

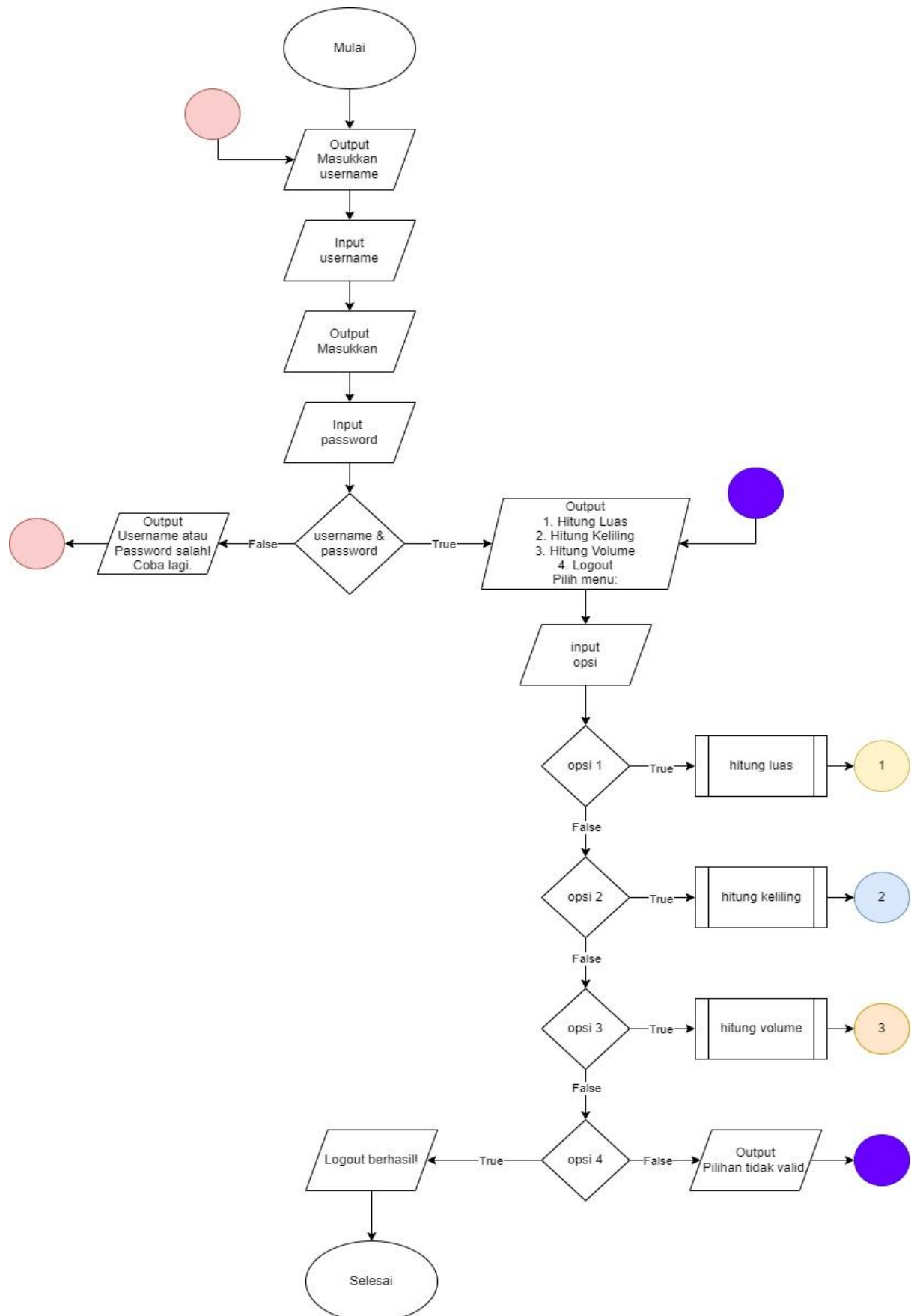
LAPORAN PRAKTIKUM
POSTTEST 1
