PENGENALAN JAVA (2)

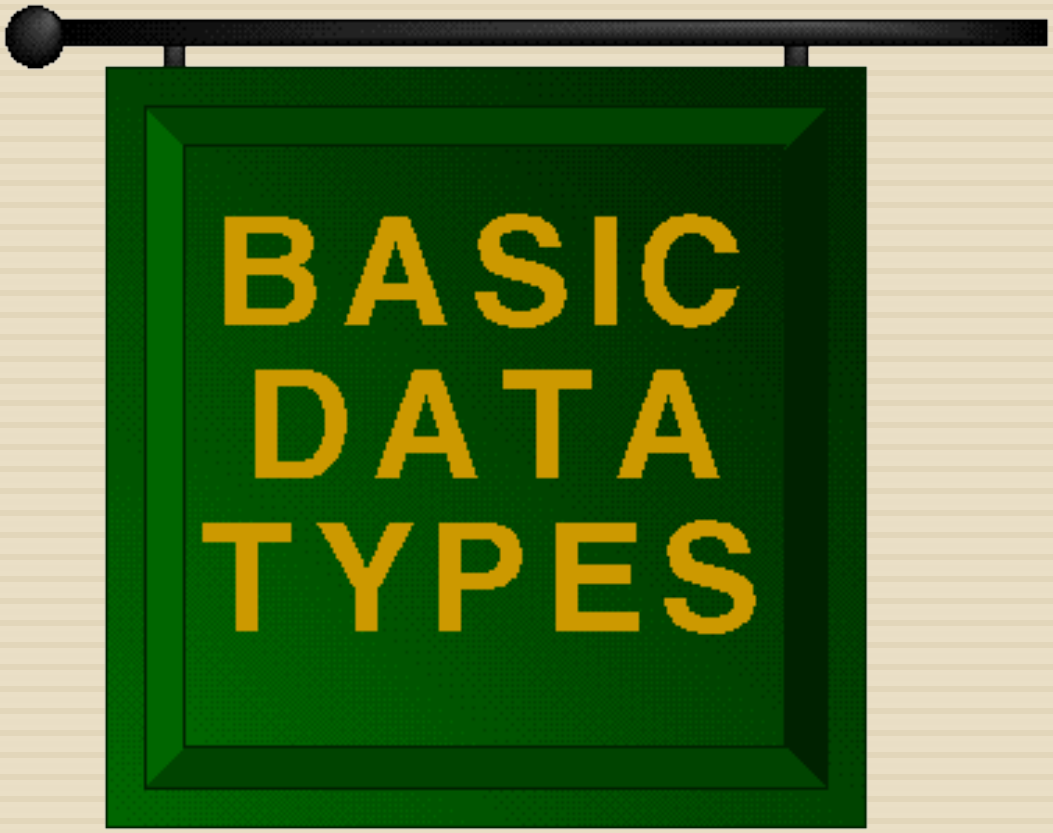
http://www.brigidaarie.com

Sub Pokok Bahasan

- Tipe data
- Variabel
- Reader input
- Java output
- Formating output in java

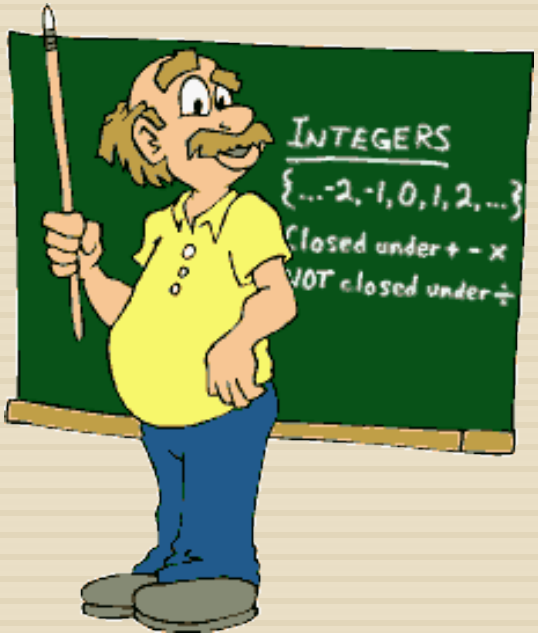
Pengelompokan tipe data dalam Java

- Integer
  - Byte
  - Short
  - Int
  - Long
- Floating Point
  - Float
  - Double
- Char
- Boolean



