



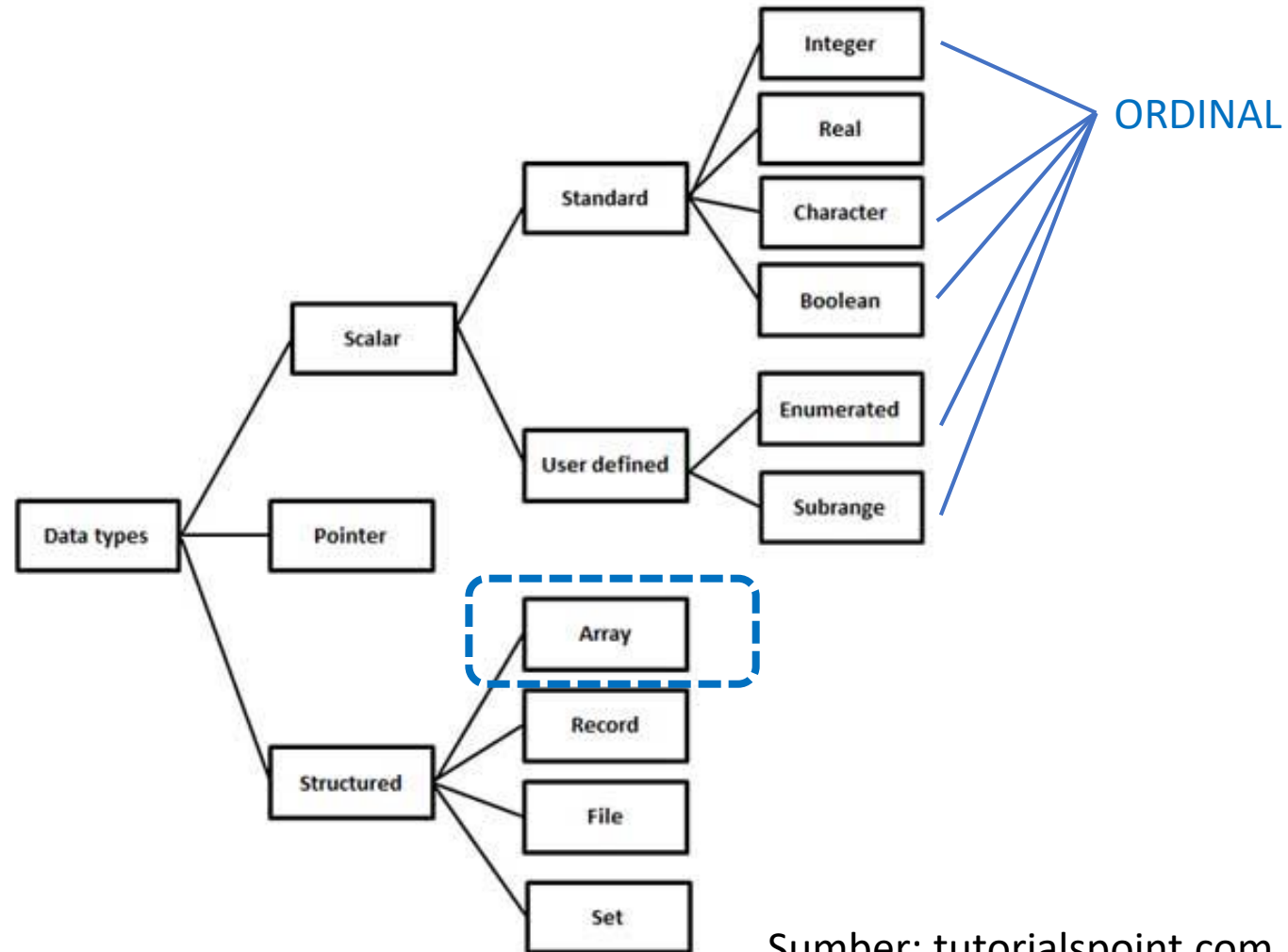
POLITEKNIK STATISTIKA STIS

For Better Official Statistics

Array

Nori Wilantika, S.S.T., M.T.I.





