



Responsi 7 (part 2)

Praktikum Pemrograman Berbasis Objek
Asisten IF2210 2022/2023

Outline

1. Reflection
2. Class
3. Field
4. Method

Reflection



Reflection

- Reflection mampu memungkinkan developer mengubah kode saat runtime.
- Contoh yang bisa dilakukan reflection
 - menambahkan method ke sebuah kelas yang ada
 - mengakses member yang private

Reflection

- Kenapa reflection?
 - Membuat testing
 - Menginjeksi dependency
 - Mengetes kode mahasiswa saat praktikum :)
 - Buat plugin
 - Membuat program magic
 - Mengurangi *nguli*
 - Contoh: di sebuah program Game. Ada 1 package isinya 500 kelas yang extend Skill, dan setiap kelas akan dimasukkan ke Map<String,Skill>, untuk memasukkan ke dalam map tersebut perlu memanggil static field String bernama ID dan instansiasi objek baru. Hal ini bisa dipermudah dengan reflection, sehingga hanya perlu for loop package tersebut dan menggunakan reflection.

Reflection

- Bayangkan:
 - Tidak perlu membuat Factory untuk tiap kelas yang butuh Factory
 - Meload member kelas dari csv/xml/json/database/lainnya tanpa membuat loader
 - Tidak bisa membuat method / atribut tambahan saat praktikum :)

Reflection

<https://www.oracle.com/technical-resources/articles/java/javareflection.html>

Class

- Java memiliki kelas Class (java.lang.Class) yang menggambarkan kelas
- Ada beberapa cara mendapatkan kelas:
 - `Class c = Person.class;`
 - `Class c = Class.forName("Person");`
 - `Person p = new Person("Fauzan Rafi Sidiq");`
`Class c = p.getClass();`
- Setelah mendapatkan kelas, ada banyak yang bisa dilakukan, misal mendapatkan daftar interface, superclass, memanggil constructor, dll
- Daftar lengkapnya ada di <https://docs.oracle.com/javase/8/docs/api/java/lang/Class.html>

Field

- Java memiliki kelas Field (`java.lang.reflect.Field`) yang menggambarkan field pada sebuah kelas
- Ada beberapa cara mendapatkan method:
 - `Field[] f = c.getFields();`
 - `Field[] f = c.getDeclaredFields();`
 - `Field f = c.getDeclaredField("name");`
- Setelah mendapatkan field, ada banyak yang bisa dilakukan, misal membaca / mengubah nilai, mendapatkan nama field, mendapatkan tipe field, dll
- Daftar lengkapnya ada di <https://docs.oracle.com/javase/8/docs/api/java/lang/reflect/Field.html>

Method

- Contoh membaca / mengubah field:
 - `Field f = c.getDeclaredField("name");`
`f.get(obj);`
`f.set(obj, "Faris-kun");`
 - `obj` merupakan objek kelas yang akan diset namanya
- Untuk mengubah field private, ubah dulu accessnya
 - `f.setAccessible(true);`
- Catatan:
 - `getDeclaredFields` mengembalikan semua private, protected, package, dan public field dari kelas, namun tidak termasuk field yang diinherit
 - `getFields` mengembalikan hanya public field dari kelas, namun termasuk field yang diinherit

Method

- Java memiliki kelas Method (`java.lang.reflect.Method`) yang menggambarkan method pada sebuah kelas
- Ada beberapa cara mendapatkan method:
 - `Method[] m = c.getMethods();`
 - `Method[] m = c.getDeclaredMethods();`
 - `Method m = c.getDeclaredMethod("getName");`
 - mendapatkan method bernama `getName` tanpa parameter
 - `Method m = c.getDeclaredMethod("setName", String.class);`
 - mendapatkan method bernama `setName` dengan 1 argumen bertipe `String`
- Setelah mendapatkan method, ada banyak yang bisa dilakukan, misal menginvoke method, mendapatkan nama method, mendapatkan tipe return, mendapatkan daftar exception yang mungkin dilempar, dll
- Daftar lengkapnya ada di <https://docs.oracle.com/javase/8/docs/api/java/lang/reflect/Method.html>

Method

- Sama seperti field, kita bisa menginvoke private method menggunakan
 - `m.setAccessible(true);`
- Contoh menginvoke method:
 - `Method m = c.getDeclaredMethod("getName");`
`m.invoke(obj);`
 - `Method m = c.getDeclaredMethod("setName", String.class);`
`m.invoke(obj, "Tuan Mor");`
 - `obj` merupakan objek kelas yang akan diset namanya
- Catatan:
 - `getDeclaredMethods` mengembalikan semua private, protected, package, dan public method dari kelas, namun tidak termasuk method yang diinherit
 - `getMethods` mengembalikan hanya public method dari kelas, namun termasuk method yang diinherit

Contoh Reflection

- Pernah menggunakan FXML pada JavaFX?
- Fun fact: tidak susah loh membuat fxml.
- Catatan: Tugas Besar 2 gak boleh pake ini yah.
- Misal ada kelas ini:

```
class CardView {  
    @FXML  
    private Label cardName;  
}
```
- Dan file fxml:

```
<Pane>  
    <children>  
        <Label id="cardName" text="Steve" />  
    </children>  
</Pane>
```

Contoh Reflection

- Kita bisa membuat Loader yang membaca file fxml.
- Lalu untuk setiap children:
 - Kita buat elemennya (baca semua properti di xml juga)
 - Tambahkan elemennya ke children
 - Cari field di kelas dengan id yang sesuai dan ada anotasi @FXML
 - Set value dari field itu ke elemen yang dibuat
- Jadi FXML di JavaFX tidak “magic”, tapi justru membantu memudahkan kita dalam membuat tampilan GUI

Catatan

- Banyak hal yang bisa dilakukan dengan reflection
- Tapi untuk praktikum kali ini, kita hanya akan fokus ke Class, Method, dan Field



Sekian.

~~Ditunggu praktikum dan tutorial berikutnya~~

Selamat, Anda sudah mencapai praktikum terakhir!