Integer (Bilangan Bulat)

Tipe Data	Ukuran (dalam bit)	Rentang
Byte	8	-128 sampai 127
Short	16	-32.786 sampai 32.767
Int	32	-2.147.483.648 sampai 2.147.483.647
long	64	-9.223.372.036.854.775.808 sampai 9.223.372.036.854.775.807



Integer (Bilangan Bulat)

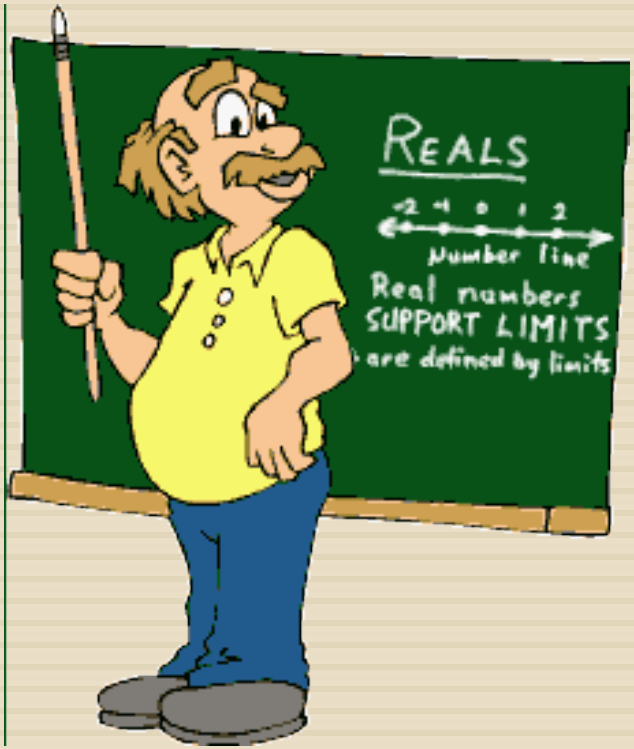
```
public class Contohperhitungan {
public static void main(String[] args) {
    byte a = 1;
    short b = 2;
    int c = 3,d;
    d = a+b+c;
    System.out.println("Hasil = " + d);}
}
```

Konvesi

```
run:
Hasil = 6
BUILD SUCCESSFUL (total time: 0 seconds)
```

Floating-Point (Bilangan Riil)

Tipe Data	Ukuran (dalam bit)	Rentang
Float	32	3.4^{e-038} sampai 3.4^{e+038}
Double	64	1.7^{e-308} sampai 1.7^{e+308}



Floating-Point (Bilangan Riil)

```
public class KelilingLingkaran {
    public static void main(String[] args) {
        double pi = 3.1416;
        double r = 2.12;
        double keliling;
        keliling = 2*pi*r;
        System.out.println("Keliling Lingkaran = " + keliling);}
}
```

```
run:
Keliling Lingkaran = 13.320384
BUILD SUCCESSFUL (total time: 0 seconds)
```

Karakter

```
public class DemoKarakter {
    public static void main(String[] args) {
        char ch1 = 65;
        char ch2 = 'B';

        System.out.println ("ch1 =  " + ch1);
        System.out.println ("ch2 =  " + ch2);}
}
```

ASCII

```
run:
ch1 =  A
ch2 =  B
BUILD SUCCESSFUL (total time: 0 seconds)
```

Karakter

```
public class DemoKarakter2 {
    public static void main(String[] args) {
        char ch = 'A';

        System.out.println ("ch =  " + ch);

        ch++;

        System.out.println ("ch =  " + ch);}

}
```

