

Array 1 Dimensi

Mengapa Perlu Array

- Variabel dengan tipe dasar hanya dapat menyimpan sebuah nilai
- 5 buah nilai dengan tipe yang sama dapat saja disimpan dalam 5 buah variabel, tetapi bagaimana dengan 100 nilai? Disimpan dengan 100 variabel?

Mengapa Perlu Array

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;
```

Code di atas bukanlah contoh yang baik untuk memperlihatkan proses penginisialisasian dan penggunaan variabel, terutama jika variabel tersebut digunakan untuk tujuan yang sama.

Apa yang Dibutuhkan ...?

- Sebuah variabel yang merupakan kumpulan tipe-tipe variabel lain
 - Variabel ini dapat diubah dan digunakan dalam program sebagai sebuah kesatuan
 - Di saat bersamaan, setiap nilai di dalamnya dapat diakses
- Sebuah array

Apa yang Dibutuhkan ...?

- Baik Java maupun bahasa pemrograman lain, memiliki kemampuan untuk menggunakan satu variabel yang dapat menyimpan sebuah data list, kemudian memanipulasinya dengan lebih efektif.
- Tipe variabel inilah yang disebut sebagai **array**.
- Sebuah array akan menyimpan beberapa item data dengan tipe data yang sama di dalam sebuah blok memori yang berdekatan yang kemudian dibagi menjadi beberapa slot.

Array

- Koleksi (kumpulan) sejumlah objek (variabel) yang memiliki indeks dan memiliki tipe data yang sama.

nilaiMahasiswa [1]

[2]

[3]

[4]

[5]

Pendeklarasian array

Untuk mendeklarasikan array :

- tulis tipe datanya,
- diikuti dengan tanda kurung [],
- serta diikuti oleh nama identifier.

Contoh:

```
int []ages;
```

atau

```
int ages[];
```

Array instantiation

Setelah proses pendeklarasian, kita harus membuat array dan menentukan berapa panjangnya dengan sebuah pernyataan konstruktor.

Instantiation

Di Java, ini berarti pembuatan

Konstruktor

- Untuk meng-*instantiate* sebuah obyek, kita membutuhkan sebuah konstruktor.
- Sebuah konstruktor merupakan sebuah method yang dipanggil untuk membuat object

Array instantiation

Untuk meng-*instantiate* (atau membuat) sebuah array,

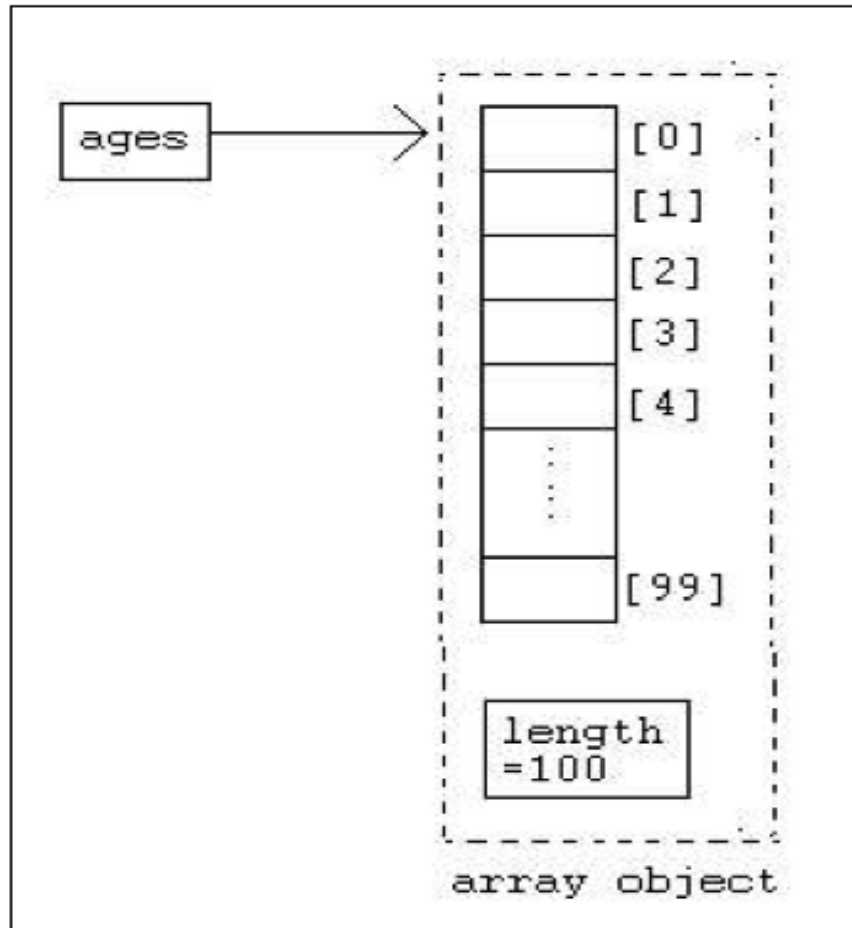
- Tulislah **new** keyword,
- Diikuti oleh tanda kurung [] yang berisi angka dari elemen yang Anda inginkan pada array tersebut.
- Contoh,

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

atau, juga dapat ditulis,

