PRAKTIKUM 9

ARRAY

A. Tujuan

1. Menjelaskan tentang array berdimensi satu
2. Menjelaskan tentang array berdimensi dua
3. Menjelaskan tentang array berdimensi banyak
4. Menjelaskan tentang inisialisasi array tak berukuran.
5. Menjelaskan array sebagai parameter fungsi

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.

Array bisa berupa array berdimensi satu, dua, tiga atau lebih. Array berdimensi satu *(one-dimensional array)* mewakili bentuk suatu vektor. Array berdimensi dua *(two-dimensional array)* mewakili bentuk dari suatu matriks atau table. Array berdimensi tiga *(three-dimensional array)* mewakili bentuk suatu ruang.

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];`

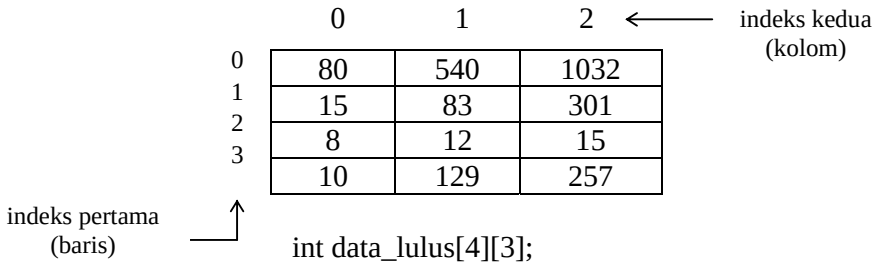
ARRAY BERDIMENSI DUA

Array berdimensi satu dapat disimpan pada sebuah array berdimensi dua.

Pendeklarasian array berdimensi dua adalah sebagai berikut :

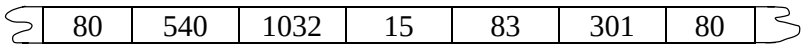
```
int data_lulus[4][3];
```

Nilai 4 untuk menyatakan banyaknya baris dan 3 menyatakan banyaknya kolom. Gambar 10.1 memberikan ilustrasi untuk memudahkan pemahaman tentang array berdimensi dua.



Gambar 10.1 Array berdimensi dua

Sama halnya pada array berdimensi satu, data array aka ditempatkan pada memori yang berurutan. Perhatikan Gambar 10.2.



Gambar 10.2 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.

Perhatikan contoh potongan program di bawah ini.

```
/* Memberikan data ke array */

     data_lulus[0][0] = 80;
     data_lulus[0][1] = 540;
     data_lulus[0][2] = 1032;

     INT HURUF_A[8][8] = {
             { 0, 1, 1, 1, 1, 1, 0, 0 } ,
             { 0, 1, 0, 0, 0, 1, 0, 0 } ,
             { 0, 1, 0, 0, 0, 1, 0, 0 } ,
             { 1, 1, 1, 1, 1, 1, 1, 0 } ,
             { 1, 1, 0, 0, 0, 0, 1, 0 } ,
             { 1, 1, 0, 0, 0, 0, 1, 0 } ,
             { 1, 1, 0, 0, 0, 0, 1, 0 } ,
             { 0, 0, 0, 0, 0, 0, 0, 0 }
        };
```

atau bisa juga ditulis sebagai berikut :

```
     int huruf_A[8][8] =
        {  0, 1, 1, 1, 1, 1, 0, 0,
           0, 1, 0, 0, 0, 1, 0, 0,
           0, 1, 0, 0, 0, 1, 0, 0,
           1, 1, 1, 1, 1, 1, 1, 0,
           1, 1, 0, 0, 0, 0, 1, 0,
           1, 1, 0, 0, 0, 0, 1, 0,
           1, 1, 0, 0, 0, 0, 1, 0,
           0, 0, 0, 0, 0, 0, 0, 0
        };
```

Array Berdimensi Banyak.

C memungkinkan untuk membuat array yang dimensinya lebih dari dua. Bentuk umum pendeklarasian array berdimensi banyak :

```
tipe nama_var[ukuran 1][ukuran2}…[ukuranN];
```

sebagai contoh :

```
int data_huruf[2][8][8];
```

merupakan pendeklarasian array **data_huruf** sebagai array berdimensi tiga.

