

LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT



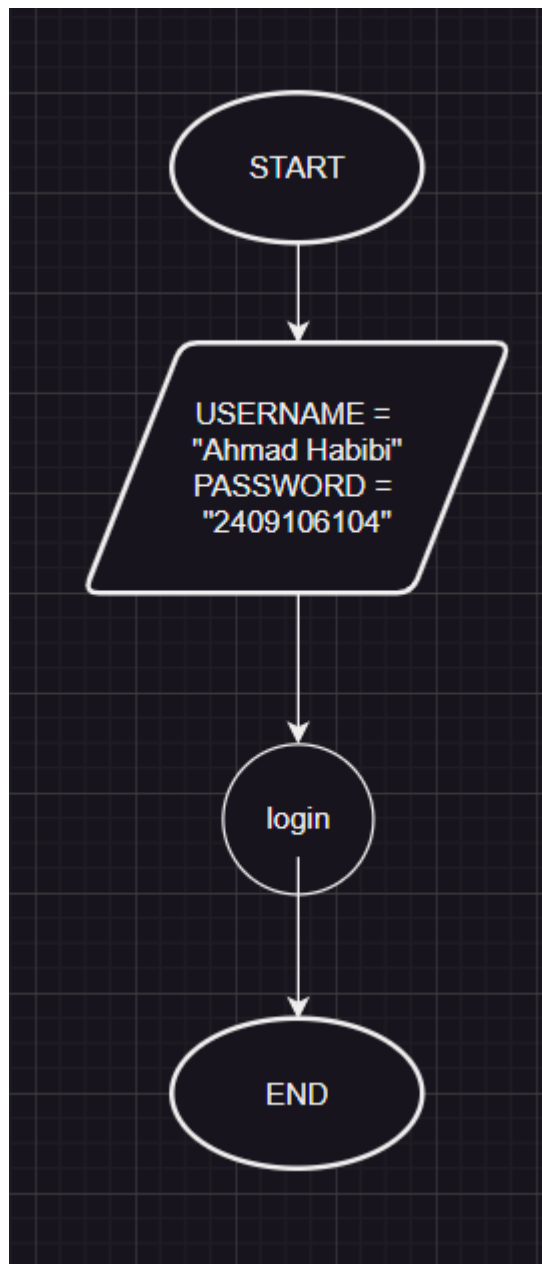
Disusun oleh:
Ahmad Habibi (2409106104)
Kelas (C1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

