

# **LECTURE NOTES**

**COMP6598**

**Introduction to Programming**

**Elementary Programming (Data Type,  
Input and Output)**

**Minggu 2**

**Sesi 3**

**Programming Dasar  
(Tipe Data, Input dan Output)**

## LEARNING OUTCOMES

1. LO2 : Peserta diharapkan dapat memahami dasar – dasar dalam pemrograman Java

### OUTLINE MATERI(Sub-Topic):

1. Menulis program sederhana
2. Membaca input dari console
3. Identifier
4. Variabel
5. Konstanta
6. Tipe Data Numerik
7. Konversi Tipe Numerik
8. Tipe Data Karakter, String dan Boolean

# ISI MATERI

## Menulis Program Sederhana

Terdapat beberapa aturan dasar dalam pembuatan program pada Java :

1. Setiap file **.java** pada pemrograman Java dapat memiliki beberapa **class**, namun **wajib** memiliki satu **public class** yang memiliki nama yang sama dengan nama file dari kelas tersebut.
2. Sebuah program java mungkin dapat terdiri lebih dari satu file .java namun dari beberapa file .java tersebut harus memiliki hanya satu **fungsi utama** (*main method*) karena ketika program di eksekusi *Java Compiler* akan secara otomatis mencari fungsi main pada file – file Java pada proyek tersebut dan kelas yang didalamnya terdapat fungsi utama akan menjadi **kelas utama** (*main class*) yang berarti program dimulai dari kelas dan fungsi tersebut.
3. Fungsi main wajib bersifat **static** yang artinya dapat dijalankan meskipun objek belum dibuat. Karena pada pemrograman Java, seharusnya fungsi baru dapat dijalankan apabila objek kelas tersebut telah dibuat, namun jika diberikan *keyword static* maka fungsi tersebut tetap dapat dijalankan tanpa membuat objek dari kelas tersebut.
4. Fungsi main juga wajib bersifat **public** agar dapat diakses dari luar kelas tersebut.

Contoh Program ComputeArea untuk menghitung luas dari lingkaran

```
public class ComputeArea {  
    public static void main(String[] args){  
        double radius;  
        double area;  
        radius = 20;  
        area = radius * radius * 3.14;  
        System.out.println("The area for the circle of radius " +radius + " is " + area);  
    }  
}
```

Output :

```
The area for the circle of radius 20.0 is 1256.0
```

Penjelasan :

1. Public adalah access modifier (akan dijelaskan di bab berikutnya)
2. ComputeArea adalah nama class. Nama ini harus sama dengan nama file yang program tersebut. Harap diingat JAVA adalah *Case Sensitive* sehingga huruf besar dan huruf kecil akan dibaca berbeda
3. Fungsi main wajib ada dan bersifat static
4. double adalah tipe data yang digunakan untuk variabel radius dan area
5. radius dan area adalah dua variabel yang digunakan untuk menampung data. dimana radius adalah untuk menampung data jari - jari, sedangkan area untuk menampung data luas yang merupakan hasil perhitungan.
6. System.out.println adalah untuk mencetak tulisan ke monitor dan setelah dicetak maka kursor akan berpindah ke baris berikutnya. Untuk mencetak atau menampilkan ke monitor bisa juga digunakan statement system.out.print.

### Membaca input dari console

Untuk membaca input dari console kita dapat menggunakan class **Scanner** untuk membuat object yang dapat membaca dari System.in

```
Scanner Input = new Scanner(System.in)
```

Input adalah nama variable dimana tipe-nya adalah **Scanner**. Kita dapat mengubah nama variabel tersebut dengan namai lain, contoh

```
Scanner baca = new Scanner (System.in)
```

Perhatikan contoh berikut

```
1 import java.util.Scanner;
2 public class ComputeArea {
3     public static void main(String[] args){
4         double radius, area;
5         // Create a Scanner object
6         Scanner input = new Scanner(System.in);
7
8         // Prompt the user to enter a radius
9         System.out.print("Enter a number for radius: ");
10        radius = input.nextDouble();
11
12        // Compute area
13        area = radius * radius * 3.14159;
14
15        // Display results
16        System.out.println("The area for the circle of radius " +radius + " is " + area);
17    }
18 }
```