ALGORITMA PEMROGRAMAN LANJUT

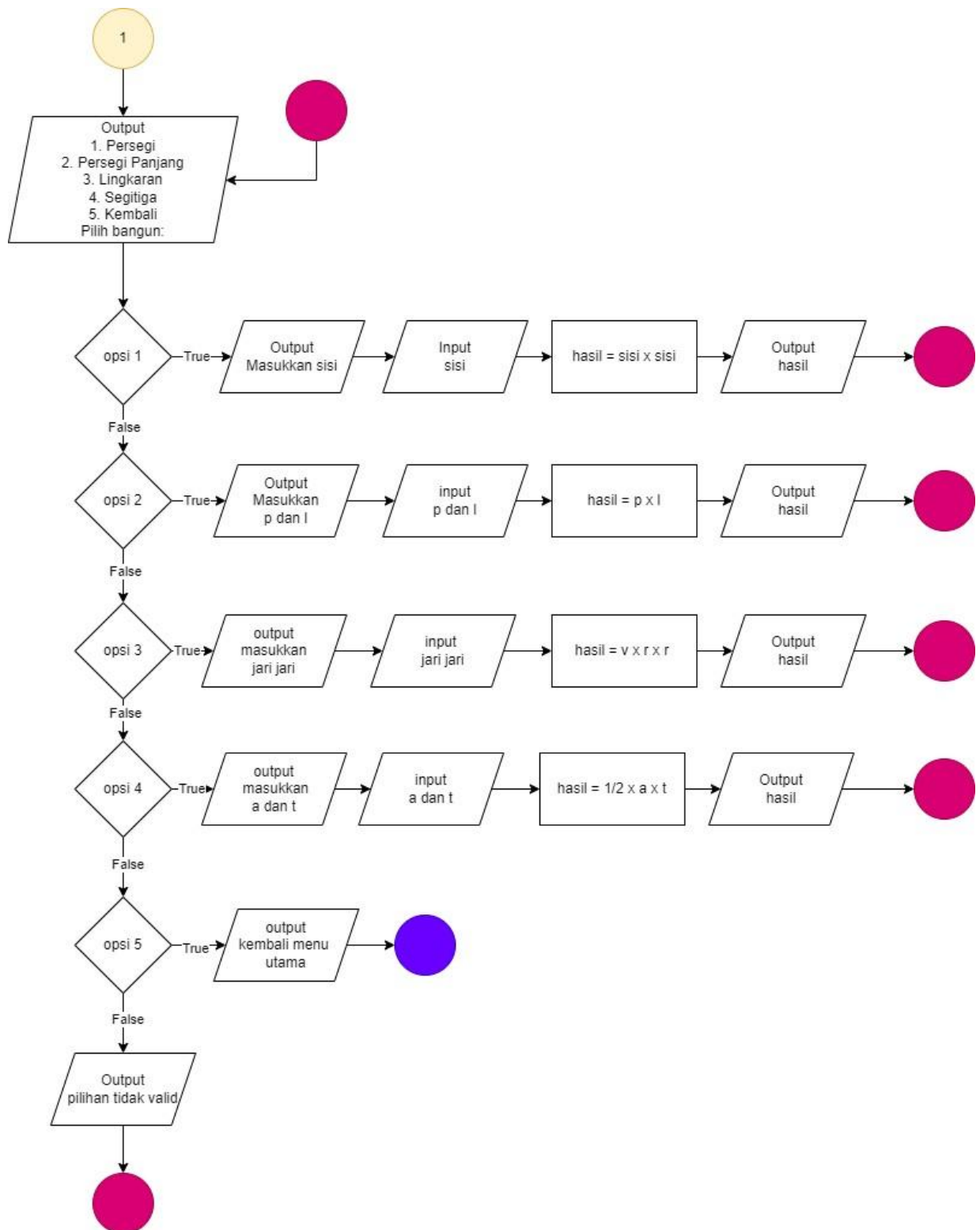


