



Matriks

Tim Pengajar KU1072



KU1072/Pengenalan Teknologi Informasi B
Tahap Tahun Pertama Bersama
Institut Teknologi Bandung





Tujuan Perkuliahan

- Mahasiswa memahami matriks dan representasinya sebagai array of array
- Mahasiswa dapat menggunakan notasi pendefinisian dan pengacuan matriks dengan benar
- Mahasiswa dapat melakukan beberapa operasi dasar terkait matriks
- Mahasiswa dapat membuat program dengan menggunakan matriks

- 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 (sukesor/predesesor), misalnya integer.
- Matriks adalah struktur data dengan memori internal. Struktur ini praktis untuk dipakai tetapi memakan memori!
 - Matriks integer 100 x 100 memakan 10000 x tempat penyimpanan integer.

Memori Matriks (1/2)

- Matriks adalah struktur data statik (ukurannya ditentukan dari awal)
- Untuk keperluan ini, sering kali memori dipesan berlebihan untuk kemudian dipakai sebagian saja

	1	2	3	4	5	6	7	8	9	10
1	1	1	1							
2	2	2	2							
3	3	3	3							
4	4	4	4							
5										
6										
7										
8										
9										
10										

Contoh-1

- MatNamaHari
 - Merepresentasikan nama hari ke-1 s.d. 7 dalam 3 bahasa (Indonesia, Inggris, Prancis)
 - Indeks baris : 1 s.d. 7; indeks kolom : 1 s.d. 3
 - Elemen matriks ber-type string

	1 = INDONESIA	2 = INGGRIS	3 = FRANCIS
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

Contoh-2

- MatUkur
 - Indeks (i, j) merepresentasikan suatu titik koordinat
 - Elemen matriks merepresentasikan hasil pengukuran pada suatu titik koordinat tertentu
 - Indeks baris : 1 s.d. 5, indeks kolom : 1 s.d. 5
 - Elemen matriks ber-type real

	1	2	3	4	5
1	12.1	7.0	8.9	0.7	6.6
2	0.0	1.6	2.1	45.9	55.0
3	6.1	8.0	0.0	3.1	21.9
4	9.0	1.0	2.7	22.1	6.2
5	5.0	0.8	0.8	2.0	8.1

Contoh-3

- MatSat
 - Merupakan matriks satuan, yaitu matriks dengan elemen bernilai hanya 0 atau 1
 - Indeks baris : 1 s.d. 4; indeks kolom 1 s.d. 4
 - Elemen matriks ber-type integer

	1	2	3	4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

Contoh-4

- MatSurvey:
 - Matriks hasil survey pada titik kordinat.
 - Setiap elemen matriks merepresentasikan hasil pengukuran <temperatur, kecepatan angin> pada titik koordinat (i, j)

	1	2	3	4
1	<24,5>	<24,5>	<30,5>	<25,5>
2	<23,56>	<3,6>	<40,5>	<2,2>
3	<22,73>	<7,3>	<60,6>	<8,3>
4	<21,56>	<8,5>	<9,8>	<7,4>
5	<23,56>	<12,50>	<3,36>	<30,6>
6	<20,0>	<2,56>	<5,46>	<20,99>
7	<30,0>	<9,0>	<15,0>	<27,0>

Contoh Pemakaian

- Operasi “biasa” matriks dalam matematika : penjumlahan, perkalian, menentukan determinan, menginvers sebuah matriks, transpose, dll.
 - Semua "perhitungan" itu menjadi tidak primitif, harus diprogram
- Sistem persamaan linier dan *operational research*
- Persoalan algoritmik: 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++ (1)

- Memori matriks diimplementasikan sebagai **array 2 dimensi**
- Suatu elemen matriks diakses dengan indeks baris dan kolom bertipe integer
- Elemen matriks dideklarasikan memiliki type yang sama (homogen)
- Cara deklarasi:

```
<type-elmt> <namamatriks>[<#baris>][<#kolom>;
```



Implementasi Matriks di C++ (2)

- Contoh:

```
int MatSatuan [10][10];
```

Matriks bernama `MatSatuan` dengan setiap elemen bertipe `integer`, dengan `banyaknya baris = 10` dan `banyaknya kolom = 10`; dengan alamat setiap elemen diakses melalui `indeks baris 0 s.d. 9` dan `indeks kolom 0 s.d. 9`.

