

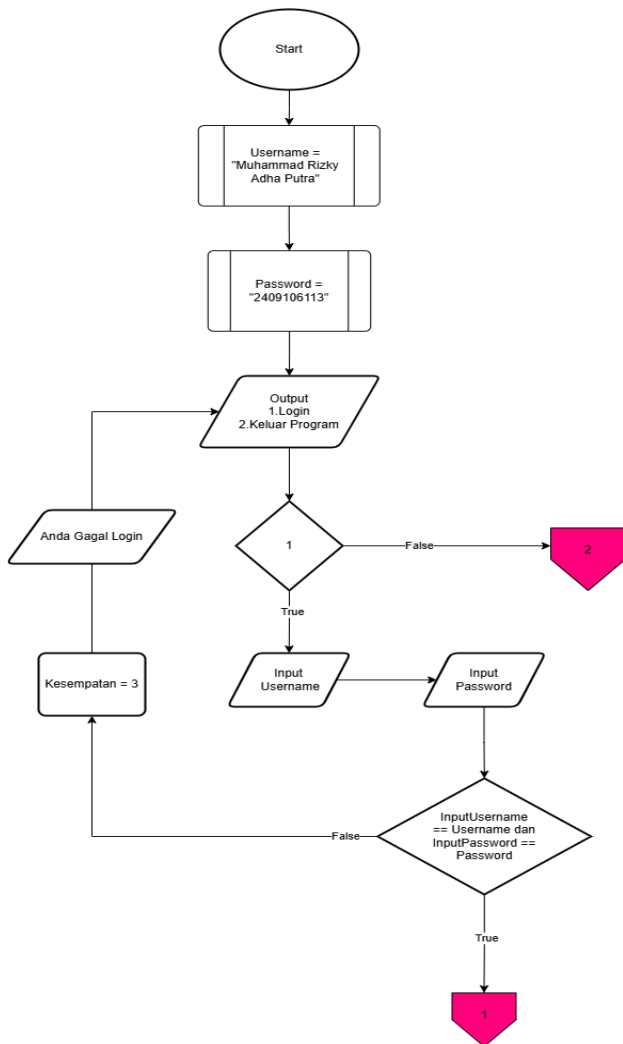
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



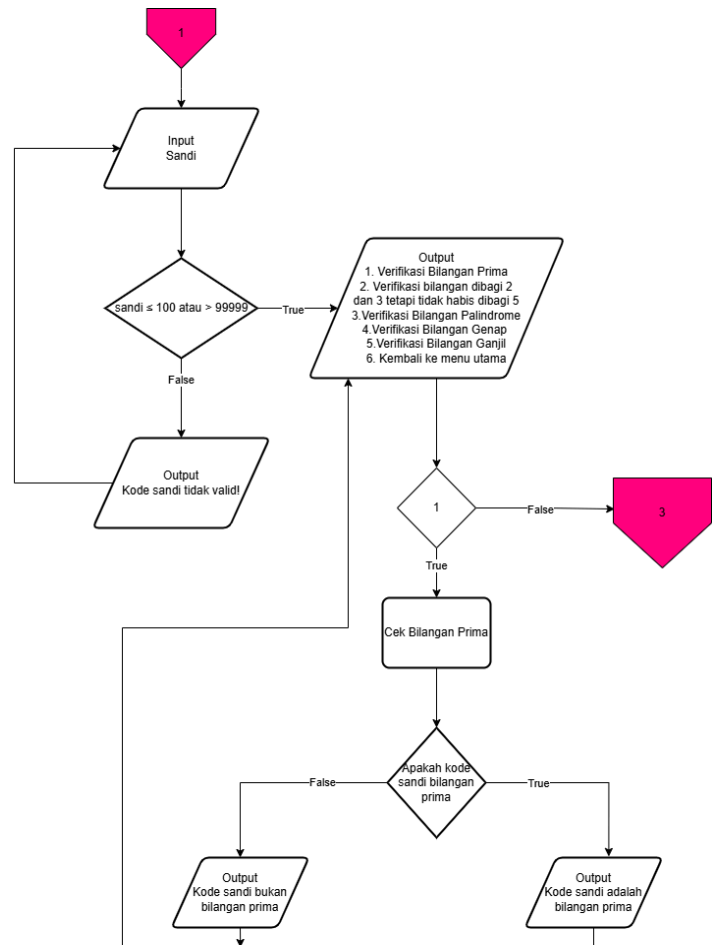
Disusun oleh:
Muhammad Rizky Adha Putra (2409106113)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

