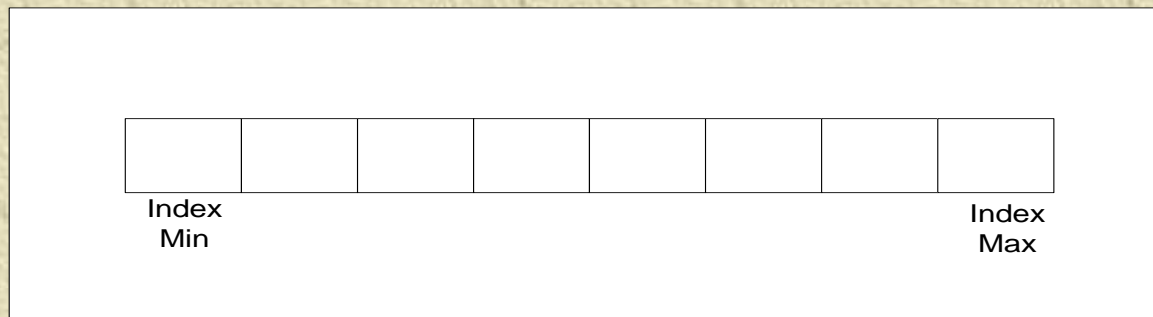


# ARRAY/LARIK

# Definisi Array

---

- ✦ Tipe Array adalah tipe yang mengacu kepada sebuah atau sekumpulan elemen melalui indeks[i] Elemen array dapat diakses langsung jika dan hanya jika indeks terdefinisi (ditentukan harganya sesuai dengan domain yang didefinisikan untuk indeks tersebut).



# Definisi Array

---

- ✦ Array digunakan untuk merepresentasikan sekumpulan informasi yang bertipe sama, dan disimpan dengan urutan yang sesuai dengan definisi indeks secara kontigu dalam memori komputer
- ✦ Indeks harus berupa struktur data ordinal
- ✦ Array disebut juga tabel/vektor/larik

# Deklarasi Array

---

- Deklarasi sebagai tipe

```
type
  Nama_type_array =
    array [IndexMin..IndexMax] of tipe_Elemen
Var
  Nama_var      : Nama_type_array
```

- Deklarasi sebagai variabel

```
Var
  Nama_Var : array [StartIndex..EndIndex]
             of tipe_elemen
```

# Contoh Deklarasi Array

---

- Deklarasi tipe array

```
type
    TabNamaHari = array [1..7] of string
Var
    Hari      : TabNamaHari
```

- Deklarasi variabel array

```
Var
    TabFrek : array ['a'..'z'] of integer
```

# Pengaksesan Array

---

- Cara Pengaksesan array

```
Nama_Var[indeks]
```

- Contoh Pengaksesan array

```
X := TabNamaHari[7]  
Y := TabFrek['d']  
For i:=1 to 7 do  
    Writeln (TabNamaHari[i])
```

# Skema Pemrosesan Pada Array

---

- ✦ Proses pada array dapat berupa proses inisialisasi, pengisian elemen array, output elemen array, dan proses kalkulasi lainnya.
- ✦ Inisialisasi : memberikan harga awal untuk seluruh elemen array
- ✦ Pengisian array berarti memberikan nilai terhadap elemen baik melalui perintah assignment atau input dari piranti masukan

# Skema Pemrosesan Pada Array

Pemrosesan beruntun pd array adalah pemrosesan secara berurutan mulai elemen pertama array, berurutan hingga elemen terakhir.

## SKEMA\_UMUM\_PEMROSESAN\_ARRAY

### KAMUS

Const N : integer = 100 {banyak elemen array}  
type larik : array[1..N] of integer {dekl array}  
i : integer[1..N] {pencacah indeks array}  
A : larik

### ALGORTIMA

For i  $\leftarrow$  1 to N do  
    Proses (A[i])  
endfor



# Inisialisasi Array

---

```
procedure init_array(var A:larik);  
{menginisialisasi setiap elemen A dengan nol  
IS : sembarang  
FS : seluruh elemen larik A berharga nol }  
{kamus lokal}  
var  
    i : integer;  
{algoritma}  
begin  
    for i:=1 to N do  
        A[i] := 0;  
end;
```