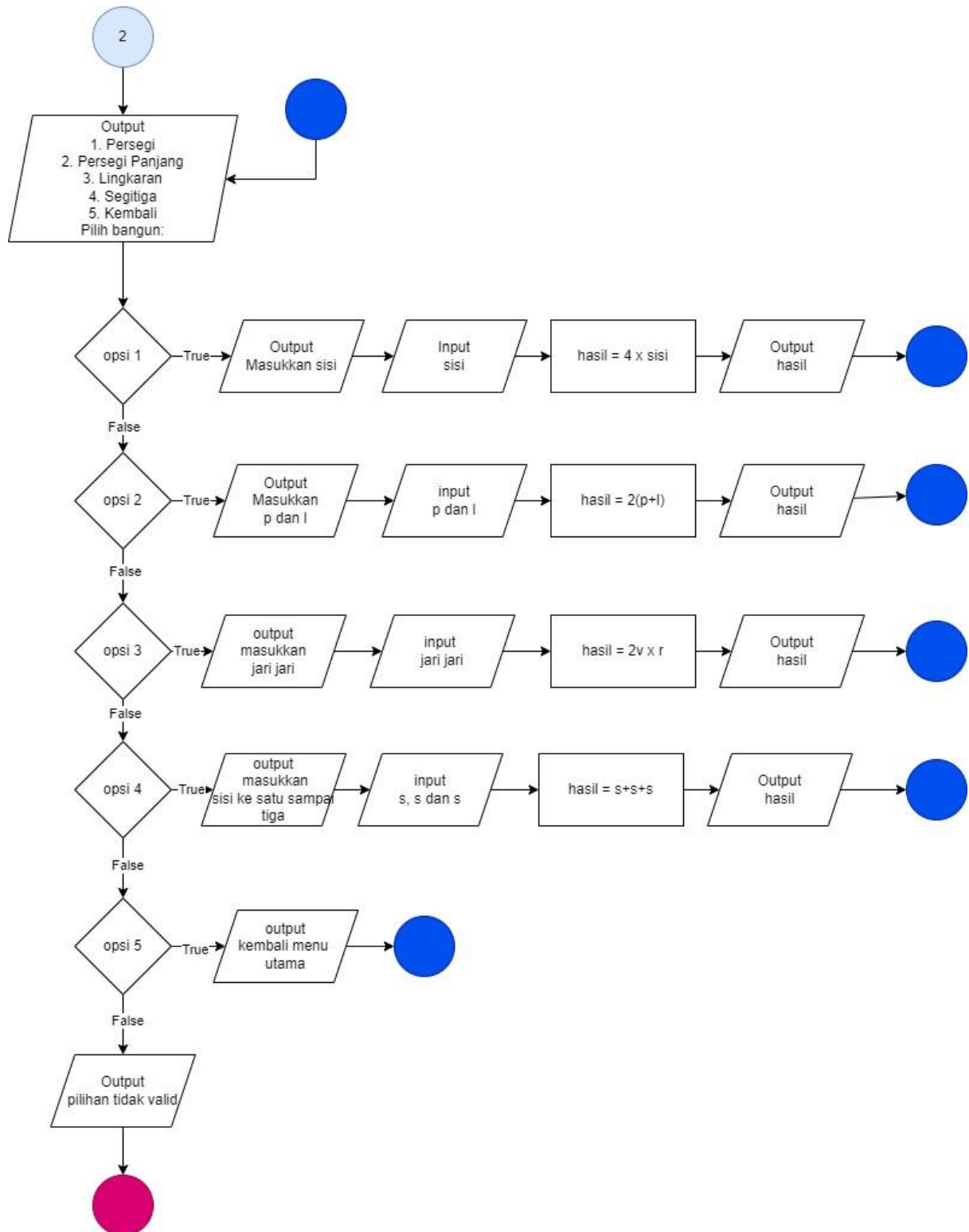
Disusun oleh:
Muhammad Kurniawan (2409106091)
Kelas (C'24)

