



POLITEKNIK MASAMY INTERNASIONAL

SK Menristekdikti RI Nomor: 731/KPT/I/2018

Jalan Ikan Paus No.10-15 Kertosari Banyuwangi - 68411
Telp (0333) 3384593 – <http://polmain.info>

PROGRAM STUDI D3 TEKNIK KOMPUTER

Form:
B.Ak/eva/04/20

Nama Dosen : *Arif Fahmi, S.T.,M.T.*

Mata Kuliah : Pemrograman Berorientasi Object

Semester : 4 (Empat)

Kode Mata Kuliah : TKV4044

Th. Akdm : 2019/2020.

BAB

ARRAY PADA PEMROGRAMAN JAVA

SUB BAB :

1. Pengenalan Array
2. Pendeklarasian Array
3. Pengaksesan Array
4. Panjang Array
5. Dimensi Array

TUJUAN MATERI

1. Mahasiswa dapat memahami pengertian Array pada Java
2. Mahasiswa dapat mengetahui bentuk umum dari Array
3. Mahasiswa dapat mengetahui jenis-jenis Array pada Java
4. Mahasiswa dapat mendeklarasikan dan membuat program menggunakan Array pada Java

REFERENSI

1. J.Eck, D. (2006). *Introduction to Programming Using Java*. Geneva.
2. Ady Wicaksono, Dasar – Dasar Pemrograman Java 2, Penerbit PT Elex Media Komputindo, Jakarta, 2002

ARRAY PEMROGRAMAN JAVA

1. Pengenalan Array

Pada Bab sebelumnya, kita telah mendiskusikan bagaimana cara pendeklarasian berbagai macam variabel dengan menggunakan tipe data primitif. Dalam pendeklarasian variabel, kita sering menggunakan sebuah tipe data beserta nama variabel atau *identifier* yang unik, dimana untuk menggunakan variabel tersebut, kita akan memanggil dengan nama *identifier*-nya.

Sebagai contoh, kita memiliki tiga variabel dengan tipe data int yang memiliki *identifier* yang berbeda untuk tiap variabel.

```
int number1;  
int number2;  
int number3;  
  
number1 = 1;  
number2 = 2;  
number3 = 3;
```

Seperti yang dapat Anda perhatikan pada contoh diatas, hanya untuk menginisialisasi dan menggunakan variabel terutama pada saat variabel-variabel tersebut memiliki tujuan yang sama, dirasa sangat membingungkan. Di Java maupun di bahasa pemrograman yang lain, mereka memiliki kemampuan untuk menggunakan satu variabel yang dapat menyimpan sebuah data list dan kemudian memanipulasinya dengan lebih efektif. Tipe variabel inilah yang disebut sebagai **array**.

| | | | |
|---------|---|---|---|
| | 0 | 1 | 2 |
| number: | 1 | 2 | 3 |

Sebuah array akan menyimpan beberapa item data yang memiliki tipe data sama didalam sebuah blok memori yang berdekatan yang kemudian dibagi menjadi beberapa slot. Bayangkanlah array adalah sebuah variabel - sebuah lokasi memori tertentu yang memiliki satu nama sebagai *identifier*, akan tetapi ia dapat menyimpan lebih dari sebuah value.

2. Pendeklarasian Array

Array harus dideklarasikan seperti layaknya sebuah variabel. Apabila Anda mendeklarasikan array, Anda harus membuat sebuah list dari tipe data, yang diikuti oleh tanda kurung buka dan kurung tutup, yang diikuti oleh nama *identifier*. Sebagai contoh,

```
int []ages;
```

atau Anda dapat menempatkan kurung buka dan kurung tutupnya setelah *identifier*. Sebagai contoh,

```
int ages[];
```

Setelah pendeklarasian, kita harus membuat array dan menentukan berapa panjangnya dengan sebuah **konstruktor**. Proses ini di Java disebut sebagai *instantiation* (Kata dalam Java yang berarti membuat). Untuk meng-*instantiate* sebuah obyek, kita membutuhkan sebuah konstruktor. Kita akan membicarakan lagi mengenai *instantiate* obyek dan pembuatan

