

BAB I. LARIK (ARRAY)

Array (larik) ialah penampung sejumlah data sejenis (homogen) yang menggunakan satu *identifier* (pengenal).

Masing-masing elemen larik diakses menggunakan indeks (*subscript*) dari nol sampai n-1 (n menyatakan jumlah elemen larik).

Pengolahan data larik harus per elemen. Elemen larik dapat diakses langsung (acak), maksudnya untuk memanipulasi elemen keempat tidak harus melalui elemen kesatu, kedua dan ketiga.

Berdasarkan banyaknya indeks larik dibagi menjadi satu dimensi dan multi dimensi (dua dimensi, tiga dimensi).

1.1 Larik Satu Dimensi

Bentuk umum larik satu dimensi dideklarasikan dengan

```
tipe_data nama_larik[ukuran];
```

tipe_data menyatakan jenis elemen larik (int, float, char, unsigned, dan lain-lain), tidak boleh jenis void.

nama_larik adalah nama larik, harus memenuhi ketentuan pengenal.

ukuran menyatakan jumlah maksimal elemen larik, normalnya lebih besar dari satu.

Contoh:

```
int nilai[4];
```

?	?	?	?
nilai[0]	nilai[1]	nilai[2]	nilai[3]

untuk memberi nilai ke elemen larik dengan cara

- memberikan nilai langsung (*assignment*)
`nilai[2] ← 5 (nilai[2]=5);`, artinya kita memberikan nilai 5 ke elemen larik yang berindeks 2;
- memasukkan nilai melalui papan ketik (*keyboard*)
`cin >> nilai[2];` atau `scanf("%d", &nilai[2]);`

untuk mengakses (membaca) elemen larik dengan cara akses berikut.

```
nama_larik[indeks];  
contoh: nilai[2];  
atau  
cout << nilai[2]; atau printf("%d", nilai[2]);
```

Elemen larik dapat juga langsung diberi nilai awal waktu larik dideklarasikan. Dalam hal ini ukuran larik boleh dituliskan atau dikosongkan.

```
tipe_data nama_larik[] = {konstanta_1, konstanta_2, ..., konstanta_n};
```

konstanta_1, konstanta_2, ..., konstanta_n adalah nilai awal elemen larik dan harus setipe.

Contoh deklarasi larik

```
char huruf[] = {'a', 'b', 'c'};
```

'a'	'b'	'c'
huruf[0]	huruf[1]	huruf[2]

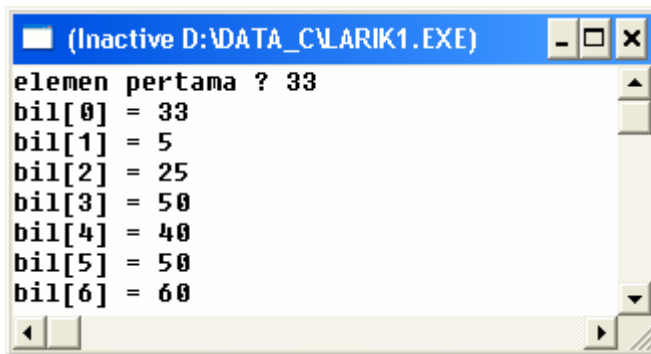
maksudnya huruf[0] = 'a'; huruf[1]='b'; huruf[2]='c';

Contoh Program Larik:

```
#include <stdio.h>
#include <conio.h>

main()
{ int bil[7], i;
  clrscr();
  printf("elemen pertama ? "); scanf("%d", &bil[0]);
  bil[1] = 5;
  bil[2] = bil[1] + 20;
  for (i = 4; i < 7; i++) bil[i] = i * 10;
  bil[3] = bil[bil[1]];
  for (i = 0; i < 7; i++)
  printf("bil[%d] = %d \n", i, bil[i]);
}
```

Hasil program



1.1.1 Menghitung Jumlah Elemen Array

Karena fungsi *sizeof()* mengembalikan jumlah byte yang sesuai dengan argumennya, maka operator tersebut dapat digunakan untuk menemukan jumlah elemen array, misalnya:

```
int array[ ] = {26,7,82,166};
cout<<sizeof(array)/sizeof(int);
```

akan mengembalikan nilai 4, yaitu sama dengan jumlah elemen yang dimiliki array.

1.1.2 Melewatkan Array Sebagai Argumen Fungsi

Array dapat dikirim dan dikembalikan oleh fungsi pada saat array dikirim ke dalam fungsi, nilai aktualnya dapat dimanipulasi.

Contoh :

```
#include <iostream.h>
void ubah(int x[]);
void main()
{
    int ujian[] = {90,95,78,85};
    ubah(ujian);
    cout<<" Elemen kedua dari array ujian adalah "<<ujian[1]<<endl;
}

void ubah(int x[])
{
    x[1] = 100;
}
```

Keluarannya : Elemen kedua dari array ujian adalah 100

1.2 Larik Dua Dimensi (Matriks)

Matriks adalah sekumpulan informasi yang setiap individu elemennya diacu dengan menggunakan dua buah indeks (yang biasanya dikonotasikan dengan baris dan kolom).

