

Writeup Tutorial 5

a. Rangkuman

Pada tutorial minggu ini, saya mempelajari bagaimana cara mengakses dan mengelola hubungan many-to-many yang terdapat pada database dengan menggunakan Spring Boot dan Thymeleaf. Selain itu saya juga mempelajari cara untuk memanggil method lain dalam sebuah method untuk menampilkan data tertentu dari gabungan beberapa tabel.

b. Latihan

1. Mengubah method `selectAllStudents` agar halaman `viewall` menampilkan semua student beserta daftar mata kuliah yang diambil

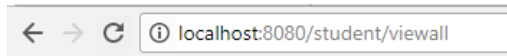
Method `selectAllStudents` di `StudentMapper`:

```
//changed
@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 courses taken by a particular student
@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);
```

Method `selectAllStudents()` akan menampilkan student hasil output query pada method `selectCourses()`, yaitu daftar course yang diambil oleh student dengan NPM tertentu. Anotasi `@Result` akan memetakan hasil query select ke class model.

Tampilan halaman viewall.html:



All Students

No. 1

NPM = 12345

Name = xyz

GPA = 3.5

Kuliah yang diambil

- DDP1-4 sks
- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

No. 2

NPM = 22222

Name = budi

GPA = 4.0

Kuliah yang diambil

- PSP-4 sks
- DDP1-4 sks

2. Membuat view pada halaman /course/view/{id} agar menampilkan data course beserta Student yang mengambil course tersebut

Method viewCourse pada StudentMapper.java:

```
//view course detail
@Select("SELECT id_course, name, credits FROM course where id_course = #{idCourse}")
@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 = "selectStudents"))
})
CourseModel viewCourse(@Param("idCourse") String id_course);
```

```
//select students that are enrolled on a particular course
@Select("SELECT student.npm, name "
      + "FROM studentcourse JOIN student "
      + "ON student.npm = studentcourse.npm "
      + "WHERE studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudents(@Param("id_course") String id_course);
```

Method viewCourse() akan menampilkan data course dengan ID tertentu dan detail mahasiswa yang mengambil course tersebut. Property “students” didapat dari hasil output query yang dijalankan method selectStudents().

View detail masing-masing course:



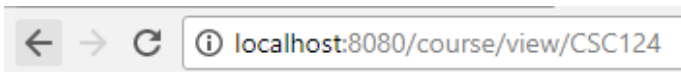
ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 22222 - budi
- 33333 - c



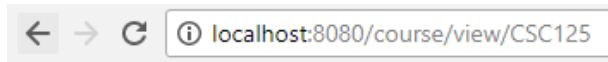
ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 33333 - c



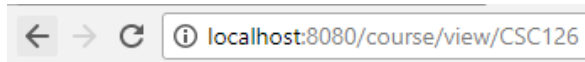
ID = CSC125

Nama = DDP1

SKS = 4

Mahasiswa yang mengambil

- 12345 - xyz
- 22222 - budi



ID = CSC126

Nama = MPKT

SKS = 6

Mahasiswa yang mengambil

- 12345 - xyz
- 22222 - budi