Sama halnya dengan array berdimensi satu atau dua, array berdimensi banyak juga bisa diinisialisasi. Contoh inisialisasi array berdimensi tiga :

```
int data_huruf [2][8][8] =
     { { { 0, 1, 1, 1, 1, 1, 0, 0 } ,
         { 0, 1, 0, 0, 0, 1, 0, 0 } ,
         { 0, 1, 0, 0, 0, 1, 0, 0 } ,
         { 1, 1, 1, 1, 1, 1, 1, 0 } ,
         { 1, 1, 0, 0, 0, 0, 1, 0 } ,
         { 1, 1, 0, 0, 0, 0, 1, 0 } ,
         { 1, 1, 0, 0, 0, 0, 1, 0 } ,
         { 0, 0, 0, 0, 0, 0, 0, 0 }
       },
       { {1, 1, 1, 1, 1, 1, 0, 0 } ,
         {1, 0, 0, 0, 0, 1, 0, 0 } ,
         {1, 0, 0, 0, 0, 1, 0, 0 } ,
         {1, 1, 1, 1, 1, 1, 1, 0 } ,
         {1, 1, 0, 0, 0, 0, 1, 0 } ,
         {1, 1, 0, 0, 0, 0, 1, 0 } ,
         {1, 1, 1, 1, 1, 1, 1, 0 } ,
         {0, 0, 0, 0, 0, 0, 0, 0 }
       } };
```

atau bisa juga ditulis menjadi

```
int data_huruf [2][8][8] =
          0, 1, 1, 1, 1, 1, 0, 0,
          0, 1, 0, 0, 0, 1, 0, 0,
          0, 1, 0, 0, 0, 1, 0, 0,
          1, 1, 1, 1, 1, 1, 1, 0,
          1, 1, 0, 0, 0, 0, 1, 0,
          1, 1, 0, 0, 0, 0, 1, 0,
          1, 1, 0, 0, 0, 0, 1, 0,
          0, 0, 0, 0, 0, 0, 0, 0,
          1, 1, 1, 1, 1, 1, 0, 0,
          1, 0, 0, 0, 0, 1, 0, 0,
```

```
            1, 0, 0, 0, 0, 1, 0, 0,
            1, 1, 1, 1, 1, 1, 1, 0,
            1, 1, 0, 0, 0, 0, 1, 0,
            1, 1, 0, 0, 0, 0, 1, 0,
            1, 1, 1, 1, 1, 1, 1, 0,
            0, 0, 0, 0, 0, 0, 0, 0
          };

int i, j, k;
int data_huruf[2][8][8] = {
     {{ 0, 1, 1, 1, 1, 1, 0, 0 },
     { 0, 1, 0, 0, 0, 1, 0, 0 },
     { 0, 1, 0, 0, 0, 1, 0, 0 },
     { 1, 1, 1, 1, 1, 1, 1, 0 },
     { 1, 1, 0, 0, 0, 0, 1, 0 },
     { 1, 1, 0, 0, 0, 0, 1, 0 },
     { 1, 1, 0, 0, 0, 0, 1, 0 },
     { 0, 0, 0, 0, 0, 0, 0, 0 }
     },
     {{1, 1, 1, 1, 1, 1, 0, 0 },
     {1, 0, 0, 0, 0, 1, 0, 0 },
     {1, 0, 0, 0, 0, 1, 0, 0 },
     {1, 1, 1, 1, 1, 1, 1, 0 },
     {1, 1, 0, 0, 0, 0, 1, 0 },
     {1, 1, 0, 0, 0, 0, 1, 0 },
     {1, 1, 1, 1, 1, 1, 1, 0 },
     {0, 0, 0, 0, 0, 0, 0, 0 }
     }
};
```

ARRAY SEBAGAI PARAMETER

Array juga dapat dilewatkan sebagai parameter fungsi. Sebagai contoh ditujukan pada program **sorting.c**. Program digunakan untuk memasukkan sejumlah data, kemudian data tersebut diurutkan naik *(ascending)* dan dicetak ke layar.

Untuk melakukan sorting (proses pengurutan data), cara yang dipakai yaitu metode *buble sort* (suatu metode pengurutan yang paling sederhana, dan memiliki kecepatan pengurutan yang sangat lambat).

Algoritma pada metode pengurutan ini adalah sebagai berikut :

1. Atur i bernilai 0
2. Bandingkan x[i] dengan x[j], dg j berjalan dari i + 1 sampai dengan n-1.
3. Pada setiap pembandingan, jika x[i] > x[j], maka isi x[i] dan x[j] ditukarkan
4. Bila i < (n – 1), ulangi mulai langkah 2.

Catatan: i = indeks array

x = nama array untuk menyimpan data

n = jumlah data

Algoritma diatas berlaku untuk pengurutan menaik *(ascending)*. Untuk pengurutan menurun *(descending)*, penukaran dilakukan jika x[i] < x[j].

