

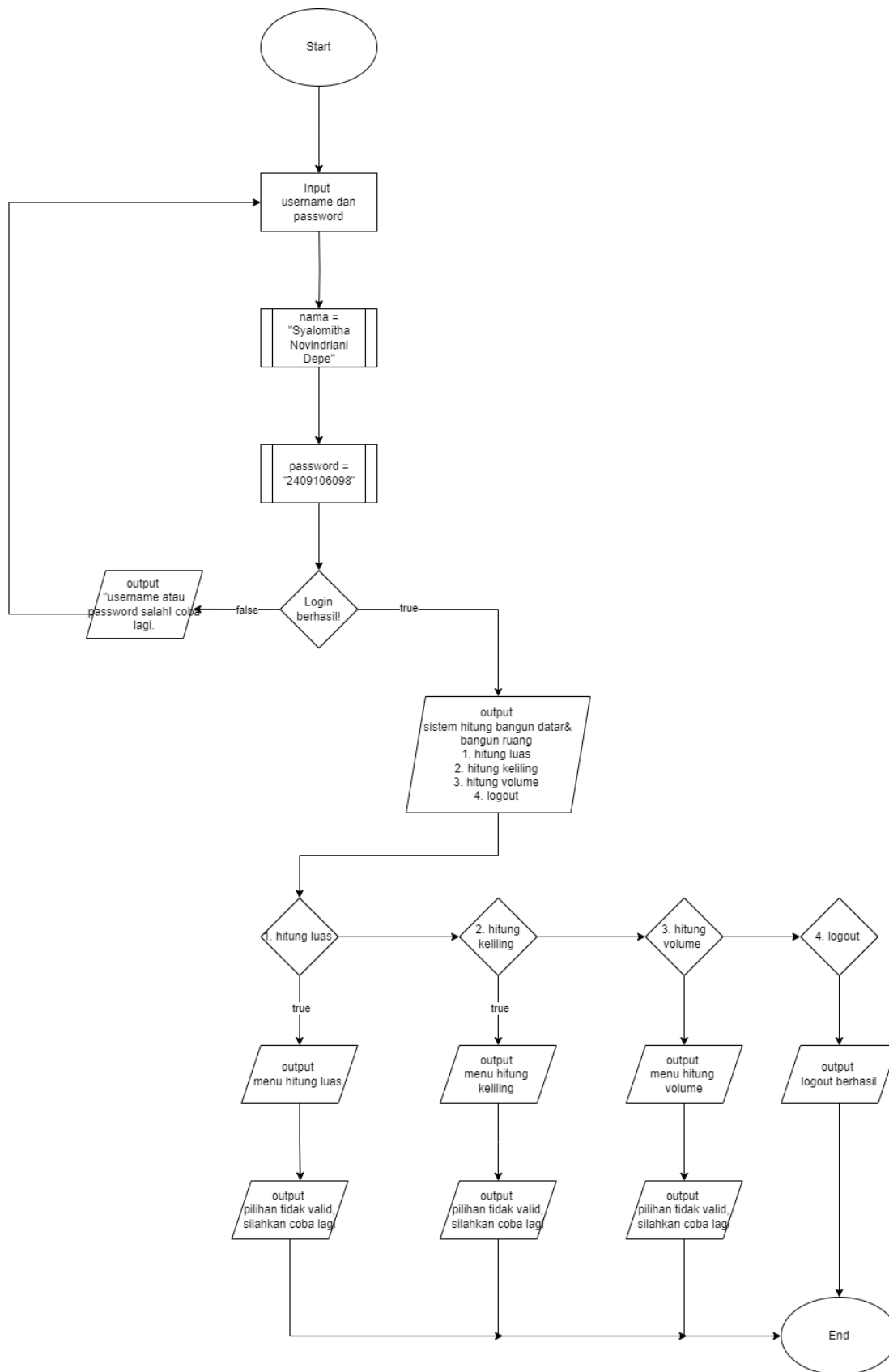
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Syalomitha Novindriani Depe (2409106098)
Kelas (C1'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

2.1 Deskripsi Singkat Program

Tujuan:

Program ini dibuat untuk membantu pengguna dalam menghitung luas, keliling, dan volume berbagai bangun datar serta bangun ruang secara otomatis, akurat, dan efisien. Program ini juga dilengkapi dengan sistem login untuk memastikan hanya pengguna yang berhak dapat mengaksesnya.

