

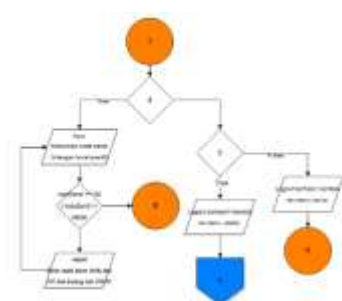
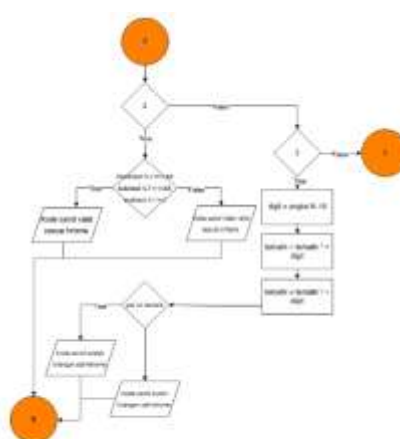
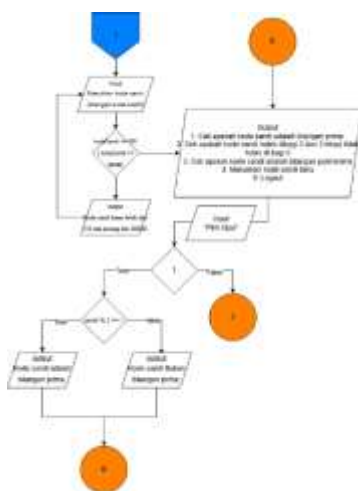
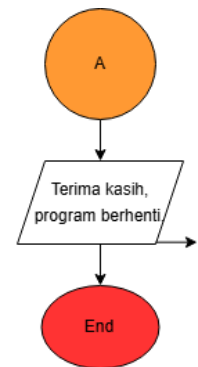
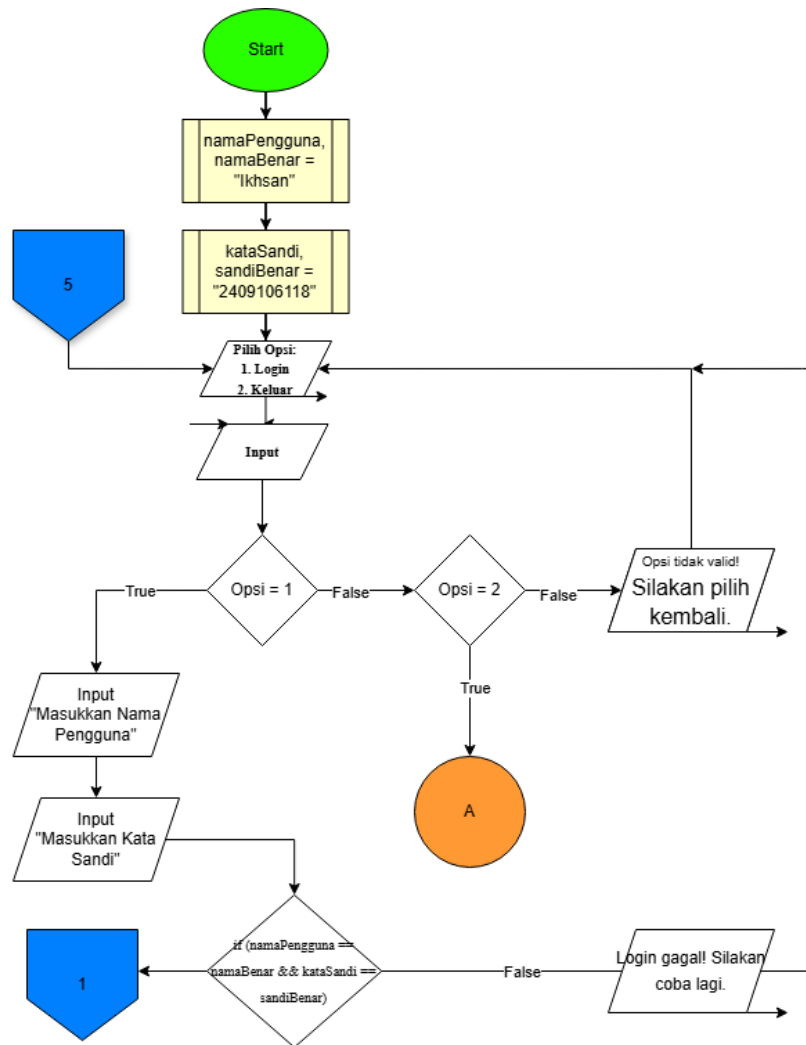
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Ikhsan (2409106118)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dibuat untuk memvalidasi kode sandi yang dimasukkan oleh pengguna setelah berhasil melakukan login. Validasi ini mencakup beberapa aspek, yaitu mengecek apakah kode sandi merupakan bilangan prima, memenuhi aturan pembagian tertentu, serta apakah kode sandi merupakan bilangan palindrome. Program ini juga dilengkapi dengan sistem login yang memastikan hanya pengguna yang memiliki kredensial yang benar yang dapat mengakses fitur-fitur di dalamnya.

