

MODUL 5

PERULANGAN

A. TUJUAN

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu:

- Mengenal dan memahami penggunaan statement perulangan
- Membuat program sederhana dengan menerapkan konsep perulangan

B. PETUNJUK

1. Awali setiap aktivitas anda dengan doa, agar anda lancer dalam belajar.
2. Pahami tujuan, dasar teori dan latihan- latihan praktikum dengan baik.
3. Kerjakan tugas-tugas praktikum dengan baik, jujur dan sabar.
4. Tanyakan kepada asisten praktikum apabila ada hal-hala yang kurang jelas.

C. ALAT dan BAHAN

- PC/Laptop yang memenuhi kapasitas untuk pemrograman Bahasa C++
- Software C++
- Modul/handout materi yang akan di praktikkan.

D. DASAR TEORI

Dalam bahasa C++ tersedia suatu fasilitas yang digunakan untuk melakukan proses yang berulang-ulang sebanyak keinginan kita. Misalnya saja, bila kita ingin menginput dan mencetak bilangan dari 1 sampai 100 bahkan 1000, tentunya kita akan merasa kesulitan. Namun dengan struktur perulangan proses, kita tidak perlu menuliskan perintah sampai 100 atau 1000 kali, cukup dengan beberapa perintah saja. Struktur perulangan dalam bahasa C mempunyai bentuk yang bermacam-macam. Sebuah/kelompok instruksi diulang untuk jumlah pengulangan tertentu. Baik yang terdefiniskan sebelumnya ataupun tidak. Struktur pengulangan terdiri atas dua bagian:

- (1) Kondisi pengulangan yaitu ekspresi boolean yang harus dipenuhi untuk melaksanakan pengulangan;
- (2) Isi atau badan pengulangan yaitu satu atau lebih pernyataan (aksi) yang akan diulang.

❖ Statement FOR

Statement FOR digunakan untuk menyatakan perulangan. Struktur perulangan for biasa digunakan untuk mengulang suatu proses yang telah diketahui jumlah perulangannya. Dari segi penulisannya, struktur perulangan for tampaknya lebih efisien karena susunannya lebih simpel dan sederhana. Pernyataan for digunakan untuk melakukan looping. Pada umumnya looping yang dilakukan oleh for telah diketahui batas awal, syarat looping dan perubahannya. Selama kondisi terpenuhi, maka pernyataan akan terus dieksekusi. Bentuk umum perulangan for adalah sebagai berikut:

JOB SHEET TEUM

```

For ( inisialisasiNilai; SyaratPerulangan; PerubahanNilai )
{
Statement yang diulang;
}

```

- Ungkapan1 merupakan statement awal (inisialisasi)
- Ungkapan2 merupakan kondisi/syarat perulangan dilakukan
- Ungkapan3 merupakan statement control untuk perulangan
- Statement merupakan pernyataan/perintah yang dijalankan jika syarat terpenuhi.

```

for(a=1;a<=5;a++)
{
cout<<"Hello World! \n"
}

```

Perintah diatas menampilkan kalimat "Hello World!" sebanyak 5 baris.

Tanda "a=1" adalah nilai awal variabel a.
Tanda "a<=5" syarat pengulangan.
Tanda "a++" kondisi pengulangan.

```

for(huruf="Z";huruf>="A";huruf--)
{
Cout<<"Abjad " << huruf << "\n";
}

```

Perintah diatas menampilkan abjad Z – A.

Perhatikan perintah operator --, operator – digunakan untuk decrement

Contoh lain :

```
for (angka = 1; angka <= 6; angka+=2)
{
cout << "Isi dari angka = " << angka << endl;
}
```

***Perintah di atas akan menampilkan angka 1, 3, 5.
Mengapa terjadi demikian? Perhatikan perintah angka+=2!***

Di bawah ini adalah program untuk mencetak bilangan genap yang kurang dari n (n diperoleh dari input).

```
#include <iostream.h>
#include <conio.h>
void main(){
int bil, n;
cout << "Masukkan n = ";
cin >> n;
for (bil = 0; bil < n; bil++)
{
if (bil % 2 == 0) cout << bil << " ";
}
}
```

❖ Statement WHILE

Perulangan WHILE banyak digunakan pada program yang terstruktur. Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui. Proses perulangan akan terus berlanjut selama kondisinya bernilai benar ($\neq 0$) dan akan berhenti bila kondisinya bernilai salah ($= 0$).

Berikut gambaran umumnya:

```
while (syarat pengulangan){
    statement yang dijalankan;
    statement control:
}
```

Dua perintah di bawah ini adalah identik.

```
for (a = 1; a <= 5; a++)
{
    cout << "Hello world \n";
}
```

```
a = 1;
while (a <= 5){
    cout << "Hello world \n";
    a++;
}
```



Jika Anda menggunakan WHILE, pastikan bahwa suatu saat bagian kondisi sampai bernilai FALSE. Apabila tidak, proses perulangan akan terus berjalan selamanya.

