



Modul 4

Pengulangan (1)

Praktikum Dasar Pemrograman

Jurusan Teknik Informatika Fakultas Sains dan Teknologi

Dosen:

Rifqi Syamsul Fuadi, ST.

A. Tujuan

1. Mahasiswa memahami manfaat dan penggunaan pengulangan.
2. Mahasiswa mampu menyelesaikan masalah menggunakan pengulangan *while* dan *do-while*.

B. Pokok Bahasan

Pengulangan *while* dan *do-while*.

C. Dasar Teori

Pengulangan

Pengulangan (*looping*) adalah bagian yang bertugas melakukan kegiatan/proses secara berulang-ulang sesuai dengan yang diinginkan. Terdapat 4 bagian pengulangan:

- a. Inisialisasi, berfungsi sebagai keadaan awal sebelum pengulangan dilakukan.
- b. Kondisi pengulangan, berfungsi untuk menentukan pengulangan akan dilakukan atau tidak. Selama kondisi pengulangan terpenuhi/benar, maka pengulangan akan dilakukan. Sebaliknya jika kondisi pengulangan tidak terpenuhi/salah, maka pengulangan dihentikan.
- c. Proses, yaitu bagian yang dilakukan berulang-ulang selama kondisi pengulangan terpenuhi atau bernilai benar.
- d. Pengatur iterasi, berfungsi untuk mengatur langkah iterasi. Langkah iterasi bisa dilakukan secara menaik atau menurun.

Terdapat tiga jenis pengulangan:

- *while*
- *do-while*
- *for*

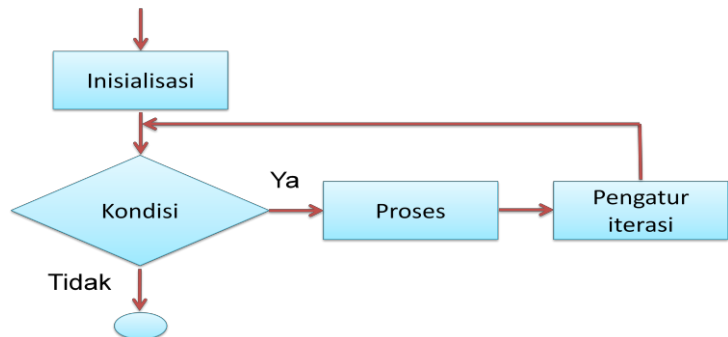
Pengulangan *while*

Pengulangan *while* akan melakukan pemeriksaan kondisi terlebih dahulu. Jika kondisi tidak terpenuhi, ada kemungkinan pengulangan tidak dikerjakan sama sekali. Selama

kondisinya terpenuhi/benar, maka pengulangan akan dikerjakan. Pengulangan *while* biasanya digunakan untuk kasus dimana banyaknya pengulangan yang akan dilakukan tidak diketahui secara pasti.

Bentuk pengulangan *while*:

```
inisialisasi;  
while(kondisi)  
{  
    proses;  
    pengatur iterasi;  
}
```



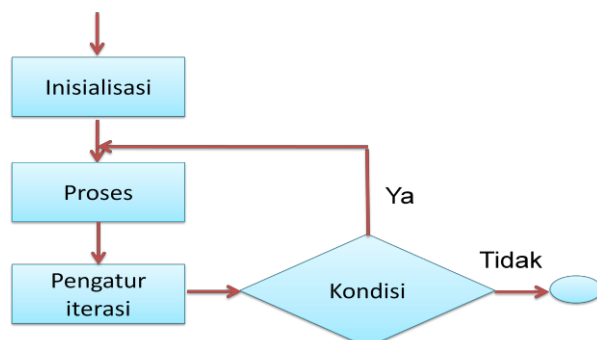
Flowchart pengulangan *while*

Pengulangan *do-while*

Pengulangan *do-while* tidak melakukan pemeriksaan kondisi di awal, sehingga pengulangan akan dilakukan minimal 1 kali sekalipun kondisinya tidak terpenuhi. Perbedaan dengan *while*, kondisi pengulangan *do-while* berada di bawah blok pengulangan. Sama dengan *while*, pengulangan *do-while* biasanya digunakan untuk kasus dimana banyaknya pengulangan yang akan dilakukan tidak diketahui secara pasti.

Bentuk pengulangan *do-while*:

```
inisialisasi;  
do  
{  
    proses;  
    pengatur iterasi;  
} while(kondisi);
```



Flowchart pengulangan *do-while*

D. Praktikum

Program 4.1

	<i>while</i>
1	int i;
2	//inisialisasi
3	i = 1;
4	//kondisi pengulangan
5	while(i <= 10)
6	{
7	//proses
8	cout << "MinGW C++\n";
9	//pengatur iterasi
10	i++; //sama dgn i = i + 1
11	}

Program 4.2

	<i>do-while</i>
1	int i;
2	//inisialisasi
3	i = 1;
4	do
5	{
6	//proses
7	cout << "MinGW C++\n";
8	//pengatur iterasi
9	i++; //sama dgn i = i + 1
10	} while(i <= 10);
11	//kondisi pengulangan

Program 4.3

	<i>while</i>
1	int k;
2	k = 1;
3	cout << "Bilangan 1 - 10:\n";
4	while(k <= 10)
5	{
6	cout << k << " ";
7	k++;
8	}
9	

Program 4.4

	<i>do-while</i>
1	int j, bil;
2	j = 5;
3	cout << "Masukkan 5 buah bilangan:\n";
4	do
5	{
6	cout << "Bil ke-" << j << " : ";
7	cin >> bil;
8	j--;
9	} while(j >= 1);

Program 4.4

	<i>do-while</i>
1	int bil;
2	char lagi;
3	do
4	{
5	cout << "Input bilangan: ";
6	cin >> bil;
7	cout << "Mau lagi?(Y/T): ";
8	cin >> lagi;
9	}while(lagi == 'Y');

Program 4.5

	<i>while</i>
1	int bil;
2	bil = 1;
3	while(bil != 999)
4	{
5	cout << "Masukkan bilangan : ";
6	cin >> bil;
7	}

Latihan 4.1

Buatlah program dengan tampilan berikut:

```
Bilangan genap positif kurang dari 20:  
2 4 6 8 10 12 14 16 18
```

Latihan 4.2

Buatlah program yang menerima sebuah inputan n , dimana n merupakan batas akhir dari bilangan yang akan ditampilkan.

```
Masukkan nilai n: 7 (input)  
1 2 3 4 5 6 7
```

Latihan 4.3

Buatlah program dengan tampilan berikut:

```
Bilangan ganjil positif sebanyak n:  
Masukkan nilai n: 7 (input)  
1 3 5 7 9 11 13
```

Latihan 4.4

Buatlah program yang menginputkan bilangan sebanyak n , kemudian menampilkan hasil penjumlahan semua bilangan tersebut.

E. Tugas

1. Buatlah program yang menerima inputan n buah bilangan, kemudian hitung rata-ratanya. Gunakan *while*!
2. Buatlah program untuk menampilkan tahun kabisat antara tahun ke- x sampai tahun ke- y . Gunakan *while*! (x dan y diinput).
3. Modifikasi program tentang berat badan ideal (tugas modul 3 no. 3) sehingga dapat dilakukan berulang-ulang selama *user* menginginkan untuk mengulangi lagi. Gunakan *do-while*!
4. Buat rangkuman materi tentang pengulangan *for* dan 1 contoh program yang menggunakan pengulangan *for*.

F. Materi Praktikum Selanjutnya

Pengulangan *for*.