Penjelasan :

- Baris 1, dikarenakan **Scanner** ada di package **java.util**, maka kita perlu memasukkannya dengan menggunakan statement import **java.util.Scanner**
- Baris 6, membuat object scanner dengan nama input. Input ini bisa saja diganti dengan nama lain.
- Baris 10, membaca input yang dimasukkan ke dalam variabel Radius. Dikarenakan Radius memiliki tipe data Double , maka digunakan method `.nextDouble()`
- Baris 13, dilakukan perhitungan area
- Baris 16, menampilkan tulisan dan hasil perhitungan area

Tabel berikut adalah method dari object Scanner

<i>Method</i>	<i>Description</i>
<code>nextByte()</code>	reads an integer of the <b>byte</b> type.
<code>nextShort()</code>	reads an integer of the <b>short</b> type.
<code>nextInt()</code>	reads an integer of the <b>int</b> type.
<code>nextLong()</code>	reads an integer of the <b>long</b> type.
<code>nextFloat()</code>	reads a number of the <b>float</b> type.
<code>nextDouble()</code>	reads a number of the <b>double</b> type.
<code>next()</code>	reads a string that ends before a whitespace character.
<code>nextLine()</code>	reads a line of text (i.e., a string ending with the <i>Enter</i> key pressed).

## Identifier

Identifier adalah nama yang mengidentifikasi element seperti class, method, constant dan variabel di dalam sebuah program. Contoh Identifier adalah area, main, radius, ComputeArea.

Semua identifier harus memenuhi aturan berikut :

- Sebuah identifier adalah serangkaian karakter yang terdiri dari huruf, angka, dan garis bawah (`_`) dan tanda dollar (`$`)
- Sebuah identifier harus dimulai dengan sebuah huruf, garis bawah (`_`) atau tanda dollar (`$`). Identifier tidak dapat diawali dengan angka.

- Sebuah identifier tidak dapat berupa reserved word (daftar bisa dilihat di halaman berikutnya)
- Sebuah identifier tidak dapat berupa true, false atau null

Contoh : \$2, ComputeArea, radius, area, number1. Harap diperhatikan bahwa Java bersifat *Case Sensitive* artinya Area, area, AREA adalah identifier yang berbeda.

Disarankan penamaan identifier untuk variabel, konstanta, method, class dan package sesuai dengan fungsinya, misalnya saja untuk variabel untuk luas lingkaran maka dapat dituliskan luasLingkaran.

Berikut adalah reserved word yang tidak dapat digunakan sebagai identifier.

abstract	default	while	package	this	class	continue
assert	do	if	private	throw	finally	float
boolean	double	implements	protected	synchronized	native	new
break	else	import	public	transient	super	switch
byte	enum	instanceof	return	TRUE	for	null
case	extends	int	short	try	char	final
catch	FALSE	interface	static	void	long	volatile

## Variabel

Variabel digunakan untuk merepresentasikan nilai yang dapat berubah di dalam sebuah program, dapat juga dikatakan bahwa variabel adalah sebuah penampung nilai. Dari contoh berikut terlihat bahwa radius dan area adalah variabel dengan tipe data **double**. Disebut variabel karena nilai dari radius dan area dapat diubah. Pada data pertama nilai radius adalah 1.0 sehingga menghasilkan area 3.14159, pada data yang kedua nilai radius diubah menjadi 2.0, sehingga area menjadi 12.56636.

```

1 public class ComputeArea {
2     public static void main(String[] args){
3         double radius, area;
4         // Compute the first area
5         radius = 1.0;
6         area = radius * radius * 3.14159;
7         System.out.println("The first area is " + area + " for radius " + radius);
8
9         // Compute the second area
10        radius = 2.0;
11        area = radius * radius * 3.14159;
12        System.out.println("The second area is " + area + " for radius " + radius);
13    }
14 }

```

Output :

```

The first area is 3.14159 for radius 1.0
The second area is 12.56636 for radius 2.0

```

Sebelum variabel dapat digunakan, maka harus dilakukan deklarasi variabel. Deklarasi variabel ini berarti memberitahu compiler nama variabel dan tipe data yang dapat disimpan. Saat deklarasi variabel juga berarti memberitahukan *compiler* untuk mengalokasikan besaran memory sesuai dengan ukuran data tersebut.

