

[Home](#) / [Tutorial](#) / [C++](#) / [Pernyataan For](#)

Pernyataan For

Posted on Januari 23, 2018 | Last Modified Januari 23, 2018

```
#include <iostream>
using namespace std;

int main()
{
    int banyak;
    cout<<"Berapa banyak : ";cin>>banyak;
    for(int i=1; i<=banyak; i++){
        for(int j=1; j<=1; j++){
            cout<<" ";
        }
        cout<<endl;
    }
}
```



 BelajarC++

Pernyataan pengulangan FOR adalah pernyataan yang dapat memanipulasi aliran pengesekusian CPU dan memungkinkan kita untuk membuat program yang berjalan dengan dinamis berdsarkan keinginan pengguna. Pernyataan pengulangan FOR berfungsi untuk melakukan pengesekusian beberapa pernyataan secara berulang-ulang. Dan merupakan pernyataan pengulangan yang sangat umum dan sering digunakan oleh para programmer C++.

Secara garis besar pernyataan FOR dan WHILE memiliki fungsi dan kerja yang mirip tapi. Jika dalam pernyataan WHILE dibutuhkan satu ekspresi sebagai conditional expression yang bertugas untuk penentu jalanya pengulangan. Dan hal tersebut adalah perbedaan antara FOR dan WHILE.

FOR adalah pernyataan pengulangan yang dikhususkan untuk pengulangan yang secara tepat mengetahui berapa kali pengulangan akan terjadi.

Bentuk Penulisan :

```
for (init-statement; condition-expression; end-expression){  
    Pernyataan;  
}
```

Untuk membuat pernyataan FOR dibutuhkan keyword “for” untuk memulainya, diikuti dengan 3 ekspresi di dalam tanda kurung. Setiap ekspresi dipisahkan dengan tanda titik koma (;). Ketiga ekspresi tersebut digunakan untuk penentu jalanya pengulangan, masing-masing memiliki peran yang berbeda.

init-statement umumnya adalah tempat dimana sebuah variabel didirikan dan diinisialisasi. Karena tempat itu adalah tempat yang akan dievaluasi satu kali ketika pernyataan FOR dimulai.

condition-expression adalah sebuah ekspresi penentu jalanya pengulangan, yang merupakan bilangan Boolean atau operasi yang menghasilkan bilangan Boolean. Jika ekspresi bernilai 1 (true) maka pengulangan akan terjadi tapi jika bernilai 0 (false) maka pengulangan akan berhenti atau pengulangan for akan diabaikan.

end-expression adalah tempat dimana sebuah operasi kenaikan atau penurunan dari variabel pada init-statement. Setelah pernyataan di badan pengulangan for dilakukan maka end-expression akan dievaluasi.

Dan di dalam badan pengulangan for terdapat beberapa pernyataan sebagai perintah atau aksi yang harus dilakukan oleh CPU jika pengulangan terjadi. Jika terdapat banyak pernyataan maka diwajibkan untuk mengurungnya dengan sepasang tanda kurung kurawal, jika hanya terdapat satu pernyataan kita bisa untuk tidak menggunakan sepasang tanda kurung kurawal.

Contoh program :

```
#include <iostream>
using namespace std;

int main()
{
    for(int i=1;i<=10;i++){
        cout<<i<<endl;
    }

    return 0;
}
```

Di atas adalah contoh penggunaan pernyataan for. Program di atas akan menampilkan keluaran angka 1 sampai 10 secara berurutan.

Ketika CPU bertemu dengan pernyataan for, maka langkah pertama CPU adalah mengevaluasi init-statement, dan melakukan perintahnya. Biasanya init-statement digunakan untuk mendeklarasi dan inisialisasi variabel secara bersamaan atau juga bisa memberi nilai pada variabel yang sudah ada. Dan pada contoh di atas dilakukan pembuatan variabel dengan nilai inisialisasi 1. Evaluasi init-statement hanya akan dilakukan satu kali di awal mulainya pernyataan FOR.

Langkah kedua CPU adalah mengevaluasi nilai dari condition-expression, biasanya diisi dengan sebuah ekspresi berbentuk bilangan Boolean atau operasi yang menghasilkan bilangan Boolean. Jika condition-expression bernilai 0 (false) maka pengulangan tidak akan terjadi (diabaikan), jika bernilai 1 (true) maka CPU akan mengeksekusi pernyataan pada badan pernyataan FOR.

Langkah ketiga CPU adalah mengevaluasi end-expression, merupakan tempat dimana dilakukan penaikan atau penurunan variabel pada init-statement.

