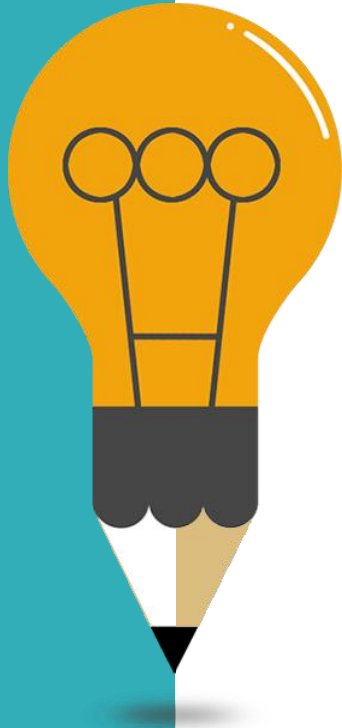


ARRAY

BY
TAUFIK HIDAYAT, MT.

TODAY'S LESSON



01

DEFINISI ARRAY

02

DIMENSI ARRAY

03

MAPPING ARRAY

04

CONTOH PROGRAM

DEFINISI ARRAY



❑ Sekumpulan data yang memiliki tipe data yang sama dapat disimpan secara beruntun dalam memori komputer. Struktur penyimpanan data seperti ini dinamakan

larik (*array*)

❑ Setiap elemen dapat diakses langsung melalui indeksnya.

Contoh deklarasi array:

```
int a[5];
```

Kode di atas mendeklarasikan array bertipe integer sebanyak 5 elemen. Indeks untuk mengakses setiap elemen adalah:

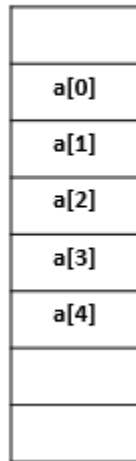
```
a[0], a[1], a[2], a[3], a[4]
```

DEFINISI ARRAY



- Dari kode sebelumnya pada saat array dideklarasikan maka yang terjadi pada memori adalah menyiapkan tempat yang berurutan sebanyak jumlah array yang dideklarasikan.

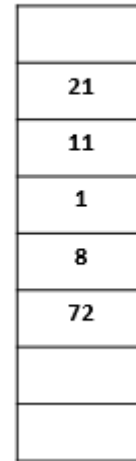
```
int a[5];
```



Alamat array a
dalam memori
berurutan

Misal dilakukan
pengisian nilai
ke dalam array:

a[0]=21
a[1]=11
a[2]=1
a[3]=8
a[4]=72



array a diisi
dengan nilai
integer

DEFINISI ARRAY



- ❑ Array adalah struktur data yang **statis**
Jumlah elemen larik harus sudah diketahui (dituliskan) sebelum kode program dieksekusi.
 - ✓ Jumlah elemen tidak dapat ditambah dan dikurangi selama program berjalan.

- ❑ Saat melakukan deklarasi array, yang dilakukan adalah:
 - ✓ Mendefinisikan banyaknya elemen array
 - ✓ Mendefinisikan tipe data array

DEFINISI ARRAY



- ❑ Pengaksesan elemen array dilakukan dengan menggunakan notasi :
 - *Nama_array[indeks]*
- ❑ Dalam hal ini dimulai dari 0(nol)
Contoh :
Cacah[0] = 1;
- ❑ Yang berarti pernyataan untuk mengisi nilai 1 ke elemen pertama pada array cacah.

KELEBIHAN VS KEKURANGAN ARRAY



Kelebihan

- ❑ Nilai yang disimpan di array terletak pada alamat yang urut.
- ❑ Dapat diakses langsung menggunakan urutan indeks elemen.

Kekurangan

- ❑ Penggunaan memori yang tidak dinamis
 - Jumlah tempat yang disediakan dalam memori harus sebanyak elemen yang dituliskan.
 - Tempat memori tersebut tidak dapat digunakan untuk sementara oleh CPU untuk melakukan proses yang lain.

