

Enrico jano Fiandi
1506689490
ADPAP – A

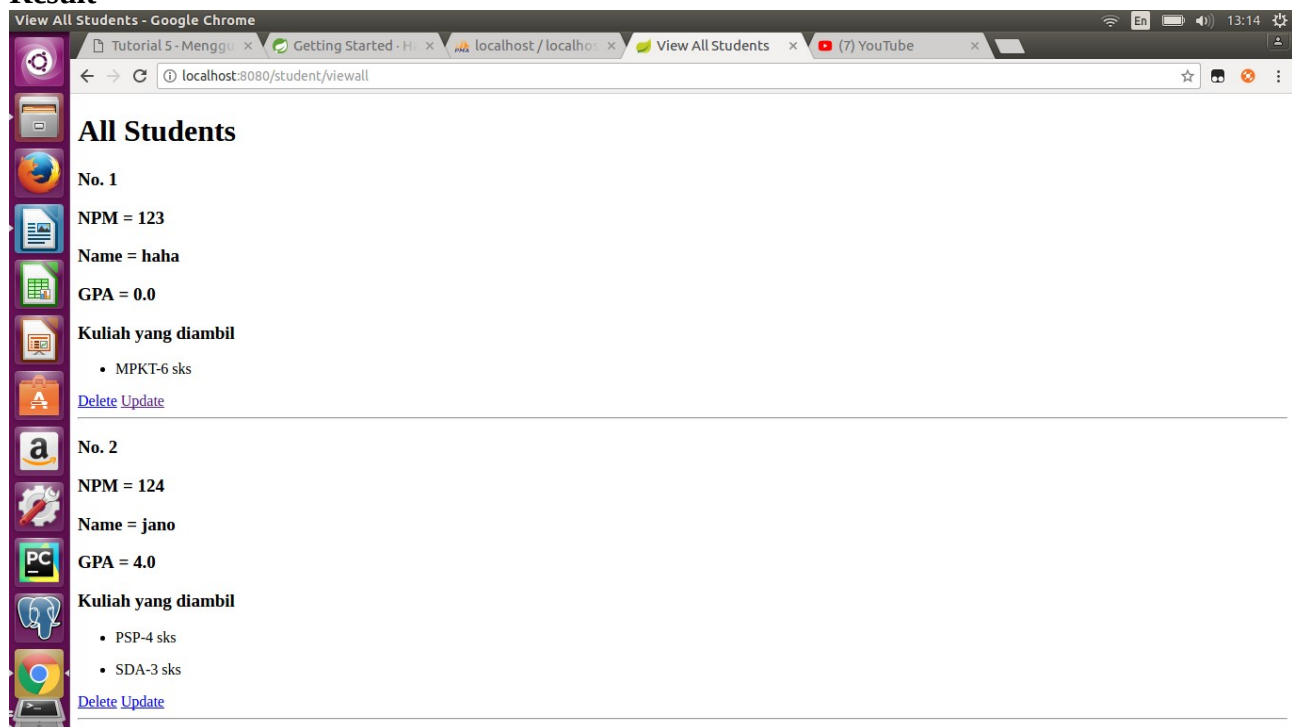
1. Merubah SelectAllStudents

```
@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 ();
```