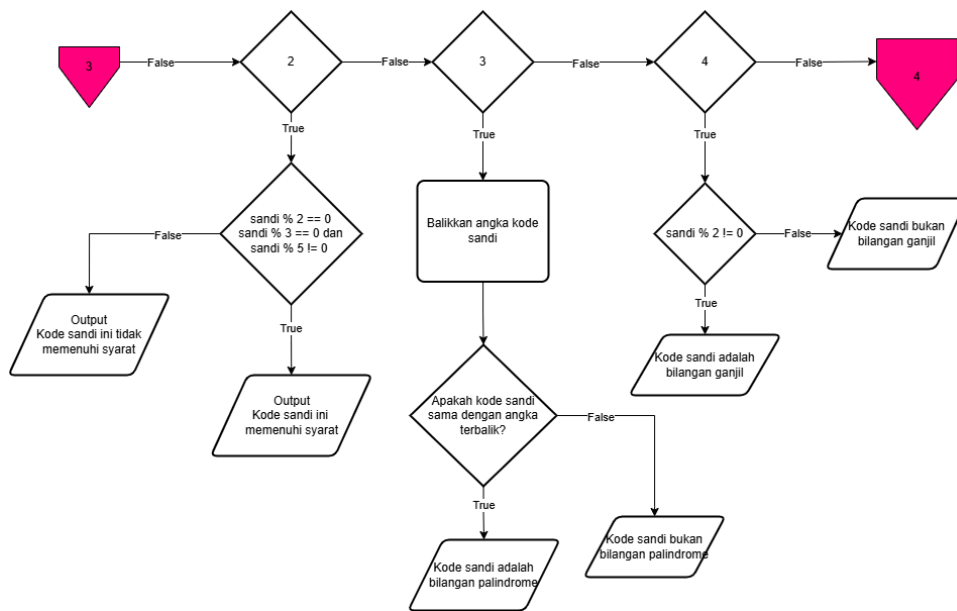
1. Flowchart



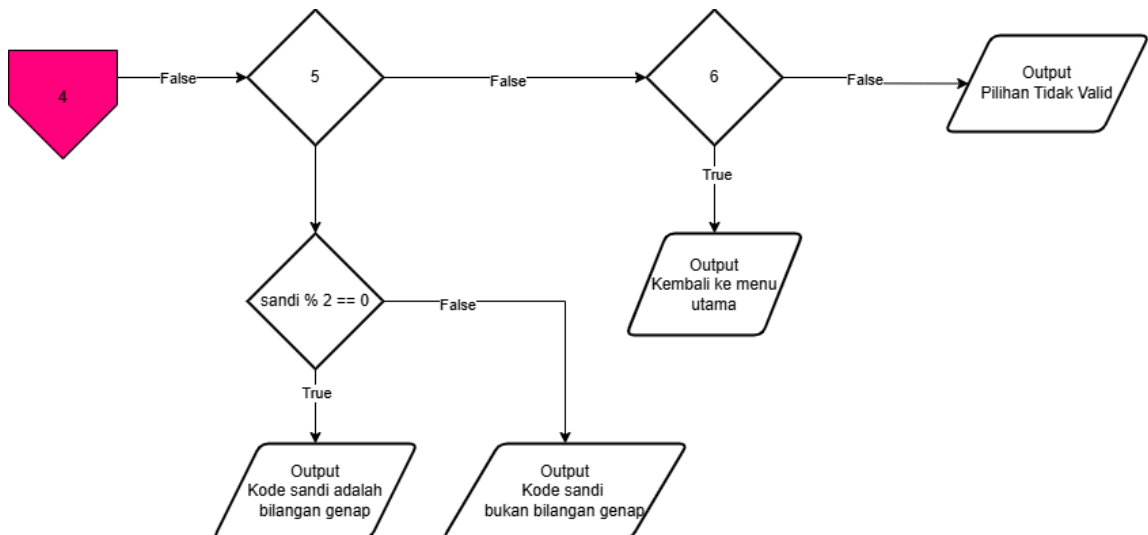
Gambar 1.1
Gambar Flowchart



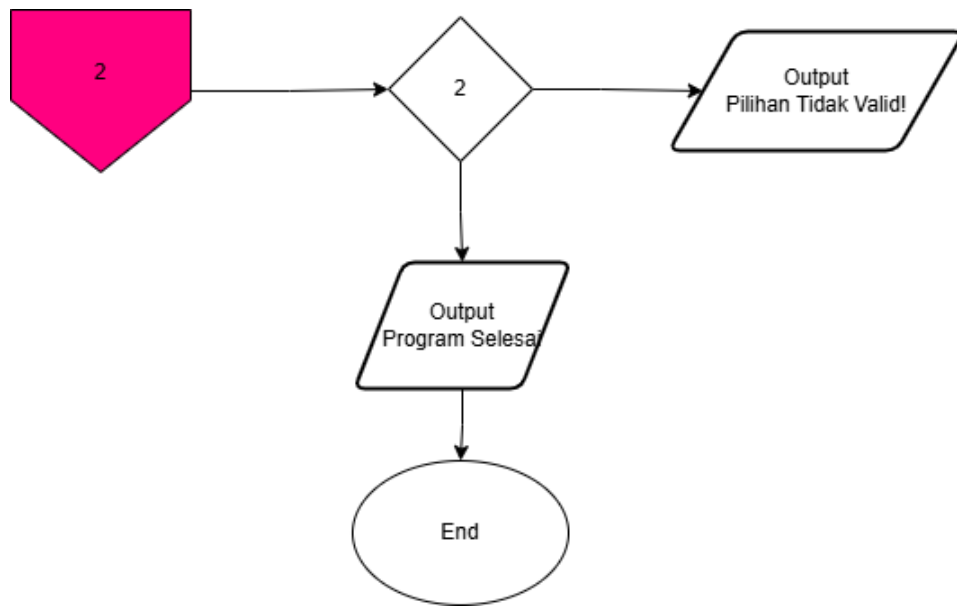
Gambar 1.2
Gambar Flowchart



Gambar 1.3
Gambar Flowchart



Gambar 1.4
Gambar Flowchart



Gambar 1.5
Gambar Flowchart

2. Analisis Program

2.1 Deskripsi Singkat Program

Pada post test 1 ini melakukan verifikasi kode sandi dengan berbagai fitur pengecekan atau verifikasi. Program ini memiliki fitur login dengan username dan password yang dirancang khusus oleh penggunanya, serta validasi kode sandi sebelum pengguna dapat mengecek properti numerik dari kode sandi tersebut. Beberapa fitur verifikasi kode sandi mencakup pengecekan apakah angka tersebut merupakan bilangan prima, bilangan palindrome, bilangan ganjil, bilangan genap, atau memenuhi aturan tertentu dalam pembagian.

2.2 Penjelasan Alur & Algoritma

1. Menampilkan Menu Utama

- Program menampilkan dua pilihan: Login dan Keluar.
- Jika pengguna memilih Login, program meminta username dan password.
- Jika pengguna memilih Keluar, program akan berhenti.

2. Proses Login

- Program memeriksa apakah username dan password yang dimasukkan cocok dengan kredensial yang telah ditentukan sesuai dengan nama pembuat
- Pengguna diberikan 3 kali kesempatan untuk memasukkan data yang benar.