Sumber: tutorialspoint.com

- Pengenalan array
- Array satu dimensi
- Array multidimensi



Pengenalan Array



- Misalkan kita ingin menyimpan nilai dari sejumlah mahasiswa perguruan tinggi

```
VAR
    Nilai1,Nilai2,Nilai3:integer;
BEGIN
    Nilai1 := 80;
    Nilai2 := 90;
    Nilai3 := 100;
END.
```

- Bagaimana jika mahasiswanya berjumlah ratusan atau ribuan?
- Untuk memudahkannya, diperlukan suatu struktur data yang memudahkan dalam mengumpulkan sekumpulan data yang sejenis tersebut.
- Sehingga struktur data yang cocok akan seperti:

```
Nilai [1]
Nilai [2]
Nilai [3]
. . .
```

- Array adalah variabel yang terdiri dari sekumpulan nilai yang **memiliki tipe data yang sama** dan dinyatakan dengan **nama yang sama**.
- Array merupakan konsep yang penting dalam pemrograman, karena array memungkinkan untuk menyimpan data maupun referensi objek dalam jumlah banyak dan terindeks.
- Dalam Pascal, array menggunakan indeks untuk menentukan urutan elemen-elemennya. Contoh: elemen pertamanya dimulai dari indeks 1, elemen kedua memiliki indeks 2, dan seterusnya.

Dengan kata lain :

array adalah variabel yang dapat menampung sejumlah data dengan tipe dan nama sama, tetapi ditandai dengan suatu indeks pada masing-masing elemennya.

- Mendeklarasikan variabel array dengan tipe data yang diinginkan, caranya hampir sama dengan mendeklarasikan variabel biasa.
- Misalnya untuk mendeklarasikan array dengan elemen bertipe integer, dapat dilakukan dengan cara :

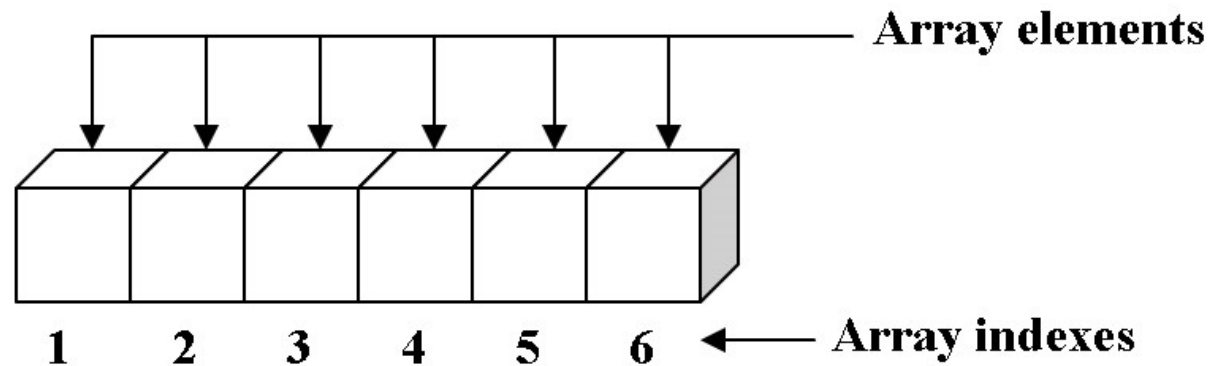
```
var x : array [1..10] of integer;
```

Diagram illustrating the declaration of an array variable:

- `x` is labeled as **Nama Array** (Name of Array).
- `[1..10]` is labeled as **Panjang Array (Indeks)** (Length of Array (Index)).
- `integer` is labeled as **Tipe Elemen** (Element Type).

- Dengan adanya pendefinisian panjang array, berarti array memiliki ukuran yang tetap dalam arti tidak dapat membesar atau mengecil ukurannya setelah didefinisikan.
- Setelah didefinisikan, maka variabel dengan nama array yang telah dideklarasikan dapat menyimpan sebanyak nilai yang berada dalam [].

- Contoh yang sebelumnya, berarti terdapat 10 angka x yang bertipe integer yang dapat diakses melalui indeks 1 s.d 10.
- Pada masing-masing elemen x hanya menampung satu nilai integer.