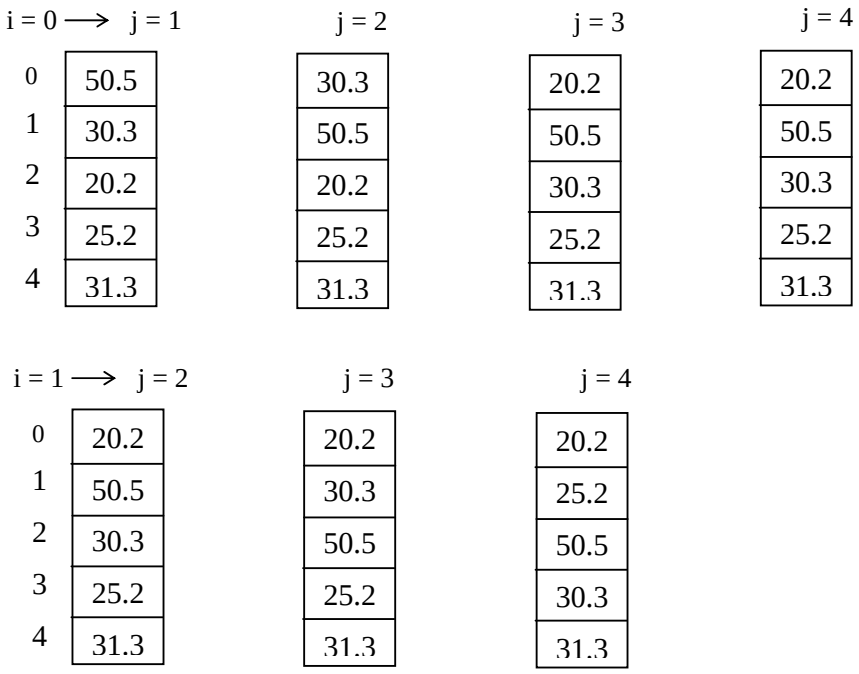
Penjelasan proses pengurutan terhadap 5 buah data ditunjukkan pada gambar 10.3

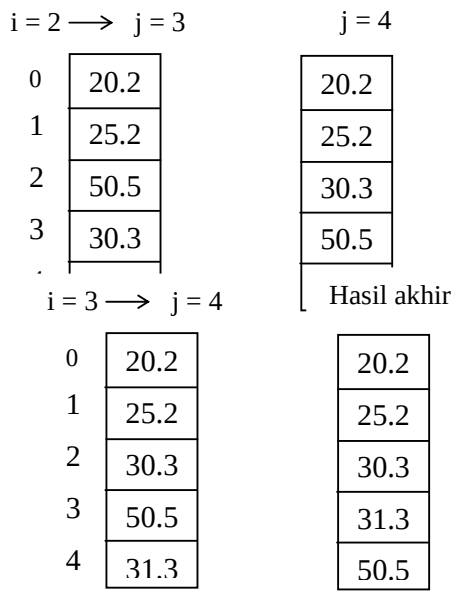
Data semula (sebelum pengurutan) adalah

50,5 30,3 20,2 25,2 31,3

Setelah pengurutan menaik (secara *ascending*), hasil yang diharapkan berupa

20,2 25,2 30,3 31,3 50,5






```
int i, j;
float smtr;

for(i=0; i<jumlah-1; i++)
     for(j=i+1; j<jumlah; j++)
          if(x[i] > x[j])          /* penukaran data */
          {
               smtr = x[i];
               x[i] = x[j];
               x[j] = smtr;
          }
```

C. TUGAS PENDAHULUAN

1. Buatlah program untuk mencari nilai rata-rata seorang mahasiswa

```
Tampilan:
Masukkan nilai matematika:60
Masukkan nilai Fisika:70
Masukkan nilai Kimia:80
Nilai rata-ratanya adalah 70
```

D. PERCOBAAN

1. Buatlah program yang membaca sebuah array karakter,'a'…'z'. Kemudian menghitung frekuensi kemunculan tiap karakter.

```
Tampilan:
Masukkan jumlah karakter yang akan dihitung: 5
Masukkan karakter ke-1:a
Masukkan karakter ke-2:b
Masukkan karakter ke-3:c
Masukkan karakter ke-3:a
Masukkan karakter ke-3:a

Frekuensi a=3
Frekuensi b=1
Frekuensi c=1
```

2. Buat matriks 2 dimensi dengan menggunakan inisialisasi langsung, kemudian lakukan penjumlahan .

3. Buat program untuk mencari nilai rata-rata n mahasiswa.

Input : Jumlah mahasiswa : 3
Nama Mahasiswa-1 : Ani
Jumlah nilai : 3
Nilai-1 : 60
Nilai-2 : 70
Nilai-3 : 80

Nama Mahasiswa-2 : Amir
Jumlah nilai : 3
Nilai-1 : 60
Nilai-2 : 50
Nilai-3 : 40
Nama Mahasiswa-3 : Ali
Jumlah nilai : 3
Nilai-1 : 50
Nilai-2 : 60
Nilai-3 : 70

Output :

Nilai rata-rata Ani adalah 70.
Nilai rata-rata Amir adalah 50.
Nilai rata-rata Ali adalah 50.

4. Buat program untuk menginputkan 10 bilangan dari keyboard. Kemudian urutkan bilangan tersebut mulai dari yang paling besar.

D. LAPORAN RESMI

1. Buatlah desain flowchart untuk setiap soal dalam percobaan