



Dr. Riza Satria Perdana, S.T., M.T.

Teknik Informatika - STEI ITB

Reflection

Kelas Class

Pemrograman Berorientasi Objek



Kelas Class

- Java memiliki kelas yang bernama Class (dengan C besar) untuk mengakses isi sebuah kelas, nama lengkapnya adalah `java.lang.Class`
 - perlu diingat bahwa semua package `java.lang` tidak perlu diimport
- Kelas Class perlu diinstansiasi dengan cara khusus (tidak bisa dengan **new**)
 - Lihat bagian berikut

Instansiasi objek Class

- Sebuah objek Class pasti berhubungan dengan sebuah nama kelas yang sudah dikenal Java
 - Instansiasi dapat dilakukan dengan 3 cara

- Dengan nama kelas:

```
Class c = Class.forName("java.lang.String");
```

- Dengan literal nama kelas:

```
Class c = String.class;
```

- Dengan objek yang sudah dimiliki sebelumnya:

```
String s = "hello";
```

```
Class c = s.getClass();
```

Method pada kelas Class

- Kelas Class memiliki banyak method, beberapa yang penting adalah:
 - `newInstance()` yang membuat objek baru
 - `getClasses()` yang mengembalikan interface dan kelas parent
 - `getFields()` dan `getMethods()` yang mengembalikan daftar field (attribut) dan method yang dimiliki
 - jika nama method atau field sudah diketahui, maka ada juga method `getField()` dan `getMethod()` (tunggal, bukan jamak)

Instansiasi Objek dengan objek Class

- Jika kita sudah memiliki objek Class (dengan salah satu cara yang sudah diberikan):

```
Class c = String.class;
```

```
String s = (String)c.newInstance();
```

```
Class cv = class.forName("java.lang.Vector");
```

```
Vector v = (Vector)cv.newInstance();
```

- Akan berguna dalam pembuatan plugin (ekstensi program yang ditambahkan tanpa mengkompilasi program [akan dijelaskan di bagian lain])



Kelas Field dan kelas Method

- Java juga memiliki kelas yang bernama Field dan Method (keduanya diawali huruf besar)
- bisa diakses dengan `java.lang.reflect.*`
- Kelas Field memungkinkan akses terhadap Field dalam sebuah kelas
 - mendapatkan nama, tipe, membaca dan mengeset nilai field
- Kelas Method memungkinkan akses terhadap sebuah method dalam sebuah kelas
 - mendapatkan nama, return value, dan parameter (singkatnya: signature) sebuah method
 - memanggil method

Contoh

- Mendapatkan daftar method kelas String:

```
Class c = String.class;  
Method m[] = c.getMethods();  
for (Method e: m) {  
    System.out.println(e.getName());  
}
```

Kegunaan Reflection

- Digunakan IDE untuk design GUI
 - IDE tidak perlu mengetahui sejak awal method apa saja yang ada di sebuah komponen sehingga komponen bisa ditambah kapan saja
- Digunakan Application Server (misalnya J2EE)
 - Suatu aplikasi yang menjalankan aplikasi lain dan memberikan pengaturan tambahan (dalam kasus J2EE: konkurensi, fitur terdistribusi)
- Digunakan untuk plugin
 - Editor seperti JEdit memakai plugin



Terima Kasih