- Setelah pendefinisian array, maka memori akan dialokasikan untuk menyimpan data dari array. Besar memori yang dialokasikan tergantung dari tipe data variabel array dan jumlah elemen array yang didefinisikan.
- Elemen array bisa bertipe data apa saja. Namun indeks Array hanya bisa diisi oleh tipe data ordinal. **Real tidak bisa dijadikan indeks Array.**


```
Var Array1 : array [1..10] of integer;
```

```
Var Array2 : array [-20..8] of real;
```

```
Var Array3 : array ['a' .. 'z'] of char; {character}
```

```
ch_array : array[char] of 1..26;
```

```
Var NilaiHuruf : array[1..5] of char; {subrange}
```

```
Var X : array[byte] of char; {0 sampai dengan 255}
```

```
Var Jumlah : array[(Jan, Feb, Mar, Apr, Mei, Jun)] of real; {enum}
```

```
color = ( red, black, blue, silver, beige);
```

```
car_color = array [color] of boolean;
```

- Suatu array tidak hanya dapat dideklarasikan sebagai variabel, tetapi juga dapat berupa suatu konstanta yang dideklarasikan di bagian deklarasi konstanta.

- Contoh:

Const

```
X : array[1..5] of integer = (6,25,375,5,2)
```

```
Xbyte : array[1..3] of byte = (6,25,5);
```

```
XWord : array[1..3] of word = (500,2500,5000);
```

```
XShortint : array[1..3] of ShortInt = (-100,-5,25);
```

```
XLongint : array[1..3] of LongInt = (100000,-5000,1000000);
```

```
XReal : array[1..3] of real = (18.2,85.98,82.10);
```

```
Nilai : array[1..3] of boolean = (True,False,False)
```

```
Bulan: array[1..12] of string[3]= ('Jan', 'Feb', 'Mar', 'Apr', 'Mei',  
'Jun', 'Jul','Agt', 'Sep', Okt', 'Nov', 'Des');
```

Misalkan terdapat pendeklarasian sebagai berikut

TYPE

Range = 1..7;

Color = (Red, Green, Blue);

int = INTEGER;

Tentukan mana pendefinisian pada VAR yang tidak benar

- a) MyArray :ARRAY[Range];
- b) HisArray :ARRAY[1..Range] OF INTEGER;
- c) YourArray :ARRAY[Color] OF INT;
- d) HerArray :ARRAY[1..Color] OF INTEGER;
- e) Score :ARRAY[1..5, Range] OF INTEGER;

- Untuk mengakses setiap elemen array, kita menggunakan indeksnya.
- Contoh:

```
var X: array [1..10] of integer;
```

```
...
```

```
X[1] := 10;
```

```
X[2] := X[1] - 5;
```

```
X[3] := X[2] + X[1];
```

```
Writeln(X[3]); {output = 15}
```

```
X[11] := ??; {error atau 0?}
```

```
...
```

- **Ingat! Indeks Array tidak selalu angka.**

```
program exArrays;
uses crt;
var
    (* n is an array of 10 integers *)
    n: array ['a'..'j'] of integer;
    i, j: integer;

begin
    clrscr;

    n['a'] := 1;
    writeln(n['a']);
end.
```

```

program tipe_array;
uses crt;
var
    nilai: array[0..9] of integer;

begin
    clrscr;

    nilai[0] := 23;
    nilai[1] := 13;
    nilai[2] := 98;
    nilai[3] := 106;

    writeln('nilai0: ', nilai[0]);
    writeln('nilai1: ', nilai[1]);
    writeln('nilai2: ', nilai[2]);
    writeln('nilai3: ', nilai[3]);

    readln;
end.

```

Inisialisasi tidak harus secara berurutan

```

1  program tipe_array;
2  uses crt;
3  var
4      nilai: array[0..9] of integer;
5
6      begin
7          clrscr;
8
9          nilai[3] := 23;
10         nilai[9] := 13;
11         nilai[2] := 98;
12         nilai[0] := 106;
13
14         writeln('nilai3: ', nilai[3]);
15         writeln('nilai9: ', nilai[9]);
16         writeln('nilai2: ', nilai[2]);
17         writeln('nilai0: ', nilai[0]);
18
19         writeln('nilai1: ', nilai[1]);
20         writeln('nilai7: ', nilai[7]);
21
22         readln;
23     end.

```

```
program exArrays;
var
    (* n is an array of 10 integers *)
    n: array [1..10] of integer;
    i, j: integer;

begin
    (* initialize elements of array n to 0 *)
    for i := 1 to 10 do
        n[ i ] := i + 100;

    (* output each array element's value *)
    for j:= 1 to 10 do
        writeln('Element[' , j , ']' = ' , n[j] );
    end.
```

OUTPUT

```
Element[1] = 101
Element[2] = 102
Element[3] = 103
Element[4] = 104
Element[5] = 105
Element[6] = 106
Element[7] = 107
Element[8] = 108
Element[9] = 109
Element[10] = 110
```

ARRAY LOOP (2)

```
Uses Wincrt;  
Var  
    x : array [1..100] of integer;  
    n,i :integer;  
  
Begin  
    Writeln('Contoh 1 Program Array');  
    Writeln('=====');  
    Writeln;  
    Write('Silahkan Input Banyak Data: ');  
    readln(n);  
    Writeln;  
  
    For i:= 1 to n do Readln(x[i]);  
  
    Writeln;  
    Write('Data Yang Telah Dimasukkan: ');  
  
    For i:= 1 to n do Write(x[i], ' ');  
End.
```

Program akan menerima inputan dari user, yang kemudian inputan user tersebut akan disimpan dalam bentuk array **x**.