- Main

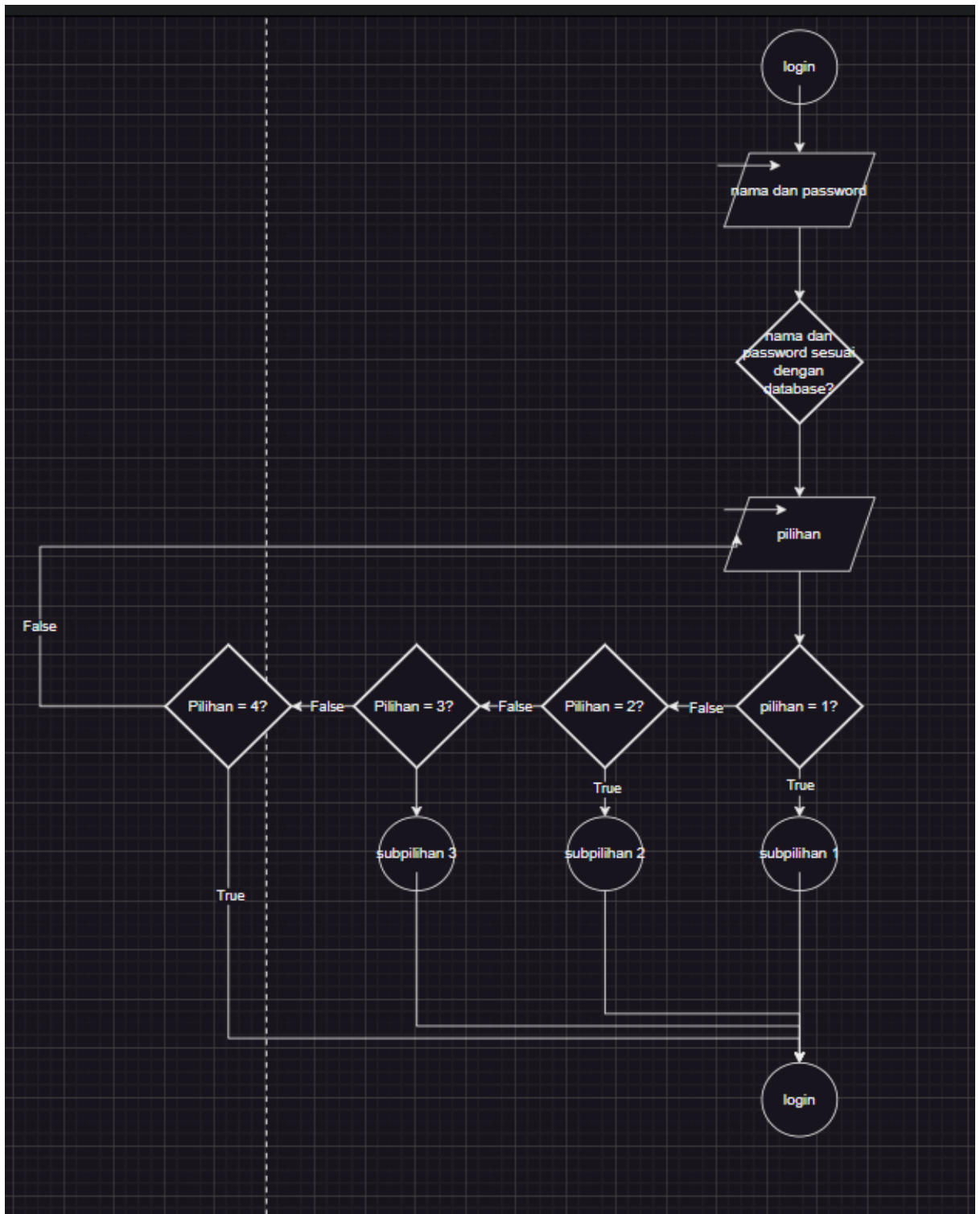
Pada bagian main, terdapat database untuk username dan password yang akan di masukkan saat login nanti, terdapat juga on page reference login.



Gambar 1.1 Main

- Login

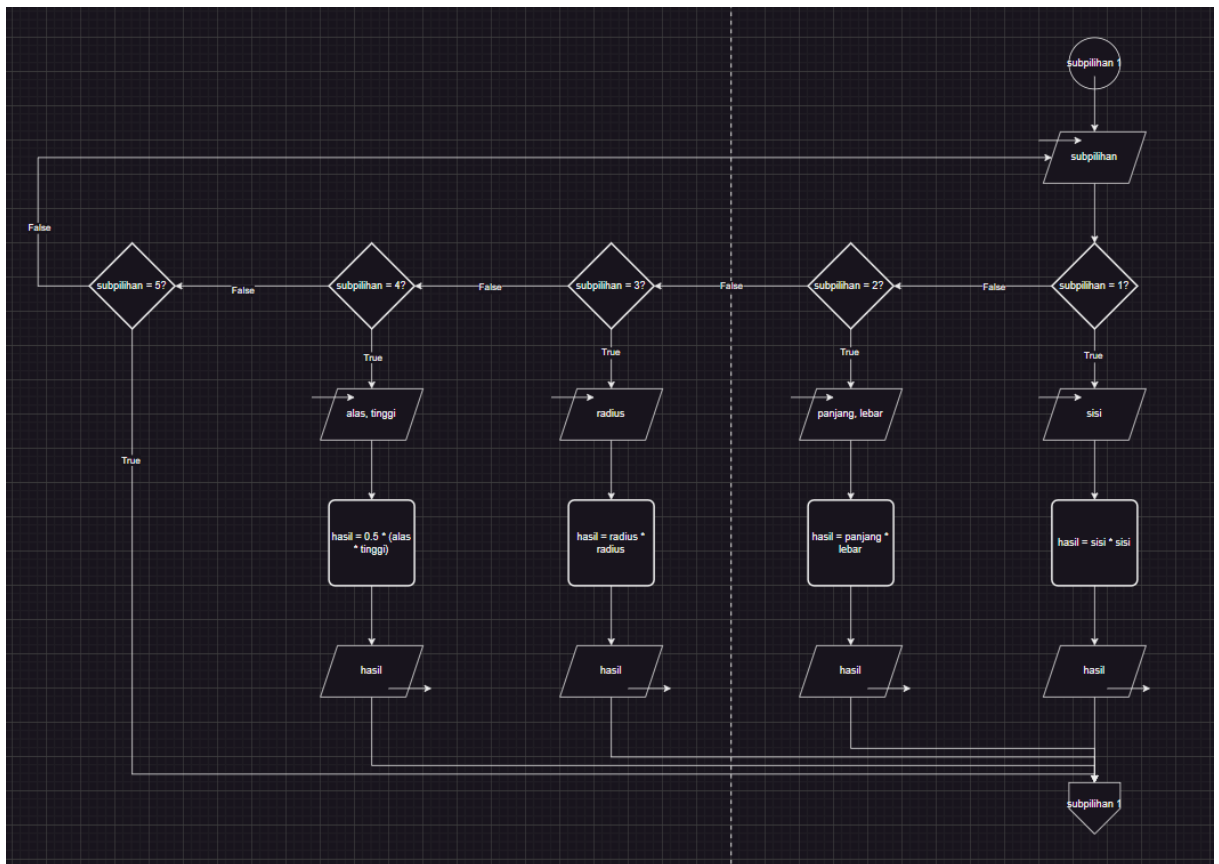
Pada on page reference login, terdapat input nama dan password, jika tidak sesuai dengan yang ada di database, program akan otomatis berhenti, jika sesuai akan dilanjut ke input pilihan, user tidak bisa memilih selain pilihan yang sudah diberikan, program akan menanyakan ulang jika user menginput yang lain.



