

STRUKDAT DATA

MATRIKS C++

Definisi Matriks

Sekumpulan informasi yang setiap individu elemennya terdefinisi berdasarkan dua buah indeks (yang biasanya dikonotasikan dengan baris dan kolom). Setiap elemen matriks dapat diakses secara langsung jika kedua indeks diketahui. Setiap elemen matriks mempunyai type yang homogen. Indeks baris dan kolom harus bertipe yang mempunyai keterurutan (suksesor/predesesor), misalnya integer.

Contoh Matriks

1. Matriks Nama Hari

- Merepresentasikan nama hari ke-1 sampai dengan 7 dalam 3 bahasa (Indonesia, Inggris, dan Prancis)
- Indeks baris = 1 sampai dengan 7 ; indeks kolom = 1 sampai dengan 3
- Elemen matriks yang bertipe string

Indeks	1 = Indonesia	2 = Inggris	3 = Prancis
1	Senin	Monday	Lundi
2	Selasa	Tuesday	Mardi
3	Rabu	Wednesday	Mercredi
4	Kamis	Thursday	Jeudi
5	Jumat	Friday	Vendredi
6	Sabtu	Saturday	Samedi
7	Minggu	Sunday	Dimanche

2. Matriks Satuan

- Matriks satuan merupakan matriks dengan elemen berupa angka atau numeric
- Indeks baris = 1 sampai dengan 4 ; indeks kolom = 1 sampai dengan 4
- Elemen matriks bertipe integer.

Indeks	1	2	3	4
1	1	6	9	2
2	5	8	1	7
3	4	5	3	3
4	2	6	0	9

Contoh Pemakaian Matriks

- Operasi biasa matriks dalam matematika :
 1. Penjumlahan
 2. Perkalian
 3. Menentukan determinan
 4. Transpose
- Sistem persamaan linier dan operational research
- Persoalan algoritmik yaitu untuk menyimpan informasi yang cirinya ditentukan oleh 2 dimensi (diterjemahkan dalam baris dan kolom). Contoh : cell pada sebuah spreadsheet, ruangan Gedung bertingkat.

Implementasi Matriks di C++

- Memori matriks diimplementasikan sebagai array 2 dimensi
- Suatu elemen matriks diakses dengan indeks baris dan kolom bertipe integer
- Elemen matriks di deklarasikan memiliki tipe yang sama (homogen)
- Cara deklarasi :
`<type-element><nama_matriks>[<baris>][<kolom>];`

Contoh

```
Int Matriks1 [5][5];
```

Penjelasan :

Matriks Bernama Matriks1 dengan setiap elemen bertipe integer, dengan banyaknya baris = 5 dan banyaknya kolom = 5; dengan alamat setiap diakses melalui indeks baris 0 sampai dengan 4 dan indeks kolom 0 sampai dengan 4.

- Cara akses elemen matriks :
`<nama_matriks>[<baris>][<kolom>];`

Contoh :

Int Matriks2 [4][4];

Indeks	0	1	2	3
0	1	6	9	2
1	5	8	1	7
2	4	5	3	3
3	2	6	0	9

Cout << Matriks2[1][3]; => maka akan tercetak : 7

Int a = Matriks2[2][1] * Matriks2[1][1]; => maka nilai a bernilai 32

Matriks2[2][2] = 10; elemen baris 2, kolom 2 menjadi 10

- Hanya Sebagian dari memori matriks yang akan benar-benar digunakan untuk menyimpan elemen matriks

Contoh : deklarasi matriks 10x10, namun yang secara actual dipakai hanya 4x3.

Sekali matriks
didefinisikan
dimungkinkan
berubah
ukuran baris
dan kolomnya

-Jun-13

	0	1	2	3	4	5	6	7	8	9
0	1	1	1							
1	2	2	2							
2	3	3	3							
3	4	4	4							
4										
5										
6										
7										
8										
9										

- Dibutuhkan :
 1. Banyaknya baris efektif (misal Nbaris) => banyaknya baris yang dipakai
 2. Banyaknya kolom efektif (missal Nkolom) => banyaknya kolom yang dipakai
- Elemen matriks yang boleh diakses hanya yang dipakai

Operasi-operasi pada Matriks

- Mendeklarasikan dan mendefinisikan isi matriks
- Menuliskan isi matriks ke layer
- Operasi 1 matriks :
 1. Menghitung total semua elemen matriks
 2. Mengalikan isi matriks dengan sebuah konstanta
 3. Transpose matriks
- Operasi 2 matriks :

