

5. Suplemen (1)

By : Mahendar Dwi Payana, S.ST., M.T

Referensi :

1. Buku "Pemrograman Berorientasi Objek. Teori dan Implementasi Java"
2. Pengarang : Raden Budiarto
3. Tahun : 2018
4. Jenis : E-Book - [ebook play.books.com](http://ebook.play.books.com)

5.1 Konversi Tipe Data

JAVA memungkinkan tipe data untuk dikonversi dari deklarasi awal. Konversi ini terkadang diperlukan untuk beberapa kondisi pada pemrograman **java**. Ada berbagai bentuk konversi data yang dapat digunakan, mulai dari tipe data **numerik** ke **numerik**, **numerik** ke **string** hingga **string** menjadi **boolean**. Secara umum ada dua metode konversi tipe data yaitu menggunakan *casting* dan menggunakan *helper class*.

- konversi tipe data numerik
 - Urutan jangkauan tipe data numerik dari yang paling besar adalah **double**, **float**, **long**, **int**, **short**, **byte**.
- jika kita mengkonversi dari kecil ke besar misal dari **int** ke **double** maka kita dapat mengkonversinya secara langsung.
- Jika kita mengkonversi dari besar ke kecil maka kita harus menggunakan tipe *cast* secara eksplisit.

Contoh konversi data dari kecil ke besar pada **JAVA** adalah sebagai berikut.

```
// konversi dari kecil ke besar
int a = 20;

double d = a; // nilai d = 20.0
```

Contoh konversi data dari besar ke kecil pada **JAVA** adalah sebagai berikut.

```
// konversi dari besar ke kecil
double d = 20.0;
int a = (int) d; // nilai a = 20
```

Contoh konversi tipe data helper class pada **JAVA** adalah sebagai berikut.

```
// konversi tipe data helper class
int a = 20;

boolean b = Boolean.valueOf(a); // nilai b = true

String s = String.valueOf(a); // nilai s = "20"
```

5.2 Sistem Input dan Output

Java menyediakan beberapa metode untuk menampilkan output dan menerima input dari pengguna. Metode yang disediakan oleh **Java** untuk menampilkan output adalah `System.out.print()` dan `System.out.println()`. Sedangkan untuk menerima input dari pengguna, **Java** menyediakan metode `Scanner`.

Selain itu terdapat juga metode menampilkan error yang disebut dengan `System.err.print()` dan `System.err.println()`. Metode ini digunakan untuk menampilkan pesan error pada program.

Java juga memiliki metode seperti karakter escape yang dapat digunakan untuk menampilkan karakter khusus seperti `\n` untuk baris baru, `\t` untuk tab, `\\` untuk backslash, `\"` untuk double quote, dan `\'` untuk single quote.

5.2.1 Sistem Output

Contoh penggunaan metode `System.out.print()` dan `System.out.println()` adalah sebagai berikut.

```
// metode print
System.out.print("Hello World"); // Hello World. perintah ini
tidak membuat baris baru
System.out.println("Hello World"); // Hello World. perintah ini
membuat baris baru
```

5.2.2 Sistem ERROR

Contoh penggunaan metode `System.err.print()` dan `System.err.println()` adalah sebagai berikut.

```
// metode print
System.err.print("Hello World"); // Hello World. perintah ini
tidak membuat baris baru
System.err.println("Hello World"); // Hello World. perintah ini
membuat baris baru
```

5.2.3 Sistem Input

Contoh penggunaan metode `Scanner` adalah sebagai berikut.

```
// metode Scanner
Scanner scanner = new Scanner(System.in); // membuat objek scanner
System.out.print("Masukkan nama anda : "); // menampilkan pesan
String nama = scanner.nextLine(); // menerima input dari pengguna
System.out.println("Halo " + nama); // menampilkan pesan
```

Untuk menggunakan input `Scanner` `java.util.Scanner` harus diimport terlebih dahulu. Untuk mengimport `Scanner` kita dapat menggunakan perintah `import java.util.Scanner;`

Menggunakan `Scanner` kita harus menyesuaikan tipe datanya. Jika kita ingin menerima input `int` maka kita harus menggunakan `scanner.nextInt()`. Jika kita ingin menerima input `double` maka kita harus menggunakan `scanner.nextDouble()`. Jika kita ingin menerima input `String` maka kita harus menggunakan `scanner.nextLine()`. Penjelasan masing-masing adalah sebagai berikut :

1. `scanner.nextInt()` digunakan untuk menerima input `int` dari pengguna.
2. `scanner.nextDouble()` digunakan untuk menerima input `double` dari pengguna.
3. `scanner.nextLine()` digunakan untuk menerima input `String` dari pengguna.
4. `scanner.next()` digunakan untuk menerima input `String` dari pengguna. Namun, `scanner.next()` hanya menerima input sampai spasi saja. Jika kita ingin menerima input `String` yang mengandung spasi maka kita harus menggunakan `scanner.nextLine()`
5. `scanner.nextBoolean()` digunakan untuk menerima input `boolean` dari pengguna.
6. `scanner.nextByte()` digunakan untuk menerima input `byte` dari pengguna.
7. `scanner.nextShort()` digunakan untuk menerima input `short` dari pengguna.
8. `scanner.nextLong()` digunakan untuk menerima input `long` dari pengguna.
9. `scanner.nextFloat()` digunakan untuk menerima input `float` dari pengguna.

Sebenarnya selain dari input `Scanner` kita juga dapat menggunakan `BufferedReader` untuk menerima input dari pengguna. Namun, `BufferedReader` membutuhkan lebih banyak kode untuk menerima input dari pengguna. Contoh penggunaan `BufferedReader` adalah sebagai berikut.

```
// metode BufferedReader
BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in)); // membuat objek BufferedReader
System.out.print("Masukkan nama anda : "); // menampilkan pesan
String nama = reader.readLine(); // menerima input dari pengguna
System.out.println("Halo " + nama); // menampilkan pesan
```

5.2.4 Karakter Escape

Contoh penggunaan karakter escape adalah sebagai berikut.

```
// karakter escape
System.out.println("Hello \n World"); // Hello
                                         // World
System.out.println("Hello \t World"); // Hello      World
System.out.println("Hello \\ World"); // Hello \ World
System.out.println("Hello \" World"); // Hello " World
System.out.println("Hello \' World"); // Hello ' World
```

```
String heading1 = "Nama";
String heading2 = "Nilai";
double nilai = 100.0;

System.out.printf("%-15s %15s\n", heading1, heading2);

System.out.println("")
```

Daftar Karakter Escape adalah sebagai berikut.

Karakter	Deskripsi
<code>\n</code>	Baris baru
<code>\t</code>	Tab
<code>\\</code>	Backslash
<code>\"</code>	Double quote
<code>\'</code>	Single quote
<code>\b</code>	Backspace
<code>\r</code>	Carriage return
<code>\f</code>	Form feed