Gambar 1.2 Login

- Sub pilihan 1

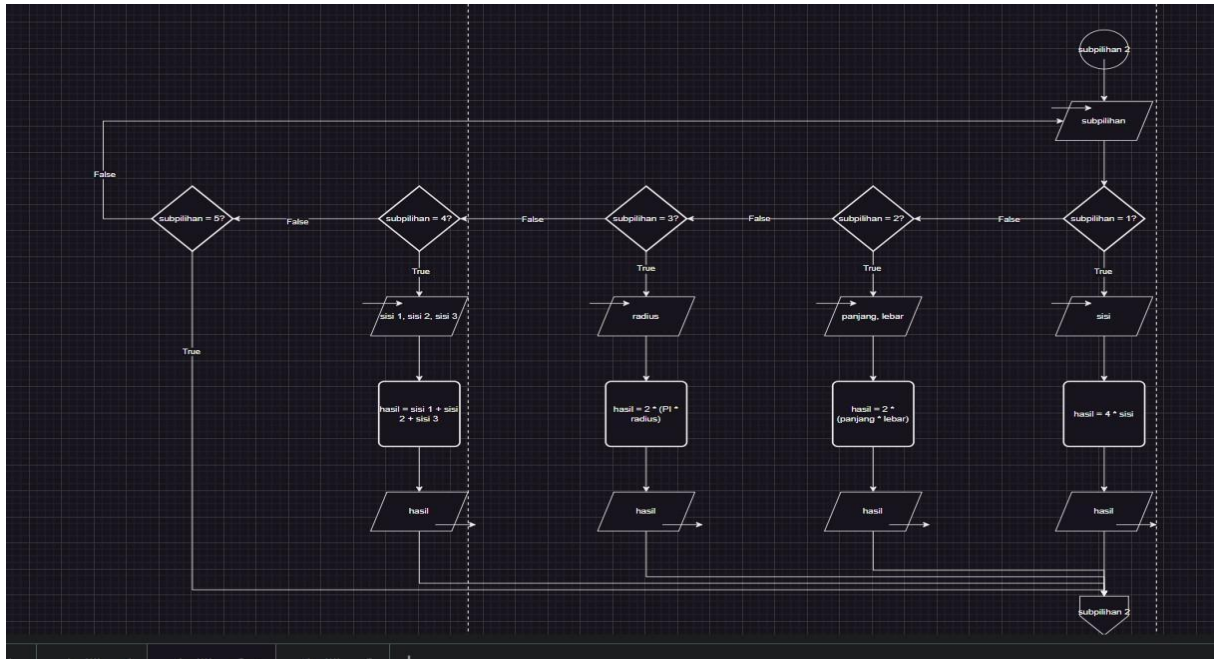
User akan diminta untuk menginput pilihan untuk menghitung luas apa, lalu akan muncul output setelah perhitungan.



