

Cara Mengambil Input dan Menampilkan Output Java

5 min read · Oct 2, 2025

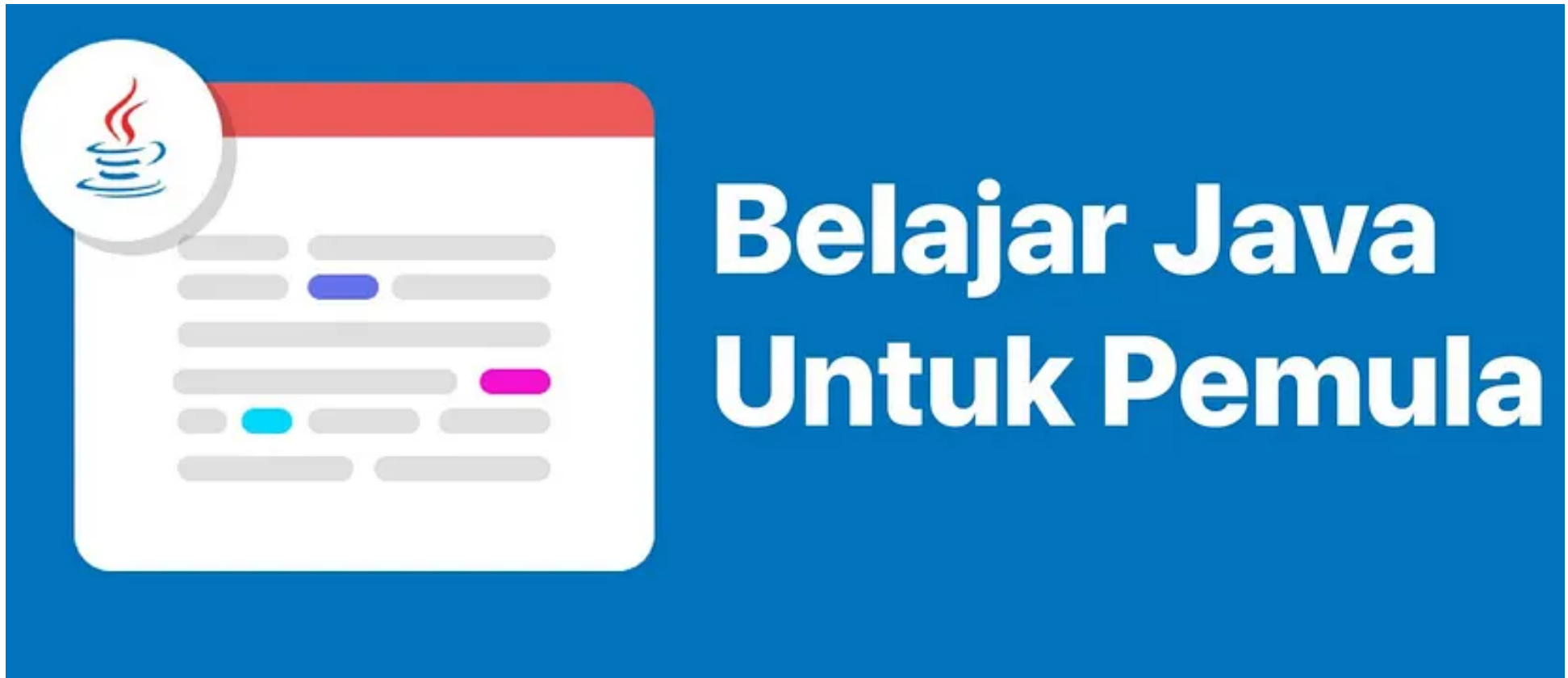


RifkyAkbarVetian

Follow



Share



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:

1. Class Scanner;
2. Class BufferedReader;
3. 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.

Tetapi, yang akan kita bahas adalah Class Scanner dan Class BufferedReader.