Increment

```
run:
ch =  A
ch =  B
BUILD SUCCESSFUL (total time: 0 seconds)
```

Escape Sequence

Escape Sequence	Keterangan
\ddd	Karakter oktal (ddd)
\uxxxx	Karakter Unicode heksadesimal (xxxx)
\’	Petik tunggal
\”	Petik ganda
\\	Backslash
\r	Carriage return
\n	Baris baru (disebut juga dengan line feed)
\f	Form feed
\t	Tab
\b	Backspace

Escape Sequence

```java
public class DemoEscapeSequence {
    public static void main (String[] args){
        System.out.print ("Hari\t\t: Jum\'at\n");
        System.out.print ("Tanggal\t\t: 1 Februari 1987\n");
    }
}
```

```
run:
Hari            : Jum'at
Tanggal         : 1 Februari 1987
BUILD SUCCESSFUL (total time: 0 seconds)
```

Boolean

```
public class DemoBoolean {
    public static void main (String[] args){
        boolean b;
        b = true;
        System.out.println("Nilai b = " +b);
        if (b){
            System.out.println("Statemen ke-1 dieksekusi");
        }
        b = false;
        System.out.println ("Nilai b = " +b);

        if (b) {
            System.out.println ("Statemen ke-2 tidak akan dieksekusi");
        }
        if (!b) {
            System.out.println ("Statemen ke-3 dieksekusi");
        }
        System.out.println ("5 <= 10 mengembalikan nilai " + (5<=10));
        System.out.println("4 > 6 mengembalikan nilai " + (4>6));
    }
}
```

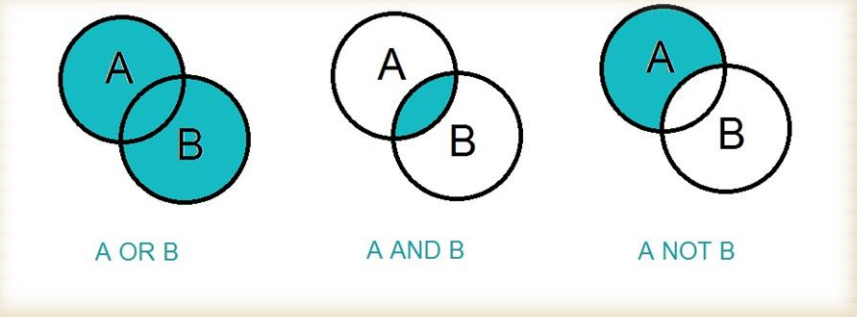
Boolean

```
run:
Nilai b = true
Statemen ke-1 dieksekusi
Nilai b = false
Statemen ke-3 dieksekusi
5 <= 10 mengembalikan nilai true
4 > 6 mengembalikan nilai false
BUILD SUCCESSFUL (total time: 0 seconds)
```

