

CARA MENGAMBIL INPUT DAN MENAMPILKAN OUTPUT



KOMPONEN UTAMA SISTEM

Seperti yang kita ketahui, program komputer terdiri dari tiga komponen utama, yaitu: input, proses, dan output.

- Input: nilai yang kita masukan ke program
- Proses: langkah demi langkah yang dilakukan untuk mengelola input menjadi sesuatu yang berguna
- Output: hasil pengolahan

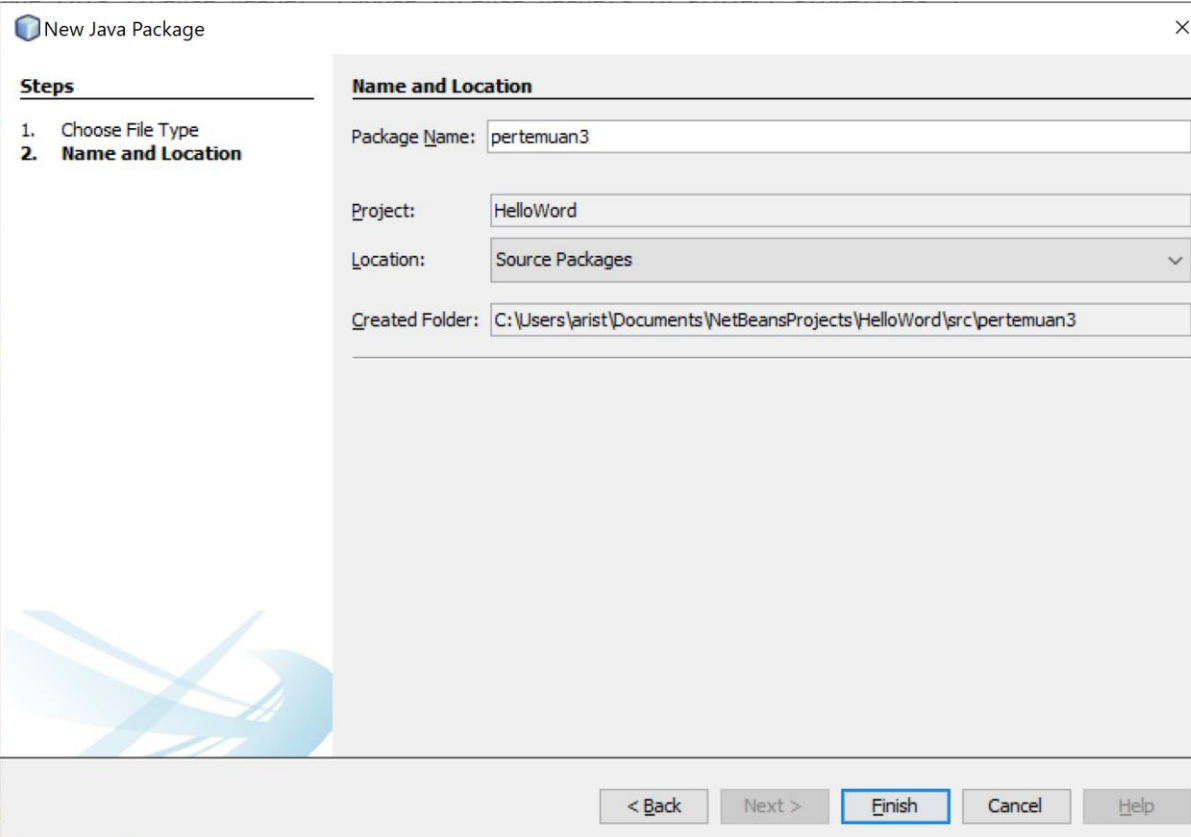
- Semua bahasa pemrograman telah menyediakan fungsi-fungsi untuk melakukan input dan output.
- Java sendiri sudah menyediakan tiga class untuk mengambil input:
 - Class Scanner;
 - Class BufferedReader;
 - dan Class Console.
- Tiga class tersebut untuk mengambil input pada program berbasis teks (console).
- Sedangkan untuk GUI menggunakan class yang lain seperti JOptionPane dan inputbox pada form.
- Sementara untuk outputnya, Java menyediakan fungsi print(), println(), dan format().