Syntax deklarasi variabel adalah :

***datatype variableName;***

berikut beberapa contoh deklarasi variabel

**int** count; // mendeklarasikan count sebagai variabel integer

**double** radius; //mendeklarasikan radius sebagai variable double

**char** flag; //mendeklarasikan flag sebagai variabel karakter

Jika beberapa variabel mempunyai tipe data yang sama dapat pula dituliskan sebagai berikut

***datatype variable1, variable2, ..., variablen;***

Contoh :

**int** count, temp; // mendeklarasikan count dan temp sebagai variabel integer

**double** radius, area; // mendeklarasikan radius dan area sebagai variabel double

Variabel juga bisa diberikan nilai awal. Deklarasi variabel dan pemberian nilai awal dapat dilakukan dalam satu langkah. Contoh sebagai berikut :

```
int count = 1;
```

Penulisan code di atas sama dengan code berikut :

```
int count;
```

```
count = 1;
```

### Konstanta (*Constant*)

**Constant** adalah sebuah identifier yang merepresentasikan nilai tetap / permanent. Jika pada variabel nilainya dapat diubah, namun pada constant nilai tidak dapat diubah.

Syntax penulisan constant sebagai berikut :

```
final datatype CONSTANTNAME = value;
```

```
1 public class ComputeArea {
2     public static void main(String[] args) {
3         // Declare a constant
4         final double PI = 3.14159;
5
6         // Assign a radius
7         double radius = 20;
8
9         // Compute area
10        double area = radius * radius *PI ;
11
12        // Display results
13        System.out.println("The area for the circle of radius " +radius + " is " + area);
14    }
15 }
```

**Constant** harus dideklarasikan terlebih dahulu sebelum dapat digunakan. Saat mendeklarasikan sebuah constant maka keyword **final** digunakan. Pada contoh diatas PI dideklarasikan sebagai constant dengan tipe datanya adalah integer. Nilai PI tidak dapat diubah tidak seperti nilai radius yang dapat diubah. Nilai PI hanya dapat diubah di source code.



## Tipe Data Numerik

Java memiliki 6 tipe data numerik yaitu 4 tipe untuk integer dan 2 tipe untuk floating point, Setiap tipe data mempunyai batasan dan ukuran masing masing.

<i>Name</i>	<i>Range</i>	<i>Storage Size</i>
<b>byte</b>	$-2^7$ to $2^7-1$ (-128 to 127)	8-bit signed
<b>short</b>	$-2^{15}$ to $2^{15}-1$ (-32768 to 32767)	16-bit signed
<b>int</b>	$-2^{31}$ to $2^{31}-1$ (-2147483648 to 2147483647)	32-bit signed
<b>long</b>	$-2^{63}$ to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
<b>float</b>	Negative range: $-3.4028235\text{E}+38$ to $-1.4\text{E}-45$ Positive range: $1.4\text{E}-45$ to $3.4028235\text{E}+38$	32-bit IEEE 754
<b>double</b>	Negative range: $-1.7976931348623157\text{E}+308$ to $-4.9\text{E}-324$ Positive range: $4.9\text{E}-324$ to $1.7976931348623157\text{E}+308$	64-bit IEEE 754

Tabel : Tipe Data Numerik (Sumber : Y. Daniel Liang, 2012)

Bilangan **integer** merupakan bilangan bulat sedangkan **floating point** adalah bilangan pecahan. Adapun operator yang digunakan untuk tipe data numerik tersebut adalah + , - , \* , / , dan %. Apabila anda mengetahui bahwa sebuah variabel akan bernilai diantara range byte, maka gunakan tipe data byte untuk variabel tersebut.