query untuk mengambil course yang diambil oleh 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);
```

Result



The screenshot shows a web browser window with the URL `localhost:8080/student/viewall`. The page title is "All Students". It displays a list of two students:

- No. 1**
 - NPM = 123
 - Name = haha
 - GPA = 0.0
 - Kuliah yang diambil
 - MPKT-6 sks
 - [Delete](#) [Update](#)
- No. 2**
 - NPM = 124
 - Name = jano
 - GPA = 4.0
 - Kuliah yang diambil
 - PSP-4 sks
 - SDA-3 sks
 - [Delete](#) [Update](#)

Penjelasan

Sama halnya seperti pada query `selectStudent`, saya menggunakan anotasi `@Result` dan `@Results` untuk melakukan mapping records dari database ke `List<StudentModel>`, dan menggunakan anotasi `@Many` untuk menyimpan Course yang diambil oleh setiap student, hal ini dikarenakan hubungan table student dengan studentcourse adalah *one-to-many* sehingga saya menggunakan anotasi `@Many` dan nantinya hasil query dari method `selectCourses` akan disimpan kedalam variable `courses` didalam model `StudentModel` dalam bentuk `List<CourseModel>`

2. Menambahkan view pada course

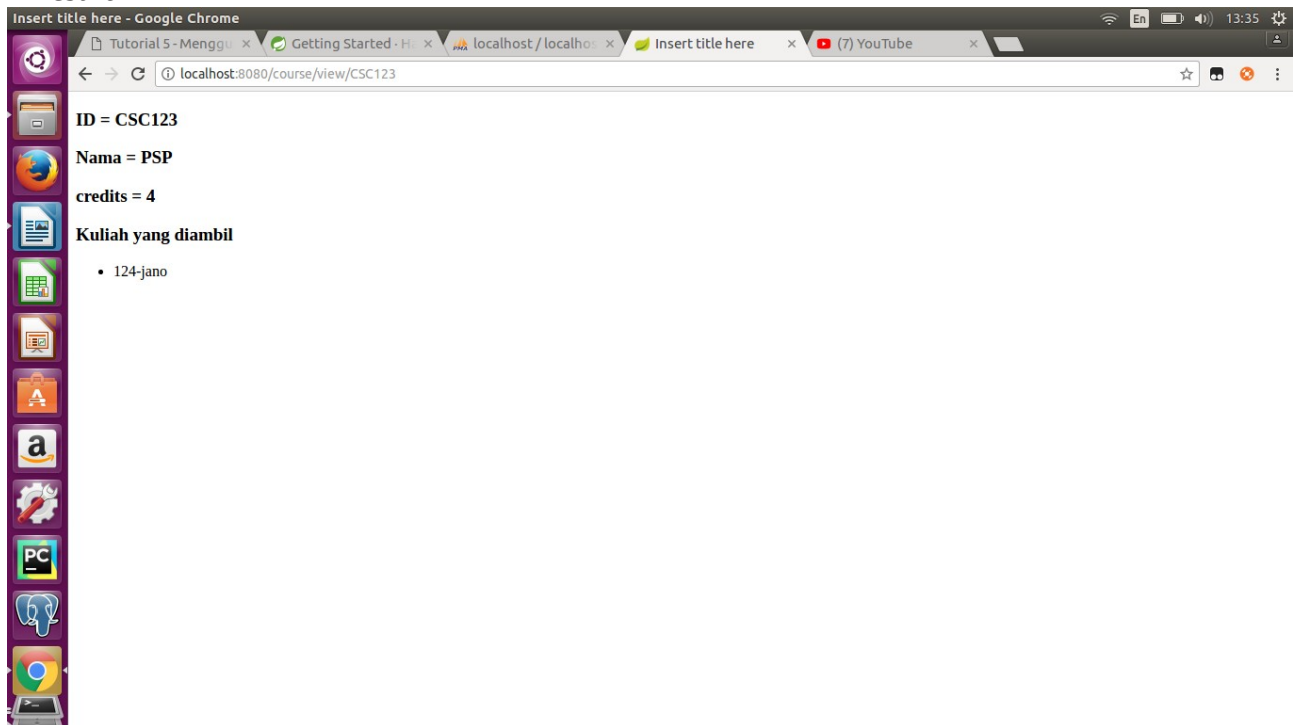
Method selectCourse

```
@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="selectParticipants"))
})
CourseModel selectCourse(@Param("id") String id);
```

Query untuk mengambil student yang mengikuti suatu course

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

Result



Penjelasan

Pertama” saya menambahkan variable `List<StudentModel> students` di dalam `CourseModel`, lalu saya melakukan query untuk menampilkan course dengan bantuan anotasi `@Results` dan `@Result` untuk melakukan proses mapping record dari database ke objek `CourseModel`. Untuk mendapatkan list student yang mengambil suatu course saya menggunakan query join antara table `studentcourse` dan `student` (method mapper `selectParticipant`) yang nantinya akan disimpan di dalam variable `students` di dalam objek `CourseModel`, saya menggunakan anotasi `@Many` karena hubungan antara table `student` dan table `student course` adalah *one-to-many*.

Hal yang saya pelajari

Pada tutorial ini saya mempelajari bagaimana cara melakukan operasi join antara 2 table di dalam menggunakan MyBatis. Saya juga belajar bagaimana cara menggunakan anotasi `@Results`, `@Result`, `@Many`, dan `@One` (anotasi `@One` saya pelajari melalui Google) untuk melakukan mapping records dari database ke Objek Model.