

Pemrograman Berorientasi Objek

Java Sebagai Bahasa yang mendukung OOP dan implementasi algoritma

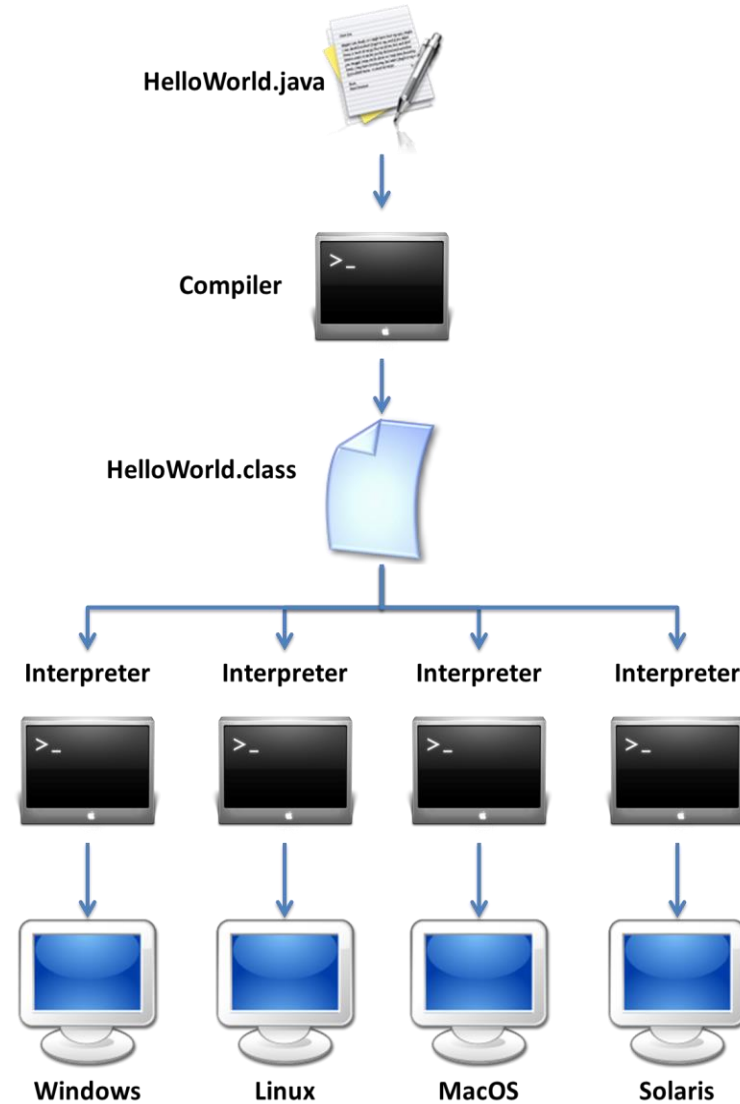
Bahasan

- Pengenalan Java Sebagai Bahasa yang mendukung OOP
- Variabel
- Assignment
- Input/ Output

Kelebihan Bahasa Java

- Multiplatform
- Open Source
- OOP
- Keamanan yang Baik
- Mendukung Multithreading
- Bergaya C++

Cara Kerja Bahasa Java



Perangkat Lunak yang Dibutuhkan

- *Java Development Kit (JDK)*
- Text Editor
 - NotePad++, UltraEdit, JCreator, Gel, dsb
- *Software IDE (Integrated Development Environment)*
 - IDE Netbeans dan IDE Eclipse

Contoh Program Java Pertama

```
public class HelloWorld{  
    public static void main (String[]args){  
        System.out.println("Hello Selamat Belajar");  
    }  
}
```

Tipe Data

- Tipe Data Dasar / Primitif
 - boolean (tipe data logika)
 - char (tipe data karakter)
 - byte, short, int dan long (tipe data bilangan bulat)
 - double dan float (tipe data bilangan pecahan/ *floating point*)

Dalam java, String bukanlah suatu tipe data melainkan suatu kelas.

