**Java Reflection**

## **Pendahuluan**

**Sebelumnya harus sudah paham materi terkait** :

* Java Dasar
* Java Object Oriented Programming
* Apache Maven
* Java Unit Test

**Agenda**

* Pengenalan Java Reflection
* Class
* Fields
* Method
* Property
* Constructor
* Array
* Annotation
* Parameterized Type
* Dan lain-lain

## **Pengenalan Java Reflection**

* Java Reflection merupakan fitur Java dimana memperbolehkan program Java untuk mempelajari atau memodifikasi dirinya sendiri
* Misal saja kitab isa melihat struktur sebuah Java class dari mulai semua fields, method, constructor dan lain -lain saat aplikasi berjalan
* Bahkan Java Reflection bisa digunakan untuk memodifikasi kode program yang sedang berjalan
* Java Reflection banyak sekali digunakan oleh framework-framework karena memang sangat powerfull sekali

**Package Java Reflection**

* Fitur Java Reflection terdapat dalam dua package, yaitu java.lang dan java.lang.reflect
* <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/package-summary.html>
* <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/reflect/package-summary.html>
* Ada banyak sekali class yang terdapat dalam Java Reflection, dan akan kita coba baahs satu persatu dalam kelas ini

## **Class**

**Class<T>**

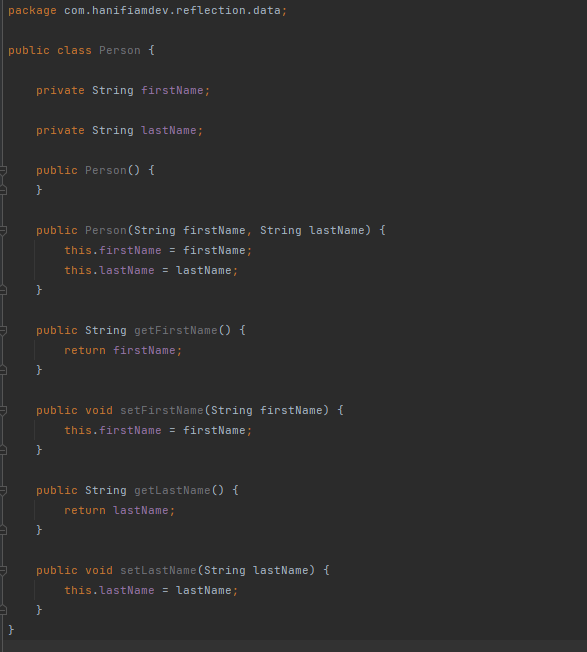
* Java.lang.Class<T> merupakan representasi dari reflection untuk Java Class, Interface dan Enum
* Saat kita membuat Java Class, Interface atau Enum, kadang kita menambahkan field dan method
* Dengan kemampuan java.lang.Class. kita bisa membaca seluruh data member yang terdapat pada Java Class, Interface atau Enum pada saat aplikasi nya berjalan
* <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/Class.html>

**Membuat Class<T>**

* Untuk membuat Class<T>, kitab isa melakukan beberapa cara
* Cara yang pertama dengan meggunakan kata kunci .class setelah nama Java Class, Interface atau Enum nya, misal Person.class, Repository.class, atau Gender.class
* Atau kita juga bisa membuat Class<T> dari sebuah String, menggunakan static method Class.forClass(“com.example.blabla.Person”)
* Atau kita juga bisa mengambil Class<T> dari object, dengan menggunakan method getClass()

**Class<T> Method**

* Class<T> memiliki banyak sekali method
* Seperti untuk mendapatkan method, fields, constructor, annotation, superclass, interface, dan lain – lian
* Semua detail method nya bis akita baca di Javadoc
* <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/Class.html>



Membuat unit test