```
//declare and instantiate object  
int ages[] = new int[100];
```

Array instantiation



Array instantiation

Anda juga dapat meng-*instantiate* sebuah array dengan cara menginisialisasinya secara langsung dengan data.

Contoh,

```
int arr[] = {1, 2, 3, 4, 5};
```

Pernyataan ini mendeklarasikan dan meng-*instantiate* sebuah array dari integer dengan lima elemen (diinisialisasi dengan nilai 1, 2, 3, 4, dan 5).

Contoh program

```
//membuat sebuah array dari variabel boolean pada sebuah identifier hasil.  
//Array ini berisi 4 elemen untuk nilai {true, false, true, false}
```

```
boolean results[] = { true, false, true, false };
```

```
//Membuat sebuah array yang terdiri dari penginisialisasian 4 variabel  
//untuk nilai {100, 90, 80, 75};
```

```
double []grades = {100, 90, 80, 75};
```

```
//Membuat sebuah array String dengan identifier days  
//terdiri dari 7 elemen.
```

```
String days[] = { "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"};
```

Pengaksesan elemen array

Untuk mengakses sebuah elemen dalam array , atau mengakses sebagian elemen dari array , Anda menggunakan penomoran yang disebut sebagai **index** atau **subscript**.

Index atau **Subscript**

- Diberikan kepada tiap anggota array, agar program dapat mengakses anggota dari array.
- Dimulai dari nol dan kemudian akan terus bertambah sampai list value dari array tersebut berakhir.
- CATATAN: elemen di dalam array anda dimulai dari 0 sampai (sizeofArray-1)

Pengaksesan elemen array

Contoh, pada array yang telah 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]);
```

Pengaksesan elemen array

CATATAN:

- Jika array telah dideklarasikan dan dikonstruksi, nilai yang disimpan dalam setiap anggota array akan diinisialisasi sebagai nol.
- Untuk tipe data reference seperti String, nilai dari String tidak akan diinisialisasi ke string kosong "", sehingga Anda tetap harus membuat array String secara eksplisit.

Pengaksesan elemen array

Berikut ini merupakan contoh, untuk mencetak seluruh elemen di dalam array. Terdapat penggunaan loop pada contoh yang membuat kode lebih pendek

```
1 public class ArraySample{
2     public static void main( String[] args ){
3         int[] ages = new int[100];
4         for( int i=0; i<100; i++ ){
5             System.out.print( ages[i] );
6         }
7     }
8 }
```


Petunjuk penulisan program

1. Akan lebih baik jika anda menginisialisasi atau meng-*instantiate* array setelah mendeklarasikannya.

Contoh deklarasi,

```
int []arr = new int[100];
```

juga diperbolehkan,

```
int []arr;  
arr = new int[100];
```

Petunjuk penulisan program

2. Elemen-elemen dalam array dengan **n-elemen** memiliki index dari **0** sampai **n-1**.

Perhatikan bahwa tidak ada elemen array *arr[n]*, karena hal ini akan menyebabkan **array-index-out-of-bounds exception** .

3. Anda tidak dapat mengubah ukuran sebuah array.

Panjang array

Untuk mengetahui jumlah elemen di dalam sebuah array, Anda dapat menggunakan **length** (panjang) field pada array.

Panjang field dalam array akan me-*return value* dari array itu sendiri.

Atau dapat ditulis,

arrayName.length

Panjang array

```
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] );  
        }  
    }  
}
```

Panduan penulisan program

1. Pada saat pembuatan loop untuk memproses elemen-elemen dalam array, gunakanlah `length` field di dalam *statement* pengkondisian dalam loop.
Hal ini akan menyebabkan loop secara otomatis menyesuaikan diri terhadap ukuran array yang berbeda-beda.
2. **constant** biasanya digunakan untuk mempermudah pendeklarasian ukuran array di Java.
Sebagai contoh,

```
final int ARRAY_SIZE = 1000; //declare a  
constant
```

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

Review

1. Dengan bantuan variabel array, simpan dan tampilkan sepuluh jenis musik ini: pop, dangdut, jazz, keroncong, reggae, acapella, rock, heavy metal, instrumental, blues