

## 1. Tujuan

- menjelaskan OOP dan beberapa konsepnya
- perbedaan antara class dan object
- perbedaan antara instance variables/method dan class (static) variable/method
- menjelaskan method apa dan bagaimana memanggil method parameter
- mengidentifikasi beberapa jangkauan dari sebuah variable
- memilih tipe data primitive dan object
- membandingkan objects dan menjabarkan class dari objects.

## 2. Latar Belakang

Pada sesi ini, kita akan mengantarkan beberapa konsep dasar dari Object-Oriented objects, dan Programming (OOP). Selanjutnya kita akan membahas konsep dari classes dan bagaimana menggunakan class dan anggotanya. Perubahan dan pemilihan object juga akan dibahas. Sekarang, kita akan focus dalam menggunakan class yang telah dijabarkan dalam Java Class library, kita akan membahas nanti tentang bagaimana membikin class anda sendiri.

OOP berputar pada konsep dari object sebagai dasar element dari program anda. Ketika kita membandingkan dengan dunia nyata, kita dapat menemukan beberapa objek disekitar kita, seperti mobil, singa, manusia dan seterusnya. Object ini dikarakterisasi oleh sifat / attributnya dan tingkah lakunya.

Contohnya, objek sebuah mobil mempunyai sifat tipe transmisi, warna dan manufaktur. Mempunyai kelakuan berbelok, mengerem dan berakselerasi. Dengan cara yang sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari singa. Coba perhatikan table dibawah ini sebagai contoh perbandingan :

<i>Object</i>	<i>Properties</i>	<i>Behavior</i>
Car	type of transmission manufacturer color	turning braking accelerating



<i>Object</i>	<i>Properties</i>	<i>Behavior</i>
Lion	Weight Color hungry or not hungry tamed or wild	roaring sleeping hunting

### 3. Percobaan

#### Percobaan 1 Pass By Value:

```
public class TestPastByValue {  
    public static void main(String[] args) {  
        int i = 10;  
        System.out.println(i);  
        test(i);  
        System.out.println(i);  
    }  
    public static void test(int j){  
        j = 33;  
    }  
}
```



### Hasil Percobaan 1 Output pass by value :

```
<terminated> TestPastByValue [Java Applicat  
10  
10
```

### Percobaan 2 Pass by reference :

```
public class TestPassByReference {  
    public static void main(String[] args) {  
        int []ages = {10,11,12};  
        for(int i=0;i<ages.length;i++){  
            System.out.println(ages[i]);  
        }  
        test(ages);  
        for(int i=0;i<ages.length;i++){  
            System.out.println(ages[i]);  
        }  
    }  
    public static void test(int []arr){  
        for(int i=0;i<arr.length;i++){  
            arr[i]=i+50;  
        }  
    }  
}
```

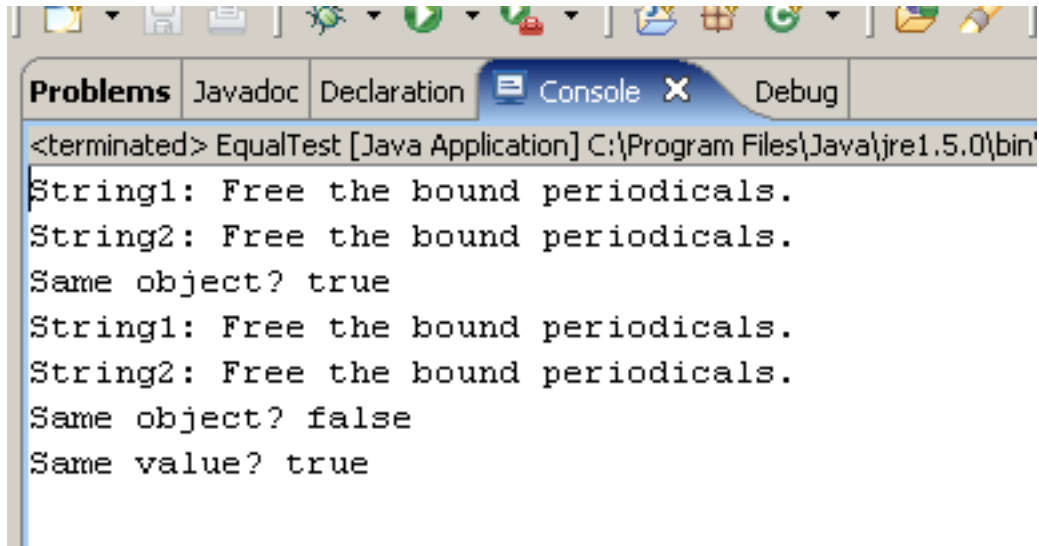
## Hasil Percobaan 2 Output Pass by reference :

```
Problems | Javadoc | Declaration |  
<terminated> TestPassByReference  
10  
11  
12  
50  
51  
52
```

## Percobaan 3 Perbandingan Object :

```
class EqualsTest {  
    public static void main(String[] arguments) {  
        String str1, str2;  
        str1 = "Free the bound periodicals.";  
        str2 = str1;  
        System.out.println("String1: " + str1);  
        System.out.println("String2: " + str2);  
        System.out.println("Same object? " + (str1 == str2));  
        str2 = new String(str1);  
        System.out.println("String1: " + str1);  
        System.out.println("String2: " + str2);  
        System.out.println("Same object? " + (str1 == str2));  
        System.out.println("Same value? " + str1.equals(str2));  
    }  
}
```

### Hasil Percobaan 3 Output Frame Control :

A screenshot of a Java IDE's console window. The window title is "<terminated> EqualTest [Java Application] C:\Program Files\Java\jre1.5.0\bin". The console output shows two test cases for string equality. In the first case, two identical strings are compared, resulting in "Same object? true". In the second case, two identical strings are compared, resulting in "Same object? false" and "Same value? true".

```
<terminated> EqualTest [Java Application] C:\Program Files\Java\jre1.5.0\bin
String1: Free the bound periodicals.
String2: Free the bound periodicals.
Same object? true
String1: Free the bound periodicals.
String2: Free the bound periodicals.
Same object? false
Same value? true
```