

Yang dipelajari pada tutorial ini

Saya mempelajari bahwa hasil dari pengambilan data database dapat menggunakan anotasi @Results dan @Result. Dengan menggunakan @Results dan @Result, maka kita tetap dapat menjamin OOP dari java itu sendiri dikarenakan @Results dan @Result mendukung penggunaan model (di dalam latihan ini yaitu StudentModel dan CourseModel).

Latihan

1. Ubah method selectAllStudents pada kelas StudentMapper agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.

viewall.html :

```
<body>
  <h1>All Students</h1>

  <div th:each="student, iterationStatus: ${students}">
    <a th:href="/student/delete/" + ${student.npm}> Delete Data</a><br/><br/>
    <a th:href="/student/update/" + ${student.npm}> Update Data</a><br/>
    <h3 th:text="'No. ' + ${iterationStatus.count}>No. 1</h3>
    <h3 th:text="'NPM = ' + ${student.npm}>Student NPM</h3>
    <h3 th:text="'Name = ' + ${student.name}>Student Name</h3>
    <h3 th:text="'GPA = ' + ${student.gpa}>Student GPA</h3>

    <h3>Kuliah yang diambil</h3>
    <ul th:each="course, courseIteration: ${student.courses}">
      <li th:text="${course.name} + ' - ' + ${course.credits} + ' SKS'">
        Nama Kuliah - X SKS
      </li>
    </ul>
  </div>
</body>
```

StudentController.java:

```
@RequestMapping("/student/viewall")
public String view (Model model)
{
    List<StudentModel> students = studentDAO.selectAllStudents ();
    model.addAttribute ("students", students);

    return "viewall";
}
```

StudentMapper.java:

```
@Select("select npm, name, gpa from student")
@Results(value = {
    @Result(property="npm", column="npm"),
    @Result(property="name", column="name"),
    @Result(property="gpa", column="gpa"),
    @Result(property="courses", column="npm",
        javaType = List.class,
        many = @Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents ();

@Select("select course.id_course, name, credits " +
    "from studentcourse join course " +
    "on studentcourse.id_course = course.id_course " +
    "where studentcourse.npm = #{npm}")
List<CourseModel> selectCourses(@Param("npm") String npm);
```

StudentModel.java:

```
package com.example.model;

import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class StudentModel
{
    @NotNull
    private String npm;

    @NotNull
    private String name;

    @NotNull
    private double gpa;

    private List<CourseModel> courses;

}
```

Teuku Amrullah Faisal

1506723231

APAP - C

2. Buatlah view pada halaman `http://localhost:8080/course/view/{id}` untuk Course sehingga dapat menampilkan data course beserta Student yang mengambil.

view-course.html:

```
<head>
  <title>View Course by ID</title>
</head>
<body>
  <h3 th:text="'ID = ' + ${course.idCourse}>ID</h3>
  <h3 th:text="'Nama = ' + ${course.name}>Nama</h3>
  <h3 th:text="'SKS = ' + ${course.credits}>SKS</h3>

  <h3>Mahasiswa yang Mengambil</h3>
  <ul th:each="student, iterationStatus: ${course.students}">
    <li th:text="${student.npm} + ' - ' + ${student.name}>
      Nama Kuliah - X SKS
    </li>
  </ul>
</body>
```

CourseModel.java:

```
package com.example.model;

import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class CourseModel
{
    @NotNull
    private String idCourse;

    @NotNull
    private String name;

    @NotNull
    private Integer credits;

    private List<StudentModel> students;
}
```

StudentServiceDatabase.java:

```
@Override
public CourseModel selectCourse(String id) {
    CourseModel course = studentMapper.selectCourse(id);
    log.info ("select course with id " + course.getIdCourse());
    return course;
}
```

StudentMapper.java:

```
@Select("select id_course, name, credits from course where id_course = #{id}")
@Results(value = {
    @Result(property="idCourse", column="id_course"),
    @Result(property="name", column="name"),
    @Result(property="credits", column="credits"),
    @Result(property="students", column="id_course",
        javaType = List.class,
        many = @Many(select="selectStudentCourses"))
})
CourseModel selectCourse(@Param("id") String id);

@Select("select student.npm, name, gpa " +
    "from studentcourse join student " +
    "on studentcourse.npm = student.npm " +
    "where studentcourse.id_course = #{id}")
List<StudentModel> selectStudentCourses(@Param("id") String id);
```

StudentController.java:

```
@RequestMapping("/course/view/{id}")
public String viewCourse(Model model,
    @PathVariable(value = "id") String id) {

    CourseModel course = studentDAO.selectCourse(id);

    model.addAttribute("course", course);

    return "view-course";
}
```