

Pertemuan 1

Konsep Struktur Data & Array

STRUKTUR DATA

Struktur Data adalah : suatu koleksi atau kelompok data yang dapat dikarakteristikan oleh organisasi serta operasi yang didefinisikan terhadapnya.

Pemakaian Struktur Data yang tepat didalam proses pemrograman, akan menghasilkan Algoritma yang lebih jelas dan tepat sehingga menjadikan program secara keseluruhan lebih sederhana.

Pada garis besarnya, Data dapat dikategorikan menjadi :

A. Type Data Sederhana / Data Sederhana

Terdiri dari :

1. Data Sederhana Tunggal

Misalnya : Integer, Real/Float, Boolean dan Character

2. Data Sederhana Majemuk

Misalnya : String

B. Struktur Data

Terdiri dari :

1. Struktur Data Sederhana

Misalnya Array dan Record

2. Struktur Data Majemuk

Terdiri dari :

a. Linier

Misalnya : Stack, Queue dan Linear Linked List.

b. Non Linier

Misalnya : Pohon (Tree), Pohon Biner (Binary Tree), Pohon Cari Biner (Binary Search Tree), General Tree serta Graph.

Array Dimensi 1 & 2

- ❖ Array / Larik : 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.

Definisi Array

- ❖ 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

Definisi Array

- ❖ **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)

Array Dimensi Satu

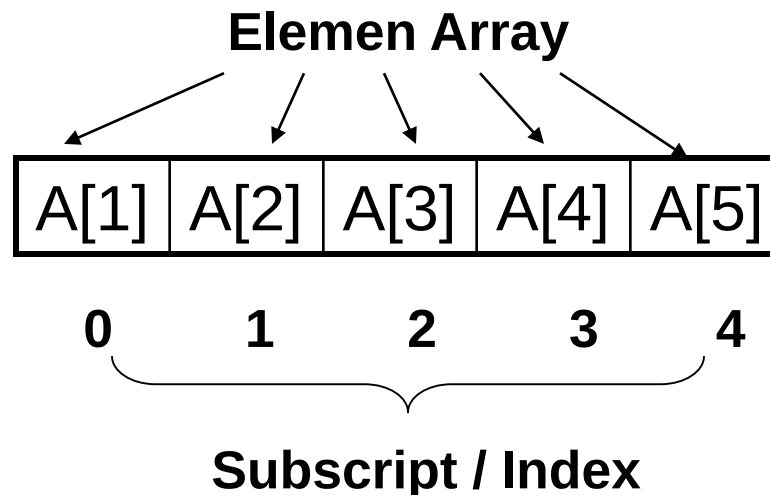
1.ARRAY DIMENSI SATU (One Dimensional Array)

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 :



Array Dimensi Satu

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

C:\ D:\CPP\array1.exe

Masukkan 5 bilangan genap :

Bilangan 1 : 45

Bilangan 2 : 50

Bilangan 3 : 100

Bilangan 4 : 75

Bilangan 5 : 30

5 bilangan genap yang dimasukkan :

45 50 100 75 30

Array Dimensi Satu

Rumus untuk menentukan jumlah elemen dalam Array :

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

(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] ?

PEMETAAN (MAPPING) ARRAY DIMENSI SATU KE STORAGE

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

Diketahui :

$$@A[i] = A[3]$$

$$B = 0011 \text{ (H)}$$

$$i = 3$$

$$L = 2$$

Penyelesaian :

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

$$= 0011_{(H)} + 4 \text{ (D)}$$

$$= 0011_{(H)} + 4 \text{ (H)}$$

$$= 0015_{(H)}$$

4 Desimal = 4 Hexa

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



KONVERSI BILANGAN

1. Decimal adalah bilangan berbasis sepuluh yang terdiri dari 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9
2. Hexadecimal adalah bilangan berbasis enam belas yang terdiri dari 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F

Tabel di bawah adalah contoh **konversi bilangan** Decimal, dan Hexadecimal

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Contoh KONVERSI ANTAR BILANGAN

Konversi Bilangan Decimal ke Hexadecimal

Contoh $254_{(10)} = \dots\dots(16)$

Caranya dengan membagi bilangan tersebut dengan enam belas sampai bilangan tersebut tidak bisa lagi dibagi enam belas (kurang dari enam belas) dengan mencatat setiap sisa pembagian.