DIMENSI ARRAY



ARRAY DIMENSI SATU

Merupakan Bentuk sederhana dari array. Setiap elemen memiliki satu subskrip.

Fungsi subskrip :

- ✓ Menyatakan posisi elemen
- ✓ Membedakan antar elemen



DIMENSI ARRAY



ARRAY DIMENSI SATU

- ❑ Merupakan Bentuk sederhana dari array. Setiap elemen memiliki satu subskrip.

Fungsi subskrip :

- ✓ Menyatakan posisi elemen
- ✓ Membedakan antar elemen

- ❑ Bentuk Umum

$A(L : U)$

- ❑ Banyak Elemen

$U - L + 1$

DIMENSI ARRAY



STRUKTUR DATA - ARRAY DIMENSI SATU

Array A satu dimensi :

8 indeks (1 s/d 8) dan data 1, 7, 18 dst.

1	7	18	03	69	24	08	70
1	2	3	4	5	6	7	8

DIMENSI ARRAY



ARRAY DIMENSI DUA

- ☐ Setiap elemen memiliki dua subskrip.
- ☐ Setiap elemen adalah sebuah array.

Fungsi subskrip :

- 1 : Menyatakan baris
- 2 : Menyatakan kolom

- ☐ Bentuk Umum
 $A (L1 : U1, L2 : U2)$

- ☐ Banyak Elemen
 $(U1 - L1 + 1) * (U2 - L2 + 1)$

DIMENSI ARRAY



STRUKTUR DATA - ARRAY DIMENSI DUA

Array B dua dimensi (matriks) :

- jumlah baris 2, kolom 3
- data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	03	69
2	24	08	70

DIMENSI ARRAY



ARRAY DIMENSI TIGA

- ❑ Setiap elemen memiliki 3 subskrip.
- ❑ Setiap elemen adalah sebuah array.

Fungsi subskrip :

- 1 : Menyatakan baris
- 2 : Menyatakan kolom
- 3 : Menyatakan tinggi

- ❑ Bentuk Umum

$A (L1 : U1, L2 : U2, L3 : U3)$

- ❑ Banyak Elemen

$(U1 - L1 + 1) * (U2 - L2 + 1) * (U3 - L3 + 1)$

MAPPING ARRAY



ARRAY DIMENSI SATU

Hal yang perlu diketahui untuk melihat alamat awal elemen array :

- Base Location
- Size

Rumus yang digunakan...

$$\mathbf{B + (I - L) * S}$$

MAPPING ARRAY



ARRAY DIMENSI DUA

Bentuk Umum Array...

$A (L1, U1 : L2, U2)$

Bentuk Umum Elemen Array...

$A (I, J)$

Dua cara melinierkan array dimensi dua

1 : Row Major

$$B + (I-L1) * (U2-L2+1) * S + (J-L2) * S$$

2 : Coloum Major

$$B + (J-L2) * (U1-L1+1) * S + (I-L1) * S$$

MAPPING ARRAY



Array B dua dimensi (matriks) :

- jumlah baris 2, kolom 3
- data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	3	69
2	24	8	70



MAPPING ARRAY



Array B dua dimensi (matriks) :

- jumlah baris 2, kolom 3
- data 18, 03, 69, 24, 08, 70.

	1	2	3
1	18	3	69
2	24	8	70

MAPPING ARRAY



TRIANGULAR ARRAY

2 Jenis Triangular Array:

- ✓ Upper Triangular
Array dengan elemen di bawah diagonal utama adalah 0.
- ✓ Lower Triangular Array dengan elemen di atas diagonal utama adalah 0.