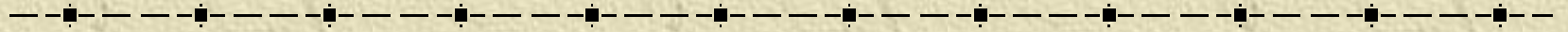
# Mengisi Elemen Array

```
procedure Isi_Larik(Neff: integer; var A:larik);  
{mengisi elemen larik dengan harga dari piranti  
masukan  
IS : sembarang  
FS : seluruh elemen larik A berisi nilai dari  
piranti masukan}  
var  
    i : integer;  
begin  
    for i:=1 to Neff do  
        begin  
            write('Elemen ke-',i,' : ');  
            readln(A[i]);  
        end;  
    end;  
end;
```

# Menulis Elemen Array

```
procedure Tulis_Larik(Neff:integer; A:larik);  
{menampilkan elemen larik [1..Neff] ke piranti  
keluaran  
IS : sembarang  
FS : seluruh elemen larik A[1..Neff] tercetak ke  
piranti keluaran}  
var  
    i : integer;  
begin  
    for i:=1 to Neff do  
        begin  
            writeln('Elemen ke-',i,': ',A[i]);  
        end;  
    readln;  
end;
```

# Contoh : Pencarian pd Array



Mencari suatu nilai dalam array, secara sekuensial.

dapat dilakukan dengan berbagai cara :

- ✦ tanpa peubah boolean
- ✦ dengan peubah boolean

```

Procedure Pencarian1 (A : larik; X : integer; var
idx : integer);
{Mencari nilai X pada array, versi tanpa boolean}
IS : harga X dan elemen array A[1..N] terdefinisi
FS : idx berisi indeks X ditemukan, jika tdk idx
    bernilai -9999 }
var
    i      : integer; {indeks array}
Begin
    i := 1;
    while ((I < Neff) and (A[i] <> X)) do
        i := i + 1;
    {(I = N) or (A[i] = X)}
    if (A[i] = X) then {X ditemukan}
        idx := I
    else {A[i] <> X}
        idx := -9999;
    end;
end;

```

```

Procedure Pencarian2 (A : larik; X : integer; var
indeks : integer);
{Mencari nilai X pada array, versi dgn boolean
var
  i : integer; {indeks array}
  ketemu : boolean;
Begin
  i := 1; ketemu := false;
  while ((I < Neff) and (not ketemu)) do
    if A[i]= X then
      ketemu := true
    else
      i := i + 1;
  if (ketemu) then {X ditemukan}
    indeks := i
  else {A[i] <> X}
    indeks := -9999;
end;

```

# Contoh : Mencari Harga Ekstrim (min/max)

```
Function Nilai_Max (A : larik) : integer;  
{Mencari Nilai maksimum dari suatu tabel}  
-----  
var  
    i      : integer;      {pencacah indeks array}  
    Max    : integer;      {nilai maksimum}  
    idx    : integer;      {indeks dari nilai maksimum}  
Begin  
    i      := 2; Max := A[1]; idx := 1;  
    while (i <= Neff) do  
        begin  
            if (A[i] > Max) then  
                idx := i;  
            i := i + 1;  
        end;  
    Nilai_Max := idx;  
End;
```

Bbrp fungsi/prosedur tsb dpt dipakai dlm program utama sbb :

```
program Proses_Larik;
uses crt;
-----

var
    A      : larik;
    Neff   : integer;
    z      : integer;
    indeks : integer;

{Deklarasi fungsi/prosedur}
procedure init_array(var A:larik);
procedure Isi_Larik(Neff: integer; var A:larik);
procedure Tulis_Larik(Neff:integer; A:larik);
Procedure Pencarian1 (A:larik; X:integer; var
indeks:integer);
Function Nilai_Max (A : larik) : integer;
```



## Lanjutan program :

```
{Program Utama}
```

```
Begin
```

```
write('jumlah elemen yang akan diinputkan : ');
```

```
readln(Neff);
```

```
Isi_Larik(Neff,A);
```

```
writeln('Elemen Array Yang Diinputkan :');
```

```
Tulis_Larik(Neff,A);
```

```
write('Inputkan nilai yang akan dicari : ');
```

```
readln(z);
```

```
pencarian1(A,z,indeks);
```

```
if indeks <> -9999 then
```

```
    writeln('Nilai',z,'ditemukan pd indeks ke',indeks)
```

```
else
```

```
    writeln('nilai tidak ditemukan');
```

```
writeln('Nilai maksimum = ',A[Nilai_Max(A)]);
```

```
end.
```