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:

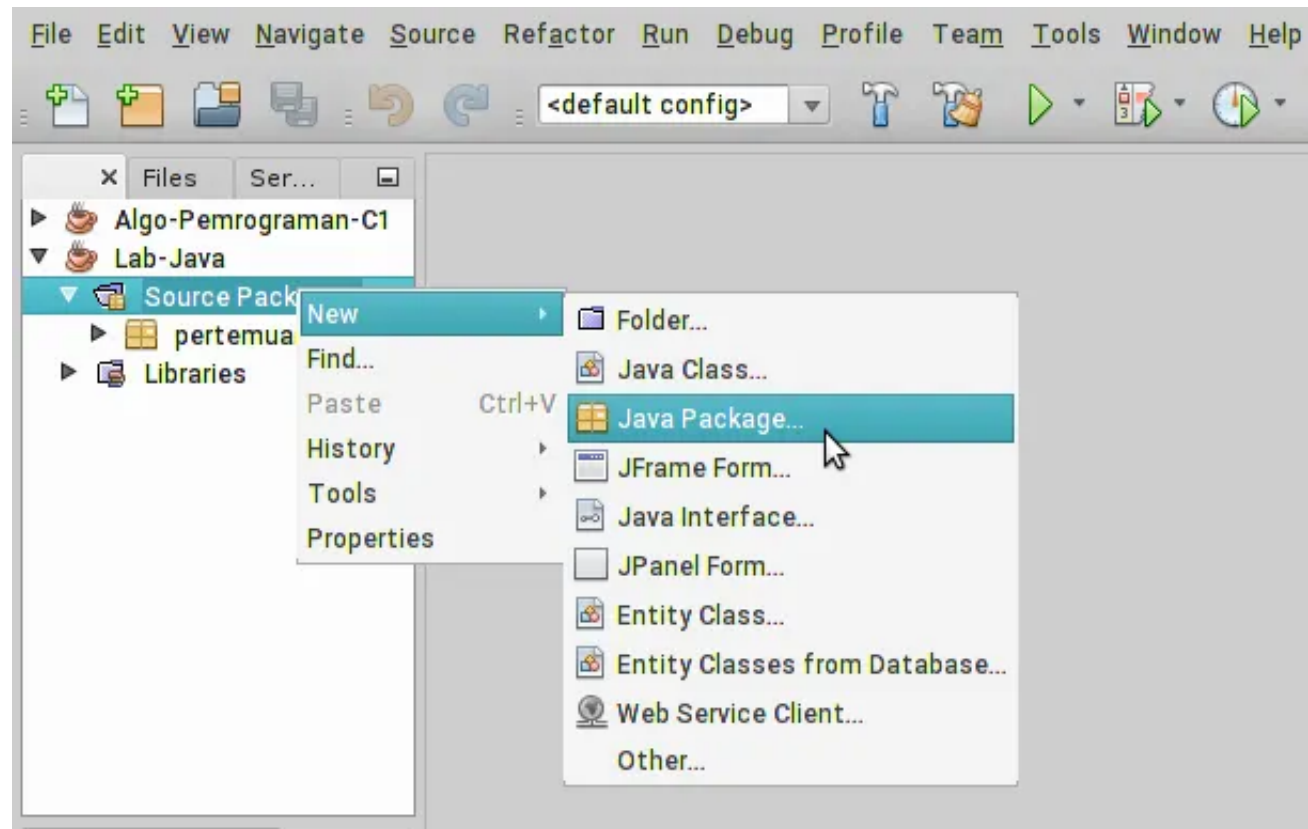
```
import java.util.Scanner;
```

Lebih jelasnya...