Gambar 1.3 Sub-Pilihan-1

- Sub Pilihan 2

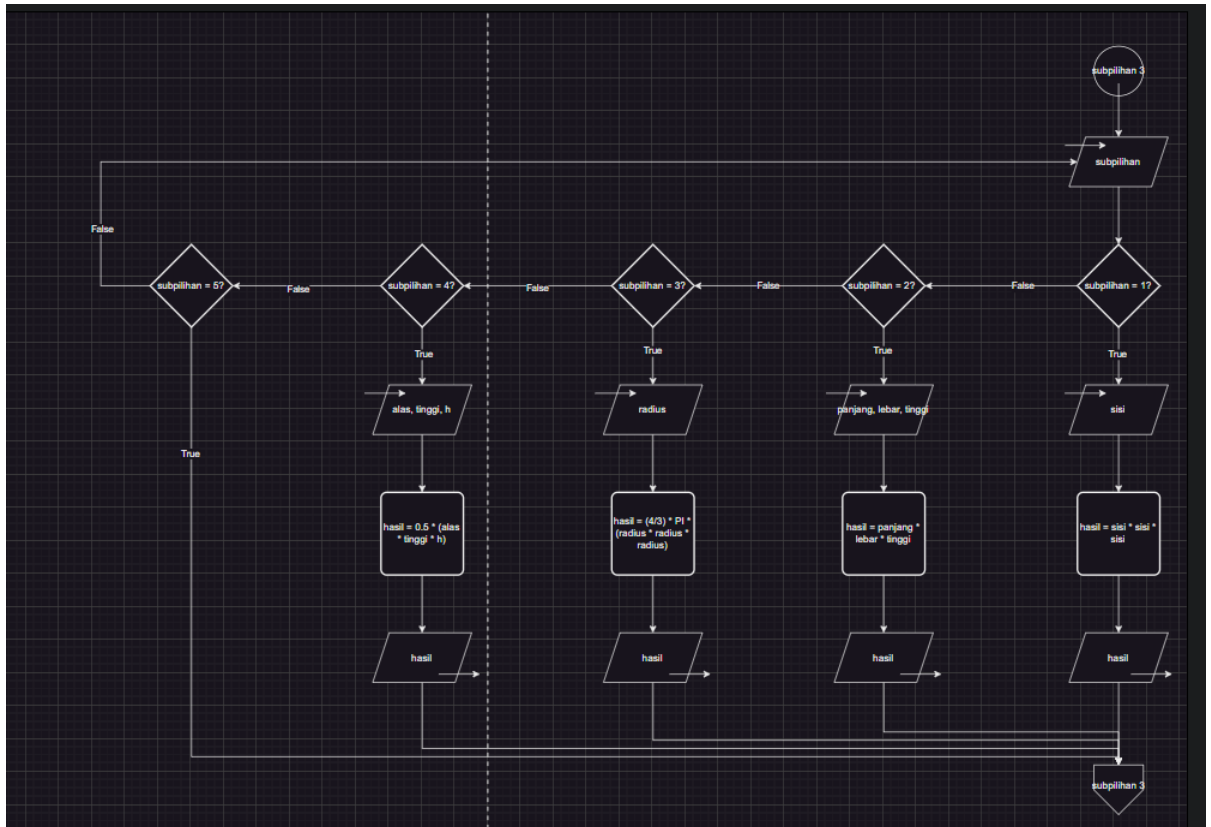
User akan diminta untuk menginput pilihan untuk menghitung keliling apa, lalu akan muncul output setelah perhitungan.



Gambar 1.4 Sub-Pilihan-2

- Sub Pilihan 3

User akan diminta untuk menginput pilihan untuk menghitung Volume apa, lalu akan muncul output setelah perhitungan.



Gambar 1.5 Sub-Pilihan-3

2. Analisis Program

2.1 Deskripsi Singkat Program

Program kali ini berfungsi untuk perhitungan bangun datar dan bangun ruang, terdapat menu untuk memilih untuk menghitung Luas, Keliling, dan Volume. program tidak akan berhenti hingga user memintanya untuk logout.