Fungsi/Manfaat:

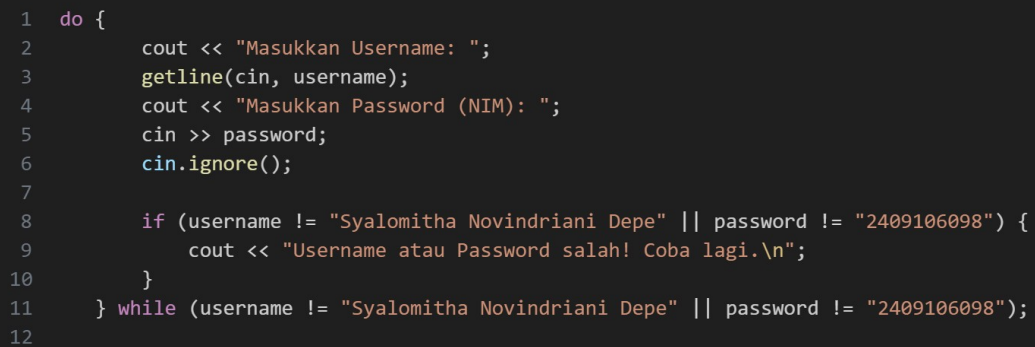
1. **Menghitung Luas** dari persegi, persegi panjang, lingkaran, dan segitiga.
2. **Menghitung Keliling** dari persegi, persegi panjang, lingkaran, dan segitiga.
3. **Menghitung Volume** dari kubus, balok, bola, dan kerucut.
4. **Meningkatkan Efisiensi** dengan validasi input untuk mencegah kesalahan dalam perhitungan.
5. **Keamanan Data** melalui sistem login untuk memastikan hanya pengguna tertentu yang dapat mengakses program.
- 6.

2.2 Penjelasan Alur & Algoritma

Program ini merupakan sistem perhitungan luas, keliling, dan volume bangun geometri yang dilengkapi dengan sistem login untuk membatasi akses pengguna. Berikut adalah alur kerja program dari awal hingga akhir:

1. Login Pengguna

- Program meminta pengguna memasukkan **Username** dan **Password (NIM)**.
- Jika username dan password sesuai, login berhasil dan pengguna dapat mengakses menu utama.
- Jika salah, pengguna harus mengulangi proses login.



```
1  do {
2      cout << "Masukkan Username: ";
3      getline(cin, username);
4      cout << "Masukkan Password (NIM): ";
5      cin >> password;
6      cin.ignore();
7
8      if (username != "Syalomitha Novindriani Depe" || password != "2409106098") {
9          cout << "Username atau Password salah! Coba lagi.\n";
10     }
11 } while (username != "Syalomitha Novindriani Depe" || password != "2409106098");
12
```

Fungsi do-while memastikan pengguna harus memasukkan kredensial yang benar sebelum lanjut.

2. Menu Utama

Setelah login berhasil, pengguna disajikan menu utama dengan pilihan:

1. **Hitung Luas**
2. **Hitung Keliling**
3. **Hitung Volume**

4. Logout

```
1  int pilihan;
2  do {
3      cout << "\n== SISTEM HITUNG BANGUN DATAR DAN BANGUN RUANG ==\n";
4      cout << "1. Hitung Luas\n";
5      cout << "2. Hitung Keliling\n";
6      cout << "3. Hitung Volume\n";
7      cout << "4. Logout\n";
8      cout << "Pilih menu: ";
9      cin >> pilihan;
10
11     switch (pilihan) {
12         case 1: hitungLuas(); break;
13         case 2: hitungKeliling(); break;
14         case 3: hitungVolume(); break;
15         case 4: cout << "Logout Berhasil."; break;
16         default: cout << "Pilihan tidak valid.\n"; break;
17     }
18 } while (pilihan != 4);
```

switch-case digunakan untuk mengeksekusi pilihan menu pengguna.