Tipe Data boolean

- Hanya akan menghasilkan nilai benar dan salah (*true* atau *false*)
- Nilai ***default*** dari tipe data boolean adalah ***false***

```
boolean hasil; // awalnya hasil bernilai false
hasil = true; // hasil sekarang bernilai true
hasil = (5<4); // hasil sekarang bernilai false
if(5+3 != 2*4) // pernyataan ini akan menghasilkan nilai false
    System.out.println("aksi true");
else
    System.out.println("aksi false"); // tulisan ini akan tampil
```


Tipe Data char

- **Karakter** adalah semua anggota/tombol yang ada pada keyboard, meliputi :
 - Huruf kecil dan besar/kapital: 'a'-'z' dan 'A'-'Z'
 - Angka: '0'-'9'
 - Tombol fungsi: F1-F12
 - Simbol-simbol khusus: '\$', '%', '/', '*', '&', '+', dst.
 - Tombol perintah: Space, Enter, Esc, Home, End, dst.
- 'A' ≠ 'a', 'B' ≠ 'b', dst.**

Tipe Data char (2)

Ciri khas tipe data char adalah selalu diapit dengan tanda petik satu/*single quotes* (').

```
char index;  
index = 'a'; //index bernilai 'a'  
index = '\n'; //index bernilai '\n' (pindah baris)  
boolean cek = ('a'=='A'); //hasil perbandingan  
                bernilai false
```

| |
|--|
| <ul style="list-style-type: none">\n digunakan untuk pindah baris\r digunakan untuk menyatakan nilai balik\b digunakan sebagai backspace\t digunakan sebagai tab (tabulasi)\' digunakan untuk menampilkan tanda ' (petik satu)\\" digunakan untuk menampilkan tanda " (petik dua) |
|--|

Tipe data byte, short, int, long

| Tipe Data | Panjang | Jangkauan |
|-----------|---------|--------------------------|
| byte | 8-bits | -2^7 s/d 2^7-1 |
| short | 16-bits | -2^{15} s/d $2^{15}-1$ |
| int | 32-bits | -2^{31} s/d $2^{31}-1$ |
| long | 64-bits | -2^{63} s/d $2^{63}-1$ |

```
byte  bulat1 = 25;
short bulat2 = 0774;
int   bulat3 = 0x456B;
long  bulat4 = -304986;
System.out.println(bulat1); //output: 25
System.out.println(bulat2); //output: 508
System.out.println(bulat3); //output: 17771
System.out.println(bulat4); //output: -304986
```

double dan float

‘E’/‘e’ (*exponential*), ‘F’/‘f’ (float) dan ‘D’/‘d’ (double)

| |
|---------------------------------------|
| 5.12 //bilangan pecahan sederhana |
| 4.03E12 //bilangan pecahan yang besar |
| 9.876F //bilangan pecahan float |
| 987.4E+40D //bilangan pecahan double |

| Type Data | Panjang | Jangkauan |
|-----------|---------|--------------------------|
| float | 32-bits | -2^{31} s/d $2^{31}-1$ |
| double | 64-bits | -2^{63} s/d $2^{63}-1$ |

```
float pecahan1 = (22/7);  
double pecahan2 = 987.4E+40D;  
System.out.println(pecahan1); //output: 3.0  
System.out.println(pecahan2); //output: 9.874E42
```

String

- **String** adalah kumpulan dari karakter
- String selalu diapit oleh tanda *double quotes*/ petik dua(").

```
String nim  = "10506357"; //cara pertama  
String nama = new String("Hardiyana"); //cara kedua  
System.out.println(nim);    //output: 10506357  
System.out.println(nama);    //output: Hardiyana
```

Java *Identifier*

Java *identifier* adalah nama yang mewakili class, interface, atribut, metode, argument, dsb

Aturan penamaan (*identifier*) yang bersifat umum, yaitu:

1. Tidak boleh sama dengan daftar *keyword* dalam bahasa Java
2. Tidak boleh menggunakan space dan simbol-simbol khusus
3. Harus selalu diawali dengan huruf
4. Tidak boleh menggunakan *identifier* yang sudah pernah dideklarasikan
5. Boleh menggunakan karakter *underscore* (`_`), tapi ini tidak direkomendasikan
6. Boleh menggunakan karakter angka, tapi tidak boleh digunakan diawal
7. Tidak direkomendasikan menggunakan *identifier* dengan huruf kapital semua.
8. Jika *identifier* memiliki dua atau lebih kata, maka huruf awal pada kata tersebut adalah huruf besar. Contoh: `IniAdalahContohClass`, `iniAdalahContohMetode`

Java *Identifier* (2)

```
nim (BENAR)
namaMahasiswa (BENAR)
jenis_Obat (BENAR, TAPI TIDAK DIREKOMENDASIKAN)
1Kelas (SALAH)
_1 (BENAR JIKA IDENTIFIER BUKAN CLASS, TAPI TIDAK
    DIREKOMENDASIKAN)
_ (BENAR JIKA IDENTIFIER BUKAN CLASS, TAPI TIDAK
    DIREKOMENDASIKAN)
LUAS (BENAR, TAPI TIDAK DIREKOMENDASIKAN)
Tinggi$ (SALAH)
```

Java *Identifier* (3)

Aturan penamaan (*identifier*) yang bersifat khusus, yaitu:

1. ***Identifier class***

- a. Huruf awal harus diawali dengan huruf kapital/besar
- b. Nama class yang dideklarasikan sebagai public harus sama dengan nama file

Contoh: NamaKelas, PraktikumPertama, dsb.

2. ***Identifier atribut, variable dan argument/parameter***

Huruf awal harus diawali dengan huruf kecil

Contoh: x, tinggiPersegi, phi, dsb

3. ***Identifier metode***

- a. Huruf awal harus diawali dengan huruf kecil
- b. Nama metode selalu diakhiri dengan tanda kurung buka dan tutup ('namaMetode()')

Contoh: metodeUtama(), cetak(), inputData(), dsb

Komentar dalam Bahasa Java

- Penulisan komentar bergaya C++
hanya berlaku untuk **satu baris** saja. Diawali dengan garis miring 2x (//)

```
int x; // tulisan ini adalah komentar  
      tulisan ini bukan komentar, bisa menyebabkan ERROR
```

- Penulisan komentar bergaya C
berlaku untuk **lebih dari satu baris**.
diawali dengan /* dan diakhiri dengan */

```
/* ini adalah komentar baris ke-1  
   ini adalah komentar baris ke-2  
   ini adalah komentar baris ke-N */
```

- Penulisan komentar bergaya javadoc
bisa berlaku untuk **lebih dari satu baris**
diawali dengan /** dan diakhiri dengan */

```
/** ini adalah komentar baris ke-1  
    ini adalah komentar baris ke-2  
    ini adalah komentar baris ke-N */
```

Statement dan Blok dalam Bahasa Java

- **Statement** adalah suatu aksi yang dijalankan oleh komputer. Ditandai dengan ***semicolon (;)***.
- **Blok** adalah satu/lebih statement yang ditulis dalam **kurung kurawal ('{}')**.

```
public static void main(String[] args) {  
    System.out.print("Halo-Halo ");  
    System.out.print("Bandung");  
}
```

Struktur Dasar Bahasa Java

```
package alamatPaket;           (A)
//blok package ditulis (0-1)

import alamatPaketYgDiImport;  (B)
//blok import ditulis (0-N)

<modifier> class <NamaClass>{   (C)
    <modifier> <tiipeData> <namaAtribut> [=inisialisasi]; (D)
    //blok deklarasi atribut ditulis (0-N)

    <modifier> <NamaClass>(<argumen>){ (E)
    }
    //blok deklarasi constructor ditulis (0-N)

    <modifier> <tiipeData> <namaMethod>(<argumen>){ (F)
    }
    //blok deklarasi method ditulis (0-N)

    public static void main(String[] args){ (G)
    }
    //blok deklarasi method main ditulis (0-1)
}
```

