

9.2. Array Berdimensi Dua

Pemanfaatan array tidak hanya dapat digunakan untuk menyimpan data dalam bentuk satu dimensi tetapi juga dapat digunakan untuk menyimpan data dalam bentuk 2 dimensi. Misal ada data dalam bentuk representasi sebagai berikut :

Nomhs	Tugas 2	Tugas 3
2001	80	80
2002	75	80
2003	90	75
2004	65	60

Deklarasi Array

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

Tipe_data nama_variabel[subscript_baris][subscript_kolom]

Pemberian nomor array dimulai dari baris 0 dan kolom 0, sehingga dalam contoh tabel di atas dapat dideklarasikan :

`int data[4][3]`

pada pendefinisian diatas :

4 menyatakan jumlah baris

3 menyatakan jumlah kolom

Inisialisasi variabel array

Inisialisasi atau memberikan nilai pada array hampir sama dengan variabel tunggal, hanya dalam memberikan nilai ini diperjelas dengan memberi nomor array.

`x[0][0] = -45;`

`x[0][1] = 6;`

`x[1][0] = 0;`

`x[1][1] = 72;`

`x[2][0] = 4;`

`x[2][1] = 34;`

	0	1
0	-46	6
1	0	73
2	4	34

Cara lain untuk memberi nilai atau inisialisasi dapat dilakukan langsung dalam proses pendeklarasian variabel array

```
int x[3][2] = { {-45, 6},
               { 0, 72 },
               { 4, 34 }, };
```

Menampilkan isi variabel array

Cara menampilkan isi array dilakukan dengan menyebutkan nama array dan nomor array yang ditampilkan. Contoh

```
cout<<"isi array 4 "<<x[1][1];
```

Contoh

```
#include <iostream.h>
main()
{   int data[4][3];
    data[0][0]= 2001;
    data[0][1]= 80;
    data[0][2]= 80;
    data[1][0]= 2002;
    data[1][1]= 75;
    data[1][2]= 80;
    data[2][0]= 2003;
    data[2][1]= 90;
    data[2][2]= 75;
    data[3][0]= 2004;
    data[3][1]= 65;
    data[3][2]= 60;
}
```

Penjelasan:

- int data[4][3];
⇒ memesan deklarasi variabel array 2 dimensi sebanyak 4 baris dan 3 kolom
- mengisi variabel array 2 dimensi
data[2][0]= 2003; → mengisi data baris 2 kolom 0 dengan 2003
data[2][1]= 90; → mengisi data baris 2 kolom 1 dengan 90
data[2][2]= 75; → mengisi data baris 2 kolom 2 dengan 75

Contoh

```
#include <iostream.h>
main()
{   int x[2][2]= {
                                { 1,2},
                                {3,4},
                                };
    int huruf[8][8] =
    {
        {1,2,3,4,5,6,7,8},
        {1,1,0,0,0,1,0,0},
        {1,1,0,0,0,1,0,0},
        {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},
        {1,1,1,1,1,1,1,0},
    };
}
```

Penjelasan :

- `int x[2][2]= { { 1,2},
 {3,4},
 };`
 ⇒ mendeklarasikan variabel x sebanyak 2 baris dan 2 kolom serta memberikan inisialisasi variabel array tersebut
- `int huruf[8][8]` → mendeklarasikan array 2 dimensi dan memberi nilai awal

contoh

```
#include <iostream.h>
main()
{ int data[4][3],i,j;
  data[0][0]= 2001;
  data[0][1]= 80;
  data[0][2]= 80;
  data[1][0]= 2002;
  data[1][1]= 75;
  data[1][2]= 80;
  data[2][0]= 2003;
  data[2][1]= 90;
  data[2][2]= 75;
  data[3][0]= 2004;
  data[3][1]= 65;
  data[3][2]= 60;

  for(i=0;i<=3;i++)
  {
    for(j=0;j<=2;j++)
      cout<<data[i][j]<<" | ";
    cout<<endl;
  }
}
```

