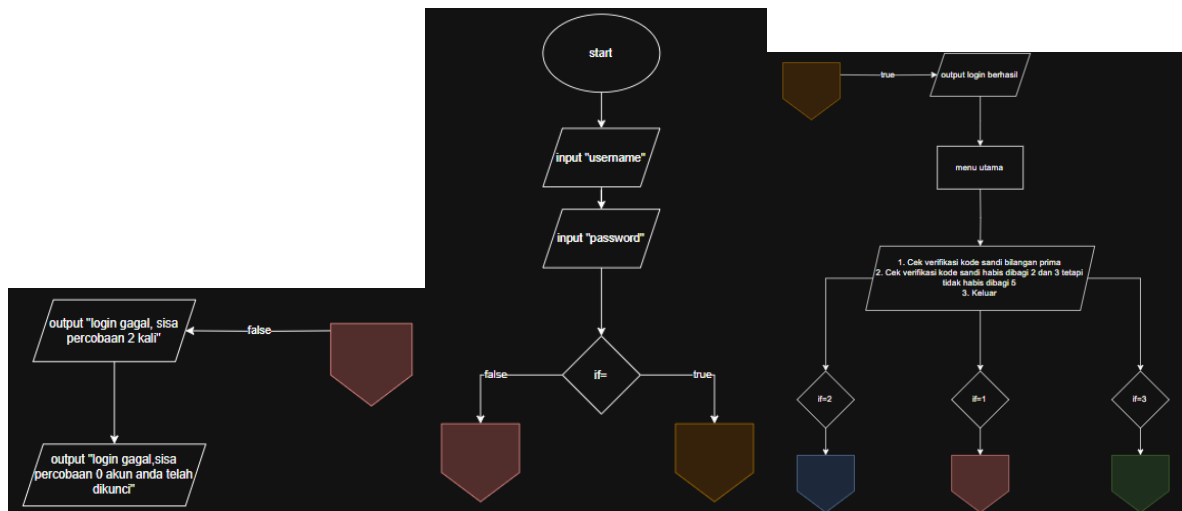
**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

# 1. Flowchart



## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini memiliki fitur **login** dengan username dan password yang telah ditentukan. Setelah berhasil masuk, pengguna dapat memasukkan **passcode** yang harus berada dalam rentang **101 hingga 99999**. Program kemudian menyediakan **menu pilihan** untuk:

1. **Memeriksa apakah passcode adalah bilangan prima.**
2. **Memeriksa apakah passcode habis dibagi 2 dan 3 tetapi tidak 5.**
3. **Keluar dari program.**

### 2.2 Penjelasan Alur & Algoritma

1. Mulai
2. Login

jika gagal 3 kali → Program berhenti

Jika berhasil → Lanjut ke menu

3. Menu ditampilkan

1. Cek bilangan prima

2. Cek divisibilitas (2 & 3 tapi bukan 5)
  3. Keluar dari program
  4. User memasukkan passcode
- Jika passcode tidak valid Minta input ulang
5. Hasil ditampilkan sesuai pilihan
  6. Program kembali ke menu hingga user memilih keluar
  7. Selesai

## Algoritma

### 1. Mulai Program

1. Panggil fungsi login() untuk otentikasi pengguna.

### 2. Login Sistem

3. Set variabel username dan password yang valid.
4. Set batas maksimal percobaan login (3 kali).
5. Looping selama percobaan login masih ada:
  - o Minta input username dan password dari pengguna.
  - o Jika benar, tampilkan "Login Berhasil" dan lanjut ke menu.
  - o Jika salah, kurangi jumlah percobaan dan beri pesan "Login gagal! Sisa percobaan: 2".
6. Jika gagal login setelah 3 kali percobaan, tampilkan pesan "Akses dikunci" dan akhiri program.

