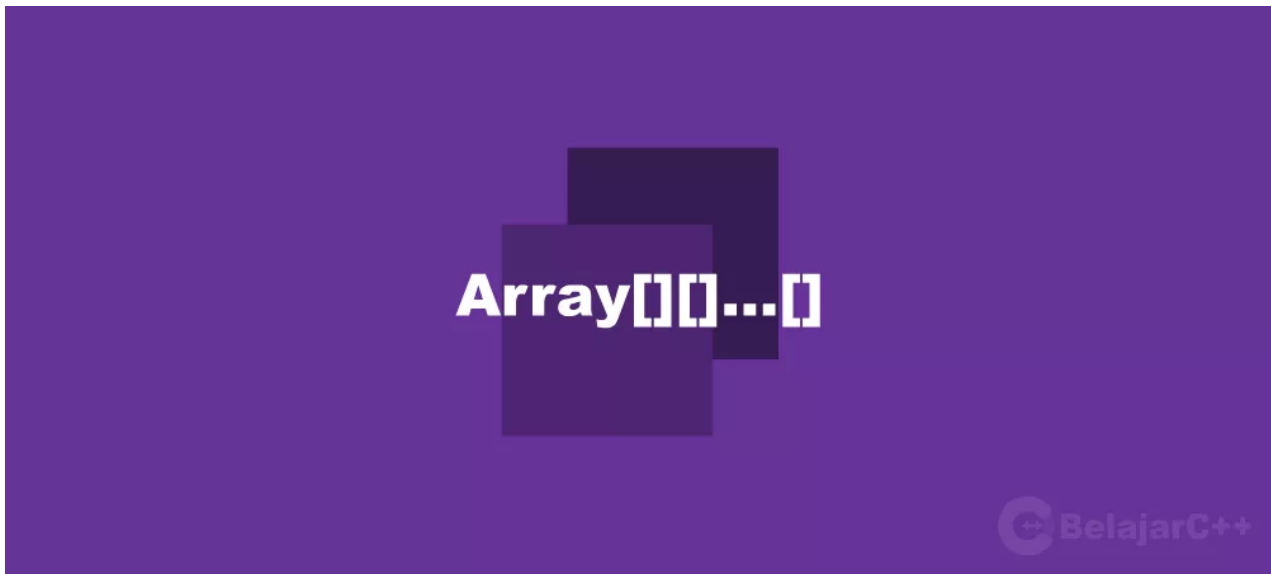


[Home](#) / [Tutorial](#) / [C++](#) / [Array Multidimensi](#)

# Array Multidimensi

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



Array adalah fitur dari bahasa pemrograman c++ yang memungkinkan kita untuk memesan banyak memori yang memiliki tipe data yang sama dengan hanya menggunakan satu identitas. Dalam variabel kita dimungkinkan untuk menggunakan array lebih dari satu buah, dan hal itu disebut sebagai Array Multidimensi.

Untuk mengubah variabel biasa menjadi variabel array adalah dengan menambahkan operator subscript dengan ukuran di dalamnya, dalam C++ kita dimungkinkan untuk menggunakan banyak subscript, hal itu akan membuat variabel tersebut menjadi array multidimensi.

## Bentuk Penulisan

```
tipeData identifier[array-1][array-2]...[array-n];
```

## Array Dua Dimensi

Untuk membuat sebuah variabel dengan array dua dimensi, kita membutuhkan dua buah operator subscript ( `[]` ) dengan masing-masing ukuran array.

### Bentuk Penulisan

```
tipeData identifier[size1][size2];
```

### Contoh Penulisan

```
string nama[2][3];
```

array dua dimensi memungkinkan anda untuk menggunakan array di dalam array, setiap array memiliki array di dalamnya. Bagi penulis jujur saya sulit untuk menjelaskan array multidimensi. Tapi untuk array dua dimensi bisa digambarkan seperti sebuah tabel yang mempunyai baris dan kolom.

Kita dapat membayangkan sebuah array dua dimensi seperti sebuah table yang memiliki baris dan kolom, untuk array atau subscript pertama adalah baris dan subscript kedua adalah kolom.

**string nama[2][3]**

	0	1	2
0			
1			

*Ilustrasi Array*

### **Inisialisasi Array Dua Dimensi**

Untuk menginisialisasi array dua dimensi adalah dengan cara menggunakan tanda kurung kurawal di dalam tanda kurung kurawal.

```
string nama[3][2]={  
{"joko", "dadang"}, //baris ke satu dengan 2 kolom  
{"fajar", "nick"}, // baris ke dua dengan 2 kolom  
{"laksono", "eren"} // baris ke tiga dengan 2 kolom  
}
```

Tanda kurung kurawal pertama akan menyatakan keseluruhan array dari variabel, di dalamnya terdapat 3 sepasang kurung kurawal yang dipisahkan dengan tanda koma, yang menyatakan array dari subscript pertama. Di dalam masing-masing tanda subscript pertama berdiri 2 nilai sesuai ukuran array subscript kedua, masing-masing nilai dipisahkan dengan tanda koma.

Untuk bentuk penulisan bebas menurut keinginan anda, seperti anda bisa menulis dalam satu baris atau seperti di atas, hal yang penting adalah penggunaan tanda-tanda baca dan cara penulisannya harus benar seperti di atas, bentuk penulisan di atas dibentuk seperti halnya sebuah tabel hanya untuk mempermudah kita dalam membaca nilai-nilai tersebut.

