

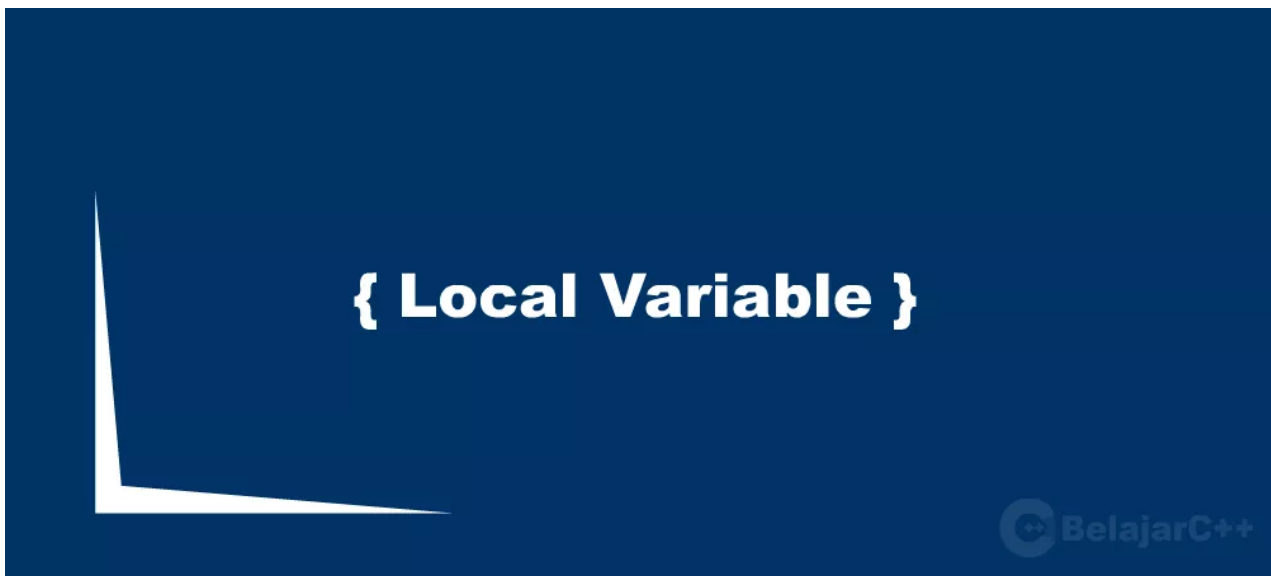
[Home](#) / [Tutorial](#) / [C++](#) / [Variabel Lokal](#)

# Variabel Lokal

---

Posted on Januari 22, 2018 | Last Modified Januari 22, 2018

---



Variable digunakan untuk menyimpan data dan didirikan dengan menggunakan identitas supaya kita dapat memanggilnya untuk menggunakan variabel tersebut. pada artikel sebelumnya “Scope”, secara tidak langsung menjelaskan bahwa deklarasi seperti variabel akan memiliki sifat akses tertentu tergantung pada penempatan dan apa yang digunakan.

Seperti ketika kita mendirikan variabel di dalam scope maka variabel itu akan bersifat sebagai Local Variable (Variabel Lokal), dan jika didirikan di luar scope maka akan memiliki sifat sebagai Global Variable. Di artikel ini penulis akan jelaskan mengenai sifat-sifat akses variabel berdasarkan scope.

## Local Variable

Local Variable atau Variabel Lokal adalah variabel yang biasanya didirikan di dalam suatu block oleh scope (seperti: function). Variabel tersebut hanya akan terlihat dan hanya dapat digunakan di dalam scope itu saja.

Local Variable adalah Automatic duration yang mempunyai arti bahwa variabel tersebut akan dibuat disaat jalur eksekusi dari kompilasi masuk ke dalam definisi dari scope tersebut dan dihancurkan setelah keluar dari scope tersebut.

Contoh Program :

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

int main ()
{
    int id = 1; //[Local Variable] Variabel didirikan dan
    terlihat setelah pernyataan ini.
    cout<<id<<endl;

    return 0;
} //variabel "id" dihancurkan dan berakhir di sini.
```

Local variable didirikan didalam scope dan hanya terlihat dan dapat digunakan di scope tersebut, dan jika scope berakhir maka local variable akan dihancurkan. Dalam scope kita juga dimungkinkan untuk melakukan nested scope (Scope di dalam scope). Hal ini akan berlaku sama untuk variabel lokal dari masing-masing scope, variabel lokal akan dihancurkan ketika bertemu dengan blok akhir dari scope. Untuk variabel lokal dari induk scope juga akan terlihat dalam anak scope.

Contoh Program :

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

int main ()
{
    int id = 1; //[Local Variable] Variabel didirikan dan
    terlihat setelah pernyataan ini.
    {
        int umur = 2; //[Local Variable] Variabel didirikan dan
        terlihat setelah pernyataan ini
        cout<<id<<endl;
        cout<<umur<<endl;
    }//Variabel "umur" dihancurkan dan berakhir di sini
    return 0;
} //variabel "id" dihancurkan dan berakhir di sini.
```

Local variabel "id" didirikan di scope pertama (Induk scope) dan juga akan terlihat di dalam nested scope (anak scope). Itu karena nested scope termasuk dalam induk scope.

