Adam Mukharil Bachtiar
English Class
Informatics Engineering 2011

Algorithms and Programming

# Array

# Steps of the Day

1. Definition of Array
2. One Dimension Array
3. Two Dimensions Array

Let's Start

# 1 Definition of Array

# Background of Array

I need a program to process students data but i want it to **keep all data temporary in memory** so i can use it **until the program is shut down**.

# What is Array

Data structure that **saves a group of variables** which **have same type**.

# Ilustration of Array

An Array was named **bil**, has **integer** type,

and consists of **five** elemens.

| bil | 5 | 1 | 4 | 2 | 7 |
|---|---|---|---|---|---|
| | [1] | [2] | [3] | [4] | [5] |

SUBSCRIPT / INDEX

# Types of Array

- One Dimension Array
- Two Dimensions Array
- **Many Dimensions Array (i will not explain THIS!!!)**

# 2 One Dimension Array

Definition and Structures of One Dimension Array

# What is One Dimension Array

Array that only has **one subscript / index**.

# Declaration of One Dimension Array

- As variable
- As user-defined data type
- Define size of array as constant

# Declaration As Variable (Algorithm)

```
Kamus:

  NamaArray : array [1..MaxSize] of TipeData

Contoh:

Kamus:

  bil : array [1..5] of integer

  NamaDosen : array [1..20] of string

  Pecahan : array [1..100] of real
```

# Declaration As Variable (Pascal)

```
var

    NamaArray : array [1..MaxSize] of TipeData;

Contoh:

var

    bil : array [1..5] of integer;

    NamaDosen : array [1..20] of string[30];

    Pecahan : array [1..100] of real;
```

# Declaration As User-Defined Data Type (Algorithm)

```
Kamus:

  type

    NamaArray = array [1..MaxSize] of TipeData

  NamaVariabel_1:NamaArray

  NamaVariabel_2:NamaArray
```

# Declaration As User-Defined Data Type (Algorithm)

```
Contoh:

Kamus:

   type

     bil = array [1..5] of integer

   bilbulat:bil

   bilpositif:bil
```

# Declaration As User-Defined Data Type (Pascal)

```
type
    NamaArray = array [1..MaxSize] of TipeData;
var
    NamaVariabel_1:NamaArray;
    NamaVariabel_2:NamaArray;
```

# Declaration As User-Defined Data Type (Pascal)

```
Contoh:

type

   bil = array [1..5] of integer;

var

   bilbulat:bil;

   bilpositif:bil;
```

# Define Size of Array As Constant (Algorithm)

```
Kamus:
  const
    MaxSize = VALUE
  type
    NamaArray = array [1..MaxSize] of TipeData
  NamaVariabel_1:NamaArray
  NamaVariabel_2:NamaArray
```

# Define Size of Array As Constant (Algorithm)

```
Contoh:

Kamus:

  const

    maks = 5

  type

    bil = array [1..maks] of integer

  bilbulat:bil
```

# Define Size of Array As Constant (Pascal)

```
const
    MaxSize = VALUE;
type
    NamaArray : array [1..MaxSize] of TipeData;
var
    NamaVariabel:NamaArray;
```

# Define Size of Array As Constant (Pascal)

```
Contoh:

const

   maks = 5;

type

   bil = array [1..maks] of integer;

var

   bilbulat:bil;
```

# Get and set the Value from Array

To fill and access the value in array, **call the name of array** and **its subscript** that you want to access

# Illustration of Setting and Getting Value in Array

| bil | 5 | 1 | 4 | 2 | 7 |
|---|---|---|---|---|---|
| | [1] | [2] | [3] | [4] | [5] |

**bil[1]=5** → it means fill 5 in [1]

**a=bil[2]** → a will be filled by 1

# Format of Accessing Array (Algorithm)

```
namaarray[indeks] ← nilai

input(namaarray[indeks])

namaarray[indeks] ← namaarray[indeks] + 1

output(namaarray[indeks])
```

# Format of Accessing Array (Algorithm)

```
namaarray[indeks] := nilai;

readln(namaarray[indeks]);

namaarray[indeks] := namaarray[indeks] + 1;

writeln(namaarray[indeks]);
```

## Operation in Array

- Creation
- Traversal
- **Searching**
- **Sorting**
- Destroy

## Array Creation

- **Prepare** array to be **accessed/processed**.

