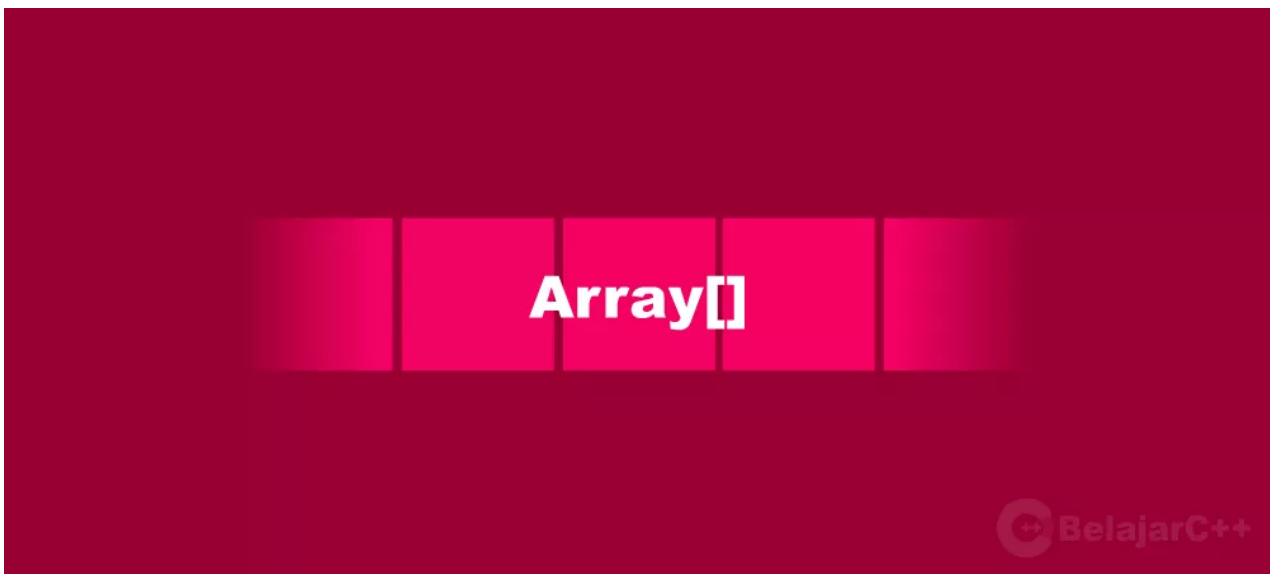


[Home](#) / [Tutorial](#) / [C++](#) / [Array](#)

Array

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



Array atau Larik adalah sebuah variabel yang memiliki serangkaian elemen dari jenis tipe data yang sama. Elemen-elemen tersebut dirangkai di dalam memori yang berdekatan dengan elemen lainnya. konsep ini mirip seperti struct dan class yang dapat merupakan kumpulan dari variabel, perbedaan dengan mereka adalah array merupakan kumpulan variabel dengan satu indentifier dan satu tipe data yang sama, dan dalam istilah-istilah array kumpulan dari variabel tersebut adalah elemen.

Bayangkan bahwa jika pembaca sedang bekerja membuat program untuk suatu organisasi sekolah, tepatnya digunakan untuk mendata nilai-nilai dari siswa pada suatu kelas, dan dalam satu kelas terdapat 50 anak. Dengan arti kita membutuhkan 50 tempat penyimpanan (variabel).

```
double nilai1, nilai2, nilai3 .....nilai50;
```

mungkin untuk pembaca yang baru mengenal dunia programmer akan berpikiran untuk membuat 50 variabel secara individu seperti pernyataan di atas. Cara di atas adalah sangat-sangat tidak efektif atau salah. Bagaimana jika satu kelas tidak selalu memiliki 50 anak,

mungkin di masa depan akan memiliki murid lebih dari 50 anak di dalam kelas. Dengan arti program itu akan selalu membutuhkan modifikasi, bukan hanya pada variabel tersebut tapi keseluruhan program, seperti input, proses dan output. Untungnya, dalam bahasa pemrograman C++ dan hampir semua bahasa pemrograman, menyediakan fitur yang bernama “Array” atau “Larik”.