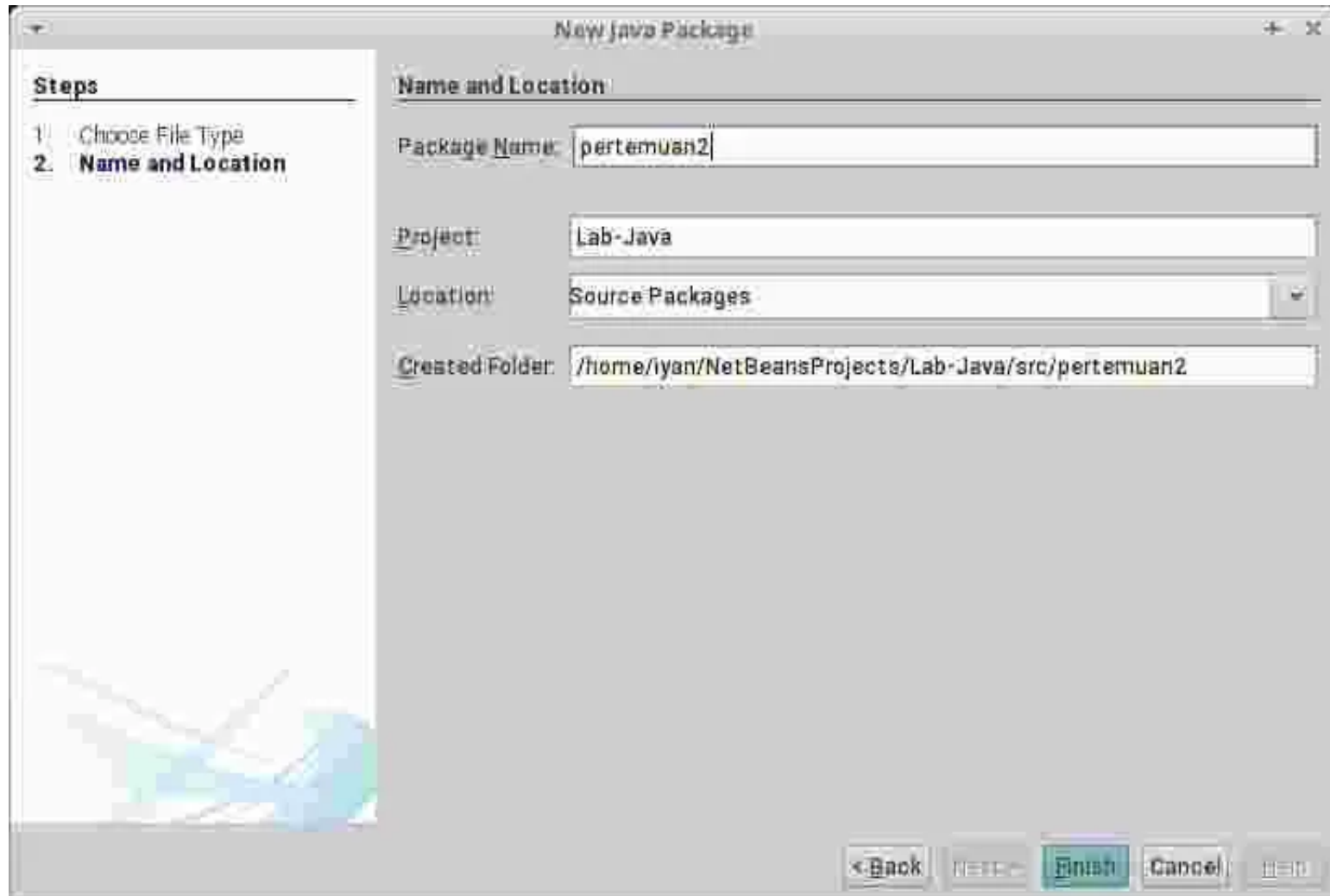
Mari Kita Praktik

Sekarang adalah pertemuan ke-2, jadi buatlah paket baru bernama *pertemuan2* pada *source packages*.