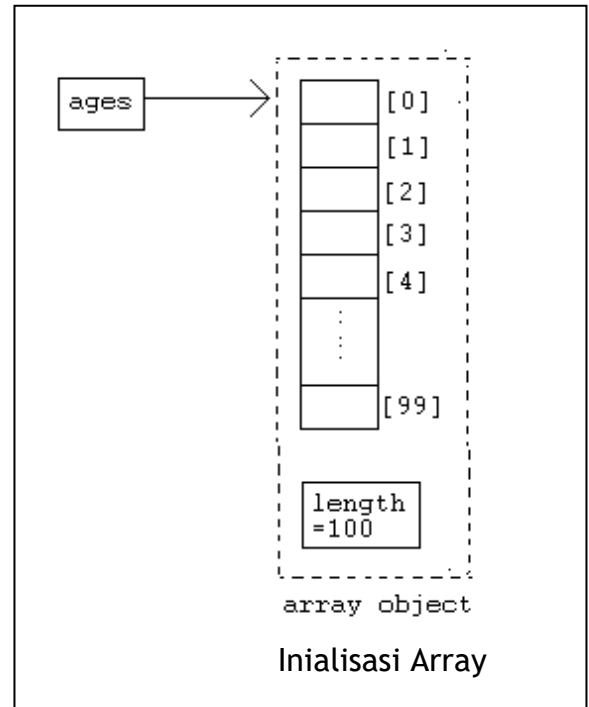
konstruktor pada bagian selanjutnya. Perlu dicatat, bahwa ukuran dari array tidak dapat diubah setelah Anda menginisialisasinya. Sebagai contoh,

```
//deklarasi  
  
int ages[];  
  
//instantiate obyek  
ages = new int[100];
```

atau bisa juga ditulis dengan,

```
//deklarasi dan instantiate obyek  
int ages[] = new  
int[100];
```

Pada contoh diatas, deklarasi akan memberitahukan kepada compiler Java, bahwa identifier **ages** akan digunakan sebagai nama array yang berisi data-data integer, dan kemudian untuk membuat atau meng-instantiate sebuah array baru yang terdiri dari 100 elemen. Selain menggunakan sebuah keyword baru untuk meng-instantiate array, Anda juga dapat secara otomatis mendeklarasikan array, membangunnya, kemudian memberikan sebuah value. Sebagai contoh,



```
//membuat sebuah array yang berisi variabel-variabel //boolean pada sebuah  
identifier. Array ini terdiri dari 4 //elemen yang diinisialisasikan sebagai  
value //{true,false,true,false} boolean results[] = { true, false, true,  
false };  
  
//Membuat sebuah array yang terdiri dari penginisialisasian //4 variabel  
double bagi value {100,90,80,75} double []grades = {100, 90, 80, 75};  
  
//Membuat sebuah array String dengan identifier days. Array //ini terdiri  
dari 7 elemen.  
String days[] = { "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"};
```

3. Pengaksesan Array

Untuk mengakses sebuah elemen dalam array, atau mengakses sebagian dari array, Anda harus menggunakan sebuah nomor atau yang disebut sebagai **index** atau subscript. Sebuah **nomor index** atau **subscript** telah diberikan kepada tiap anggota array, sehingga program dan programmer dapat mengakses setiap value apabila dibutuhkan. Index **selalu** dalam integer. Dimulai dari nol, kemudian akan terus bertambah sampai list value dari array tersebut berakhir. Perlu dicatat, bahwa elemen-elemen didalam array dimulai dari 0 sampai dengan (**ukuranArray-1**).

Sebagai contoh, pada array yang kita deklarasikan tadi, kita mempunyai,

```
//memberikan nilai 10 kepada elemen pertama  
array ages[0] = 10;  
  
//mencetak elemen array yang terakhir  
System.out.print(ages[99]);
```

Perlu diperhatikan bahwa sekali array dideklarasikan dan dikonstruksi, nilai yang disimpan dalam setiap anggota array akan diinisialisasi sebagai nol. Oleh karena itu, apabila Anda menggunakan tipe data reference seperti String, ia tidak akan diinisialisasi ke string kosong "", sehingga Anda tetap harus membuat String array secara eksplisit.

Berikut ini adalah contoh, bagaimana untuk mencetak seluruh elemen didalam array. Dalam contoh ini digunakanlah loop, sehingga kode kita menjadi lebih pendek.

```
public class ArraySample{
    public static void main( String[] args ){
        int[] ages = new int[100];
        for( int i=0; i<100; i++ ){
            System.out.print( ages[i] );
        }
    }
}
```

Petunjuk penulisan program:

Biasanya, lebih baik menginisialisasi atau meng-instantiate array setelah Anda mendeklarasikannya. Sebagai contoh pendeklarasiannya

```
int []arr = new int[100];
lebih disarankan daripada,
int []arr;
arr = new int[100];
```

