

Faiz Muhammad Refano
1506721895
APAP-A

Yang Saya dapatkan pada tutorial 5 ini adalah menambahkan table pada program dengan membuat controller course ,course mapper,course service yang sudah diajarkan pada tutorial 4 minggu lalu untuk membuat output sesuai dengan yang diminta oleh soal.

1.Method SelectAllStudent yang sudah diubah:

```
@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 ();
```

Perubahannya adalah dengan menambahkan @Result dan menambahkan property dan column dimana guna @Result adalah untuk memetakan hasil dari query ke coursemodel.

2.Method yang ditambahkan untuk menview Course adalah:

-Saya membuat CourseMapper baru yang berisi

```
@Mapper
public interface CourseMapper {
    @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="selectStudent"))
    })
    CourseModel selectCourse(@Param("id")String id);

    @Select("SELECT student.npm , name FROM studentcourse join student ON studentcourse.npm = student.npm WHERE studentcourse.id_course =#{id}")
    List<StudentModel> selectStudent(@Param("id") String id);
}
```

Dimana CourseMapper ini berisi method untuk menselect course dan juga terdapat query untuk memilih student npm

Faiz Muhammad Refano
1506721895
APAP-A

dan juga nama student yang nanti akan ditampilkan pada view.

-Saya Membuat CourseController baru yang berisi:

```
@Controller
public class CourseController {
    @Autowired
    CourseService courseDAO;

    @RequestMapping("/course/view/{id}")
    public String viewCourse (Model model, @PathVariable(value="id") String id)
    {
        CourseModel course = courseDAO.selectCourse(id);
        if(course!=null) {
            model.addAttribute("course" , course);
            return "viewCourse";
        }
        else {
            model.addAttribute("id" , id);
            return "not-found-course";
        }
    }
}
```

Method ini berisi path yang akan diakses oleh kita dimana method ini berisi tentang pengecekan terhadap course apakah ada isinya atau tidak jika ada maka akan diarahkan ke viewCourse untuk menampilkan output yang sesuai dengan permintaan soal. Jika Course tidak ada isinya maka akan di arahkan ke not-found-course dimana outputnya berupa tulisan course tidak ada.

-Tampilan View Course

```
<title>View Course by id</title>
</head>
<body>
<h3 th:text="'ID = ' + ${course.idCourse}">Course Id</h3>
<h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
<h3 th:text="'SKS = ' + ${course.credits}">Course Credits</h3>

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

File ViewCourse berguna untuk menampilkan Course Name ,Course Id ,dan Course Credits.

Faiz Muhammad Refano
1506721895
APAP-A

-Tampilan not-found-course

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

file not-found-course berguna untuk menampilkan output error jika id course yang kita cari tidak ada pada database.