

## ARRAY (LARIK)

### **Kompetensi Umum:**

Dapat menjelaskan Penggunaan Tipe Data *Array*

### **Kompetensi Umum:**

1. Memahami tipe data *array* dan keuntungan yang dapat diberikan
2. Memahami *array* yang memiliki dimensi lebih dari satu
3. Dapat mengimplementasikan tipe data *array* dalam program
4. Memahami cara menentukan tipe data bentukan dan menggunakannya dalam program

### 8.1 Array

Tipe data array adalah tipe data terstruktur yang merujuk kepada sebuah atau sekumpulan elemen yang mempunyai tipe data yang sama melalui indeks. *Array* biasanya disebut juga sebagai **tabel**, **vector** atau **larik**.

Elemen dari *array* dapat diakses langsung jika dan hanya jika indeks terdefinisi (telah ditentukan nilainya sesuai dengan domain yang didefinisikan untuk indeks tersebut). Struktur data array disimpan dengan urutan yang sesuai dengan definisi indeks secara kontinu (berurutan) dalam memori komputer. Karena dari indeks haruslah merupakan suatu tipe data yang memiliki keterurutan (ada suksesor dan predecesor), misal tipe integer dan karakter.

Dengan kata lain Array adalah himpunan elemen (variabel) dengan tipe yang sama dan disimpan secara berurutan dalam memory yang ditandai dengan memberikan indeks pada suatu nama variabel. Contohnya kita dapat menyimpan 5 nilai dengan tipe **int** tanpa harus mendeklarasikan 5 identifier variabel yang berbeda.

Untuk mendeklarasikan sebuah array menggunakan tanda [ ] (bracket). Adapun bentuk umum dari pendeklarasian array adalah sebagai berikut:

**type name [elements];**

Maka contoh array diatas dideklarasikan sebagai berikut:

**int nama [5];**

Dilihat dari dimensinya, *array* dapat dibagi menjadi *Array* Satu Dimensi, *Array* Dua Dimensi dan *Array* Multi-Dimensi.

### 8.1.1 *Array* Satu Dimensi

Sebelum digunakan, variabel array perlu dideklarasikan terlebih dahulu. Cara mendeklarasikan variabel array sama seperti deklarasi variabel yang lainnya, hanya saja diikuti oleh suatu indek yang menunjukkan jumlah maksimum data yang disediakan.

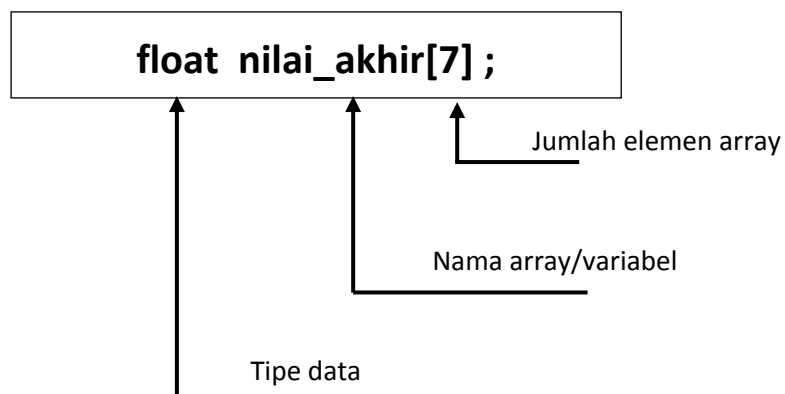
Bentuk umum pendeklarasian array:

**tipe-data nama\_variabel[ukuran]**

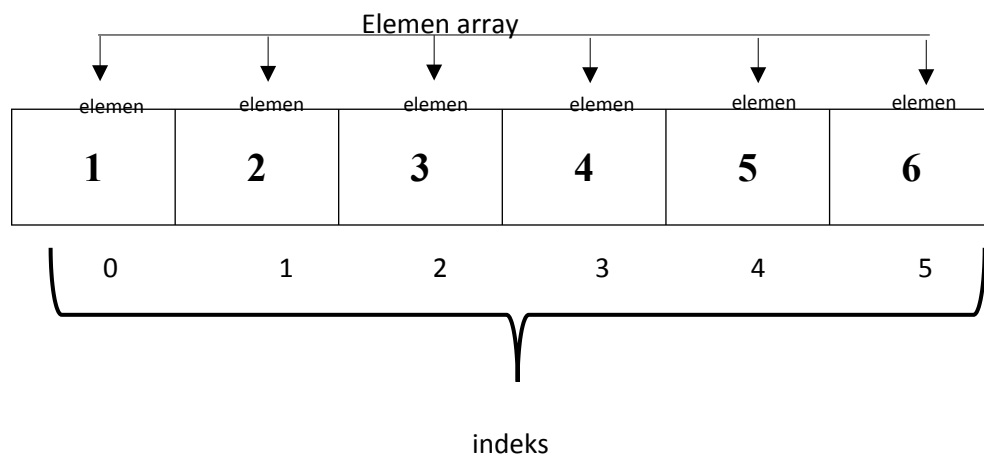
Keterangan:

- Tipe Data : untuk menyatakan tipe data yang digunakan.
- Ukuran : untuk menyatakan jumlah maksimum elemen array

#### Contoh Pendeklarasian Array



Suatu array dapat digambarkan sebagai kotak panjang yang berisi kotak-kotak kecil didalam kotak panjang tersebut.



