

Keg. Pembelajaran 5 : Perulangan dalam C++

1. Tujuan Kegiatan Pembelajaran

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

- 1) Mengetahui bentuk perulangan `while`, `do while` dan `for` dengan benar
- 2) Menjelaskan prinsip kerja bentuk-bentuk perulangan dengan tepat.
- 3) Dapat menggunakan bentuk-bentuk perulangan dalam aplikasinya pada pembuatan program secara tepat
- 4) Dapat mengembangkan bentuk-bentuk perulangan dalam pemrograman secara benar

2. Uraian Materi

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.

Perintah atau notasi dalam struktur pengulangan adalah :

- 1) Pernyataan **while**
- 2) Pernyataan **do..while**
- 3) Pernyataan **for**
- 4) Pernyataan **continue dan break**
- 5) Pernyataan **go to**

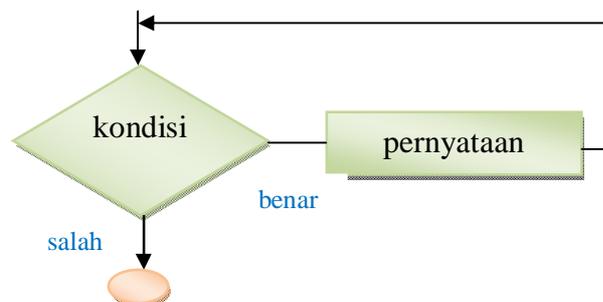
(1) Struktur Perulangan " WHILE "

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 ($=0$).

Bentuk Umumnya :

```
while (kondisi)
{
    Pernyataan ;
}
```

Pengujian ungkapan pada while dilakukan sebelum bagian pernyataan, Oleh karena itu ada kemungkinan bagian pernyataan pada while tidak dijalankan sama sekali, yaitu kalau kondisi yang pertama kali bernilai salah. Perlihatkan gambar flowchart berikut :



Gambar. 21. Diagram flowchart Perulangan dengan While

Catatan :

Jika menggunakan while pastikan bahwa pada suatu saat ungkapan pada while bernilai salah. Jika tidak demikian pernyataan yang mengikutinya akan dijalankan selamanya.

Berikut contoh program perulangan dengan while untuk menampilkan " C++ " sebanyak 10 kali.

```

//Program menampilkan " C++ " sebanyak 10 kali dengan while
# include <iostream.h>
# include <conio.h>
void main()
{
    int i;          // sebagai variabel pencacah untuk menyatakan
                  // jumlah tulisan sebanyak 10 kali.

    clrscr ();
    i = 0          // mula-mulai I diisi dengan 0

    while ( i < 10)
    {
        cout << " c++ " << endl;
        i ++;      // menaikkan pencacah sebesar 1
    }
}

```

(2) Struktur do-while

Pada dasarnya struktur perulangan do...while sama saja dengan struktur while, hanya saja pada proses perulangan dengan while, seleksi berada di while yang letaknya di atas sementara pada perulangan do...while, seleksi while berada di bawah batas perulangan. Jadi dengan menggunakan struktur do...while sekurang-kurangnya akan terjadi satu kali perulangan.

Pada struktur *do-while* kondisi pengecekan ditempatkan di bagian akhir. Hal ini menyebabkan struktur pengulangan ini minimal akan melakukan satu kali proses walaupun kondisi yang didefinisikan tidak terpenuhi (bernilai salah). Bentuk umum dari struktur *do-while*

```

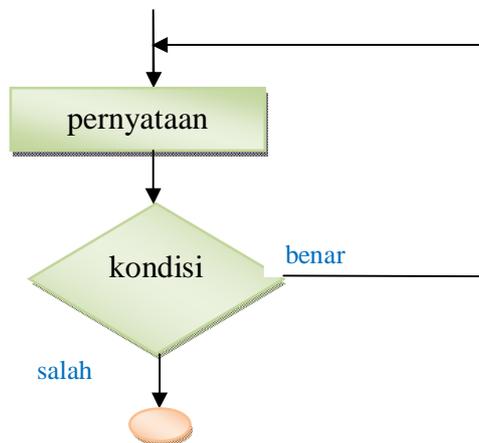
do
{
    pernyataan 1;
    pernyataan 2;
    ....
    pernyataan n ;
}while (ungkapan)