- Jika gagal login setelah 3 kali, program akan kembali ke menu utama.

3. Validasi Kode Sandi

- Setelah login berhasil, pengguna diminta memasukkan kode sandi yang harus lebih dari 100 dan tidak boleh lebih dari 99999.
- Jika kode sandi tidak memenuhi syarat, program meminta pengguna untuk memasukkan ulang kode sandi hingga valid.

4. Menu Verifikasi Kode Sandi

1. Setelah kode sandi divalidasi, program menampilkan menu verifikasi dengan pilihan berikut:
 1. Cek apakah kode sandi adalah bilangan prima
 2. Cek apakah kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5
 3. Cek apakah kode sandi adalah bilangan palindrome
 4. Cek apakah kode sandi bilangan ganjil
 5. Cek apakah kode sandi bilangan genap
 6. Kembali ke menu utama

5. Implementasi Algoritma di Setiap Pilihan Menu Verifikasi

1. Bilangan Prima: Menggunakan fungsi yang dimana akan memeriksa apakah suatu angka hanya bisa dibagi oleh 1 dan dirinya sendiri.
2. Habis Dibagi 2 dan 3 tetapi Tidak 5: Mengecek apakah angka tersebut habis dibagi 2 dan 3 sekaligus, tetapi tidak habis dibagi 5
3. Bilangan Palindrome: Menggunakan fungsi yang akan membalik angka dan membandingkannya dengan angka aslinya untuk melihat apakah sama.
4. Bilangan Ganjil: Mengecek apakah angka tersebut tidak habis dibagi 2.
5. Bilangan Genap: Mengecek apakah angka tersebut habis dibagi 2.

