



MATRIKS (LARIK 2 DIMENSI)



Institut Teknologi Sumatera

OUTLINE

- Konsep matriks
- Inisialisasi matriks
- Mengakses elemen matriks
- Pencarian nilai di dalam matriks

TUJUAN PERKULIAHAN

- Mahasiswa memahami konsep dan cara kerja matriks
- Mahasiswa mampu membuat program sederhana menggunakan matriks

REVIEW LARIK 1 DIMENSI

- Bayangkan jika anda diminta untuk membuat variabel yang menampung integer berupa nilai tugas dari seorang mahasiswa. Masing-masing nilai dari mahasiswa ditampung dalam sebuah variabel yang sama dengan nama mahasiswa dan banyaknya tugas adalah 5 buah.
- Dengan apa anda membuatnya ?

REVIEW LARIK 1 DIMENSI

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int Doni[5], Susilo[5], Andika[5], Rahmat[5];
```

```
}
```

Lalu bagaimana jika masing-masing siswa mempunyai 3 mata pelajaran berbeda, dan masing-masing mata pelajaran punya 5 nilai tugas. Bisakah ditampung dalam sebuah variabel `int Nama_Siswa;` saja ?

LARIK 2 DIMENSI (MATRIKS)

- Maka anda memerlukan sebuah larik (array) yang memiliki 2 dimensi.

Doni[0][0]	Doni[0][1]	Doni[0][2]	Doni[0][3]	Doni[0][4]
Doni[1][0]	Doni[1][1]	Doni[1][2]	Doni[1][3]	Doni[1][4]
Doni[2][0]	Doni[2][1]	Doni[2][2]	Doni[2][3]	Doni[2][4]

LARIK 2 DIMENSI (MATRIKS)

Doni[0][0]	Doni[0][1]	Doni[0][2]	Doni[0][3]	Doni[0][4]
Doni[1][0]	Doni[1][1]	Doni[1][2]	Doni[1][3]	Doni[1][4]
Doni[2][0]	Doni[2][1]	Doni[2][2]	Doni[2][3]	Doni[2][4]

Cara Inisiasi Variabel Larik 2 Dimensi :

```
tipe_data nama_var_array[batas_baris][batas_kolom];
```

LATIHAN PENGANTAR LARIK 2 DIMENSI

Prodi	2014	2015	2016	2017
1. Teknik Informatika	35	45	80	120
2. Teknik Industri	100	110	70	101
3. Teknik Fisika	10	15	20	17

Bentuk seperti pada tabel diatas dapat dituangkan pada array berdimensi dua.
Pendefinisiannya :

Pada pendefinisian diatas :

3 menyatakan jumlah baris (mewakili prodi)

4 menyatakan jumlah kolom (mewakili tahun kelulusan).

SOLUSI

Prodi	2014	2015	2016	2017
1. Teknik Informatika	35	45	80	120
2. Teknik Industri	100	110	70	101
3. Teknik Fisika	10	15	20	17

```
#include <iostream>

using namespace std;

int main() {

    int data_lulus[3][4]={35,45,80,120,100,110,70,101,10,15,20,17};

}
```

LATIHAN PENGANTAR LARIK 2 DIMENSI

Prodi	2014	2015	2016	2017
1. Teknik Informatika	35	45	80	120
2. Teknik Industri	100	110	70	101
3. Teknik Fisika	10	15	20	17

Bagaimana jika program ingin menampilkan data seperti dibawah ini :

Masukkan Kode Jurusan : 2

Tahun ke : 3

70

 Run Succeeded | Time 11 ms

SOLUSI

Prodi	2014	2015	2016	2017
1. Teknik Informatika	35	45	80	120
2. Teknik Industri	100	110	70	101
3. Teknik Fisika	10	15	20	17

```
#include <iostream>

using namespace std;

int main() {

    int data_lulus[3][4]={35,45,80,120,100,110,70,101,10,15,20,17};

    int jurusan,tahun;

    cout<<"Masukkan Kode Jurusan : ";cin>>jurusan;cout<<endl;

    cout<<"Tahun ke : ";cin>>tahun;cout<<endl;

    cout<<data_lulus[jurusan-1][tahun-1];