Fungsi dan Manfaat Utama

1. Sistem Login yang Aman

- a) Program menerapkan sistem login sederhana dengan nama pengguna dan kata sandi yang harus sesuai sebelum pengguna dapat mengakses fitur utama.
- b) Jika login gagal, pengguna diminta mencoba kembali hingga berhasil.

2. Validasi Kode Sandi dengan Berbagai Metode

- a) Cek Bilangan Prima → Menentukan apakah kode sandi yang dimasukkan merupakan bilangan prima atau bukan.
- b) Cek Aturan Pembagian → Mengevaluasi apakah kode sandi habis dibagi 2 dan 3, tetapi tidak habis dibagi 5.
- c) Cek Bilangan Palindrome → Mengecek apakah kode sandi memiliki susunan angka yang sama jika dibaca dari depan maupun belakang.

3. Keamanan Input dan Validasi Data

- a) Program menangani berbagai kemungkinan kesalahan input dengan menggunakan `cin.clear()` dan `cin.ignore()`, sehingga tidak akan terjadi crash jika pengguna memasukkan nilai yang tidak valid.
- b) Rentang kode sandi yang diterima dibatasi antara 100 hingga 99999 untuk memastikan angka yang masuk dapat diuji dengan benar.

4. Struktur Modular dan Mudah Dikembangkan

- a) Program menggunakan fungsi terpisah (cekPrima(), cekPalindrome(), login()) untuk meningkatkan modularitas dan kemudahan pemeliharaan kode.
- b) Menu interaktif memungkinkan pengguna untuk mengganti kode sandi, melakukan pengecekan ulang, atau logout tanpa harus keluar dari program.

2.2 Penjelasan Alur & Algoritma

Program ini memiliki beberapa tahapan utama, yaitu **login**, **input kode sandi**, **validasi kode sandi**, serta **menu interaktif** untuk melakukan pengecekan angka berdasarkan kriteria tertentu. Berikut adalah penjelasan alur kerja secara berurutan:

1. Program Dimulai dan Menampilkan Menu Utama

Ketika program dijalankan, pengguna akan disambut dengan **menu utama** yang berisi dua pilihan:

1. **Login** → Memasukkan nama pengguna dan kata sandi untuk mengakses fitur utama.
2. **Keluar** → Menghentikan program jika pengguna memilih opsi ini.

Kode terkait:

```
int opsi;
cout << "===== SELAMAT DATANG =====\n";
cout << "1. Login\n";
cout << "2. Keluar\n";
cout << "Pilih opsi: ";
```

Jika pengguna memilih opsi **1 (Login)**, program akan memanggil fungsi login(). Jika memilih **2 (Keluar)**, program akan berhenti dengan return 0.