## Output Program :

```
C:\TP\BIN\TURBO.EXE
PROGRAM ARRAY
=====

MENGINPUTKAN ELEMEN ARRAY
jumlah elemen yang akan diinputkan : 5
Elemen ke-1: 62
Elemen ke-2: 785
Elemen ke-3: 4
Elemen ke-4: 15
Elemen ke-5: 985

Elemen Array Yang Diinputkan :
Elemen ke-1: 62
Elemen ke-2: 785
Elemen ke-3: 4
Elemen ke-4: 15
Elemen ke-5: 985

PENCARIAN NILAI PADA ARRAY
Inputkan nilai yang akan dicari : 4
Nilai 4 ditemukan pada indeks ke-3
Nilai maksimum = 985
```



---

# MATRIKS

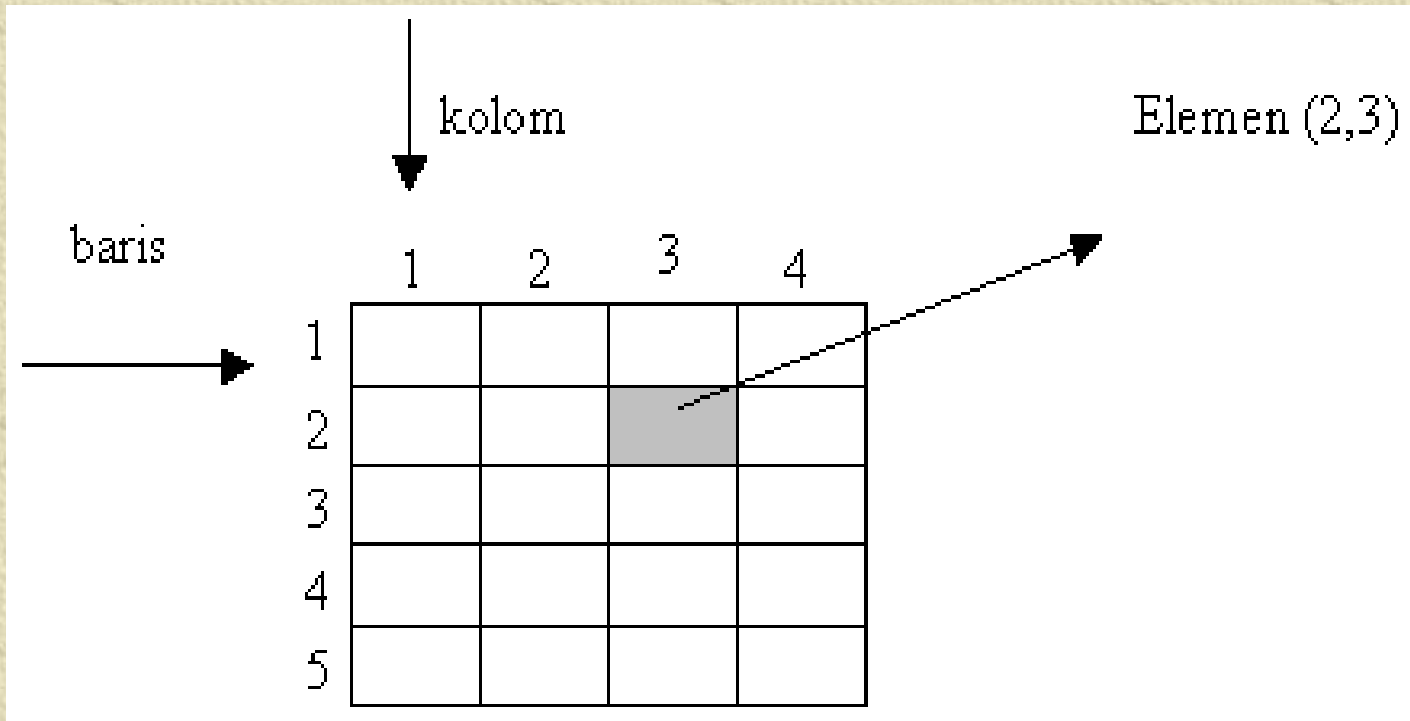
# Definisi Matriks

---

- ✦ Matriks merupakan struktur penyimpanan data dalam memori yang setiap elemennya diacu dengan menggunakan dua buah indeks (biasanya dikenal dengan nama baris dan kolom).
- ✦ matriks disebut juga sebagai array dua dimensi [2].