1. Menambahkan dua matriks
2. Mengalikan 2 matriks

Contoh membuat matriks ordo 2x2

```
#include <iostream>

using namespace std;

int main(){

    int Matriks[2][2] = {1,2,3,4};

    cout << Matriks[0][0] << " " << Matriks[0][1] << endl;

    cout << Matriks[1][0] << " " << Matriks[1][1] << endl;

    return 0;

}
```

Menampilkan matriks dengan menggunakan pointers array

```
#include <iostream>

using namespace std;

void printMatriks(int *ptrArray, int baris, int kolom){

    int index = 0;

    for(int i = 0; i<baris; i++){

        for(int j = 0; j<kolom; j++ ){

            cout << *(ptrArray + index) << " ";

            index++;

        }

        cout << endl;

    }

}

int main(){

    int Matriks[2][2] = {1,2,3,4};

    printMatriks(*Matriks,2,2);

    return 0;

}
```

Membuat matriks ordo 2x2 dengan standart library array

```
#include <iostream>
#include <array>

using namespace std;

int main(){
    const int kolom = 3;
    const int baris = 2;
    array < array <int,kolom>, baris> Nmatriks =
{1,2,3,4,5,6};
    cout << Nmatriks[0][0] <<" ";
    cout << Nmatriks[0][1] <<" ";
    cout << Nmatriks[0][2] <<" "<<endl;

    cout << Nmatriks[1][0] <<" ";
    cout << Nmatriks[1][1] <<" ";
    cout << Nmatriks[1][2] <<" "<<endl;
    return 0;
}
```

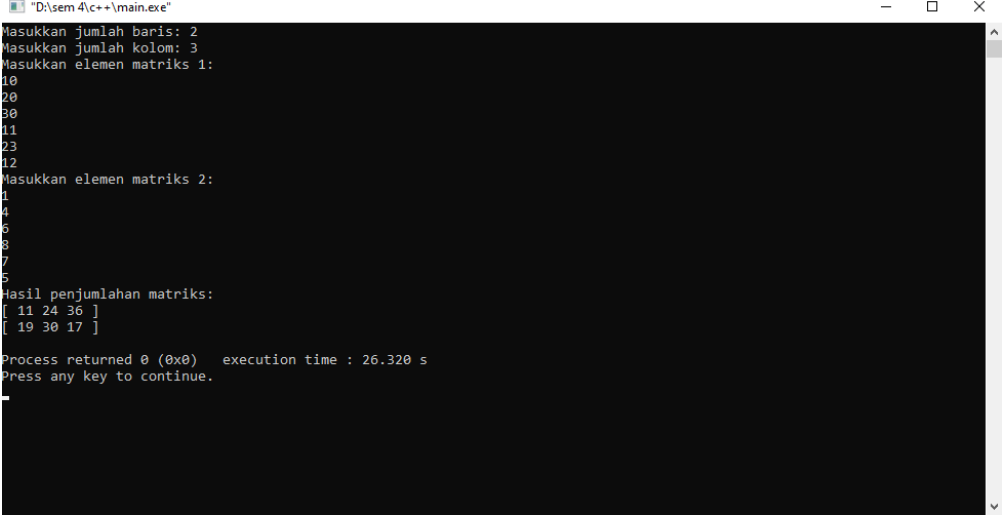
Menampilkan matriks dengan menggunakan pointers array di std library array

```
#include <iostream>
#include <array>
using namespace std;
const int kolom = 3;
const int baris = 2;
void printMatriks(array < array <int,kolom>, baris>
Nilaiarray){
    for (array<int,kolom> vectorbaris : Nilaiarray){
        for(int nilaikolom: vectorbaris){
            cout << nilaikolom <<" ";
        }
        cout << endl;
    }
}

int main(){
    array < array <int,kolom>, baris> Nmatriks =
{1,2,3,4,5,6};
    printMatriks(Nmatriks);
    return 0;
}
```

Tugas Praktikum

Buatlah program penjumlahan matriks tanpa menggunakan std library array dengan ordo 2x3, dimana baris dan kolom inputan dari user. (noted : elemen yang ada di dalam matriks harus sesuai dengan output yang disediakan)



```
"D:\sem 4\c++\main.exe"
Masukkan jumlah baris: 2
Masukkan jumlah kolom: 3
Masukkan elemen matriks 1:
10
20
30
11
23
12
Masukkan elemen matriks 2:
1
4
6
8
7
5
Hasil penjumlahan matriks:
[ 11 24 36 ]
[ 19 30 17 ]

Process returned 0 (0x0)   execution time : 26.320 s
Press any key to continue.
```