2. Sistem Login

- Program meminta pengguna untuk memasukkan **nama pengguna** dan **kata sandi**.
- Jika input cocok dengan kredensial yang sudah ditentukan (namaBenar dan sandiBenar), program akan menampilkan pesan "**Login berhasil!**" dan lanjut ke tahap berikutnya.
- Jika input salah, pengguna akan diminta untuk mencoba kembali hingga berhasil.

Kode terkait (fungsi login()):

```
bool login() {
    string namaPengguna, namaBenar = "Ikhsan";
    string kataSandi, sandiBenar = "2409106118";

    cin.ignore();
    while (true) {
        cout << "==== LOGIN ====\\n";
        cout << "Nama Pengguna: ";
        getline(cin, namaPengguna);
        cout << "Kata Sandi: ";
        getline(cin, kataSandi);

        if (namaPengguna == namaBenar && kataSandi == sandiBenar) {
            cout << "Login berhasil!\\n";
            return true;
        } else {
            cout << "Login gagal! Silakan coba lagi.\\n";
        }
    }
}
```

Jika login berhasil, program akan meminta **kode sandi** dari pengguna.

3. Memasukkan Kode Sandi

- Setelah login berhasil, pengguna diminta memasukkan kode sandi dalam bentuk **bilangan bulat positif**.
- Program akan memvalidasi input agar **berada dalam rentang 100 hingga 99999**. Jika angka tidak memenuhi syarat, pengguna harus memasukkan ulang.

Kode terkait:

```
cout << "\nMasukkan kode sandi (bilangan bulat positif): ";
while (!(cin >> kodeSandi)) {
    cout << "Input tidak valid! Masukkan angka.\nMasukkan kode
sandi: ";
    cin.clear();
    cin.ignore(10000, '\n');
}

if (kodeSandi <= 100 || kodeSandi >= 99999) {
    cout << "Kode sandi harus lebih dari 100 dan kurang dari
99999!\n";
    continue;
}
```

Jika kode sandi valid, program akan menampilkan **menu utama** untuk melakukan pengecekan terhadap kode sandi tersebut.

4. Menu Pengecekan Kode Sandi

Setelah kode sandi dimasukkan, pengguna akan diberikan **beberapa opsi** untuk melakukan pengecekan terhadap angka tersebut:

1. **Cek apakah kode sandi 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. **Masukkan kode sandi baru**
5. **Logout (kembali ke menu utama)**

Kode terkait:

```
cout << "\n==== MENU =====\n";
cout << "1. Cek apakah kode sandi adalah bilangan prima\n";
cout << "2. Cek apakah kode sandi habis dibagi 2 dan 3
tetapi tidak habis dibagi 5\n";
cout << "3. Cek apakah kode sandi adalah bilangan
palindrome\n";
cout << "4. Masukkan kode sandi baru\n";
cout << "5. Logout\n";
cout << "Pilih menu: ";
```

Jika pengguna memilih **1**, **2**, atau **3**, program akan menjalankan fungsi yang sesuai. Jika memilih **4**, pengguna akan diminta memasukkan kode sandi baru. Jika memilih **5**, pengguna akan logout dan kembali ke menu utama.

5. Pengecekan Kode Sandi

Setiap opsi dalam menu akan melakukan pemeriksaan tertentu terhadap kode sandi yang sudah dimasukkan.

a. Cek Bilangan Prima

- Program akan memanggil fungsi cekPrima() untuk menentukan apakah kode sandi adalah bilangan **prima** atau bukan.
- Logikanya: Jika angka lebih kecil dari **2**, langsung dikembalikan **false**. Jika angka memiliki faktor selain **1** dan **dirinya sendiri**, maka bukan prima.