3. Perhitungan Luas

- Jika pengguna memilih **Hitung Luas**, program menampilkan sub-menu:
 1. **Persegi**
 2. **Persegi Panjang**
 3. **Lingkaran**
 4. **Segitiga**
 5. **Kembali ke Menu Utama**
- Program meminta input nilai yang sesuai untuk bentuk yang dipilih.
- Setelah input diterima, program menghitung luas dan menampilkan hasilnya.

```

1 void hitungLuas() {
2     int pilihan;
3     double sisi, panjang, lebar, jari, alas, tinggi;
4
5     do {
6         cout << "\n== HITUNG LUAS ==\n";
7         cout << "1. Persegi\n2. Persegi Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\n";
8         cout << "Pilih bentuk: ";
9         cin >> pilihan;
10
11         switch (pilihan) {
12             case 1:
13                 do {
14                     cout << "Masukkan sisi persegi: "; cin >> sisi;
15                 } while (sisi <= 0);
16                 cout << "Luas: " << sisi * sisi << "\n";
17                 break;
18             case 2:
19                 do {
20                     cout << "Masukkan panjang: "; cin >> panjang;
21                     cout << "Masukkan lebar: "; cin >> lebar;
22                 } while (panjang <= 0 || lebar <= 0);
23                 cout << "Luas: " << panjang * lebar << "\n";
24                 break;
25             case 3:
26                 do {
27                     cout << "Masukkan jari-jari lingkaran: "; cin >> jari;
28                 } while (jari <= 0);
29                 cout << "Luas: " << PI * jari * jari << "\n";
30                 break;
31             case 4:
32                 do {
33                     cout << "Masukkan alas: "; cin >> alas;
34                     cout << "Masukkan tinggi: "; cin >> tinggi;
35                 } while (alas <= 0 || tinggi <= 0);
36                 cout << "Luas: " << 0.5 * alas * tinggi << "\n";
37                 break;
38             }
39         } while (pilihan != 5);
40     }

```

Program menggunakan validasi input agar nilai yang dimasukkan tidak negatif atau nol.

4. Perhitungan Keliling

- Jika pengguna memilih **Hitung Keliling**, program menampilkan sub-menu yang sama seperti luas.
- Program meminta input angka dan menghitung keliling sesuai rumus bangun yang dipilih.

```

1 void hitungKeliling() {
2     int pilihan;
3     double sisi, panjang, lebar, jari, alas, tinggi, miring;
4
5     do {
6         cout << "\n== HITUNG KELILING ==\n";
7         cout << "1. Persegi\n2. Persegi Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\n";
8         cout << "Pilih bentuk: ";
9         cin >> pilihan;
10
11         switch (pilihan) {
12             case 1:
13                 do {
14                     cout << "Masukkan sisi persegi: "; cin >> sisi;
15                 } while (sisi <= 0);
16                 cout << "Keliling: " << 4 * sisi << "\n";
17                 break;
18             case 2:
19                 do {
20                     cout << "Masukkan panjang: "; cin >> panjang;
21                     cout << "Masukkan lebar: "; cin >> lebar;
22                 } while (panjang <= 0 || lebar <= 0);
23                 cout << "Keliling: " << 2 * (panjang + lebar) << "\n";
24                 break;
25             case 3:
26                 do {
27                     cout << "Masukkan jari-jari lingkaran: "; cin >> jari;
28                 } while (jari <= 0);
29                 cout << "Keliling: " << 2 * PI * jari << "\n";
30                 break;
31             case 4:
32                 do {
33                     cout << "Masukkan alas: "; cin >> alas;
34                     cout << "Masukkan tinggi: "; cin >> tinggi;
35                     cout << "Masukkan sisi miring: "; cin >> miring;
36                 } while (alas <= 0 || tinggi <= 0 || miring <= 0);
37                 cout << "Keliling: " << alas + tinggi + miring << "\n";
38                 break;
39             }
40         } while (pilihan != 5);
41     }

```

Konsep yang sama digunakan untuk keliling.

5. Perhitungan Volume

- Jika pengguna memilih **Hitung Volume**, program menampilkan sub-menu:
 1. Kubus
 2. Balok
 3. Bola
 4. Kerucut
 5. Kembali