Proses terakhir, program akan menampilkan isi dari array **x**.

- Suatu string dianggap sebagai suatu array tipe Char dengan indeks **mulai dari 0** sampai dengan panjang dari string tersebut. Misalnya nilai string berikut:

`X := 'ABCD' ;`

- Maka string X bisa dipandang sebagai suatu array bertipe Char, sehingga:

`X[1] = 'A'`

`X[2] = 'B'`

`X[3] = 'C'`

`X[4] = 'D'`

- Indeks ke-0 dari nilai string ini berisi nilai string yang menunjukkan panjang dari stringnya.

`X[0] → #4 ;`

- Pada fungsi, mentransfer array sebagai parameter akan menghasilkan error.
- Solusinya, kita perlu mendeklarasikan Array sebagai tipe data bentukan (*user-defined*), dan kemudian menggunakan tipe itu sebagai parameter untuk fungsi.

```

program arrayToFunction;
const
    size = 5;
type
    a = array [1..size] of integer;
var
    balance: a = (1000, 2, 3, 17, 50);
    average: real;

function avg(arr: a) : real;
var
    i :1..size;
    sum: integer;

begin
    sum := 0;
    for i := 1 to size do sum := sum + arr[i];
    avg := sum / size;
end;

begin
    (* Passing the array to the function *)
    average := avg( balance ) ;

    (* output the returned value *)
    writeln( 'Average value is: ', average:7:2);
end.
    
```

deklarasi array sebagai tipe data baru dengan nama 'a'

