MINGGU 13

Praktikum Pemrograman Berbasis Objek
oleh Asisten IF2210 2019/2020

OUTLINE

1. Apa dan kenapa reflection?
2. Class
3. Field
4. Method

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 :)
- 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 :)

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("Hojun");`
    `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, "Qila");
    ```
    - 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, "Hojun");
    ```
    - 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 PEGGUNAAN REFLECTION

- Pernah menggunakan FXML pada JavaFX?
- Fun fact: tidak susah loh membuat fxml
- Misal ada kelas ini:

```
class CardView {
    @FXML
    private Label cardName;
}
```

- Dan file fxml:

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

CONTOH PEGGUNAAN 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 TAMBAHAN

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