MENGAMBIL INPUT DENGAN CLASS SCANNER

- Scanner merupakan class yang menyediakan fungsi-fungsi untuk mengambil input dari keyboard.
- Agar kita bisa menggunakan Scanner, kita perlu mengimpornya ke dalam kode:

```
import java.util.Scanner;
```

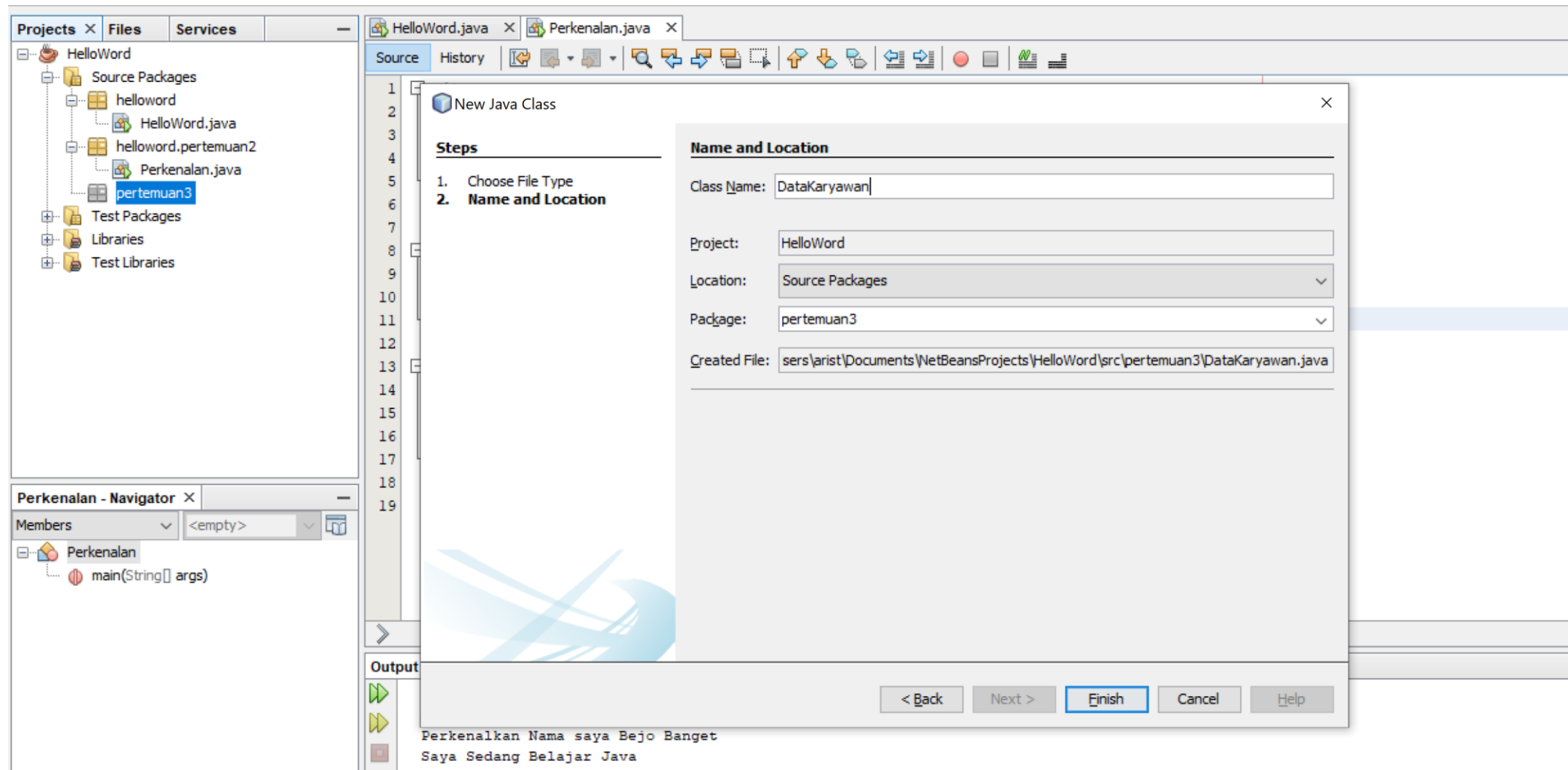
- Sekarang adalah pertemuan ke-3, jadi buatlah paket baru bernama pertemuan3 pada source packages.
- Klik kanan pada souce packages, kemudian pilih new package:



The screenshot shows the 'New Java Package' dialog box. On the left, under 'Steps', step 2 'Name and Location' is selected. The main area is titled 'Name and Location' and contains the following fields:

- Package Name:** pertemuan3
- Project:** HelloWord
- Location:** Source Packages (dropdown menu)
- Created Folder:** C:\Users\arist\Documents\NetBeansProjects\HelloWord\src\pertemuan3

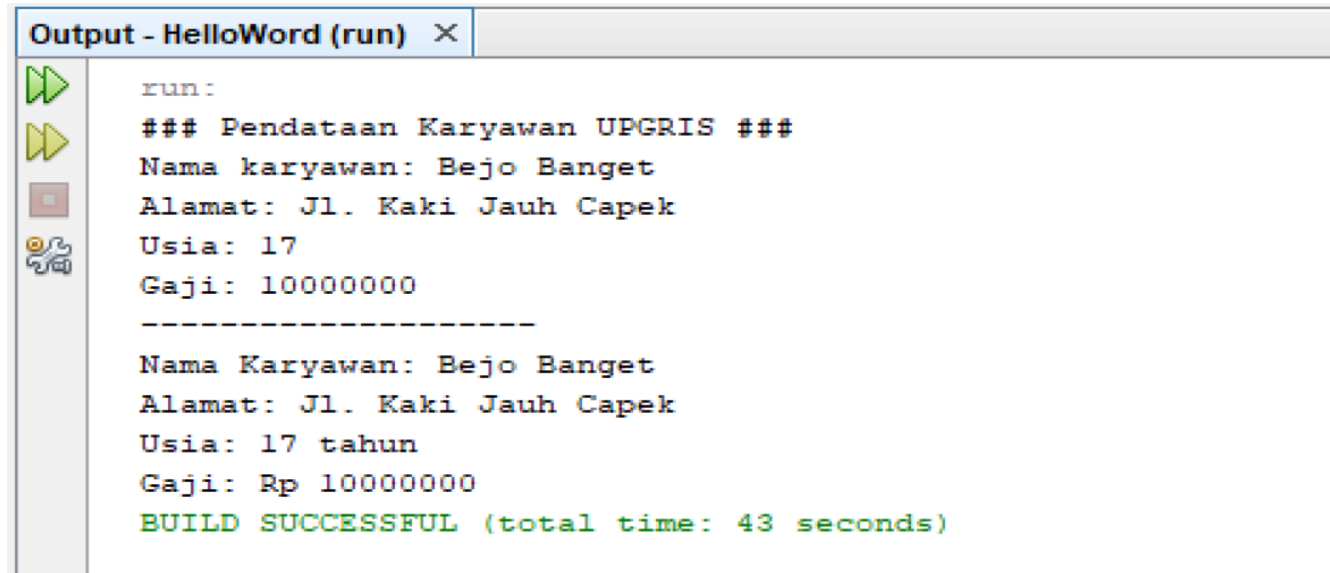
At the bottom, there are five buttons: '< Back', 'Next >', 'Finish' (highlighted with a blue border), 'Cancel', and 'Help'.



SETELAH ITU, SILAHKAN IKUTI KODE BERIKUT INI:

```
1 package pertemuan3;
2
3 import java.util.Scanner;
4
5
6 public class DataKaryawan {
7     // mengimpor Scanner ke program
8
9     public static void main(String[] args) {
10         // deklarasi variabel
11         String nama, alamat;
12         int usia, gaji;
13
14         // membuat scanner baru
15         Scanner keyboard = new Scanner(System.in);
16
17         // Tampilkan output ke user
18         System.out.println("### Pendataan Karyawan UPGRIS ###");
19         System.out.print("Nama karyawan: ");
20         // menggunakan scanner dan menyimpan apa yang diketik di variabel nama
21         nama = keyboard.nextLine();
22         // Tampilkan output lagi
23         System.out.print("Alamat: ");
24         // menggunakan scanner lagi
25         alamat = keyboard.nextLine();
26
27         System.out.print("Usia: ");
28         usia = keyboard.nextInt();
29
30         System.out.print("Gaji: ");
31         gaji = keyboard.nextInt();
32
33         // Menampilkan apa yang sudah simpan di variabel
34         System.out.println("-----");
35         System.out.println("Nama Karyawan: " + nama);
36         System.out.println("Alamat: " + alamat);
37         System.out.println("Usia: " + usia + " tahun");
38         System.out.println("Gaji: Rp " + gaji);
39     }
40
41 }
```