}
```

KEUNTUNGAN MENGGUNAKAN LARIK

- Array sangat cocok untuk pengaksesan acak. Sembarang elemen di array dapat diacu secara langsung tanpa melalui elemen-elemen lain.
- Jika berada di suatu lokasi elemen, maka sangat mudah menelusuri ke elemenelemen tetangga, baik elemen pendahulu atau elemen penerus
- Jika elemen-elemen array adalah nilai-nilai independen dan seluruhnya harus terjaga, maka penggunaan penyimpanannya sangat efisien

KEKURANGAN MENGGUNAKAN LARIK

- Array harus bertipe homogen. Kita tidak dapat mempunyai array dimana satu elemen adalah karakter, elemen lain bilangan, dan elemen lain adalah tipe-tipe lain
- Kebanyakan bahasa pemrograman mengimplementasikan array statik yang sulit diubah ukurannya di waktu eksekusi. Bila penambahan dan pengurangan terjadi terus-menerus, maka representasi statis
- Tidak efisien dalam penggunaan memori
- Menyiakan banyak waktu komputasi
- Pada suatu aplikasi, representasi statis tidak dimungkinkan

LATIHAN #1 LARIK 2 DIMENSI

Bagaimana mengisi larik 2 dimensi yang memiliki nilai berurutan, dimana pengguna memasukkan dimensi larik
Keluaran berupa bentuk matriks.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

```
Masukkan dimensi larik yang anda inginkan :  
4  
2  
  
1 2  
3 4  
5 6  
7 8
```

✓ Run Succeeded | Time 20 ms

```
#include <iostream>
```

```
using namespace std;
```

```
int main() {
```

```
    int r=1;
```

```
    int x,y,m=0,n=0;
```

```
    cout<<"Masukkan dimensi larik yang anda inginkan : "<<endl;
```

```
    cin>>x;
```

```
    cin>>y;
```

```
    cout<<endl;
```

```
    int a[x][y];
```

```
    for(m=0; m<x; m++){
```

```
        for(n=0; n<y; n++){
```

```
            a[m][n]=r;
```

```
            r++;
```

```
        }
```

```
    }
```

```
    for(m=0; m<x; m++){
```

```
        for(n=0; n<y; n++){
```

```
            cout<<a[m][n]<<" ";
```

```
        }
```

```
        cout<<endl;
```

```
    }
```

```
}
```

SOLUSI LATIHAN

#1 LARIK 2

DIMENSI

LATIHAN #2 LARIK 2 DIMENSI

Modifikasilah program yang sebelumnya sudah dibuat agar bisa melakukan operasi antara matriks A + B. Dimana dimensi matriks A dan B ditentukan oleh pengguna

Contoh apabila pengguna memasukkan 4 dan 2 sebagai dimensi matriks

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} + \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}$$

```
Masukkan dimensi larik yang anda inginkan :  
4  
2  
  
1 2  
3 4  
5 6  
7 8  
  
+  
1 2  
3 4  
5 6  
7 8  
HASIL :  
2 4  
6 8  
10 12  
14 16
```


LATIHAN #3 LARIK 2 DIMENSI

Modifikasilah program yang sebelumnya sudah dibuat agar bisa melakukan operasi antara matriks A + B. Dimana dimensi matriks A dan B ditentukan oleh pengguna, dan matriks B merupakan urutan nilai yang menurun

Contoh apabila pengguna memasukkan 4 dan 2 sebagai dimensi matriks

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix} + \begin{bmatrix} 8 & 7 \\ 6 & 5 \\ 4 & 3 \\ 2 & 1 \end{bmatrix}$$

```
Masukkan dimensi larik yang anda inginkan :  
4  
2  
  
1 2  
3 4  
5 6  
7 8  
  
+  
8 7  
6 5  
4 3  
2 1  
HASIL :  
9 9  
9 9  
9 9  
9 9
```

Terima Kasih