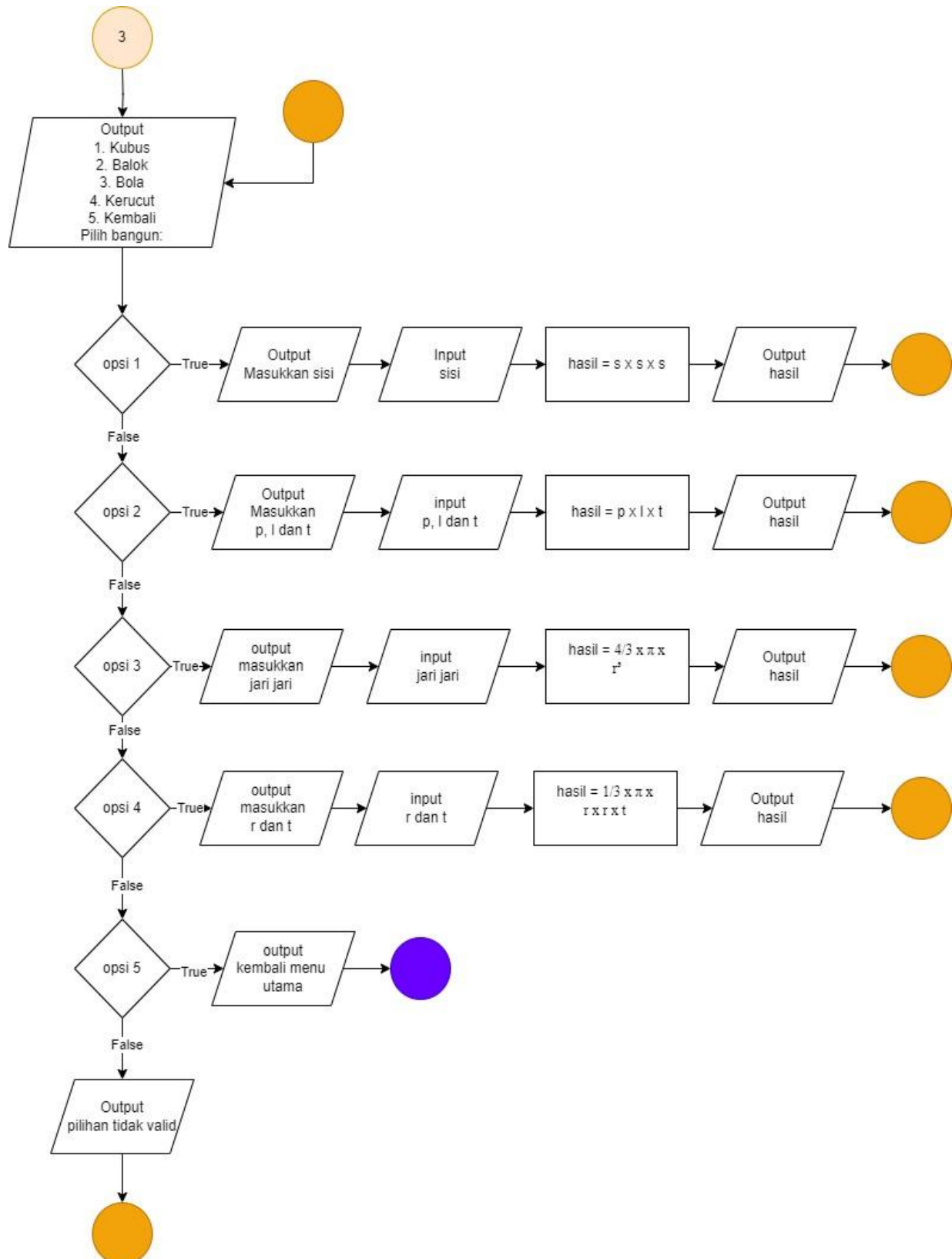
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart









2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan sistem perhitungan bangun datar dan bangun ruang dengan fitur login dan validasi input.

Tujuan:

1. Membuat sistem perhitungan bangun datar dan bangun ruang
2. Memastikan hanya pengguna dengan kredensial yang benar yang dapat mengakses sistem.
3. Menerapkan validasi input supaya hanya bisa memasukkan angka yang valid
4. Menyediakan pilihan menu dan bisa memilih sesuai perhitungan yang dibutuhkan 5.

Program akan terus berjalan sampai pengguna memilih untuk keluar

Fungsi:

1. Mempermudah pengguna dalam menghitung luas, keliling, dan volume berbagai bangun
2. Menghindari kesalahan input dengan validasi angka
3. Meningkatkan pemahaman konsep C++

2.2 Penjelasan Alur & Algoritma

Login Sistem:

- Pengguna diminta memasukkan username dan password.
- Jika salah, pengguna diminta mengulangi hingga memasukkan yang benar.
- Jika benar, pengguna masuk ke menu utama.