Inisialisasi Array

informasi panjang array

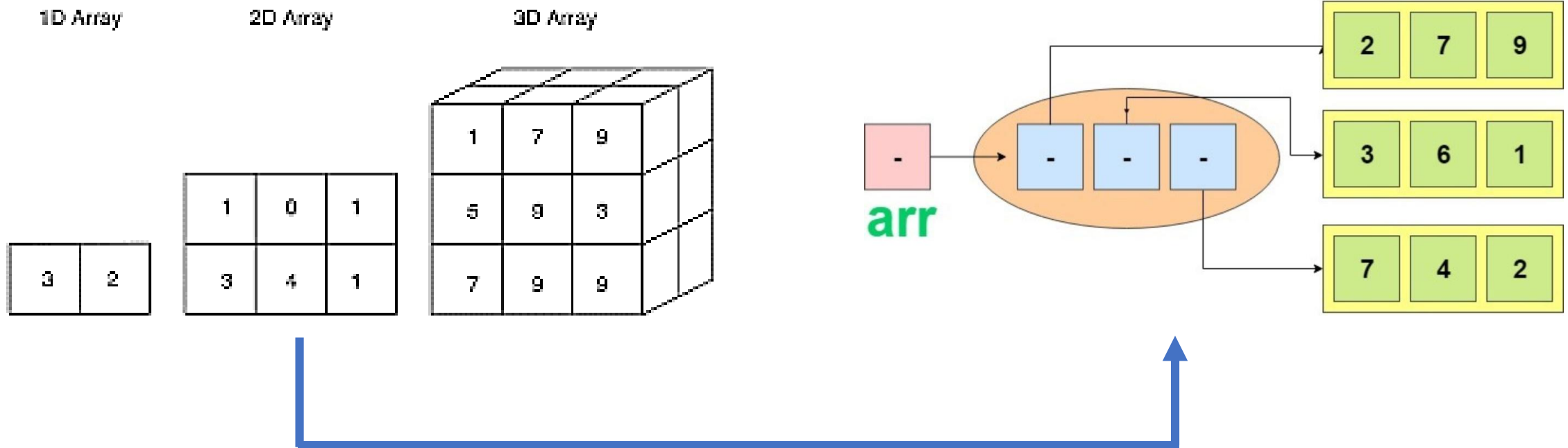
Passing Array



ARRAY MULTIDIMENSI



- Array yang telah kita pelajari pada pembahasan sebelumnya merupakan array satu dimensi.
- Indeks atau subscript pada array jumlahnya hanya sebuah, sehingga array demikian disebut sebagai array satu dimensi.
- Seringkali dibutuhkan suatu susunan struktur penyimpanan data dengan mendefinisikan suatu array dalam array. Hal ini dinamakan sebagai **array multidimensi**.
- Jika jumlah indeks atau subscriptnya lebih dari satu maka array tersebut disebut sebagai **array multidimensi**, yang bisa saja berdimensi dua (array dua dimensi), berdimensi tiga (array tiga dimensi), dan seterusnya.



Array berdimensi dua merepresentasikan suatu bentuk tabel atau matriks, dimana indeks atau subscript yang pertama menunjukkan baris, dan indeks yang kedua menunjukkan kolom dari tabel atau matriks.

- Kita ingin mendefinisikan sebuah array untuk menyimpan data setiap 1000 mahasiswa dalam beberapa kurun waktu 10 tahun terakhir.

Dalam hal ini kita dapat mendefinisikan array seperti ini:

```
mhs : array [1..10] of array [1..1000] of integer;
```

atau

```
mhs: array [1..10, 1..1000] of integer;
```

Indeks
Pertama

Indeks
Kedua

- Di dalam komputer data mahasiswa tersebut akan disimpan dengan urutan sebagai berikut :

```
mhs [1,1] mhs [1,2] . . . mhs[1,1000] mhs[2,1] . . . mhs [10, 1000]
```

atau $mhs[i,j]$ menyatakan mahasiswa pada tahun ke i urutan ke j .


```
var
  a: array [0..3, 0..3] of integer;
  i, j : integer;

begin
  for i:= 0 to 3 do
    for j:= 0 to 3 do
      a[i,j]:= i * j;
    ...
  end;
```

Menggunakan
nested loop

terdapat 2 indeks

- Berikut adalah contoh kode untuk mencetak data mahasiswa dalam 10 tahun terakhir.

```
for i := 1 to 10 do  
  begin  
    for j:=1 to 1000 do write(mhs[i,j], ' ');  
  end;
```

	0	1	2
0	(0,0)	(0,1)	(0,2)
1	(1,0)	(1,1)	(1,2)
2	(2,0)	(2,1)	(2,2)

Column Index

Row Index

- Contoh lainnya:

Var

```
Tabel : array[1..3,1..2] of byte;
```

- Cara mengakses elemen Tabel:

```
Tabel[1,1] := 5;
```

```
Tabel[1,2] := 25;
```

```
Tabel[2,1] := 200;
```

```
Tabel[2,2] := 22;
```

```
Tabel[3,1] := 75;
```

