Praktikum 8 (2/3)
ARRAY

A. TUJUAN

1. Menjelaskan tentang array berdimensi dua
2. Menjelaskan tentang array berdimensi banyak
3. Menjelaskan tentang inisialisasi array tak berukuran.

B. DASAR TEORI

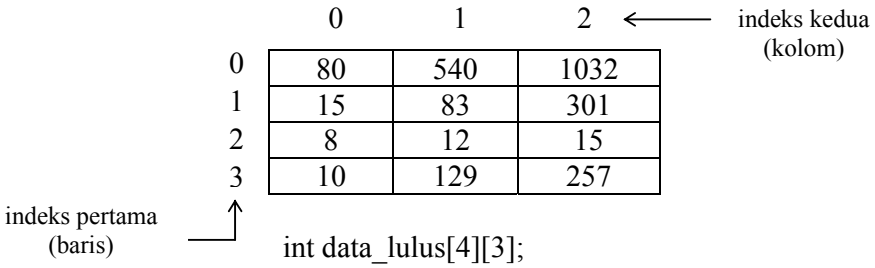
Array Berdimensi Dua

Array berdimensi satu dapat disimpan pada sebuah array berdimensi dua.

Pendeklarasian array berdimensi dua adalah sebagai berikut :

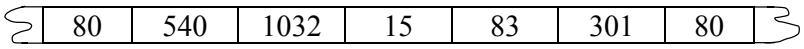
```
int data_lulus[4][3];
```

Nilai 4 untuk menyatakan banyaknya baris dan 3 menyatakan banyaknya kolom. Gambar 6.1 memberikan ilustrasi untuk memudahkan pemahaman tentang array berdimensi dua.



Gambar 6.1 Array berdimensi dua

Sama halnya pada array berdimensi satu, data array aka ditempatkan pada memori yang berurutan. Perhatikan Gambar 6.2.



Gambar 6.2 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 }
     }
};
```

C. TUGAS PENDAHULUAN

Buatlah desain flowchart untuk setiap soal dalam percobaan

D. PERCOBAAN

1. Deklarasikan sebuah variabel array (misalkan nama variabelnya = **letters[3][3]**) yang mengalokasikan untuk 9 elemen bertipe *char*. Kemudian masukkan karakter ‘Z’ pada elemen yang nilai baris= kolom dari array **letters.**

2. Ulangi soal no 1 dengan menggunakan loop for dan if statement.

3. Buat penjumlahan dua matrik A[2][2]dengan B[2][2]. . Hasil penjumlahan tersebut disimpan dalam matrik C.

4. 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.

5. Buat program untuk mencocokan apakah karakter sebuah karakter yang diinputkan dari keyboard ada dalam array yang telah dideklarasikan dan hitung berapa banyak karakter tersebut.

E. LAPORAN RESMI

1. Tulis listing program dari percobaan-percobaan diatas. Kemudian tulis outputnya.

2. Tuliskan kesimpulan dari percobaan yang telah anda lakukan.