



Array of Objects

TIM AJAR

ALGORITMA DAN STRUKTUR DATA

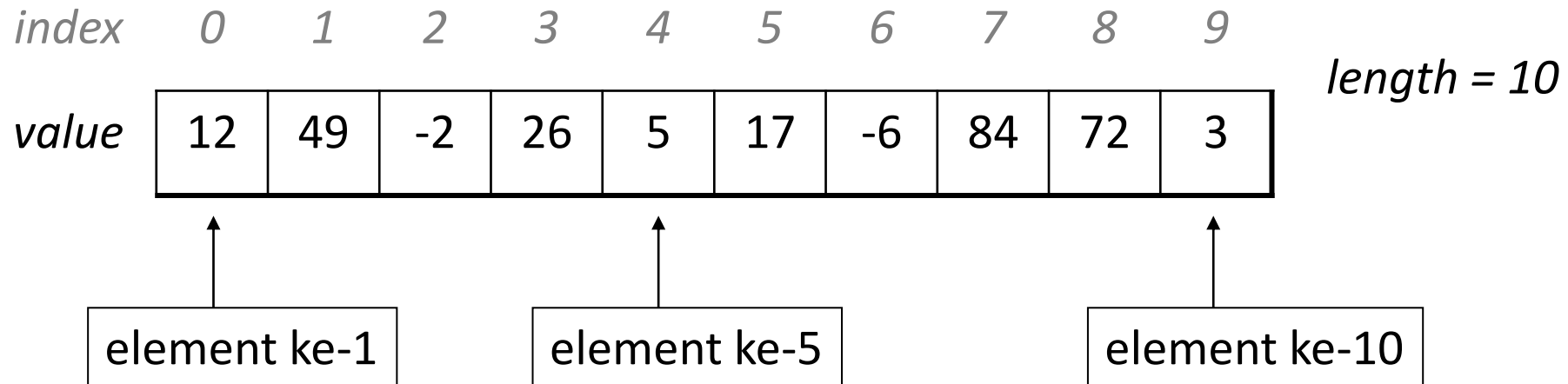
2023/2024

Array

- Array merupakan suatu struktur data yang berisi sekumpulan nilai (elemen) dengan tipe data yang sama.
- Setiap elemen array bisa diakses dengan menggunakan indeks yang unik

Sifat Array

- Homogen
 - Seluruh elemen di dalam struktur array mempunyai tipe data yang sama.
- Random Access
 - Setiap elemen di dalam struktur array dapat diakses secara individual, langsung ke lokasi elemen yang diinginkan, tidak harus melalui elemen pertama.
- Termasuk reference variable.



Array of Integer

```
int[] daftarNilai = new int[5];
```

```
daftarNilai[0] = 7;  
daftarNilai[1] = 3;  
daftarNilai[2] = 8;  
daftarNilai[3] = 2;  
daftarNilai[4] = 1;
```

0	1	2	3	4
0	0	0	0	0

0	1	2	3	4
7	3	8	2	1

Array of String

```
String[] itemNames = new String[3];
```

0	1	2
<i>null</i>	<i>null</i>	<i>null</i>

```
itemNames[0] = "Nasi goreng";
```

```
itemNames[1] = "Donat gula";
```

```
itemNames[2] = "Cimory";
```

0	1	2
Nasi goreng	Donat gula	Cimory

Array of Objects

- Selain elemen dengan tipe data primitif seperti int, float, double, dan sebagainya, array juga dapat menyimpan elemen bertipe object.
- Array of int → kumpulan elemen bertipe int
- Array of String → kumpulan elemen bertipe String
- Array of objects → kumpulan elemen bertipe object

Array of Objects

0



Name= "Nasi goreng"
Price = 15000
Stock = 10

1



Name = "Donat gula"
Price = 3000
Stock = 20


2



Name = "Cimory"
Price = 7000
Stock = 15


```
public class Item {  
    public String name;  
    public double price;  
    public int stock;  
  
    public void displayInfo(){  
        System.out.println("Name: " + name);  
        System.out.println("Price: " + price);  
        System.out.println("Stock: " + stock);  
    }  
}
```

```
public class Item {  
    public String name;  
    public double price;  
    public int stock;  
  
    public void displayInfo(){  
        System.out.println("Name: " + name);  
        System.out.println("Price: " + price);  
        System.out.println("Stock: " + stock);  
    }  
}
```



```
public class CanteenDemo {  
    Run | Debug  
    public static void main(String[] args) {  
        Item item1 = new Item();  
        item1.name = "Nasi goreng";  
        item1.price = 15000;  
        item1.stock = 10;  
  
        Item item2 = new Item();  
        item2.name = "Donat gula";  
        item2.price = 3000;  
        item2.stock = 20;  
  
        Item item3 = new Item();  
        item3.name = "Cimory";  
        item3.price = 7000;  
        item3.stock = 15;  
    }  
}
```

```
✓ public class Item {  
    public String name;  
    public double price;  
    public int stock;  
  
    ✓ public Item(){  
        }  
  
    ✓ public Item(String itemName, double itemPrice, int itemStock) {  
        name = itemName;  
        price = itemPrice;  
        stock = itemStock;  
    }  
  
    ✓ public void displayInfo(){  
        System.out.println("Name: " + name);  
        System.out.println("Price: " + price);  
        System.out.println("Stock: " + stock);  
    }  
}
```