Langkah keempat CPU adalah kembali mengevaluasi condition-expression. Jika condition-expression bernilai 0 (false) maka pengulangan akan dihentikan, jika bernilai 1 (true) maka langkah kedua CPU sampai langkah keempat CPU akan dilakukan berulang kali hingga condition-expression bernilai 0 (false).

Contoh program :

```
#include <iostream>
using namespace std;

int main()
{
    int sampai = 0;
    cout<<"Lakukan penghitungan sampai : ";cin>>sampai;
    for(int i=1;i<=sampai;i++){
        cout<<i<<endl;
    }

    return 0;
}
```

Menghilangkan Ekspresi

Dalam bahasa pemrograman C++ kita dimungkinkan untuk tidak menggunakan salah satu ekspresi atau semua ekspresi pada pernyataan for.

Jika pernyataan FOR tidak menggunakan init-statement maka pernyataan FOR akan tetap bisa berjalan tanpa melakukan inisialisasi pada awal sebelum pengulangan.

Jika pernyataan FOR tidak menggunakan condition-expression, pernyataan for akan melakukan pengulangan tak berhingga, meskipun condition-expression ditinggalkan kosong maka CPU akan tetap melakukan pengulangan (dianggap bernilai true).

Jika pernyataan for tidak menggunakan end-expression, Maka pernyataan for akan tetap berjalan tanpa adanya operasi di antara proses pengulangan terjadi.

Jika pernyataan FOR tidak menggunakan semua ekspresi maka pengulangan for akan menjadi pengulangan tak terhingga.

Contoh program :

```
#include <iostream>
using namespace std;

int main()
{
    int i = 0;
    for(;;i++){
        cout<<i<<endl;
    }

    return 0;
}
```

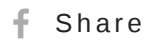
Nested FOR

Sebenarnya dalam semua bentuk pernyataan pengulangan atau penyeleksian kita dimungkinkan untuk membuat sebuah pernyataan di dalam pernyataan. Dan untuk pernyataan FOR kita dimungkinkan untuk membuat pernyataan FOR di dalam pernyataan FOR.

Contoh Program :

```
#include <iostream>
using namespace std;

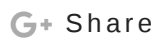
int main()
{
    int banyak = 0;
    cout<<"Berapa banyak : ";cin>>banyak;
    for(int i=1;i<=banyak;i++){
        for(int j=1;j<=i;j++){
            cout<<j<<" ";
        }
        cout<<endl;
    }
}
```



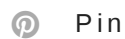
Share



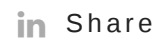
Tweet



Share



Pin

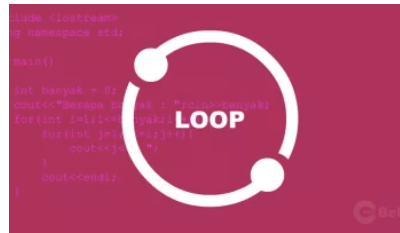


Share

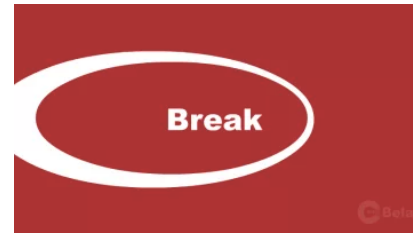
Terkait



Pernyataan Do-while
Januari 23, 2018
dalam "C++"



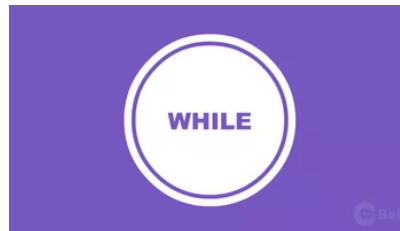
Pernyataan Pengulangan
Januari 23, 2018
dalam "C++"



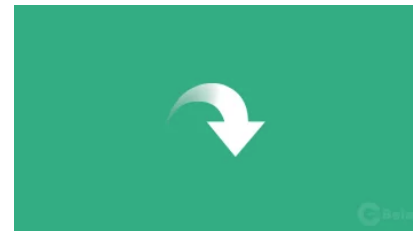
Pernyataan Break
Januari 23, 2018
dalam "C++"



Pernyataan dan Kontrol Aliran
Januari 23, 2018
dalam "C++"



Pernyataan While
Januari 23, 2018
dalam "C++"



Pernyataan Lompatan
Januari 23, 2018
dalam "C++"