# Definisi Matriks

Ilustrasi matriks yang terdiri dari 4 baris 5 kolom



# Deklarasi Matriks

-----  
Deklarasi dalam bentuk tipe :

```
type
  Nama_type_Matriks =
    array [IndexMinBaris..IndexMaxBaris,
          IndexMinKolom..IndexMaxKolom]
    of tipe_Elemen
Var
  Nama_var      : Nama_type_Matriks
```

Contoh :

```
type
  Matriks = array [1..10,1..10] of integer;
Var
  M      : Matriks
```

# Deklarasi Matriks

-----  
Deklarasi matriks dalam bentuk variabel

```
var  
  Nama_var :  
    array[IndexMinBaris..IndexMaxBaris,  
          IndexMinKolom..IndexMaxKolom]  
  of tipe_Elemen
```

Contoh :

```
Var  
  Matriks = array [1..10,1..10] of integer;
```

# Pengaksesan Matriks

Pengaksesan terhadap Matriks dilakukan mengacu pada kedua indeksnya sbb :

```
Nama_Var[indeksBaris, indeksKolom]
```

Contoh :

$M[4, 5] := 8$        $\rightarrow$  elemen baris 4 kolom 5 diisi dgn 8  
 $Z := M[2, 2]$        $\rightarrow$  Z diisi dgn elemen baris 2 kolom 2



# Skema Pemrosesan Pada Matriks

Pemrosesan beruntun pd array adalah pemrosesan mulai elemen pertama matriks  $M[1,1]$ , berurutan hingga elemen terakhir  $M[Nb,Nk]$ .

## SKEMA\_UMUM\_PEMROSESAN\_MATRIKS

### KAMUS

#### const

Nb = 6      {jumlah baris maksimum}

Nk = 6      {jumlah kolom maksimum}

#### type

Matriks = array[1..Nb,1..Nk] of integer

M : Matriks

i, j : integer[1..N]

### ALGORTIMA

```
For i ← 1 to Nb do  
    For j ← 1 to Nk do  
        Proses (M[i, j])  
endfor
```

# Inisialisasi Matriks

```
-----  
Procedure InitMatriks (var M :Matriks; bar,kol :  
integer) ;  
{Menginisiliasi elemen matriks dengan nilai 0  
IS : -  
FS : seluruh elemen matriks berisi dengan nilai 0 }  
  
Var  
    i : integer;    {indeks kolom}  
    j : integer;    {indeks baris}  
  
Begin  
    For i := 1 to bar do  
        For j := 1 to kol do  
            M[I,J] := 0  
        end;  
    end;  
end;
```

# Pengisian Elemen Matriks

```
Procedure Isi_Matriks (var M: Matriks; bar, kol
: integer);
{Mengisi elemen matriks dari piranti masukan}
IS : -
FS : seluruh elemen matriks berisi dengan nilai yang
diinputkan user dari piranti masukan }

Var
    i : integer; {indeks kolom}
    j : integer; {indeks baris}
Begin
    For I := 1 to bar do
        For J := 1 to kol do
            begin
                Write ('[', i, ', ', j, ' ] = ' );
                Readln (M[I, J]);
            end;
        end;
    end;
end;
```

# Menulis Elemen Matriks

```
Procedure Tulis_Matriks(var M :Matriks; bar,kol :  
integer);
```

```
{Menuliskan elemen matriks ke piranti keluaran  
IS : elemen matriks sudah terdefinisi harganya  
FS : seluruh elemen matriks tertulis dilayar}
```

Var

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

Begin

```
  For i := 1 to bar do  
    begin  
      For j := 1 to kol do  
        Write(M[I,J]:3,' ');  
        writeln; {pindah baris}  
      end;  
    end;
```

```
end;
```

# Contoh Kasus : Penjumlahan Matriks

Menjumlahkan matriks  $A + B$ , dengan asumsi  $A$  dan  $B$  merupakan matriks yang memiliki dimensi kolom dan baris yang sama, dan sudah berisi nilai. Hasil penjumlahan akan ditampung di matriks  $C$ , dimana matriks  $C$  berdimensi sama dengan matriks  $A$  dan  $B$ .