Elemen-elemen dalam n-elemen array memiliki index dari 0 sampai n-1. Perhatikan disini bahwa tidak ada elemen array arr[n]. Hal ini akan menyebabkan array-index out-of-bounds exception. Anda tidak dapat mengubah ukuran dari sebuah array

4. Panjang Array

Untuk mengetahui berapa banyak element didalam sebuah array, Anda dapat menggunakan **length (panjang)** field dalam array. Panjang field dalam array akan mengembalikan ukuran dari array itu sendiri. Sebagai contoh,

```
arrayName.length
```

Pada contoh sebelumnya, kita dapat menuliskannya kembali seperti berikut ini,

```
public class ArraySample
{
    public static void main( String[] args ){
        int[] ages = new int[100];
        for( int i=0; i<ages.length; i++ ){
            System.out.print( ages[i] );
        }
    }
}
```

5. Dimensi Array

A. Array Dimensi Satu

Variabel Larik atau lebih dikenal dengan array adalah tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama. Sebelum digunakan, variabel array perlu dideklarasikan terlebih dahulu. Cara mendeklarasikan variabel array sama seperti deklarasi variabel yang lainnya, hanya saja diikuti oleh suatu indek yang menunjukkan jumlah maksimum data yang disediakan. Bentuk Umum pendeklarasian array;

```
tipe_data[] nama_var_array;  
nama_var_array = new  
tipe_data[ukuran];
```

Atau bisa juga secara langsung seperti dibawah ini :

```
tipe_data[]nama_var_array = new tipe_data[ukuran];
```

Keterangan :

Type Data : Untuk menyatakan type data yang digunakan.

Ukuran : Untuk menyatakan jumlah maksimum elemen array.

Berikut contoh pendeklarasian array dan membuat objek array;

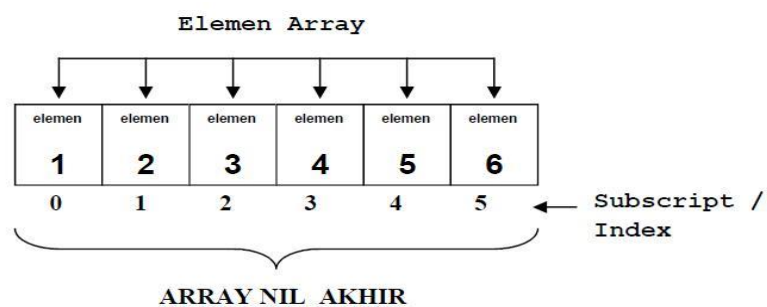
```
int[] nil_akhir;  
nil_akhir = new int[6];
```

Contoh Pendeklarasian Array sekaligus membuat objek array,

```
int[] nil_akhir = new int[6];
```

Diagram showing the components of the array declaration:
- **int[]**: Tipe data elemen array (indicated by an arrow)
- **nil_akhir**: Nama Array (indicated by an arrow)
- **new int[6]**: Jumlah Elemen Array (indicated by an arrow)

Pada pendeklarasian diatas, variabel nil_akhir sebagai array-of-int, banyak elemen yang dapat ditampung sebanyak 6 (enam) elemen. Suatu array dapat digambarkan sebagai kotak panjang yang berisi kotak-kotak kecil didalam kotak panjang tersebut.



Suatu array dapat diakses dengan menggunakan subscript atau indexnya. Bentuk umum pengaksesan dengan bentuk :

```
nama_var_array[nomor_index_array];
```

Contoh :

```
Nil_Akhir[3];  
Nil_Akhir[1];
```

Sebagai contoh program memasukan data kedalam elemen array dan mengambil data dari dalam elemen array sebagai berikut :

```
public class LatArray1 {
    public static void main(String[] args)
    {
        int i;
        int[] nil_akhir;
        // deklarasi variabel array
        nil_akhir = new int[6]; // membuat objek array
        Scanner input = new Scanner(System.in);
        for(i=0; i<6; i++)
        {
            System.out.print("Masukkan Array ke " + i + " = ");
            nil_akhir[i] = input.nextInt();
        }
        System.out.println("\n\nData Yang Diinput ke Elemen Array \n");
        for(i=0; i<6; i++)
        {
            System.out.print("Nilai Akhir Index " + i );
            System.out.println(" = " + nil_akhir[i]);
        }
    }
}
```

Inisialisasi adalah memberikan nilai awal terhadap suatu variabel. Bentuk pendefinisian suatu array dapat dilihat dari contoh berikut :

```
tipe_data[] nama_array = { nilai array };
```