Penjelasan:

- `int data[4][3];`
 ⇒ memesan deklarasi variabel array 2 dimensi sebanyak 4 baris dan 3 kolom
- mengisi variabel array 2 dimensi
 `data[2][0]= 2003;` → mengisi data baris 2 kolom 0 dengan 2003
 `data[2][1]= 90;` → mengisi data baris 2 kolom 1 dengan 90
 `data[2][2]= 75;` → mengisi data baris 2 kolom 2 dengan 75
- `for(i=0;i<=3;i++)`
 { `for(j=0;j<=2;j++)`
 `cout<<data[i][j]<<" | ";` → menampilkan isi array 2 dimensi
 `cout<<endl;`
 }

Contoh

```
#include <iostream.h>
main()
{
    int i,j;
    int x[2][2]= {
                { 1,2},
                { 3,4},
            };

    for (i=0;i<=1;i++)
    {
        for (j=0;j<=1;j++)
            cout<<x[i][j];
        cout<<endl;
    }
}
```

**Penjelasan:**

```
for (i=0;i<=1;i++)
{
    for (j=0;j<=1;j++)
        cout<<x[i][j];
    cout<<endl;
}
```

⇒ untuk menampilkan data variabel 2 dimensi harus menggunakan for dalam for

Contoh :**Penjumlahan 2 matrik****Logika**

$$\begin{Bmatrix} 2 & 3 & 4 \\ 4 & 2 & 1 \end{Bmatrix} + \begin{Bmatrix} 3 & 1 & 3 \\ 2 & 2 & 3 \end{Bmatrix}$$

Dalam menjumlah suatu matrik harus ada syarat yang harus diperhatikan, 2 matrik dapat dilakukan operasi penjumlahan bila ke dua matrik tersebut mempunyai baris dan kolom yang sama. Untuk memepromudah proses komputasinya diagram alir yang dibuat **harus menggunakan variabel larik**. Penggunaan variabel larik ini adalah variabel laarik yang menggunakan 2 dimensi.

$$\begin{matrix} A & A(1,1) = 2 & A(1,2) = 3 & A(1,3) = 4 \\ & A(2,1) = 4 & A(2,2) = 2 & A(2,3) = 1 \end{matrix}$$

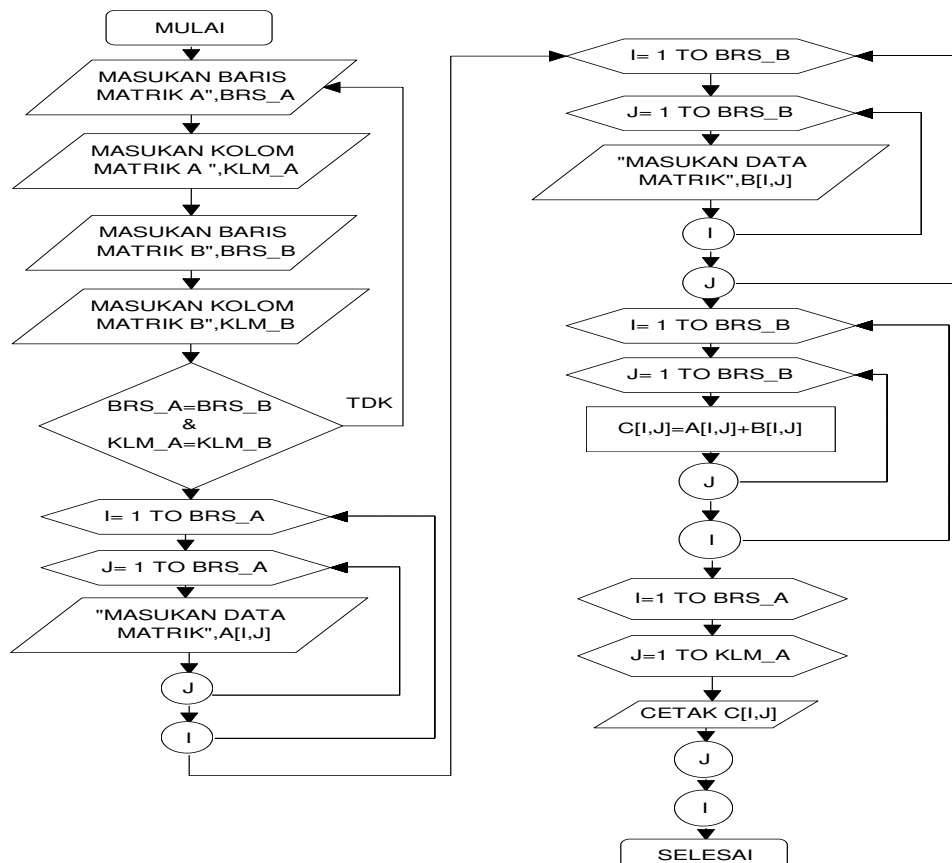
$$\begin{matrix} B & B(1,1) = 3 & B(1,2) = 1 & B(1,3) = 3 \\ & B(2,1) = 2 & B(2,2) = 2 & B(2,3) = 3 \end{matrix}$$