Kode terkait:

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

Jika angka adalah **bilangan prima**, program mencetak:

Kode sandi adalah bilangan prima.

Jika bukan, program mencetak:

Kode sandi bukan bilangan prima.

b. Cek Pembagian dengan 2, 3, dan 5

- Program memeriksa apakah kode sandi **habis dibagi 2 dan 3, tetapi tidak habis dibagi 5**.
- Jika memenuhi syarat, program mencetak:

Kode sandi valid sesuai kriteria.

- Jika tidak, program mencetak:

Kode sandi tidak valid sesuai kriteria.

Kode terkait:

```
if (kodeSandi % 2 == 0 && kodeSandi % 3 == 0 &&
kodeSandi % 5 != 0)
    cout << "Kode sandi valid sesuai kriteria.\n";
else
    cout << "Kode sandi tidak valid sesuai kriteria.\n";
```

c. Cek Bilangan Palindrome

- Program akan memanggil fungsi cekPalindrome(), yang membalikkan angka lalu membandingkannya dengan angka asli.
- Jika angka sama sebelum dan sesudah dibalik, maka itu adalah **palindrome**.

Kode terkait:

```
bool cekPalindrome(int angka) {
    int asli = angka, terbalik = 0, digit;
    while (angka > 0) {
        digit = angka % 10;
        terbalik = terbalik * 10 + digit;
        angka /= 10;
    }
    return asli == terbalik;
}
```

Jika angka adalah **palindrome**, program mencetak:

Kode sandi adalah bilangan palindrome.

Jika bukan, program mencetak:

Kode sandi bukan bilangan palindrome.

6. Logout atau Masukkan Kode Sandi Baru

- Jika pengguna memilih **Logout**, program kembali ke menu utama.
- Jika memilih **Masukkan kode sandi baru**, program akan meminta input angka lagi dan mengulangi proses validasi.

3. Source Code

```
#include <iostream>
using namespace std;

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

bool cekPalindrome(int angka) {
    int asli = angka, terbalik = 0, digit;
    while (angka > 0) {
        digit = angka % 10;
        terbalik = terbalik * 10 + digit;
        angka /= 10;
    }
    return asli == terbalik;
}

bool login() {
    string namaPengguna, namaBenar = "Ikhsan";
    string kataSandi, sandiBenar = "2409106118";

    cin.ignore();
    while (true) {
        cout << "==== LOGIN ==== \n";
        cout << "Nama Pengguna: ";
        getline(cin, namaPengguna);
        cout << "Kata Sandi: ";
        getline(cin, kataSandi);

        if (namaPengguna == namaBenar && kataSandi == sandiBenar) {
            cout << "Login berhasil! \n";
            return true;
        } else {
            cout << "Login gagal! Silakan coba lagi. \n";
        }
    }
}

int main() {
    while (true) {
        int opsi;
        cout << "==== SELAMAT DATANG ==== \n";
```

```

cout << "1. Login\n";
cout << "2. Keluar\n";
cout << "Pilih opsi: ";

while (!(cin >> opsi)) {
    cout << "Input tidak valid! Masukkan angka.\nPilih opsi: ";
    cin.clear();
    cin.ignore(10000, '\n');
}

if (opsi == 2) {
    cout << "Terima kasih, program berhenti.\n";
    return 0;
} else if (opsi != 1) {
    cout << "Opsi tidak valid! Silakan pilih kembali.\n";
    continue;
}

if (!login()) {
    continue;
}

while (true) {
    int kodeSandi;
    cout << "\nMasukkan kode sandi (bilangan bulat positif): ";
    while (!(cin >> kodeSandi)) {
        cout << "Input tidak valid! Masukkan angka.\nMasukkan kode
sandi: ";

        cin.clear();
        cin.ignore(10000, '\n');
    }

    if (kodeSandi <= 100 || kodeSandi >= 99999) {
        cout << "Kode sandi harus lebih dari 100 dan kurang dari
99999!\n";

        continue;
    }

    while (true) {
        int pilihan;
        cout << "\n===== MENU =====\n";
        cout << "1. Cek apakah kode sandi adalah bilangan prima\n";
        cout << "2. Cek apakah kode sandi habis dibagi 2 dan 3
tetapi tidak habis dibagi 5\n";
        cout << "3. Cek apakah kode sandi adalah bilangan
palindrome\n";

        cout << "4. Masukkan kode sandi baru\n";
        cout << "5. Logout\n";
    }
}

