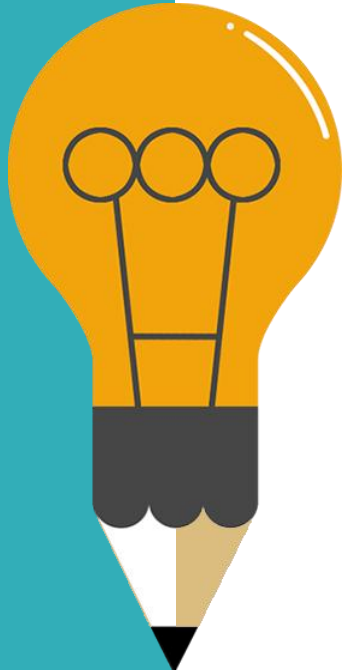


MATRIKS

BY
TAUFIK HIDAYAT, MT.

TODAY'S LESSON



01

DEFINISI MATRIKS

02

PEMROESESAN MATRIKS

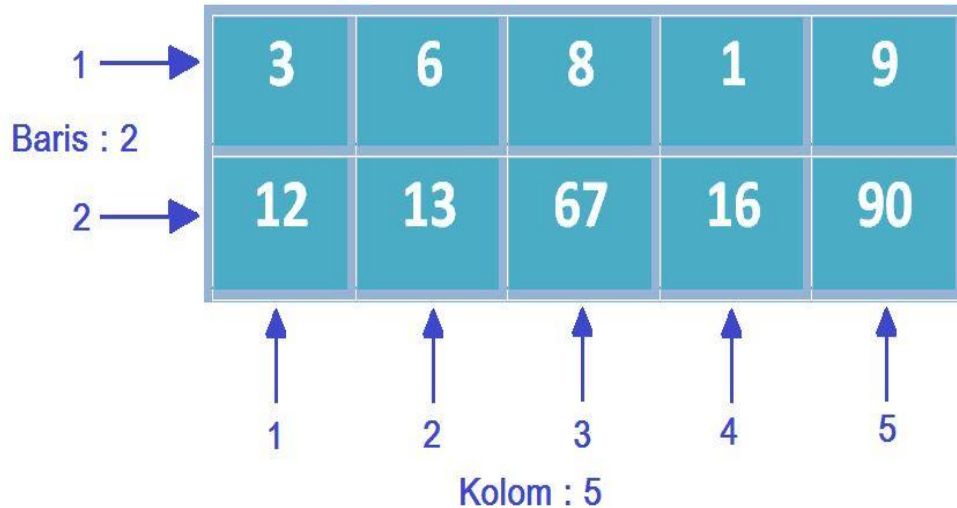
03

OPERASI MATRIKS

04

CONTOH PROGRAM

DEFINISI MATRIKS



The diagram illustrates a 2x5 matrix. The rows are indexed from 1 to 2 on the left, and the columns are indexed from 1 to 5 at the bottom. The matrix contains the following values:

