Aisyah Husna
1506689686
ADPAP - C

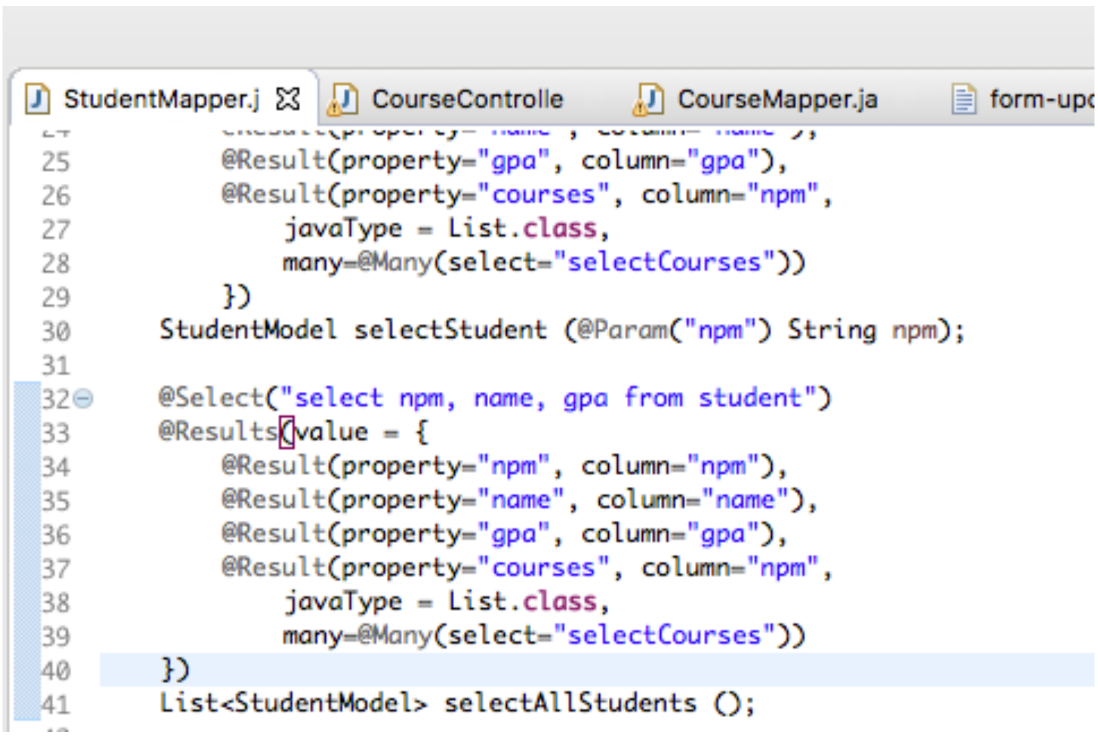
Tutorial 5

Menggunakan Database serta Relasi Database dalam Project Spring Boot

Latihan

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

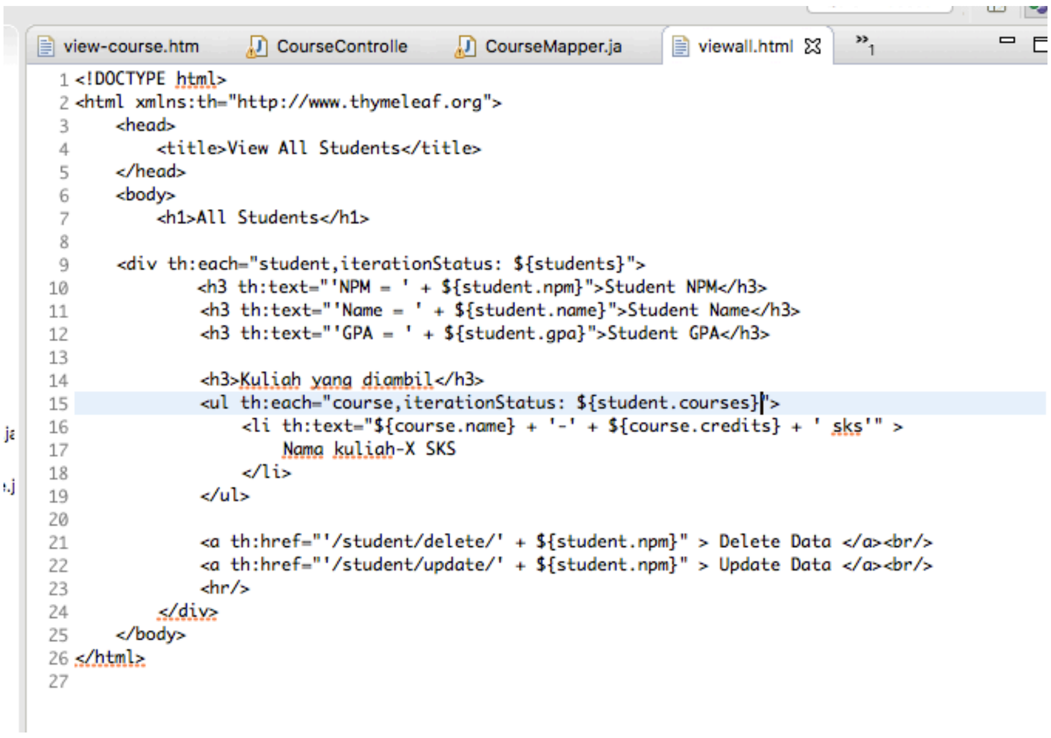
   Tambahkan anotasi *@Results* seperti yang dijelaskan pada tutorial pada method *selectStudent(String npm)* sehingga *courses* dapat terisikan untuk setiap *students* pada *List<studentModel>*. Berikut tambahan pada method:



```
        @Result(property="gpa", column="gpa"),
        @Result(property="courses", column="npm",
            javaType = List.class,
            many=@Many(select="selectCourses"))
    })
    StudentModel selectStudent (@Param("npm") String npm);

    @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 ();
```