- Menggunakan objek-objek bertipe Item dengan constructor berparameter

```
public class CanteenDemo {  
    Run | Debug  
    public static void main(String[] args) {  
        Item item1 = new Item("Nasi goreng", 15000, 10);  
        Item item2 = new Item("Donat gula", 3000, 20);  
        Item item3 = new Item("Cimory", 7000, 15);  
    }  
}
```

Deklarasi & Instansiasi Array

- Berikut beberapa cara melakukan deklarasi, instansiasi, dan inisialisasi array:
 - `int[] daftarNilai1 = new int[3];`
 - `int[] daftarNilai2 = {75, 90, 80};`
 - `int[] daftarNilai3 = new int[] {75, 90, 80};`

Deklarasi & Instansiasi Array

- Seperti array pada umumnya, array of objects diinstansiasi menggunakan keyword **new** beserta informasi **jumlah elemen**

```
tipeData[] namaArray = new tipeData[jumlahElemen];
```

- `int[] daftarNilai = new int[3];`
- `String[] itemNames = new String[3];`
- `Item[] items = new Item[3];`
- `Book[] books = new Book[3];`
- `Student[] students = new Student[30];`

Array of Integer

```
int[] daftarNilai = new int[5];
```

```
daftarNilai[0] = 7; ✓
```

```
daftarNilai[1] = 3; ✓
```

```
daftarNilai[2] = "Delapan"; ✗
```

Array of String

```
String[] itemNames = new String[3];
```

```
itemNames[0] = "Nasi goreng";
```



```
itemNames[1] = "Donat gula";
```



```
itemNames[2] = 1000;
```



Array of Objects

```
Item[] items = new Item[3];
```

```
items[0] = "Nasi goreng";
```



```
items[0] = item1;
```



```
items[1] = new Item();
```



```
items[2] = new Item("Roti", 5000, 25);
```



Atribut & Method dari Elemen Array

- Bagaimana cara mengakses atribut dan memanggil method dari element bertipe object di dalam array?
- Cara #1 Gunakan nama object yang tadi dibuat:

```
item1.name = "Nasi goreng ayam";  
item2.stock = 18;  
item2.displayInfo();
```

Atribut & Method dari Elemen Array

- Cara #2 Gunakan nama array dan nomor index:

```
items[0].name = "Nasi goreng ayam";  
items[1].stock = 18;  
items[2].displayInfo();
```

Atribut & Method dari Elemen Array

- Cara #3 Simpan dulu elemennya ke variable baru:

```
Item myItem = items[2];  
myItem.name = "Roti ayam";  
myItem.displayInfo();
```

Default Value

- `int[] listNilai = new int[5];`

0	1	2	3	4
0	0	0	0	0

- `boolean[] daftarStatus = new boolean[2]`

0	1
<i>false</i>	<i>false</i>

- `Item[] items = new Item[3]`

0	1	2
<i>null</i>	<i>null</i>	<i>null</i>

NullPointerException

- *NullPointerException* adalah error yang muncul saat program berusaha melakukan operasi terhadap nilai null
- Contoh:

```
Item[] items = new Item[9];  
items[0].name = "Nasi kuning";
```
- Kode program di atas akan menyebabkan *NullPointerException* karena pada array items dengan indeks 0 belum ada object bertipe Item

NullPointerException (2)

- Object bertipe Item harus diinstansiasi terlebih dahulu dengan menggunakan constructor dan keyword new

```
Item[] items = new Item[9];  
items[0] = new Item();  
items[0].name = "Nasi kuning";
```

Looping untuk membuat objek di semua indeks array

- Looping dapat digunakan untuk membuat objek baru dan menjadikannya sebagai elemen dalam array

```
for (int i = 0; i < items.length; i++) {  
    items[i] = new Item();  
}
```


Looping untuk mengakses elemen dalam array



```
for (int i = 0; i < items.length; i++) {  
    System.out.println(items[i].name);  
    System.out.println(items[i].price);  
    System.out.println(items[i].stock);  
}
```

Latihan

1. Ingat kembali project akhir mata kuliah Dasar Pemrograman. Identifikasi 2 class beserta atribut dan methodnya kemudian buatlah class diagram untuk setiap class.

Latihan

2. Ingat kembali sintaks foreach. Bagaimana padanan kode program berikut jika looping menggunakan foreach?

```
int[] daftarNilai = new int[5];  
daftarNilai[0] = 7;  
daftarNilai[1] = 3;  
daftarNilai[2] = 8;  
daftarNilai[3] = 2;  
daftarNilai[4] = 1;  
  
for (int i = 0; i < daftarNilai.length; i++) {  
    System.out.println(daftarNilai[i]);  
}
```