Contoh program dibawah ini digunakan untuk menjumlahkan data angka. Angka yang akan dijumlahkan diinputkan satu-persatu. Proses pemasukan data angka akan berhenti ketika dimasukkan angka 1-5. Setelah itu tampil hasil penjumlahannya.

JOBSHIELD

```

#include <iostream.h>
#include <conio.h>
void main()
{
int data, jumlah, cacah;
jumlah = 0;
data = 0;
cacah = 0;
while (data != -1)
{
cout << "Masukkan data angka : ";
cin >> data;
jumlah += data;
cacah++;
}
cout << "Jumlah data adalah : " << jumlah << endl;
cout << "Rata-rata : " << jumlah/cacah;
}

```

Kondisi dalam WHILE dapat menggunakan operator logika:

```

while ((kondisi1) && (kondisi2))
{
    Blok Pernyataan
}

```

❖ Statement DO-WHILE

Perintah do...while hampir sama dengan WHILE sebelumnya. Gambaran secara umumnya adalah:

```

do
{
    Blok Pernyataan
}
while (kondisi);

```

Perbedaan dengan WHILE sebelumnya yaitu bahwa DO WHILE statement dilakukan terlebih dahulu baru kemudian mengecek kondisinya. Sedangkan WHILE statement, kondisi dicek terlebih dahulu kemudian statement perulangan dijalankan. Akibat dari hal ini dalam penggunaan DO WHILE statemen minimal terdapat 1x perulangan. Sedangkan WHILE statement dimungkinkan perulangan tidak pernah terjadi sama sekali yaitu ketika kondisinya bernilai FALSE.

```
a = 1;
do
{
cout << "Hello world \n";
a++;
}
while(a==0)
```

Perintah di atas akan muncul satu buah Hello World. Bandingkan dengan yang berikut ini:

```
a = 1;
while(a==0)
{
cout << "Hello world \n";
a++;
}
```

Perintah di atas sama sekali tidak menampilkan Hello World, karena kondisinya langsung FALSE.

E. LATIHAN

1. LAT1

Buatlah program C++ untuk membuat tampilan segitiga seperti di bawah ini menggunakan FOR melalui inputan tinggi segitiga.

```

*
**
***
****
*****

```

Dst

```

D:\SEMESTER 1\DASAR PEMROGRAMAN KOMPUTER\PRAKTIKUM\PROJECT PR...
-----Program Segitiga-----
-----Ananda Putri Syaviri-----
-----130533608243-----
MASUKKAN NILAI
9
*
**
***
****
*****
*****
*****
*****
*****
*****
Lakukan pengulangan kembali <HAI/IE>:H
MASUKKAN NILAI
3
*
**
***
Lakukan pengulangan kembali <HAI/IE>:I
_

```

2. LAT2

Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan **WHILE STATEMENT**


```

MENU PILIHAN
1. Baca Data
2. Ubah Data
3. Hapus Data
4. Exit
Pilihan Anda (1/2/3/4) ? ...

```

Apabila dipilih menu no 1, maka akan tampil teks "Anda memilih menu 1". Demikian pula untuk menu 2 dan 3. Kemudian setelah itu muncul teks "Tekan ENTER untuk kembali ke menu utama". Artinya begitu kita tekan ENTER menu pilihan akan muncul kembali, dst. Akan tetapi bila yang dipilih menu 4 (EXIT), program langsung berhenti.

3. LAT3

Buatlah program untuk mencetak (gunakan perulangan while atau for)

```

Malang
alang
lang
ang
ng
g

```

F. TUGAS PRAKTIKUM

1. TGP1

Buatlah program menentukan nama hari

A = SENIN C = RABU E = JUM'AT G = AHAD

B = SELASA D = KAMIS F = SABTU

Masukkan Kode Hari (A..G) :

2. TGP2

Buatlah program dengan C++ untuk menampilkan semua penyelesaian persamaan $x + y + z = 20$. Dengan x, y, z bilangan bulat ≥ 0

```

-----
x y    z
-----
0 0    20
0 1    19
0 2    18
dst

```

G. TUGAS RUMAH

1. Buatlah program untuk menampilkan bilangan ganjil dari kecil ke besar sampai batas tertentu sampai menemukan break, program selesai.
2. Buatlah program untuk rekening tabungan di bank sebesar Rp. 50.000,- (Saldo awal). Selanjutnya ia dapat menyetor atau mengambil tabungannya. Buatlah program dengan C++ untuk keperluan transaksi di bank tsb. Tampilan utamanya seperti berikut:

```

-----
PT. BANK SYARIAH INDEPENDEN
-----

```

Saldo :

Menu Transaksi

1. Setor Tabungan

2. Ambil Tabungan

3. Exit

Pilihan menu (1/2/3) ? ...

Ketentuan:

Bank membuat kebijakan bahwa saldo minimum yang harus disisakan di tabungan sebesar Rp.25.000,-