

PRAKTIKUM 3

Perulangan For dan While

Tujuan Praktikum :

1. Memperkenalkan kepada mahasiswa struktur kontrol perulangan/looping
2. Memahami dan mengerti penggunaan struktur kontrol perulangan FOR ..., WHILE
3. Dapat menerapkan struktur kontrol perulangan for ..., while ... untuk menyelesaikan masalah dalam kehidupan sehari-hari

Operasi perulangan selalu dijumpai didalam bahasa pemrograman, disini akan dibahas beberapa perintah perulangan diantaranya.

Percobaan 3.1: Pengulangan dengan Statement “ for “

Perulangan yang pertama adalah **for**. Bentuk umum pernyataan **for** sebagai berikut :

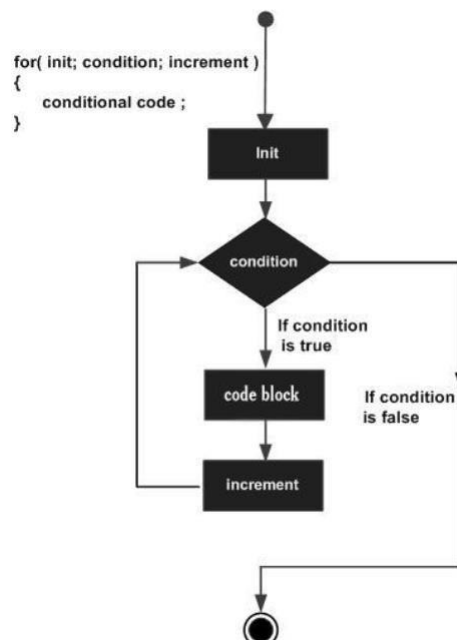
```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
```

Bila pernyataan didalam **for** lebih dari satu maka pernyataan-pernyataan tersebut harus diletakan didalam tanda kurung.

```
for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
{
    pernyataan / perintah;
    pernyataan / perintah;
    pernyataan / perintah;
}
```

Kegunaan dari masing-masing argumen **for** diatas adalah :

- **Inisialisasi**: merupakan bagian untuk memberikan nilai awal untuk variabel-variabel tertentu.
- **Syarat Pengulangan**: memegang kontrol terhadap pengulangan, karena bagian ini yang akan menentukan suatu perulangan diteruskan atau dihentikan.
- **Pengubah Nilai Pencacah**: mengatur kenaikan atau penurunan nilai pencacah.



```
#include <iostream>
using namespace std;
main() {
    for(int i=0; i<4; i++){
        cout << i;
    }
    cout<<endl;
    for(int i=3;i>0;i--){
        cout<<i;
    }
    cout << endl;
    cout << endl;

    int n, sigma = 0;
    cout << "Masukkan bilangan integer positif : "; cin >> n;
    for (int i = 1; i <= n; i++) {
        sigma += i;
    }
    cout << "Jumlah dari " << n << " bilangan pertama adalah : " << sigma << endl;
}
```

Percobaan 3.2: Pengulangan dengan Statement " while "

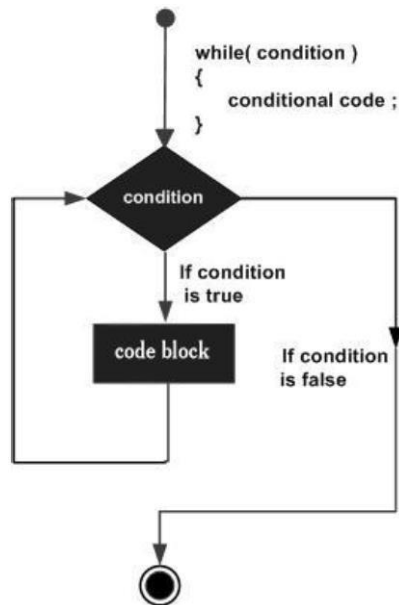
Pernyataan perulangan **while** merupakan instruksi perulangan yang mirip dengan perulangan **for**. Bentuk perulangan **while** dikendalikan oleh syarat tertentu, yaitu perulangan akan terus dilaksanakan selama syarat tersebut terpenuhi.

Bentuk umum perulangan **while**, sebagai berikut:

```
while ( syarat )
    Pernyataan / perintah ;
```

Bentuk umum perulangan **while**, dengan lebih dari perintah / pernyataan, sebagai berikut:

```
while ( syarat )
{
    Pernyataan / perintah ;
    Pernyataan / perintah ;
}
```



```

#include <iostream>
using namespace std;
main() {
    int i=1, n, sigma = 0;
    cout << "Masukkan bilangan integer positif : "; cin >> n;
    while (i<=n) {
        sigma += i;
        i++;
    }
    cout << "Jumlah dari "<< n <<" bilangan pertama adalah : "<< sigma << endl;
}
  
```

Percobaan 3.3: Pengulangan Dengan Statement "do while"

Pernyataan perulangan **do-while** merupakan bentuk perulangan yang melaksanakan perulangan terlebih dahulu dan pengujian perulangan dilakukan dibelakang.