- Perlu diperhatikan, penggunaan fungsi untuk mengambil data bergantung dari tipe data yang digunakan.
- Misal, tipe datanya adalah String, maka fungsi atau method yang dipakai adalah `nextLine()`.
- Begitu juga dengan tipe data lain, Integer menggunakan `nextInt()`, Double menggunakan `nextDouble()`, dsb.
- Setelah selesai membuat program, silahkan dijalankan.
- Klik kanan kemudian pilih Run File atau tekan tombol `[Shift]+[F6]`.



```
run:
### Pendataan Karyawan UPGRIS ###
Nama karyawan: Bejo Banget
Alamat: Jl. Kaki Jauh Capek
Usia: 17
Gaji: 10000000
-----
Nama Karyawan: Bejo Banget
Alamat: Jl. Kaki Jauh Capek
Usia: 17 tahun
Gaji: Rp 10000000
BUILD SUCCESSFUL (total time: 43 seconds)
```


MENGAMBIL INPUT DENGAN CLASS BUFFERREADER

- Class **BufferedReader** sebenarnya tidak hanya untuk mengambil input dari keyboard saja.
- Class ini juga dapat digunakan untuk membaca input dari file dan jaringan.
- Class ini terletak di dalam paket **java.io**.
- Silahkan diimpor untuk dapat menggunakan class **BufferedReader**.

