

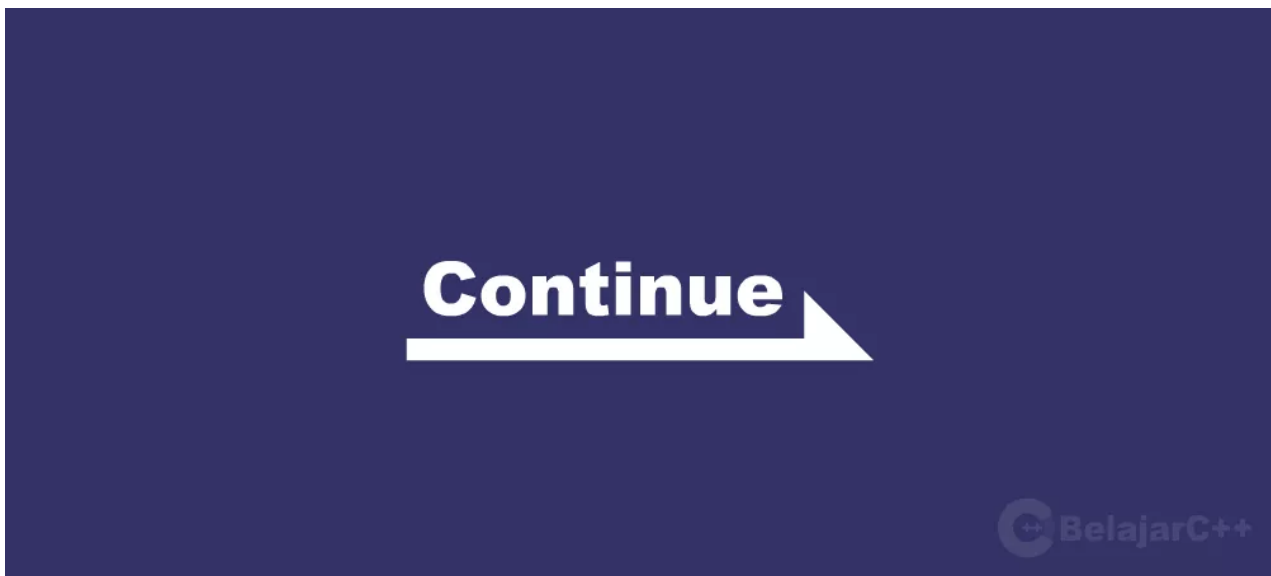
[Home](#) / [Tutorial](#) / [C++](#) / [Pernyataan Continue](#)

## Pernyataan Continue

---

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

---



Pernyataan CONTINUE adalah pernyataan yang memungkinkan kita mengatur aliran eksekusi CPU terhadap program buatan kita. Pernyataan CONTINUE adalah keyword yang berfungsi untuk menyelesaikan perulangan saat itu dan melanjutkan ke perulangan selanjutnya dengan mengabaikan sisa pernyataan pada badan perulangan tersebut. Pernyataan CONTINUE hanya dapat digunakan di dalam pernyataan pengulangan.

Bentuk Penulisan :

```
continue;
```

Pernyataan CONTINUE sangat berguna jika kita ingin mengakhiri perulangan pada saat itu dan melanjutkan ke perulangan selanjutnya. Pada saat CPU bertemu dengan pernyataan CONTINUE, CPU akan meloncat mengabaikan pernyataan-pernyataan yang ada di bawah pernyataan CONTINUE menuju ke akhir baris dari pernyataan perulangan, dengan hal itu perulangan selanjutnya akan segera dimulai.

Contoh Program :

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

int main ()
{
    int jumlah=0;
    cout<<"Jumlah : ";cin>>jumlah;
    for (int i = 1; i <= jumlah; i++){//Perulangan dimulai
        if ((i % 2)==0) continue; //jika tereksekusi CPU akan
        meloncat
        cout<<i<<endl; //pernyataan ini akan diabaikan
        //CPU akan mendarat di sini
    }

    return 0;
}
```

Contoh program di atas adalah contoh penggunaan dari pernyataan CONTINUE. Program di atas akan meminta anda untuk memasukkan dan akan menampilkan hitungan angka ganjil sebanyak angka yang pengguna masukan.

Dari program di atas dapat menampilkan hitungan ganjil karena jika pernyataan CONTINUE tereksekusi oleh CPU maka CPU akan meloncat ke baris akhir dari pernyataan pengulangan dan jika kondisi perulangan terpenuhi maka perulangan akan melanjutkan perulangan selanjutnya. Loncatan tersebut akan mengabaikan sisa dari badan pernyataan pengulangan, sehingga pernyataan keluaran di bawah pernyataan CONTINUE tidak akan dieksekusi oleh CPU.

Setelah penjelasan di atas memang CONTINUE adalah pernyataan yang sangat berguna, tapi hati-hati dalam menggunakannya, Perhatikan aliran program yang sedang anda buat, jika tidak anda bisa membuat suatu program yang sulit untuk dipahami dan anda juga bisa membuat program yang gagal berfungsi.

Salah satu kasus adalah anda bisa membuat sebuah program dengan pengulangan tak terhingga seperti di bawah ini.

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

int main ()
{
    int jumlah=0;
    cout<<"Jumlah : ";cin>>jumlah;
    int i  = 1;
    while(i <= jumlah){
        if ( i == 5) continue;
        cout<<i<<endl;
        i++;
    }

    return 0;
}
```

