Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

Menggunakan Database serta Relasi Database dalam Project Spring Boot

MENAMBAHKAN MODEL

1. Membuat CourseModel pada package com.example.model

```
package com.example.model;

import java.util.List;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class CourseModel {
    private String idCourse;
    private String name;
    private Integer credits;
    private List<StudentModel> students;
}
```

2. Menambahkan list of CourseModel pada StudentModel

```
public class StudentModel
{
     private String npm;
     private String name;
     private double gpa;
     private List<CourseModel> courses;

}
```

Pada StudentController, mengubah inisalisasi constructor menjadi:

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

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

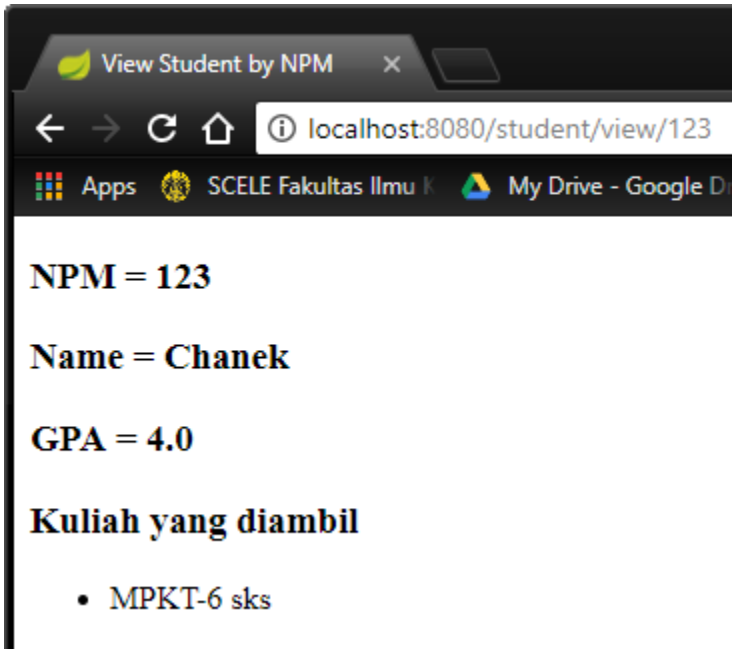
3. Membuat method pada StudentMapper

```
@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);
```

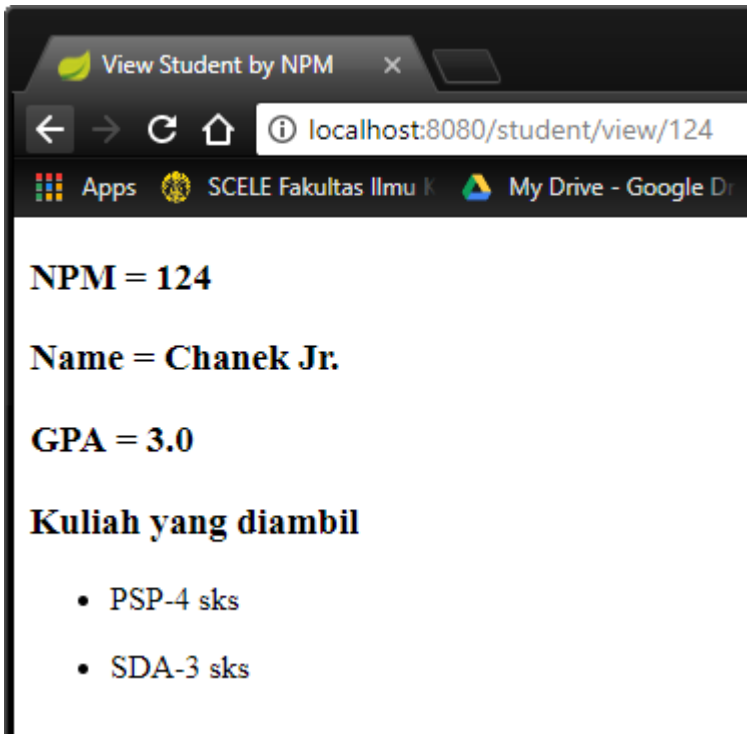
4. Menambahkan pada view.html

```
<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>
```

5. Percobaan menjalankan :
   - localhost:8080/student/view/123



- localhost:8080/student/view/124



LATIHAN 1

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

1. Mengubah StudentMapper

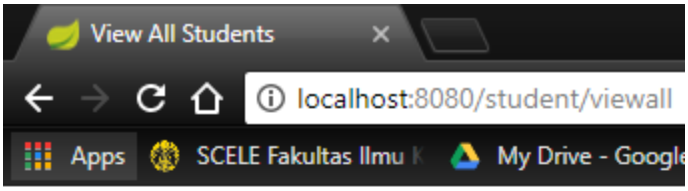
```
@Select("select npm, name, gpa from student")
@Results(value = {
            @Result(property="npm", column="npm"),
            @Result(property="name", column="name"),
            @Result(property="gpa", column="gpa"),
            @Result(property="courses", column="npm",
                      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

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

2. Mengubah viewall.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>
```

3. Percobaan :



All Students

No. 1

NPM = 123

Name = Chanek

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

Delete data
Update data

Update data

No. 2

NPM = 124

Name = Chanek Jr.