6. Navigasi dalam Program

- Setelah melakukan verifikasi, pengguna dapat memilih untuk kembali ke menu utama atau melakukan pengecekan ulang pada kode sandi.
- Jika memilih opsi 6 (Kembali ke menu utama), program akan kembali menampilkan menu utama.

3. Source Code

A. Memeriksa Bilangan Prima

Source Code:

```
// Fungsi untuk memeriksa apakah bilangan prima
bool BilanganPrima(int bp) {
    if (bp <= 1) return false;
    for (int i = 2; i * i <= bp; i++) {
        if (bp % i == 0) return false;
    }
    return true;
}
```

B. Memeriksa Bilangan Palindrome

Source Code:

```
// Fungsi untuk memeriksa apakah bilangan palindrome
bool BilanganPalindrome(int angka) {
    if (angka < 0) return false;

    int BlganAseli = angka, balikan = 0;
    while (angka > 0) {
        int digit = angka % 10;
        balikan = balikan * 10 + digit;
        angka /= 10;
    }
    return BlganAseli == balikan;
}
```

C. Fitur Login Username dan Password Khusus

Source Code:

```
// Fungsi untuk memeriksa Login
bool login(string inputUsername, string inputPassword) {
    string Username = "Muhammad Rizky Adha Putra";
    string Password = "2409106113";

    return (inputUsername == Username && inputPassword == Password);
}
```

```
}
```

D. Mulai Program

```
int main() {  
    int menuUtama;  
    string username, password;  
    int sandi;
```

E. Perulangan 3 kali Login

```
    if (menuUtama == 1) {  
        int kesempatan = 3;  
  
        while (kesempatan > 0) {  
            cout << "== MENU LOGIN ==" << endl;  
            cout << "Masukkan Username: ";  
            getline(cin, username);  
  
            cout << "Masukkan Password: ";  
            getline(cin, password);  
  
            if (login(username, password)) {  
                cout << "Login berhasil! Selamat datang, " << username  
                << "!" << endl;  
                break;  
            } else {  
                kesempatan--;  
                cout << "Login gagal! Kesempatan tersisa: " <<  
                kesempatan << " kali." << endl;  
                if (kesempatan == 0) {  
                    cout << "Anda gagal login 3 kali. Kembali ke menu  
utama." << endl;  
                }  
            }  
        }  
    }  
}
```


F. Validasi Kode Sandi

Source Code:

```
do {  
    cout << "Masukkan kode sandi (harus lebih dari 100 dan tidak boleh lebih  
dari 99999): ";  
    cin >> sandi;  
  
    if (sandi <= 100 || sandi > 99999) {  
        cout << "Kode sandi tidak valid! Harus lebih dari 100  
dan tidak boleh lebih dari 99999." << endl;  
    }  
} while (sandi <= 100 || sandi > 99999);
```