Di atas adalah program contoh dari kesalahan penggunaan pernyataan CONTINUE, program di atas mempunyai maksud untuk menampilkan penghitungan 1 ke angka yang dimasukan oleh pengguna, dengan melewati angka 5.

Tapi hasil dari program di atas akan memiliki perulangan yang tak terhingga, program di atas jika dijalankan terlihat berhenti dan tidak berakhir. Hal itu karena pernyataan CONTINUE diletakan sebelum penaikan nilai variabel yang digunakan untuk conditional expression oleh pernyataan pengulangan. Hal tersebut akan menyebabkan pernyataan penaikan variabel akan diabaikan.

Ketika variabel "i" bernilai 5 CPU akan mengeksekusi pernyataan CONTINUE, melompat dan mengabaikan penaikan variabel tersebut. di saat evaluasi pernyataan pengulangan, variabel "i" akan bernilai 5 karena pernyataan penaikan variabel tersebut di abaikan. Karena hal itu CPU akan kembali lagi untuk mengeksekusi pernyataan CONTINUE. Dan hal itu akan terjadi berulang-ulang tanpa henti.

Di bawah ini adalah contoh peulisan program yang benar untuk menghindari hal tersebut.

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

int main ()
{
    int jumlah=0;
    cout<<"Jumlah : ";cin>>jumlah;
    int i  = 1;
    while(i <= jumlah){
        if ( i == 5) {
            i++;
            continue;
        }
        cout<<i<<endl;
        i++;
    }

    return 0;
}
```

Atau

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

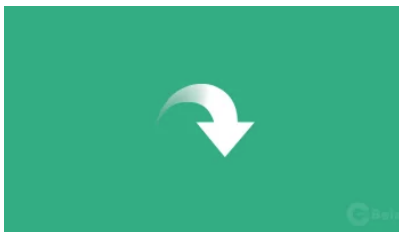
int main ()
{
    int jumlah=0;
    cout<<"Jumlah : ";cin>>jumlah;
    int i  = 1;
    do{
        if ( i == 5) continue;
        cout<<i<<endl;
    }while(++i <= jumlah);

    return 0;
}
```

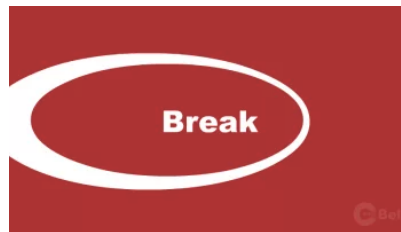
Saat anda menggunakan pernyataan lompatan anda harus berhati-hati dalam penggunaan pernyataan tersebut, dan cermat dalam menganalisis bagaimana aliran program anda.

[!\[\]\(e2376d476d06eb31946dc01a69a4403a\_img.jpg\) Share](#)[!\[\]\(74d4806277d7e73349d8e8c0897931e9\_img.jpg\) Tweet](#)[!\[\]\(0aff635c4179ba9e710b00f4b01d3b20\_img.jpg\) Share](#)[!\[\]\(830769b31eeeaca920791081939ff8ba\_img.jpg\) Pin](#)[!\[\]\(0b5e7e25e8775f7e7e80906ada4f0021\_img.jpg\) Share](#)

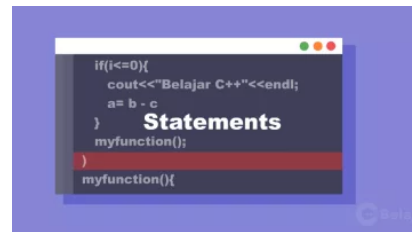
#### Terkait



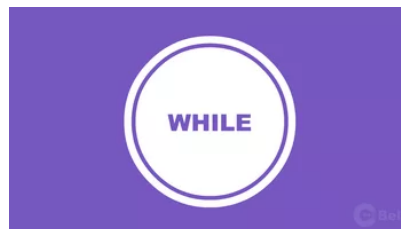
Pernyataan Lompatan  
Januari 23, 2018  
dalam "C++"



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



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



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

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

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