

The logo for Oracle Academy is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by two horizontal dark gray bars, one at the top and one at the bottom.

# ORACLE

## Academy

# Yayasan Jawa

3-5

Masukan Papan Ketik

**ORACLE**  
Academy



Hak Cipta © 2022. Oracle dan/atau afiliasinya, Oracle, Java, dan MySQL adalah merek dagang terdaftar dari Oracle dan/atau afiliasinya. Nama lain mungkin merupakan merek dagang dari pemiliknya masing-masing.

# Tujuan

- Pelajaran ini mencakup tujuan-tujuan berikut:

- Memahami masukan pengguna
- Buat JOptionPane untuk mengumpulkan input pengguna
- Gunakan Pemindai untuk mengumpulkan masukan dari konsol
- Gunakan Pemindai untuk mengumpulkan input dari file
- Memahami bagaimana Pemindai menangani token dan pembatas



## Mengapa Anda Harus Mendapatkan Input Pengguna?

- Saat Anda menetapkan nilai ke variabel secara manual, ini dikenal sebagai nilai hard-coding:

```
Masukan string = "Ini adalah String";
```

- Anda dapat dengan mudah mengubah nilai hard-code karena Anda memiliki kode sumber dan Java IDE:

```
Input string = "Ini adalah String yang berbeda";
```

• Tetapi saat Anda mendistribusikan perangkat lunak, pengguna Anda tidak akan memiliki kemewahan yang sama

## Jenis Masukan Pengguna

- Contoh input pengguna termasuk ...

- Menekan tombol pada pengontrol permainan

- Memasukkan alamat pada GPS
  - Memasukkan angka

- dan fungsi ke dalam kalkulator
  - Memberitahukan nama Anda kepada

- orang lain
- Namun tanpa input pengguna ...

- Kapan game akan membuat karakter Anda melompat?

- Di mana GPS Anda akan memandu Anda?

- Angka berapa yang akan dihitung oleh kalkulator Anda?

- Apa yang akan orang panggil Anda?

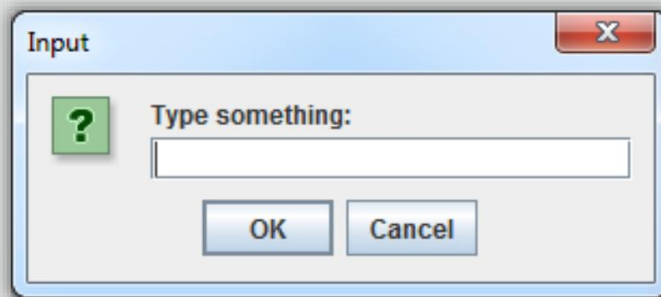
## Cara Mendapatkan Masukan Pengguna

- Ada banyak cara untuk mendapatkan input pengguna:
  - Tombol (fisik atau virtual)
  - Roda dan kenop
  - Pengenalan suara
  - Kotak dialog teks
  - File properti
- Java menawarkan banyak cara untuk mendapatkan input pengguna, termasuk
  - ...
  - Ayunkan JOptionPane
  - JavaFX (penerus Swing, dibahas nanti)
  - Pemindai

# JOptionPane

- Ini adalah cara sederhana untuk mendapatkan masukan dari pengguna:

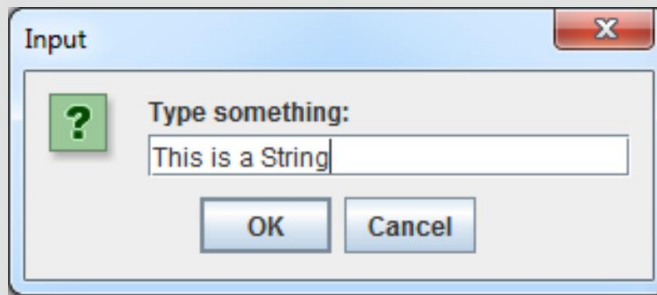
```
JOptionPane.showInputDialog("Ketik sesuatu:");
```



## JOptionPane Mengembalikan String

- Input dapat disimpan sebagai String:

```
Input string = JOptionPane.showInputDialog("Ketik sesuatu:");
```



- Ini setara dengan menulis:

```
Masukan string = "Ini adalah String";
```



## Latihan 1, Bagian 1

- Buat proyek baru dan tambahkan file Input01.java ke proyek
- Buat JOptionPane: NetBeans akan komplain mengikuti saran NetBeans untuk mengimpor javax.swing.JOptionPane Kita akan membahas pengimporan di bagian lain

## Latihan 1, Bagian 2

- Simpan input ini sebagai sebuah String
- Cetak variabel String
- Parse String sebagai variabel int yang terpisah
  - Anda harus memasukkan nilai yang dapat diuraikan
  - Cetak nilai ini +1
- Coba buat kotak dialog, parsing, dan inisialisasi int dalam satu baris
- Anda seharusnya hanya memiliki satu titik koma (;)