- Program meminta input angka dan menghitung volume.

```
1 void hitungVolume() {
2     int pilihan;
3     double sisi, panjang, lebar, tinggi, jari;
4
5     do {
6         cout << "\n== HITUNG VOLUME ==\n";
7         cout << "1. Kubus\n2. Balok\n3. Bola\n4. Kerucut\n5. Kembali\n";
8         cout << "Pilih bentuk: ";
9         cin >> pilihan;
10
11         switch (pilihan) {
12             case 1:
13                 do {
14                     cout << "Masukkan sisi kubus: "; cin >> sisi;
15                 } while (sisi <= 0);
16                 cout << "Volume: " << sisi * sisi * sisi << "\n";
17                 break;
18             case 2:
19                 do {
20                     cout << "Masukkan panjang: "; cin >> panjang;
21                     cout << "Masukkan lebar: "; cin >> lebar;
22                     cout << "Masukkan tinggi: "; cin >> tinggi;
23                 } while (panjang <= 0 || lebar <= 0 || tinggi <= 0);
24                 cout << "Volume: " << panjang * lebar * tinggi << "\n";
25                 break;
26             case 3:
27                 do {
28                     cout << "Masukkan jari-jari bola: "; cin >> jari;
29                 } while (jari <= 0);
30                 cout << "Volume: " << (4.0/3) * PI * pow(jari, 3) << "\n";
31                 break;
32             }
33         } while (pilihan != 5);
34     }
35 }
```

Program menggunakan pow() dari <cmath> untuk menghitung pangkat pada volume bola.

6. Logout dan Selesai

- Jika pengguna memilih **Logout**, program menampilkan pesan dan berhenti.



```
1 case 4: cout << "Logout Berhasil."; break;
```

Program keluar dari do-while dan berhenti.

3. Source Code

```
#include <iostream>
#include <cmath>

#define _USE_MATH_DEFINES

using namespace std;

void hitungLuas();
void hitungKeliling();
void hitungVolume();

int main() {
    string username, password;

    // Sistem Login
    do {
        cout << "Masukkan Username: ";
        getline(cin, username);
        cout << "Masukkan Password (NIM): ";
        cin >> password;
        cin.ignore();

        if (username != "Syalomitha Novindriani Depe" || password !=
"2409106098") {
            cout << "Username atau Password salah! Coba lagi.\n";
        }
    } while (username != "Syalomitha Novindriani Depe" || password !=
"2409106098");

    cout << "Login Berhasil!\n";

    int pilihan;
```

```

do {
    cout << "\n== SISTEM HITUNG BANGUN DATAR DAN BANGUN RUANG ==\n";

    cout << "1. Hitung Luas\n";
    cout << "2. Hitung Keliling\n";
    cout << "3. Hitung Volume\n";
    cout << "4. Logout\n";
    cout << "Pilih menu: ";
    cin >> pilihan;

    switch (pilihan) {
        case 1: hitungLuas(); break;
        case 2: hitungKeliling(); break;
        case 3: hitungVolume(); break;
        case 4: cout << "Logout Berhasil.\n"; break;
        default: cout << "Pilihan tidak valid.\n"; break;
    }
} while (pilihan != 4);

return 0;
}

void hitungLuas() {
    int pilihan;
    double sisi, panjang, lebar, jari, alas, tinggi;

    do {
        cout << "\n== HITUNG LUAS ==\n";
        cout << "1. Persegi\n2. Persegi Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\n";
        cout << "Pilih bentuk: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Masukkan sisi persegi: "; cin >> sisi;
                cout << "Luas: " << sisi * sisi << "\n";
                break;
            case 2:
                cout << "Masukkan panjang: "; cin >> panjang;
                cout << "Masukkan lebar: "; cin >> lebar;
                cout << "Luas: " << panjang * lebar << "\n";
                break;
            case 3:
                cout << "Masukkan jari-jari lingkaran: "; cin >>
jari;

                cout << "Luas: " << M_PI * jari * jari << "\n";
                break;

```

```

        case 4:
            cout << "Masukkan alas: "; cin >> alas;
            cout << "Masukkan tinggi: "; cin >> tinggi;
            cout << "Luas: " << 0.5 * alas * tinggi << "\n";
            break;
        case 5: return;
        default: cout << "Pilihan tidak valid.\n"; break;
    }
} while (true);
}

void hitungKeliling() {
    int pilihan;
    double sisi, panjang, lebar, jari, alas, tinggi, miring;

    do {
        cout << "\n== HITUNG KELILING ==\n";
        cout << "1. Persegi\n2. Persegi Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\n";
        cout << "Pilih bentuk: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Masukkan sisi persegi: "; cin >> sisi;
                cout << "Keliling: " << 4 * sisi << "\n";
                break;
            case 2:
                cout << "Masukkan panjang: "; cin >> panjang;
                cout << "Masukkan lebar: "; cin >> lebar;
                cout << "Keliling: " << 2 * (panjang + lebar) << "\n";
                break;
            case 3:
                cout << "Masukkan jari-jari lingkaran: "; cin >> jari;
                cout << "Keliling: " << 2 * M_PI * jari << "\n";
                break;
            case 4:
                cout << "Masukkan alas: "; cin >> alas;
                cout << "Masukkan tinggi: "; cin >> tinggi;
                cout << "Masukkan sisi miring: "; cin >> miring;
                cout << "Keliling: " << alas + tinggi + miring << "\n";
                break;
            case 5: return;
            default: cout << "Pilihan tidak valid.\n"; break;
        }
    }
}