Boolean

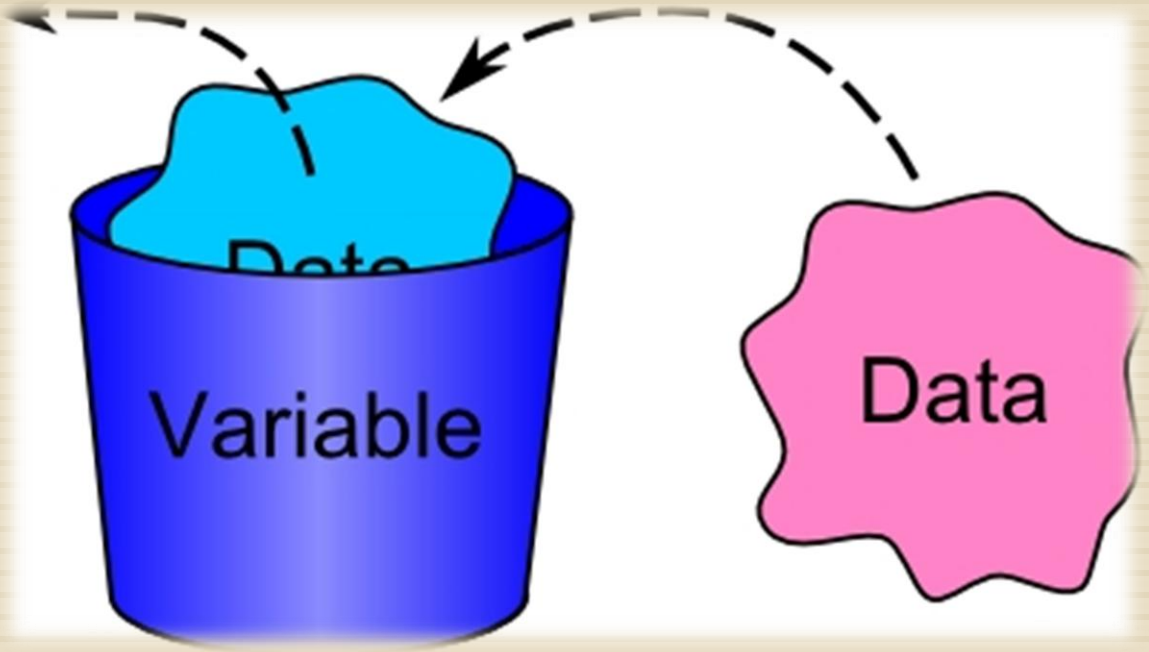
```
public class DemoBoolean2 {
    public static void main (String [] args){
        int index = 0;
        while (index < 5) {
            System.out.println ("Baris ke-" + (index+1));
            index++;
        }
    }
}
```

```
run:
Baris ke-1
Baris ke-2
Baris ke-3
Baris ke-4
Baris ke-5
BUILD SUCCESSFUL (total time: 1 second)
```



Variabel

- Unit dasar yang diperlukan untuk menyimpan suatu nilai dengan tipe data tertentu di dalam memori komputer



Mendeklarasikan Variabel

Tipe namavariabel;
Tipe namavariabel1, variabel2, …

- Tidak boleh mengandung spasi
- Tidak boleh berupa angka atau diawali oleh karakter numerik (angka)
- Tidak boleh mengandung karakter simbol, kecuali tanda $
- Tidak boleh menggunakan kata kunci yang sudah didefinisikan di dalam java
- Dalam satu program,nama variabel harus bersifat unik atau berbeda satu sama lain
- Usahakan nama variabel sedeskriptif mungkin sehingga kode program dapat lebih mudah untuk dipahami oleh orang lain (teruatam bila bekerja dalam tim)
- Selalu diawali dengan huruf kecil