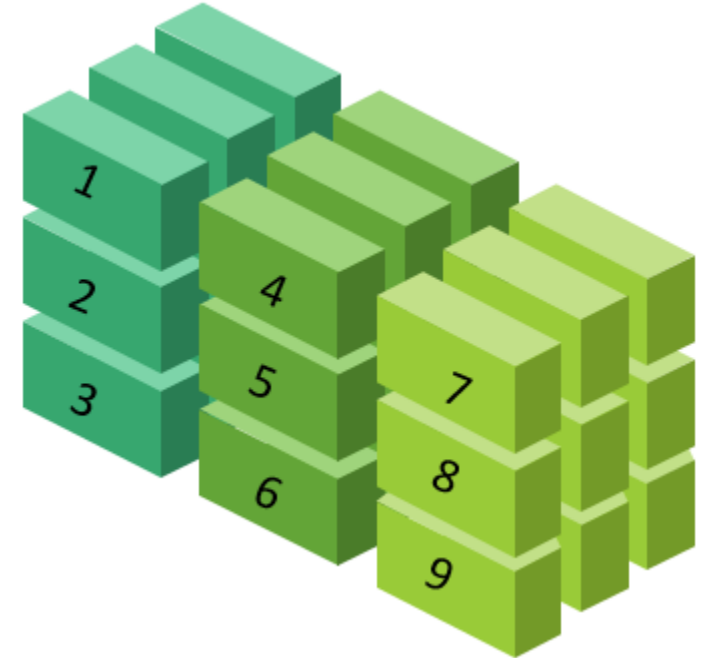
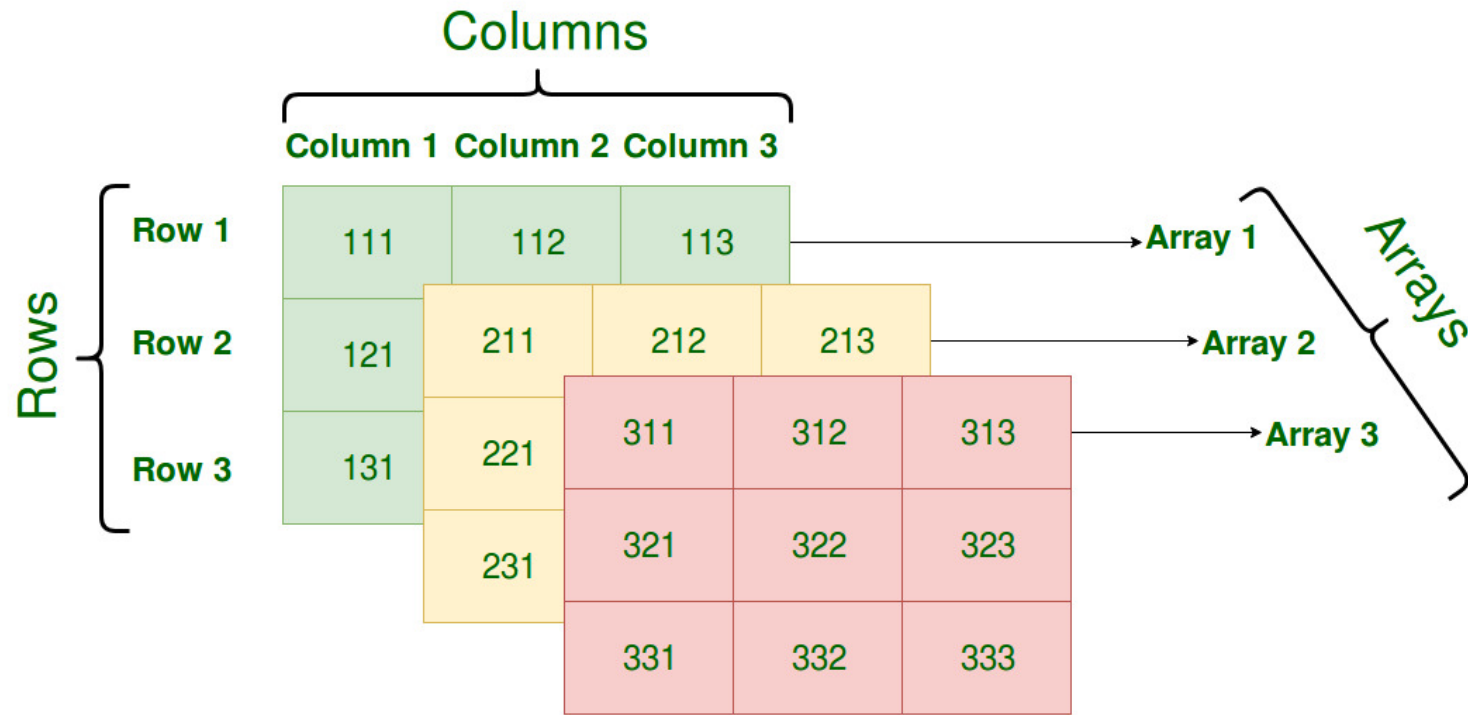
```
Tabel[3,2] := 50
```

```
program ex2dimarray;
var
  a: array [0..3, 0..3] of integer;
  i,j : integer;
begin
  for i:=0 to 3 do
    for j:=0 to 3 do
      a[i,j]:= i * j;
    end;
  for i:=0 to 3 do
  begin
    for j:=0 to 3 do write(a[i,j]:2,' ');
    writeln;
  end;
end.
```

OUTPUT

```
0 0 0 0
0 1 2 3
0 2 4 6
0 3 6 9
```

Secara prinsip, array berdimensi lebih dari dua penanganannya sama saja dengan array berdimensi dua.



```
program tipe_array;  
uses crt;  
var  
    nilai: array[0..1,0..2,0..3] of integer;  
begin  
    clrscr;  
  
    nilai[0,0,2]:= 2;  
    writeln(nilai[0,0,2]);  
  
    nilai[1,2,3]:= 999;  
    writeln(nilai[1,2,3]);  
  
    readln;  
end.
```

OUTPUT

2

999



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