$254 : 16 = 15$ sisa 14 atau E (lihat tabel di atas)

$15 : 16 =$ sisa 15 atau F (lihat tabel di atas)

Jadi $254_{(10)} = FE_{(16)}$ diurutkan dari sisa pembagian terakhir.

Contoh Penerapan Array Dimensi 1 Pada Program C++

0	1	2	3	4	5	6	7	indeks
								value
21d2	21d4	21d6	21d8	21da	21dc	21de	21e0	alamat

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

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

Array Dimensi Dua

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		

Array Dimensi Dua

Menentukan jumlah elemen dalam Array dimensi dua:

n
 π (Elemen array)
 $i=1$

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 = 4

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

	0	1	2	← index
0	0011 _(H)			
1				
2				
3				

↑ index

Lanjutan Contoh Pemetaan :

Penyelesaian :

Secara Baris Per Baris (Row Major Oder / RMO)

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

$$\begin{aligned} X[3][2] &= 0011_{(H)} + \{(3 - 1) * 3 + (2 - 1)\} * 4 \\ &= 0011_{(H)} + 28_{(D)} \quad \longrightarrow \quad 1C_{(H)} \\ &= 0011_{(H)} + 1C_{(H)} \end{aligned}$$

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

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 program array dua dimensi

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a[3][5];
    for (int i=0;i<3;i++)
    {
        for (int j=0;j<5;j++)
        {
            printf("%x ",&a[j][i]);
        }
        printf("\n");
    }
    getch();
}
```

```
12ff50 12ff64 12ff78 12ff8c 12ffa0
12ff54 12ff68 12ff7c 12ff90 12ffa4
12ff58 12ff6c 12ff80 12ff94 12ffa8
```

Contoh program array dua dimensi

```
#include<stdio.h>
#include<conio.h>
main()
{
    int 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();
}
```

```
12ff50 12ff54 12ff58 12ff5c 12ff60
12ff64 12ff68 12ff6c 12ff70 12ff74
12ff78 12ff7c 12ff80 12ff84 12ff88
```

Latihan :

1. Suatu array A dideklarasikan sbb :
int A[50] dengan alamat awal berada di 0011(H). Tentukan berapa alamat array A[20] dan A[40]?
2. Suatu array X dideklarasikan sbb :
Float X[4][5] dengan alamat awal berada pada 0011(H). Tentukan berapa alamat array X[4][3], berdasarkan cara pandang baris dan kolom?

Latihan Soal Struktur Data (Pertemuan 1)

1. Tipe data int memiliki nilai jangkauan antara.....
 - a. -32768..32767
 - b. 32766..32829
 - c. -32500..32500
 - d. 32342..32768
 - e. 32342...32829

2. Tipe Data yang terdiri dari aksara (simbol) yang meliputi digit numerik, alfabetik adalah tipe data?.....
 - a. Float
 - b. Integer
 - c. BOOL
 - d. Character
 - e. String

2. Tipe Data yang terdiri dari aksara (simbol) yang meliputi digit numerik, alfabetik adalah tipe data?.....
- a. Float
 - b. Integer
 - c. BOOL
 - d. Character
 - e. String

```
3. void main()
{
....(a).... x,y,z;
clrscr();
cout<<"\n input nilai X="; cin >>x;
cout<<"\n input nilai Y="; cin >>y;
z = x + y;
cout <<"\n hasil penjumlahan ="<<z;
getch();
}
```

Type data yang tepat untuk (a) adalah

- a. Boolean b. String
a. c. Char d. Integer e. Array

4. Terdapat Array : $A[2][2]$ maka jumlah elemen Array tersebut adalah
- a. 2
 - b. 4
 - c. 22
 - d. 15
 - e. 20
5. Diketahui float $A[5]$ dan lokasi awal terletak di alamat $00F_{(H)}$, maka lokasi $A[3]$ adalah
- a. $00FC_{(H)}$
 - b. $017_{(H)}$
 - c. $071_{(H)}$
 - d. $01B_{(H)}$
 - e. $111_{(H)}$

5. Diketahui float $A[5]$ dan lokasi awal terletak di alamat $00F_{(H)}$, maka lokasi $A[3]$ adalah

a. $00FC_{(H)}$

b. $017_{(H)}$

c. $071_{(H)}$

d. $01B_{(H)}$

e. $111_{(H)}$

1. Tipe data int memiliki nilai jangkauan antara.....

a. **-32768..32767**

b. 32766..32829

c. -32500..32500

d. 32342..32768

e. 32342...32829