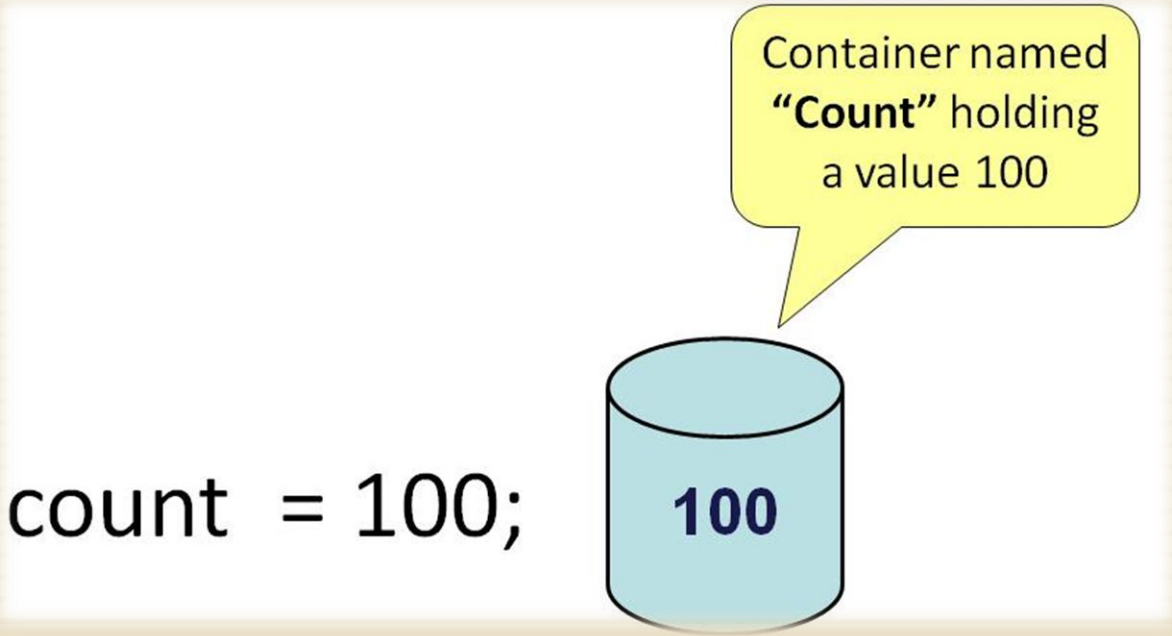
Inisialisasi Variabel

Tipe namavariabel = nilai;
Tipe namavariabel1 = nilai1, namavariabel2 = nilai2 …

Contoh :

Int index =0;

Char ch = 'A';



Lingkup dan Daur Hidup Variabel

```
10  public class LingkupVariabel {
11      public static void main (String[] args) {
12          int a = 10;
13
14          if (a>5){
15              int b =15;
16              System.out.println ("Nilai a di dalam blok if : " +a);
17              System.out.println ("Nilai b di dalam blok if : " +b);
18          }
19          System.out.println("Nilai a di luar blok if : " +a);
            System.out.println("Nilai b di luar blok if : " +b);
21      }
22  }
```

```
15
16
17
18
19
            System.out.prin
```

cannot find symbol
  symbol: variable b
  location: class LingkupVariabel

(Alt-Enter shows hints)

```
run:
C:\Users\user\AppData\Local\NetBeans\Cache\7.3.1\executor-snippets\run.xml:48:
Cancelled by user.
BUILD FAILED (total time: 2 seconds)
```

Typecasting

- Proses pemeranan (casting) tipe data dari satu tipe tertentu ke tipe data lainnya

Konversi otomatis dalam java

Syarat

- Kedua tipe data tersebut kompatibel (tipe data asal dan tipe data tujuan)
- Tipe data tujuan memiliki rentang lebih besar dari tipe data asal

Konversi tipe data yang tidak kompatibel

```
public class DemoKonversi {
  public static void main (String [] args) {
      int a = 257;
      double d = 274.5678;

      byte b;
      b = (byte) a;
      System.out.println ("Typecasting dari tipe int ke tipe byte");
      System.out.println("int : "+a);
      System.out.println ("byte : "+b);

      int x;
      x =(int)d;
      System.out.println("\nTypecasting dari tipe double ke tipe int");
      System.out.println ("double :" +d);
      System.out.println ("int : " +x);

      b = (byte)d;
      System.out.println("\nTypecasting dari tipe double ke tipe byte");
      System.out.println ("double :" +d);
      System.out.println ("byte : " +b);  }
}
```