Contoh:

```
float[] nilai= {56.5, 66.7, 87.45, 98,5, 78.9 };
```

Sebagai contoh program memasukan data kedalam elemen array dengan cara inisialisasi objek array dan mengambil data dari dalam elemen array, secara selengkapnya seperti dibawah ini:

```
public class InisialisasiArray1 {
    public static void main(String[] args)
    {
        int i;
        double[] nil_akhir = {56.5, 66.7, 87.6, 98,5, 78.9, 85.4};

        System.out.println("\nData Yang Diinput ke Elemen Array \n");
        //menampilkan data dari elemen array
        for(i=0; i<6; i++)
        {
            System.out.print("Nilai Akhir Index " + i );
            System.out.println(" = " + nil_akhir[i]);
        }
    }
}
```

B. Array Dimensi Dua

Array dimensi dua tersusun dalam bentuk baris dan kolom, dimana indeks pertama menunjukan baris dan indeks kedua menunjukan kolom.

Deklarasi Array Bentuk Umum pendeklarasian array :

```
tipe_data[][] nama_var_array;

nama_var_array = new
tipe_data[ukuran_baris][ukuran_kolom];
```

Atau bisa juga secara langsung seperti dibawah ini :

```
tipe_data[][] nama_var_array = new tipe_data[ukuran_baris][ukuran_kolom];
```

Keterangan :

Type Data : Untuk menyatakan type data yang digunakan.

Ukuran_baris : Untuk menyatakan jumlah maksimum elemen baris dalam array.

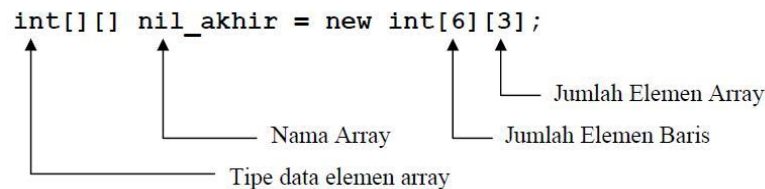
Ukuran_kolom : Untuk menyatakan jumlah maksimum elemen kolom dalam array.

Berikut contoh pendeklarasian array dan membuat objek array;

```
int[][] nil_akhir;  
nil_akhir = new int[6][3];
```

Contoh Pendeklarasian Array sekaligus membuat objek array.

```
int[][] nil_akhir = new int[6][3];
```



Tipe data elemen array

Nama Array

Jumlah Elemen Baris

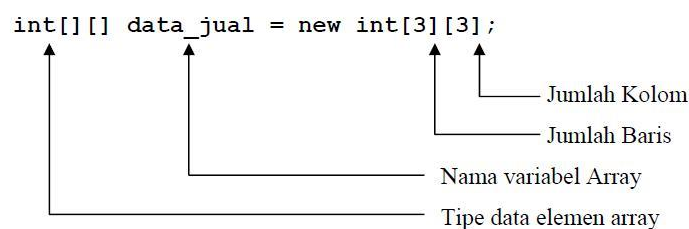
Jumlah Elemen Array

Sebagai contoh pendeklarasian dan pengaksesan data, kita gunakan adalah pengolahan data penjualan, berikut dapat anda lihat pada tabel berikut :

| No | Tahun Penjualan | | |
|----|-----------------|------|------|
| | 2001 | 2002 | 2003 |
| 1 | 150 | 159 | 230 |
| 2 | 100 | 125 | 150 |
| 3 | 210 | 125 | 156 |

Jika anda lihat dari tabel diatas maka dapat dituliskan kedalam array dimensi dua berikut :

```
int[][] data_jual = new int[3][3];
```



Tipe data elemen array

Nama variabel Array

Jumlah Baris

Jumlah Kolom

Sebagai contoh program memasukan data kedalam elemen array berdimensi dua dan mengambil data dari dalam elemen array berdimensi dua, secara selengkapnya seperti dibawah ini:

```
public class LatArray2 {  
    public static void main(String[] args)  
    {  
        int i, j;  
        int[][] data_jual;  
        data_jual = new int[3][3];  
        Scanner input = new Scanner(System.in);  
        for(i=0; i<3; i++)  
        {  
            for(j=0; j<3; j++)  
            {  
                System.out.print("Masukkan Data Jual ke ["+i+""]["+j+"] =");  
                data_jual[i][j] = input.nextInt();  
            }  
            System.out.println();  
        }  
        System.out.println("\n\nData Jual Yang Diinput ke Elemen Array \n");  
        for(i=0; i<3; i++)  
        {  
            for(j=0; j<3; j++)  
            {  
                System.out.print("Nilai Data Jual ["+i+""]["+j+"] ");  
                System.out.println(" = " + data_jual[i][j]);  
            }  
            System.out.println();  
        }  
    }  
}
```