1	3	6	8	1	9
2	12	13	67	16	90
	1	2	3	4	5

Baris : 2

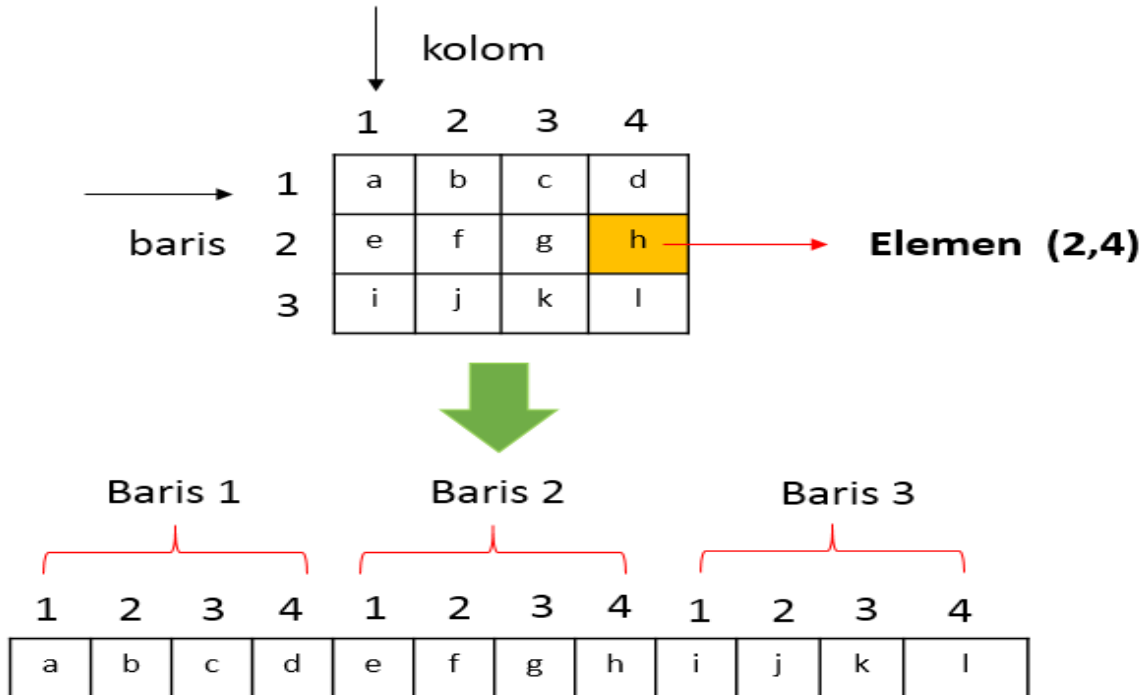
Kolom : 5

Matriks → struktur penyimpanan data di dalam memori yang setiap individu elemennya diacu dengan menggunakan dua indeks (baris dan kolom).

Karena ada dua indeks yang diperlukan maka matriks disebut sebagai array 2 dimensi.

Ilustrasi Matriks

Meskipun matriks merupakan array dua dimensi, nilai matriks yang disimpan di dalam memori tetap berurutan.



DEKLARASI MATRIKS

Sebagai nama peubah

DEKLARASI

M : array [1..5, 1..4] of interger

Sebagai tipe

DEKLARASI

type Mat : array [1..5, 1..4] of interger

M : Mat

Mendefiniskan ukuran maksimum matriks sebagai konstanta

DEKLARASI

const NbarisMaks = 20

const NkolomMaks = 20

M : array [1..NbarisMaks, 1..NKolomMaks] of integer



PEMROSESAN MATRIKS

Pemrosesan dengan menggunakan for

```
procedure ProsesMatriks1(input M : MatriksInt, input Nbar, Nkol : integer)
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per kolom.}
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
{K.Akhir : Setiap elemen matriks M telah diproses.}
```

DEKLARASI

i : integer

j : integer

ALGORITMA:

```
for i ← 1 to Nbar do
  for j ← 1 to Nkol do
    Proses(M[i, j])
```

endfor

endfor



PEMROSESAN MATRIKS

Pemrosesan dengan menggunakan while

```
procedure ProsesMatriks2(input M : MatriksInt, input Nbar, Nkol : integer)
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per kolom.}
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
{K.Akhir : Setiap elemen matriks M telah diproses.}
```

DEKLARASI

i : integer

j : integer

ALGORITMA:

i ← 1

while i ≤ Nbar do

j ← 1

while j ≤ Nkol do

proses (M[i, j])

j ← j+1

endwhile

i ← i+1

endwhile



PEMROSESAN MATRIKS

Pemrosesan dengan menggunakan repeat - until

```
procedure ProsesMatriks3(input M : MatriksInt, input Nbar, Nkol : integer)
{Pemrosesan elemen matriks M[1..Nbar, 1..Nkol] per baris per kolom.}
{K.Awal : Matriks M sudah terdefinisi elemen-elemennya.}
{K.Akhir : Setiap elemen matriks M telah diproses.}
```