**Gambar 8.1** Elemen dan Indeks Array

Suatu array, dapat diakses dengan menggunakan subscript atau indeksny. Bentuk umum pengaksesan adalah:

**Nama\_array[subscript/indeks]**

Contoh:

```
nilai_akhir[3];  
nilai_akhir[1];  
nilai_akhir[0];
```

```

//*=====*
//*Contoh Program menggunakan Array *
//*=====*

#include <iostream>
using namespace std;

int main() {
float suhu[5]; //Array dengan elemen 5 bertipe
float
//membaca data dari keyboard dan meletakkan array
cout<< "Masukan 5 buah data suhu"<<endl;
for (int i = 0; i < 5; i++){
    cout<<i+1<<". ";
    cin>>suhu[i];

}
//menampilkan isi array ke layar
cout<< "Data suhu yang anda masukan : "<<endl;
for(int i = 0; i<5; i++){
cout<<suhu[i]<<endl;

}
return 0;
}

```

**Gambar 8.2** Program menggunakan Array

Hasil dari program diatas:

```

Masukan 5 buah data suhu
1.27.5
2.28
3.27.5
4.30.5
5.27
Data suhu yang anda masukan :
27.5
28
27.5
30.5
27

Process returned 0 (0x0)   execution time : 29.913 s
Press any key to continue.

```

**Gambar 8.3** Hasil Program menggunakan Array

Tampak diatas terdapat pernyataan:

- **float suhu[5];** menyatakan array **suhu** dapat menyimpan 5 (lima) buah data bertipe **float**.
- **Suhu[i]** menyatakan elemen suhu dengan *subscript* sama dengan **i**.
- **cin>>suhu[i];** membaca data dari keyboard dan meletakkan ke elemen nomor **i** pada array **suhu**.
- **cout<< suhu[i];** akan menampilkan elemen bernomor **i** pada array **suhu**.

Dibawah ini contoh program array 1 dimensi:

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

int main() {
    // Deklarasi Array dengan 5 elemen
    int angka[5];
    // Mengisi Array dengan nilai
    angka[0] = 10;
    angka[1] = 20;
    angka[2] = 30;
    angka[3] = 40;
    angka[4] = 50;

    // Menampilkan nilai-nilai dalam Array
    cout << "Nilai dalam Array adalah: ";
    for (int i = 0; i < 5; i++) {
        cout << angka[i] << " ";
    }
    return 0;
}
```

**Gambar 8.4** Program menggunakan Array 1 Dimensi

Hasil dari program diatas;

```
Nilai dalam Array adalah: 10 20 30 40 50
Process returned 0 (0x0)   execution time : 0.018 s
Press any key to continue.
```

**Gambar 8.5** Program menggunakan Array 1 Dimensi

### 8.1.2 Array Dua Dimensi

*Array* dua dimensi merupakan *array* yang terdiri dari m buah baris (row) dan n buah kolom (column). Bentuk *array* semacam ini menggunakan 2 (dua) buah kelompok indeks yang masing-masing direpresentasikan sebagai indeks baris dan kolom. Jika ingin memasukkan atau membaca sebuah nilai pada matriks maka, harus diketahui terlebih dahulu indeks baris dan kolomnya.

Untuk mendeklarasikan sebuah array dua dimensi dalam C++, kita harus menggunakan tanda [ ] (bracket). Adapun bentuk umum dari pendeklarasian array dua dimensi adalah sebagai berikut:

**tipe\_data nm\_array [jmlh\_elemen\_baris] [jmlh\_elemen\_kolom];**

Sebagai contoh misal kita ingin mendeklarasikan sebuah array dua dimensi (misalnya dengan nama LARIK) yang mempunyai jumlah elemen baris sebanyak 3, jumlah elemen kolom sebanyak 5 dan array LARIK memiliki tipe data int, maka bentuk array nya adalah sebagai berikut:

**int LARIK [3][5];**

Seandainya array LARIK akan kita beri nilai saat pendeklarasian (inisialisasi), maka contoh penulisannya dapat ditulis sebagai berikut:

**int LARIK [3][5] = {{1,2,3,4,5},{6,7,8,9,10},{11,12,13,14,15}};**

Dari pendeklarasian sekaligus inisialisasi array LARIK diatas, dapat diartikan bahwa kita telah memesan tempat pada memori komputer sebanyak 15 tempat, dengan indeks dari LARIK[0][0] sampai LARIK[2][4], dimana nilai-nilai akan dimasukkan ke elemen array secara berturut-turut, mulai dari LARIK[0][0] akan di isi dengan nilai '1' sampai LARIK[2][4]

yang di isi dengan nilai ‘15’. untuk lebih jelasnya berikut kami sajikan ilustrasi dari array LARIK.

**int LARIK [3][5]={1,2,3,4,5},{6,7,8,9,10 }{11,12,13,14,15}}**

Dibawah ini juga contoh inisialisasi nilai array 2 dimensi:

**int A[3][4]={3,4,8,0},{3,9,2,1},{6,3,0,2}};**

Untuk dapat melihat lebih jelas inisialisasi array yang dibuat diatas dapat digambarkan lewat tabel berikut:

**Tabel 8.1** Inisialisasi Array

<b>A</b>	<b>Kolom ke 0</b>	<b>Kolom ke 1</b>	<b>Kolom ke 2</b>	<b>Kolom ke 3</b>
Baris ke 0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
Baris ke 1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
Baris ke 2	A[2][0]	A[2][1]	A[2][2]	A[2][3]

<b>A</b>	<b>Kolom ke 0</b>	<b>Kolom ke 1</b>	<b>Kolom ke 2</b>	<b>Kolom ke 3</b>
Baris ke 0	3	4	8	0
Baris ke 1	3	9	2	1
Baris ke 2	6	3	0	2

Karena index array dimulai dari 0 maka setiap elemem baris dan kolom dimulai dari 0. Nilai 3 akan menempati pada baris ke 0 dan kolom ke 0. nilai 4 menempati baris ke 0 kolom ke 1 nilai 8 pada baris ke 0 dan kolom ke 2 dan seterusnya.

