

## Write Up Tutorial 5

### 1. Penjelasan pada perubahan method selectAllStudents

Pada method selectAllStudent di StudentMapper dilakukan perubahan sebagai berikut

```
@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 ();
```

Perubahan dilakukan dengan menambahkan anotasi @Results seperti yang dilakukan di method selectStudent.

Di viewAll.html juga dilakukan penambahan seperti berikut

```
<h3>Kuliah yang akan diambil</h3>
<ul th:each="course, iterationStatus: ${student.courses}">
    <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
        Nama kuliah-X SKS
    </li>
</ul>
```

Adapun hasil yang diperoleh ketika menampilkan localhost:8080/student/viewall akan seperti ini

# All Students

No. 1

NPM = 100

Name = ChanneK

GPA = 3.33

[Delete Data](#)

[Update Data](#)

Kuliah yang akan diambil

---

No. 2

NPM = 123

Name = Sergio Agüero

GPA = 3.77

[Delete Data](#)

[Update Data](#)

2. Penjelasan untuk melihat course sesuai dengan id nya beserta student yang mengambilnya

Pada dasarnya tahapan-tahapan yang dilakukan untuk fitur ini mirip dengan apa yang sudah dijelaskan di file tutorial mengenai melihat student beserta courses yang diambilnya. Yang pertama kali dilakukan adalah menambahkan method viewCoursePath di StudentController

```
@RequestMapping("/course/view/{idCourse}")
public String viewCoursePath (Model model,
    @PathVariable(value = "idCourse") String idCourse)
{
    CourseModel course = studentDAO.selectCourse (idCourse);

    if (course != null) {
        model.addAttribute ("course", course);
        return "viewCourse";
    }
}
```

```

    } else {
        model.addAttribute ("idCourse", idCourse);
        return "CourseNotFound";
    }
}

```

Setelah itu, dilakukan penambahan di StudentService.

```
CourseModel selectCourse(String idCourse);
```

Method selectCourse ini kemudian diperjelas dengan memanggil mapper di StudentServiceDatabase

```

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

```

Pada StudentMapper dilakukan penambahan method selectCourse. Penambahan anotasi @Results untuk menghubungkan course yang dipilih dari query dengan suatu method yang akan memanggil query untuk melihat semua student yang memilih course tersebut yang kemudian dipetakan ke List students di CourseModel yang sebelumnya telah dibuat.

```

@Select("select id_course, credits, name 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 selectCourse (@Param("idCourse") String idCourse);

```

```

@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);

```

Tidak lupa juga ditambahkan layer view

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Course by ID</title>
    </head>
    <body>
        <h3 th:text="'ID = ' + ${course.idCourse}">Class ID</h3>
        <h3 th:text="'Nama = ' + ${course.name}">Class Name</h3>
        <h3 th:text="'SKS = ' + ${course.credits}">Class Credits</h3>
    </body>
</html>

```

```

        <h3>Student yang mengambil</h3>
        <ul th:each="student, iterationStatus: ${course.students}">
            <li th:text="${student.name} + ' - NPM : ' + ${student.npm}">
                Nama - NPM :-X
            </li>
        </ul>
    </body>
</html>

```

Berikut adalah view ketika course tidak ditemukan

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Class not found</title>
    </head>
    <body>
        <h1>Class not found</h1>
        <h3 th:text="'ID = ' + ${idCourse}">Class Id</h3>
    </body>
</html>

```

Tampilan pada <http://localhost:8080/course/view/CSC123>

**ID = CSC123**

**Nama = PSP**

**SKS = 4**

**Student yang mengambil**

- Sergio Aguero - NPM : 123
- Chritian Abbiati - NPM : 124
- den - NPM : 127