

1. Hal yang dipelajari

Pada tutorial kali ini, saya mempelajari cara penggunaan database yaitu dengan menggunakan relasi database pada spring. Selain itu, saya juga mempelajari bagaimana pemetaan antara nama atribut pada class model dengan nama column pada database. Saya juga mempelajari bagaimana suatu fungsi dapat menyimpan hasil dari fungsi lain pada mapper.

2. Method Select All Students

```
@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 ();
```

Saya menggunakan cara yang mirip dengan method selectStudent. Dimana akan dilakukan pemetaan column npm ke atribut npm, lalu column name ke atribut name, column gpa ke atribut gpa, serta memetakan hasil dari query yang terdapat pada method selectCourses dengan paramater npm dan menyimpan hasilnya ke atribut courses.

```
@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);

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All Students</title>
    </head>
    <body>
        <h1>All Students</h1>
        <div th:each="student, iterationStatus: ${students}">
            <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>
            <a th:href="/student/delete/' + ${student.npm}">Delete Data</a><br>
            <a th:href="/student/update/' + ${student.npm}">Update Data</a><br>
            <ul th:each="course, iterationStatus: ${student.courses}">
                <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
                    Nama kuliah-X SKS
                </li>
            </ul>
            <hr/>
        </div>
    </body>
</html>
```

3. Method View Course

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

    if (course != null) {
        model.addAttribute ("course", course);
        return "view-course";
    } else {
        model.addAttribute ("id", id);
        return "not-found";
    }
}
```

Method akan menerima url yang bersesuaian lalu memanggil method selectCourseInfo dengan parameter id di kelas StudentService yang kemudian diteruskan ke mapper

```
@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="selectStudentsTakeCourses"))
})
CourseModel selectCourseInfo (@Param("id")String id);
```

Method akan memetakan column id_course ke atribut idCourse, lalu column name ke atribut name, column credits ke atribut credits, serta memetakan hasil dari query yang terdapat pada method selectStudentsTakeCourses dengan paramater id_course dan menyimpan hasilnya ke atribut students.

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

Method akan mengambil data student yang mengambil course dengan idCourse id dan mengembalikannya ke method selectCourseInfo
Lalu program akan menampilkannya di view-course.html

Abdul Raziq Andri
1506757560
Tutorial 5 APAP

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Course Info</title>
  </head>
  <body>
    <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
    <h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
    <h3 th:text="'Credits = ' + ${course.credits}">Course Credits</h3>
    <h3>Mahasiswa yang mengambil</h3>
    <ul th:each="student, iterationStatus: ${course.students}">
      <li th:text="${student.npm} + ' - ' + ${student.name}">
        Id nama
      </li>
    </ul>
  </body>
</html>
```