Jika anda tidak mengisi salah satu cell atau semua cell maka anda akan menginisialisasi array dalam variabel array tersebut menjadi bernilai 0 atau kosong.

```
string nama[3][2]={  
{"joko", "dadang"}, //baris ke satu dengan 2 kolom [1][1] =  
"joko", [1][2]="dadang"  
{"fajar"}, // baris ke dua dengan 2 kolom, [2][1] = "fajar", [2]  
[2] ""  
{} // menginisialisasi satu baris dengan angka 0, [3][1] = "",  
[3][2]=""  
}
```

dan juga bisa seperti ini

```
string nama[3][2]={}
```

### Deklarasi Variabel Array Multidimensi Tanpa Ukuran

array multidimensi adalah array, kita masih dimungkinkan untuk mendeklarasi variabel array tanpa menyertakan ukuran dari array tersebut, ukuran otomatis akan terhitung berdasarkan

jumlah nilai pada proses inisialisasi.

## Mengakses Array Dua Dimensi

```
nama[2][1]
```

untuk mengakses sebuah cell pada array dimensi dua, sama seperti kita mengakses array dimensi satu. Tapi disini kita membutuhkan dua subscript sebagai alamat cell tersebut.

untuk dapat megakses setiap cell di dalam array dua dimensi dengan dinamis anda dapat menggunakan pernyataan pengulangan nested (nested loop).

### Contoh Program

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

const int MAX_ROW = 3;
const int MAX_COL = 3;

int main ()
{
    string nama[MAX_ROW][MAX_COL]={
        {"joko", "dadang", "aziz"},
        {"fajar", "nick"},
        {"laksono", "eren"}
    };

    for(int i=0;i<MAX_ROW;++i){
        for(int j=0;j<MAX_COL;++j){
            cout<<"["<<(i+1)<<"]["<<(j+1)<<"]. "<<nama[i][j]
<<endl;
        }
    }

    return 0;
}
```

## Array Tiga Dimensi ke atas

Jika pembaca sudah mengerti mengenai konsep dari array, anda akan dengan mudah untuk memodifikasi array, menambahkan beberapa subscript untuk menjadikannya array multidimensi. Secara garis besar, pembuatan, penggunaan, pengaksesan array multi dimensi memiliki konsep yang sama seperti apa yang telah penulis jelaskan di atas pada array dua dimensi.

## Pembekakan memori

Pada artikel [belajarcpp.com](http://belajarcpp.com) sebelumnya telah penulis sampaikan bahwa array bisa memakan banyak memori, terutama array multi dimensi. Karena apa yang kita pesan adalah berkali-kali lipat lebih besar dari variabel biasa.

Misalnya anda mendirikan variabel array satu dimensi "int nomer[5]", besar memori untuk membuat variabel integer adalah 4 byte (di beberapa komputer bisa berbeda-beda), dikali dengan jumlah elemen yaitu 5 elemen, berarti apa yang telah anda pesan adalah variabel dengan jumlah memori 20 byte.

dan ada yang lebih memakan memori adalah saat anda memesan variabel array multi dimensi, misalnya anda memesan variabel empat dimensi "int nomer[3][4][5][2]", dengan hitungan :

```
int * 10 * 40 * 50 * 20
4 byte * 10 * 40 * 50 * 20 = 1600000 byte
```

Jumlah itu merupakan jumlah yang tidak kecil, anda bisa membebani program anda dengan pembekakan memori, jadi saran penulis adalah ketika anda mendirikan variabel array berhati-hatilah dalam memesan jumlah elemen pada array tersebut, pesanlah sesuai yang anda butuhkan, jangan memesan memori terlalu banyak yang nantinya bisa menjadi memori yang tidak digunakan.

f Share

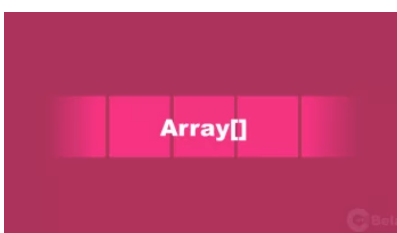
🐦 Tweet

G+ Share

📌 Pin

in Share

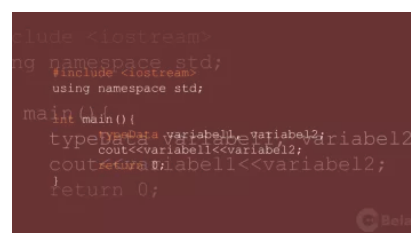
### Terkait



Array



C-Style String



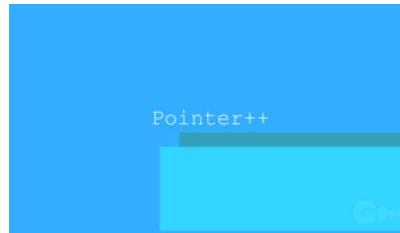
Variabel

Januari 23, 2018  
dalam "C++"



Pointer  
Januari 23, 2018  
dalam "C++"

Januari 23, 2018  
dalam "C"



Aritmetika pada Pointer  
Januari 23, 2018  
dalam "C++"

Januari 22, 2018  
dalam "C++"



Tipe Data  
Januari 22, 2018  
dalam "C++"