```

```

        cout << "Pilih menu: ";

        while (!(cin >> pilihan)) {
            cout << "Input tidak valid! Masukkan angka.\nPilih menu: ";
            cin.clear();
            cin.ignore(10000, '\n');
        }

        if (pilihan == 1) {
            cout << (cekPrima(kodeSandi) ? "Kode sandi adalah bilangan prima.\n" : "Kode sandi bukan bilangan prima.\n");
        }
        else if (pilihan == 2) {
            if (kodeSandi % 2 == 0 && kodeSandi % 3 == 0 && kodeSandi % 5 != 0)
                cout << "Kode sandi valid sesuai kriteria.\n";
            else
                cout << "Kode sandi tidak valid sesuai kriteria.\n";
        }
        else if (pilihan == 3) {
            cout << (cekPalindrome(kodeSandi) ? "Kode sandi adalah bilangan palindrome.\n" : "Kode sandi bukan bilangan palindrome.\n");
        }
        else if (pilihan == 4) {
            break;
        }
        else if (pilihan == 5) {
            cout << "Logout berhasil! Kembali ke menu utama.\n";
            goto kembaliKeMenu;
        }
        else {
            cout << "Pilihan tidak valid!\n";
        }
    }

    kembaliKeMenu;;
}
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

A. Pengujian Login

Skenario 1.1: Login Berhasil

Input: Nama pengguna = "Ikhsan", Kata sandi = "2409106118"

Output yang diharapkan: "Login berhasil!"

Skenario 1.2: Login Gagal

Input: Nama pengguna salah atau kata sandi salah (misalnya "Admin", "1234")

Output yang diharapkan: "Login gagal! Silakan coba lagi." (program terus meminta input hingga benar)

B. Pengujian Input Menu Utama

Skenario 2.1: Memilih Opsi "Keluar"

Input: 2

Output yang diharapkan: "Terima kasih, program berhenti."

Skenario 2.2: Memilih Opsi Tidak Valid

Input: Angka selain 1 atau 2 (misalnya 3, -1, huruf)

Output yang diharapkan: "Opsi tidak valid! Silakan pilih kembali."

C. Pengujian Input Kode Sandi

Skenario 3.1: Input Kode Sandi Valid

Input: Bilangan antara 101 dan 99998 (misalnya 12345)

Output yang diharapkan: Lanjut ke menu pemrosesan kode sandi

Skenario 3.2: Input Kode Sandi Tidak Valid

Input: Bilangan ≤ 100 atau ≥ 99999 (misalnya 99, 100000)

Output yang diharapkan: "Kode sandi harus lebih dari 100 dan kurang dari 99999!"

Skenario 3.3: Input Bukan Bilangan

Input: Karakter selain angka (misalnya "abc", "!@#")

Output yang diharapkan: "Input tidak valid! Masukkan angka."

D. Pengujian Menu Pemrosesan Kode Sandi

Skenario 4.1: Memeriksa Bilangan Prima

Input: Pilihan 1, Kode sandi = 101

Output yang diharapkan: "Kode sandi adalah bilangan prima."

Input: Pilihan 1, Kode sandi = 102

Output yang diharapkan: "Kode sandi bukan bilangan prima."

Skenario 4.2: Memeriksa Kelipatan 2 dan 3, tetapi bukan 5

Input: Pilihan 2, Kode sandi = 102

Output yang diharapkan: "Kode sandi valid sesuai kriteria."

Input: Pilihan 2, Kode sandi = 101

Output yang diharapkan: "Kode sandi tidak valid sesuai kriteria."

Skenario 4.3: Memeriksa Bilangan Palindrome

Input: Pilihan 3, Kode sandi = 101

Output yang diharapkan: "Kode sandi adalah bilangan palindrome."

Input: Pilihan 3, Kode sandi = 12345

Output yang diharapkan: "Kode sandi bukan bilangan palindrome."

Skenario 4.4: Memasukkan Kode Sandi Baru

Input: Pilihan 4

Output yang diharapkan: Kembali ke input kode sandi

Skenario 4.5: Logout

Input: Pilihan 5

Output yang diharapkan: "Logout berhasil! Kembali ke menu utama."

Skenario 4.6: Memilih Opsi Tidak Valid

Input: Angka selain 1-5 atau karakter lain

Output yang diharapkan: "Pilihan tidak valid!"

4.2 Hasil Output

A. Pengujian Login