  Array will be filled with **default value**.

- For **numeric array** will be filled with **0** and

  for **alphanumeric array** will be filled with **‘ ’**

  **(Null Character)**

# Array Creation (Algorithm)

```
Procedure create (output NamaVarArray:NamaArray)
{I.S: elemen array diberi harga awal agar siap digunakan}
{F.S: menghasilkan array yang siap digunakan}

Kamus:
   indeks:integer

Algoritma:
   for indeks ← 1 to maks_array do
      nama_var_array[indeks]← 0 {sesuaikan dengan tipe array}
   endfor

EndProcedure
```

# Array Creation (Pascal)

```
procedure create (var NamaVarArray:NamaArray);


var
   indeks:integer;


begin
   for indeks := 1 to maks do
      NamaVarArray[indeks] := 0;
end;
```

## Array Traversal

The process of **visiting all elements** of array **one by one**, from **the first element** until **last element**.

# Traversal Processes

- Fill elements array with data
- Output all elements of array
- Adding data to array
- Insert data in particular index
- Delete data in particular index
- Determine maximum and minimum data in array
- Count mean value in array

# General Form for Array Traversal (Algorithm)

```
Procedure traversal (I/O NamaVarArray:NamaArray)
{I.S: maksimum array sudah terdefinisi}
{F.S: menghasilkan array yang sudah diproses}

Kamus:

Algoritma:
   for indeks ← 1 to maks do
      {proses traversal}
   endfor
   Terminasi {sifatnya optional}

EndProcedure
```

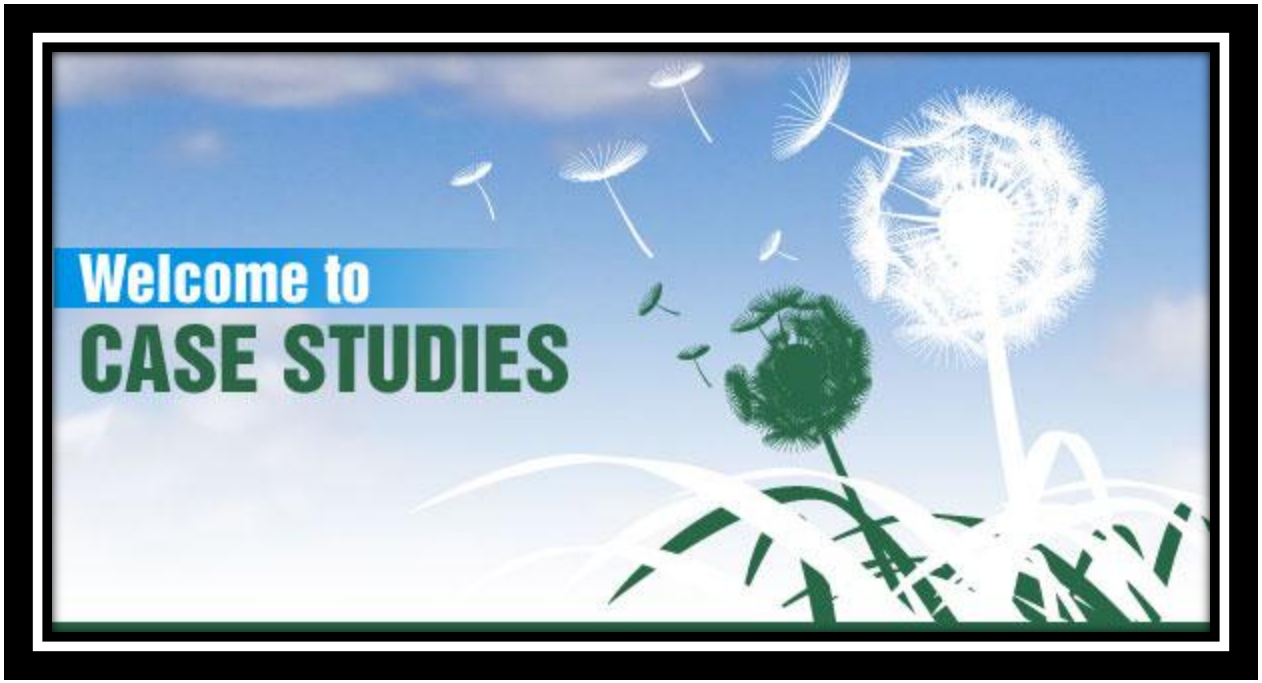
# General Form for Array Traversal (Pascal)

```
procedure traversal(var NamaVarArray:NamaArray);


begin
   for indeks := 1 to maks do
      {proses traversal yang dipilih}
   terminasi {sifatnya optional}
end;
```