```

```

    } while (true);
}

void hitungVolume() {
    int pilihan;
    double sisi, panjang, lebar, tinggi, jari;

    do {
        cout << "\n== HITUNG VOLUME ==\n";
        cout << "1. Kubus\n2. Balok\n3. Bola\n4. Kerucut\n5. Kembali\n";

        cout << "Pilih bentuk: ";
        cin >> pilihan;

        switch (pilihan) {
            case 1:
                cout << "Masukkan sisi kubus: "; cin >> sisi;
                cout << "Volume: " << pow(sisi, 3) << "\n";
                break;
            case 2:
                cout << "Masukkan panjang: "; cin >> panjang;
                cout << "Masukkan lebar: "; cin >> lebar;
                cout << "Masukkan tinggi: "; cin >> tinggi;
                cout << "Volume: " << panjang * lebar * tinggi << "\n";
                break;
            case 3:
                cout << "Masukkan jari-jari bola: "; cin >> jari;
                cout << "Volume: " << (4.0/3) * M_PI * pow(jari, 3) << "\n";
                break;
            case 4:
                cout << "Masukkan jari-jari kerucut: "; cin >> jari;
                cout << "Masukkan tinggi kerucut: "; cin >> tinggi;
                cout << "Volume: " << (1.0/3) * M_PI * pow(jari, 2) * tinggi << "\n";
                break;
            case 5: return;
            default: cout << "Pilihan tidak valid.\n"; break;
        }
    } while (true);
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

A. Pengujian Login

Skenario 1.1: Login Berhasil

Input: Masukkan Username: Syalomitha Novindriani Depe

Masukkan Password (NIM): 2409106098

Output yang diharapkan: Login Berhasil!

Skenario 1.2: Login Gagal

Input: Masukkan Username: Admin

Masukkan Password (NIM): 1234

Output yang diharapkan: Username atau Password salah! Coba lagi.

B. Pengujian Input Menu Utama

Skenario 2.1: Memilih Opsi "Logout"

Input: Pilih menu: 4

Output yang diharapkan: Logout Berhasil.

Skenario 2.2: Memilih Opsi Tidak Valid

Input: Pilih menu: 5

Output yang diharapkan: Pilihan tidak valid.

C. Pengujian Perhitungan Luas

Skenario 3.1: Menghitung Luas Persegi

Input: Pilih menu: 1

Pilih bentuk: 1

Masukkan sisi persegi: 4

Output yang diharapkan: Luas: 16

Skenario 3.2: Menghitung Luas Lingkaran

Input: Pilih menu: 1

Pilih bentuk: 3

Masukkan jari-jari lingkaran: 7

Output yang diharapkan: Luas: 153.938

D. Pengujian Perhitungan Keliling

Skenario 4.1: Menghitung Keliling Persegi Panjang

Input: Pilih menu: 2

Pilih bentuk: 2

Masukkan panjang: 5

Masukkan lebar: 3

Output yang diharapkan: Keliling: 16

E. Pengujian Perhitungan Volume

Skenario 5.1: Menghitung Volume Kubus

Input: Pilih menu: 3

Pilih bentuk: 1

Masukkan sisi kubus: 3

Output yang diharapkan: Volume: 27

F. Pengujian Perhitungan Tidak Valid

Skenario 6.1: Memilih Opsi Tidak Valid dalam Perhitungan

Input: Pilih menu: 1

Pilih bentuk: 6

Output yang diharapkan: Pilihan tidak valid.

4.2 Hasil Output

A. Pengujian Login

```
Masukkan Username: Admin
Masukkan Password (NIM): 1234
Username atau Password salah! Coba lagi.
Masukkan Username: Syalomitha Novindriani Depe
Masukkan Password (NIM): 2409106098
Login Berhasil!
```

Gambar 4.2.1 Pengujian Login

B. Pengujian Input Menu Utama