Array memungkinkan kita untuk membuat sebuah variabel yang dapat menyimpan banyak nilai. Di dalam array terdapat banyak elemen yang merupakan tempat untuk penyimpanan setiap nilai.

Bentuk Penulisan

```
tipeData identifier[ukuran];
```

untuk mendirikan sebuah variabel array sama seperti kita mendirikan sebuah variabel biasa yang dimana membutuhkan tipe data, identifier, dan titik koma. Untuk merubah variabel biasanya menjadi variabel array adalah dengan menambahkan operator subscript (`[]`).

Untuk parameter di dalam operator subscript disebut sebagai index atau subscript, tempat dimana kita memberitahukan compiler berapa jumlah variabel yang akan di alokasi sesuai yang kita inginkan. dan dalam array, istilah dari banyak variabel yang dialokasi itu disebut sebagai elemen.

Contoh Penulisan

```
int nilai[50];
```

Di atas adalah contoh penulisan dari penggunaan array, variabel di atas memiliki identifier “nilai” bertipe data integer sebanyak 50 elemen. Tersebut memungkinkan kita untuk menyimpan 50 nilai berbeda dalam satu identifier variabel “nilai”.

Pernyataan di atas dinamakan Array statis atau Fixed array, karena kita mendirikan variabel array dengan menggunakan konstanta bernilai 50 sebagai index saat penulisan program. Dan pemberian index hanya dapat dilakukan menggunakan konstanta dan dilakukan sebelum kompilasi (Compile-time), tidak saat program berjalan (Runtime).

Untuk mendirikan sebuah variabel array dibutuhkan tipe data, identifier dan index di dalam subscript. Setiap tipe data untuk mendirikan variabel, dimungkinkan untuk digunakan pada array. untuk identifier kita bebas dalam memberikan identitas tapi harus di bawah peraturan pembuatan identitas yang telah penulis jelaskan. Untuk index adalah tempat dimana sebuah konstanta diletakan sebagai penentu banyaknya elemen.

Inisialisasi Array

Saat mendirikan variabel array kita juga dimungkinkan untuk memberi nilai saat deklarasi variabel array secara bersamaan, inisialisasi array disebut sebagai "Initializer list". apa yang kita butuhkan untuk inisialisasi sebuah variabel array adalah sepasang tanda kurung kurawal yang mengapit semua nilai tersebut.

Contoh Penulisan

```
int grade[50] = {100, 90, 80, 70, 60, 50, 40, 30, 20, 10};
```

di atas adalah contoh dari cara inisialisasi sebuah variabel array yang memiliki 50 elemen. Untuk melakukan inisialisasi kita membutuhkan tanda sama dengan setelah itu nilai dari setiap elemen. Nilai-nilai setiap elemen akan di kurung di dalam sepasang tanda kurung kurawal. Setiap nilai dipisahkan dengan tanda koma.

Contoh di atas adalah pernyataan yang valid, meskipun variabel tersebut memesan 50 elemen tetapi penulis hanya memberikan 10 nilai. Dari 10 nilai tersebut akan mengisi elemen pertama hingga ke sepuluh. Untuk elemen 10 sampai 50 akan di inisialisasi dengan nilai "0" atau kosong.

Inisialisasi adalah sebuah pilihan, kita bebas untuk membuat variabel dengan inisialisasi atau tanpa inisialisasi. seperti apa yang telah penulis jelaskan pada artikel tentang "Inisialisasi". inisialisasi variabel itu penting meskipun tidak wajib. Jika variabel array di atas tidak melakukan inisialisasi maka otomatis setiap elemen akan mengandung nilai acak yang berasal dari pengguna memori sebelumnya.

Mendeklarasikan tanpa ukuran array

Pada bahasa pemrograman C++ dalam mendirikan sebuah variabel array kita dimungkinkan untuk mendirikan variabel array tanpa menyertakan ukuran, hanya operator subscript tanpa nilai di dalamnya. Tapi untuk melakukan hal itu, kita diwajibkan untuk menyertai dengan inisialisasi. bukan berarti jika variabel array tidak menggunakan ukuran akan memiliki elemen yang dinamis. Variabel array dengan cara tersebut masih dalam golongan variabel array statis. Dan compiler selalu akan membutuhkan ukuran array untuk alokasi (elemen) yang dibutuhkan.