```

Catatan :

- Bagian pernyataan1 hingga N dijalankan secara berulang sampai dengan ungkapan bernilai salah (=0).
- Berbeda dengan while, pengujian ungkapan dilakukan di bagian belakang (setelah bagian pernyataan).
- Dengan demikian bagian pernyataan pada pengujian do – while minimal akan dijalankan satu kali.

Perhatikan bentuk flowchart do .. while berikut ini:



Gambar 22. Diagram flowchart perulangan dengan Do - While

Berikut program modifikasi prgram sebelumnya untuk menampilkan tulisan C++ 10 kali dengan do – while.

```
//Program menampilkan " C++ " sebanyak 10 kali dengan do –  
while *  
# include <iostream.h>  
# include <conio.h>  
void main()  
{  
    int i;          // sebagai variabel pencacah untuk menyatakan  
                  jumlah tulisan sebanyak 10 kali.  
    clrscr ();  
  
    i = 0          // mula-mulai I diisi dengan 0
```

```

do
{
    cout << " c++ " << endl;
    i ++;          // menaikkan pencacah sebesar 1
} while ( i < 10);
}

```

(3) Struktur Perulangan "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 ( ungkapan1; ungkapan2; ungkapan3)
    Pernyataan;

```

Keterangan :

- Ungkapan1 merupakan pernyataan inisialisasi
- Ungkapan 2 sebagai kondisi yang menentukan pengulangan thd pernyataan
- Ungkapan 3 sebagai pengatur variabel yang digunakan di ungkapan1

Contoh program FOR untuk menampilkan bilangan genap

```

//Program FOR untuk menentukan bilangan genap, yang
//ditampilakn dari besar ke kecil menuji nol
# include <iostream.h>
# include <conio.h>
void main()
{
    int n;
    clrscr ();

```

```

cout <<" menampilkan deret bilangan genap "<<endl;
cout <<" kurang atau sama dengan n"<<endl;
cin >> n;
        // jika bilangan ganjil, maka kurangi sebesar 1
if (n % 2)
n- - ;
        // tampilkan deret bilangan genap dari besar
        //ke kecil menuju nol

for ( ; n>= 0; n-=2)
        cout << n << ' ';
}

```

Pernyataan *for* dapat berada di dalam pernyataan *for* lainnya yang biasa disebut ***nested for***.

Contoh program For di dalam for sbb :

```

//Program untuk menampilkan bentuk segitiga karakter '*'
//dengan menggunakan for di dalam for
# include <iostream.h>
# include <conio.h>
void main()
{
    int tinggi,          //menyatakan tinggi puncak
        baris,          //pencacah untuk baris
        kolom;          //pencacah untuk kolom

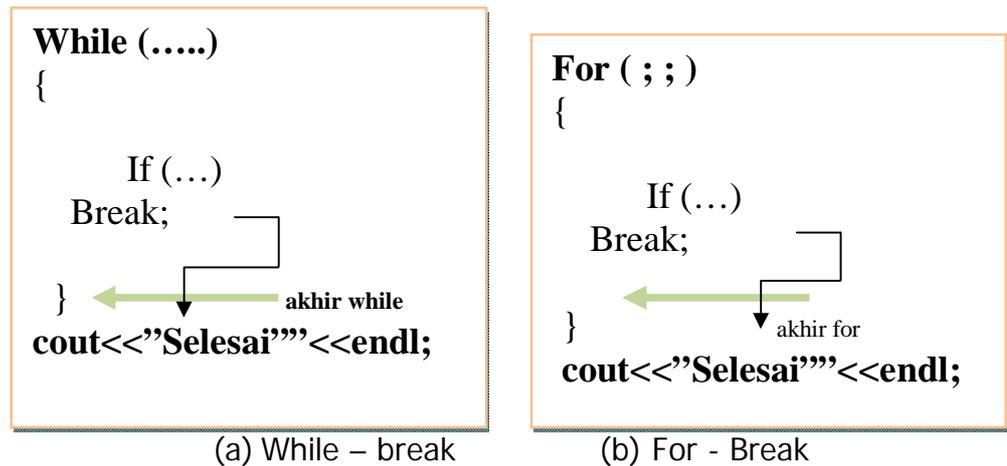
    clrscr ();
    cout << "Tinggi segi tiga : ";
    cin >> tinggi;