2.2 Penjelasan Alur & Algoritma

Program dimulai dengan mengimpor library iostream untuk input/output dan cmath untuk operasi matematika seperti pow. PI didefinisikan sebagai konstanta dengan nilai 3.14, dan USERNAME serta PASSWORD diinisialisasi sebagai string yang berisi username dan password yang valid. Di dalam fungsi main(), program meminta pengguna untuk

memasukkan username dan password. Jika username atau password yang dimasukkan tidak sesuai dengan yang telah ditentukan, program akan berhenti. Jika login berhasil, program akan menampilkan menu utama yang memungkinkan pengguna untuk memilih antara menghitung luas, keliling, volume, atau logout.

Jika pengguna memilih untuk menghitung luas, program akan menampilkan submenu untuk memilih bentuk geometri yang ingin dihitung luasnya, seperti persegi, persegi panjang, lingkaran, atau segitiga. Program akan meminta input yang diperlukan (misalnya, sisi untuk persegi, panjang dan lebar untuk persegi panjang, jari-jari untuk lingkaran, atau alas dan tinggi untuk segitiga), menghitung luas berdasarkan rumus yang sesuai, dan menampilkan hasilnya. Jika pengguna memilih untuk menghitung keliling, program akan menampilkan submenu serupa untuk memilih bentuk geometri. Program akan meminta input yang diperlukan (misalnya, sisi untuk persegi, panjang dan lebar untuk persegi panjang, jari-jari untuk lingkaran, atau tiga sisi untuk segitiga), menghitung keliling berdasarkan rumus yang sesuai, dan menampilkan hasilnya.

Jika pengguna memilih untuk menghitung volume, program akan menampilkan submenu untuk memilih bentuk geometri yang ingin dihitung volumenya, seperti kubus, balok, bola, atau prisma segitiga. Program akan meminta input yang diperlukan (misalnya, sisi untuk kubus, panjang, lebar, dan tinggi untuk balok, jari-jari untuk bola, atau alas, tinggi segitiga, dan tinggi prisma untuk prisma segitiga), menghitung volume berdasarkan rumus yang sesuai, dan menampilkan hasilnya. Jika pengguna memilih untuk logout, program akan berhenti. Selama pengguna tidak memilih logout, program akan terus menampilkan menu utama setelah setiap perhitungan selesai, memungkinkan pengguna untuk melakukan perhitungan lain atau logout. Setiap kali pengguna memasukkan input, program memastikan bahwa input tersebut valid (misalnya, tidak negatif) dengan menggunakan loop do-while yang akan terus meminta input sampai input yang valid diberikan. Setelah perhitungan selesai, hasilnya ditampilkan, dan pengguna dapat memilih untuk kembali ke menu sebelumnya atau melanjutkan dengan perhitungan lain. Program berakhir ketika pengguna memilih logout.

3. Source Code

A. Fitur login.

Program akan dimulai dengan user yang diminta untuk memasukkan username dan password yang harus sesuai dengan yang ada di database yaitu “Ahmad Habibi” untuk Username dan “2409106104” untuk Passwordnya. Jika user menginput yang tidak sesuai dengan database maka program akan terhenti otomatis.

Source Code :

```
int main() {
    string username, password;
    cout << "\n== SISTEM LOGIN ==\n";
    cout << "Username: ";
    getline(cin, username);
    cout << "Password: ";
    cin >> password;
    cin.ignore();
    if (username != USERNAME || password != PASSWORD) {
        cout << "Login gagal!";
        return 0;
    }
}
```

B. Menu Utama dan fitur menghitung luas.

Setelah login berhasil, program menampilkan menu utama yang berisi empat pilihan:

1. Hitung Luas: Untuk menghitung luas berbagai bentuk geometri.
2. Hitung Keliling: Untuk menghitung keliling berbagai bentuk geometri.
3. Hitung Volume: Untuk menghitung volume berbagai bentuk geometri.
4. Logout: Untuk keluar dari program.

Jika pengguna memilih "Hitung Luas", program akan menampilkan submenu untuk memilih bentuk geometri yang ingin dihitung luasnya. Pilihan yang tersedia adalah:

1. Persegi: Program meminta input panjang sisi, kemudian menghitung luas menggunakan rumus $sisi * sisi$.
2. Persegi Panjang: Program meminta input panjang dan lebar, kemudian menghitung luas menggunakan rumus $panjang * lebar$.
3. Lingkaran: Program meminta input jari-jari, kemudian menghitung luas menggunakan rumus $PI * r * r$.

