

MODUL

4

Perulangan (**Looping**)

Modul **Praktikum**
C++
Pemrograman Dasar

Semester Genap 2017/2018

MODUL 4

PERULANGAN (LOOPING)

A. Tujuan

Setelah mempelajari bab ini diharapkan mahasiswa akan mampu :

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

B. Dasar Teori

1. Definsi Perulangan

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.

1. Statement For

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 :

```
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 << " ";
    }
}

```

2. 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 di bawah ini digunakan untuk menjumlahkan sejumlah data angka. Angka yang akan dijumlahkan diinputkan satu-persatu. Proses pemasukan data angka akan berhenti ketika dimasukkan angka -1 . Setelah itu tampil hasil penjumlahannya.

```

#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 juga dapat menggunakan **operator logika**. Misalnya:

```

while((kondisi 1) && (kondisi2))
{
    Blok Pernyataan;
}

```

3. Statement Do-While

Perintah DO ... WHILE hampir sama dengan WHILE sebelumnya. Gambaran secara umum:

```

do
{
    Blok Pernyataan;
}
while(kondisi);

```

Perbedaan dengan WHILE sebelumnya yaitu bahwa pada DO WHILE statement perulangannya dilakukan terlebih dahulu baru kemudian di cek kondisinya. Sedangkan WHILE kondisi dicek dulu baru kemudian statement perulangannya dijalankan. Akibat dari hal ini adalah dalam DO WHILE minimal terdapat 1x perulangan. Sedangkan WHILE dimungkinkan perulangan tidak pernah terjadi yaitu ketika kondisinya langsung bernilai FALSE.

Contoh:

```
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.

Latihan Praktikum

1. Latihan 1

Buatlah program untuk menampilkan bilangan bulat 1 s/d 10

```
1  #include <iostream.h>
2  #include <conio.h>
3
4  int main()
5  {
6      cout << "Menampilkan bilangan bulat 1-10 \n";
7
8      for(int a=1; a<=10; a++){
9          cout << a << endl;
10     }
11
12     getch();
13 }
```

2. Latihan 2

Buatlah program untuk menampilkan bilangan genap dari besar ke kecil dengan batas awal dan batas akhir bilangan dengan menggunakan perulangan WHILE.

```
1  #include <iostream.h>
2  #include <conio.h>
3
4  int main()
5  {
6      int awal, akhir;
7
8      cout << "Masukkan nilai awal bilangan : ";
9      cin >> awal;
10     cout << "Masukkan nilai akhir bilangan : ";
11     cin >> akhir;
12
13     cout << endl;
14     cout << "Menampilkan bilangan genap dari " << awal << " hingga " << akhir << endl;
15     while(awal<=akhir)
16     {
17         if(awal%2==0){
18             cout << awal << endl;
19         }
20
21         awal++;
22     }
23
24     getch();
25 }
```

3. Latihan 3

Buatlah program untuk menampilkan bentuk segitiga seperti output dibawah ini dengan memasukkan tinggi segitiga dan menggunakan perulangan FOR.

```

1  #include <iostream.h>
2  #include <conio.h>
3
4  int main()
5  {
6      int tinggi;
7
8      cout << "Masukkan Tinggi Segitiga : ";
9      cin >> tinggi;
10
11     cout << endl;
12     cout << "Segitiga Siku-siku dengan tinggi " << tinggi << endl;
13
14     for(int a=1; a<=tinggi; a++)    //perulangan untuk tinggi segitiga
15     {
16         for(int b=1; b<=a; b++)    //perulangan untuk menampilkan *
17         {
18             cout << "*" ;
19         }
20         cout << endl ;
21     }
22
23     getch();
24 }

```

4. Latihan 4

Buatlah simulasi menu program seperti output dibawah ini dengan menggunakan DO WHILE.

```

1  #include <iostream.h>
2  #include <conio.h>
3
4  int main()
5  {
6      int pilihan=0;
7      do{
8          cout<<"MENU PILIHAN :\n";
9          cout<<"1. Baca Data\n";
10         cout<<"2. Ubah Data\n";
11         cout<<"3. Hapus data\n";
12         cout<<"4. Exit\n";
13         cout<<"Pilihan Anda (1/2/3/4) : ";
14         cin>>pilihan;
15
16         if(pilihan==1){
17             cout<<"Anda Memilih menu 1\n";
18             cout<<"Tekan ENTER untuk kembali ke menu utama\n"<<endl;
19             getch();
20         }else if(pilihan==2){
21             cout<<"Anda Memilih menu 2\n";
22             cout<<"Tekan ENTER untuk kembali ke menu utama\n"<<endl;
23             getch();
24         }else if(pilihan==3){
25             cout<<"Anda Memilih menu 3\n";
26             cout<<"Tekan ENTER untuk kembali ke menu utama\n"<<endl;
27             getch();
28         }else if(pilihan==4){
29             system("exit");
30         }
31     }while(pilihan!=4);
32 }

```


Tugas Praktikum

1. Tugas Praktikum 1

Buatlah program untuk mencetak kata seperti dibawah ini dengan menggunakan perulangan WHILE atau FOR.

```
Masukkan Kata = Malang
Malang
alang
lang
ang
ng
g

Terminated with return code 0
Press any key to continue ...
```

2. Tugas Praktikum 2

Menampilkan konversi suhu dari Celcius ke Fahrenheit, Reamur, dan Kelvin menggunakan bentuk perulangan, seperti contoh dibawah ini :

°C	°F	°R	°K
10	??	??	??
20	??	??	??
...
...
100	??	??	??

3. Tugas Praktikum 3

Buatlah program untuk menampilkan * seperti output dibawah ini.

```
Masukkan tinggi segitiga : 7
*****
*****
*****
****
***
**
*
*
**
***
****
*****
*****
*****

Terminated with return code 0
Press any key to continue ...
```

Tugas Rumah

1. Tugas Rumah 1

Buatlah program untuk menampilkan semua penyelesaian persamaan $a + b = c$, dimana c adalah 20, dengan ketentuan bilangan a , b , dan c adalah bilangan bulat ≥ 0 seperti contoh output dibawah ini :

<u>A</u>	<u>B</u>	<u>C</u>	
0	20	20	
1	19	20	
2	18	20	
....	dan seterusnya.

2. Tugas Rumah 2

Buatlah program untuk menghitung KOMBINASI dalam matematika dengan memasukkan nilai n dan r , seperti output dibawah ini :

```
===== RUMUS KOMBINASI =====
          n!
        -----
        r! * (n-r)!

Masukkan nilai n : 10
Masukkan nilai r : 5

Hasil 10 Kombinasi 5 adalah : 252

Terminated with return code 0
Press any key to continue ...
```

3. Tugas Rumah 3

Seseorang mempunyai rekening tabungan dibank sebesar Rp. 100.000,- (saldo awal), selanjutnya dia dapat menyetorkan ataupun mengambil uang ditabungannya. Buatlah program untuk keperluan transaksi dibank tersebut. Setiap terdapat transaksi (menyetorkan / mengambil uang) maka nilai saldo akan selalu berubah dan Bank membuat kebijakan bahwa saldo minimum yang harus disisakan dalam rekening adalah Rp. 50.000,-.

Tampilan menu utamanya seperti dibawah ini :

BANK INDONESIA JAYA

Saldo : 100000

Menu Transaksi :

1. Setor Tabungan
2. Ambil Tabungan
3. Keluar

Pilihan Menu Transaksi (1/2/3) : ...