```
== SISTEM HITUNG BANGUN DATAR DAN BANGUN RUANG ==
1. Hitung Luas
2. Hitung Keliling
3. Hitung Volume
4. Logout
Pilih menu: 1
```

Gambar 4.2.2 Pengujian Input Menu Utama

C. Pengujian Input Hitung Luas

```
== HITUNG LUAS ==
1. Persegi
2. Persegi Panjang
3. Lingkaran
4. Segitiga
5. Kembali
Pilih bentuk: 1
Masukkan sisi persegi: 25
Luas: 625
```

Gambar 4.2.3 Pengujian Hitung Luas

D. Pengujian Input Hitung Keliling

```
== HITUNG KELILING ==  
1. Persegi  
2. Persegi Panjang  
3. Lingkaran  
4. Segitiga  
5. Kembali  
Pilih bentuk: 1  
Masukkan sisi persegi: 25  
Keliling: 100
```

Gambar 4.2.4 Pengujian Hitung Keliling

E. Pengujian Input Hitung Volume

```
== HITUNG VOLUME ==  
1. Kubus  
2. Balok  
3. Bola  
4. Kerucut  
5. Kembali  
Pilih bentuk: 1  
Masukkan sisi kubus: 25  
Volume: 15625
```

Gambar 4.2.5 Penngujian Hitung Volume

5. GIT

```
ASUS@LAPTOP-GDSVM82C MINGW64 ~/Documents/GITHUB/Praktikum-APL (main)
$ git status
On branch main
nothing to commit, working tree clean

ASUS@LAPTOP-GDSVM82C MINGW64 ~/Documents/GITHUB/Praktikum-APL (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (5/5), 1.33 KiB | 454.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/syalomithaaaa/Praktikum-APL.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ASUS@LAPTOP-GDSVM82C MINGW64 ~/Documents/GITHUB/Praktikum-APL (main)
$ |
```

1. Mengatur Email dan Nama Pengguna di Git

Selain mengatur email, sebaiknya juga mengatur nama pengguna agar identitas commit lebih jelas.

```
git config --global user.email "syalomithachello1111@gmail.com"
```

Pengaturan ini akan digunakan untuk setiap commit yang dibuat di Git.

2. Memeriksa dan Menginisialisasi Repository Git

Sebelum menjalankan git init, ada baiknya memeriksa apakah folder sudah merupakan repository Git dengan perintah:

```
git status
```

Jika sudah ada, tidak perlu menginisialisasi ulang. Jika belum, jalankan.

```
git init
```

Ini akan membuat repository Git di dalam folder tersebut.

3. Menambahkan File ke Staging Area

Untuk memasukkan semua file yang ada ke dalam staging area sebelum commit, gunakan:

`git add .`

Staging area berfungsi sebagai tempat penyimpanan sementara sebelum perubahan dikomit ke repository.

4. Menambahkan Remote Repository (Jika Sudah Ada, Perbarui URL-nya)

Jika perintah berikut gagal karena remote dengan nama origin sudah ada:

`git remote add origin https://github.com/Syalomitha-098/Praktikum-Apl.git`

Maka, ubah URL remote yang sudah ada dengan perintah:

5. Melakukan Commit dengan Pesan Deskriptif

Untuk menyimpan perubahan dengan commit, gunakan perintah:

`git commit -m "mita belajar git"`

Pastikan pesan commit deskriptif dan menjelaskan perubahan yang dilakukan.

6. Mengunggah Perubahan ke Repository Remote

Untuk mengirimkan perubahan ke GitHub pada branch main, gunakan:

`git push -u origin main`

Karena ini adalah push pertama, flag `-u` digunakan agar branch main di lokal terhubung dengan branch main di repository remote.

7. Menghindari Mengunggah File yang Tidak Diperlukan

Jika ada file yang seharusnya tidak dikomit (seperti file `.exe`), buat file `.gitignore` dengan isi:

Lalu jalankan perintah berikut agar Git tidak lagi melacak file tersebut:

`git rm --cached Post-test/Post-test-1/2409106098-SyalomithaNovindrianiDepe-PT-1.exe`

8. Memeriksa Riwayat Commit Sebelum Push

Untuk memastikan commit sudah benar sebelum di-push, gunakan:

`git log --oneline --graph --all`

Ini akan menampilkan riwayat commit dengan cara yang lebih ringkas dan mudah dipahami.