### 3. Menampilkan Menu Utama

7. Jika login berhasil, tampilkan menu utama:
7. **Minta pengguna memilih opsi menu.**

### 4. Proses Pemilihan Menu

9. Jika pengguna memilih "3", tampilkan "Terima kasih telah menggunakan program" dan keluar.
10. Minta input passcode dari pengguna.
11. Periksa apakah passcode berada dalam rentang 101 - 99999:
  - Jika tidak valid, tampilkan pesan "Passcode tidak valid!" dan kembali ke menu utama.
  - Jika valid, lanjut ke langkah berikutnya sesuai pilihan menu.

## 5.Mengecek Bilangan Prima (Jika Pilih 1)

12. Gunakan fungsi isPrime(passcode):

- Jika passcode adalah bilangan prima, tampilkan "Passcode adalah bilangan prima."
- Jika bukan bilangan prima, tampilkan "Passcode bukan bilangan prima."

13. Kembali ke menu utama.

## 6.Mengecek Bilangan Habis Dibagi 2 dan 3 Tetapi Tidak 5 (Jika Pilih 2)

14. Gunakan fungsi isDivisibleBy2And3ButNot5(passcode):

- Jika passcode memenuhi syarat, tampilkan "Passcode memenuhi syarat."
- Jika tidak, tampilkan "Passcode tidak memenuhi syarat."

15. Kembali ke menu utama.

## 7.Program Berjalan Secara Berulang

16. Ulangi proses dari langkah 7 sampai pengguna memilih keluar.

17. Program selesai.

## 3. Source Code

Menu Login Sistem

user harus memasukkan username dan password yang benar untuk bisa masuk ke program.

Jika salah, user diberi maksimal 3 kali percobaan sebelum program mengunci akses.

```
int main() {  
    string username = "Gilbert Finsen Zakaria Lian";
```

```

string password = "2409106110";
string input_user, input_pass;
int attempts = 3;

while (attempts > 0) {
    cout << "Masukkan Username: ";
    getline(cin, input_user);
    cout << "Masukkan Password: ";
    cin >> input_pass;
    cin.ignore();

    if (input_user == username && input_pass == password) {
        cout << "Login berhasil!\n\n";
        break;
    } else {
        attempts--;
        cout << "Login gagal! Sisa percobaan: " << attempts <<
"\n\n";
    }
}

if (attempts == 0) {
    cout << "Anda telah dikunci.\n";
    return 0;
}

```

**Gambar 3.1**

**Catatan:**

Jika login berhasil, program lanjut ke menu utama.

Jika login gagal 3 kali berturut-turut, program berhenti.

**B. Masukan kode sandi tidak kurang dari 100 dan lebih dari 99999**

```

bool is_valid_passcode(int n) {
    return n > 100 && n <= 99999;
}

```

**Gambar 3.2**

### C. Verifikasi kode sandi apakah bilangan prima atau bukan

```
bool is_prime(int n) {  
    if (n < 2) return false;  
    for (int i = 2; i * i <= n; i++) {  
        if (n % i == 0) return false;  
    }  
    return true;  
}
```

Gambar 3.3

### D. Cek Verifikasi kode sandi (habis di bagi 2 dan 3, tapi tidak boleh oleh 5

```
bool is_divisible(int n) {  
    return n % 2 == 0 && n % 3 == 0 && n % 5 != 0;  
}
```

Gambar 3.4

### E.Menu keluar program

```
if (choice == 3) {  
    cout << "Program selesai.\n";  
    break;  
}
```

Gambar 3.

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Skenario 1 = Nama saya Gilbert Finsen Zakaria Lian dengan Nim 2409106110 saya akan masuk ke program dan ternyata berhasil

Skenario 2 = Saya ingin masuk dengan menggunakan nama gilbert dengan Nim 2409106110 dan ternyata tidak bisa masuk

Skenario 3 = saya coba lagi dengan nama Lian tapi dengan Nim 110 yang berbeda namun tetap gagal, sampai 3 kali dan saya pun terkunci dan keluar dari program.