## Kode Terkondensasi

- Anda dapat menyebarkan input, parsing, dan penghitungan di beberapa baris:

```
String inputString = JOptionPane.showInputDialog("??"); int masukan
= Integer.parseInt(inputString); masukan++;
```

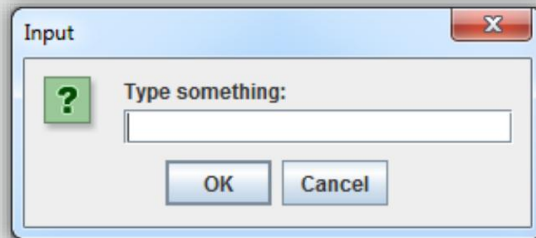
- Atau padatkan ini menjadi satu baris:

```
int masukan = Integer.parseInt(JOptionPane.showInputDialog("??")) +1;
```

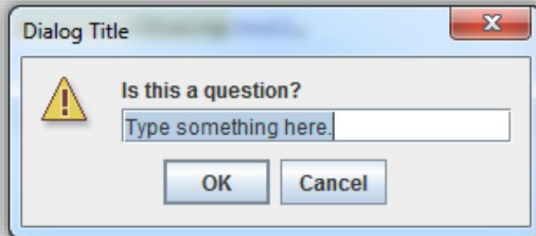
- Pilihan ini adalah masalah preferensi pribadi  
Tetapi jika Anda perlu mereferensikan nilai-nilai tertentu lagi nanti, akan sangat membantu untuk menyimpan nilai-nilai ini dalam sebuah variabel

## InputDialog yang berbeda

- Kami membuat InputDialog sederhana:



- Dengan kode yang lebih rumit, kita dapat mengkustomisasi InputDialog lainnya:



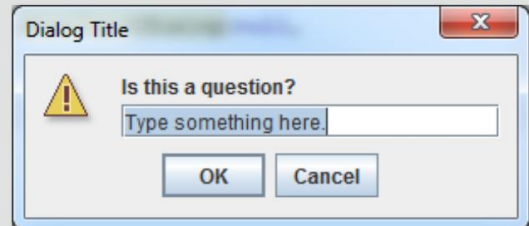
## Opsi Lainnya dengan InputDialogs

- Versi InputDialog ini tidak mengembalikan sebuah String

- Hasilnya harus dilemparkan ke a String yang dapat digunakan:

**Casting**

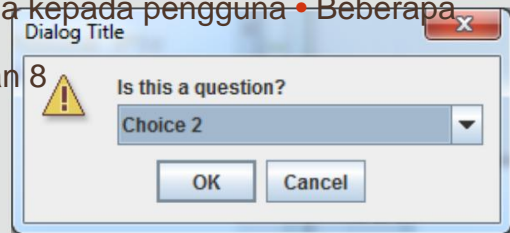
```
Input string = (String)JOptionPane.showInputDialog(null, "Apakah ini pertanyaan?", "Judul Dialog", 2, null, null, "Ketikkan sesuatu di sini.");
```



Bingung tentang kode ini? Jangan khawatir. Bahkan pemrogram berpengalaman pun bisa bingung saat melihat kode baru. Cara yang sangat membantu untuk mengembangkan pemahaman Anda adalah dengan memodifikasi kode yang ada dan melihat apa yang terjadi. Kita akan melakukannya di latihan berikutnya.

## Opsi Lainnya dengan InputDialogs

- Untuk menghindari masukan yang tidak diinginkan, dimungkinkan untuk hanya memberikan nilai yang dapat diterima kepada pengguna • Beberapa sintaks ini dibahas secara lebih rinci di Bagian 8

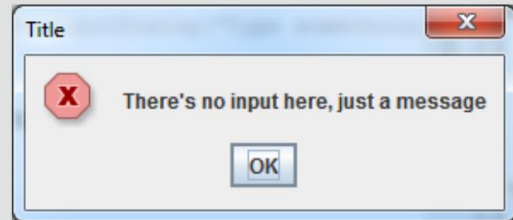


```
String[] AcceptableValues = {"Pilihan 1", "Pilihan 2", "Pilihan 3"}; Input2
string= (String)JOptionPane.showInputDialog(null, "Apakah ini pertanyaan?",
        "Judul Dialog", 2, null, AcceptableValues, AcceptableValues
        [1]);
```

# showMessageDialog

- A showMessageDialog tidak menyediakan kolom untuk input

- Ada banyak variasi lain dari JOptionPane



```
JOptionPane.showMessageDialog(  
    batal,  
    "Tidak ada input di sini, hanya pesan",  
    "Judul",  
    0);
```

## Latihan 2

- Buat proyek baru dan tambahkan file Input02.java ke proyek
- Eksperimen dengan kode dan coba ubah ...

