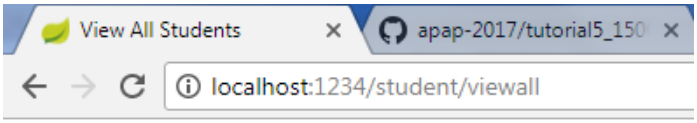
1. Ubah method selectAllStudents pada kelas StudentMapper agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.



All Students

No. 1

NPM = 123

Name = koi

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

Delete Data
Update Data

No. 2

NPM = 124

Name = lala

GPA = 4.0

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

Delete Data
Update Data

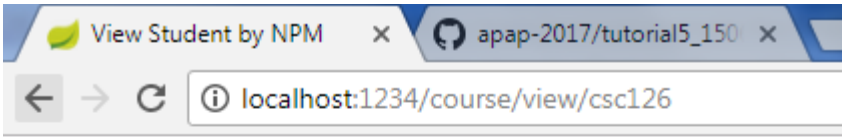
Pertama saya mengubah method selectAllStudents() pada studentMapper dengan menambahkan property courses yang merupakan List dari CourseModel.

```
@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="selectAllCourses"))
    })
    List<StudentModel> selectAllStudents ();
```

Lalu saya membuat method selectAllCourses. Method ini bertujuan untuk menselect data dari tabel course dengan menyamakan id_course yang ada di tabel studentcourse dan id_course yang ada di tabel course serta menyamakan juga dengan npm yang ada di tabel studentcourse dengan npm dari parameter. Ini bertujuan untuk memfilter data mata kuliah apa saja yang diambil oleh seorang mahasiswa.

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

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



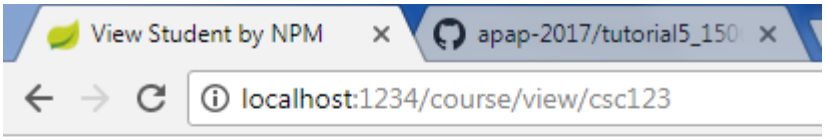
ID = CSC126

Nama = MPKT

SKS = 6

Mahasiswa yang mengambil

- 123-koi



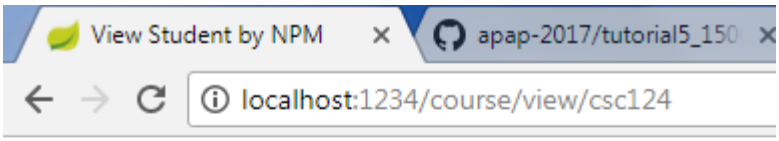
ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124-lala



ID = CSC124

Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 124-lala

Pertama saya membuat method selectCourse yang menerima parameter string id_course dan method selectStudentCourses() di StudentService

```
CourseModel selectCourse (String id_course);

List<StudentModel> selectStudentCourses ();
```

Lalu membuat ini di StudentServiceDatabase

```
@Override
    public List<StudentModel> selectStudentCourses ()
    {
        Log.info ("select all courses and students");
        return studentMapper.selectStudentCourses();
       }

@Override
    public CourseModel selectCourse (String id_course)
    {
        Log.info ("select course with id {}", id_course);
        return studentMapper.selectCourse(id_course);
       }
```

Lalu membuat method ini di StudentMapper

```
@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="selectStudentCourses"))
    })
       CourseModel selectCourse (@Param("id_course") String id_course);


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

Method ini bertujuan untuk menselect data dari tabel student dengan menyamakan npm yang ada di tabel studentcourse dan npm yang ada di tabel student serta menyamakan juga dengan id_course yang ada di tabel studentcourse dengan id_Course dari parameter. Ini bertujuan untuk memfilter data siapa saja mahasiswa yang mengambil pada suatu mata kuliah tertentu.

Apa yang saya pelajari dari tutorial ini adalah bagaimana cara membuat relasi antar suatu entiti di database. Yaitu dengan membuat entiti baru atau tabel di database. Lalu membuat model dari suatu entiti yang baru itu dengan menambahkan class-class yang diperlukan.