Konversi Byte ke Integer : nilai integer/256

Konversi tipe data yang tidak kompatibel

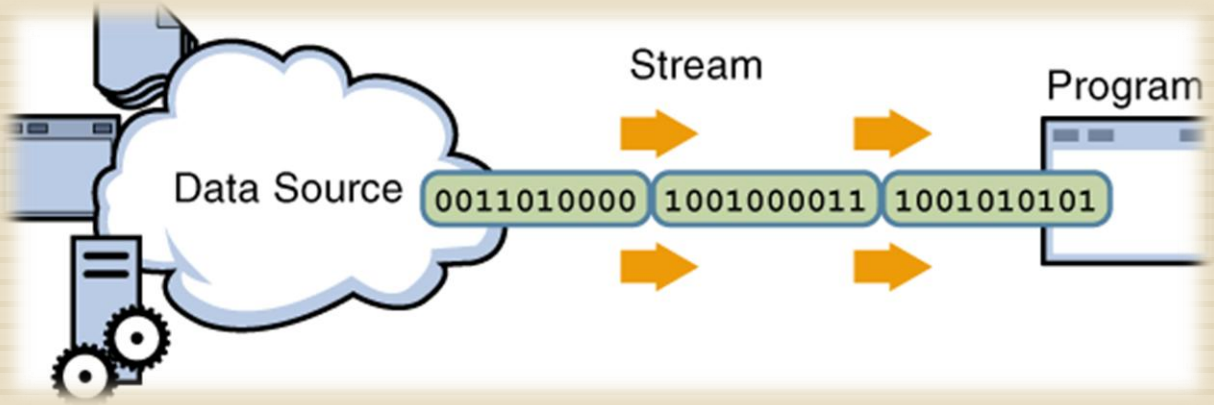
```
run:
Typecasting dari tipe int ke tipe byte
int : 257
byte : 1

Typecasting dari tipe double ke tipe int
double :274.5678
int : 274

Typecasting dari tipe double ke tipe byte
double :274.5678
byte : 18
BUILD SUCCESSFUL (total time: 0 seconds)
```

Input dan Output

- Java menyediakan dukungan terhadap proses I/O dengan menghadirkan paket **java.io**

Pengertian Stream

- Program Java melakukan proses I/O melalui stream
- Stream adalah sebuah abstraksi yang dapat memberikan atau mendapatkan informasi
- Stream dapat dihubungkan dengan peralatan fisik yang terdapat dalam sistem I/O Java, seperti : keyboard, file, layar console, dll



Tipe Stream

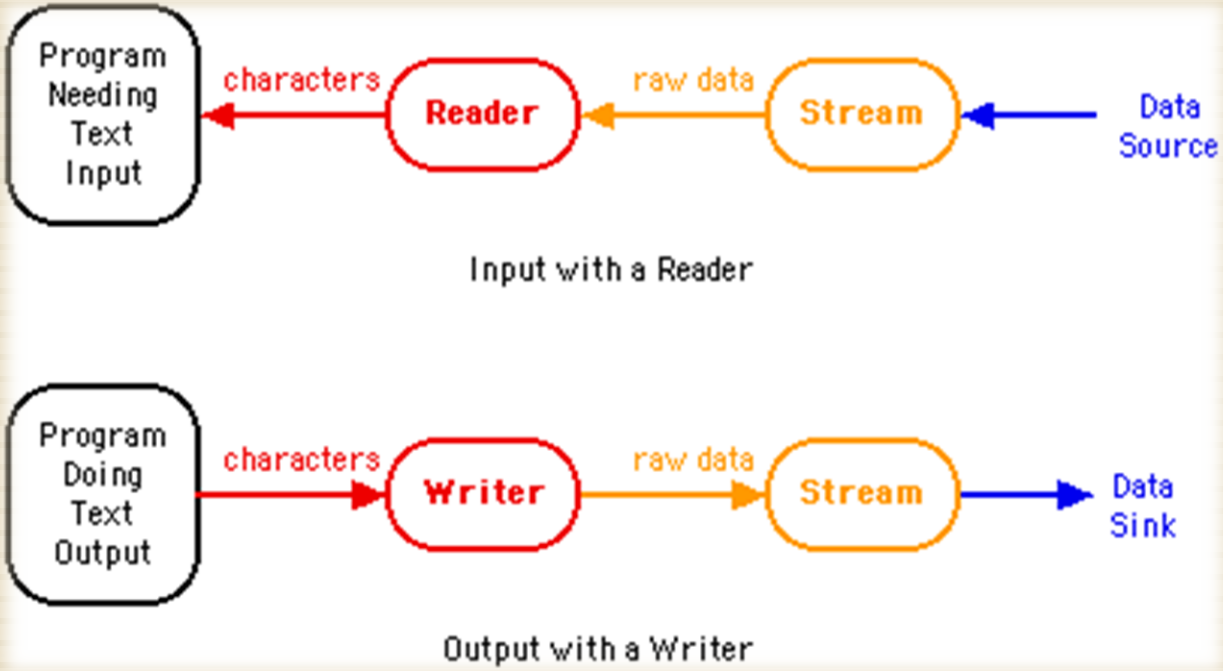
- Stream Byte
  - Memberikan atau menyimpan informasi data dalam bentuk byte. Ex : menulis dan membaca file biner.
- Stream Karakter
  - Melakukan proses I/O yang melibatkan data-data dalam bentuk karakter. Ex : proses baca/tulis ke file teks. Stream karakter menggunakan karakter unicode.

