

Java Developer Fundamentals

# Module

Array pada Java

A circular inset image showing a person from behind, wearing a headset and working on a laptop in a coding bootcamp environment. Other people and laptops are visible in the background.

**CODING  
BOOTCAMP**

# Arrays

1. Array berfungsi untuk menyimpan kumpulan(list) data yang sejenis
2. Array memiliki struktur :
  - a. name
  - b. type
  - c. length
3. Struktur diatas harus dideklarasikan sebelum dapat digunakan.
4. Dimensi/ ruang array tidak dapat ditambah atau dikurangi setelah terbentuk.

myArray = 

<b>3</b>	<b>6</b>	<b>3</b>	<b>1</b>	<b>6</b>	<b>3</b>	<b>4</b>	<b>1</b>
0	1	2	3	4	5	6	7

myArray mempunyai ruang untuk 8 elements

- element dapat diakses dengan menggunakan indexnya
- Pada Java, array index dimulai dari 0

# Declaring Arrays

```
int myArray[];
```

deklarasi *myArray* untuk menjadi array penampung integers

```
myArray = new int[8];
```

sets up 8 integer-sized ruangan memory, labelled *myArray[0]* sampai *myArray[7]*

```
int myArray[] = new int[8];
```

combines dua statements menjadi satu baris perintah

# Assigning Values

1. Untuk menyimpan data harus menunjuk ke indexnya.

```
myArray[0] = 3;
```

```
myArray[1] = 6;
```

```
myArray[2] = 3; ...
```

2. Membuat dan langsung me -inisialisasi array dengan nilai:

```
int myArray[] = {3, 6, 3, 1, 6, 3, 4, 1};
```

# Assigning Values

## Beberapa Cara Penulisan Array

:

```
(public class MateriArray {  
    public static void main(String[] args) {  
        int [] nilai ={1,2,3,4,5};  
        System.out.println(nilai[2]);  
    }  
}
```

### Cara 2

```
public class MateriArray {  
    public static void main(String[] args) {  
        int [] nilai = new int[4];  
        nilai[0]=1;  
        nilai[1]=2;  
        nilai[2]=3;  
        nilai[3]=4;  
        System.out.println(nilai[2]);  
    }  
}
```

Perbedaannya hanya kalau di cara kedua sudah menetapkan berapa banyak nilai (dimensi) yang akan dimasukkan ke dalam array

# Loop through an array

Kita bisa melooping sebuah array, berdasarkan dari jumlah array tersebut

```
public class MateriArray {  
    public static void main(String[] args) {  
        String[] cars = {"Volvo", "BMW", "Ford", "Mazda"};  
        for (int i = 0; i < cars.length; i++) {  
            System.out.println(cars[i]);  
        }  
    }  
}
```

Disini menggunakan cars.length untuk melooping sesuai banyaknya data di dalam array

# Array Multidimensi

Array multidimensi adalah array yang berisi array atau sekumpulan array. Secara matematis, array multidimensi disebut juga sebagai matriks. Untuk menulis array dua dimensi, menambahkan masing-masing array dengan kurung kurawal. Contoh;

```
int [][] angkaKu = {{1,2,3,4},{5,6,7}};
```

AngkaKu sekarang merupakan array dengan dua array di dalamnya. Untuk mengakses elements angkaKu, perlu disediakan dua index: satu untuk array, dan satunya lagi untuk element di dalam array. Contohnya mengakses element ketiga (2) in array kedua (1) dari angkaKu. Contoh

```
int [][] angkaKu = {{1,2,3,4},{5,6,7}};
```

```
int x= angkaKu[1][2];
```

```
System.out.println(x);
```



# Array Multidimensi

Kita juga bisa melakukan looping mendapatkan element dari array dua dimensi

Contoh

```
public class MateriArray {  
    public static void main(String[] args) {  
        int[][] myNumbers = { { 1, 2, 3, 4 }, { 5, 6, 7 } };  
        for (int i = 0; i < myNumbers.length; ++i) {  
            for (int j = 0; j < myNumbers[i].length; ++j) {  
                System.out.println(myNumbers[i][j]);  
            }  
        }  
    }  
}
```

# Mengurutkan Elemen pada Array

- Mengurutkan data adalah hal yang umum dilakukan dalam pemrograman. Banyak algoritma yang digunakan untuk mengurutkan data, salah satunya adalah *selection sort*.
- Selection sort melakukan pencarian terhadap nilai yang paling besar dan menempatkan nilai tersebut di paling belakang. Kemudian mencari nilai terbesar lagi pada data yang tersisa, dan menempatkannya kembali di bagian belakang.

# Mengurutkan Elemen pada Array

**SOURCE CODE DIMINTA SAJA KE TRAINER**

# Pencarian Elemen pada Array

- Pencarian (searching) adalah proses mencari spesifik elemen di dalam array, sebagai contoh menemukan nilai tertentu ada pada array atau tidak. Ada banyak algoritma dan struktur data yang digunakan pada proses pencarian. Dua algoritma yang banyak digunakan adalah *linear search* dan *binary search*.

# Linear Search

Linear search membandingkan elemen kunci (*key*) dengan setiap elemen di dalam array. Pencarian dilakukan sampai *key* sama dengan elemen di dalam array atau pencarian mencapai ujung dari array tanpa ditemukan elemen yang sama.

```
for(int i=0; i<myArray.length; i++) {  
    if(key == myArray[i]){  
        return i;  
    }  
}  
return -1;
```

# Binary Search

Agar binary search bisa bekerja, maka elemen di dalam array harus sudah terurut (misalnya secara nilai). Binary search dilakukan dengan melihat tiga hal berikut ini:

- Jika key kurang dari elemen yang ada di tengah array, maka pencarian key dilakukan pada setengah array pertama.
- Jika key sama dengan elemen yang ada di tengah array, pencarian berhenti dengan elemen ditemukan.
- Jika key lebih besar dari elemen yang ada di tengah array, maka pencarian dilakukan pada setengah array kedua.

# Binary Search

**SOURCE CODE DIMINTA SAJA KE TRAINER**

# Inserting Array into Another Array

myArray1=

<b>3</b>	<b>6</b>	<b>3</b>	<b>3</b>	<b>4</b>	<b>1</b>
0	1	2	3	4	5

myArray2=

<b>1</b>	<b>6</b>
0	1

insertAt= 3

myArray3=

<b>3</b>	<b>6</b>	<b>3</b>	<b>1</b>	<b>6</b>	<b>3</b>	<b>4</b>	<b>1</b>
0	1	2	3	4	5	6	7



# Inserting Array into Another Array

```
int[] myArray1 = {3,6,3,3,4,1};
int[] myArray2 = {1,6};
int insertAt = 3;

int[] myArray3 = new int[myArray1.length+myArray2.length];

for (int i = 0; i<insertAt; i++) {
    myArray3[i] = myArray1[i];
}

for (int j = insertAt; j < insertAt + myArray2.length ; j++) {
    myArray3[j] = myArray2[j-insertAt];
}

for (int k = insertAt + myArray2.length ; k < myArray3.length ; k++) {
    myArray3[k] = myArray1[k-myArray2.length];
}
```