    cout <<endl;
    for (baris = 1; baris <= tinggi; baris++)
    {
        for (kolom = 1; kolom <= baris ; kolom++)
            cout <<'*';
        cout << endl;
    }
}

```

(4) Pernyataan Break (dalam Perulangan)

Di samping break digunakan untuk pernyataan switch, juga bisa digunakan pada For, while dan do-while. Intinya sama yakni untuk keluar dari pernyataan-pernyataan itu.



Gambar 23. Bentuk perulangan menggunakan Break

Efek dari break dari while dapat dicontohkan program sbb :

```
//Contoh program efek break pada while
# include <iostream.h>
# include <conio.h>

void main()
{
    Int bil = 1;
    clrscr ();
    while (bil <= 10)
    {
        if (bil > 5)
            break;
        cout << bil<< endl;
        bil++;
    }
}
```

Berikut contoh program efek break pada for

```
# include <iostream.h>
# include <conio.h>
void main()
{
    char karakter;
    clrscr ();
    cout <<" anda bisa ngetik apasaja" <<endl;
    cout <<" sampai tombol esc anda tekan" <<endl;

    for ( ;; )
    {
        karakter = getch();    //baca tombol
        if (karakter == 27)    //escape
            break;
    }
}
```

(5) Pernyataan continue

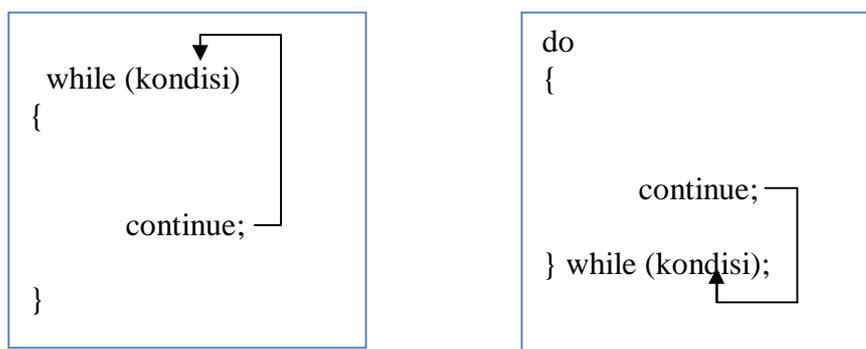
Kegunaannya kebalikannya dari break. Dipakai untuk mengarahkan eksekusi ke putaran atau iterasi berikutnya pada pernyataan perulangan. Efek dari perintah continue:

1) Pada for:

Ungkapan ke tiga pada for (ungkapan terkanan yg terletak di dalam () pada for) akan dijalankan dan kemudian ungkapan ke dua diuji lagi.

2) Pada while dan do- while :

Pengujian terhadap ungkapan pada while dilakukan kembali.



Gambar 24. Bentuk perulangan menggunakan Continue

Contoh untuk menunjukkan efek continue pada pernyataan for:

```
//Program continue pada for
# include <iostream.h>
# include <conio.h>

void main()
{
    int bil;

    clrscr ();
    for (bil = 0; bil < 10; bil++)
    {
        cout << bil << ' ';
        continue;
        cout <<" perhatikan apakah saya muncul ? " << endl;
    }
}
```

Dalam hal ini `cout <<" perhatikan apakah saya muncul ? " << endl;` tidak pernah muncul. Yang muncul adalah bilangan : 0 1 2 3 4 5 6 7 8 9

Berikut contoh penggunaan `continue` pada `while`. Dalam hal ini jika nilai yang dimasukkan melebihi 100, pernyataan `continue` akan mengarah ke `while (1)` kondisi.