Kelas Abstrak Stream Byte

- Kelas abstrak yang dirancang sebagai kelas induk atau superclass untuk kelas-kelas yang termasuk ke dalam kategori stream byte
  - InputStream
  - OutputStream

Kelas Abstrak Stream Karakter

- Kelas abstrak yang akan diturunkan menjadi kelas-kelas baru yang termasuk ke dalam kategori stream karakter
  - Reader
  - Writer



Stream Byte

Nama Kelas	Keterangan
BufferedInputStream	Stream input yang telah ter-buffer
BufferedOutputStream	Stream output yang telah ter-buffer
ByteArrayInputStream	Stream input yang membaca dari array byte
ByteArrayOutputStream	Stream output yang membaca dari array byte
DataInputStream	Stream input yang berisi method-method untuk membaca tipe data standar yang terdapat dalam java
DataOutputStream	Stream output yang berisi method-method untuk membaca tipe data standar yang terdapat dalam java

Stream Byte (lanjutan)

Nama Kelas	Keterangan
FileInputStream	Stream input yang membaca dari sebuah file
FileOutputStream	Stream output yang membaca dari sebuah file
FilterInputStream	Mengimplementasikan input stream
FilterOutputStream	Mengimplementasikan output stream
InputStream	Kelas abstrak yang menjelaskan stream input
OutputStream	Kelas abstrak yang menjelaskan stream output
PipedInputStream	Penyalur input
PipedOutputStream	Penyalur output
PrintStream	Stream output yang berisi method println()

Stream Karakter

Nama Kelas	Keterangan
BufferedReader	Stream karakter input yang telah ter-buffer
BufferedWriter	Stream karakter output yang telah ter-buffer
CharArrayReader	Stream input yang membaca dari array karakter
CharArrayWriter	Stream output yang menulis ke array karakter
FileReader	Stream input yang membaca dari file
FileWriter	Stream output yang menulis ke file
InputStreamReader	Stream input yang menerjemahkan byte ke karakter
LineNumberReader	Stream input yang menghitung jumlah baris
OutputStreamWriter	Stream input yang menerjemahkan karakter ke byte
PipedReader	Penyalur input
PipedWriter	Penyalur output
PrintWriter	Stream output yang berisi method print () dan println()
Reader	Kelas abstrak yang menjelaskan stream karakter input

Stream Karakter (lanjutan)

Nama Kelas	Keterangan
StringReader	Stream input yang membaca dari sebuah string
StringWriter	Stream output yang menulis ke sebuah string
Writer	Stream output

Membaca Input Data Karakter

```java
import java.io.*;
public class DemoInputKarakter {
    public  static void main (String [] args) throws IOException{
        System.out.print ("Masukkan sembarang karakter : ");
        char ch;
        InputStreamReader isr = new InputStreamReader (System.in);
        BufferedReader br = new BufferedReader (isr);

        ch = (char) br.read();

        System.out.println ("Karakter yang dimasukkan adalah \'" + ch + "\'");
    }
}
```

```
run:
Masukkan sembarang karakter : t
Karakter yang dimasukkan adalah 't'
BUILD SUCCESSFUL (total time: 10 seconds)
```