Menampilkan Menu Utama:

- Program menampilkan pilihan:
 - 1 Hitung Luas
 - 2 Hitung Keliling
 - 3 Hitung Volume
 - 4 Logout (Keluar dari program)
- Pengguna memilih salah satu opsi.

Jika pengguna memilih opsi perhitungan:

- Program menampilkan sub-menu sesuai pilihan (luas, keliling, atau volume).
- Pengguna memilih bangun yang ingin dihitung (misalnya: persegi, lingkaran, bola, dll.).
- Program meminta pengguna memasukkan input angka (panjang, jari-jari, tinggi, dll.).
- Program memvalidasi input, memastikan tidak ada nilai negatif.
- Program menghitung dan menampilkan hasil perhitungan.
- Setelah selesai, pengguna bisa memilih lagi atau kembali ke menu utama.

Jika pengguna memilih Logout (4):

- Program menampilkan pesan "Logout berhasil!" lalu berhenti.

3. Source Code

A. Fitur Validasi Input

Fitur ini berguna untuk memvalidasi nilai yang dimasukkan untuk menghitung tidak boleh negatif

Source Code:

```
double validateInput(string message) {    double value;    do {  
    cout << message;        cin >> value;        if (value < 0) {  
    cout << "Input tidak boleh negatif! Silakan masukkan ulang.\n";        }  
    } while (value < 0);  
    return value;  
}
```

B. Fitur Login

Fitur ini digunakan untuk mengidentifikasi user atau pengguna

Source code

```
int main() {    string username, password;  
    string correctUsername = "MuhammadKurniawan";  
    string correctPassword = "2409106091";  
    do {        cout << "Masukkan  
Username: ";  
        getline(cin, username); //getline agar bisa menangkap input kosong  
  
        cout << "Masukkan Password: ";  
        getline(cin, password);  
  
        if (username.empty() || password.empty()) {            cout << "Username  
dan Password tidak boleh kosong! Silakan masukkan kembali.\n";  
        } else if (username != correctUsername || password != correctPassword) {  
            cout << "Username atau Password salah! Coba lagi.\n";  
        }  
    } while (true);  
}
```



```
    } while (username.empty() || password.empty() || username !=  
correctUsername || password != correctPassword);
```

C. Fitur Menu Utama

Fitur ini adalah menu utama dari program, dimana terdapat pilihan untuk menghitung luas, keliling, volume serta pilihan untuk logout dari program

