

BAB XI ARRAY (LARIK)

1. PENDAHULUAN

Bab ini memiliki kompetensi dasar untuk mendefinisikan dan menggunakan tipe data array. Tipe data array adalah salah satu tipe terstruktur dalam pemrograman. Tipe data array ini dapat dipakai untuk salah satu contohnya adalah perhitungan matriks dalam matematika (bahasan matriks akan diperdalam pada kuliah algoritma pemrograman 2).

2. PENYAJIAN

2.1. Pengertian Array

Array atau larik adalah koleksi data dimana setiap elemen memakai nama yang sama dan bertipe sama dan setiap elemen diakses dengan membedakan indeks arraynya.

Array adalah variabel berindeks. Indeks harus bertipe yang memiliki keturunan (ada suksesor dan predesor), misal integer, byte, character dan boolean.

Jadi array dipakai untuk menyajikan sekumpulan data yang bertipe sama dan disimpan dengan urutan sesuai dengan indeks secara kontinyu.

2.2. Deklarasi Array Dimensi Satu

Algoritma :

Bentuknya :

```
DEKLARASI
    NamaArray : Array[range_indeks]of tipe
Atau
    TipeArray : TYPE Array[range_indeks]of tipe
    NamaArray : TipeArray
```

Cara mengakses suatu elemen :

```
NamaArrayindeks
```

Contoh :

X : Array[1..10] of integer

Artinya : mendefinisikan 10 variabel bertipe integer

Yaitu : $X_1, X_2, X_3, \dots X_{10}$

Contoh lain :

NamaHari : Array [1..7] of String

Nilai : Array [1..10] of Char

Frekuensi : Array['A'..'E'] of Real

Cara Memberikan Nilai/Harga pada Array

NilaiMka : Array[1..10] of Char

NilaiMka₁ ← 'A'

NilaiMka₂ ← 'C'

NilaiMka₃ ← ' '

Bahasa C++ :

Variabel array dideklarasikan dengan mencantumkan tipe dan nama variabel yang diikuti dengan banyaknya lokasi memori yang ingin dibuat.

Bentuknya :

```
tipe nama_var[ukuran];
```

Dengan :

- tipe : menyatakan jenis elemen array (int, char, unsigned, dan lain-lain)
- ukuran : menyatakan jumlah maksimal elemen array

Contoh :

- `int c[5];`

C++ secara otomatis akan menyediakan lokasi memori sesuai dengan yang dideklarasikan, dimana nomor indeks selalu dimulai dari 0.

- `int c[5] = {-12, 0, 20, 85, 1551};`

Nilai suatu variabel array dapat juga diinisialisasi secara langsung seperti yang terdapat di dalam tanda kurung kurawal pada saat deklarasi di atas.

- `int x[5] = {0};`

Deklarasi variable array sekaligus mengisi setiap lokasi memorinya dengan nilai 0.

Contoh Algoritma :

Algoritma Array1D

DEKLARASI

N : array[1..5] of integer

i : integer

DESKRIPSI

$N_1 \leftarrow 25$

$N_2 \leftarrow 12$

$N_3 \leftarrow 17$

$N_4 \leftarrow 10$

$N_5 \leftarrow 15$

For i \leftarrow 1 to 5 do

Output (N_i)

endfor

Contoh Program dalam bahasa C++ :

```
#include<iostream.h>
main()
{
    int N[5]={25,12,17,10,15};
    int i;
    for(i=0; i<=4; i++)
        cout << N[i];
}
```

atau :

```
#include<iostream.h>
main()
{
    int N[5]={25,12,17,10,15};
    int i;
    for(i=0; i<=4; i++)
        printf("%d \n",N[i]);
}
```

Output :

25
12
17
10
15

B. Array Dimensi Dua**Algoritma :**

Array dua dimensi hampir sama dengan array berdimensi satu, namun biasanya array berdimensi dua banyak digunakan untuk penyajian data berbentuk tabel atau juga berbentuk matriks.