Ketika kita mendeklarasikan sebuah variabel array tanpa ukuran array, maka ukuran dari variabel array tersebut otomatis akan dihitung berdasarkan jumlah nilai pada proses inisialisasi.

Contoh Penulisan

```
int nilai[]={2,4,7,9}
```

pada contoh penulisan di atas compiler akan mendeklarasikan sebuah variabel array yang memiliki 4 elemen yang dihitung berdasarkan jumlah nilai pada proses inisialisasi.

Mengakses Variabel Array

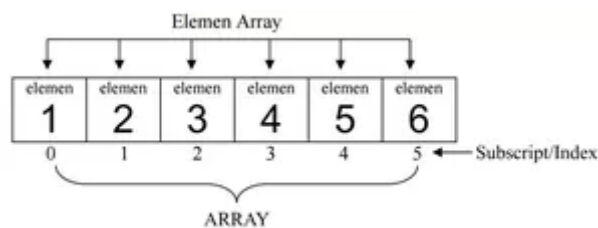
Untuk mengakses variabel array tidak jauh beda dengan cara mengakses variabel biasa, untuk mengakses variabel array membutuhkan Operator subscript (`[]`) dengan index di dalamnya yang diletakan setelah identitas variabel.

Contoh Penulisan

```
grade[2];
```

contoh di atas akan mengakses elemen ketiga dari variabel "grade". Index yang berada di dalam subscript digunakan untuk mengakses nilai elemen berdasarkan konstanta yang diberikan. Pada contoh di atas kita mencoba untuk mengakses elemen ketiga dari variabel "grade".

Bahasa pemrograman C++ tepatnya pada fitur array selalu menggunakan bilangan cacah (bilangan dimulai dari 0). Dan setiap elemen yang kita pesan akan di nomori berdasarkan urutan dari 0 sampai sebanyak yang kita pesan. Misalnya kita memesan grade[50] maka elemen pertama berada pada urutan nomer 0 dan elemen terakhir berada pada urutan 49.



Elemen pada array

Jika kita menuliskan nomer 10 maka elemen yang akan di akses adalah elemen nomer urutan nomer 11.

Contoh Program

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

int main ()
{
    char grade[3]={};

    grade[0] = 'a';
    grade[1] = 'b';
    grade[2] = 'c';

    cout<<"0 : "<<grade[0]<<endl;
    cout<<"1 : "<<grade[1]<<endl;
    cout<<"2 : "<<grade[2]<<endl;

    return 0;
}
```

Hati-hati dalam mengakses array, dalam pemrograman C++ kita dimungkinkan untuk mengakses elemen dari array yang bukan merupakan anggota array tersebut. misalnya kita mencoba akses index nomer 50, tetapi variabel array itu sendiri hanya memiliki 10 elemen.

Kita dimungkinkan untuk mengakses index dari array meskipun bukan merupakan anggota dari variabel array tersebut. itu dapat di lakukan tapi hal itu dilarang untuk dilakukan. Karena apa yang coba kita akses bukan merupakan memori milik program anda. Bisa saja memori tersebut dimiliki oleh program lain.

Pengulangan dengan Array

Pada contoh program di atas kita menggunakan pernyataan tunggal untuk mengakses setiap elemen array. jika array memiliki 50 elemen bukan berarti kita harus membuat 50 pernyataan tunggal untuk mengakses setiap elemen tersebut. termasuk penyuntingan ketika anda merubah index array.

Untuk mengakses array secara dinamis, tidak memerlukan banyak waktu untuk membuat pernyataan menyunting program jika ada perubahan pada array. kita dapat memanfaatkan pernyataan pengulangan.

Contoh Program

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

int main ()
{
    const int MAX_ARRAY = 5;
    string nama[MAX_ARRAY]={};

    for(int i=0;i<MAX_ARRAY;++i){
        cout<<"Masukan Nama : ";cin>>nama[i];
    }

    cout<<endl<<"=== Daftar Nama ==="<<endl;
    for(int i=0;i<MAX_ARRAY;++i){
        cout<<(i+1)<<". "<<nama[i]<<endl;
    }

    return 0;
}
```

Pernyataan pengulangan FOR memang merupakan pernyataan yang efektif untuk digunakan dan sering digunakan oleh para programmer dalam mengelolah variabel array. di atas adalah program yang akan meminta 5 nama dan nama tersebut akan di tampilkan di bawahnya sebagai daftar nama. Program di atas mencontohkan program yang cukup efisien. Jika kita ingin mengubah banyak elemen pada array, apa yang harus dilakukan hanyalah mengubah nilai pada variabel konstanta "MAX_ARRAY".

