

# BAB 7

## Java Array

### 7.1 Tujuan

Dalam bab ini, kita akan mendiskusikan mengenai array dalam Java. Pertama, kita akan mendefinisikan apa yang dimaksud dengan array, kemudian kita juga akan mendiskusikan bagaimana mendeklarasikannya dan menggunakannya dalam Java.

Pada akhir pembahasan, diharapkan pembaca dapat :

- Mendeklarasikan dan membuat array
- Mengakses elemen-elemen didalam array
- Menentukan jumlah element didalam sebuah array
- Mendeklarasikan dan membuat array multidimensi

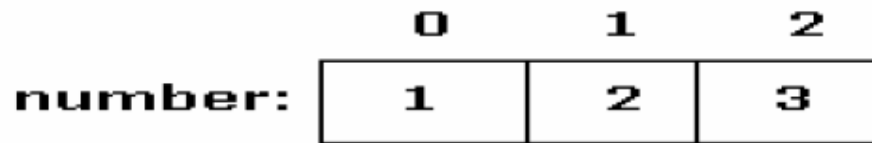
### 7.2 Pengenalan Array

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. Apabila kita ingin menggunakan variabel tersebut, kita akan memanggil dengan nama *identifier*-nya.

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

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

Seperti yang dapat Anda perhatikan pada contoh diatas, kode tersebut akan sia-sia karena harus menginisialisasi dan menggunakan setiap variabel padahal sebenarnya variabel-variabel tersebut digunakan untuk tujuan yang sama. Pada bahasa pemrograman Java maupun di bahasa pemrograman yang lain, terdapat sebuah kemampuan untuk menggunakan satu variabel yang dapat menyimpan beberapa data dan memanipulasinya dengan lebih efektif. Tipe variabel inilah yang disebut sebagai **array**.



Gambar 1: Contoh dari Integer Array

Sebuah array akan menyimpan beberapa item data yang memiliki tipe data sama didalam sebuah blok memori yang berdekatan yang kemudian dibagi menjadi beberapa ruang. Array adalah sebuah variabel/sebuah lokasi tertentu yang memiliki satu nama sebagai *identifier*, namun identifier ini dapat menyimpan lebih dari sebuah nilai.

### 7.3 Pendeklarasian Array

Array harus dideklarasikan seperti layaknya sebuah variabel. Pada saat mendeklarasikan array, anda harus membuat sebuah daftar dari tipe data, yang diikuti oleh sepasang tanda kurung [], lalu diikuti oleh nama *identifier*-nya. Sebagai contoh,

```
int []ages;
```

atau Anda dapat menempatkan sepasang tanda kurung [] sesudah nama *identifier*. Sebagai contoh,

```
int ages[];
```

Setelah pendeklarasian array , kita harus membuat array dan menentukan berapa panjangnya dengan sebuah konstruktor. Proses ini di Java disebut sebagai *instantiation* (istilah 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. Sebagai catatan 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, pendeklarasian tersebut akan memberitahukan kepada compiler Java, bahwa identifier ages akan digunakan sebagai nama array yang berisi data bertipe integer, dan dilanjutkan dengan membuat atau meng-*instantiate* sebuah array baru yang terdiri dari 100 elemen.

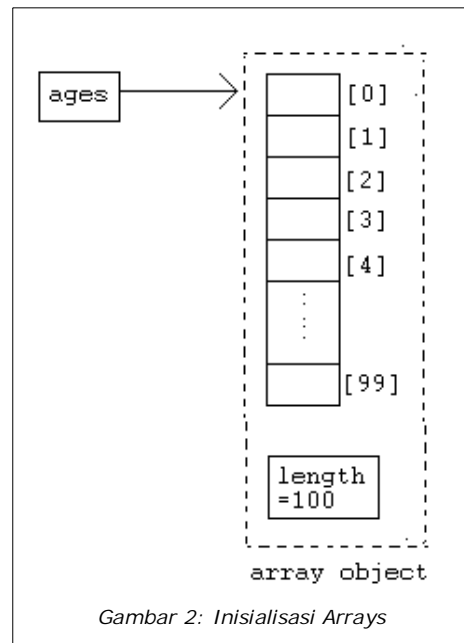
Selain menggunakan sebuah pernyataan *new* untuk meng-instantiate array, Anda juga dapat mendeklarasikan, membangun, kemudian memberikan sebuah nilai pada array sekaligus dalam sebuah pernyataan.