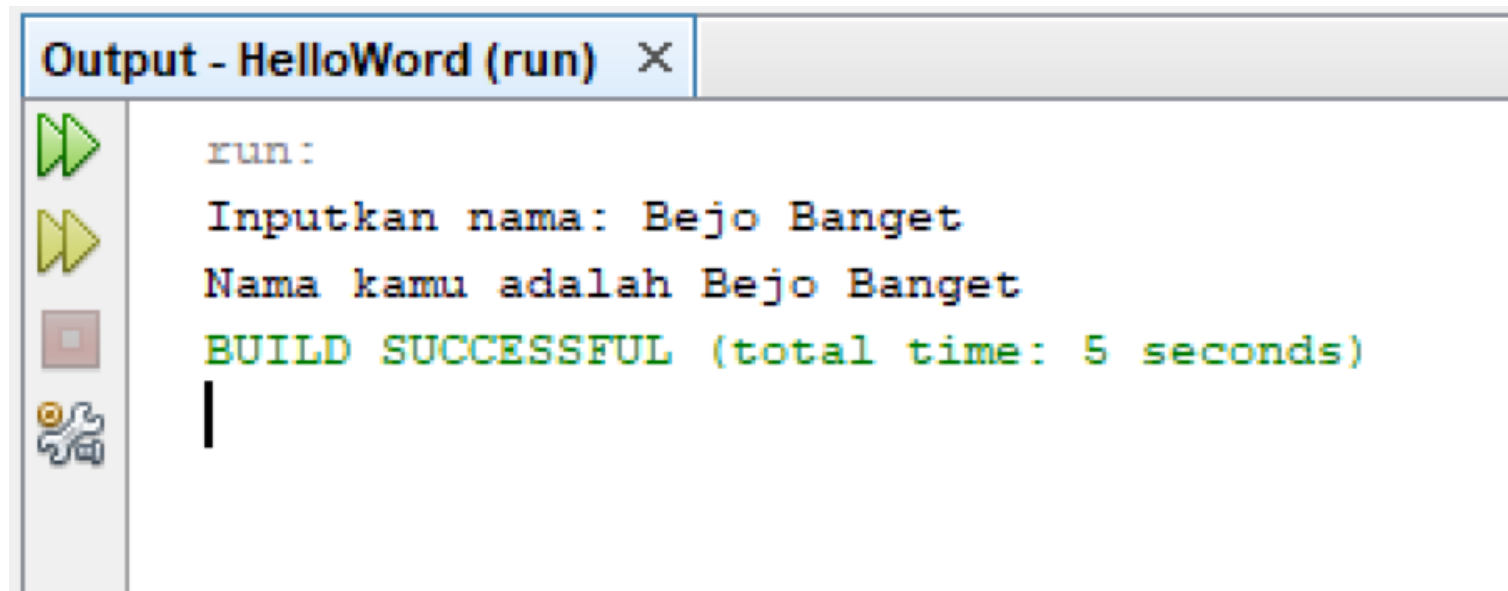
```
import java.io.BufferedReader;
```

MARI KITA COBA...

- Silahkan buat class baru bernama ContohBufferReader, kemudian isi dengan kode berikut ini.

```
1  package pertemuan3;
2
3  import java.io.BufferedReader;
4  import java.io.IOException;
5  import java.io.InputStreamReader;
6
7  public class ContohBufferReader {
8
9      public static void main(String[] args) throws IOException {
10
11          String nama;
12
13          // Membuat objek inputstream
14          InputStreamReader isr = new InputStreamReader(System.in);
15
16          // membuat objek bufferreader
17          BufferedReader br = new BufferedReader(isr);
18
19          // Mengisi variabel nama dengan Bufferreader
20          System.out.print("Inputkan nama: ");
21          nama = br.readLine();
22
23          // tampilkan output isi variabel nama
24          System.out.println("Nama kamu adalah " + nama);
25
26      }
27
28  }
```

- Ternyata class `BufferedReader` tidak bisa bekerja sendirian. Dia juga butuh teman yaitu: class `InputStreamReader` dan class `IOException`.
- Sekarang mari kita coba jalankan programnya:



The screenshot shows an IDE's Output window titled "Output - HelloWord (run)". On the left side of the window, there is a vertical toolbar with four icons: a green double arrow (run), a yellow double arrow (debug), a red square (stop), and a gear icon (preferences). The main area of the window displays the following text:

```
run:
Inputkan nama: Bejo Banget
Nama kamu adalah Bejo Banget
BUILD SUCCESSFUL (total time: 5 seconds)
|
```

- Perbedaan `BufferedReader` dengan `Scanner` terlihat dari fungsi atau *method* yang dipakai.
- `Scanner` menggunakan `next()`, sedangkan `BufferedReader` menggunakan `readLine()`.
- Lalu untuk tipe data integer, `BufferedReader` menggunakan fungsi `read()` saja. Hal ini dapat kita lihat dalam *hint autocomplete*.
- **Tips:** Tekan **Ctrl+Spasi** saat menulis kode untuk menampilkan hint autocomplete

```
usia = br.read  
// tapi  
System.o
```

○ <code>read()</code>	<code>int</code>
○ <code>read(CharBuffer cb)</code>	<code>int</code>
○ <code>read(char[] chars)</code>	<code>int</code>
○ <code>read(char[] chars, int i, int il)</code>	<code>int</code>
○ <code>readLine()</code>	<code>String</code>
○ <code>ready()</code>	<code>boolean</code>

MENGAMBIL INPUT DENGAN CLASS CONSOLE

- Class Console hampir sama dengan BufferedReader. Dia juga menggunakan fungsi `readLine()` untuk mengambil input.

Akan tetapi...

- Class ini hanya bisa digunakan di lingkungan console saja, seperti Terminal dan CMD.
- Class Console tidak bisa digunakan langsung di Netbeans.
- Maka dari itu, kita harus kompilasi secara manual.
- Untuk menggunakan class ini, kita perlu mengimpornya terlebih dahulu.

```
import java.io.Console;
```

MARI KITA COBA...

- Buatlah file baru bernama InputConsole.java dengan isi sebagai berikut:

```
1 package pertemuan3;
2
3 import java.io.Console;
4
5 public class InputConsole {
6     public static void main(String[] args) {
7
8         String nama;
9         int usia;
10
11         // membuat objek console
12         Console con = System.console();
13
14         // mengisi variabel nama dan usia dengan console
15         System.out.print("Inputkan nama: ");
16         nama = con.readLine();
17         System.out.print("Inputkan usia: ");
18         usia = Integer.parseInt(con.readLine());
19
20         // mengampilkan isi variabel nama dan usia
21         System.out.println("Nama kamu adalah: " + nama);
22         System.out.println("Saat ini berusia " + usia + " tahun");
23     }
24 }
```

MENAMPILKAN OUTPUT

Kita sudah mengenal beberapa cara mengambil input dari keyboard untuk program berbasis teks.