Array sebagai Parameter Function

Variabel array sebagai parameter di function akan berbeda dengan variabel sebagai parameter pada umumnya. Jika anda mengikuti pembelajaran pada belajarcpp.com, penulis menyadari bahwa materi function belum penulis sampaikan di beberapa materi sebelumnya. Jadi penulis menyarankan anda untuk mengerti hal ini sebagai sebuah teori. Jadi ketika anda bertemu dengan materi function, anda akan tau apa yang akan terjadi ketika parameter dalam function tersebut adalah variabel array.

Sifat parameter pada umumnya adalah menyalin nilai dari variabel yang diajukan sebagai argument. Jadi di antara variabel parameter dengan variabel asli tidak saling berhubungan. Ketika anda mencoba mengubah nilai pada parameter maka variabel asli tidak akan terpengaruh, begitu pula sebaliknya.

Tapi jika parameter itu merupakan array, dan jika anda mencoba untuk mengubah nilai dari salah satu variabel maka keduanya akan terpengaruh. Hal itu terjadi karena variabel array bisa sangat memakan banyak memori, untuk menghindari lamanya proses penyalinan dan pembekakan memori program, C++ membuat parameter array tidak menyalin nilai dari variabel array asli, tapi dia menyalin alamat memori pada variabel asli.

Contoh Program

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

const int MAX_ARRAY = 5;

void ubah(string str[]){
    for(int i=0;i<MAX_ARRAY;++i){
        str[i]="Berubah";
    }
}

int main ()
{
    string nama[MAX_ARRAY]={};

    for(int i=0;i<MAX_ARRAY;++i){
        cout<<"Masukan Nama : ";cin>>nama[i];
    }

    ubah(nama);

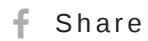
    cout<<endl<<"=== Daftar Nama ==="<<endl;
    for(int i=0;i<MAX_ARRAY;++i){
        cout<<(i+1)<<". "<<nama[i]<<endl;
    }

    return 0;
}
```

Tambahan : Mengakses hanya menggunakan identitas

Operator subscript dan nilai di dalamnya digunakan untuk mengakses elemen. Jika kita mencoba untuk mengakses sebuah variabel array tanpa menggunakan subscript maka apa yang akan kita dapat adalah alamat memori dari elemen pertama variabel array.

Tapi berbeda dengan array C-Style String, array dengan tipe data char akan menampilkan sebuah string dari keseluruhan nilai pada setiap elemen.



Share



Tweet



Share

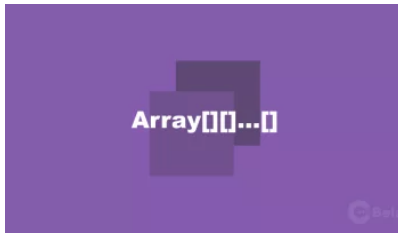


Pin



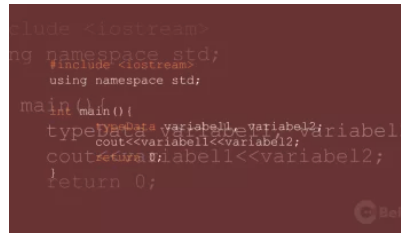
Share

Terkait



Array Multidimensi

Januari 23, 2018
dalam "C++"



Variabel

Januari 22, 2018
dalam "C++"



C-Style String

Januari 23, 2018
dalam "C"



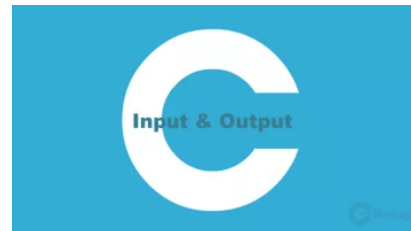
Assignment Operator

Januari 22, 2018
dalam "C++"



Tipe Data

Januari 22, 2018
dalam "C++"



Dasar Input Output pada C

Januari 22, 2018
dalam "C"