

Pertemuan 2

ARRAY DIMENSI 1 & 2

- ❖ Array atau Larik merupakan Struktur Data Sederhana yang dapat didefinisikan sebagai pemesanan alokasi memory sementara pada komputer.
- ❖ Array dapat didefinisikan sebagai suatu himpunan hingga elemen yang terurut dan homogen.
- ❖ Terurut : Dapat diartikan bahwa elemen tersebut dapat diidentifikasi sebagai elemen pertama, elemen kedua dan seterusnya sampai elemen ke-n.
- ❖ Homogen : Adalah bahwa setiap elemen dari sebuah Array tertentu haruslah mempunyai type data yang sama.

❖ Sebuah Array dapat mempunyai elemen yang seluruhnya berupa integer atau character atau String bahkan dapat pula terjadi suatu Array mempunyai elemen berupa Array.

❖ **Karakteristik Array :**

1. Mempunyai batasan dari pemesanan alokasi memory (Bersifat Statis)
2. Mempunyai Type Data Sama (Bersifat Homogen)
3. Dapat Diakses Secara Acak

- ❖ **3 Hal** yang harus diketahui dalam mendeklarasikan array :
 - a. Type data array
 - b. Nama variabel array
 - c. Subskrip / index array

- ❖ **Jenis Array** (yang akan dipelajari) adalah :
 - a. Array Dimensi Satu (One Dimensional Array)
 - b. Array Dimensi Dua (Two Dimensional Array)
 - c. Array Dimensi Tiga (Thee Dimensional Array)

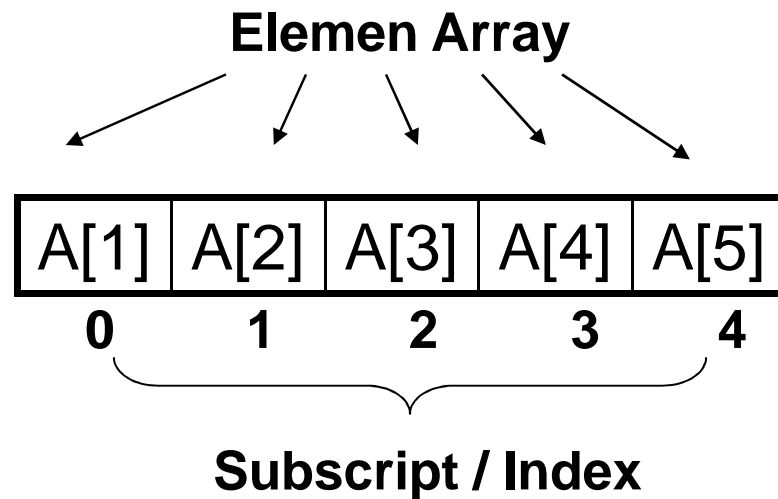
1. ARRAY DIMENSI SATU (One Dimensional Array)

Dapat disebut juga dengan istilah vektor yang menggambarkan data dalam suatu urutan

Deklarasi : Type_Data Nama_Variabel [index]

Misalnya : int A[5];

Penggambaran secara Logika :



- Contoh aplikasi untuk Array dimensi 1 adalah seperti program input bilangan dibawah ini

input 5 bilangan genap : bilangan 1 = 45
 bilangan 2 = 50
 bilangan 3 = 100
 bilangan 4 = 75
 bilangan 5 = 30

Dengan hasil output sebagai berikut :

45	50	100	75	30
----	----	-----	----	----

```
#include <iostream.h>
#include <conio.h>

void main()
{
float bil [5];
clrscr;

cout<<"Masukkan 5 bilangan genap : "<<endl;
for (int i = 0; i < 5; i++)
{
cout<< i + 1 <<" : ";
cin>> bil[i];
cout<<endl;
}
cout<<"5 bilangan genap yang dimasukkan : "<<endl;
for (int i = 0; i < 5; i++)
cout<<" "<<bil[i];
getch();
}
```

Rumus untuk menentukan jumlah elemen dalam Array :

$$\prod_{i=1}^n (\text{Elemen Array})$$

π = Perkalian dari elemen sebelumnya
(untuk array dimensi dua & tiga)

Contoh :