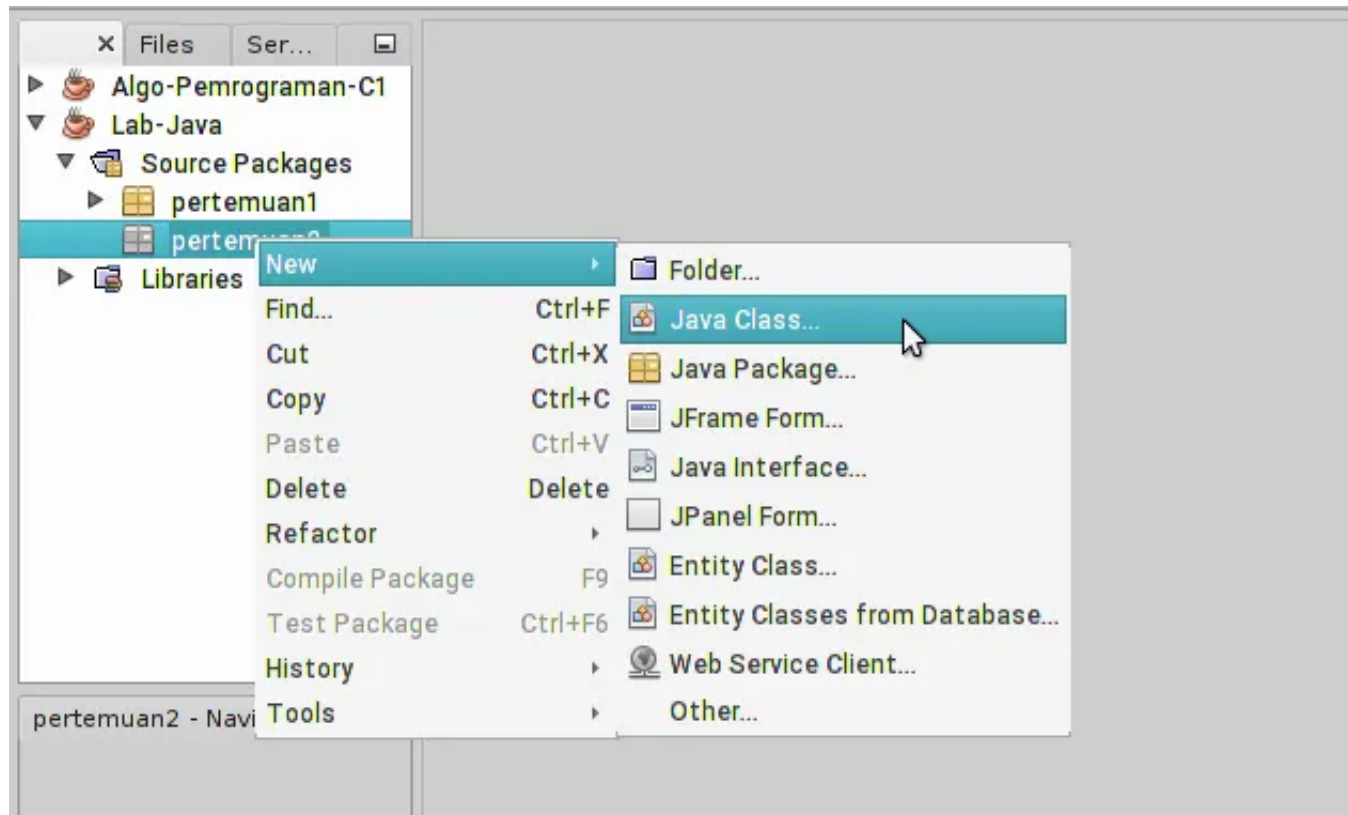
Klik kanan pada *source packages*, kemudian pilih *new package*:



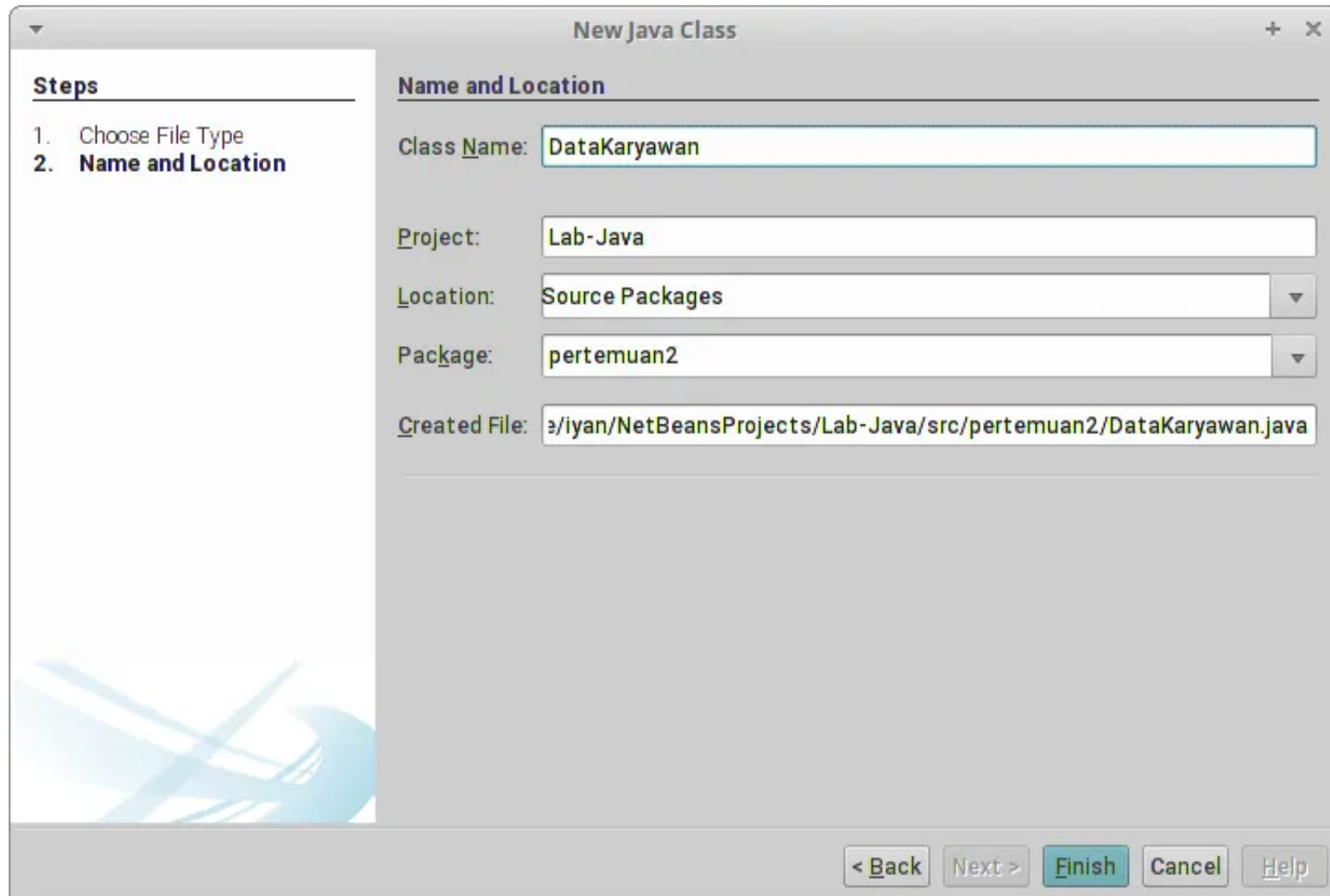
Kemudian isi nama paket dengan *pertemuan2*, lalu klik *Finish*:



Setelah itu, lanjutkan dengan membuat kelas (*java class*) baru di paket *pertemuan2*:



Berikan nama: *DataKaryawan*. Kemudian klik *Finish*:



Setelah itu, silakan ikuti kode berikut ini:

```
package pertemuan2;
```

```

// mengimpor Scanner ke program
import java.util.Scanner;

public class DataKaryawan {

    public static void main(String[] args) {
        // deklarasi variabel
        String nama, alamat;
        int usia, gaji;

        // membuat scanner baru
        Scanner keyboard = new Scanner(System.in);

        // Tampilkan output ke user
        System.out.println("### Pendataan Karyawan Universitas Putera Batam ###");
        System.out.print("Nama karyawan: ");
        // menggunakan scanner dan menyimpan apa yang diketik di variabel nama
        nama = keyboard.nextLine();
        // Tampilkan output lagi
        System.out.print("Alamat: ");
        // menggunakan scanner lagi
        alamat = keyboard.nextLine();

        System.out.print("Usia: ");
        usia = keyboard.nextInt();

        System.out.print("Gaji: ");
        gaji = keyboard.nextInt();

        // Menampilkan apa yang sudah simpan di variabel
        System.out.println("-----");
    }
}

```



```
        System.out.println("Nama Karyawan: " + nama);  
        System.out.println("Alamat: " + alamat);  
        System.out.println("Usia: " + usia + " tahun");  
        System.out.println("Gaji: Rp " + gaji);  
    }  
  
}
```

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, silakan dijalankan. Klik kanan kemudian pilih *Run File* atau tekan tombol [Shift]+[F6].

```
--- exec:3.1.0:exec (default-cli) @ Pertemuan2 ---  
### Pendataan Karyawan Universitas Putera ###  
Nama karyawan: Rifky  
Alamat: Tembesi  
Usia: 28  
Gaji: 5000000  
-----  
Nama Karyawan: Rifky  
Alamat: Tembesi  
Usia: 28 tahun  
Gaji: Rp 5000000  
-----  
BUILD SUCCESS  
-----  
Total time: 27.947 s  
Finished at: 2025-10-02T00:43:39+07:00  
-----
```

Mengambil Input dengan Class BufferedReader

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`.