Bentuk umum perulangan **do-while**, sebagai berikut :

```

do
    pernyataan / perintah ;
while ( syarat );
  
```

Bentuk umum perulangan **do-while**, dengan lebih dari perintah / pernyataan, sebagai berikut:

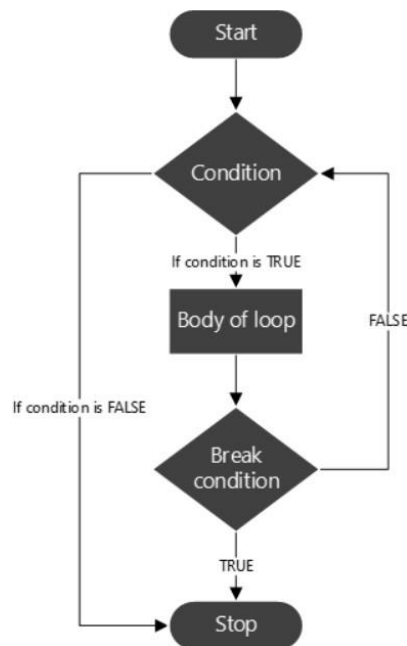
```

do
{
    Pernyataan / perintah ;
    Pernyataan / perintah ;
}
while ( syarat );
  
```

```
#include <iostream>
using namespace std;
main() {
    int n, f = 1;
    cout << "Masukkan bil integer positif : "; cin >> n;
    do {
        f *= n;
        n--;
    } while (n >= 1);
    cout << n << " faktorial adalah : " << f << endl;
}
```

Percobaan 3.4: Pernyataan Break

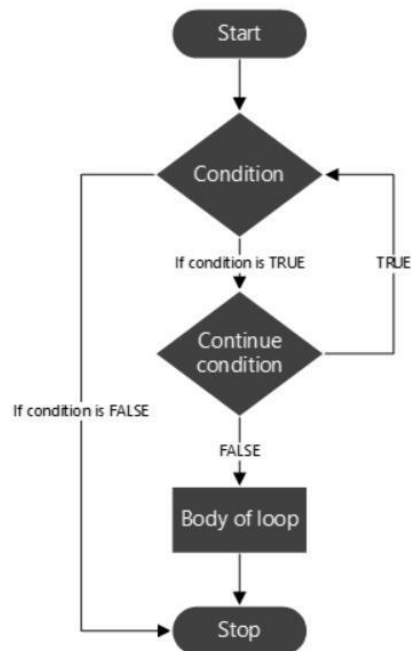
Pernyataan **break** telah dibahas pada pernyataan pengambilan keputusan **switch**. Pernyataan **break** ini berfungsi untuk keluar dari struktur **switch**. Selain itu pernyataan **break** berfungsi keluar dari perulangan (**for**, **while** dan **do-while**). Jika pernyataan **break** dikerjakan, maka eksekusi akan dilanjutkan ke pernyataan yang terletak sesudah akhir dari badan perulangan (*loop*).



```
#include <iostream>
using namespace std;
main(){
    int i=1, n, sigma = 0;
    cout << "Masukkan bilangan integer positif : "; cin >> n;
    while (1) {
        if (i > n) break; // loop berhenti disini ketika i>n
        cout << i << endl;
        i++;
    }
}
```

Percobaan 3.5: Pernyataan Continue

Pernyataan **continue** digunakan untuk mengarahkan eksekusi ke iterasi (proses) berikutnya pada **loop** yang sama, dengan kata lain mengembalikan proses yang sedang dilaksanakan ke-awal **loop** lagi, tanpa menjalankan sisa perintah dalam **loop** tersebut.



```
#include <iostream>
using namespace std;
main(){
    int n;
    for (; ; ) {
        cout << "Masukkan integer : "; cin >> n;
        if (n % 2 == 0) continue;
        else if (n % 2 > 0) break;
    }
    cout << "\tAkhir loop" << endl;
}
```



Percobaan 3.6: Nested Loop

Pernyataan Nested Loop adalah suatu perulangan didalam perulangan yang lainnya.

```
#include <iostream>
using namespace std;
main() {
    for (int i=1; i<=3; i++) {
        for (int j=1; j<=5; j++) {
            cout << " * ";
        }
        cout << endl;
    }

    cout << endl;

    int i=1;
    while(i<=3) {
        int j=1;
        while (j<=5 ) {
            cout << " * ";
            j++;
        }
        cout << endl;
        i++;
    }
}
```

Sumber/Referensi:

- (1) Munir, R. (2012). Algoritma dan Pemrograman. Jilid 1 Bandung: Penerbit Informatika.
- (2) Wirth, N. (1990). Algorithms + Data Structures = Programs. India: Prentice-Hall Of India Pvt. Limited.
- (3) Hubbard, J.R. (1996). Programming With C++, Schaum's outlines Series. USA: McGraw Hill
- (4) Suarga, M.Math., (2012). Algoritma Dan Pemrograman (Edisi 2). Yogyakarta: Penerbit Andi.
- (5) Shalahuddin, M dan AS. Rosa. (2007) Belajar Bahasa Pemrograman dengan C++ dan Java: Penerbit Informatika.
- (6) Joel Adams-Sanford Leestma-Larry Nyhoof, (1995). C++ An Introduction To Computing: Prentice-Hall, Inc
- (7) James P. Cohoon-Jack W.Davidson, McGraw-Hill, (1997). C++ Programming Design.
- (8) Budi Raharjo, (2004). Mengungkap Rahasia Pemrograman Dalam C++: Penerbit Informatika.
- (9) Abdul Kadir, (2003). Pemrograman C++, Andi, Yogyakarta
- (10) Yosua Onesimus Suheru, (2004). Trik Memecahkan Masalah Dengan Tiga bahasa Pemrograman – C++, Pascal dan Visual Basic, Gava Media, Yogyakarta