

## Java Language

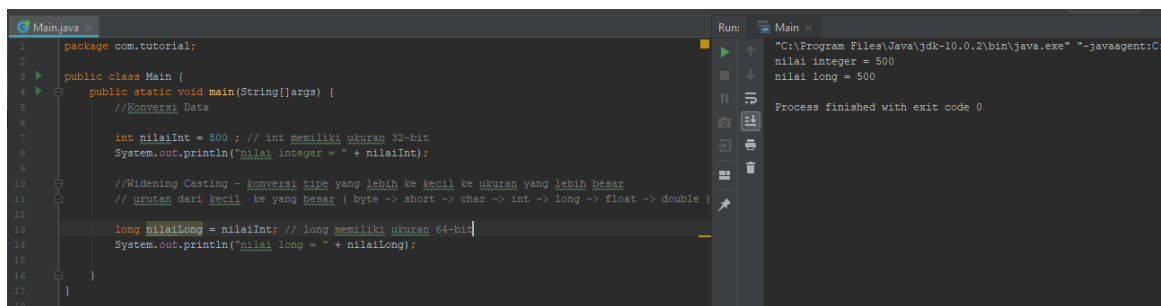
### Konversi (casting) Tipe Data Primitive

Konversi tipe data adalah teknik mengubah nilai yang awalnya dari tipe data a, menjadi tipe data b.

Ada dua jenis tipe casting :

- Widening Casting (automatically) - mengonversi tipe yang lebih kecil ke ukuran tipe yang lebih besar **byte -> short -> char -> int -> long -> float -> double**
- Narrowing Casting (manually) - mengonversi tipe yang lebih besar ke tipe ukuran yang lebih kecil **double -> float -> long -> int -> char -> short -> byte**

### Widening Casting

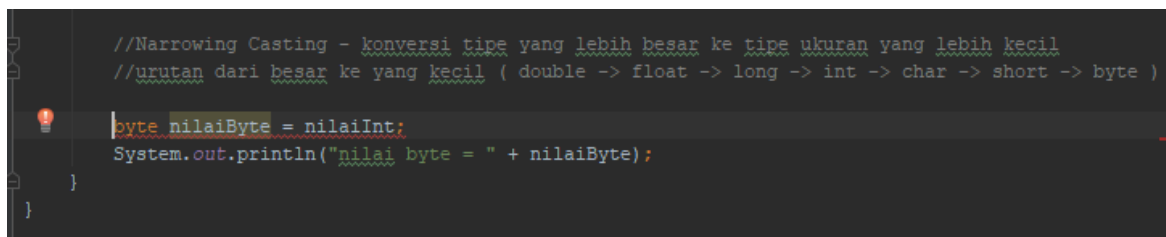


```
1 package com.tutorial;
2
3 public class Main {
4     public static void main(String[] args) {
5         //Konversi Data
6
7         int nilaiInt = 500; // int memiliki ukuran 32-bit
8         System.out.println("nilai integer = " + nilaiInt);
9
10        //Widening Casting - konversi tipe yang lebih kecil ke ukuran yang lebih besar
11        // urutan dari kecil ke yang besar ( byte -> short -> char -> int -> long -> float -> double )
12
13        long nilaiLong = nilaiInt; // long memiliki ukuran 64-bit
14        System.out.println("nilai long = " + nilaiLong);
15    }
16 }
17
18
```

Run: Main  
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\...  
nilai integer = 500  
nilai long = 500  
Process finished with exit code 0

Hasil dari output menghasilkan nilai yang sama karena jika mengkonversi tipe data dari yang kecil ke yang lebih besar tidak akan mengubah nilainya.

