

Java Fundamental

Saniati

saniati@teknokrat.ac.id

STMIK Teknokrat, Bandar Lampung

String

- Tidak termasuk primitiv data type
- Termasuk object / user defined datatype
- Digunakan untuk menyimpan deretan karakter
- Valuenya harus diapit dengan tanda “ (double quote)

```
String s = "Hello Lampung";  
System.out.println(s);
```

- Sebaiknya tidak menggunakan operator == untuk tipe data object, sebaiknya gunakan method equals();

Membandingkan object String

```
String st1= "bandar lampung kota yang indah";  
String st2= "bandar lampung kota yang indah";  
String st3= "bandar lampung ";  
String st4= "kota yang indah";  
String st5= st3+st4;
```

```
System.out.println("statement 1 : "+st1);  
System.out.println("statement 2 : "+st2);  
System.out.println("statement 3 : "+st3);  
System.out.println("statement 4 : "+st4);  
System.out.println("statement 5 : "+st5);
```

```
System.out.println("membandingkan dengan operator ==");  
System.out.println("statement1 == statement2 : "+(st1==st2));  
System.out.println("statement1 == statement5 : "+(st1==st5));
```

```
System.out.println("membandingkan dengan method equal");  
System.out.println("statement1.equal(statement2) : "+(st1.equals(st2)));  
System.out.println("statement1.equal(statement5) : "+(st1.equals(st5)));
```

Operator == jika di apply ke object akan membandingkan alamat memori

Method equal akan membandingkan isi memori

Hasil eksekusi

```
membandingkan dengan operator ==  
statement1 == statement2 : true  
statement1 == statement5 : false
```

```
membandingkan dengan method equal  
statement1.equal(statement2) : true  
statement1.equal(statement5) : false
```

Compare String

- Method `compareTo()` digunakan untuk membandingkan posisi sebuah string
- Jika return 0, berarti kedua string secara leksikal order sama
- Jika return -, berarti kedua pertama memiliki leksikal order lebih kecil
- Jika return +, berarti kedua pertama memiliki leksikal order lebih besar

Sub Program

- Definisi dari sub program:
 - blok kode
 - punya nama
 - bisa dipanggil
 - digunakan untuk melakukan task yang sering dijalankan
- Kadang disebut sebagai method
- Ada 2 jenis:
 - Procedure (void method)
 - Method yang jika dipanggil tidak mengembalikan value
 - Function (non void method)
 - Method yang dipanggil mengembalikan value
 - Cirinya, hasil bisa ditangkap variabel dan dapat diumpankan pemanggilannya menggunakan `System.out.print()`

Procedure

- Buat procedure

```
static public void cetakArrayString(String s[]){  
    for(int i=0;i<s.length;i++){  
        System.out.println(s[i]);  
    }  
}
```

- Panggil procedure dari method main

```
String kota[] = {"bandung","surabaya","jakarta"};  
cetakArrayString(kota);  
  
String negara[] = {"indonesia","singapura","malaysia"};  
cetakArrayString(negara);
```

Function

- Buat fungsi, dengan return integer

```
static public int kali(int a, int b){  
    int hasil = a*b;  
    return (hasil);  
}
```

- Panggil fungsi dari method main

```
System.out.println(kali(4,8));  
  
int hasil = kali(10,5);  
System.out.println(hasil);
```

Method Overloading

- Deklarasi metode dengan nama yang sama, namun parameter memiliki jumlah atau tipe data yang berbeda.

```
static public void cetakArray(int n[]){
    for(int i=0;i<n.length;i++){
        System.out.println(n[i]);
    }
}

static public void cetakArray(String s[]){
    for(int i=0;i<s.length;i++){
        System.out.println(s[i]);
    }
}

static public void cetakArray(char c[]){
    for(int i=0;i<c.length;i++){
        System.out.println(c[i]);
    }
}

static public void cetakArray(double n[]){
    for(int i=0;i<n.length;i++){
        System.out.println(n[i]);
    }
}
```

Contoh disamping
berbeda tipe data

Main Method

- Starting point untuk menjalankan sebuah java program
- Program java dapat diexecute dengan command:
 - java NamaClass
 - dengan syarat memiliki method main
- Program java yang memiliki method main disebut driver class

```
public static void main(String[] args) {  
    //tambahkan kode yang ingin ditulis  
}
```

Parameter Pada Method Main

- Parameter pada method main dapat dimanfaatkan sebagai masukan ketika program pertama kali dijalankan.

```
public class Test
{
    public static void main(String [] args)
    {
        if(args.length==0){
            System.out.println("anda tidak memasukkan parameter");
        }else{
            System.out.println("parameter anda : ");
            cetakArray(args);
        }
    }
    static public void cetakArray(String s[]){
        for(int i=0;i<s.length;i++){
            System.out.println(s[i]);
        }
    }
}
```

```
E:\pelatihan>java Test satu dua tiga
parameter anda :
satu
dua
tiga
```

Input

- Untuk dapat membaca data masukan dari device (ex. keyboard) dapat menggunakan library (import) `java.util.Scanner`.

```
import java.util.Scanner;
public class Test
{
    public static void main(String [] args)
    {
        Scanner s = new Scanner(System.in);
        String nama;
        String alamat;

        System.out.print("Masukkan nama : ");
        nama = s.nextLine();

        System.out.print("Masukkan alamat : ");
        alamat = s.nextLine();

        System.out.print("Selamat datang "+nama+" dari " + alamat);

    }
}
```

```
Masukkan nama : Saniati
Masukkan alamat : Bandar Lampung
Selamat datang Saniati dari Bandar Lampung
```

Reference

- Pemaparan materi TOT Java Fundamental oleh bapak Tri Haryoko (7-11 Nopember 2016, Bandar Lampung)
- <https://docs.oracle.com/javase/tutorial/java>
- “Thinking in Java”, Bruce Eckel