### 4.1 Hasil Output

Hasil nya jika login berhasil

```
Masukkan Username: Gilbert Finsen Zakaria Lian
Masukkan Password: 2409106110
Login berhasil!
```

**Gambar 4.1**

Hasil nya jika login gagal

```
Masukkan Username: gilbert
Masukkan Password: 2409106110
Login gagal! Sisa percobaan: 2

Masukkan Username: lian
Masukkan Password: 110
Login gagal! Sisa percobaan: 1

Masukkan Username: lian
Masukkan Password: 110
Login gagal! Sisa percobaan: 0

Anda telah dikunci.

-----
Process exited after 38.5 seconds with return value 0
Press any key to continue . . .
```

**Gambar 4.2**

### 4.3 Uji Coba

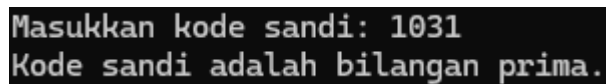


Skenario 1 = Saya mencoba memasukan kode sandi 1031 dan saya ingin ngecek apakah kode sandi tersenbut adalah bilangan prima, dan ternyata sandi tersebut adalah bilangan prima

Skenario ke 2 = Saya mencoba menggunakan kode sandi pertama yaitu 2025, dan saya ingin ngecek apakah sandi tersebut adalah bilangan prima, ternyata sandi tersebut bukan bilangan prima

## 4.2 Hasil Output

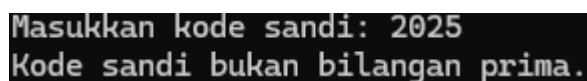
Dengan menggunakan kode sandi 1031



```
Masukkan kode sandi: 1031
Kode sandi adalah bilangan prima.
```

**Gambar 4.3**

Dengan menggunakan kode sandi 2025



```
Masukkan kode sandi: 2025
Kode sandi bukan bilangan prima.
```

**Gambar 4.4**

## 4. Uji Coba dan Hasil Output

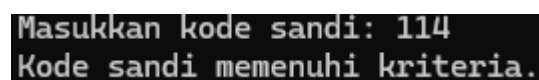
### 4.1 Uji Coba

Skenario ke 1 = Saya ingin mengecek menggunakan kode sandi 114 apakah sandi tersebut memenuhi kriteria, dan ternyata sandi tersebut memenuhi kriteria

Skenario ke 2 = Saya ingin mengecek menggunakan kode sebelumnya 110 apakah kode sandi tersebut memenuhi kriteria, ternyata sandi tersebut tidak memenuhi kriteria.

## 4.2 Hasil Output

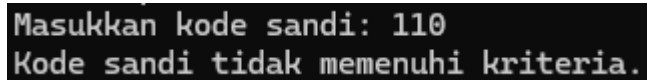
Jika menggunakan kode 114



```
Masukkan kode sandi: 114
Kode sandi memenuhi kriteria.
```

**Gambar 4.7**

Jika menggunakan kode sandi 110



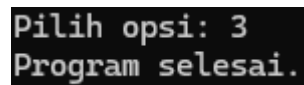
```
Masukkan kode sandi: 110
Kode sandi tidak memenuhi kriteria.
```

**Gambar 4.8**

#### 4.1 Uji Coba

Skenario 1 = saya ingin memberhentikan program dan memilih menu 3 dan program pun otomatis berhenti

