

# LAPORAN PROGRAM PERULANGAN

Miftahul ulum (25104410084)

## 1: PENDAHULUAN DAN TUJUAN PROGRAM

### 1.1 Latar Belakang

Program ini dibuat untuk mendemonstrasikan penggunaan struktur perulangan (*loop*) dan percabangan (*conditional statement*) dalam bahasa C++. Program menerima sejumlah data integer dari pengguna, kemudian mengelompokkan dan menjumlahkan data tersebut berdasarkan sifat genap atau ganjilnya.

### 1.2 Tujuan

- Memahami penggunaan perulangan for untuk input data berulang
- Memahami penggunaan percabangan if-else untuk pengambilan keputusan
- Memahami operasi modulus (%) untuk menentukan bilangan genap atau ganjil
- Menghitung total jumlah bilangan genap dan ganjil dari data yang dimasukkan

## 2: PENJELASAN KODE PROGRAM

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

int main() {
    int n;
    int jumGenap = 0, jumGanjil = 0;

    cout << "Masukkan jumlah data: ";
    cin >> n;

    for (int i = 0; i < n; i++) {
        int data;
        cout << "Masukkan data ke-" << i << ": ";
        cin >> data;

        if (data % 2 == 0) {
            jumGenap += data;
        }
        else {
            jumGanjil += data;
        }
    }

    cout << "Jumlah bilangan genap = " << jumGenap << "\n";
    cout << "Jumlah bilangan ganjil = " << jumGanjil << "\n";

    return 0;
}
```

### 2.1 Struktur Program

Program terdiri dari:

- Deklarasi variabel
- Input jumlah data

- Perulangan untuk memasukkan data
- Percabangan untuk menentukan genap/ganjil
- Penghitungan jumlah
- Output hasil

## 2.2 Penjelasan Kode Program

Program dimulai dengan mendeklarasikan variabel `n` untuk menyimpan jumlah data, serta `jumGenap` dan `jumGanjil` yang diinisialisasi dengan nilai 0 untuk menampung total bilangan genap dan ganjil.

Pengguna diminta memasukkan jumlah data yang akan diproses. Kemudian program menggunakan perulangan `for` untuk meminta input data sebanyak `n` kali. Setiap data yang dimasukkan akan diperiksa apakah genap atau ganjil menggunakan operator modulus `%`. Jika `data % 2 == 0`, maka data tersebut adalah genap dan ditambahkan ke `jumGenap`. Jika tidak, data tersebut ganjil dan ditambahkan ke `jumGanjil`.

Setelah semua data diproses, program menampilkan total jumlah bilangan genap dan ganjil.

## 3: CONTOH HASIL OUTPUT DENGAN DATA 0-9

### 3.1 Contoh Eksekusi Program

Input:

```
Masukkan jumlah data: 10
Masukkan data ke-0: 0
Masukkan data ke-1: 1
Masukkan data ke-2: 2
Masukkan data ke-3: 3
Masukkan data ke-4: 4
Masukkan data ke-5: 5
Masukkan data ke-6: 6
Masukkan data ke-7: 7
Masukkan data ke-8: 8
Masukkan data ke-9: 9
```

Output:

```
Jumlah bilangan genap = 20
Jumlah bilangan ganjil = 25
```

### 3.2 Analisis Output

**Bilangan Genap:** 0, 2, 4, 6, 8

**Total Genap** =  $0 + 2 + 4 + 6 + 8 = 20$

**Bilangan Ganjil:** 1, 3, 5, 7, 9

**Total Ganjil** =  $1 + 3 + 5 + 7 + 9 = 25$

### 3.3 Verifikasi

- **Total semua data:**  $20 + 25 = 45$
- **Jumlah bilangan genap:** 5
- **Jumlah bilangan ganjil:** 5

### 3.4 Kesimpulan

Program berhasil memproses 10 data input (angka 0-9) dengan benar. Program mampu mengidentifikasi dan mengelompokkan bilangan genap dan ganjil, kemudian menghitung total masing-masing kelompok dengan akurat. Perulangan for bekerja dengan efektif untuk menangani input data yang berulang, sedangkan percabangan if-else berfungsi dengan baik untuk klasifikasi data.