

Tutorial 5 APAP

1. Method yang diubah pada Latihan Merubah **SelectAllStudents**.

a. Mengubah method 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 ();
```

Method ini menghubungkan dirinya dengan method selectCourses agar saat kita melakukan select pada variable List<Course> juga akan terisi. Method ini tidak jauh berbeda dengan method selectStudent(), hanya saja method ini untuk melihat semua student yang ada pada database yang mengambil suatu matakuliah dan menampilkannya(method /student/viewall pada controller).

b. Mengubah view-all.html

```
<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>
```

Ini untuk menampilkan matakuliah yang diambil oleh semua mahasiswa yang ada pada database.

2. Method yang dibuat pada Latihan Menambahkan **View** pada **Course**.

a. Method CourseModel viewCourse(String idCourse); pada StudentService.java

Method ini nantinya untuk mengambil data course dari database.

b. Implementasi method CourseModel viewCourse(String idCourse); pada StudentServiceDatabase.java

```
@Override
public CourseModel viewCourse(String idCourse) {
    log.info("course" + idCourse);
    return studentMapper.viewCourse(idCourse);
}
```

Method ini mengimplementasikan method yang sudah dibuat pada StudentService.java

c. Method select pada StudentMapper.java

```
@Select("select * 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="joinCourse"))
})
CourseModel viewCourse (@Param("idCourse") String idCourse);
```

Method ini untuk mengambil(Select) data pada table course didatabase. Method ini menghubungkan dirinya dengan method joinCourse.

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

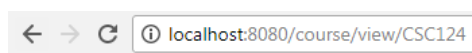
Method ini berisikan query yang akan melakukan join antara table studentcourse dengan student berdasarkan npm dan di filter berdasarkan idCourse yang diberikan.

d. Method pada Controller

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

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

Method ini nanti nya akan menampilkan view-course.html dihalaman browser ketika menggunakan /course/view/idCourse. Halaman browser nantinya akan menampilkan seperti gambar dibawah ini:



ID = CSC124

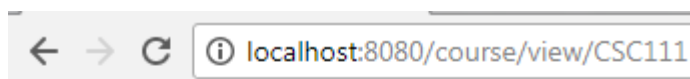
Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 124-hehe

Jika ID tidak ditemukan, maka halaman browser akan menampilkan seperti gambar dibawah ini:



Course not found

ID = CSC111

e. html view-course dan not-found-course

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>View Course by ID</title>
5   </head>
6   <body>
7     <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
8     <h3 th:text="'Nama = ' + ${course.name}">Course Name</h3>
9     <h3 th:text="'SKS = ' + ${course.credits}">SKS </h3>
10
11     <h3>Mahasiswa yang mengambil</h3>
12     <ul th:each="student, iterationStatus: ${course.students}">
13       <li th:text="${student.npm} + '-' + ${student.name}">
14         Nama mahasiswa
15       </li>
16     </ul>
17   </body>
18 </html>

```

Nanti nya akan menampilkan data-data course bawa halaman browser jika ID course ditemukan.

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Course not found</title>
5   </head>
6   <body>
7     <h1>Course not found</h1>
8     <h3 th:text="'ID = ' + ${idCourse}">Student NPM</h3>
9   </body>
10 </html>

```

Nantinya akan menampilkan pada halaman browser jika id course tidak ditemukan.