Membaca Input Data String

```
import java.io.*;
public class DemoInputString {
    public  static void main (String [] args) throws IOException{
        System.out.print ("Masukkan nama Anda : ");
        String nama;

        InputStreamReader isr = new InputStreamReader (System.in);
        BufferedReader br = new BufferedReader (isr);

        nama = br.readLine();

        System.out.println ("Halo " + nama + " sudahkan Anda mengerti Java?");
    }
}
```

```
run:
Masukkan nama Anda : Tyas
Halo Tyas sudahkan Anda mengerti Java?
BUILD SUCCESSFUL (total time: 15 seconds)
```

Membaca Input Data Numerik

```
import java.io.*;
public class DemoInputNumerik {
    public  static void main (String [] args) throws IOException{
        System.out.print ("Masukkan sebuah bilangan bulat : ");
        String temp;
        int bilangan = 0;

        InputStreamReader isr = new InputStreamReader (System.in);
        BufferedReader br = new BufferedReader (isr);

        temp = br.readLine();

        try{
            bilangan = Integer.parseInt(temp);

        } catch (NumberFormatException nfe){
            System.out.println("Data yang dimasukkan bukan bilangan bulat");
            System.exit(1);
        }
        System.out.println ("Bilangan yang dimasukkan adalah " +bilangan);
    }
}
```

Membaca Input Data Numerik (lanjutan)

```
run:
Masukkan sebuah bilangan bulat : 5
Bilangan yang dimasukkan adalah 5
BUILD SUCCESSFUL (total time: 3 seconds)
```

```
run:
Masukkan sebuah bilangan bulat : tyas
Data yang dimasukkan bukan bilangan bulat
Java Result: 1
BUILD SUCCESSFUL (total time: 7 seconds)
```

Membaca Input Data Numerik (lanjutan)

```
import java.io.*;
public class DemoInputNumerik2 {
    public  static void main (String [] args) throws IOException{
        System.out.print ("Masukkan sebuah bilangan riil : ");
        String temp;
        double bilangan = 0;

        InputStreamReader isr = new InputStreamReader (System.in);
        BufferedReader br = new BufferedReader (isr);

        temp = br.readLine();

        try{
            bilangan = Double.parseDouble(temp);

        } catch (NumberFormatException nfe){
            System.out.println("Data yang dimasukkan bukan bilangan");
            System.exit(1);
        }
        System.out.println ("Bilangan yang dimasukkan adalah " +bilangan);
    }
}
```

Membaca Input Data Numerik (lanjutan)

```
run:
Masukkan sebuah bilangan riil : 5.9
Bilangan yang dimasukkan adalah 5.9
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
run:
Masukkan sebuah bilangan riil : tyas
Data yang dimasukkan bukan bilangan
Java Result: 1
BUILD SUCCESSFUL (total time: 3 seconds)
```

Konsep Object Oriented Programming

- Ciri OOP
  - Abstraction
  - Encapsulation
  - Inheritance
  - Polymorphism

Abstraction (Abstraksi)

- Penyembunyian kerumitan dari suatu proses

Encapsulation (Pembungkusan)

- Cara atau mekanisme untuk melakukan abstraksi
  - Private (hanya dapat diakses oleh kelas yang meilikinya saja)
  - Protected (dapat diakses oleh kelas yang memilikinya dan juga oleh kelas-kelas yang masih meiliki hubungan turunan)
  - Public (kebalikan dari private, dapat diakses oleh semua bagian di dalam program)

Inheritance (Pewarisan)

- Suatu kelas dapat diturunkan menjadi kelas-kelas baru (sub class) lainnya yang akan mewarisi beberapa sifat atau perilaku dari kelas induknya (super class)
- Misal
  - Kelas A → induk
  - Kelas B → turunan
  - Sifat dan perilaku baru dapat ditambahkan ke kelas B, yang sebelumnya tidak dimiliki oleh kelas A

Polimorfisme

- Kemampuan suatu objek untuk mengungkapbanyak hal melalu satu cara yang sama
- Misal :
  - Kelas A → induk
  - Kelas B, C, dan D → turunan
  - Method-method yang terdapat pada kelas B, C, dan D dapat dijalankan hanya dari objek yang diinstansiasi.