Implementasi Matriks di C++ (3)

- Cara akses elemen matriks:
`<namamatriks>[<brs>][<kol>]`
- Contoh: `int M1[5][5];`

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10
2	11	12	13	14	15
3	16	17	18	19	20
4	21	22	23	24	25

`cout << M1[4][0]; // akan tercetak: 21`

`int x = M1[0][0] * M1[3][3]; // x bernilai 19`

`M1[1][1] = 8; // Elemen brs. 1, kol. 1 menjadi 8`

`M1[0][5] ?? // tidak terdefinisi!`

Implementasi Matriks di C++ (4)

- Hanya sebagian dari memori matriks yang akan benar-benar digunakan untuk menyimpan elemen matriks
- Contoh: deklarasi matriks 10x10, namun yang secara aktual dipakai hanya 4x3

Sekali matriks didefinisikan dimungkinkan berubah ukuran baris dan kolomnya

	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										

Implementasi Matriks di C++ (5)

- Dibutuhkan:
 - **Banyaknya baris efektif** (misal **NBrS**) : banyaknya baris yang dipakai
 - **Banyaknya kolom efektif** (misal **NKol**) : banyaknya kolom yang dipakai
- Elemen matriks yang boleh diakses hanya yang dipakai

	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										

NKol = 3

NBrS = 4



Operasi-Operasi pada Matriks

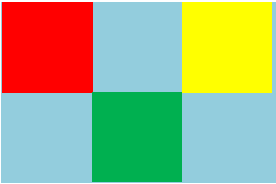
- Mendeklarasikan dan mendefinisikan isi matriks
- Menuliskan isi matriks ke layar
- Operasi 1 matriks:
 - Menghitung total semua elemen matriks
 - Mengalikan isi matriks dengan sebuah konstanta
 - Transpose matriks
- Operasi 2 matriks:
 - Menambahkan dua matriks
 - Mengalikan dua matriks



Mengisi dan Menuliskan Isi Sebuah Matriks

- Deklarasi matriks dan inisialisasi baris+kolom efektif
- Membaca isi matriks dari hasil kalkulasi
- Menampilkan ke layar

```
//Mendefinisikan isi sebuah matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M[10][10]; //Deklarasi matriks
    int NBrS, NKol; //Baris dan kolom efektif
    int i, j;
    // ALGORITMA
    // Mengisi matriks
    NBrS = 5; NKol = 5;
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            M[i][j] = i*j;
        }
    }
    // Menuliskan isi matriks ke layar
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            cout << M[i][j] << " ";
        }
        cout << endl; //ganti baris
    }
    return 0;
}
```

Membaca dan Menuliskan Isi Sebuah Matriks

- Deklarasi matriks
- Inisialisasi baris+kolom efektif dari masukan user
- Membaca isi matriks dari user
- Menampilkan isi matriks ke layar

```
//Mendefinisikan isi sebuah matriks
//dari masukan pengguna
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M[10][10];
    int NBrS, NKol;
    int i, j;
    // ALGORITMA
    cout << "Baris efektif = "; cin >> NBrS;
    cout << "Kolom efektif = "; cin >> NKol;
    //asumsi NBrS dan NKol selalu benar
    //Membaca isi matriks
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            cout << "Elemen ke-[" << i << ", " << j
            << "] = ";
            cin >> M[i][j];
        }
    }
    // Menuliskan isi matriks ke layar
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            cout << M[i][j] << " ";
        }
        cout << endl; //ganti baris
    }
    return 0;
}
```

Menghitung Total Elemen

- Men-total seluruh elemen yang ada di matriks dan menampilkan hasilnya ke layar

```
//Mendefinisikan isi sebuah matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M[10][10]; //Deklarasi matriks
    int NBrS, NKol; //Baris dan kolom efektif
    int i, j, sum;
    // ALGORITMA
    // Mengisi matriks
    // M diasumsikan sudah diisi

    // Mnghitung sum elemen
    sum = 0;
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            sum = sum + M[i][j];
        }
    }
    // Menuliskan hasil sum ke layar
    cout << "Total semua elemen = " << sum;
    return 0;
}
```



Mengalikan Elemen Matriks dgn Konstanta

- Setiap elemen matriks dikalikan dengan suatu konstanta, misal X

12-Jun-13

```
// Mengalikan isi matriks dengan sebuah
// konstanta
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M[10][10];
    int NBrS, NKol;
    int i, j, x;

    // ALGORITMA
    // Membaca isi matriks : buat sebagai
    // latihan

    // Mengalikan elemen matriks dengan x
    cout << "Faktor pengali = "; cin >> x;
    for (i=0; i<NBrS; i++) {
        for (j=0; j<NKol; j++) {
            M[i][j] = M[i][j] * x;
        }
    }

    // Menuliskan matriks baru ke layar : buat
    // sebagai latihan

    return 0;
}
```