4. Segitiga: Program meminta input alas dan tinggi, kemudian menghitung luas menggunakan rumus $0.5 * \text{alas} * \text{tinggi}$.

5. Kembali: Untuk kembali ke menu utama.

Program menggunakan loop do-while untuk memastikan input yang dimasukkan valid (tidak negatif). Setelah perhitungan selesai, hasilnya ditampilkan, dan pengguna dapat memilih untuk kembali ke submenu atau melanjutkan dengan perhitungan lain.

Source Code :

```
int pilihan;
do {
    cout << "\n== SISTEM HITUNG ==\n1. Hitung Luas\n2. Hitung
Keliling\n3. Hitung Volume\n4. Logout\nPilihan: ";
    cin >> pilihan;
    if (pilihan == 1) {
        int subPilihan;
        do {
            cout << "\n== HITUNG LUAS ==\n1. Persegi\n2. Persegi
Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\nPilihan: ";
            cin >> subPilihan;
            double hasil;
            if (subPilihan == 1) {
                double s;
                do {
                    cout << "Masukkan sisi: ";
                    cin >> s;
                } while (s < 0);
                hasil = s * s;
            } else if (subPilihan == 2) {
                double p, l;
                do {
                    cout << "Masukkan panjang: ";
                    cin >> p;
                    cout << "Masukkan lebar: ";
                    cin >> l;
                } while (p < 0 || l < 0);
                hasil = p * l;
            } else if (subPilihan == 3) {
                double r;
                do {
                    cout << "Masukkan jari-jari: ";
                    cin >> r;
                } while (r < 0);
                hasil = PI * r * r;
            }
        }
    }
}
```

```

    } else if (subPilihan == 4) {
        double a, t;
        do {
            cout << "Masukkan alas: ";
            cin >> a;
            cout << "Masukkan tinggi: ";
            cin >> t;
        } while (a < 0 || t < 0);
        hasil = 0.5 * a * t;
    }
    if (subPilihan != 5) cout << "Hasil luas: " << hasil <<
    "\n";
} while (subPilihan != 5);
}

```

C. Menu menghitung keliling.

Jika pengguna memilih "Hitung Keliling", program akan menampilkan submenu untuk memilih bentuk geometri yang ingin dihitung kelilingnya. Pilihan yang tersedia adalah:

1. Persegi: Program meminta input panjang sisi, kemudian menghitung keliling menggunakan rumus $4 * \text{sisi}$.
2. Persegi Panjang: Program meminta input panjang dan lebar, kemudian menghitung keliling menggunakan rumus $2 * (\text{panjang} + \text{lebar})$.
3. Lingkaran: Program meminta input jari-jari, kemudian menghitung keliling menggunakan rumus $2 * \text{PI} * r$.
4. Segitiga: Program meminta input tiga sisi segitiga, kemudian menghitung keliling menggunakan rumus $\text{sisi1} + \text{sisi2} + \text{sisi3}$.
5. Kembali: Untuk kembali ke menu utama.

Seperti pada fitur luas, program memastikan input yang dimasukkan valid menggunakan loop do-while. Hasil perhitungan keliling ditampilkan, dan pengguna dapat memilih untuk kembali ke submenu atau melanjutkan dengan perhitungan lain.

Source Code :

```
else if (pilihan == 2) {
    int subPilihan;
    do {
        cout << "\n== HITUNG KELILING ==\n1. Persegi\n2. Persegi
Panjang\n3. Lingkaran\n4. Segitiga\n5. Kembali\nPilihan: ";
        cin >> subPilihan;
        double hasil;
        if (subPilihan == 1) {
            double s;
            do {
                cout << "Masukkan sisi: ";
                cin >> s;
            } while (s < 0);
            hasil = 4 * s;
        } else if (subPilihan == 2) {
            double p, l;
            do {
                cout << "Masukkan panjang: ";
                cin >> p;
                cout << "Masukkan lebar: ";
                cin >> l;
            } while (p < 0 || l < 0);
            hasil = 2 * (p + l);
        } else if (subPilihan == 3) {
            double r;
            do {
                cout << "Masukkan jari-jari: ";
                cin >> r;
            } while (r < 0);
            hasil = 2 * PI * r;
        } else if (subPilihan == 4) {
            double a, b, c;
            do {
                cout << "Masukkan sisi 1: ";
                cin >> a;
                cout << "Masukkan sisi 2: ";
                cin >> b;
                cout << "Masukkan sisi 3: ";
                cin >> c;
            } while (a < 0 || b < 0 || c < 0);
            hasil = a + b + c;
        }
        if (subPilihan != 5) cout << "Hasil keliling: " << hasil <<
"\n";
    } while (subPilihan != 5);
}
```