```
===== SELAMAT DATANG =====
1. Login
2. Keluar
Pilih opsi: 1
===== LOGIN =====
Nama Pengguna: Lorem Ipsum
Kata Sandi: 123
Login gagal! Silakan coba lagi.
===== LOGIN =====
Nama Pengguna: Ikhsan
Kata Sandi: 2409106118
Login berhasil!
```

Gambar 4.2.1 Pengujian Login

B. Pengujian Input Menu Utama

```
===== SELAMAT DATANG =====  
1. Login  
2. Keluar  
Pilih opsi: 3  
Opsi tidak valid! Silakan pilih kembali.  
===== SELAMAT DATANG =====  
1. Login  
2. Keluar  
Pilih opsi: 2  
Terima kasih, program berhenti.  
PS D:\GITHUB\Praktikum Apl\Post-test\Post-test-1>
```

Gambar 4.2.2 Pengujian Input Menu Utama

C. Pengujian Input Kode Sandi

```
Masukkan kode sandi (bilangan bulat positif): 12345  
===== MENU =====  
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. Masukkan kode sandi baru  
5. Logout  
Pilih menu: |
```

Gambar 4.2.3 Pengujian Input Kode Sandi

```
Masukkan kode sandi (bilangan bulat positif): 99  
Kode sandi harus lebih dari 100 dan kurang dari 99999!  
Masukkan kode sandi (bilangan bulat positif):
```

Gambar 4.2.4 Pengujian Input Kode Sandi

```
Masukkan kode sandi (bilangan bulat positif): abc  
Input tidak valid! Masukkan angka.  
Masukkan kode sandi:
```

Gambar 4.2.5 Pengujian Input Kode Sandi

D. Pengujian Menu Pemrosesan Kode Sandi

```
Masukkan kode sandi: 101

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 1
Kode sandi adalah bilangan prima.

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 4
Masukkan kode sandi (bilangan bulat positif): 102

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 1
Kode sandi bukan bilangan prima.
```

Gambar 4.2.6 Pengujian Menu Pemrosesan Kode Sandi

```
Masukkan kode sandi (bilangan bulat positif): 102

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 2
Kode sandi valid sesuai kriteria.

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 4
Masukkan kode sandi (bilangan bulat positif): 101

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 2
Kode sandi tidak valid sesuai kriteria.
```

Gambar 4.2.7 Pengujian Menu Pemrosesan Kode Sandi

```
Masukkan kode sandi (bilangan bulat positif): 101

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 3
Kode sandi adalah bilangan palindrome.

