

Chapter 7 : Perulangan

Oleh : Fiftin Noviyanto

What Is a Loop?

- Loop dapat diibaratkan sebagai perputaran.
- Loop dapat digunakan untuk memberikan perintah agar suatu statement dieksekusi berkali-kali.

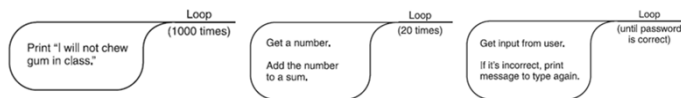
Perulangan pada program

- Loop mendeskripsikan suatu group, satu baris atau beberapa baris program dieksekusi sebanyak beberapa kali.

- Loop merupakan salah satu control statement
- Loop akan memaksa program berada pada baris/blok tertentu sampai diperbolehkan keluar dari baris / blok tersebut.

Langkah perulangan pada suatu Loop

- Contoh



Counter Statement

- counter statement adalah statement yang memungkinkan variabel menambah nilainya sendiri—biasanya dengan penambahan 1
- Contoh :

```
x = x + 1;
(3 + 1)
maka
x ← 4
```

Using the Counter Statement



Dua perbedaan jenis Loops

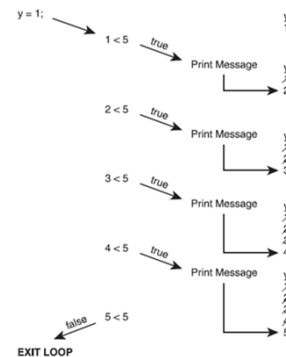
- Fixed Iterative Loops vs. Conditional Loops
- Apabila akan dicetak pesan sebanyak 100 kali pada layar, maka dapat digunakan Fixed Iterative Loops.
- Namun jika tidak diketahui berapa kali jumlah perulangan, maka menggunakan Conditional Loops. Contoh : pada permainan game. Ketika ada pertanyaan “Apakah anda akan mencoba lagi?”

Perulangan For

- Perulangan dengan **for** adalah fixed iterative loop.
- Perulangan **for** memiliki tiga kunci perintah
 - Perintah pertama adalah mendeklarasikan variabel dan memberikan nilai awal pada variabel tersebut.
 - Perintah kedua menentukan batasan.
 - Perintah ketiga perubahan nilai variabel, misalnya penambahan 1, pengurangan 1, dll.

Contoh looping dengan for

- Sebagai contoh akan dicetak pesan sebanyak lima kali.
- Gambaran :

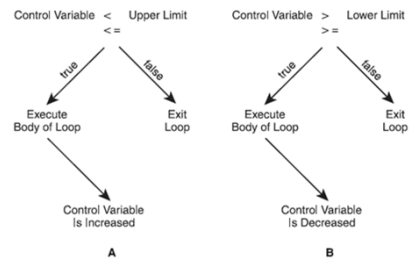


Control Variable

- Control variable adalah variable yang bekerja pada proses perulangan.
- Ketika dibandingkan dengan batas atas, nilai control variable akan dicek apakah masih dibawah batas atas.
control variable < upper limit
- Selama masih dibawah batas atas, maka body statement perulangan, masih akan dieksekusi. Setiap kali eksekusi, control variable akan bertambah.
- Namun, control variable juga dapat dibandingkan dengan batas bawah.
control variable > lower limit
- Selama control variable masih kurang dari batas bawah, maka body statement perulangan, dieksekusi. Setiap kali eksekusi, control variable akan berkurang.

The Initial Statement

- Initial statement didefinisikan pertama kali pada perulangan for.



• Contoh :

```

for ( int count =0; count < 25; count = count +1)
{
    // body of loop is here: for example
    cout << "I will not chew gum in class." << endl;
}
// program continues here after loop is executed.
  
```

• Penjelasannya???

The Conditional Loop

- Conditional loop adalah perulangan yang tidak memiliki jumlah yang tetap.
- Perulangan ini akan dijalankan selama ekspresi boolean bernilai benar.
- Setelah ekspresi boolean bernilai false, maka akan keluar dari perulangan dan melanjutkan urutan eksekusi secara sequential.

Perulangan While

- Analogi : Mahasiswa diperbolehkan mengerjakan ujian selama alarm tanda berakhir belum berbunyi.

```

⇒ while (boolean expression is true)
    ↓ {
    ↑ // the body of a loop
    ↓ statement 1;
    ↑ statement 2; etc.
    ↓ ...
    ↑ ← }
  
```

Perulangan Do . . .While

Ekspresi :

```
do
{
    execute the body of a loop
} while (boolean expression is true);
```

Contoh :

```
do
{
    cout << "Would you like to play the game again?"<<endl;
    cin >> response;
}
while (response ==yes);
```

Using a Conditional Loop with a Counter Statement

• Contoh :

```
int x;
x = 1;
while ( x < 10 )
{
    //body of the loop
    cout << x << endl;
    // counter statement to increase x
    x = x + 1;
}
```

- Angka 10 tidak akan muncul di layar, karena control variable akan berhenti ketika sampai angka 10. sedangkan ketika x=10, maka akan mengembalikan nilai FALSE.

- Contoh Counter statement pada perulangan Do..while.. :

```
int x;
x = 1;
do
{
    //body of the loop
    cout << x << endl;
    // counter statement to increase x
    x = x + 1;
}
while ( x < 10 );
```

Summary

- Another useful statement is the counter statement, which has this syntax:

```
var = var + 1.
```