Apabila sebuah variabel diberikan nilai yang melebihi range tipe data-nya , maka akan mengakibatkan terjadinya overflow. Apabila sebuah nilai sedemikian kecil sehingga mendekati nol, maka akan terjadi underflow.

Berikut adalah contoh tipe data numerik dengan operator “/” (pembagian)

```
import java.util.Scanner;
public class ComputeSpeed {

    public static void main(String[] args){
        int speed, distance, time;

        Scanner input = new Scanner(System.in);
        System.out.print("Distance (in km) = ");
        distance = input.nextInt();

        System.out.print("\nTime needed (in hour) = ");
        time = input.nextInt();

        speed = distance/time;
        System.out.println("\nThe speed needed to travel " + distance + "km in " +
            time + " is= " + speed + "km per hour");
    }
}
```

## Output

```
Distance (in km) = 100

Time needed (in hour) = 2

The speed needed to travel 100km in 2 is= 50km per hour
```

## Konversi Tipe Data Numerik

**Casting** adalah operasi yang mengkonversikan sebuah nilai dari satu tipe data ke tipe data lainnya.

Apabila terdapat dua buah variabel, dimana variabel satu memiliki tipe data lebih kecil daripada variabel kedua, maka apabila kita akan memindahkan nilai variabel satu ke variabel kedua, hal ini dimungkinkan. Dari contoh di bawah ini bisa dilihat bahwa variabel a mempunyai tipe data integer, dimana range-nya lebih kecil daripada variabel b dengan tipe data float. Dikarenakan hal tersebut maka operasi  $b = a * a$  dimungkinkan, karena variabel b mempunyai range tipe data yang lebih besar daripada variabel a.

Contoh :

```
public static void main(String[] args){  
    int a=10;  
    float b;  
    b = a*a;  
    System.out.println("b = " + b );  
}
```

Namun apabila dilakukan sebaliknya maka akan muncul error “Type Mismatch : cannot convert from float to integer”. Untuk mengatasi hal ini maka kita memerlukan type casting.

Casting sebuah tipe data dengan range lebih kecil ke lebih besar disebut dengan *widening a type*, sebaliknya disebut dengan *narrowing a type*. Untuk widening , Java dapat secara otomatis melakukannya, namun untuk narrowing perlu dituliskan secara eksplisit. Contoh casting bisa dilihat pada statement `b = (int) (a*a);` artinya kita melakukan casting dari float menjadi integer. Casting tidak akan mengubah tipe data variabel tersebut , jadi variabel a akan tetap bernilai float.

```
public static void main(String[] args){  
    float a=10;  
    int b;  
    b = (int) (a*a);  
    System.out.println("b = " + b );  
}
```

## Tipe Data Boolean, Char dan String

Tipe data boolean hanya memiliki dua nilai yaitu true atau false.

```
public class booleanTest {  
    public static void main (String[] args){  
        boolean check=false;  
  
        System.out.println("nilai variabel check awal adalah " + check);  
  
        check = 10 > 5;  
        System.out.println("nilai variabel check adalah " + check);  
  
        check = 5 > 10;  
        System.out.println("nilai variabel check adalah " + check);  
    }  
}
```

Output :

```
nilai variabel check awal adalah false  
nilai variabel check adalah true  
nilai variabel check adalah false
```

Pada awalnya variabel check diberikan nilai false. Setelah dilakukan operasi yang pertama maka didapatkan nilainya adalah true dimana 10 lebih besar dari 5. Namun saat dibalikan maka nilainya adalah false yang artinya 5 tidak benar lebih besar dari 10

### Tipe data Character

Tipe data Character merepresentasikan sebuah karakter. Perhatikan contoh berikut ini

```
char huruf = 'A ';
```

```
char numChar = '1 '
```

Statement yang pertama adalah memberikan karakter A ke variabel huruf, sedangkan statement yang kedua adalah memberikan karakter angka 1 ke variabel numChar. Perhatikan bahwa tanda petik yang diberikan adalah petik satu ( ' '). Apabila digunakan petik dua ( " "), maka variabel tersebut seharusnya bertipe data String dan bukan Char.

