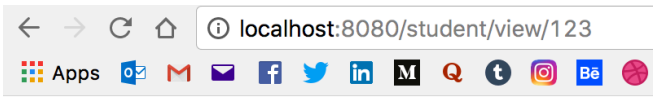
Ilyas Fahreza 1506733371 Kelas B

Tutorial 5

Menambahkan Course Model

Untuk menambahkan class course model yang akan berhubungan dengan student diperlukan sebuah kelas penengah yang menjadi intersection antara student dan course tersebut, yaitu studentcourse. Pada studentcourse data dari masing-masing primary key kedua tabel tersebut disambungkan dan membuat student dan course yang diambilnya berhubungan. Dengan begitu, setiap data student dan course yang ada dapat dilihat hubungan diambil dan mengambilnya pada sistem. Perubahan method dengan penambahan Result yang terdapat pada mapper memungkinkan hal itu terjadi:

```
@Select("select npm, name, gpa from student where npm = #{npm}")
@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"))
})
StudentModel selectStudent (@Param("npm") String npm);
```



NPM = 123

Name = ilyas

GPA = 0.0

Kuliah yang diambil

- PSP-4 sks

Latihan:

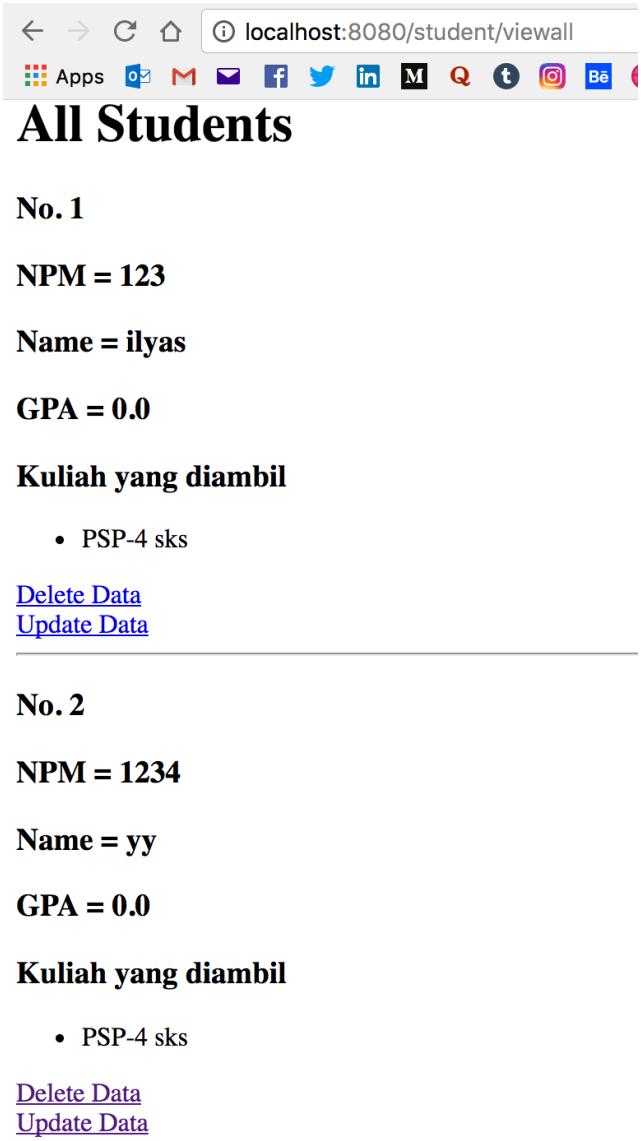
1. Ubah method selectAllStudents pada kelas StudentMapper agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.
   Jawab: Dengan menambahkan code yang sama pada selectStudent pada selectAllStudents menjadi:

```
@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 ();
```

menambahkan code yang sama pada viewAll.html untuk selectAllStudents menjadi:

```
<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>
            <h3>Kuliah yang diambil</h3>
            <ul th:each="course,iterationStatus: ${student.courses}">
                  <li th:text="${course.name} + '-' + ${course.credits} +
' sks'" >
                  Nama kuliah-X SKS
                  </li>
            </ul>
           <a th:href="'/student/delete/' + ${student.npm}" > Delete
Data</a><br/>
           <a th:href="'/student/update/' + ${student.npm}" > Update
Data</a><br/>
            <hr/>
</div>
```



2. Buatlah view pada halaman http://localhost:8080/course/view/{id} untuk Course sehingga dapat menampilkan data course beserta Student yang mengambil.
   Jawab: Dengan membuat sebuah class baru yaitu CourseController yang memiliki fungsi untuk mengambil Course:

```
@RequestMapping("/course/view/{id_course}")
    public String viewPath (Model model,
            @PathVariable(value = "id_course") String id_course)
    {
        CourseModel course = courseDAO.selectCourse(id_course);

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

```
        }
    }
```

serta menambah method untuk salah satu selectCourse() serta menambahkan selectStudents yang mengambil course tertentu, menjadi:

```
@Select("select id_course, name, credits from course where id_course =
#{id_course}")
    @Results(value = {
             @Result(property="id_course", 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="selectStudents"))
             })
       CourseModel selectCourse (@Param("id_course") String id_course);

    @Select("SELECT student.npm, name, gpa " +
             "FROM studentcourse JOIN student " +
             "ON studentcourse.npm = student.npm " +
             "WHERE studentcourse.id_course = #{id_course} ")
    List<StudentModel> selectStudents (@Param("id_course") String
id_course);
```

Selain itu, kita juga menambahkan method baru untuk interface StudentService dan implementasinya pada StudentServiceDatabase menjadi:

StudentService:

```
CourseModel selectCourse (String id_course);
```

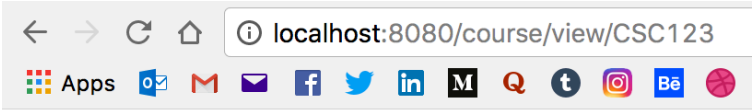
StudentServiceDatabase:

```
@Override
       public CourseModel selectCourse(String id_course) {
             return studentMapper.selectCourse (id_course);
       }
```

Terakhir, kita menambahkan view-course.html untuk menampilkan course dan mahasiswa yang mengambilnya menjadi:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Course</title>
    </head>
    <body>
             <h3 th:text="'ID = ' + ${course.id_course}">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}" >
                        NPM - NAMA
                  </li>
            </ul>
      </body>
</html>
```



ID = CSC123

Name = PSP

Credits = 4

Mahasiswa yang mengambil

- 123-ilyas
- 1234-yy