

[Home](#) / [Tutorial](#) / [C++](#) / [Inisialisasi](#)

# Inisialisasi

---

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

---



## Initialization

### Pengertian Inisialisasi

Variabel adalah suatu tempat dimana kita bisa menyimpan nilai atau data di dalamnya, dan tempat itu merupakan memori dari komputer, memori yang telah diberikan oleh CPU memungkinkan kita untuk menyimpan data dengan jenis tertentu di dalam memori komputer. Data dari variabel bisa kita salin, simpan atau bisa kita perbaharui, untuk melakukan hal tersebut kita harus melakukannya dengan memanggil identitas dari variabel yang telah kita berikan pada saat deklarasi variabel tersebut. Inisialisasi (Initialization) adalah tugas pemberian nilai awal (data awal) yang dilakukan saat deklarasi variabel atau obyek. Inisialisasi sangat disarankan untuk dilakukan setiap mendirikan sebuah deklarasi.

Cara kerja variabel adalah, ketika kita mendirikan variabel, kita sebenarnya memesan tempat (memori) pada komputer. Di dalam bahasa pemrograman ada istilah seperti pemesanan memori dan pelepasan memori dan hal itu adalah aktivitas yang selalu dilakukan oleh program-program di dalam komputer anda.

Inisialisasi disarankan untuk dilakukan karena ketika suatu program melepaskan memori, sebenarnya memori tersebut hanya dilepaskan dan tidak dikosongkan, dengan arti ketika kita memesan memori (mendirikan variabel) tanpa inisialisasi, kita akan mendapatkan data sampah pada memori yang telah kita pesan itu. Anda bisa mencobanya melalui contoh program di bawah.

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

int main()
{
    int a;
    cout<<a;

    return 0;
}
```

Variabel “a” pada contoh program di atas akan memiliki nilai yang diwariskan oleh pengguna (program) sebelumnya jika memori tersebut pernah digunakan, dan jika kita tidak melakukan inisialisasi pada variabel tersebut maka nilai tersebut dapat membuat program kita mendapatkan data yang bukan asli berasal dari operasi program kita (data asing dan tidak diharapkan).

## Macam-macam Cara Inisialisasi :

**1. C-like Initialization atau Copy Initialization**, diwariskan oleh Bahasa Pemrograman C. dengan cara menambahkan tanda sama dengan “=” di antara nama variabel dan nilai awal variabel.

### Bentuk Penulisan

```
dataType identitas = initial_value;
```

### Contoh Penulisan

```
int varInteger = 0;  
char varSingleChar = 0; //atau char varSingleChar = ' '  
char varChar[] = "";  
string varstring = "";
```

sebenarnya untuk inisialisasi kita bebas memberikan nilai apapun sebagai nilai awal dari variabel, tapi di sini penulis contohkan cara inisialisasi variabel dengan nilai kosong.

**2. Inisialisasi Constructor atau Direct initialization**, diperkenalkan pada C++, membungkus nilai awal didalam atau antara tanda kurung "()".

#### Bentuk Penulisan

```
dataType identitas (initial_value);
```

#### Contoh Penulisan

```
int varInteger (0);  
char varSingleChar ('');  
char varChar[] = ("");
```

**3. Inisialisasi Seragam (Uniform initialization)**, diperkenalkan oleh revisi C ++ standar, pada tahun 2011 , mirip dengan cara kedua, tetapi menggunakan kurung kurawal "{}" bukan tanda kurung biasa. cara uniform initialization biasanya digunakan untuk inisialisasi variabel array.

#### Bentuk Penulisan

```
dataType identitas {initial_value};
```

#### Contoh Penulisan :

```
int varInteger{0};  
char varSingleChar{0}; //atau char varSingleChar {' '  
char varChar[]{"  
string varstring{"
```

**Contoh Program :**

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

int main ()
{
    int x;          //variabel yang tidak diinisialisasi akan
    memiliki nilai awal dari nilai warisan program lain (tidak
    diharapkan)

    int a=6;        // C-like Initialization, int bernilai= 6
    int b(2);        // Inisialisasi Constructor, int bernilai=2
    int c{3};        // Inisialisasi Seragam, int bernilai=4
    int hasil;       // int tidak bernilai. value undetermined
    (karena kita yakin dan akan mengisinya setelah ini, melalui
    oprasi penugasan. Tapi hal ini tidak disarankan)

    //jika kita tidak menginisialisasi salah satu variabel maka
    akan menghasilkan nilai yang tidak diharapkan seperti dibawah
    ini

    cout <<"Hasil variabel tidak mempunyai nilai awal : "
    <<x+a<<endl;

    //Operasi dengan variabel yang mempunyai nilai awal (hasil
    sesuai dengan harapan kita)
    a = a + b;
    hasil = a - c;
    cout <<"Hasil variabel yang mempunyai nilai awal : "<<
    hasil;

    //Inisialisasi sangat penting dalam deklarasi untuk
    memastikan tidak adanya nilai asing dalam program,
    //yang kemungkinan dapat mengacaukan operasi matematika
    dalam program kita.
    return 0;
}
```

---

**Terkait**

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

int main() {
    int variabel1, variabel2;
    cout << variabel1 << variabel2;
    cout << variabel1 << variabel2;
    return 0;
}
```

**Array[]**

**\*Pointer**

**Variabel**

Januari 22, 2018  
dalam "C++"

**Array**

Januari 23, 2018  
dalam "C++"

**Pointer**

Januari 23, 2018  
dalam "C++"

**{ Local Variable }**

**Static**

**Fundamental Data Type**

**Variabel Lokal**

Januari 22, 2018  
dalam "C++"

**Variabel Statis Global**

Januari 22, 2018  
dalam "C++"

**Tipe Data**

Januari 22, 2018  
dalam "C++"