

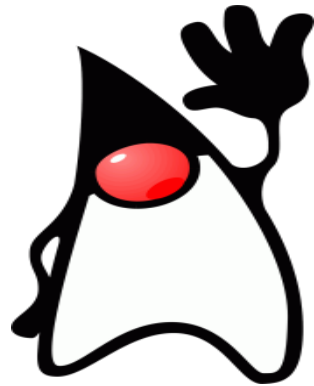


LAB PEMROGRAMAN I (JAVA FUNDAMENTAL)

PERTEMUAN 4

Dosen :

Bella Hardiyana S. Kom, M. Kom



BAB IV

FUNGSI INPUT OUTPUT DASAR (OUTPUT)

Mengenal Fungsi Input/Output Dasar

Mengenal fungsi input/output dasar adalah suatu hal yang sangat penting. Semua hal yang ada di dunia ini pasti ada input (masukan), proses dan output (keluaran). Banyak sekali fungsi/method input dan output dalam bahasa Java. Penulis hanya akan menampilkan beberapa saja.

Secara garis besar fungsi input/output terbagi menjadi dua, yaitu: input/output berbasis DOS dan input/output berbasis GUI. Penulis akan mengajarkan fungsi terlebih dahulu karena dianggap lebih mudah.

Fungsi Output

Penulis membagi fungsi output dalam empat bagian, yaitu :

- *Output* berbasis DOS menggunakan printf()
- *Output* berbasis DOS menggunakan print()
- *Output* berbasis DOS menggunakan println()
- *Output* berbasis GUI menggunakan JOptionPane

Semua output berbasis DOS terdapat pada class **System** dan melalui objek referensi **out**. Jadi, jika kita ingin melakukan inputan dari keyboard, maka sintaksnya selalu diawali dengan “**System.out**”

Output Menggunakan **printf()**

Method **printf()** berfungsi sebagai output dengan menentukan penentu format. Itulah sebabnya nama method `printf()`, huruf **f** berasal dari singkatan format.

```
System.out.printf("format-string", argumen);
```

Format-string adalah tulisan/teks yang akan tampil di layar monitor. Didalam format-string, kita menentukan penentu format yang nantinya akan ditampilkan melalui argumen. **Argumen** adalah variabel/data yang akan ditampilkan berdasarkan apa yang sudah didaftarkan pada format-string.

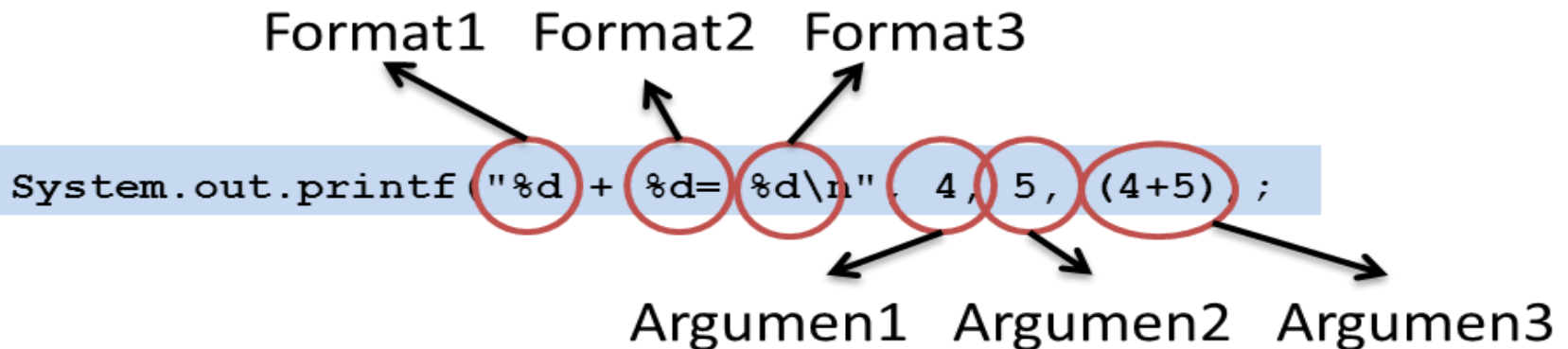
Daftar penentu format bahasa Java

Format	Untuk menampilkan variabel/data berjenis
%d	Bilangan bulat (integer)
%f	Bilangan pecahan (floating-point)
%c	Karakter 1 digit (char)
%s	Karakter banyak digit (String)
%b	Nilai true/false (boolean)

Contoh Penggunaan printf()

```
System.out.printf("%d + %d= %d\n", 4, 5, (4+5));
```