GPA = 3.0

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

Delete data
Update data

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

LATIHAN 2

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

1. Membuat interface CourseMapper

```
package com.example.dao;

import java.util.List;

import org.apache.ibatis.annotations.Many;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.annotations.Result;
import org.apache.ibatis.annotations.Results;
import org.apache.ibatis.annotations.Select;

import com.example.model.CourseModel;
import com.example.model.StudentModel;

@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);
}
```

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

2. Membuat interface CourseService

```
package com.example.service;

import com.example.model.CourseModel;

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

3. Membuat kelas CourseServiceDatabase

```
package com.example.service;

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

import com.example.dao.CourseMapper;
import com.example.model.CourseModel;

import lombok.extern.slf4j.Slf4j;

@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);
    }
}
```

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

4. Membuat viewcourse.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<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="'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 - Nama mahasiswa</li>
    </ul>
</body>
</html>
```

5. Membuat not-found-course.html

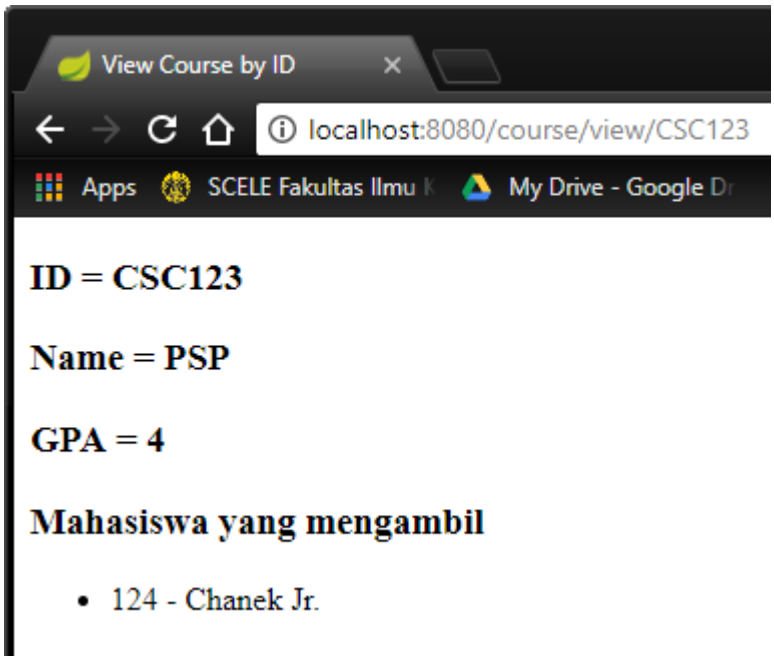
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Course not found</title>
    </head>
    <body>
          <h1>Course not found</h1>
          <h3 th:text="'ID = ' + ${id_course}">ID Course</h3>
    </body>
</html>
```

6. Membuat kelas CourseController

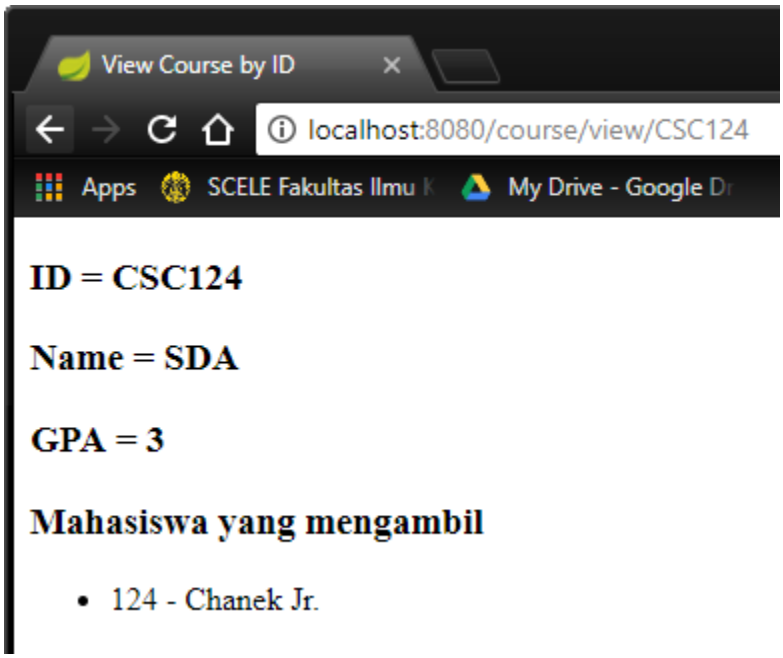
```
package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;