G. Menu Verifikasi

Source Code:

```
int pilihan;

while (true) {

    cout << "== Menu Program Verifikasi Password ==" << endl;

    cout << "Kode sandi yang Anda masukkan: " << sandi << endl;

    cout << "1. Cek kode sandi bilangan prima" << endl;

    cout << "2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak  
habis dibagi 5" << endl;

    cout << "3. Cek kode sandi bilangan palindrome" << endl;

    cout << "4. Cek kode sandi bilangan ganjil" << endl;

    cout << "5. Cek kode sandi bilangan genap" << endl;

    cout << "6. Kembali ke menu utama" << endl;

    cout << "Pilih menu: ";

    cin >> pilihan;
```

```
    cout << "Kode sandi: " << sandi << endl;

    switch (pilihan) {
        case 1:
            cout << (BilanganPrima(sandi) ? "Kode  
sandi adalah bilangan prima." : "Kode sandi bukan bilangan  
prima.") << endl;
            break;
        case 2:
            cout << ((sandi % 2 == 0 && sandi % 3 == 0  
&& sandi % 5 != 0) ? "Kode sandi memenuhi syarat." : "Kode sandi  
tidak memenuhi syarat.") << endl;
            break;
        case 3:
```

```

        cout << (BilanganPalindrome(sandi) ? "Kode
sandi adalah bilangan palindrome." : "Kode sandi bukan bilangan
palindrome.") << endl;
        break;
    case 4:
        cout << ((sandi % 2 != 0) ? "Kode sandi
adalah bilangan ganjil." : "Kode sandi bukan bilangan ganjil.") <<
endl;
        break;
    case 5:
        cout << ((sandi % 2 == 0) ? "Kode sandi
adalah bilangan genap." : "Kode sandi bukan bilangan genap.") <<
endl;
        break;
    case 6:
        cout << "Anda telah kembali ke menu utama"
<< endl;
        goto menu_utama;
    default:
        cout << "Pilihan tidak valid! Silakan coba
lagi." << endl;
        break;
    }
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Skenario 1: Login Berhasil

Input:

- Username: Muhammad Rizky Adha Putra
- Password: 2409106113

Output: "Login berhasil! Selamat datang..."

Skenario 2: Login Gagal (Salah Password 3 Kali)

Input:

- Username: Muhammad Rizky Adha Putra
- Password: pass_salah (diulang 3 kali)

Output: "Login gagal! Kesempatan tersisa: X kali."

Skenario 3: Input Kode Sandi Tidak Valid

- **Input:** 50 (Kurang dari 100) atau 150000 (Lebih dari 99999)
- **Output:** "Kode sandi tidak valid! Harus lebih dari 100 dan tidak boleh lebih dari 99999."

Skenario 4: Verifikasi Kode Sandi

- **Input Kode Sandi:** 131
- **Pengecekan:**
 1. Bilangan prima → **131 adalah bilangan prima**
 2. Habis dibagi 2 & 3 tetapi tidak 5 → **Tidak memenuhi syarat**
 3. Bilangan palindrome → **131 adalah palindrome**
 4. Bilangan ganjil → **131 adalah bilangan ganjil**
 5. Bilangan genap → **Bukan bilangan genap**
- **Output:** Program memberikan hasil sesuai kriteria di atas.

4.2 Hasil Output

1. Login Berhasil

```
== MENU UTAMA ==  
1. Login  
2. Keluar  
Pilih menu: 1  
== MENU LOGIN ==  
Masukkan Username: Muhammad Rizky Adha Putra  
Masukkan Password: 2409106113  
Login berhasil! Selamat datang, Muhammad Rizky Adha Putra!
```

Gambar 4.1 Proses Login Berhasil

2. Login Gagal

```
== MENU LOGIN ==  
Masukkan Username: siapa  
Masukkan Password: 1233  
Login gagal! Kesempatan tersisa: 2 kali.  
== MENU LOGIN ==  
Masukkan Username: apa  
Masukkan Password: 12234  
Login gagal! Kesempatan tersisa: 1 kali.  
== MENU LOGIN ==  
Masukkan Username: kok  
Masukkan Password: 11245  
Login gagal! Kesempatan tersisa: 0 kali.  
Anda gagal login 3 kali. Kembali ke menu utama.
```

Gambar 4.2 Proses Login Berhasil

3. Input Kode Sandi Tidak Valid

```
Masukkan kode sandi (harus lebih dari 100 dan tidak boleh lebih dari 99999): 50  
Kode sandi tidak valid! Harus lebih dari 100 dan tidak boleh lebih dari 99999.
```

Gambar 4.3 Proses Kode Sandi Tidak Valid

4. Verifikasi Kode Sandi

```
Login berhasil! Selamat datang, Muhammad Rizky Adha Putra!  
Masukkan kode sandi (harus lebih dari 100 dan tidak boleh lebih dari 99999): 131  
== Menu Program Verifikasi Password ==  
Kode sandi yang Anda masukkan: 131  
1. Cek kode sandi bilangan prima  
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5  
3. Cek kode sandi bilangan palindrome  
4. Cek kode sandi bilangan ganjil  
5. Cek kode sandi bilangan genap  
6. Kembali ke menu utama
```

Gambar 4.4 Proses Verifikasi Kode Sandi