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
//4variabel 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"};
```



Gambar 2: Inisialisasi Arrays

## 7.4 Pengaksesan sebuah elemen array

Untuk mengakses sebuah elemen dalam array, atau mengakses sebagian dari array, Anda harus menggunakan sebuah angka atau yang disebut sebagai **indeks** atau subscript.

Pada saat memasukkan nilai ke dalam array, sebuah **nomor indeks atau subscript** telah diberikan kepada tiap anggota array, sehingga program dan programmer dapat mengakses setiap nilai pada array apabila dibutuhkan. Nilai indeks **selalu dalam tipe integer, dimulai dari angka nol dan dilanjutkan ke angka berikutnya sampai akhir array**. Sebagai catatan bahwa indeks 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 seperti String, array tidak akan diinisialisasi menjadi string kosong "". Untuk itu Anda tetap harus membuat String array secara eksplisit.

Berikut ini adalah contoh kode untuk mencetak seluruh elemen didalam array. Dalam contoh ini digunakanlah pernyataan *for 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:**

1. 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];
```

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

## 7.5 Panjang Array

Untuk mengetahui berapa banyak elemen didalam sebuah array, Anda dapat menggunakan atribut **length** dari array. Atribut ini 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] );
        }
    }
}
```

### ***Petunjuk penulisan program:***

1. Pada saat pembuatan for loop untuk memproses elemen-elemen dalam array, gunakanlah atribut length yang diletakkan di dalam bagian pengkondisian dari for loop. Hal ini akan menyebabkan loop secara otomatis menyesuaikan diri terhadap ukuran array yang berbeda-beda.
2. Pendeklarasian ukuran array di program Java, biasanya menggunakan sebuah konstanta untuk mempermudah. Sebagai contoh,

```
final int ARRAY_SIZE = 1000;    //pendeklarasian constant
...

int[] ages = new int[ARRAY_SIZE];
```

## 7.6 Array Multidimensi

Array multidimensi diimplementasikan sebagai array yang terletak di dalam array. Array multidimensi dideklarasikan dengan menambahkan jumlah tanda kurung setelah nama array. Sebagai contoh,

```
// Elemen 512 x 128 dari integer array
int[][] twoD = new int[512][128];

// karakter array 8 x 16 x 24
char[][][] threeD = new char[8][16][24];

// String array 4 baris x 2 kolom
String[][] dogs = {{ "terry", "brown" },
                  { "Kristin", "white" },
                  { "toby", "gray" },
                  { "fido", "black" }
                  };
```

Untuk mengakses sebuah elemen didalam array multidimensi, sama saja dengan mengakses array satu dimensi. Misalnya saja, untuk mengakses elemen pertama dari baris pertama didalam array dogs, kita akan menulis,

```
System.out.print( dogs[0][0] );
```

Kode diatas akan mencetak String "terry" di layar.

## 7.7 Latihan

### 7.7.1 Hari dalam seminggu

Buatlah sebuah String array yang akan menginisialisasi 7 hari dalam seminggu. Sebagai contoh,

```
String days[] = {"Monday", "Tuesday"...};
```

Gunakan while-loop, kemudian print semua nilai dari array (Gunakan juga untuk do-while dan for-loop) Using a while-loop.

### 7.7.2 Nomor terbesar

Gunakanlah BufferedReader dan JOptionPane, tanyakan kepada user untuk 10 nomor. Kemudian gunakan array untuk menyimpan 10 nomor tersebut. Tampilkan kepada user, input terbesar yang telah diberikan user.

### 7.7.3 Buku Alamat

Berikut ini adalah array multidimensi yang menyatakan isi dari sebuah buku alamat:

```
String entry = {"Florence", "735-1234", "Manila"},  
               {"Joyce", "983-3333", "Quezon City"},  
               {"Becca", "456-3322", "Manila"}};
```

Cetak buku alamat tersebut dalam format berikut ini:

```
Name : Florence  
Tel. #      : 735-1234  
Address    : Manila  
  
Name : Joyce  
Tel. #      : 983-3333  
Address    : Quezon City  
  
Name : Becca  
Tel. #      : 456-3322  
Address    : Manila
```