

PRAKTIKUM 18

Array 1

A. TUJUAN PEMBELAJARAN

1. Memahami konsep array dan penyimpanannya dalam memori
2. Mempelajari penggunaan variabel array berdimensi satu
3. Memahami penggunaan variabel array berdimensi dua
4. Dapat menerapkan penggunaan array berdimensi satu dan dua pada program sederhana

B. DASAR TEORI

Dalam beberapa literatur, array sering disebut (diterjemahkan) sebagai larik. Array adalah kumpulan dari nilai-nilai data bertipe sama dalam urutan tertentu yang menggunakan sebuah nama yang sama. Nilai-nilai data di suatu array disebut dengan elemen-elemen array. Letak urutan dari elemen-elemen array ditunjukkan oleh suatu *subscript* atau indeks.

Mendeklarasikan Array Berdimensi Satu

Suatu array berdimensi satu dideklarasikan dalam bentuk umum berupa :

```
tipe_data nama_var[ukuran];
```

dengan :

- `tipe_data` : untuk menyatakan tipe dari elemen array, misalnya *int*, *char*, *float*.
- `nama_var` : nama variabel array
- `ukuran` : untuk menyatakan jumlah maksimal elemen array.

Contoh pendeklarasian array :

```
float nilai_tes[5];
```

menyatakan bahwa array **nilai_tes** mengandung 5 elemen bertipe *float*.

Mengakses Elemen Array Berdimensi Satu

Pada C, data array akan disimpan dalam memori yang berurutan. Elemen pertama mempunyai indeks bernilai 0. Jika variabel **nilai_tes** dideklarasikan sebagai array dengan 5 elemen, maka elemen pertama memiliki indeks sama dengan 0, dan elemen terakhir memiliki indeks 4. Bentuk umum pengaksesan array adalah sbb :

nama_var[indeks]

sehingga, untuk array **nilai_tes**, maka :

nilai_tes[0] → elemen pertama dari **nilai_tes**
nilai_tes[4] → elemen ke-5 dari **nilai_tes**

Contoh :

```
nilai_tes[0] = 70;  
scanf("%f", &nilai_tes[2]);
```

Contoh pertama merupakan pemberian nilai 70 ke **nilai_tes[0]**. Sedangkan contoh 2 merupakan perintah untuk membaca data bilangan dari keyboard dan diberikan ke **nilai_tes[2]**. Pada contoh 2 ini

&nilai_tes[2]

berarti “alamat dari **nilai_tes[2]**”. Perlu diingat bahwa *scanf()* memerlukan argumen berupa alamat dari variabel yang digunakan untuk menyimpan nilai masukan.

Inisialisasi Array Berdimensi Satu

Sebuah array dapat diinisialisasi sekaligus pada saat dideklarasikan. Untuk mendeklarasikan array, nilai-nilai yang diinisialisasikan dituliskan di antara kurung kurawal ({ }) yang dipisahkan dengan koma.

```
int jum_hari[12] =  
{31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
```

Beberapa Variasi dalam Mendeklarasikan Array

Ada beberapa variasi cara mendeklarasikan sebuah array (dalam hal ini yang berdimensi satu), di antaranya adalah sebagai berikut :

- `int numbers[10];`
- `int numbers[10] = {34, 27, 16};`
- `int numbers[] = {2, -3, 45, 79, -14, 5, 9, 28, -1, 0};`
- `char text[] = "Welcome to New Zealand.";`

- `float radix[12] = {134.362, 1913.248};`
- `double radians[1000];`

Array Berdimensi Dua

Array berdimensi satu dapat disimpan pada sebuah array berdimensi dua.

Pendeklarasian array berdimensi dua adalah sebagai berikut :

```
int data_lulus[4][3];
```

Nilai 4 untuk menyatakan banyaknya baris dan 3 menyatakan banyaknya kolom. Gambar 15.1 memberikan ilustrasi untuk memudahkan pemahaman tentang array berdimensi dua.

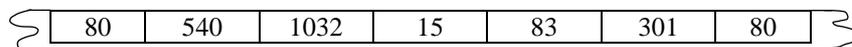
	0	1	2	← indeks kedua (kolom)
0	80	540	1032	
1	15	83	301	
2	8	12	15	
3	10	129	257	

↑ indeks pertama (baris)

```
int data_lulus[4][3];
```

Gambar 15.1 Array berdimensi dua

Sama halnya pada array berdimensi satu, data array akan ditempatkan pada memori yang berurutan. Perhatikan Gambar 15.2.



Gambar 15.2 Model penyimpanan array dimensi dua pada memori

Mengakses Elemen Array Berdimensi Dua

Array seperti **data_lulus** dapat diakses dalam bentuk `data_lulus[indeks pertama, indeks kedua]` :

(1) `data_lulus[0][1] = 540;`

merupakan instruksi untuk memberikan nilai 540 ke array **data_lulus** untuk indeks pertama = 0 dan indeks kedua bernilai 1.

(2) `printf("%d", data_lulus[2][0]);`

merupakan perintah untuk menampilkan elemen yang memiliki indeks pertama = 2 dan indeks kedua = 0.

C. TUGAS PENDAHULUAN

Untuk semua persoalan di bawah ini, desainlah algoritma dan flowchartnya.

D. PERCOBAAN

1. Deklarasikan sebuah **variabel *array of int***, **selanjutnya** isi array tsb kemudian tampilkan isi variabel tersebut menggunakan statement **for()**. Ingat apabila jumlah deklarasi *index* adalah **n** maka nilai *index*-nya adalah dimulai dari **0** sampai dengan **n-1**.
2. Deret fibonacci adalah deret yang dimulai dengan dua angka yang bernilai 0 dan 1, kemudian deret ketiga ditentukan dari penjumlahan kedua angka tersebut, sedangkan deret keempat ditentukan dari dua angka sebelumnya begitu seterusnya. Sehingga didapatkan deret fibonanci sebagai berikut: 0 1 1 2 3 5 8 13 21 ...
Buatlah program untuk meminta input dari user berupa sebuah bilangan, kemudian tampilkan deret fibonacci mulai dari 1 sampai dengan bilangan tsb.
3. Buat penjumlahan dua matriks $A[2][2]$ dengan $B[2][2]$. Hasil penjumlahan tersebut disimpan dalam matrik C. Tampilkan di layar masing-masing isi dari matriks A, B dan C
Catatan modifikasi (gunakan nested loop) :
 - indeks MAKS dibuat konstanta
 - variabel ordo matriks diinputkan
 - memungkinkan pengulangan proses dengan menginputkan ordo yang lain
4. Buatlah sebuah variabel *array of int* yang memiliki ukuran MAKS. Dengan menggunakan looping `for()`, masukkan sebanyak n data ke dalam array tsb kemudian carilah nilai terbesar yang ada dalam array tsb.

E. LAPORAN RESMI

1. Untuk setiap listing program dari percobaan-percobaan di atas, ambil *capture* outputnya.
2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.