Berikut cara kerja dari method printf()



Sehingga muncul output seperti dibawah ini.

```
4 + 5= 9
```

Contoh Penggunaan printf()

```
public class ContohFungsiOutPrintf {  
    public static void main(String[] args) {  
        System.out.printf("%s\n", "Halo Halo Bandung");  
        System.out.printf("%s %s\n", "Halo Halo", "Bandung");  
        System.out.printf("%d + %d= %d\n", 4, 5, (4+5));  
        System.out.printf("PHI= %f\n", 3.14);  
        System.out.printf("index= %c\n", 'A');  
        System.out.printf("nilai boolean= %b\n", true);  
    }  
}
```


Output Menggunakan **print()**

Method **print()** adalah method yang berfungsi untuk mencetak data, setelah data tersebut dicetak tidak diikuti dengan perpindahan baris baru. Jadi jika menginginkan fungsi ini memiliki fasilitas untuk berpindah baris, maka terpaksa kita harus mencantumkan '**\n**' diakhir sintaks. Terkadang data yang ditampilkan tidak selalu berupa teks/String. Jika kita ingin mencampur output berupa teks/String dan variabel, maka yang harus kita lakukan adalah menambahkan karakter '**+**' sebagai penyampung data antara String dan variabel.

Contoh Penggunaan print()

```
System.out.print("NIM : "+nim+", Nama : "+nama+'\n');
```

Misal isi variabel nim= "10506377" dan nama="Udin".
Berikut cara kerja dari method print()

```
System.out.print("NIM : "+nim+", Nama : "+nama+'\n');
```

NIM : 10506377, Nama : Udin //setelah itu kursor pindah baris

Sehingga muncul output seperti dibawah ini.

```
NIM : 10506377, Nama : Udin
```

Contoh Penggunaan **print()**

```
public class ContohFungsiOutPrint {  
    public static void main(String[] args) {  
        System.out.print("Halo Halo ");  
        System.out.print("Bandung\n");  
        String nim="10506357";  
        String nama="Phantom Assassin";  
  
        System.out.print("NIM : "+nim+'\n');  
        System.out.print("Nama : "+nama+'\n');  
        System.out.print("NIM : "+nim+", Nama : "+nama+'\n');  
    }  
}
```

Output Menggunakan **println()**

Fungsi **method println()** sangat mirip dengan `printf()`, yaitu berfungsi untuk mencetak data, setelah data tersebut dicetak diikuti dengan perpindahan baris baru. Jadi kita tidak perlu mencantumkan '`\n`' seperti yang dilakukan method `printf()`.

Konsep dari method `println()` sama dengan `print()`. Dimana jika kita ingin mencampur output String dan variabel, maka harus digabung dengan menggunakan tanda '+'.
`print("String " + variabel);`

Contoh Penggunaan `println()`

```
System.out.println("NIM : "+nim+", Nama : "+nama);
```

Misal isi variabel `nim= "10506377"` dan `nama="Udin"`.
Berikut cara kerja dari method `println()`

```
System.out.println("NIM : "+nim+", Nama : "+nama);
```



```
NIM : 10506377, Nama : Udin //setelah itu kursor pindah baris
```

Sehingga muncul output seperti dibawah ini.

```
NIM : 10506377, Nama : Udin
```

Contoh Penggunaan `println()`

```
public class ContohFungsiOutPrintln {  
    public static void main(String[] args) {  
        System.out.println("Halo Halo ");  
        System.out.println("Bandung");  
        String nim="10506357";  
        String nama="Phantom Assassin";  
        System.out.println("NIM : "+nim);  
        System.out.println("Nama : "+nama);  
        System.out.println("NIM : "+nim+", Nama : "+nama);  
    }  
}
```

Output Menggunakan JOptionPane

1. **Mengimport JOptionPane.class** yang ada di paket **javax.swing**. Pendeklarasian import ditulis diatas pendeklarasian class.

```
import javax.swing.JOptionPane //wajib ditulis  
public class Contoh{  
}
```

2. Memanggil method showMessageDialog().

```
JOptionPane.showMessageDialog(null, String);
```

Contoh Penggunaan JOptionPane

```
import javax.swing.JOptionPane;

public class ContohOutJOptionPane{
    public static void main(String[] args){
        String nama="Mulyana";
        JOptionPane.showMessageDialog(
            null, "Halo-Halo Bandung");
        JOptionPane.showMessageDialog(
            null, "Halo "+nama+'\n'+ "Apa Kabar?");
    }
}
```


Contoh Penggunaan JOptionPane