Transpose Matriks (1)

- Dideklarasikan 2 buah matriks, misalnya M dan MTranspose
- MTranspose menampung hasil transpose dari M
 - Ukuran baris efektif M = ukuran kolom efektif MTranspose
 - Ukuran kolom efektif M = ukuran baris efektif Mtranspose
- $Mtranspose[i][j] = M[j][i]$

Transpose Matriks (2)



```
//Transpose Matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M[10][10]; int NBrS1, NKol1;
    int MTranspose[10][10]; int NBrS2, NKol2;
    int i, j;
    // ALGORITMA
    // Membaca isi matriks M : Buat sebagai latihan

    // Deklarasi dan pengisian MTranspose
    NBrS2 = NKol1; NKol2 = NBrS1;
    for (i=0; i<NBrS2; i++) {
        for (j=0; j<NKol2; j++) {
            MTranspose[i][j] = M[j][i]; //transpose
        }
    }

    // Menuliskan hasil transpose ke layar : Buat sebagai latihan
    return 0;
}
```

Menjumlahkan 2 Matriks (1)

- Deklarasi 3 matriks, misal M1, M2, dan MHasil
- 2 matriks M1 dan M2 bisa dijumlahkan jika dimensinya sama, yaitu:
 - Ukuran baris efektif M1 = ukuran baris efektif M2
 - Ukuran kolom efektif M1 = ukuran kolom efektif M2
- Hasil ditampung di MHasil:
$$MHasil[i][j] = M1[i][j] + M2[i][j]$$

Menjumlahkan 2 Matriks (2)



```
//Menjumlahkan isi 2 matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M1[10][10]; int NBrs1, NKol1;
    int M2[10][10]; int NBrs2, NKol2;
    int MHasil[10][10]; int NBrsHsl, NKolHsl;
    int i, j;
    // ALGORITMA
    //Membaca isi matriks M1 dan M2 : Buat sebagai latihan

    // Deklarasi dan pengisian MHasil
    NBrsHsl = NBrs1; NKolHsl = NKol1;
    for (i=0; i<NBrsHsl; i++) {
        for (j=0; j<NKolHsl; j++) {
            MHasil[i][j] = M1[i][j] + M2[i][j]; //penjumlahan
        }
    }

    // Menuliskan hasil penjumlahan ke layar : Buat sebagai latihan
    return 0;
}
```

Mengalikan 2 Matriks (1)

- Deklarasi 3 matriks, misal $M1$, $M2$, dan $MHasil$
- 2 matriks $M1$ dan $M2$ bisa dikalikan jika:
 - Ukuran kolom efektif $M1$ = ukuran baris efektif $M2$
- $MHasil$ menampung hasil perkalian antara $M1$ dan $M2$
 - Ukuran baris efektif $MHasil$ = ukuran baris efektif $M1$
 - Ukuran kolom efektif $MHasil$ = ukuran kolom efektif $M2$

Mengalikan 2 Matriks (2)

- Beberapa contoh:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} ax + by \\ cx + dy \end{pmatrix}$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} = \begin{pmatrix} a\alpha + b\gamma & a\beta + b\delta \\ c\alpha + d\gamma & c\beta + d\delta \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix} = \begin{pmatrix} 1a + 2b + 3c & 1d + 2e + 3f \\ 4a + 5b + 6c & 4d + 5e + 6f \\ 7a + 8b + 9c & 7d + 8e + 9f \end{pmatrix}$$

Mengalikan 2 Matriks (3)



```
//Mengalikan isi 2 matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M1[10][10]; int NBrS1, NKol1;
    int M2[10][10]; int NBrS2, NKol2;
    int MHasil[10][10]; int NBrSHsl, NKolHsl;
    int i, j, k;
    // ALGORITMA
    //Membaca isi matriks M1 dan M2 : Buat sebagai latihan

    // Deklarasi dan pengisian MHasil dengan perkalian M1 dan M2
    NBrSHsl = NBrS1; NKolHsl = NKol2;
    for (i=0; i<NBrSHsl; i++) {
        for (j=0; j<NKolHsl; j++) {
            MHasil[i][j] = 0;
            for (k=0; k<NKol1; k++) {
                MHasil[i][j] = MHasil[i][j] + (M1[i][k] * M2[k][j]);
            }
        }
    }
    // Menuliskan hasil penjumlahan ke layar : Buat sebagai latihan
    return 0;
}
```