Sebuah karakter disimpan pada komputer sebagai rangkaian dari 0 dan 1. Pemetaan (*mapping*) sebuah karakter dengan binarynya disebut dengan *encoding*. Perhatikan contoh berikut

```
char huruf = 'A';
```

```
char huruf = '\u0041'; // Character A's Unicode is 0041
```

Kedua statement tersebut sama sama memberikan karakter A ke variabel huruf.

## Escape Characters

Apabila anda ingin mencetak sebuah pesan ke tetapi menggunakan karakter tertentu misal tanda petik, maka anda membutuhkan escape sequence.

Escape Sequence	Description
\t	Insert a tab in the text at this point.
\b	Insert a backspace in the text at this point.
\n	Insert a newline in the text at this point.
\r	Insert a carriage return in the text at this point.
\f	Insert a form feed in the text at this point.
\'	Insert a single quote character in the text at this point.
\"	Insert a double quote character in the text at this point.
\\	Insert a backslash character in the text at this point.

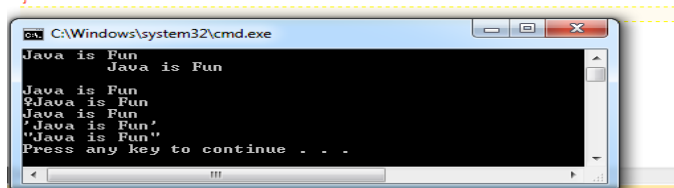
Tabel Escape sequence

Pada contoh berikut ini dituliskan bahwa apabila anda ingin tanda petik dua muncul maka bisa digunakan \" , sehingga secara lengkap bisa ditulis System.out.println(\" \"Java is Fun

```

public class EscapeSequence {
    public static void main(String[] args)
    {
        System.out.println("Java is Fun");
        System.out.println("\tJava is Fun");
        System.out.println("\nJava is Fun");
        System.out.println("\fJava is Fun");
        System.out.println("\rJava is Fun");
        System.out.println("\'Java is Fun\'");
        System.out.println("\"Java is Fun\"");
    }
}

```



\"");

## Tipe Data String

String adalah serangkaian karakter. Pada tipe data Char hanya berisi satu karakter saja, tetapi untuk serangkaian karakter maka digunakan tipe data String.

Contoh :

```
String kalimat = "Algoritma dan Pemrograman";
```

Pada tipe data String menggunakan tanda petik dua sedangkan pada char menggunakan petik satu. Untuk membaca input dari tipe data String maka dapat digunakan `.next( )` ataupun `.nextLine( )`. Perbedaanannya adalah pada `.next( )` hanya akan membaca string sampai dengan karakter *whitespace*. Karakter *whitespace* diantaranya yaitu spasi, `\t`, `\f`, `\r` atau `\n`. Namun apabila ingin membaca satu kesatuan kalimat maka dapat digunakan `.nextLine( )`

Contoh dengan menggunakan `.next()`

```
import java.util.Scanner;

public class StringTest {
    public static void main(String[] args){
        //declare string
        String s1, s2, s3;

        Scanner input = new Scanner(System.in);

        System.out.println("Enter three words separated by spaces (using .next(): ");
        s1 = input.next();
        s2 = input.next();
        s3 = input.next();
        System.out.println("s1 is " + s1);
        System.out.println("s2 is " + s2);
        System.out.println("s3 is " + s3);
    }
}
```

Output

```
Enter three words separated by spaces (using .next():
algorithm and programming
s1 is algorithm
s2 is and
s3 is programming
```

Contoh dengan menggunakan `.nextLine()`

```
import java.util.Scanner;

public class StringTest {
    public static void main(String[] args){
        //declare string
        String line;

        Scanner input = new Scanner(System.in);

        System.out.println("enter a sentence : ");
        line = input.nextLine();
        System.out.println("the sentence is : " + line);
    }
}
```

## Output

```
enter a sentence :
i like this algorithm and programming subject
the sentence is : i like this algorithm and programming
```

## Fungsi dalam *String Class*

Dikarenakan tipe data String merupakan tipe data *composite* (tipe data dalam bentuk Class yang berisi gabungan dari beberapa data single dalam hal ini char) maka class String memiliki beberapa fungsi yang dipaparkan pada table berikut.

