

PROGRAM C++

Perhitungan Bilangan Ganjil dan Genap

Menggunakan Perulangan do...while

Laporan Teknis Program Praktik

Oleh: **Falik Fabio Aulia**

Kelas: **Teknik Informatika Malam C**

Pendahuluan

Latar Belakang

Program ini dibuat untuk menyelesaikan permasalahan perhitungan jumlah bilangan ganjil dan genap dari sejumlah data input. Program menggunakan bahasa pemrograman C++ dengan struktur perulangan **do...while** untuk memproses data secara berulang dan efisien.

8 Tujuan Program

- Menerima input jumlah data dari pengguna melalui keyboard
- Menerima input bilangan secara berulang sesuai jumlah data
- Mengklasifikasikan setiap bilangan menjadi ganjil atau genap
- Menghitung jumlah total bilangan ganjil dan genap
- Menampilkan hasil perhitungan dengan format terstruktur

Syarat Program

- Syarat 1 Input data harus dari keyboard menggunakan cin
- Syarat 2 Bilangan mulai dari 0 sampai 10
- Syarat 3 Menggunakan perulangan for atau do...while

🔍 Analisis & Flowchart

📊 Alur Program dari Flowchart

Berdasarkan flowchart yang diberikan, berikut adalah alur eksekusi program:

- **Start** → Awal program dimulai
- **Input Data** → Masukkan jumlah data (n) dan inisialisasi $\text{jumGenap} = 0$, $\text{jumGanjil} = 0$
- **Perulangan** → Loop untuk input setiap bilangan
- **Decision (if data % 2 == 0)** → Kondisi untuk mengecek bilangan
 - **True (Genap)** → $\text{jumGenap} = \text{jumGenap} + \text{data}$
 - **False (Ganjil)** → $\text{jumGanjil} = \text{jumGanjil} + \text{data}$
- **Output** → Tampilkan hasil jumlah ganjil dan genap
- **End** → Akhir program

Logika Kondisi

Operator Modulo (%) digunakan untuk menentukan sisa pembagian

- Jika $\text{data \% 2} == 0$ → Bilangan GENAP (habis dibagi 2, sisa 0)
- Jika $\text{data \% 2} == 1$ → Bilangan GANJIL (tidak habis dibagi 2, sisa 1)

Penjelasan Kode C++

Header dan Namespace

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

- **#include <iostream>** → Menginclude library untuk input/output (cin dan cout)
- **using namespace std;** → Menggunakan namespace standar agar tidak perlu menulis std:: sebelum cin dan cout

Deklarasi Variabel

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

Variabel	Tipe Data	Fungsi	Nilai Awal
n	int	Menyimpan jumlah data yang akan diinput	-
data	int	Menyimpan nilai bilangan pada setiap iterasi	-
i	int	Counter untuk perulangan	0
jumGenap	int	Akumulator jumlah bilangan genap	0
jumGanjil	int	Akumulator jumlah bilangan ganjil	0

Penjelasan Kode C++ (Lanjutan)

Input Jumlah Data

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

- **cout** → Menampilkan pertanyaan ke layar (output)
- **cin >> n** → Menerima input jumlah data dari keyboard (input)

Perulangan do...while

```
do { cout << "Input data ke-" << (i + 1) << ": "; cin >> data; if (data % 2 == 0) { jumGenap = jumGenap + data; } else { jumGanjil = jumGanjil + data; } i++; } while (i < n);
```

Penjelasan Perulangan do...while

- **do { }** → Blok kode yang akan dijalankan terlebih dahulu
- **cout << "Input data ke-" << (i + 1)** → Menampilkan nomor urutan (i+1 agar dimulai dari 1, bukan 0)
- **cin >> data** → Menerima input bilangan dari pengguna
- **if (data % 2 == 0)** → Mengecek apakah bilangan genap
- **jumGenap = jumGenap + data** → Jika genap, tambahkan ke jumGenap
- **jumGanjil = jumGanjil + data** → Jika ganjil, tambahkan ke jumGanjil
- **i++** → Increment counter (i bertambah 1)

- **while (i < n)** → Cek kondisi setelah eksekusi blok kode

Output Hasil

```
cout << "\n=== HASIL ===" << endl; cout << "Jumlah total bilangan ganjil : " <<
jumGanjil << endl; cout << "Jumlah total bilangan genap : " << jumGenap <<
endl; return 0; }
```

Output & Hasil Program

Contoh Eksekusi 1: Input Campuran

Skenario: Input 5 bilangan dengan campuran ganjil dan genap

Masukkan jumlah data: 5 Input data ke-1: 2 Input data ke-2: 3 Input data ke-3: 5 Input data ke-4: 7 Input data ke-5: 9

Output:

=== HASIL === Jumlah total bilangan ganjil : 24 Jumlah total bilangan genap : 2

Penjelasan:

- Bilangan **Genap**: 2 → Total = **2**
- Bilangan **Ganjil**: 3 + 5 + 7 + 9 = **24**

Contoh Eksekusi 2: Semua Bilangan Genap

Input: 4, 6, 8, 10

Output: Ganjil = 0 | Genap = 28 (4+6+8+10)

Contoh Eksekusi 3: Semua Bilangan Ganjil

Input: 1, 3, 5, 7, 9

Output: Ganjil = 25 (1+3+5+7+9) | Genap = 0

Kesimpulan & Evaluasi

Keberhasilan Program

- ✓ Program berhasil menerima input dari keyboard dengan cin
- ✓ Perulangan do...while bekerja dengan sempurna untuk memproses data berulang
- ✓ Operator modulo (%) berhasil mengklasifikasikan bilangan ganjil dan genap
- ✓ Akumulator jumGanjil dan jumGenap menghitung total dengan akurat
- ✓ Output terstruktur dan mudah dipahami pengguna

Karakteristik Program

- **Fleksibel** → Dapat menerima jumlah data yang berbeda-beda
- **Efisien** → Menggunakan perulangan untuk menghemat kode
- **User-Friendly** → Tampilan interaktif dengan pesan yang jelas
- **Akurat** → Hasil perhitungan sesuai dengan input yang diberikan

Poin Penting

- Perulangan **do...while** mengeksekusi blok kode minimal 1 kali sebelum cek kondisi
- Operator **%** sangat berguna untuk mendeteksi bilangan ganjil/genap
- Variabel harus diinisialisasi dengan benar (jumGenap = 0, jumGanjil = 0)
- Counter **i** harus di-increment agar perulangan tidak infinite loop

Kesimpulan Akhir

Program C++ yang telah dibuat berhasil menyelesaikan permasalahan perhitungan bilangan ganjil dan genap sesuai dengan semua syarat yang diberikan. Program menggunakan struktur yang jelas, logika yang sederhana namun efektif, dan hasil output yang akurat. Dengan pemahaman yang baik tentang perulangan, kondisi, dan operator, program ini dapat dikembangkan lebih lanjut untuk kasus-kasus yang lebih kompleks.