MAPPING ARRAY



SPARSE ARRAY

- ❑ Suatu array yang elemen tidak nol nya sangat sedikit.
- ❑ Format penyimpanan elemen tidak nol
(row, coloum, value)

CONTOH PROGRAM (1)



- ❑ Ketikkan kode berikut:

```
#include <stdio.h>

void main()
{
    int i, a[5];           //deklarasi array
    for(i=0;i<5;i++)       //perulangan untuk inisialisasi array
    {
        a[i]=0;           //inisialisasi array bertipe integer dengan nilai 0.
    }
    for(i=0;i<5;i++)       //perulangan untuk menampilkan isi array
    {
        printf("%d\n",a[i]);
    }
}
```

- ❑ Apa yang terjadi jika kode tersebut dieksekusi?
 - Apa kegunaan dari variabel i pada perulangan for pertama dan kedua?

CONTOH PROGRAM (1)



- ❑ Array dapat diisi dengan elemen yang sesuai dengan tipe datanya
- ❑ Isi elemen dapat berupa nilai yang ditentukan pada kode program atau dari inputan user.

CONTOH PROGRAM (2)



- ❑ Ketikkan kode berikut:

```
#include <stdio.h>
#define MAX 5          //mendefinisikan variabel MAX dengan nilai 5 (nama variabel bebas)

void main()
{
    int i, a[MAX];      //deklarasi array
    for(i=0;i<MAX;i++)  //perulangan untuk inisialisasi
    {
        a[i]=0;
    }
    a[0]=2;
    a[MAX-3]=5;
    for(i=0;i<MAX;i++)
    {
        printf("%d\n",a[i]);
    }
}
```

- ❑ Apa yang terjadi jika kode tersebut dieksekusi?
 - Elemen pada indeks berapa sajakah yang diganti nilainya?
 - Berapa isi array tersebut setelah diganti?

CONTOH PROGRAM (3)



- ❑ Ketikkan kode berikut:

```
#include <stdio.h>
#define MAX 5          //mendefinisikan variabel MAX dengan nilai 5 (nama variabel bebas)

void main()
{
    int i, a[MAX];      //deklarasi array
    for(i=0;i<MAX;i++)  //perulangan untuk inisialisasi
    {
        a[i]=0;
    }

    for(i=0;i<MAX;i++)  //perulangan untuk mengisi array
    {
        printf("Input Nilai: ");scanf("%d",&a[i]);
    }
    puts("Isi Array");
    for(i=0;i<MAX;i++)  //perulangan untuk menampilkan
    {
        printf("%d\n",a[i]);
    }
}
```

- ❑ Apa yang terjadi jika kode tersebut dieksekusi?
- ❑ Apakah nilai array yang ditampilkan sesuai dengan yang diinputkan user?

LATIHAN



Buatlah algoritma dan program yang menyimpan pasangan data Negara dan ibukota seperti berikut :

Indonesia	_____	Jakarta
Filipina	_____	Manila
Prancis	_____	Paris
Pakistan	_____	Islamabad

LATIHAN



ALGORITMA

- Negara[0][0] = "Indonesia"
- Negara[0][1] = "jakarta"
- Negara[1][0] = "Filipina"
- Negara[1][1] = "Manila"
- Negara[2][0] = "Prancis"
- Negara[2][1] = "Paris"
- Negara[3][0] = "Pakistan"
- Negara[3][1] = "Islamabad"
- Untuk baris = 0 s/d 4
- Jika Negara[baris][0][0] = "P" maka
 - Tampilkan(Negara[baris][0], Negara[baris][1])
- Akhir jika
- Akhir untuk

LATIHAN



ALGORITMA

```
#include <iostream.h>
#include<string.h>
Int main(){
Char Negara[4][2][15] = {{“Indonesia”,”jakarta”},{“Filipina”,”Manila”},{“Prancis”,”paris”},{“Pakistan”,”Isla
mabad”}};
For (int baris = 0;baris <4 ;baris++){
    If (Negara[baris][0][0]==’P’){
        Cout<< Negara[baris][0] << “ – “
        <<Negara[baris][1] << “\n”;
        Return 0;
    }
}
}
```



Q & A