Bentuknya :

DEKLARASI

```
NamaArray : TipeElemen Array[r_indeks1, r_indeks2]
```

Cara mengakses suatu elemen :

```
NamaArrayindeks1, indeks2
```

Contoh

	1	2	3	4
1	10	1	11	15
2	20	2	21	25
3	30	3	31	35
4	40	4	41	45

$\text{Harga}_{1,1} \leftarrow 10$

$\text{Harga}_{2,3} \leftarrow 21$

$\text{Harga}_{3,1} \leftarrow 30$

Bahasa C++ :

Bentuknya :

```
tipe nama_var[ukuran1][ukuran2];
```

ukuran1 = jumlah baris

ukuran 2 = jumlah kolom

Contoh :

```
int data_huruf[2][4];
```

Contoh :

Sebuah matrik A berukuran 2x3 dapat dideklarasikan sebagai berikut :

int a[2][3] = {{11, 7, 4},{12, 3, 9}} yang akan menempati lokasi memori dengan susunan berikut :

	0	1	2
0	11	7	4
1	12	3	9

Dan definisi variabel untuk setiap elemen tersebut adalah :

	0	1	2
0	a[0][0]	a[0][1]	a[0][2]
1	a[1][0]	a[1][1]	a[1][2]

Contoh Program dalam bahasa C++:

```

/* Program : array.cpp */
#include<iostream.h>
void printArray(int [][][3]);

main()
{
    int matrik1 [2][3] = { {1, 1, 1}, {2, 2, 2}};
    int matrik2 [2][3] = { {3, 3, 3}, {4, 4, 4}};
    int matrik3 [2][3] = { {5, 5, 5}, {6, 6, 6}};
    printArray(matrik1);
    printArray(matrik2);
    printArray(matrik3);
    return 0;
}

void printArray(int a[][][3])
{
    int i, j;

    for(i=0; i<=1; i++)
        {
            for(j=0; j<=2; j++)
                cout << a[i][j]<< " ";
            cout << endl;
        }
}

```

atau :

```

/* Program : array.cpp */
#include<iostream.h>
void printArray(int [][][3]);

main()
{
    int matrik1 [2][3] = { {1, 1, 1}, {2, 2, 2}};
    int matrik2 [2][3] = { {3, 3, 3}, {4, 4, 4}};
    int matrik3 [2][3] = { {5, 5, 5}, {6, 6, 6}};
    printArray(matrik1);
    printArray(matrik2);
    printArray(matrik3);
    return 0;
}

void printArray(int a[][][3])
{
    int i, j;

    for(i=0; i<=1; i++)
        {
            for(j=0; j<=2; j++)
                printf("%d ",a[i][j]);
            printf("\n");
        }
}

```

Output :

```

1 1 1
2 2 2
3 3 3
4 4 4
5 5 5
6 6 6

```

C. Array Dimensi Banyak**Algoritma :**

Bentuknya :

DEKLARASI

```
NamaArray : TipeElemen Array[r_indeks1, r_indeks2,... , r_indeksn]
```

Cara mengakses suatu elemen :

```
NamaArrayindeks1, indeks2, indeks3
```

Bahasa C++ :

Bentuknya :

```
tipe nama_var[ukuran1][ukuran2] . . .[ukuranN];
```

Contoh :

```
int data_huruf[2][4][6];
```

3. PENUTUP

Latihan

1. Diketahui data-data sebagai berikut :

Nama	Nilai Ujian
Adi	8
Budi	7
Charlie	5
Desak	6
Sari	8

Buatlah program untuk membaca nilai-nilai tersebut dan mencari siapa sajakah yang lulus ujian (syarat lulus ujian, nilai ujian ≥ 6) .

2. Dari soal no. 1 buat program untuk menghitung rata-rata nilai ujian.

3. Diketahui matriks A dan matriks B sebagai berikut :

$$A = \begin{bmatrix} 4 & 7 & 8 \\ 2 & 3 & 6 \\ 1 & 5 & 9 \end{bmatrix}$$

$$B = \begin{bmatrix} 3 & 5 & 1 \\ 2 & 4 & 7 \\ 8 & 6 & 9 \end{bmatrix}$$

Buatlah program untuk menjumlahkan matriks C = matriks A + matriks B .