| Point | Keterangan |
|-------|---|
| A | Package adalah folder/direktori tempat kita menyimpan file java kita. Package berfungsi jika kita ingin mengelompokkan class yang kita buat dalam suatu folder. Dalam suatu class terdapat (0-1) package. |
| B | Import adalah menjabarkan/mendaftarkan library-library yang akan digunakan. Dalam satu class terdapat (0-N) import. |
| C | Pendeklarasian class umumnya menggunakan “public class <NamaClass>” atau “class <NamaClass>”. Nama Class yang dideklarasikan sebagai public harus sama dengan nama file. Misal nama class: PraktikumSatu, maka nama file: PraktikumSatu.java |
| D | Atribut adalah ciri yang melekat yang dimiliki suatu objek. Dalam satu class bisa terdapat (0-N) atribut. Penulisan atribut ditulis dibawah pendeklarasian class. Metode akan dibahas lebih lanjut ketika materi objek. |
| E | Constructor adalah metode yang akan dipanggil ketika class tersebut dibuatkan objeknya (<i>instance</i>). Nama constructor harus sama dengan nama class-nya, misal nama class: Contoh, maka nama constructor: Contoh. Dalam satu class bisa terdapat (0-N) constructor. Penulisan constructor ditulis dibawah blok atribut. |
| F | Metode adalah kemampuan yang dimiliki suatu objek. Dalam satu class bisa terdapat (0-N) metode. Penulisan metode ditulis dibawah blok Constructor. Metode akan dibahas lebih lanjut ketika materi objek. |
| G | Metode main adalah metode utama. Artinya metode yang akan pertama kali dieksekusi oleh komputer adalah metode main ini. Penulisan metode main pasti selalu ditulis dengan sintaks “public static void main(String[] args)”. Dalam pembangunan aplikasi yang baik, metode main ini hanya dimiliki oleh satu class saja, yakni class utama. |

Variabel

Variabel adalah wadah/tempat untuk menyimpan data

```
<tipoData> <namaVar> [=inisialisasi];
```

NB: untuk <tipoData> dan <namaVar> bersifat wajib, sedangkan [=inisialisasi] bersifat opsional (boleh ditulis, boleh tidak).

```
public class ContohVariabel{  
    public static void main(String[] args){  
        String nama;  
        int nilai;  
        nama="Susi Similikiti";  
        nilai=100;  
        System.out.println("Nama    : "+nama);  
        System.out.println("Nilai   : "+nilai);  
    }  
}
```

TAMPILAN OUTPUT

```
Nama    : Susi Similikiti  
Nilai   : 100
```

Operator

- Operator Aritmetika (+ - * / %)
- Operator *Increment & Decrement* serta Operator Khusus
(++ -- += -= *= /= %=)
- Operator Perbandingan (> >= < <= == !=)
- Operator Logika (&& || ^ !)
- Operator Kondisi `(kondisi) ? aksiTrue: aksiFalse;`

Fungsi *Output*

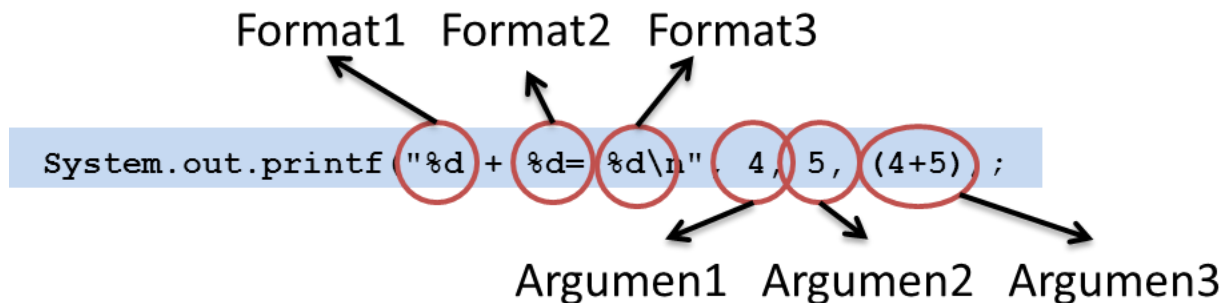
- *Output* berbasis DOS menggunakan printf()
- *Output* berbasis DOS menggunakan print()
- *Output* berbasis DOS menggunakan println()
- *Output* berbasis GUI menggunakan JOptionPane

