

## TUTORIAL 5

### Menggunakan Database serta Relasi Database dalam Project Spring Boot

#### MENAMBAHKAN MODEL

1. Menambah Model baru dengan membuat CourseModel pada package com.example.model

```
StudentController.java CourseModel.java ⌵
1 package com.example.model;
2
3 import java.util.List;
4
5 import lombok.AllArgsConstructor;
6 import lombok.Data;
7 import lombok.NoArgsConstructor;
8
9 @Data
10 @AllArgsConstructor
11 @NoArgsConstructor
12 public class CourseModel {
13     private String idCourse;
14     private String name;
15     private Integer credits;
16     private List<StudentModel> students;
17 }
```

2. Menambahkan list of CourseModel pada StudentModel

```
package com.example.model;

import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class StudentModel
{
    private String npm;
    private String name;
    private double gpa;
    private List<CourseModel> courses;
}
```

pada StudentController tambahkan parameter constructor baru seperti berikut.

```
StudentModel student = new StudentModel (npm, name, gpa, null);
studentDAO.addStudent (student);
```

3. Selanjutnya, membuat method pada StudentMapper yang mengembalikan list of course pada Student dengan npm tertentu.

```
@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);
```

\*query tersebut akan melakukan join antara table studentcourse dengan course berdasarkan id\_course dan difilter hanya pada student dengan npm yang diberikan.

4. Menghubungkan method selectStudent dengan method selectCourses agar saat melakukan select maka variable List<Course> juga akan terisi. Lalu ubah method pada class StudentMapper sebagai berikut.

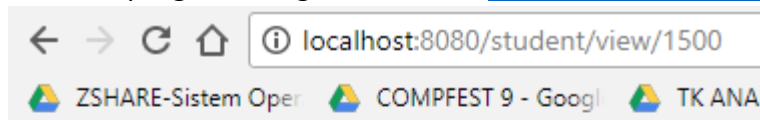
```
@Select("select npm, name, gpa from student where npm = #{npm}")
@Results(value = {
    @Result(property="npm", column="npm"),
    @Result(property="name", column="name"),
    @Result(property="gpa", column="gpa"),
    @Result(property="courses", column="npm",
        javaType = List.class,
        many=@Many(select="selectCourses"))
})
StudentModel selectStudent (@Param("npm") String npm);
```

5. Menambahkan view pada view.html untuk menampilkan list kuliah yang diambil oleh mahasiswa seperti berikut.

```
<body>
    <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
    <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
    <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>

    <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>
</body>
```

6. Jalankan program dengan membuka <http://localhost:8080/student/view/1500>



**NPM = 1500**

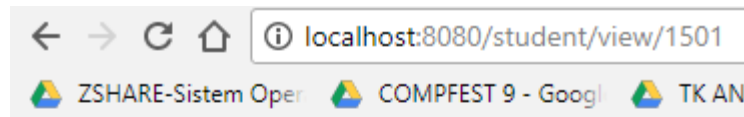
**Name = fizhasi**

**GPA = 3.98**

**Kuliah yang diambil**

- MPKT-6 sks

dan <http://localhost:8080/student/view/1501>



**NPM = 1501**

**Name = chanek**

**GPA = 3.45**

**Kuliah yang diambil**

- PSP-4 sks

#### LATIHAN

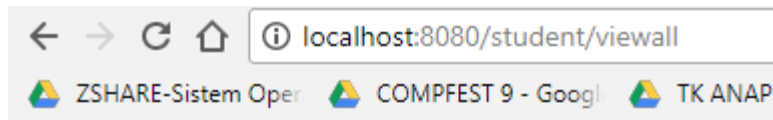
1. Ubah method selectAllStudents pada kelas StudentMapper agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.

```
@Select("select npm, name, gpa from student")
@Results(value = {
    @Result(property="npm", column="npm"),
    @Result(property="name", column="name"),
    @Result(property="courses", column="npm",
        javaType = List.class,
        many=@Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents ();
```

tambahkan pada viewall.html

```
<h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
<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>
```

2. Buka <http://localhost:8080/student/viewall>



## All Students

**No. 1**

**NPM = 1500**

**Name = fizhasi**

**GPA = 3.98**

**Kuliah yang diambil**

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

---

**No. 2**

**NPM = 1501**

**Name = chanek**

**GPA = 3.45**

**Kuliah yang diambil**

- PSP-4 sks

[Delete Data](#)

[Update Data](#)

**No. 3**

**NPM = 1502**

**Name = hera**

**GPA = 3.4**

**Kuliah yang diambil**

- SDA-3 sks

[Delete Data](#)

[Update Data](#)

---

## LATIHAN 2

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

1. Tambahkan interface CourseMapper

```
@Mapper
public interface CourseMapper {

    @Select("select id_course, name, credits from course where id_course = #{id_course}")
    @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("id_course") String id_course);

    @Select("select student.npm as 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);
}
```

2. Tambahkan interface CourseService dan class CourseServiceDatabase

```
package com.example.service;

import com.example.model.CourseModel;

public interface CourseService {
    CourseModel selectCourse (String id_course);
}
```

dan class CourseServiceDatabase  
course yang diambil

```
package com.example.service;

import org.springframework.beans.factory.annotation.Autowired;

@Slf4j
@Service
public class CourseServiceDatabase implements CourseService{
    @Autowired
    private CourseMapper courseMapper;

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

3. Membuat class CourseController.

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

    @RequestMapping("/course/view/{id}")
    public String viewPath(Model model, @PathVariable(value="id") String id_course) {
        CourseModel course = courseDAO.selectCourse(id_course);

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

4. Buat view-course.html untuk menampilkan list course yang diambil mahasiswa tertentu.

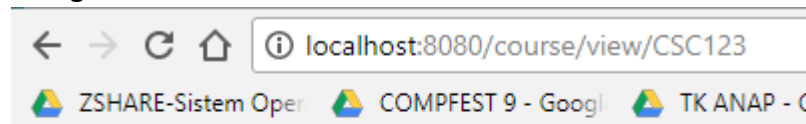
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Course</title>
  </head>
  <body>
    <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
    <h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
    <h3 th:text="'GPA = ' + ${course.credits}">Credits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

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

5. Buat not-found-course.html

```
<!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 Course</h3>
  </body>
</html>
```

6. Buka localhost:8080/course/view/CSC123 untuk melihat daftar mahasiswa yang mengambil course CSC123



**ID = CSC123**

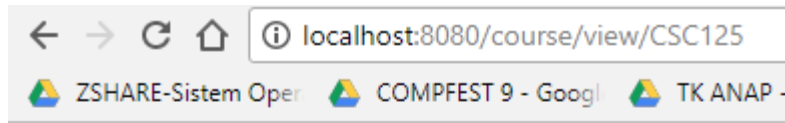
**Name = PSP**

**GPA = 4**

**Mahasiswa yang mengambil**

- 1501 - chanek

7. Buka localhost:8080/course/view/CSC125 untuk melihat list jika tidak ada mahasiswa yang mengambil course tersebut



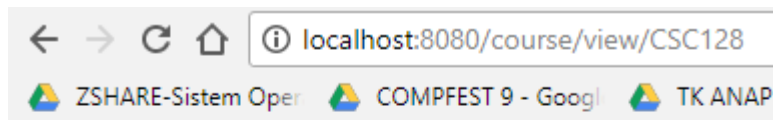
**ID = CSC125**

**Name = DDP 1**

**GPA = 4**

**Mahasiswa yang mengambil**

8. Buka localhost:8080/course/view/CSC128 untuk menampilkan page not found jika course tidak terdaftar



**Course not found**

**ID = CSC128**