```
// * Program pemakaian continue pada while
// nilai kalau lebih dari 100 akan ditolak
// program untuk menghitung nilai rerata dari masukkan
data
# include <iostream.h>
# include <conio.h>

void main()
{
    int pencacah;
    float nilai, total;

    clrscr();
    cout << " **Menghitung nilai rerata **"<<endl;
    cout <<" Akhiri dengan memasukkan nilai negatif" <<endl;
```

```

pencacah = 0;
total = 0;

while (1)          // ungkapan selalu benar
{
    cout << " nilai ke - " << pencacah+1 << " = ";
    cin >> nilai;

    if (nilai > 100)
    {
        cout << " nilai di atas diabaikan " << endl;
        continue ;
    }

    if (nilai < 0)
        break;

    pencacah++;
    total += nilai;
}

cout << " jumlah nilai      = " << pencacah << endl;
cout << " nilai rerata      = " << total/pencacah << endl;
}

```

(6) Pernyataan goto

Pernyataan **goto**, diperlukan untuk melakukan suatu lompatan ke suatu pernyataan berlabel yang ditandai dengan tanda " : " .

Bentuk umumnya adalah :

<pre> goto bawah; pernyataan1; pernyataan2; bawah : pernyataan 3; </pre>
--

Gambar 25. Bentuk perulangan menggunakan goto

Pada contoh diatas, pada saat goto ditemukan maka program akan melompat pernyataan berlabel bawah dan melakukan pernyataan 3.

Contoh :

```
// Program penggunaan goto
#include <iostream.h>
#include <conio.h>
void man ( )
{
    clrscr ( );
    cout << "Tes go to " << endl;
    goto selesai;

    cout << "Hai Aku kok Tidak disapa" <<endl;
    selesai:
    cout << " Akulah Selesai ..." << endl;
}
```

3. Rangkuman

Bentuk perulangan yang paling lazim digunakan dalam aplikasi pemrograman adalah bentuk while, do – while, dan for. Penggunaan bentuk-bentuk tersebut sangat tergantung dari tujuan dan kondisi yang ada. Namun pada prinsipnya bentuk-bentuk perulangan ini akan selalu dipergunakan dan banyak sekali manfaat dan kemudahan dalam pembuatan program aplikasi.

Di samping tiga bentuk perulangan tersebut , ada bentuk perulangan turunan, misalnya for di dalam for, ada bentuk break dan continue, kemudian ada bentuk go-to. Ini semua digunakan untuk melengkapi bentuk-bentuk perulangan. Oleh karena itu, kita harus benar-benar dapat memahami dan bisa menggunakannya secara baik dan tepat.

4. Latihan

- 1). Buatlah program untuk menampilkan bilangan ganjil dari kecil ke besar sampai batas tertentu sampai ia menemui break, program selesai.

2). Buatlah program untuk mencetak (gunakan perulangan while atau for)

Yogyakarta
ogyakarta
gyakarta
yakarta
akarta
karta
arta
rta
ta
a

Jika String Yang dimasukkan adalah "Yogyakarta".

3). Tampilkan bentuk '*' seperti contoh program sebelumnya, hanya saja tampilannya dibalik, dari kolom besar ke kolom kecil, dengan ketinggian tertentu. Contoh tampilan:

Tinggi = 4 brs
Lebar = 4 kolom

```
****  
***  
**  
*
```

4. Maenampilkan konversi suhu dari derajat celcius ke derajat fahrenheit dengan menggunakan **bentuk perulangan**. Contoh tampilan program sebagai berikut :

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