Suatu array, dapat diakses dengan menggunakan subscript atau indexnya. Bentuk umum pengaksesan dengan bentuk :

```
type_data[][] nama_array = { nilai array };
```

Contoh:

```
int[][] nil_akhir = {{150, 159, 230}, {100,125,150}, {210,125,156}};
```

Sebagai contoh program memasukan data kedalam elemen array dengan cara inisialisasi objek array dan mengambil data dari dalam elemen array, secara selengkapnya seperti dibawah ini:

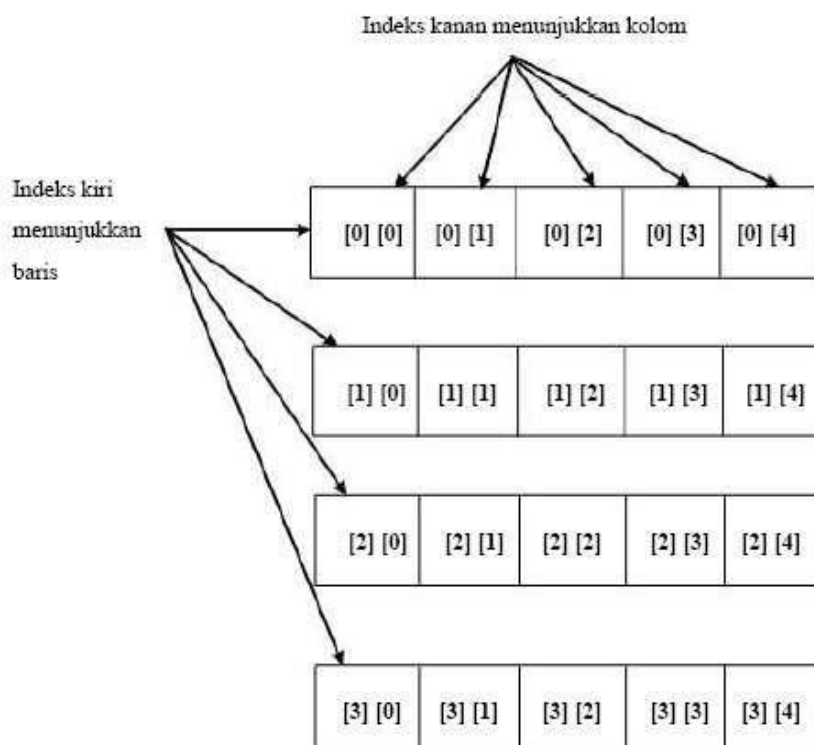
```
public class InisialisasiArray2 {  
    public static void main(String[] args)  
    {  
        int i, j;  
        int[][] nil_akhir = {{150, 159, 230}, {100,125,150}, {210,125,156}};  
  
        System.out.println("Data Yang Diinput ke Elemen Array \n");  
        //menampilkan data dari elemen array  
        for(i=0; i<3; i++)  
        {  
            for(j=0; j<3; j++)  
            {  
                System.out.print("Nilai Akhir Index ["+i+"] ["+j+"] = ");  
                System.out.println(" = " + nil_akhir[i][j]);  
            }  
        }  
    }  
}
```