#### 4.2 Hasil Output



```
Pilih opsi: 3
Program selesai.
```

### **Gambar 4.9 Keluar dari program**

*(Sertakan tangkapan layar atau hasil output dari program setelah dijalankan.)*

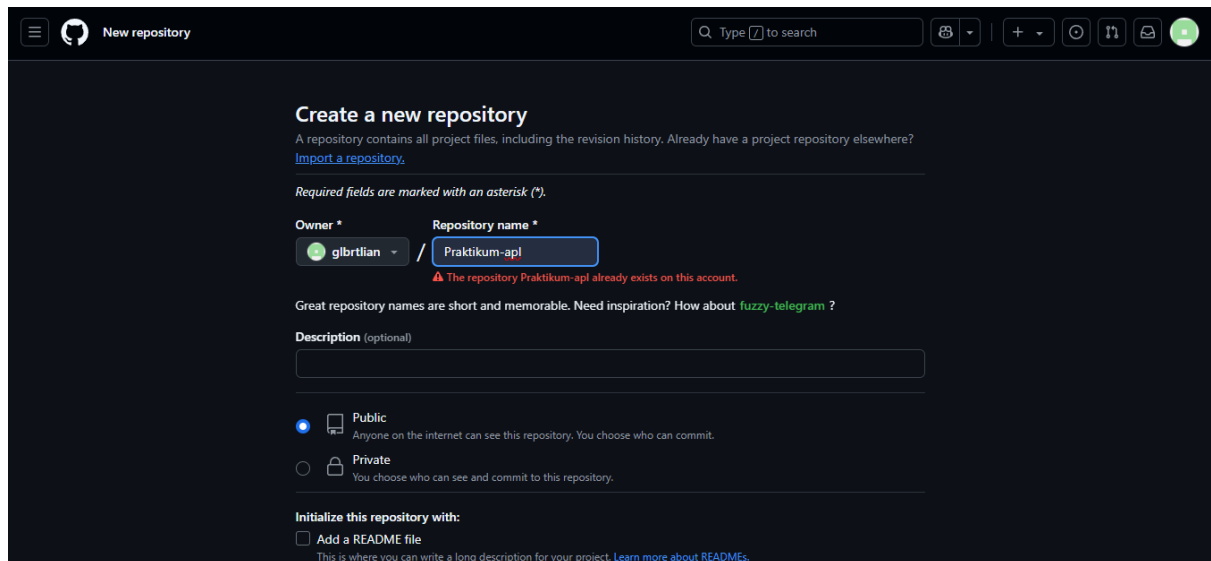
Gambar 4.1 Contoh

Gambar <nomor bab>.[index] <judul-gambar>

## 5. Sistem Github

### 1. Membuat Repository

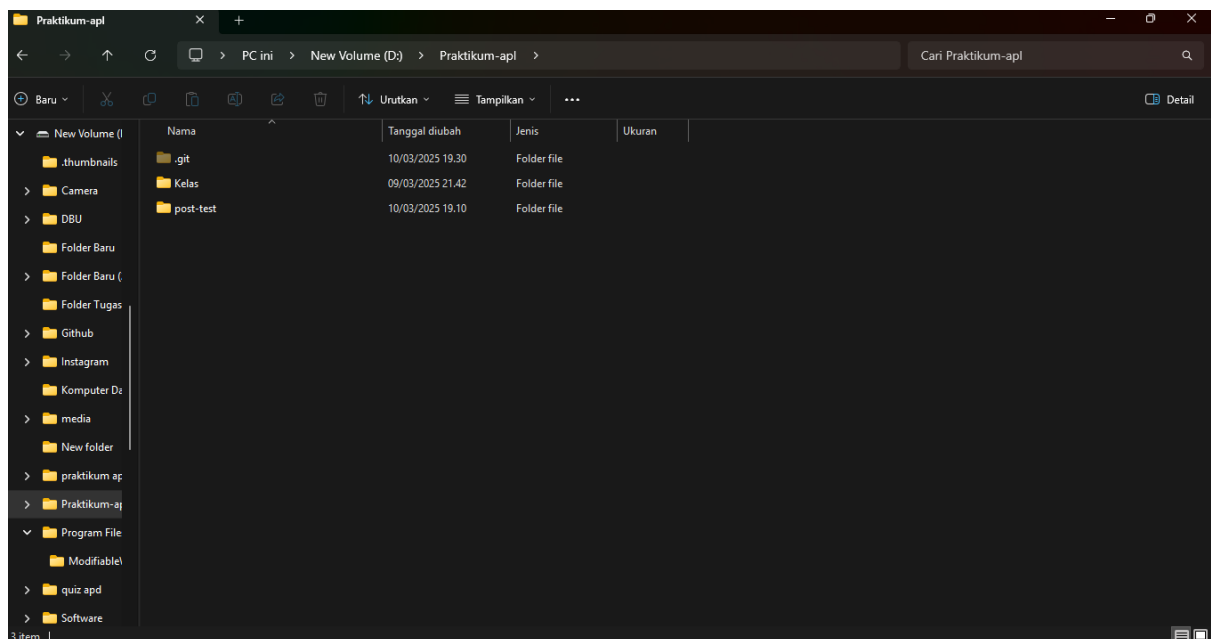
Membuat repository baru untuk menyimpan folder