Methods	Description
length()	Returns the length of this string.
charAt(index)	Returns the char value at the specified index.
concat()	Concatenates the specified string to the end of this string.
substring(start,finish)	Returns a new string that is a substring of this string.
toLowerCase()	Converts all of the characters in this String to lower case using the rules of the default locale.
toUpperCase()	Converts all of the characters in this String to upper case using the rules of the default locale.
trim()	Returns a copy of the string, with leading and trailing whitespace omitted.
replace(char oldChar, char newChar)	Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.
replaceFirst(String regex, String replacement)	Replaces the first substring of this string that matches the given regular expression with the given replacement.
split(String regex, int limit)	Splits this string around matches of the given regular expression.

## Contoh :

```
public class StringTest {
    public static void main(String[] args){
        //declare string
        String kata1, kata2;
        kata1 = "Algoritma";
        kata2 = "Pemrograman";

        System.out.println("Panjang kata1 = " + kata1.length());
        System.out.println("Menggabungkan kata1 dan kata2= " + kata1.concat(kata2));
        System.out.println("Menjadikan huruf besar semua " + kata1.toUpperCase());
        System.out.println("Menjadikan huruf kecil semua " + kata2.toLowerCase());
    }
}
```

## Output

```
Panjang kata1 = 9
Menggabungkan kata1 dan kata2= AlgoritmaPemrograman
Menjadikan huruf besar semua ALGORITMA
Menjadikan huruf kecil semua pemrograman
```

## SIMPULAN

Semua nama disebut dengan Identifier yang memiliki ketentuan yang harus dipenuhi diantaranya adalah tidak diperbolehkan diawali dengan angka. Java mengenal Case Sensitive yang artinya huruf besar dan huruf kecil akan dianggap berbeda. Perbedaan dari variabel dan konstanta adalah nilai variabel dapat diubah sedangkan konstanta tidak dapat diubah.

Setiap variabel maupun konstanta memiliki tipe data sesuai dengan kebutuhannya. Tipe data numerik terdiri atas 4 tipe untuk integer dan 2 tipe untuk floating point. Tipe integer yaitu byte, short, int, long. Floating point yaitu float dan double. Setiap tipe data mempunyai rentang nilai masing - masing. Tipe data lainnya adalah char yang memuat satu karakter, sedangkan jika membutuhkan lebih dari satu karakter dapat menggunakan tipe data String. Tipe data terakhir adalah boolean yang hanya memuat nilai true atau false.

Tipe data String merupakan *composite class* yang dimana nilainya terdiri dari penggabungan nilai sekumpulan variabel char (*array of char*) disamping itu kelas String juga memiliki **fungsi – fungsi** pada kelasnya.



## DAFTAR PUSTAKA

1. Y. Daniel Liang. (2013). Introduction to Java Programming . Edition 09. Pearson Education. New Jersey.
2. <https://netbeans.org/kb/docs/javaee/ecommerce/intro.html>
3. [https://en.wikipedia.org/wiki/Java\\_\(software\\_platform\)](https://en.wikipedia.org/wiki/Java_(software_platform))
4. [https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/lecture-notes/MIT6\\_092IAP10\\_lec01.pdf](https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/lecture-notes/MIT6_092IAP10_lec01.pdf)
5. <http://docs.oracle.com/javase/7/docs/api/java/lang/String.html>
6. [http://www.tutorialspoint.com/java/java\\_tutorial.pdf](http://www.tutorialspoint.com/java/java_tutorial.pdf)
7. <http://docs.oracle.com/javase/tutorial/java/javaOO/accesscontrol.html>