Output Menggunakan printf()

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

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

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



TAMPILAN OUTPUT

4 + 5 = 9

Output Menggunakan printf() (2)

```
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);  
    }  
}
```

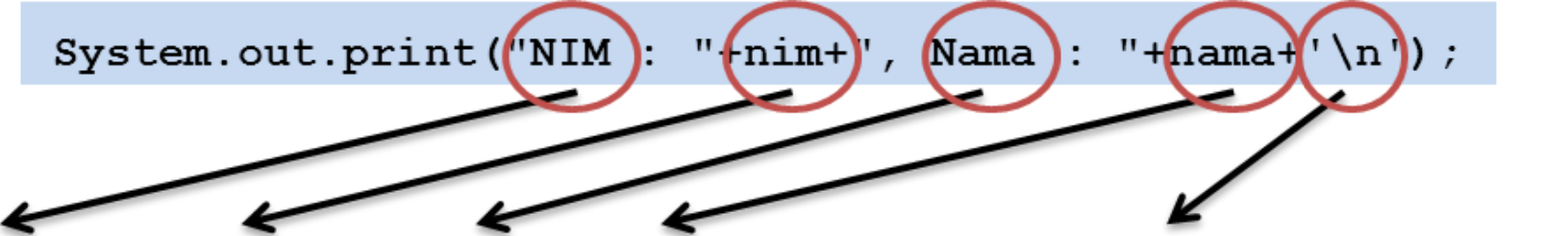
TAMPILAN OUTPUT

```
Halo Halo Bandung  
Halo Halo Bandung  
4 + 5= 9  
PHI= 3.140000  
index= A  
nilai boolean= true
```


Output Menggunakan print()

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

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



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

TAMPILAN OUTPUT

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

Output Menggunakan print() (2)

```
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');  
    }  
}
```

TAMPILAN OUTPUT

```
Halo Halo Bandung  
NIM   : 10506357  
Nama  : Phantom Assassin  
NIM : 10506357, Nama : Phantom Assassin
```

Output Menggunakan println()

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

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



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

TAMPILAN OUTPUT

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

Output Menggunakan println() (2)

```
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);  
    }  
}
```

TAMPILAN OUTPUT

```
Halo Halo  
Bandung  
NIM   : 10506357  
Nama  : Phantom Assassin  
NIM : 10506357, Nama : Phantom Assassin
```

Output Menggunakan JOptionPane

Berikut ini adalah langkah-langkah yang harus dilakukan :

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{
}
```

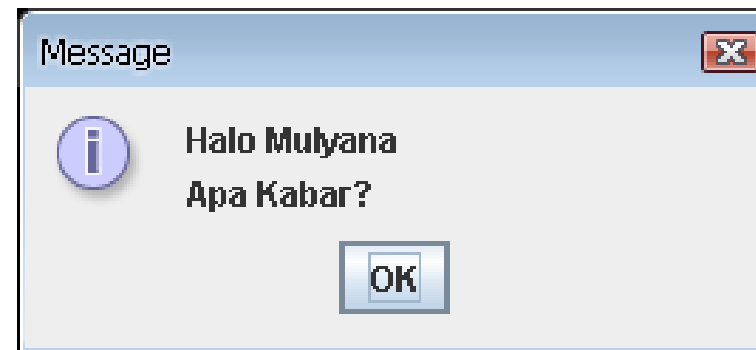
2. Memanggil metode showMessageDialog().

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

Output Menggunakan JOptionPane (2)

```
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?");
    }
}
```



Fungsi Input

- *Input* berbasis DOS menggunakan Scanner
- *Input* berbasis DOS menggunakan BufferedReader
- *Input* berbasis GUI menggunakan JOptionPane

Input menggunakan Scanner

1. Meng-***import*** kelas Scanner yang terdapat pada paket **java.util**

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

2. Membuat objek referensi sebagai media penginputan data.