$$\begin{bmatrix} 5 & 20 \\ 43 & 8 \\ 20 & 14 \end{bmatrix} + \begin{bmatrix} 10 & 11 \\ 12 & 13 \\ 14 & 15 \end{bmatrix} = \begin{bmatrix} 5 + 10 & 20 + 11 \\ 43 + 12 & 8 + 13 \\ 20 + 14 & 14 + 15 \end{bmatrix} = \begin{bmatrix} 15 & 31 \\ 55 & 21 \\ 34 & 29 \end{bmatrix}$$

Matriks A                  Matriks B                  Matriks C

# Contoh Kasus : Penjumlahan Matriks

```
Procedure JumlahMatriks(A,B : Matriks; bar,kol
:integer; var C : Matriks);
{IS : Matriks A dan B sudah terdefinisi dan terisi
nilai
FS : Menjumlahakan matriks A dan B dan menyimpan
hasil di matriks C}

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

Begin
    For i := 1 to bar do
        For j := 1 to kol do
            C[I,J] := A[I,J] + B[I,J];
end;
```

# Contoh Kasus : Perkalian Matriks

Mengalikan dua matriks  $A[1..m,1..n]$  dan  $B[1..n,1..p]$  dan menyimpan hasilnya di matriks  $C[1..m,1..p]$ .

Syarat : ukuran kolom matriks A = ukuran baris matriks B.

Matriks A

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

x

Matriks B

$$\begin{bmatrix} 10 & 11 \\ 12 & 13 \\ 14 & 15 \end{bmatrix}$$

=

$$\begin{bmatrix} (1*10) + (2*12) + (3*14) & (1*11) + (2*13) + (3*15) \\ (4*10) + (5*12) + (6*14) & (4*11) + (5*13) + (6*15) \\ (7*10) + (8*12) + (9*14) & (7*11) + (8*13) + (9*15) \end{bmatrix} = \begin{bmatrix} 76 & 82 \\ 184 & 199 \\ 292 & 316 \end{bmatrix}$$

Matriks C

# Contoh Kasus : Perkalian Matriks

Proses perkalian matriks :

- Inisialisasi matriks hasil  $C[1..m,1..p]$  dengan 0
- Untuk tiap baris pada matriks  $A[1..m,1..n]$ ,  $i = 1, 2, 3..m$ 
  - Untuk setiap kolom  $j$  pada matriks  $B$ ,  $j = 1,2,3..p$ 
    - Untuk setiap baris  $k$  pada matriks  $B$ ,  $k = 1,2,3..n$ 
$$C[i,j] = C[i,j] + A[i,k] * B[k,j]$$



# Contoh Kasus : Perkalian Matriks

```
procedure kali_matriks (M1,M2 : matriks; var M3:
matriks; m,n,p : integer; var barC,kolC : integer);
var
    i,j,k : integer;
begin
bar3 := m; kol3 := p;
{inisialisasi matriks C}
for i:=1 to barC do
    for j:= 1 to kolC do
        M3[i,j] := 0;
{operasi perkalian}
for i:= 1 to m do
    for j:=1 to p do
        for k:= 1 to n do
            M3[i,j] := M3[i,j] + M1[i,k] * M2[k,j];
end;
```

Bbrp fungsi/prosedur tsb dpt dipakai dlm program utama sbb :

```
program Penggunaan_Matriks;
Var
  M1, M2, M3 : Matriks;
  bar1, kol1 : integer;
  bar2, kol2 : integer;
  bar3, kol3 : integer;
{Deklarasi fungsi/prosedur}
.....

{program utama}
begin;
  {input dimensi matrtriks M1}
  write ('ukuran baris M1: '); readln(bar1);
  write ('ukuran kolom M1: '); readln(kol1);
  {input dimensi matrtriks M2}
  write ('ukuran baris M2: '); readln(bar2);
  write ('ukuran kolom M2: '); readln(kol2);
```

lanjutan program :

```
{Inisialisasi Matriks}
```

```
InitMatriks (M1,bar1,kol1) ;
```

```
InitMatriks (M2,bar2,kol2) ;
```

```
InitMatriks (M3,bar3,kol3) ;
```

```
{Mengisi matriks}
```

```
writeln('Pengisian Matriks Pertama : ');
```

```
Isi_Matriks (M1,bar1,kol1) ;
```

```
writeln('Pengisian Matriks Kedua : ');
```

```
Isi_Matriks (M2,bar2,kol2) ;
```

```
{Menuliskan Matriks}
```

```
tulis_Matriks (M1,bar1,kol1) ;
```

```
tulis_Matriks (M2,bar2,kol2) ;
```