## Destroy the Array

The process to **return value of array** into **default value** that was given in **array creation**.


Welcome to
CASE STUDIES

# Example of One Dimension Array (Algorithm)

```
Algoritma ArrayDasar
{I.S.: Dideklarasikan dua buah array satu dimensi}
{F.S.: Menampilkan array beserta hasil perhitungan}

Kamus:
const
     maks=5
type
     bil=array[1..maks] of integer

bil1,bil2:bil
i:integer
jumlah,jumlah2:integer
```

# Example of One Dimension Array (Algorithm)

```
Algoritma:
      {input elemen array}
      for i ← 1 to maks do
         input(bil1[i])
      endfor

      for i ← 1 to maks do
         input(bil2[i])
      endfor

      {output elemen array}
      for i ← 1 to maks do
         output(bil1[i])
      endfor
```

# Example of One Dimension Array (Algorithm)

```
28      for i ← 1 to maks do
29         output(bil2[i])
30      endfor
31
32      {proses perhitungan array}
33      jumlah←0;
34      for i ← 1 to maks do
35         jumlah←jumlah+bil1[i]
37      endfor
38      output(jumlah)
39
40      jumlah2←0;
```

# Example of One Dimension Array (Algorithm)

```
for i ← 1 to maks do
    jumlah2←jumlah2+bil2[i]
endfor
output(jumlah2)
```

# Example of One Dimension Array (Pascal)

```
1   program ArrayDasar;
2   uses crt;
3
4   const
5        maks=5;
6
7   type
8        bil=array[1..maks] of integer;
9
10  var
11       bil1,bil2:bil;
12       i:integer;
13       jumlah,jumlah2:integer;
```

# Example of One Dimension Array (Pascal)

```
14  begin
15        {input elemen array}
16        for i:=1 to maks do
17        begin
18           write('Masukkan nilai ke bil 1 [',i,'] : ');
19           readln(bil1[i]);
20        end;
21
22        writeln();
23        for i:=1 to maks do
24        begin
25           write('Masukkan nilai ke bil 2 [',i,'] : ');
26           readln(bil2[i]);
27        end;
```

# Example of One Dimension Array (Pascal)

```
28        {output elemen array}
29        for i:=1 to maks do
30        begin
31           writeln('Bil 1[',i,'] = ',bil1[i]);
32        end;
33
34        writeln();
35        for i:=1 to maks do
37        begin
38           writeln('Bil 2[',i,'] = ',bil2[i]);
39        end;
40
```

# Example of One Dimension Array (Pascal)

```
41      {proses perhitungan array}
42      writeln();
43      jumlah:=0;
44      for i:=1 to maks do
45      begin
46          jumlah:=jumlah+bil1[i];
47      end;
48      writeln('Jumlah elemen array bil 1 = ',jumlah);
49
50      writeln();
51      jumlah2:=0;
52      for i:=1 to maks do
53      begin
```