• Judul pesan

• Pesan

• Setiap teks input default



• Ikon kotak dialog

- Mengurai, memanipulasi, dan mencetak input apa pun

Petunjuk: Abaikan nulls. Jika Anda memerlukan bantuan, dokumentasi Java mungkin berguna: <https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/javax/swing/package-summary.html>.



## Mendapatkan Masukan dengan Pemindai

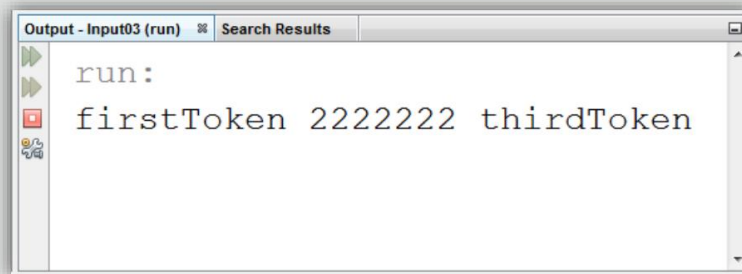
- Sebuah objek Pemindai membuka aliran untuk mengumpulkan masukan: `System.in` menyiapkan Pemindai untuk mengumpulkan masukan dari konsol `System.in` Ketikkan masukan Anda di jendela keluaran IDE Anda `System.in` Anda juga dapat menggunakan Pemindai tanpa IDE
- Praktek terbaik adalah menutup Pemindai streaming saat Anda selesai

```
public static void main(String[] args) { Scanner sc
    = new Scanner(System.in);

    sc.close(); }//
akhir metode utama
```

## Membaca Input dengan Pemindai

- Pemindai mencari token
- Token dipisahkan oleh pembatas  
• Pembatas default adalah spasi



The screenshot shows a Java IDE's output window titled "Output - Input03 (run)". The window contains the following text:

```
run:  
firstToken 2222222 thirdToken
```

# Kelas Pemindai

- Pemindai, seperti kelas lainnya, memiliki bidang dan metode •

Beberapa metode Pemindai yang berguna ...

nextInt() membaca token berikutnya sebagai int

nextDouble() membaca token berikutnya sebagai double

next() membaca token berikutnya sebagai String

```
public static void main(String[] args) { Scanner sc
    = new Scanner(System.in); x = sc.nextInt(); int
    sc.next() y = sc.nextDouble(); String z =
```

## Latihan 3

- Buat proyek baru dan tambahkan file Input03.java ke proyek • Buat Pemindai:

• IDE Anda akan mengeluh • Ikuti

saran IDE Anda untuk mengimpor `java.util.Scanner` • Ingat untuk menutup Pemindai

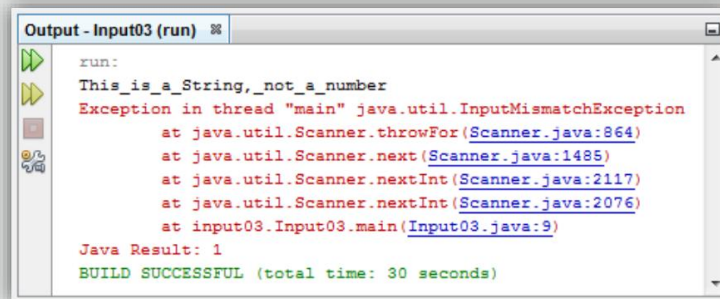
- Gunakan `Scanner` dan `System.in` untuk menulis program yang ...

• Menemukan dan mencetak jumlah dari tiga bilangan bulat yang dimasukkan oleh pengguna

- Coba masukkan kurang dari tiga token •

Coba masukkan token yang tidak dapat diuraikan sebagai `int`

# Pengecualian: InputMismatchException



```

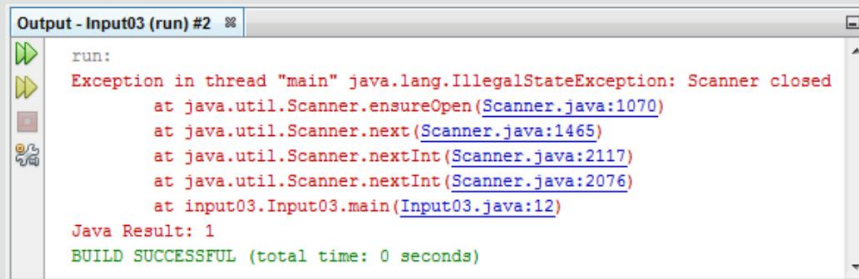
run:
This_is_a_String,_not_a_number
Exception in thread "main" java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at input03.Input03.main(Input03.java:9)
Java Result: 1
BUILD SUCCESSFUL (total time: 30 seconds)
  