DEKLARASI

i : integer

j : integer

ALGORITMA:

i ← 1

repeat

j ← 1

repeat

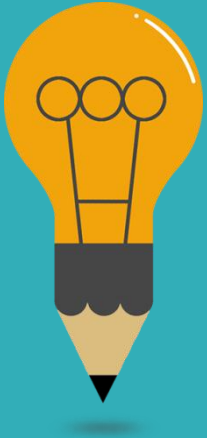
proses (M[i, j])

j ← j+1

until j>Nkol

i ← i+1

until i > Nbar



Operasi-Operasi pada Matriks

Mendeklarasikan dan mendefinisikan isi matriks

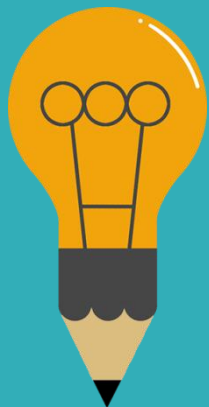
Menuliskan isi matriks ke layar

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



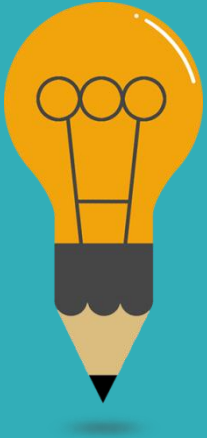
Mengisi dan Menuliskan Isi Sebuah Matriks

```
//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;
}
```

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



Membaca dan Menuliskan Isi Sebuah Matriks



```
//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;
}
```

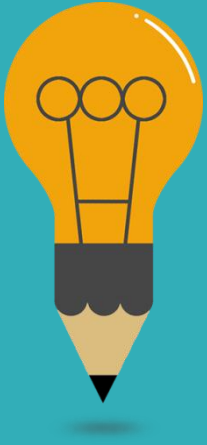
- Deklarasi matriks
- Inisialisasi baris+kolom efektif dari masukan user
- Membaca isi matriks dari user
- Menampilkan matriks

Menghitung Total Elemen

```
//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;
}
```

- Mentotal seluruh elemen yang ada di matrik dan menampilkan hasilnya kelayar



Mengalikan Elemen Matriks dgn Konstanta

```
// 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;

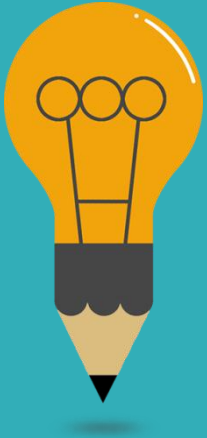
    // ALGORITMA
    // Membaca isi matriks : buat sebagai
    // latihan

    // 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

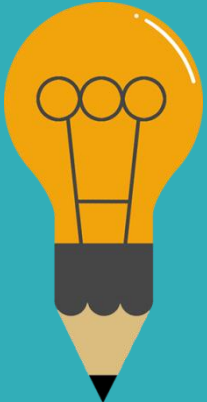
    return 0;
}
```