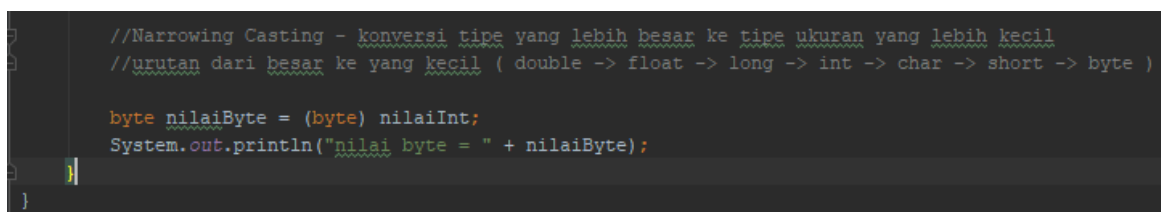
### Narrowing Casting



```
//Narrowing Casting - konversi tipe yang lebih besar ke tipe ukuran yang lebih kecil
//urutan dari besar ke yang kecil ( double -> float -> long -> int -> char -> short -> byte )

byte nilaiByte = nilaiInt;
System.out.println("nilai byte = " + nilaiByte);
}
```

Program akan terjadi eror jika konversi tipe yang lebih besar ke tipe ukuran yang lebih kecil. Maka dari itu kita harus melakukan secara manual dengan menempatkan jenis tipe data dalam tanda kurung di depan nilai. Contoh :



```
//Narrowing Casting - konversi tipe yang lebih besar ke tipe ukuran yang lebih kecil
//urutan dari besar ke yang kecil ( double -> float -> long -> int -> char -> short -> byte )

byte nilaiByte = (byte) nilaiInt;
System.out.println("nilai byte = " + nilaiByte);
}
```

```
1 package com.tutorial;
2
3 public class Main {
4     public static void main(String[] args) {
5         //Konversi Data
6
7         int nilaiInt = 500; // int memiliki ukuran 32-bit
8         System.out.println("nilai integer = " + nilaiInt);
9
10        System.out.println("=====Widening Casting=====");
11        //Widening Casting - konversi tipe yang lebih kecil ke ukuran yang lebih besar
12        //urutan dari kecil ke yang besar ( byte -> short -> char -> int -> long -> float -> double )
13
14        long nilaiLong = nilaiInt; // long memiliki ukuran 64-bit
15        System.out.println("nilai long = " + nilaiLong);
16
17        System.out.println("=====Narrowing Casting=====");
18        //Narrowing Casting - konversi tipe yang lebih besar ke tipe ukuran yang lebih kecil
19        //urutan dari besar ke yang kecil ( double -> float -> long -> int -> char -> short -> byte )
20
21        byte nilaiByte = (byte) nilaiInt;
22        System.out.println("nilai byte = " + nilaiByte);
23    }
24 }
```

```
"C:\Program Files\Java\jdk-10.0.2\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-10.0.2\bin\javaagent.jar" -classpath .;C:\Program Files\Java\jdk-10.0.2\bin\java.exe nilai integer = 500
=====Widening Casting=====
nilai long = 500
=====Narrowing Casting=====
nilai byte = -12
Process finished with exit code 0
```

Hasil dari output menghasilkan nilai yang berbeda karena jika mengkonversi tipe data dari yang besar ke yang lebih kecil akan mengubah nilainya.

## Konversi dalam pembagian

```
System.out.println("=====Konversi Pembagian=====");
//Konversi dalam Pembagian
int a = 10;
int b = 4;
float c = a / b;
System.out.printf("%d / %d = %f ", a, b, c);
```

```
=====Konversi Pembagian=====
10 / 4 = 2.000000
Process finished with exit code 0
```

Hasil dari pembagian di atas adalah 2.000000 seharusnya 2.5 . mengapa ? karena tipe data int tidak dapat menghasilkan desimal. Dan nilai 00000 di belakang angka 2 merupakan output dari float.

Untuk mengkonversi agar nilai nya sesuai yaitu sama dengan cara narrowing casting. Contoh :

```
System.out.println("=====Konversi Pembagian=====");
//Konversi dalam Pembagian
int a = 10;
int b = 4;
float c = (float) a / b;
System.out.printf("%d / %d = %f ", a, b, c);
```

```
=====Konversi Pembagian=====
10 / 4 = 2.500000
Process finished with exit code 0
```