```
#include <iostream>
using namespace std;
int main()
{
    // Membuat array 2 dimensi dengan ukuran 3 baris dan 4 kolom
    int a[3][4] = {{1, 2, 3, 4}, {5, 6, 7, 8}, {9, 10, 11, 12}};

    // Menampilkan isi elemen array
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            cout << a[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}
```

Hasil dari program diatas:

**Gambar 8.6** Program menggunakan Array 2 Dimensi

```
1 2 3 4
5 6 7 8
9 10 11 12

Process returned 0 (0x0)   execution time : 0.031 s
Press any key to continue.
```

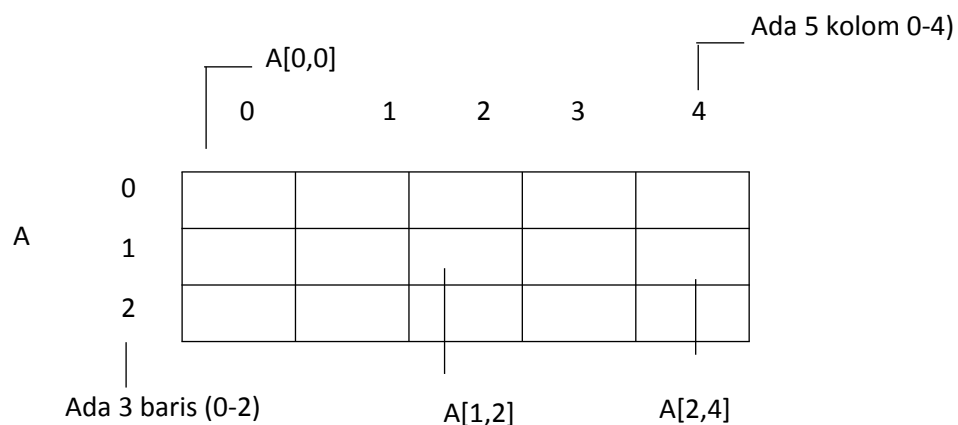
**Gambar 8.7** Hasil Program menggunakan Array 2 Dimensi

Dalam program di atas, array 2 dimensi a dideklarasikan dengan ukuran 3 baris dan 4 kolom. Kemudian, array a diinisialisasi dengan nilai-nilai yang ditentukan. Setelah itu, isi elemen array a ditampilkan dengan menggunakan perulangan for. Setiap elemen ditampilkan dengan menggunakan indeks i untuk baris dan indeks j untuk kolom.

### **Mengakses data array dua dimensi**

Seperti *array* satu dimensi, *array* dua dimensi juga diakses melalui indeksnya. Contoh:  $A[1,2]$ , menunjuk pada posisi nilai *array* pada baris 1 kolom 2.

Untuk menyimpan nilai dalam *array* dua dimensi, dapat dilakukan dengan cara sebagai berikut:





$A[0,0] \leftarrow 2$  : Simpan 2 pada *array* A baris 0, kolom 0

$A[0,1] \leftarrow 4$  : Simpan 4 pada *array* A baris 0, kolom 1

$A[1,2] \leftarrow 8$  : Simpan 8 pada *array* A baris 1, kolom 2

$A[2,2] \leftarrow A[0,0] + A[1,2]$  : tambahkan nilai pada array A baris 0, kolom 0 dengan nilai pada array A baris 1, kolom 2 dan simpan hasilnya pada array A baris 2, kolom 2.

		Ada 5 kolom (0-4)				
		A[0,0]				
		0	1	2	3	4
A	0	2	4			
	1			8		
	2			10		
Ada 3 baris (0-2)		A[1,2]			A[2,4]	