```
== Menu Program Verifikasi Password ==  
Kode sandi yang Anda masukkan: 131  
1. Cek kode sandi bilangan prima  
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5  
3. Cek kode sandi bilangan palindrome  
4. Cek kode sandi bilangan ganjil  
5. Cek kode sandi bilangan genap  
6. Kembali ke menu utama  
Pilih menu: 1  
Kode sandi: 131  
Kode sandi adalah bilangan prima.
```

Gambar 4.4 Proses Pengecekan Kode Sandi Prima

```
== Menu Program Verifikasi Password ==  
Kode sandi yang Anda masukkan: 131  
1. Cek kode sandi bilangan prima  
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5  
3. Cek kode sandi bilangan palindrome  
4. Cek kode sandi bilangan ganjil  
5. Cek kode sandi bilangan genap  
6. Kembali ke menu utama  
Pilih menu: 2  
Kode sandi: 131  
Kode sandi tidak memenuhi syarat.
```

Gambar 4.5

Proses Pengecekan Kode Sandi dibagi 2 dan 3 tetapi tidak habis dibagi 5

```
== Menu Program Verifikasi Password ==
Kode sandi yang Anda masukkan: 131
1. Cek kode sandi bilangan prima
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5
3. Cek kode sandi bilangan palindrome
4. Cek kode sandi bilangan ganjil
5. Cek kode sandi bilangan genap
6. Kembali ke menu utama
Pilih menu: 3
Kode sandi: 131
Kode sandi adalah bilangan palindrome.
```

Gambar 4.4
Proses Pengecekan Kode Sandi Palindrome

```
== Menu Program Verifikasi Password ==
Kode sandi yang Anda masukkan: 131
1. Cek kode sandi bilangan prima
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5
3. Cek kode sandi bilangan palindrome
4. Cek kode sandi bilangan ganjil
5. Cek kode sandi bilangan genap
6. Kembali ke menu utama
Pilih menu: 4
Kode sandi: 131
Kode sandi adalah bilangan ganjil.
```

Gambar 4.5
Proses Pengecekan Kode Sandi Ganjil

```
== Menu Program Verifikasi Password ==
Kode sandi yang Anda masukkan: 131
1. Cek kode sandi bilangan prima
2. Cek kode sandi habis dibagi 2 dan 3 tetapi tidak habis dibagi 5
3. Cek kode sandi bilangan palindrome
4. Cek kode sandi bilangan ganjil
5. Cek kode sandi bilangan genap
6. Kembali ke menu utama
Pilih menu: 5
Kode sandi: 131
Kode sandi bukan bilangan genap.
```

Gambar 4.6
Proses Pengecekan Kode Sandi Genap

5. Git

```
MINGW64/C:/GITHUB/Praktik x + v
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl
$ git config --global user.email "rizky75338@gmail.com"
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl
$ git init
Initialized empty Git repository in C:/GITHUB/Praktikum-Apl/.git/
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git add .
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git remote add origin https://github.com/113Rizky/praktikum-apl.git
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git commit -m "Posttest rizky"
[main (root-commit) 877ae64] Posttest rizky
 2 files changed, 131 insertions(+)
 create mode 100644 Post-test/Post-test-1/2409106113-MuhammadRizkyAdhaPutra-PT-1.cpp
 create mode 100644 Post-test/Post-test-1/2409106113-MuhammadRizkyAdhaPutra-PT-1.exe
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git branch -M main
Asus@LAPTOP-F4V7N70I MINGW64 /c/GITHUB/Praktikum-Apl (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 2 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 22.39 KiB | 3.20 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
```

1. Mengatur Email Global untuk Git

\$ git config --global user.email "rizky75338@gmail.com"

- Perintah ini digunakan untuk mengatur email global Git, yang akan digunakan dalam

2. **git init** → Inisialisasi repository lokal.
3. **git add .** → Menambahkan semua file ke dalam staging area.
4. **git remote add origin <https://github.com/113Rizky/praktikum-apl.git>** → Menghubungkan repository lokal dengan GitHub.
5. **git commit -m "Posttest rizky"** → Menyimpan perubahan dengan pesan commit.
6. **git branch -M main** → Menentukan branch utama sebagai main.
7. **git push -u origin main** → Mengunggah kode ke GitHub.