# Example of One Dimension Array (Pascal)

```
54          jumlah2:=jumlah2+bil2[i];
55      end;
56      writeln('Jumlah elemen array bil 2 = ',jumlah2);
57
58      writeln();
59      write('Tekan sembarang tombol untuk menutup...');
60      readkey();
61  end.
```

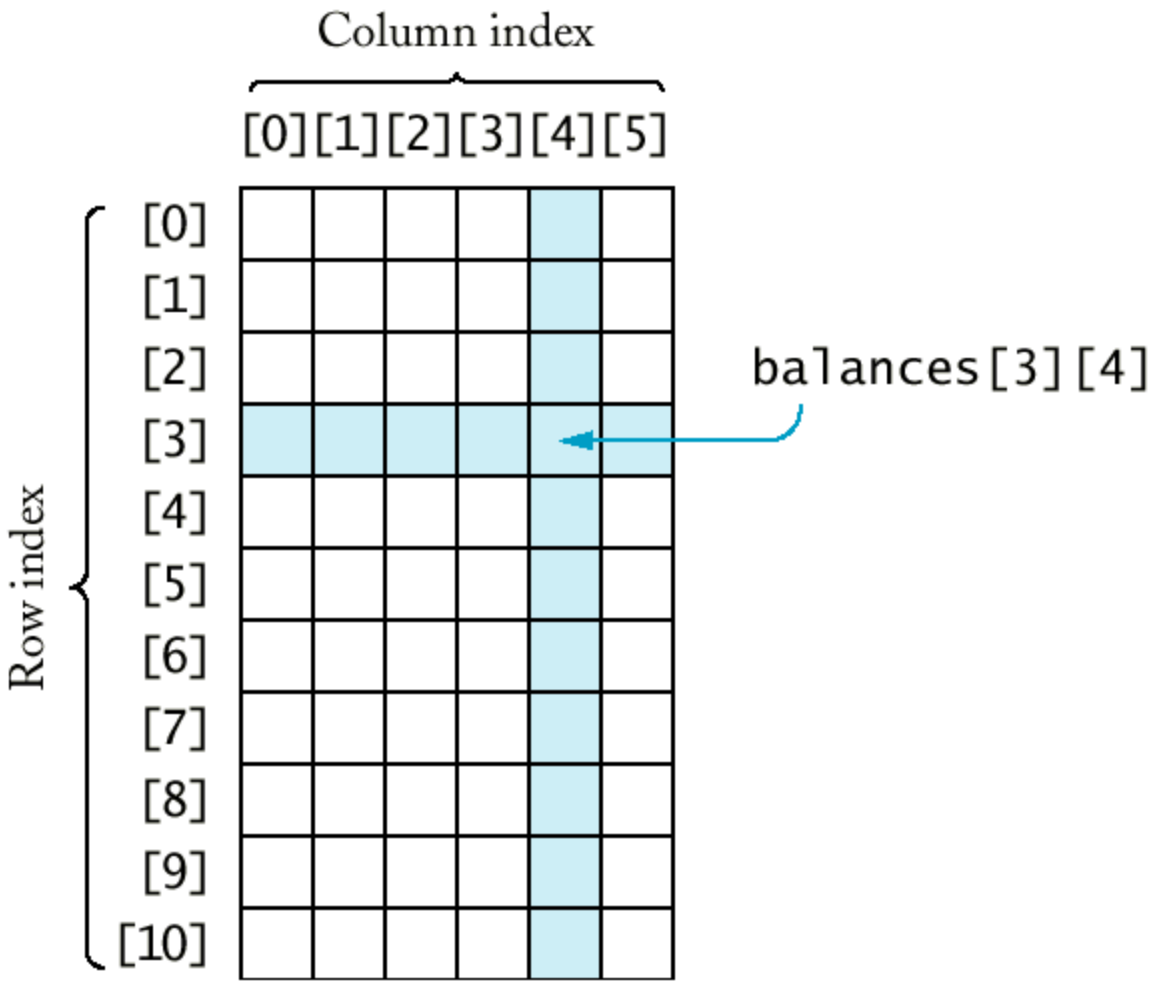
# 3 Two Dimensions Array

Definition and Structures of Two Dimensions Array

# What is Two Dimensions Array

Array that has **two subscripts** in its declaration. It often was called **matrix**.