```
Scanner objekReferensi = new Scanner(System.in);
```

Contoh :

```
Scanner input = new Scanner(System.in);  
Scanner scan = new Scanner(System.in);
```

3. Memanggil metode khusus untuk melakukan inputan data melalui objek referensi yang tadi dibuat.

Input menggunakan Scanner (2)

| Metode | Berfungsi untuk menampung |
|------------------|-----------------------------------|
| nextBigDecimal() | Nilai bilangan pecahan BigDecimal |
| nextBigInteger() | Nilai bilangan bulat BigInteger |
| nextBoolean() | Nilai berupa boolean |
| nextByte() | Nilai bilangan bulat byte |
| nextDouble() | Nilai bilangan pecahan double |
| nextFloat() | Nilai bilangan pecahan float |
| nextInt() | Nilai bilangan bulat int |
| nextLine() | Data berupa String |
| nextLong() | Nilai bilangan bulat long |
| nextShort() | Nilai bilangan bulat short |

Contoh :

```
Scanner input = new Scanner(System.in);  
String nama = input.nextLine();  
int alas = input.nextInt();  
float tinggi = input.nextFloat();  
boolean status = input.nextBoolean();
```

Input menggunakan Scanner (2)

```
import java.util.Scanner;

public class ContohInScanner{
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Input Nama : ");
        String nama = input.nextLine();
        System.out.println("Halo "+ nama);
        System.out.println();
        System.out.print("Alas    : ");
        int alas = input.nextInt();
        System.out.print("Tinggi : ");
        float tinggi = input.nextFloat();
        System.out.println("Luas Segitiga : "+ (alas*tinggi)/2);
    }
}
```

TAMPILAN OUTPUT

```
Input Nama : Kang Kabayan [enter]
Halo Kang Kabayan

Alas    : 4 [enter]
Tinggi : 8 [enter]
Luas Segitiga : 16.0
```

Input menggunakan BufferedReader

1. Meng-***import*** kelas **InputStreamReader**, **IOException** dan **BufferedReader**.

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

2. Membuat objek referensi sebagai media penginputan data.

```
BufferedReader objekReferensi = new  
BufferedReader(new InputStreamReader(System.in));
```

Contoh :

```
BufferedReader input = new  
BufferedReader(new InputStreamReader(System.in));  
BufferedReader read = new  
BufferedReader(new InputStreamReader(System.in));
```

Input menggunakan BufferedReader (2)

3. Memanggil metode **readLine()** melalui objek referensi tadi. Tapi perlu diingat, karena **hasil buffer** berupa **String** dan jika kita ingin mengubahnya dalam bentuk tipe data yang lain, maka harus melakukan ***parsing (mengkonversi data)***.

| Konversi ke- | Cara konversi |
|--------------|------------------------------|
| boolean | Boolean.parseBoolean(String) |
| float | Float.parseFloat(String) |
| double | Double.parseDouble(String) |
| byte | Byte.parseByte(String) |
| short | Short.parseShort(String) |
| int | Integer.parseInt(String) |
| long | Long.parseLong(String) |

Contoh :

```
double x = Double.parseDouble(input.readLine());  
int y = Integer.parseInt(input.readLine());
```

Input menggunakan BufferedReader (3)

4. Menuliskan poin 2 dan 3 didalam blok **try-catch** yang akan ditangkap oleh **IOException**.

```
try {  
    BufferedReader input = new  
        BufferedReader(new  
        InputStreamReader(System.in));  
    System.out.print("Input Nilai X : ");  
    double x =  
        Double.parseDouble(input.readLine());  
} catch (IOException e) {  
}
```

Input menggunakan BufferedReader (4)

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

public class ContohInBufferedReader {
    public static void main(String[] args){
        double nilai1;
        int nilai2;
        try {
            BufferedReader input = new
                BufferedReader(new InputStreamReader(System.in));
            System.out.print("Input Nilai 1 : ");
            nilai1 = Double.parseDouble(input.readLine());
            System.out.print("Input Nilai 2 : ");
            nilai2 = Integer.parseInt(input.readLine());
            System.out.println(nilai1+" "+nilai2+"="+(nilai1+nilai2));
            System.out.println(nilai1+"-"+nilai2+"="+(nilai1-nilai2));
            System.out.println(nilai1+"*"+nilai2+"="+(nilai1*nilai2));
            System.out.println(nilai1+"/"+nilai2+"="+(nilai1/nilai2));
        } catch (IOException e) {
        }
    }
}
```

