


# Casting

Sering kali kita ingin mengubah suatu tipe data tertentu ke bentuk tipe data lain dalam suatu variabel. Perubahan ini disebut dengan istilah casting. Untuk menggunakan casting di dalam Java, cukup deklarasikan tipe data yang baru di dalam statement assignment. Seperti contoh berikut ini :

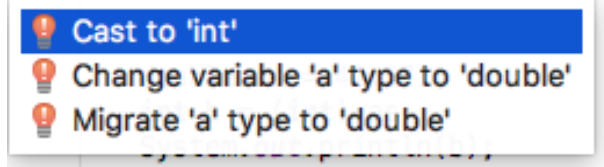


```
int a = 3.14;
```

Pada contoh di atas, kita mencoba mendeklarasikan variabel a sebagai integer. Akan tetapi nilai yang kita berikan merupakan tipe *double*. Dengan kata lain antara tipe data dan nilai tidak sesuai. Kita bisa memaksa keadaan ini dengan *casting*. Cara termudah untuk melakukan *casting* adalah arahkan *cursor* pada bagian ikon lampu berwarna merah sebelah kiri kemudian pilih *cast to 'int'*



```
int a = 3.14;
```



Dengan melakukan hal tersebut maka saat kita melakukan perubahan tipe data ke int. di dalam kode, akan muncul seperti berikut :

```
int a = (int) 3.14;
```


Apabila kita coba mencetak variabel a , maka *output*-nya akan menjadi seperti berikut:

Output

```
3
```

Mengapa dapat terjadi demikian? Jawabannya merujuk kepada tipe data yang di-*casting*. Pada nilai yang sebelumnya, tipe datanya *double*. Maka saat melakukan *casting* untuk mengubah tipe data ke bentuk *int*, nilai tersebut akan berubah menjadi *int*.

Kita dapat melakukan *casting* terhadap beberapa tipe data. Bagaimana jika kita akan melakukan *casting* dari *double* ke dalam bentuk *string*? Dengan cara mendeklarasikan variabel sebagai *String* dan menggunakan fungsi `String.valueOf`. Berikut ini contohnya:



```
1. double a = 3.14;
2. String hasil = String.valueOf(a);
3. System.out.println(hasil);
```

Output:

Perlu diingat bahwa hasil dari *output* tersebut bukanlah lagi *double* melainkan sudah menjadi *String*.

Selain untuk mengubah atau mengkonversikan tipe data menjadi tipe data lain dalam suatu variabel, casting juga dapat digunakan untuk mengkonversi kelas menjadi kelas lain. Untuk menkonversi kelas terdapat dua metode yaitu Upcasting dan Downcasting. Upcasting dan downcasting terkait erat dengan konsep pewarisan. Sangat umum untuk menggunakan variabel referensi untuk merujuk pada tipe yang lebih spesifik. Dan setiap kali kita melakukan ini, upcasting secara implisit terjadi. Upcasting adalah casting dari subclass ke superclass sementara downcasting adalah casting dari superclass ke subclass.

Contoh penggunaan upcasting dan downcasting adalah sebagai berikut:

```
1. public class Hewan {
2.     public void makan() {
3.         // ...
4.     }
5. }
```

Kita buat terlebih dahulu kelas **Hewan**

```
1. public class Kucing extends Hewan {
2.     public void makan() {
3.         // ...
4.     }
5.
6.     public void meow() {
7.         // ...
8.     }
9. }
```

Selanjutnya buat kelas Kucing yang akan meng-extends kelas **Hewan**

```
1. public class Main {
2.     public static void main(String[] args) {
3.         Kucing anggora = new Kucing();
4.
5.         Hewan hewan = anggora; //secara implisit upcasting
6.         hewan = (Hewan) anggora; //secara eksplisit melakukan casting dari subclass ke superclass
7.
8.         Hewan hewan2 = new Kucing();
9.         ((Kucing) hewan2).meow(); //downcast hewan ke kucing, casting dari superclass ke subclass
10.    }
11. }
```

```
1. Kucing anggora = new Kucing();
```

Setelah itu kita buat kelas Main, lalu kita instansiasi objek kucing anggora yang mereferensi variabel tipe kucing.

```
1. Hewan hewan = anggora; //secara implisit upcasting
```

Kita juga bisa menetapkan referensi anggora menjadi variabel tipe hewan. Karena pada kasus ini kucing adalah subclass sementara hewan adalah superclass. Dan pada saat kita menetapkan referensinya (assign).

```
1. hewan = (Hewan) anggora; //secara eksplisit melakukan casting dari subclass ke superclass
```

Pada kode program sebelumnya telah dilakukan upcasting secara implisit. Sementara untuk kode program di atas kita lakukan upcasting secara eksplisit. Kita tidak perlu melakukan upcasting secara eksplisit, karena pada Java, kompiler akan mengetahui bila kelas tersebut ada hubungan inheritance. Apabila kita hendak menggunakan metode yang terdapat pada kelas kucing untuk objek hewan, hal tersebut tidak dapat dilakukan misalnya `hewan.meow()`. Untuk bisa melakukannya, kita bisa melakukan downcasting.

```
1. Hewan hewan2 = new Kucing();
2. ((Kucing) hewan2).meow(); //downcast hewan2 ke kucing, casting dari superclass ke subclass
```

Sebelumnya kita telah mereferensikan variabel hewan ke instans kucing. Dan ketika ingin menggunakan metode yang ada pada kelas kucing kita bisa menggunakan downcast seperti pada kode program di atas untuk menggunakan metode meow yang terdapat pada kelas kucing. Downcast dilakukan hanya pada instans kucing.

Selesai sudah materi Java Fundamental. Menguasai bahasa Java adalah salah satu batu pertama yang harus kamu lompat untuk terjun ke dunia "persilatan" pengembangan aplikasi. Pondasi Java yang kuat akan menciptakan aplikasi yang excellent. Kami tunggu aplikasi dahsyat dari Anda.

Selanjutnya? Masih banyak kelas-kelas lainnya yang menarik untuk Anda ikuti.

Bisa dimulai dari kelas Reguler :

- [Belajar Membuat Aplikasi Android Pemula](#)

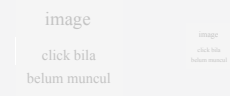
Atau bisa juga melanjutkan ke kelas Picodiploma :

- [Menjadi Android Developer Expert](#)
- [Kotlin Android Developer Expert](#)
- [Menjadi Game Developer Expert](#)
- [Membangun Progressive Web Apps](#)



Dicoding Space  
Jl. Batik Kumeli No.50, Sukaluyu,  
Kec. Cibeunying Kaler, Kota Bandung  
Jawa Barat 40123

Penghargaan



Decode Ideas  
Discover Potential

[Tentang Kami](#)

- [Blog](#)
- [Reward](#)
- [Showcase](#)
- [Hubungi Kami](#)
- [FAQ](#)