# Illustration of Two Dimensions Array



# Declaration As Variable (Algorithm)

```
Kamus:

  NamaArray : array [1..MaxBaris,1..MaxKolom] of TipeData


Contoh:

Kamus:

  matriks : array [1..5,1..5] of integer
```

# Declaration As Variable (Pascal)

```
var

 NamaArray : array [1..MaxBaris,1..MaxKolom] of TipeData;


Contoh:

var

   matriks: array [1..5,1..5] of integer;
```

# Declaration As User-Defined Data Type (Algorithm)

```
Kamus:

  type

    NamaArray = array [1..MaxBaris,1..MaxKolom] of TipeData

  NamaVariabel_1:NamaArray

  NamaVariabel_2:NamaArray
```

# Declaration As User-Defined Data Type (Algorithm)

```
Contoh:

Kamus:

   type

      matriks = array [1..5,1..5] of integer

   matriks1:matriks
```

# Declaration As User-Defined Data Type (Pascal)

```
type
    NamaArray = array [1..MaxBaris,1..MaxKolom] of TipeData;
var
    NamaVariabel_1:NamaArray;
    NamaVariabel_2:NamaArray;
```

# Declaration As User-Defined Data Type (Pascal)

```
Contoh:

type

    matriks = array [1..5,1..5] of integer;

var

    matriks1:bil;

    matriks2:bil;
```

# Define Size of Array As Constant (Algorithm)

```
Kamus:
  const
    MaxBaris = VALUE1
    MaxKolom = VALUE2
  type
    NamaArray = array [1..MaxBaris,1..MaxKolom] of TipeData
  NamaVariabel_1:NamaArray
  NamaVariabel_2:NamaArray
```

# Define Size of Array As Constant (Algorithm)

```
Contoh:

Kamus:

  const

    MaksBaris = 5

    MaksKolom = 5

  type

    matriks = array [1..MaksBaris,1..MaksKolom] of integer

  matriks1,matriks2:bil
```

# Define Size of Array As Constant (Pascal)

```
const

   MaxBaris = VALUE1;

   MaxKolom = VALUE2;

type

   NamaArray : array [1..MaxBaris,1..MaxKolom] of TipeData;

var

   NamaVariabel:NamaArray;
```

# Define Size of Array As Constant (Pascal)

```
Contoh:

const

   MaksBaris = 5;

   MaksKolom = 5;

type

   matriks = array [1..MaksBaris,1..MaksKolom] of integer;

var

   bilbulat:bil;
```

# Operation in Two Dimensions Array

Operation in two dimensions array **is same**

**as** operation in one dimensions array.

## Operation in Array

- Creation
- Traversal
- **Searching**
- **Sorting**
- Destroy

## Array Creation

- **Prepare** array to be **accessed/processed**.

  Array will be filled with **default value**.

- For **numeric array** will be filled with **0** and

  for **alphanumeric array** will be filled with **‘ ’**

  **(Null Character)**

# Array Creation (Algorithm)

```
Procedure create (output NamaVarArray:NamaArray)
{I.S: elemen array diberi harga awal agar siap digunakan}
{F.S: menghasilkan array yang siap digunakan}

Kamus:
   i,j:integer

Algoritma:
   for i ← 1 to MaksBaris do
     for j← 1 to MaksKolom do
      nama_var_array[i,j] ← 0 {sesuaikan dengan tipe array}
     endfor
   endfor

EndProcedure
```

# Array Creation (Pascal)

```
procedure create (var NamaVarArray:NamaArray);

var
   i,j:integer;

begin
   for i := 1 to MaksBaris do
   begin
     for j := 1 to MaksKolom do
        NamaVarArray[i,j] := 0;
   end;
end;
```

## Array Traversal

The process of **visiting all elements** of array **one by one**, from **the first element** until **last element**.