C. Array Multidimensi

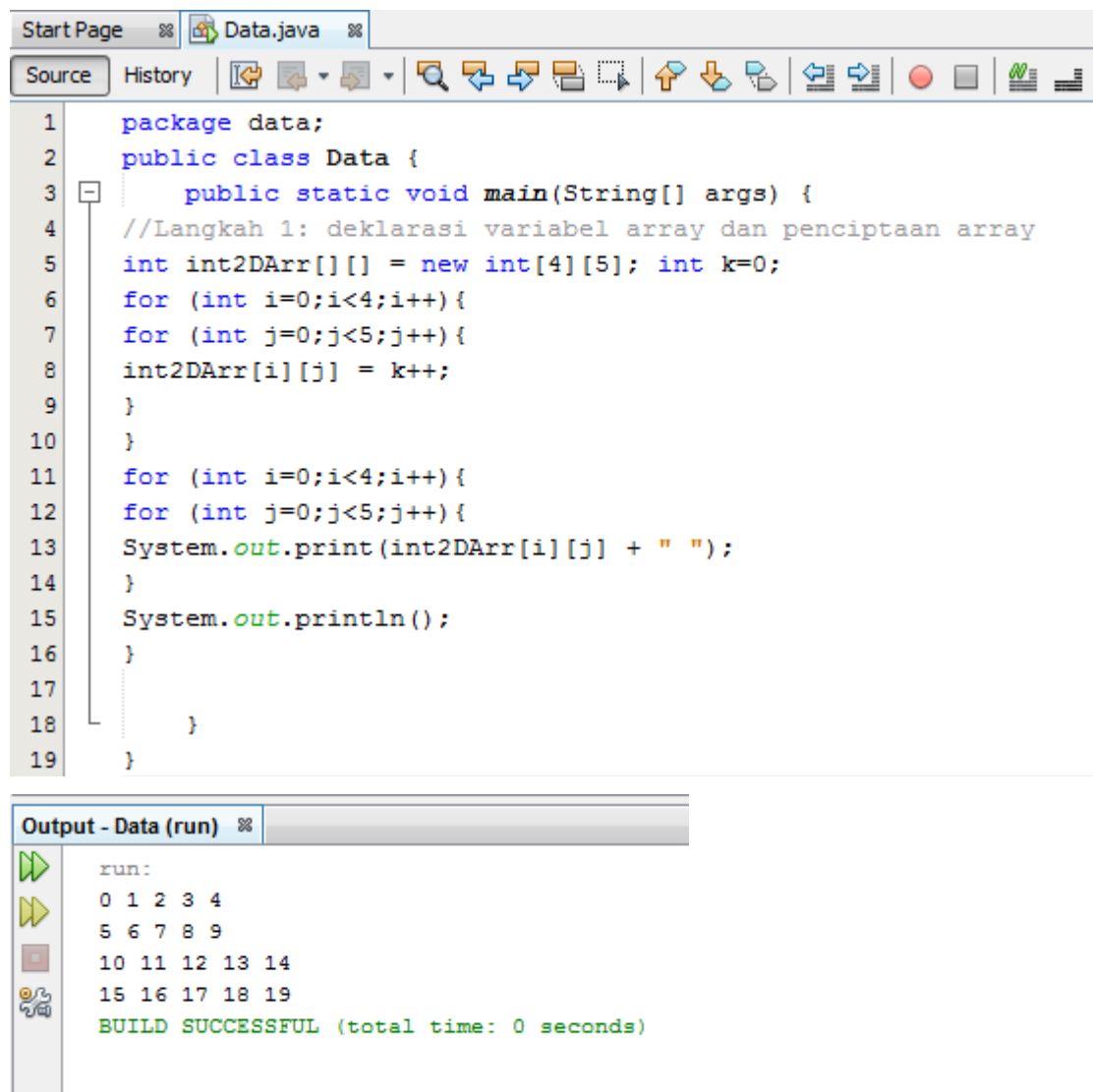
Pada pemrograman Java, *array* multidimensi sesungguhnya *array* dari *array*. Untuk mendeklarasikan variabel *array*, menspesifikasikan masing-masing indeks menggunakan himpunan kurung siku yang lain. *Array* 2 dimensi bertipe *int* bernama *intArr* dideklarasikan dan diciptakan dengan perintah sebagai berikut:

```
int int2DArr[][] = new() int [4][5];
```

Perintah ini mendeklarasikan dan mengalokasikan *array* 4 kali 5 dan memberikan ke variabel *int2DArr*. Secara internal matriks ini diimplementasikan sebagai *array* dari *array* bertipe *int*. Secara konseptual *array* ini digambarkan sebagai berikut:



Berikut contoh listing program yang memberikan angka ke masing-masing elemen di *array* dari kiri ke kanan, puncak ke bawah, kemudian menampilkan masing-masing elemen:



```
1 package data;
2 public class Data {
3     public static void main(String[] args) {
4         //Langkah 1: deklarasi variabel array dan penciptaan array
5         int int2DArr[][] = new int[4][5]; int k=0;
6         for (int i=0;i<4;i++){
7             for (int j=0;j<5;j++){
8                 int2DArr[i][j] = k++;
9             }
10        }
11        for (int i=0;i<4;i++){
12            for (int j=0;j<5;j++){
13                System.out.print(int2DArr[i][j] + " ");
14            }
15            System.out.println();
16        }
17    }
18 }
19 }
```

Output - Data (run) %

```
run:
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 18 19
BUILD SUCCESSFUL (total time: 0 seconds)
```