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 4
Masukkan kode sandi (bilangan bulat positif): 12345

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 3
Kode sandi bukan bilangan palindrome.
```

Gambar 4.2.8 Pengujian Menu Pemrosesan Kode Sandi


```

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 4

Masukkan kode sandi (bilangan bulat positif): 416

```

Gambar 4.2.8 Pengujian Menu Pemrosesan Kode Sandi

```

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 5
Logout berhasil! Kembali ke menu utama.
===== SELAMAT DATANG =====
1. Login
2. Keluar
Pilih opsi:

```

Gambar 4.2.9 Pengujian Menu Pemrosesan Kode Sandi

```

===== MENU =====
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. Masukkan kode sandi baru
5. Logout
Pilih menu: 6
Pilihan tidak valid!

```

Gambar 4.2.10 Pengujian Menu Pemrosesan Kode Sandi

5. Git

```
MINGW64/d/GITHUB/Praktikum-Apl
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git config --global user.email "hsanzxx9@gmail.com"
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git init
Reinitialized existing Git repository in D:/GITHUB/Praktikum-Apl/.git/
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git add .
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git remote add origin https://github.com/Ikhsan-118/Praktikum-Apl.git
error: remote origin already exists.
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git commit -m "Ikhsan Posttest Ganteng"
[main (root-commit) f87887d] Ikhsan Posttest Ganteng
2 files changed, 90 insertions(+)
 create mode 100644 Post-test/Post-test-1/2409106118-Ikhsan-PT-1.cpp
 create mode 100644 Post-test/Post-test-1/2409106118-Ikhsan-PT-1.exe
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$ git push -u origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 693.72 KiB | 3.03 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Ikhsan-118/Praktikum-Apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
ASUS@DESKTOP-8AN5309 MINGW64 /d/GITHUB/Praktikum-Apl (main)
$
```

Gambar 5 Terminal Git Bash

1. Mengatur Email Global untuk Git

\$ git config --global user.email "hsanzxx9@gmail.com"

- Perintah ini digunakan untuk mengatur email global Git, yang akan digunakan dalam setiap commit yang dibuat oleh pengguna.
- Email ini dikaitkan dengan identitas pengguna dalam repository Git.
-

2. Menginisialisasi Repository Git

\$ git init

- Perintah ini menginisialisasi (membuat) repository Git dalam folder Praktikum-Apl.
- Jika sudah ada repository Git di folder tersebut, maka Git hanya akan mengingatkan bahwa repository sudah ada (reinitialized).
-

3. Menambahkan File ke Staging Area

\$ git add .

- Perintah ini menambahkan semua file yang ada di dalam folder ke **staging area**.
- Staging area adalah tempat sementara sebelum file dikomit ke dalam repository.

4. Menambahkan Remote Repository (Gagal karena Sudah Ada)

\$ git remote add origin https://github.com/Ikhsan-118/Praktikum-Apl.git

- Perintah ini digunakan untuk menambahkan repository remote dengan nama origin.
- **Error: "remote origin already exists"**, ini terjadi karena sebelumnya sudah ada repository remote yang dikaitkan dengan nama origin.

•

5. Membuat Commit dengan Pesan "Ikhsan Posttest Ganteng"

\$ git commit -m "Ikhsan Posttest Ganteng"

- Perintah ini menyimpan perubahan dalam repository dengan commit dan pesan "Ikhsan Posttest Ganteng".
- File yang dikomit:
 - Post-test/Post-test-1/2409106118-Ikhsan-PT-1.cpp
 - Post-test/Post-test-1/2409106118-Ikhsan-PT-1.exe

6. Mendorong (Push) Perubahan ke Repository Remote

\$ git push -u origin main

- Perintah ini mengunggah (push) perubahan ke repository remote pada branch main.
- Karena ini adalah push pertama, flag -u digunakan untuk mengatur branch lokal main agar terhubung dengan branch main di remote repository.
- Proses yang terjadi:
 - Menghitung objek (Enumerating objects: 6).
 - Mengompresi objek sebelum mengunggahnya.
 - Menulis (mengunggah) objek ke GitHub.
 - Menampilkan informasi bahwa branch main sekarang dilacak oleh remote repository origin/main.