Suatu Array A dideklarasikan sbb :

int A[10]; maka jumlah elemen Array dimensi satu tersebut
adalah = 10

PEMETAAN (MAPPING) ARRAY DIMENSI SATU KE STORAGE

$$\text{Rumus : } @A[i] = B + (i - 1) * L$$

Dimana : @A[i] : Posisi Array yg dicari

B : Posisi awal index di memory komputer

i : Subkrip atau indeks array yg dicari

L : Ukuran / Besar memory suatu type data

Contoh :

Suatu Array A dideklarasikan sebagai berikut :

int A[5]; dengan alamat awal index berada di 0011_(H) dan ukuran memory type data integer = 2

Tentukan berapa alamat array A[3] ?

$$\text{Rumus : } @A[i] = B + (i - 1) * L$$

Diketahui :

$$\begin{aligned} @A[i] &= A[3] \\ B &= 0011 \text{ (H)} \\ i &= 3 \\ L &= 2 \end{aligned}$$

Penyelesaian :

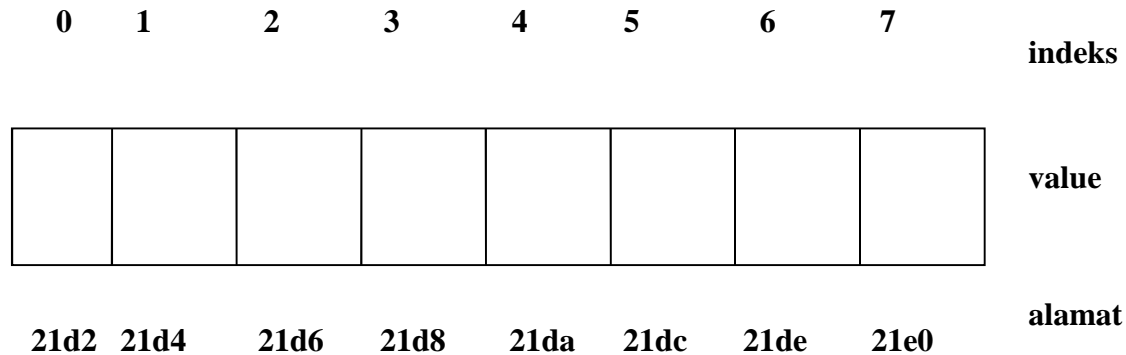
$$\begin{aligned} A[3] &= 0011\text{(H)} + (3 - 1) * 2 \\ &= 0011\text{(H)} + 4 \text{ (D)} \\ &= 0011\text{(H)} + 4 \text{ (H)} \\ &= 0015\text{(H)} \end{aligned}$$

4 Desimal = 4 Hexa

0	1	2	3	4
A[1]	A[2]	A[3]	A[4]	A[5]
0011	0013	0015	0017	0019



Contoh Penerapan Array Dimensi 1 Pada Program C++



```
#include <stdio.h>

void main() {
    int a[8];
    for (int i=0; i<8; i++) {
        printf("%x\n", &a[i]); %x adalah hexadesimal
    }
}
```

2. ARRAY DIMENSI DUA (Two Dimensional Array)

Sering digunakan dalam menterjemahkan matriks pada pemrograman.

Deklarasi : Type_Data Nama_Variabel [Index1] [index2];

Misal : int A[3][2];

Penggambaran secara Logika :

	0	1
0		
1		
2		

- Contoh aplikasi untuk Array dimensi 2 adalah seperti program input IPK mahasiswa dengan hasil output seperti berikut :

Nama Mahasiswa	IPK
Abdillah	3.50
Budiman	2.76
Indah	3.25
Khalilah	2.81
Nadya	2.93

Menentukan jumlah elemen dalam Array dimensi dua:

$$\sum_{i=1}^n \pi$$

(Elemen array)

π = Perkalian dari elemen sebelumnya
(untuk array dimensi dua & tiga)

Contoh :

Suatu Array X dideklarasikan sbb :

```
int X[4][3];
```

maka jumlah elemen Array dimensi dua tersebut adalah :

$$(4) * (3) = 12$$

PEMETAAN (MAPPING) ARRAY DIMENSI DUA KE STORAGE

Terbagi Dua cara pandang (representasi) yang berbeda :

1. Secara Kolom Per Kolom (Coloumn Major Order/CMO)

$$@M[i][j] = M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

2. Secara Baris Per Baris (Row Major Order / RMO)

$$@M[i][j] = M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

Keterangan :

@M[i][j] = Posisi Array yg dicari, M[0][0] = Posisi alamat awal index array, i = Baris, j = kolom, L = Ukuran memory type data