Setiap elemen matriks dikalikan dengan suatu konstanta, misal x



TRANSPOSE MATRIKS

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

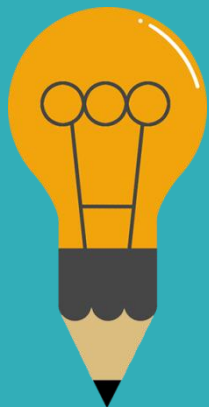


TRANSPOSE MATRIKS

```
//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

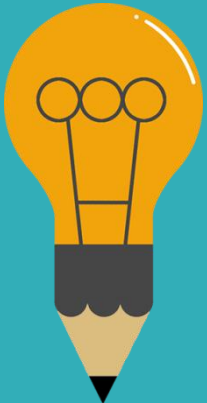
    // Deklarasi dan pengisian MTranspose
    NBrS2 = NKol1; NKol2 = NBrS1;
    for (i=0; i<NBrS2; i++) {
        for (j=0; j<NKol2; j++) {
            MTranspose[i][j] = M[j][i]; //transpose
        }
    }

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



PENJUMLAHAN MATRIKS

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



PENJUMLAHAN MATRIKS

$$\mathbf{A} = \begin{pmatrix} 3 & 5 & 6 \\ 1 & 9 & 2 \\ 5 & 1 & 2 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 0 & 4 \\ 7 & 1 & 1 \end{pmatrix}$$

Algoritma Penjumlahan Matriks

Input : A, B

Output : C

Rumus penjumlahan matriks : $c(i,j) = a(i,j) + b(i,j)$

Sehingga,

$$c[0][0] = a[0][0] + b[0][0]$$

$$c[0][1] = a[0][1] + b[0][1]$$

...dst



PENJUMLAHAN MATRIKS

```
//Menjumlahkan isi 2 matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M1[10][10]; int NBrsl, NKol1;
    int M2[10][10]; int NBrsl2, NKol2;
    int MHasil[10][10]; int NBrslHsl, NKolHsl;
    int i, j;
    // ALGORITMA
    //Membaca isi matriks M1 dan M2 : Buat sebagai latihan

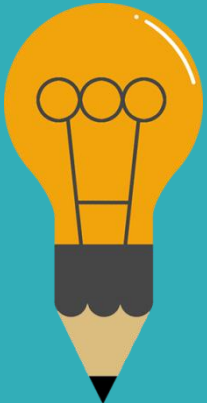
    // Deklarasi dan pengisian MHasil
    NBrslHsl = NBrsl; NKolHsl = NKol1;
    for (i=0; i<NBrslHsl; i++) {
        for (j=0; j<NKolHsl; j++) {
            MHasil[i][j] = M1[i][j] + M2[i][j]; //penjumlahan
        }
    }

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



PERKALIAN MATRIKS

- 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 $M2$



PERKALIAN MATRIKS

$$\mathbf{A} = \begin{pmatrix} 3 & 5 & 6 \\ 1 & 9 & 2 \\ 5 & 1 & 2 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 0 & 4 \\ 7 & 1 & 1 \end{pmatrix}$$

Algoritma Perkalian Matriks

Input $A[3][3], B[3][3]$

Output $C[3][3]$

Rumus perkalian matriks:

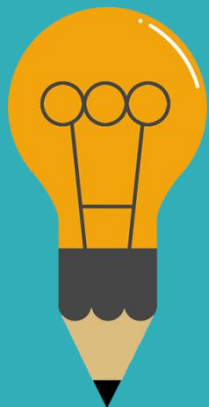
$$C[0][0] = A[0][0] * B[0][0] + A[0][1] * B[1][0] + A[0][2] * B[2][0]$$

$$C[0][1] = A[0][0] * B[0][1] + A[0][1] * B[1][1] + A[0][2] * B[2][1]$$

.. Dst

$$C[i][j] = \sum_{k=0}^{m-1} A[i][k] * B[k][j]$$

m = jumlah kolom matriks A = jumlah baris matriks B



PERKALIAN MATRIKS



```
//Mengalikan isi 2 matriks
#include <iostream>
using namespace std;
int main () {
    // KAMUS
    int M1[10][10]; int NBrsl, NKol1;
    int M2[10][10]; int NBrsl2, NKol2;
    int MHasil[10][10]; int NBrslHsl, NKolHsl;
    int i, j, k;
    // ALGORITMA
    //Membaca isi matriks M1 dan M2 : Buat sebagai latihan

    // Deklarasi dan pengisian MHasil dengan perkalian M1 dan M2
    NBrslHsl = NBrsl; NKolHsl = NKol2;
    for (i=0; i<NBrslHsl; 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;
}
```



Q & A