Source code

```
int pilihan;    do {                cout << "\n== SISTEM HITUNG BANGUN DATAR  
DAN BANGUN RUANG ==\n";          cout << "1. Hitung Luas\n2. Hitung  
Keliling\n3. Hitung Volume\n4. 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";                }  
        } while (pilihan != 4);  
  
        return 0;  
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1

skenario pertama yang saya uji adalah pada saat login, jika kita tidak mengisi username ataupun password maka terdapat pemberitahuan username dan password tidak boleh kosong.

2. Skenario 2

skenario kedua yang saya uji coba adalah validasi input sesuai dengan fitur yang saya buat, input yang dimasukkan tidak boleh bernilai negatif

4.2 Hasil Output

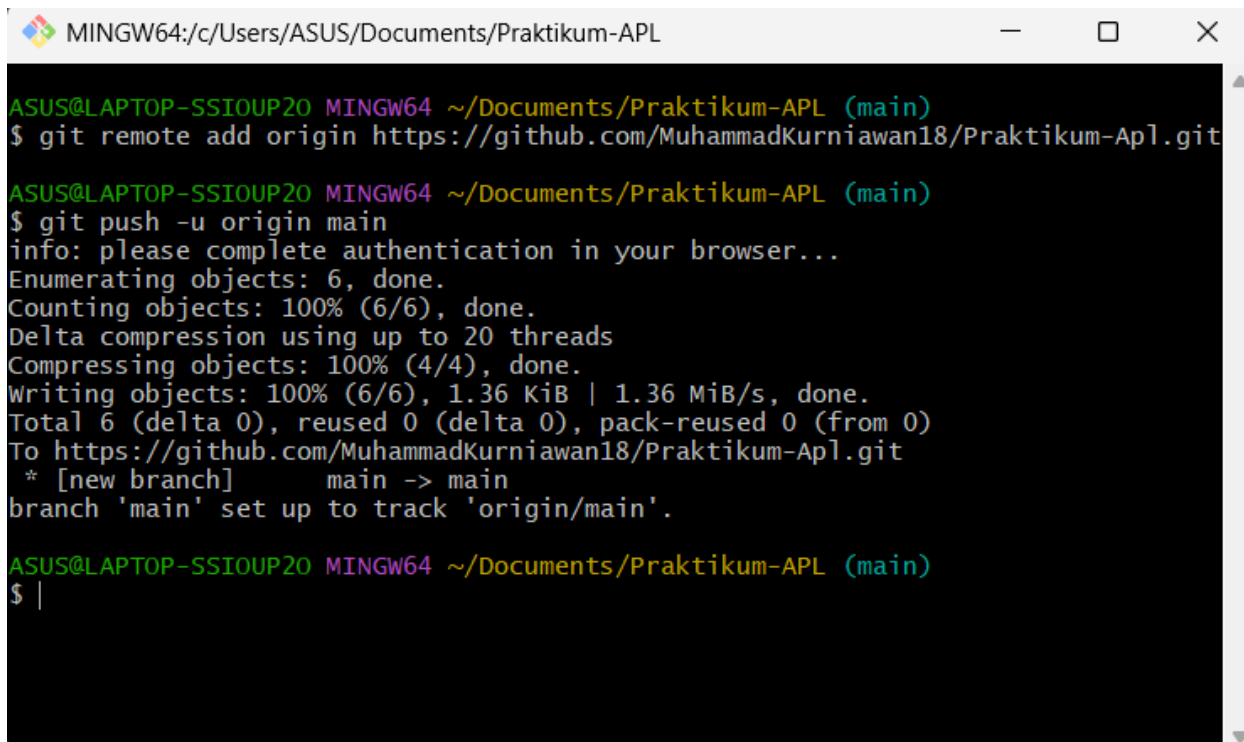
```
Masukkan Username:  
Masukkan Password:  
Username dan Password tidak boleh kosong! Silakan masukkan kembali.  
Masukkan Username: kurniawan  
Masukkan Password:  
Username dan Password tidak boleh kosong! Silakan masukkan kembali.  
Masukkan Username: 
```

Gambar 4.1 Uji coba skenario 1

```
== HITUNG LUAS ==  
1. Persegi  
2. Persegi Panjang  
3. Lingkaran  
4. Segitiga  
5. Kembali  
Pilih bangun: 1  
Masukkan sisi persegi: -3  
Input tidak boleh negatif! Silakan masukkan ulang.  
Masukkan sisi persegi: -6  
Input tidak boleh negatif! Silakan masukkan ulang.  
Masukkan sisi persegi: 2  
Luas: 4
```

Gambar 4.2 Uji coba skenario 2

5. Git



```
MINGW64:/c/Users/ASUS/Documents/Praktikum-APL
ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL (main)
$ git remote add origin https://github.com/MuhammadKurniawan18/Praktikum-Apl.git

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/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 20 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (6/6), 1.36 KiB | 1.36 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/MuhammadKurniawan18/Praktikum-Apl.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

ASUS@LAPTOP-SSIOUP20 MINGW64 ~/Documents/Praktikum-APL (main)
$ |
```

1. `git remote add origin https://github.com/MuhammadKurniawan18/Praktikum-Apl.git`: Perintah ini menambahkan remote bernama "origin" ke repositori Git lokal. Remote ini menunjuk ke repositori di GitHub dengan URL yang diberikan. Remote digunakan untuk melacak dan berinteraksi dengan repositori jarak jauh.
2. `git push -u origin main`: Perintah ini mendorong (unggah) branch lokal bernama "main" ke remote "origin". Opsi `-u` mengatur branch lokal "main" untuk melacak branch "main" di remote "origin". Ini memungkinkan Anda menggunakan perintah `git pull` dan `git push` di masa mendatang tanpa harus menentukan remote dan branch secara eksplisit.