TAMPILAN OUTPUT

```
Input Nilai 1 : 10.5 [enter]
Input Nilai 2 : 2 [enter]
10.5+2=12.5
10.5-2=8.5
10.5*2=21.0
10.5/2=5.25
```

Input menggunakan JOptionPane

1. Meng-***import*** kelas **JOptionPane** yang ada di paket **javax.swing**.

```
import javax.swing.JOptionPane;
```

2. Membuat objek referensi dari kelas String. lalu memanggil metode **showInputDialog()** melalui objek referensi tadi.

```
String objekReferensi = JOptionPane.showInputDialog(null, String);
```

Contoh :

```
String input = JOptionPane.showInputDialog(null,  
"Contoh tulisan");  
String readJOp = JOptionPane.showInputDialog(null,  
"Masukan data");
```

Input menggunakan JOptionPane (2)

3. Hasil pemanggilan **showInputDialog()** berupa **String**. Sehingga, jika kita ingin mengubahnya dalam bentuk tipe data yang lain, maka harus melakukan ***parsing*** (**mengkonversi data**).

```
String input = JOptionPane.showInputDialog(null,  
"Input Bilangan");  
double x = Double.parseDouble(input);
```


Input menggunakan JOptionPane (3)

```
import javax.swing.JOptionPane;

public class ContohInJOptionPane{
    public static void main(String[] args){
        String input = JOptionPane.showInputDialog(null, "Input NIM");
        String nim = input;

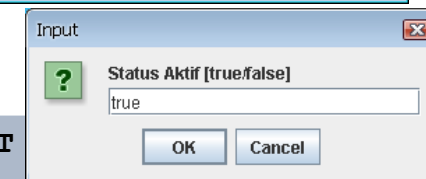
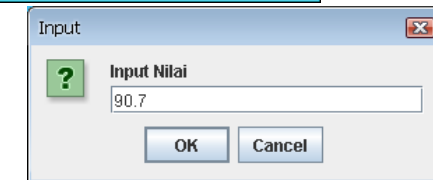
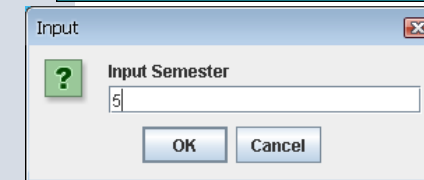
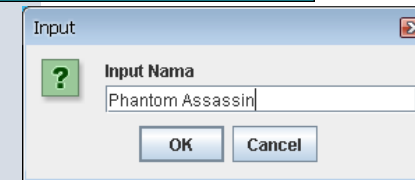
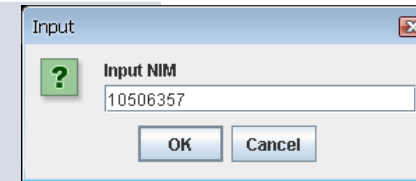
        input = JOptionPane.showInputDialog(null, "Input Nama");
        String nama = input;

        input = JOptionPane.showInputDialog(null, "Input Semester");
        int smt = Integer.parseInt(input);

        input = JOptionPane.showInputDialog(null, "Input Nilai");
        double nilai = Double.parseDouble(input);

        input = JOptionPane.showInputDialog(null, "Status Aktif [true/false]");
        boolean status = Boolean.parseBoolean(input);

        System.out.println("NIM          : "+ nim);
        System.out.println("Nama          : "+ nama);
        System.out.println("Semester      : "+ smt);
        System.out.println("Nilai         : "+ nilai);
        System.out.println("Status aktif  : "+ status);
    }
}
```



TAMPILAN OUTPUT

```
NIM          : 10506357
Nama          : Phantom Assassin
Semester      : 5
Nilai         : 90.7
Status aktif  : true
```

Terima Kasih