Proses penjumlahannya

$$\begin{array}{rclcl}
 & A(1,1) + B(1,1) & A(1,2)+B(1,2) & A(1,3)+B(1,3) \\
 A + B & A(2,1) + B(2,1) & A(2,2)+B(2,2) & A(2,3)+B(2,3)
 \end{array}$$

Algoritma :

1. Masukkan ukuran matriks A
 Ukuram baris matrik A $\rightarrow$ BRS_A
 Ukuran kolom matrik A $\rightarrow$ KLM_A
2. Masukkan ukuran matriks B
 Ukuram baris matrik B $\rightarrow$ BRS_B
 Ukuran kolom matrik B $\rightarrow$ KLM_B
3. Lakukan pengecekan apakah kedua ukuran matriknya sama
 Apakah BRS_A = BRS_B dan apakah KLM_A = KLM_B
 Jika tidak sama ulangan langkah 1
3. Masukkan isi data dari matrik A
4. Masukkan isi data dari matrik B
5. Ulang I dari 1 to BRS_A
 Ulang J dari 1 to KLM_A
 $C(I,J)=A(I,J)+B(I,J)$
6. Cetak matriks C
7. Selesai

Flowchart

Berikut adalah program sederhana untuk menghitung 2 matriks, kembangkan program di bawa ini sesuai flowchart di atas.

```
#include <iostream.h>
main()
{
    int a[3][3]={ { 2,3,5}, { 4,2,4},{ 5,3,5} };
    int b[3][3]={ { 1,2,7}, { 2,2,6},{ 4,3,7} };
    int c[3][3],i,j;

    for(i=0;i<=2;i++)
        for(j=0;j<=2;j++)
            c[i][j] = a[i][j] + b[i][j];

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

Penjelasan

```
for(i=0;i<=2;i++)
    for(j=0;j<=2;j++)
        c[i][j] = a[i][j] + b[i][j];
    ➔ Digunakan untuk proses penjumlahan matriks c=a+b

for(i=0;i<=2;i++)
{ cout<<endl;           ➔ untuk berganti baris
  for(j=0;j<=2;j++)
      cout<<c[i][j]<<" "; ➔ mencetak isi matriks, tanda <<" " untuk memberi jarak
                           dalam mencetak isi matriks
}
```

Jika `cout<<endl;` dihilangkan, maka hasilnya akan berderet untuk semua data matriks

Jika `cout<<c[i][j]<<" ";` perintah `<<" "` dihilangkan, maka hasilnya akan berderet untuk semua data matriks