K = Banyaknya elemen per kolom, N = Banyaknya elemen per baris

Penggambaran secara logika

Misal : int M[3][2];
(Array dengan 3 Baris & 2 Kolom)

	0	1
0		
1		
2		

Berdasarkan Cara pandang :

1. Kolom Per Baris (Row Major Order / RMO)

M[0,0]	M[0,1]	M[1,0]	M[1,1]	M[2,0]	M[2,1]
--------	--------	--------	--------	--------	--------

Jumlah elemen per baris = 2

2. Baris Per Kolom (Coloumn Major Order / CMO)

M[0,0]	M[1,0]	M[2,0]	M[0,1]	M[1,1]	M[2,1]
--------	--------	--------	--------	--------	--------

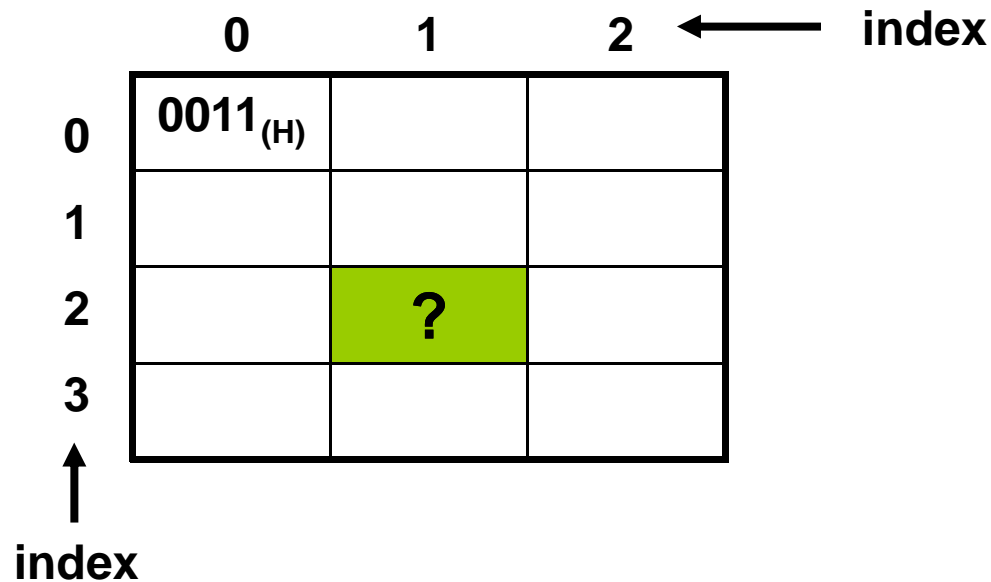
Jumlah elemen per kolom = 3

Contoh Pemetaan :

Suatu Array X dideklarasikan sebagai berikut :

Float $X[4][3]$, dengan alamat index $X[0][0]$ berada di $0011_{(H)}$
dan ukuran type data float/real = 4

Tentukan berapa alamat array $X[3][2]$ berdasarkan cara pandang baris dan kolom ?



Lanjutan Contoh Pemetaan :

Penyelesaian :

Secara Baris Per Baris (Row Major Oder / RMO)

$$@M[i][j] = @M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

$$X[3][2] = 0011_{(H)} + \{(3 - 1) * 3 + (2 - 1)\} * 4$$

$$= 0011_{(H)} + 28_{(D)} \longrightarrow 1C_{(H)}$$

$$= 0011_{(H)} + 1C_{(H)}$$

$$= 002D_{(H)}$$

Lanjutan Contoh Pemetaan :

Penyelesaian :

Secara Kolom Per Kolom (Coloumn Major Oder / CMO)

$$@M[i][j] = @M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

$$X[3][2] = 0011(H) + \{(2 - 1) * 4 + (3 - 1)\} * 4$$

$$= 0011(H) + 24 (D) \longrightarrow 18 (H)$$

$$= 0011(H) + 18 (H)$$

$$= 0029(H)$$

Contoh Penerapan Array Dimensi 2 Pada Program C++

```
#include <stdio.h>
#include <conio.h>

int main() {
    char a[3][5];
    for (int i=0; i<3; i++) {
        for (int j=0; j<5; j++) {
            printf("%x ", &a[i][j]);
        }
        printf("\n");
    }
    getch();
    return 0;
}
```

21d4	21d5	21d6	21d7	21d8
21d9	21da	21db	21dc	21dd
21de	21df	21e0	21e1	21e2