Silakan diimpor untuk dapat menggunakan class `BufferedReader` .

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

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

Mari kita coba...

Silakan buat class baru bernama `ContohBufferedReader` , kemudian isi dengan kode berikut ini.

```
package pertemuan2;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class ContohBufferedReader {

    public static void main(String[] args) throws IOException {

        String nama;

        // Membuat objek inputstream
        InputStreamReader isr = new InputStreamReader(System.in);

        // membuat objek bufferreader
        BufferedReader br = new BufferedReader(isr);
```

```
// Mengisi variabel nama dengan BufferedReader
System.out.print("Inputkan nama: ");
nama = br.readLine();

// tampilkan output isi variabel nama
System.out.println("Nama kamu adalah " + nama);

}

}
```

Ternyata class `BufferedReader` tidak bisa bekerja sendirian. Dia juga butuh teman yaitu: class `InputStreamReader` dan class `IOException`.

Sekarang mari kita coba jalankan programnya:

```
--- exec:3.1.0:exec (default-cli) @ Pertemuan2 ---  
Inputkan nama: Rifky  
Nama kamu adalah Rifky  
  
-----  
BUILD SUCCESS  
-----  
Total time: 5.960 s  
Finished at: 2025-10-02T01:00:41+07:00  
-----
```

Perbedaan `BufferedReader` dengan `Scanner` terlihat dari fungsi atau *method* yang dipakai.

Get RifkyAkbarVetian's stories in your inbox

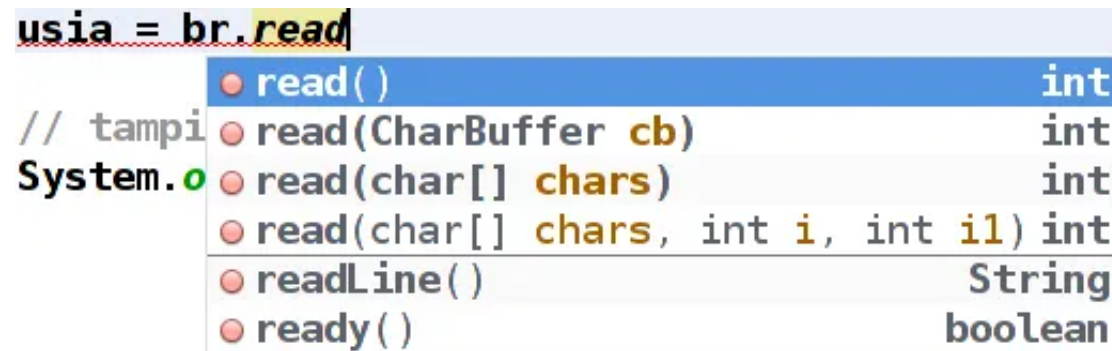
Join Medium for free to get updates from this writer.

Subscribe

`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* .

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:

1. Fungsi `System.out.print()`
2. Fungsi `System.out.println()`

3. Fungsi `System.out.format()`

Apa saja perbedaan dari fungsi-fungsi tersebut?

Mari kita bahas...

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.

Mari kita coba dalam kode:

```
package eksperimen;

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("pakai print() lagi");
    }
}
```

```
}  
  
}
```

Sekarang lihat hasil outputnya:

```
--- exec:3.1.0:exec (default-cli) @ Pertemuan2 ---  
ini teks yang dicetak dengan print()sedangkan ini teks yang dicetak dengan println()  
pakai print() lagi  
-----  
BUILD SUCCESS  
-----  
Total time: 1.545 s  
Finished at: 2025-10-02T01:26:30+07:00  
-----
```

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 = "Rifky";  
String namaBelakang = "Vetian";
```

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: `RifkyVetian`

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

Contoh:

```
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()`.

Contoh:

```
package eksperimen;
```

[Sign up](#) [Sign in](#)

Medium



```
String namaDepan = "Rifky";  
String namaBelakang = "Vetian";  
  
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.

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.

```
--- exec:3.1.0:exec (default-cli) @ Pertemuan2 ---
Nama saya Rifky Vetian
-----
BUILD SUCCESS
-----
Total time:  1.378 s
Finished at: 2025-10-02T01:32:18+07:00
-----
```

Kode di atas akan menghasilkan: `Nama saya Rifky Vetian`