D. Fitur menghitung Volume dan fitur logout.

Jika pengguna memilih "Hitung Volume", program akan menampilkan submenu untuk memilih bentuk geometri yang ingin dihitung volumenya. Pilihan yang tersedia adalah:

1. Kubus: Program meminta input panjang sisi, kemudian menghitung volume menggunakan rumus $sisi * sisi * sisi$.
2. Balok: Program meminta input panjang, lebar, dan tinggi, kemudian menghitung volume menggunakan rumus $panjang * lebar * tinggi$.
3. Bola: Program meminta input jari-jari, kemudian menghitung volume menggunakan rumus $(4.0 / 3.0) * PI * r^3$.
4. Prisma Segitiga: Program meminta input alas, tinggi segitiga, dan tinggi prisma, kemudian menghitung volume menggunakan rumus $0.5 * alas * tinggiSegitiga * tinggiPrisma$.
5. Kembali: Untuk kembali ke menu utama.

Program kembali menggunakan loop do-while untuk memastikan input yang dimasukkan valid. Hasil perhitungan volume ditampilkan, dan pengguna dapat memilih untuk kembali ke submenu atau melanjutkan dengan perhitungan lain. Jika pengguna memilih "Logout", program akan berhenti. Ini dilakukan dengan mengakhiri loop do-while yang mengontrol menu utama, sehingga program selesai dan keluar.

Source Code :

```
else if (pilihan == 3) {
    int subPilihan;
    do {
        cout << "\n== HITUNG VOLUME ==\n1. Kubus\n2.
Balok\n3. Bola\n4. Prisma Segitiga\n5. Kembali\nPilihan: ";
        cin >> subPilihan;
        double hasil;
        if (subPilihan == 1) {
            double s;
            do {
```

```

        cout << "Masukkan sisi: ";
        cin >> s;
    } while (s < 0);
    hasil = s * s * s;
} else if (subPilihan == 2) {
    double p, l, t;
    do {
        cout << "Masukkan panjang: ";
        cin >> p;
        cout << "Masukkan lebar: ";
        cin >> l;
        cout << "Masukkan tinggi: ";
        cin >> t;
    } while (p < 0 || l < 0 || t < 0);
    hasil = p * l * t;
} else if (subPilihan == 3) {
    double r;
    do {
        cout << "Masukkan jari-jari: ";
        cin >> r;
    } while (r < 0);
    hasil = (4.0 / 3.0) * PI * pow(r, 3);
} else if (subPilihan == 4) {
    double a, t, h;
    do {
        cout << "Masukkan alas segitiga: ";
        cin >> a;
        cout << "Masukkan tinggi segitiga: ";
        cin >> t;
        cout << "Masukkan tinggi prisma: ";
        cin >> h;
    } while (a < 0 || t < 0 || h < 0);
    hasil = 0.5 * a * t * h;
}
if (subPilihan != 5) cout << "Hasil volume: " <<
hasil << "\n";
    } while (subPilihan != 5);
}
} while (pilihan != 4);
return 0;
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

Pada skenario ini program dijalankan senormalnya untuk menghitung luas persegi dengan menginput angka 4 sebagai sisinya.

```
== SISTEM LOGIN ==  
Username: Ahmad Habibi  
Password: 2409106104  
  
== SISTEM HITUNG ==  
1. Hitung Luas  
2. Hitung Keliling  
3. Hitung Volume  
4. Logout  
Pilihan: 1  
  
== HITUNG LUAS ==  
1. Persegi  
2. Persegi Panjang  
3. Lingkaran  
4. Segitiga  
5. Kembali  
Pilihan: 1  
Masukkan sisi: 4  
Hasil luas: 16
```

Gambar 4,1 Hasil-Uji-Coba-Berhasil

2. Skenario 2

Pada skenario ini, user menginput username ataupun password yang salah sehingga program otomatis berhenti.