Konsep umum larik berlaku juga pada Matriks, yaitu:

1. kumpulan elemen bertipe sama;
2. setiap elemen data dapat diakses secara acak, jika indeksnya (baris dan kolom) sudah diketahui;
3. merupakan struktur data statis, artinya jumlah elemennya sudah ditentukan terlebih dahulu di dalam kamus dan tidak bisa diubah selama pelaksanaan program.

Deklarasi Matriks

```
tipe_data nama_matriks[baris] [kolom];
```

Contoh:

```
int B[3][4];
```

	0	1	2	3
0	B[0][0]	B[0][1]	B[0][2]	B[0][3]
1	B[1][0]	B[1][1]	B[1][2]	B[1][3]
2	B[2][0]	B[2][1]	B[2][2]	B[2][3]

tipe_data menyatakan jenis elemen matriks (int, float, char, unsigned, dan lain-lain), tidak boleh jenis void.

nama_matriks adalah nama matriks, harus memenuhi ketentuan pengenalan.

baris menyatakan jumlah maksimal elemen baris matriks,

kolom menyatakan jumlah maksimal elemen kolom matriks.

Untuk memberi nilai matriks:

1. memberikan nilai langsung (*assignment*)
 $B[2][1] \leftarrow 5$ ($B[2][1]=5$); artinya kita memberikan nilai 5 ke matriks pada baris 2 dan kolom 1;
2. memasukkan nilai melalui papan ketik (*keyboard*)
 $cin \gg B[2][1]$; atau $scanf("%d", \&B[2][1])$;

untuk mengakses (membaca) elemen matriks dengan cara akses berikut.

$nama_larik[baris][kolom]$; contoh $B[2][1]$;
 atau
 $cout \ll B[2][1]$; atau $printf("%d", B[2][1])$;

Contoh program memberi (mengisi) nilai suatu matriks

```
#include <stdio.h>
#include <conio.h>

main()
{ int B[3][4],i,j;
  clrscr();
  for (i=0;i<=2;i++)
  {
    for (j=0;j<=3;j++)
    {
      printf("B[%d,%d] = ",i+1,j+1); scanf("%d", &B[i][j]);
    }
  }
  printf("\n");
  for (i=0; i<=2;i++)
  {
    for (j=0; j<=3;j++)
    printf("%d ",B[i][j]);
    if (j=3) printf("\n");
  }
  return 0;
}
```

1.2.1 Inisialisasi Matriks

Misalkan ada matriks sebagai berikut:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Dapat dinyatakan dengan perintah berikut ini.

```
int mat[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

1.2.2 Menjumlahkan Dua Buah Matriks

Menjumlahkan dua buah matriks A dan B menghasilkan matriks C atau $A + B = C$ hanya dapat dilakukan bila ukuran matriks A dan ukuran matriks B sama, dan kedua matriks sudah terdefinisi harganya (elemennya sudah terisi). Matriks C juga berukuran sama dengan matriks A dan B. Penjumlahan matriks A dan B didefinisikan sebagai berikut:

$C[i,j] = A[i,j] + B[i,j]$ untuk setiap i dan j.

Contoh:

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 2 & 1 \\ 2 & 3 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 2 & 4 \\ 1 & 4 & 3 \\ 2 & 5 & 2 \end{bmatrix}$$

Prosedur untuk menjumlahkan matriks sebagai berikut.

Kamus

```
const min_baris : integer = 1
```

```
const maks_baris : integer = 3
```

```
const min_kolom : integer = 1
```

```
const maks_kolom : integer = 3
```

```
type matriks : matrix[min_baris..maks_baris, min_kolom..maks_kolom] of integer
```

atau

Kamus

```
const int maks_baris = 3
```

```
const int maks_kolom = 3
```

```
int matriks [maks_baris] [maks_kolom]
```

```

procedure Jumlah_matriks(input A : matriks, input B : matriks, output C : matriks)
{menjumlahkan matriks A dan B, yaitu  $A + B = C$ 
K.Awal : matriks A dan B sudah terdefinisi elemen-elemennya
K.Akhir : elemen matriks C berisi penjumlahan A dan B}

```

Kamus

i : integer {indeks baris}
j : integer {indeks kolom}

Algoritma

```

for i ← min_baris to mak_baris do
  for j ← min_kolom to mak_kolom do
    C[i,j] ← A[i,j] + B[i,j]
  endfor
endfor

```

```

void Jumlah_matriks(input A : matriks, input B : matriks, output C : matriks)
{menjumlahkan matriks A dan B, yaitu  $A + B = C$ 
K.Awal : matriks A dan B sudah terdefinisi elemen-elemennya
K.Akhir : elemen matriks C berisi penjumlahan A dan B}

```

Kamus

int i {indeks baris}
int j {indeks kolom}

Algoritma

```

for (i ← min_baris; i ≤ mak_baris; i++)
  for (j ← min_kolom; j ≤ mak_kolom; j++)
    C[i,j] ← A[i,j] + B[i,j]
  endfor
endfor

```

Latihan:

Buatlah algoritma dan program untuk menentukan:

1. jumlah dan rata-rata sejumlah data menggunakan larik.
2. penjumlahan dan pengurangan dua buah matriks.
3. perkalian dari dua buah matriks.