Di dalam peraturan pemberian identifier, Deklarasi tidak boleh memiliki identitas yang sama dalam satu ruang lingkup. dengan arti kita dapat mendirikan nama yang sama pada ruang lingkup yang berbeda, itu juga berlaku pada anak ruang lingkup, kita dapat mendirikan deklarasi dengan identitas yang sama meskipun identitas tersebut sudah digunakan di induk ruang lingkup yang merupakan lokal deklarasi dari luar ruang lingkup (induk ruang lingkup).

Jika kita mendirikan deklarasi memiliki identitas yang sama dengan deklarasi luar ruang lingkup. Identitas dari deklarasi di dalam ruang lingkup akan menggantikan deklarasi luar ruang lingkup tersebut, hal ini disebut sebagai Shadowing atau Name Hiding.

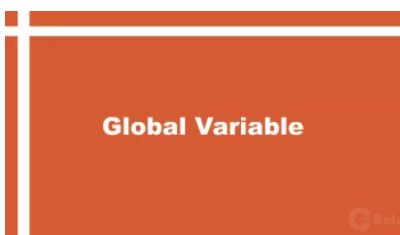
```
#include <iostream>
using namespace std;

int main ()
{
    int id = 1; //[id1 - Local Variable] Variabel didirikan dan
    terlihat setelah pernyataan ini.
    {
        int id = 2; //[ id2 - Local Variable] Variabel didirikan
        dan terlihat setelah pernyataan ini
        cout<<id<<endl;

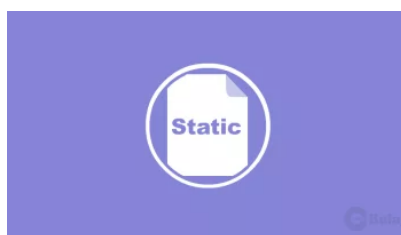
        }//Variabel "id2" dihancurkan dan berakhir di sini
        cout<<id<<endl;
        return 0;
    } //variabel "id1" dihancurkan dan berakhir di sini.
```

[!\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5\_img.jpg\) Share](#)[!\[\]\(e78f798d4ea5c530c9db49e7d26e6b95\_img.jpg\) Tweet](#)[!\[\]\(23d9fc146e83b5c3013cfa32c784f8d5\_img.jpg\) Share](#)[!\[\]\(c694a3ff3b077d76910920a6a1593ab4\_img.jpg\) Pin](#)[!\[\]\(ec9132f1d27c8919987d92907322654d\_img.jpg\) Share](#)

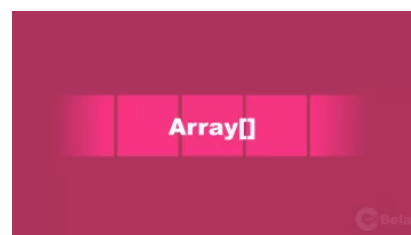
#### Terkait



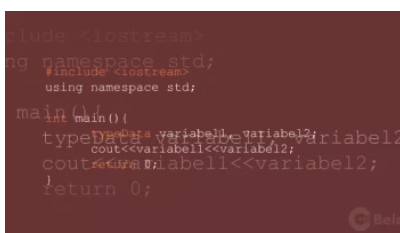
Variabel Global  
Januari 22, 2018  
dalam "C++"



Variabel Statis Global  
Januari 22, 2018  
dalam "C++"



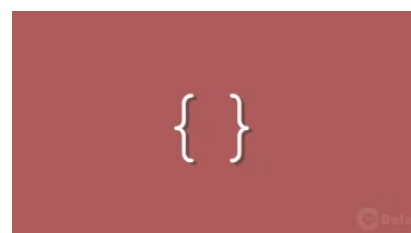
Array  
Januari 23, 2018  
dalam "C++"



Variabel  
Januari 22, 2018



Assignment Operator  
Januari 22, 2018



Scope  
Januari 22, 2018

dalam "C++"

dalam "C++"

dalam "C++"