Sekarang bagaimana dengan outputnya?

Ada beberapa fungsi yang sudah disediakan oleh Java:

- Fungsi `System.out.print()`
- Fungsi `System.out.println()`
- Fungsi `System.out.format()`

Apa saja perbedaan dari fungsi-fungsi tersebut?

FUNGSI PRINT() VS PRINTLN()

- Fungsi `print()` dan `println()` sama-sama digunakan untuk menampilkan teks.

Lalu apa bedanya?

- Fungsi `print()` akan menampilkan teks apa adanya.
- Sedangkan `println()` akan menampilkan teks dengan ditambah baris baru.

```
public class PrintVsPrinln {  
  
    public static void main(String[] args) {  
  
        System.out.print("ini teks yang dicetak dengan print()");  
        System.out.println("sedangkan ini teks yang dicetak dengan println()");  
        System.out.print("pake print() lagi");  
  
    }  
  
}
```

MENGGABUNGKAN STRING

Ketika menggunakan fungsi `print()` maupun `println()`, kadang kita perlu mengambil teks dari variabel dan menggabungkannya dengan teks yang lain.

Misalnya seperti ini:

Kita punya variabel `namaDepan` dan `namaBelakang`:

```
String namaDepan = "Bejo";  
String namaBelakang = "Banget";
```

- Kemudian kita ingin menampilkannya dengan fungsi `print()`, maka kita hanya perlu memasukkannya ke sana.

```
System.out.print(namaDepan);  
System.out.print(namaBelakang);
```

Kode tersebut akan menghasilkan: **BejoBanget**

- Sebenarnya kita tidak perlu menggunakan dua fungsi print(), karena kita bisa menggabungkannya dengan operator +.

```
System.out.print(namaDepan + namaBelakang);
```

Agar ada spasi, tinggal ditambahkan saja spasi:

```
System.out.print(namaDepan + " " + namaBelakang);
```

FORMAT STRING

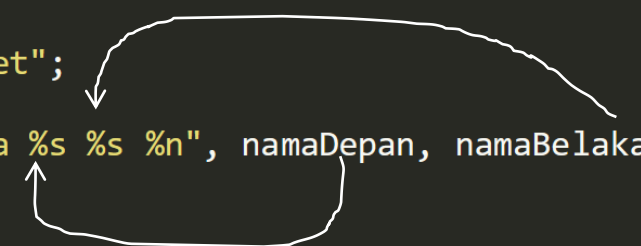
Sedangkan untuk menggabungkan String yang lebih kompleks, kita bisa menggunakan fungsi `format()`.

```
public class FormatString {  
  
    public static void main(String[] args) {  
  
        String namaDepan = "Bejo";  
        String namaBelakang = "Banget";  
  
        System.out.format("Nama saya %s %s %n", namaDepan, namaBelakang);  
  
    }  
  
}
```

Perhatikan: di sana kita menggunakan simbol **%s** untuk mengambil nilai dari variabel di sampingnya.

%s artinya string.

```
public static void main(String[] args) {  
    String namaDepan = "Bejo";  
    String namaBelakang = "Banget";  
    System.out.format("Nama saya %s %s %n", namaDepan, namaBelakang);  
}
```

A diagram with two curved arrows. One arrow starts from the variable 'namaDepan' on the line 'String namaDepan = "Bejo";' and points to the first '%s' in the format string '"Nama saya %s %s %n"'. The other arrow starts from the variable 'namaBelakang' on the line 'String namaBelakang = "Banget";' and points to the second '%s' in the same format string.

Selain **%s**, ada juga simbol lain:

- **%d** untuk desimal atau angka;
- **%f** untuk bilangan pecahan;
- **%n** untuk baris baru, bisa juga pakai `\n`;
- dan masih banyak lagi, cek di [dokumentasi java](#).