```

- Terjadi karena input tidak dapat diuraikan sebagai tipe yang diharapkan:

```

public static void main(String[] args) { Scanner sc
    = new Scanner(System.in);
    System.out.println(sc.nextInt()); sc.close(); }//
akhir metode utama
  
```

# Pengecualian: IllegalStateException

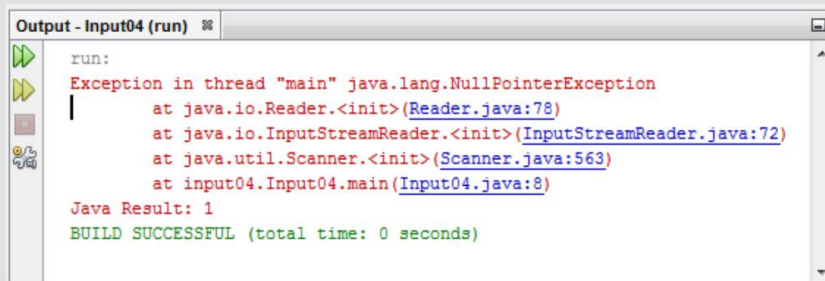


```
run:
Exception in thread "main" java.lang.IllegalStateException: Scanner closed
    at java.util.Scanner.ensureOpen(Scanner.java:1070)
    at java.util.Scanner.next(Scanner.java:1465)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at input03.Input03.main(Input03.java:12)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Terjadi karena aliran diakses setelah itu tertutup:

```
public static void main(String[] args) { Scanner sc =
    new Scanner(System.in); sc.close();
    System.out.println(sc.nextInt()); } //akhiri metode
utama
```

## Pengecualian: NullPointerException



```
Output - Input04 (run)
run:
Exception in thread "main" java.lang.NullPointerException
    at java.io.Reader.<init>(Reader.java:78)
    at java.io.InputStreamReader.<init>(InputStreamReader.java:72)
    at java.util.Scanner.<init>(Scanner.java:563)
    at input04.Input04.main(Input04.java:8)
Java Result: 1
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Terjadi karena "fakeFile.txt" tidak ada, lupa ekstensi .txt juga merupakan kesalahan umum

```
public static void main(String[] args) {
    Pemindai sc = Pemindai baru (
        Input04.class.getResourceAsStream("fakeFile.txt"));
    sc.close(); } //akhiri metode utama
```

Remember the extension

# Membaca dari File

•Java menawarkan beberapa cara untuk

membaca file •Metode Pemindai yang lebih berguna

meliputi: `nextLine()` memajukan Pemindai ini melewati baris saat ini dan mengembalikan input yang dilewati

`findInLine("StringToFind")` Mencoba menemukan input berikutnya

terjadinya pola yang dibangun dari String yang ditentukan, mengabaikan pembatas

**public static void main(String[] args) {**

**Pemindai sc = Pemindai baru (**

**Input04.class.getResourceAsStream("fakeFile.txt")); **int** x =  
sc.nextInt(); String seluruhLine = sc.nextLine(); sc.close(); } //akhiri  
metode utama**



## Latihan 4, Bagian 1

- Buat proyek baru dan tambahkan file Input04.java ke proyek
- Jalankan kode dan periksa hasilnya
- Baca setiap baris berikutnya sampai Anda menemukan "BlueBumper"
- Dua angka setelah "**BlueBumper**" adalah xPosition dan yPosition objek. Simpan koordinat ini sebagai bilangan bulat dan cetak
- Periksa input04text.txt, jika perlu

## Latihan 4, Bagian 2

- Periksa Level05.txt jika Anda penasaran:

• Ini adalah bagaimana data level disimpan untuk Java Puzzle Ball

• Membaca dan mengurai data level sedikit lebih rumit daripada yang telah Anda lakukan dalam latihan ini

• Tetapi jika Anda menyelesaikan latihan ini, Anda hampir memahami cara melakukannya

# Ringkasan

- Dalam pelajaran ini, Anda seharusnya telah mempelajari cara untuk:

- Memahami masukan pengguna
- Buat JOptionPane untuk mengumpulkan input pengguna
- Gunakan Pemindai untuk mengumpulkan input dari konsol
- Gunakan Pemindai untuk mengumpulkan input dari file
- Memahami bagaimana Pemindai menangani token dan pembatas



The Oracle Academy logo is centered on a light gray background. It features the word "ORACLE" in a bold, orange, sans-serif font. Below it, the word "Academy" is written in a smaller, dark gray, sans-serif font. The entire logo is framed by a thin black border, with dark gray horizontal bars at the top and bottom.

**ORACLE**  
Academy