# Traversal Processes

- Fill elements array with data
- Output all elements of array
- Adding data to array
- Insert data in particular index
- Delete data in particular index
- Determine maximum and minimum data in array
- Count mean value in array

# General Form for Array Traversal (Algorithm)

```
Procedure traversal (I/O NamaVarArray:NamaArray)
{I.S: maksimum array sudah terdefinisi}
{F.S: menghasilkan array yang sudah diproses}


Kamus:


Algoritma:
   for i ← 1 to MaksBaris do
     for j ← 1 to MaksKolom do
      {proses traversal}
     endfor
   endfor
   Terminasi {sifatnya optional}
EndProcedure
```

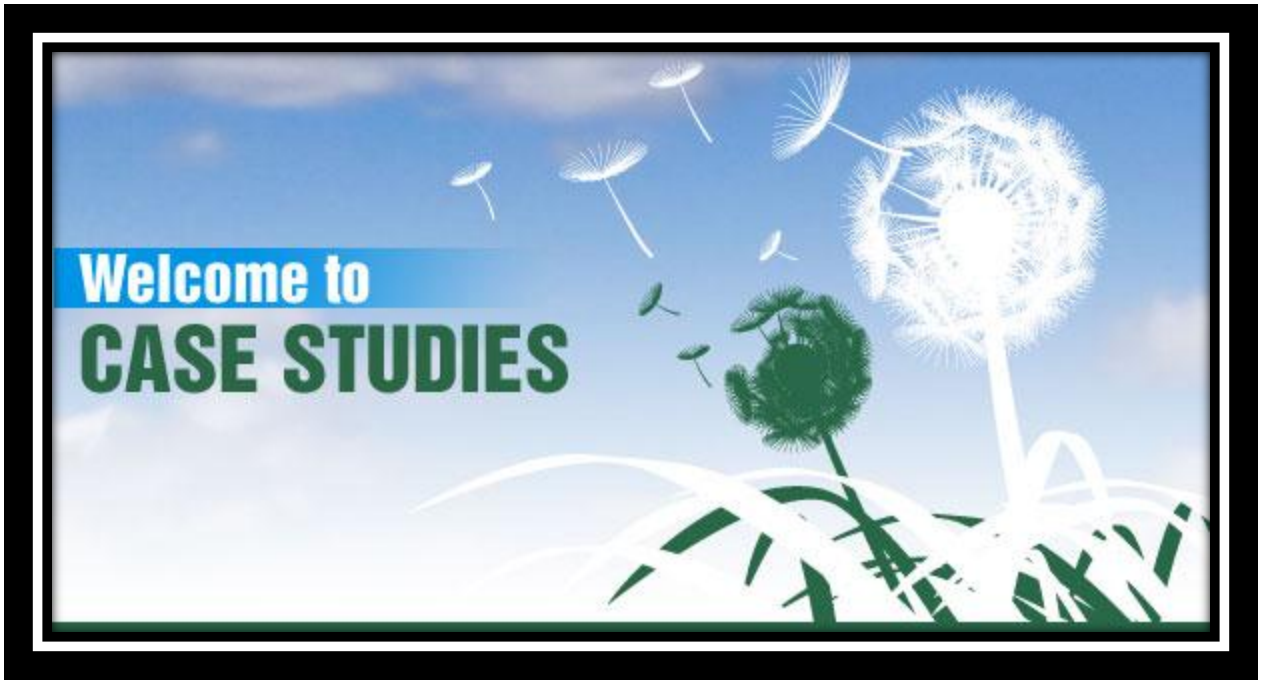
# General Form for Array Traversal (Pascal)

```
procedure traversal(var NamaVarArray:NamaArray);


begin
   for i := 1 to MaksBaris do
   begin
     for j := 1 to MaksKolom do
      {proses traversal yang dipilih}
   end;
   terminasi {sifatnya optional}
end;
```

## Destroy the Array

The process to **return value of array** into **default value** that was given in **array creation**.