lanjutan program :

```
{Penjumlahan Matriks}
  if(bar1 = bar2) and (kol1 = kol2) then
    begin
      -----bar3:=bar1; kol3 := kol1;-----
      JumlahMatriks (M1,M2,bar3,kol3,M3) ;
      tulis_Matriks (M3,bar3,kol3) ;
    end
  else
    write ('Matriks M1 & M2 tdk dpt dijumlahkan');

{Perkalian matriks}
  if (kol2 = bar1) then
    begin
      kali_matriks (M1,M2,M3,bar1,kol1,bar2,bar3,kol3) ;
      tulis_Matriks (M3,bar3,kol3) ;
    end
  else
    write ('Matriks M1 dan M2 tidak dapat dikalikan');
end.
```

# Contoh Output Program

```
C:\> C:\TP\BIN\TURBO.EXE
PROGRAM PENJUMLAHAN MATRIKS
=====
ukuran baris M1: 2
ukuran kolom M1: 2
ukuran baris M2: 2
ukuran kolom M2: 2

Isi M1 Setelah Inisialisasi :

  0   0
  0   0

Pengisian Matriks Pertama :

[1,1] = 1
[1,2] = 2
[2,1] = 3
[2,2] = 4

Pengisian Matriks Kedua :

[1,1] = 5
[1,2] = 6_
```

```
C:\> C:\TP\BIN\TURBO.EXE

[1,1] = 5
[1,2] = 6
[2,1] = 7
[2,2] = 8

Isi Matriks Pertama :

  1   2
  3   4

Isi Matriks Kedua :

  5   6
  7   8

Matriks Hasil Penjumlahan M1 dan M2 :

  6   8
 10  12

Matriks hasil Perkalian M1 dan M2 :

 19  22
 43  50
```

# Latihan Soal

---

## ✦ Kasus Palindrom

Buatlah fungsi yang meminta user menginputkan sejumlah karakter yang membentuk sebuah kata, fungsi kemudian mengecek apakah suatu kata yang tersimpan dalam array of char adalah kata yang palindrom atau bukan.

Fungsi akan mengembalikan nilai true jika palindrom dan false jika tidak.

Contoh kata :

K A T A K

→ Palindrom

K A S U R

→ Bukan Palindrom

KASURUSAK

→ Palindrom

# Latihan Soal

---

## ✧ Mencari Nilai X terdapat pada suatu matriks.

Buatlah sebuah fungsi untuk mencari elemen X pada sebuah matriks. Fungsi akan mengembalikan nilai True jika X ditemukan dan false jika tidak

## ✧ Mencari Nilai minimum ekstrim pada matriks

Buatlah prosedur untuk mencari nilai ekstrim (minimum dan maksimum) yang terdapat pada suatu matriks. Prosedur akan mengeluarkan output berupa posisi serta nilai ekstrim yang ditemukan.

## ✧ Matriks diagonal

Buatlah prosedur untuk membuat suatu matriks diagonal. Matriks diagonal didefinisikan sebagai matriks bujur sangkar dengan elemen untuk posisi diagonal = 1 dan elemen untuk posisi lainnya = 0. Dimensi matriks diinputkan oleh user dari piranti masukan.

# Latihan Soal

---

## ✦ Matriks Transpose

Buatlah prosedur untuk melakukan transpose matriks. Matriks Transpose diperoleh dengan membuat seluruh elemen baris ke-I matriks asal menjadi elemen-elemen kolom ke-J (pada matriks transpose) atau secara matematis dituliskan  $A_{transpose}[j,i] = A[i,j]$

## ✦ Matriks segitiga bawah

Buatlah fungsi yang mengecek apakah sebuah matriks bujursangkar merupakan matriks segitiga bawah. Matriks segitiga bawah merupakan matriks dengan elemen di atas diagonal utama adalah 0, dan nilai pada diagonal utama tidak nol



# Referensi

---

1. **Diktat Kuliah Algotirma & Pemrograman Bag I. Inggriani Liem, ITB**
2. **Algoritma & Pemrograman, jilid 1 dan 2 Rinaldi Munir & Leoni Lidya, Penerbit Informatika**