```
== SISTEM LOGIN ==  
Username: Ahmad Habibi  
Password: 2409106030  
Login gagal!
```

Gambar 4.2 Hasil-Uji-Coba-Gagal

4.2 Hasil Output

Contoh hasil output dari perhitungan Volume Bola.

```
== SISTEM LOGIN ==  
Username: Ahmad Habibi  
Password: 2409106104  
  
== SISTEM HITUNG ==  
1. Hitung Luas  
2. Hitung Keliling  
3. Hitung Volume  
4. Logout  
Pilihan: 3  
  
== HITUNG VOLUME ==  
1. Kubus  
2. Balok  
3. Bola  
4. Prisma Segitiga  
5. Kembali  
Pilihan: 3  
Masukkan jari-jari: 24  
Hasil volume: 57876.5
```

Gambar 4.3 Output

5. Git

1. Inisialisasi Repo git

```
PS D:\dump tugas\project 0\praktikum-apl> git init
Reinitialized existing Git repository in D:/dump tugas/project 0/praktikum-apl/.git/
```

Gambar 5.1 Inisialisasi-Repository

Pada gambar 5.1, perintah git init dijalankan di dalam direktori D:\dump tugas\project 0\praktikum-apl, yang menghasilkan pesan bahwa Git telah berhasil diinisialisasi dalam folder tersebut. Jika pertama kali menjalankan perintah ini, Git akan membuat folder tersembunyi bernama .git, yang berisi semua informasi terkait version control dari proyek tersebut.

2. Menambahkan File untuk dicommit

```
PS D:\dump tugas\project 0\praktikum-apl> git add .
```

Gambar 5.2 Menambahkan-File

Setelah repository Git diinisialisasi, langkah berikutnya adalah menambahkan file ke dalam staging area. Ini adalah tahap sebelum file benar-benar disimpan dalam repository Git. Perintah yang digunakan untuk menambahkan file ke staging area. Pada gambar 5.2, perintah git add . digunakan. Titik (.) dalam perintah ini berarti semua file dalam direktori saat ini akan ditambahkan ke staging area.

3. Commit

```
PS D:\dump tugas\project 0\praktikum-apl> git commit -m "Finish Post Test 1"
[main 38ac0bc] Finish Post Test 1
3 files changed, 0 insertions(+), 0 deletions(-)
rename post-test/post-test-apl-1/{24091060104_AhmadHabibi-PT-1.cpp => 24091060104-AhmadHabibi-PT-1.cpp} (100%)
rename post-test/post-test-apl-1/{24091060104_AhmadHabibi-PT-1.exe => 24091060104-AhmadHabibi-PT-1.exe} (100%)
rename post-test/post-test-apl-1/{24091060104_AhmadHabibi-PT-1.cpp => 24091060104-AhmadHabibi-PT-1.cpp} (100%)
rename post-test/post-test-apl-1/{24091060104_AhmadHabibi-PT-1.exe => 24091060104-AhmadHabibi-PT-1.exe} (100%)
rename post-test/post-test-apl-1/{24091060104_AhmadHabibi-PT-1.pdf => 24091060104-AhmadHabibi-PT-1.pdf} (100%)
```

Gambar 5.3 Commit

Commit adalah proses menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository Git. Commit membantu mendokumentasikan perubahan yang dilakukan pada proyek dengan memberikan pesan deskriptif. Pada gambar 5.3, perintah git commit -m "Finish Post Test 1" digunakan untuk menyimpan perubahan dengan pesan commit "Finish Post Test 1".

4. push untuk mengupload file ke github

```
PS D:\dump tugas\project 0\praktikum-apl> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 353 bytes | 176.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/aeBeeBoy/praktikum-apl.git
   c5bfa10..38ac0bc  main -> main
PS D:\dump tugas\project 0\praktikum-apl>
```

Gambar 5.4 Push

Setelah melakukan commit, langkah selanjutnya adalah mengunggah perubahan dari repository lokal ke repository GitHub. Pada gambar 5.4, perintah `git push` digunakan untuk mengunggah commit terbaru ke repository GitHub. Perintah ini akan mengunggah perubahan ke *branch* yang aktif (default-nya adalah `main` atau `master`).