import com.example.model.CourseModel;
import com.example.service.CourseService;

@Controller
public class CourseController {
    @Autowired
    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 "viewcourse";
            } else {
                    model.addAttribute("id_course", id_course);
                    return "not-found-course";
            }
    }
}
```

Tengku Chavia Zagita
1506757642
APAP – A
Tutorial 5

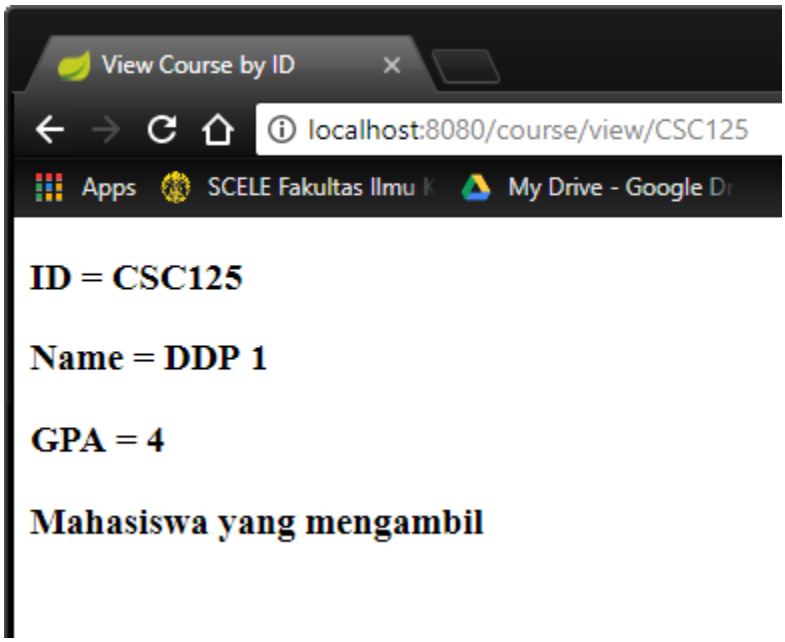
7. Percobaan
    a. CSC123



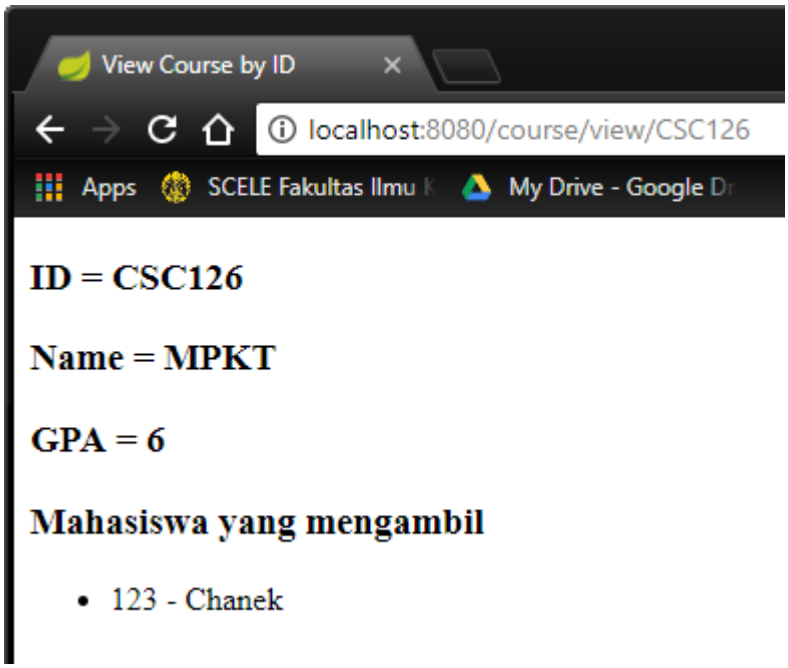
    b. CSC124



c. CSC125 (tidak ada mahasiswa yang mengambil)



d. CSC126



e. CSC120 (tidak ada di database)