Welcome to
CASE STUDIES

# Example of Two Dimensions Array (Algorithm)

```
Algoritma ArrayDasar
{I.S.: Dideklarasikan dua buah array dua dimensi}
{F.S.: Menampilkan isi array}

Kamus:
const
     MaksBaris=5
     MaksKolom=5

type
     bil=array[1..MaksBaris,1..MaksKolom] of integer

matriks1,matriks2:bil
i,j:integer
```

# Example of Two Dimensions Array (Algorithm)

```
Algoritma:
      {input elemen array}
      for i ← 1 to MaksBaris do
         for j ← 1 to MaksKolom do
             input(bil1[i,j])
         endfor
      endfor

      for i ← 1 to MaksBaris do
         for j ← 1 to MaksKolom do
             input(bil2[i,j])
         endfor
      endfor

```

# Example of Two Dimensions Array (Algorithm)

```
{output elemen array}
for i ← 1 to MaksBaris do
   for j ← 1 to MaksKolom do
       output(bil1[i,j])
   endfor
endfor

for i ← 1 to MaksBaris do
   for j ← 1 to MaksKolom do
       output(bil1[i,j])
   endfor
endfor
```

# Example of Two Dimensions Array (Pascal)

```
program ArrayDuaDimensiDasar;
uses crt;

const
     MaksBaris=3;
     MaksKolom=3;
type
     matriks = array[1..MaksBaris,1..MaksKolom] of
integer;

var
     matriks1,matriks2:matriks;
     baris,kolom:integer;
```

# Example of Two Dimensions Array (Pascal)

```
14  begin
15        {input matriks}
16        writeln('Input Matriks Pertama');
17        for baris:=1 to MaksBaris do
18        begin
19           for kolom:=1 to MaksKolom do
20           begin
21                gotoxy(kolom*5+1,baris+3);
22                readln(matriks1[baris,kolom]);
23           end;
24         end;
25
26        writeln();
27        writeln('Input Matriks Kedua');
```

# Example of Two Dimensions Array (Pascal)

```
28      for baris:=1 to MaksBaris do
29      begin
30         for kolom:=1 to MaksKolom do
31         begin
32              gotoxy(kolom*5+1,baris+9);
33              readln(matriks2[baris,kolom]);
34         end;
35      end;
37
38      {output matriks}
39      clrscr();
40      writeln('Output Matriks Pertama');
```

# Example of Two Dimensions Array (Pascal)

```
41      for baris:=1 to MaksBaris do
42      begin
43         for kolom:=1 to MaksKolom do
44         begin
45              gotoxy(kolom*5+1,baris+3);
46              write(matriks1[baris,kolom]);
47         end;
48      end;
49
50      writeln();writeln();
51      writeln('Output Matriks Kedua');
52      for baris:=1 to MaksBaris do
53      begin
```

# Example of Two Dimensions Array (Pascal)

```
54          for kolom:=1 to MaksKolom do
55          begin
56               gotoxy(kolom*5+1,baris+9);
57               write(matriks2[baris,kolom]);
58          end;
59      end;
60
61      writeln();
62      write('Tekan sembarang tombol untuk menutup...');
63      readkey();
64  end.
```

THANK YOU

# GRACIAS

**Contact Person:**
Adam Mukharil Bachtiar
Informatics Engineering UNIKOM
Jalan Dipati Ukur Nomor. 112-114 Bandung 40132
Email: adfbipotter@gmail.com
Blog: http://adfbipotter.wordpress.com

**Copyright © Adam Mukharil Bachtiar 2011**