Gambar 5.1 Respository

### 2. Membuat Folder

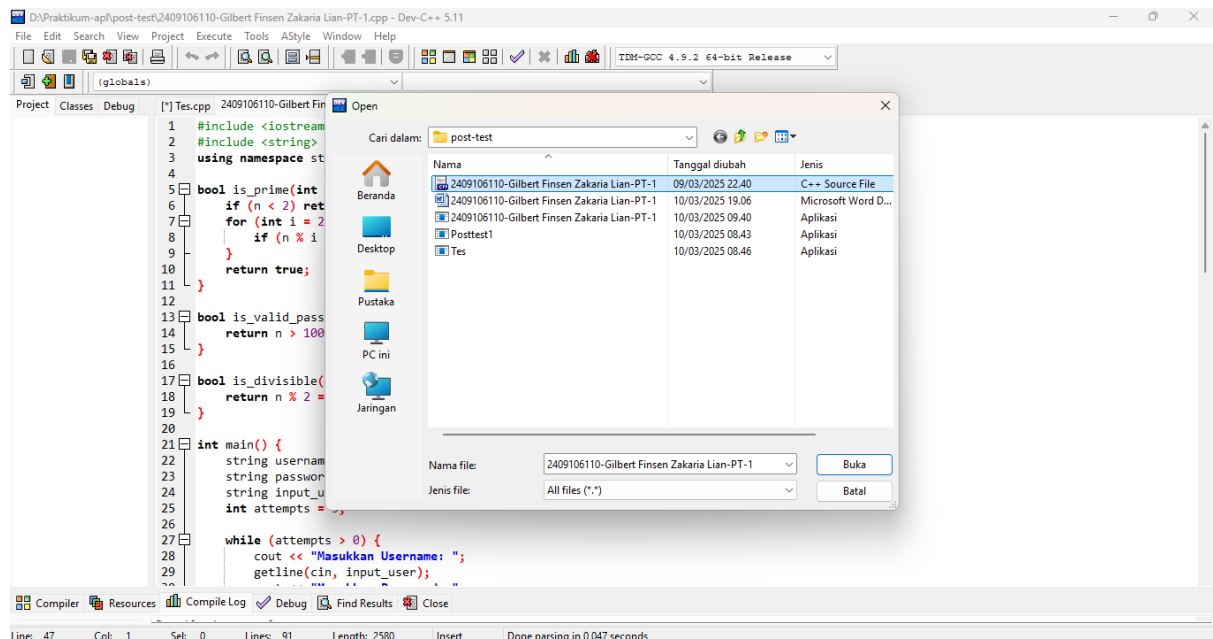
Buat folder eksplor dengan nama praktikum-apl yang sesuai dengan repository github, dan di dalam folder buat 2 folder lagi dengan nama kelas dan posttest



Gambar 5.2 Folder

### 3. Membuka

Buka folder yang sudah di buat di Dev C++



**Gambar 5.3 Vscode**

### 4. Langkah Langkah Git

Pertama klik kanan pada folder lalu pilih show open witch lalu pilih open git bash

Kedua ketika sudah di terminal ketik git config –global user.email”sesuaikan dengan email yang ada di akun github”

Ketiga git init untuk menginisiasi repository git.

Keempat ketik git add untuk menambah file yang akan di komit dan git comit –m

“Update” untuk membuat check point

Kelima ketik git remove untuk menghubungkan repository yang ada dilokal computer dengan cara mengcopy link yang ada di github

```

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl
$ git config --global user.email "glian6020@gmail.com"

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl
$ git init
Initialized empty Git repository in D:/Praktikum-apl/.git/

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (master)
$ git add .

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (master)
$ git commit -m "Gilbert Ganz"
[master (root-commit) c82a5f4] Gilbert Ganz
7 files changed, 183 insertions(+)
create mode 100644 post-test/2409106110-Gilbert Finsen Zakaria Lian-PT-1.cpp
create mode 100644 post-test/2409106110-Gilbert Finsen Zakaria Lian-PT-1.exe
create mode 100644 post-test/Gilbert Lian - Template Laporan APL (2).docx
create mode 100644 post-test/Posttest1.exe
create mode 100644 post-test/Tes.exe
create mode 100644 post-test/~$lbert Lian - Template Laporan APL (2).docx
create mode 100644 post-test/~WRL3056.tmp

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (master)
$ git branch -M main

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (main)
$ git remote add origin https://github.com/glbtrlan/Praktikum-apl.git

```

**Gambar 5.4 Langkah Langkah Git**

## 5. Git push

Melakukan gitpush untuk mengupload semua yang ada di repository

```

ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 834.55 KiB | 4.54 MiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/glbtrlan/Praktikum-apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

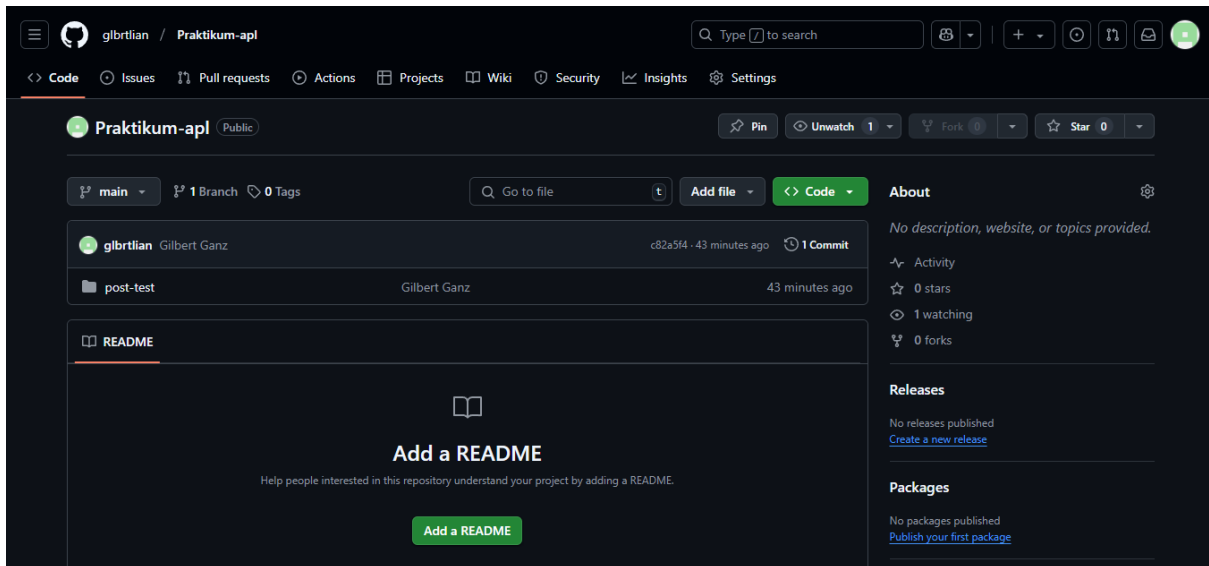
ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl (main)
$

```

**Gambar 5.5 GitPush**

## 6. Reload Github

Reload github kalian pada browser atau google choreme kalian



Gambar 5.6 Reload