   Ubah halaman *viewall.html* agar semua *courses* setiap *students* dapat diiterasikan:

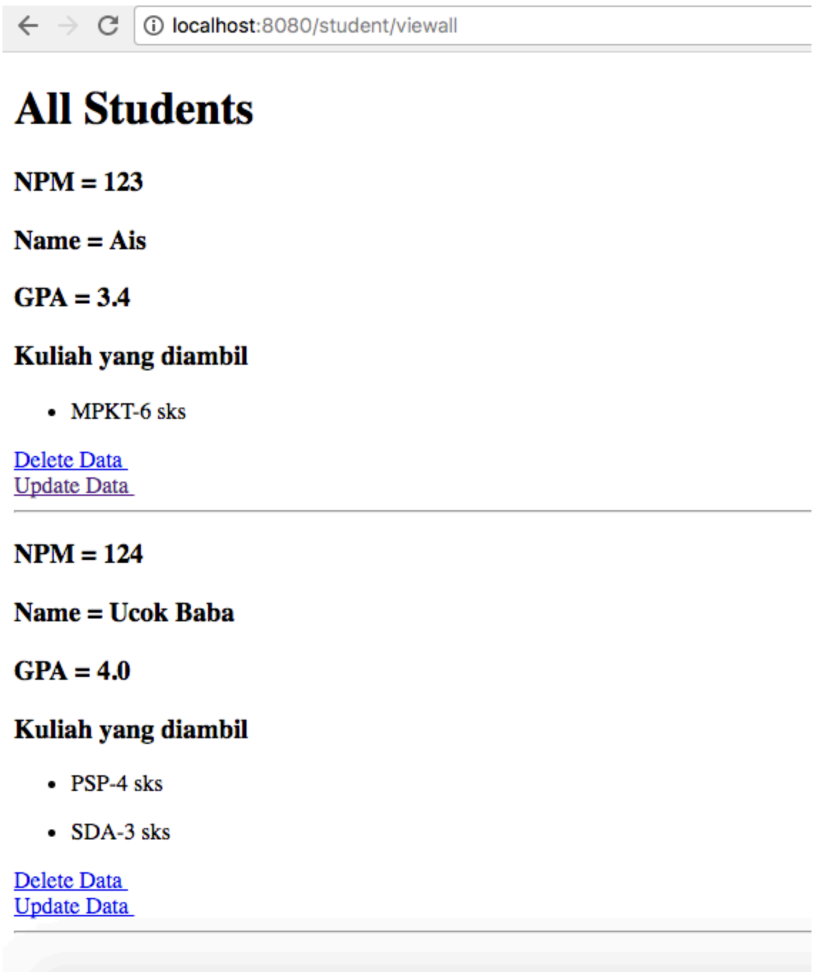
Aisyah Husna
1506689686
ADPAP - C



view-course.htm | CourseControlle | CourseMapper.ja | viewall.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View All Students</title>
    </head>
    <body>
        <h1>All Students</h1>

    <div th:each="student,iterationStatus: ${students}">
            <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>

            <a th:href="'/student/delete/' + ${student.npm}" > Delete Data </a><br/>
            <a th:href="'/student/update/' + ${student.npm}" > Update Data </a><br/>
            <hr/>
        </div>
    </body>
</html>
```

Contoh keluaran *viewall:*



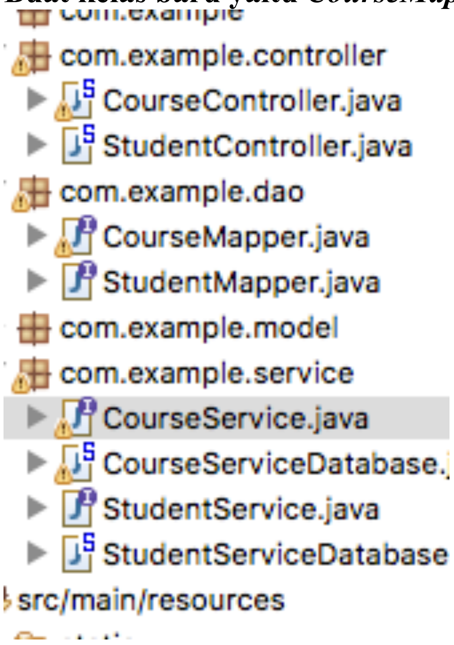
localhost:8080/student/viewall

```
All Students

NPM = 123

Name = Ais

GPA = 3.4

Kuliah yang diambil

  • MPKT-6 sks

Delete Data
Update Data
________________________________

NPM = 124

Name = Ucok Baba

GPA = 4.0

Kuliah yang diambil

  • PSP-4 sks

  • SDA-3 sks

Delete Data
Update Data
________________________________
```

Aisyah Husna
1506689686
ADPAP - C

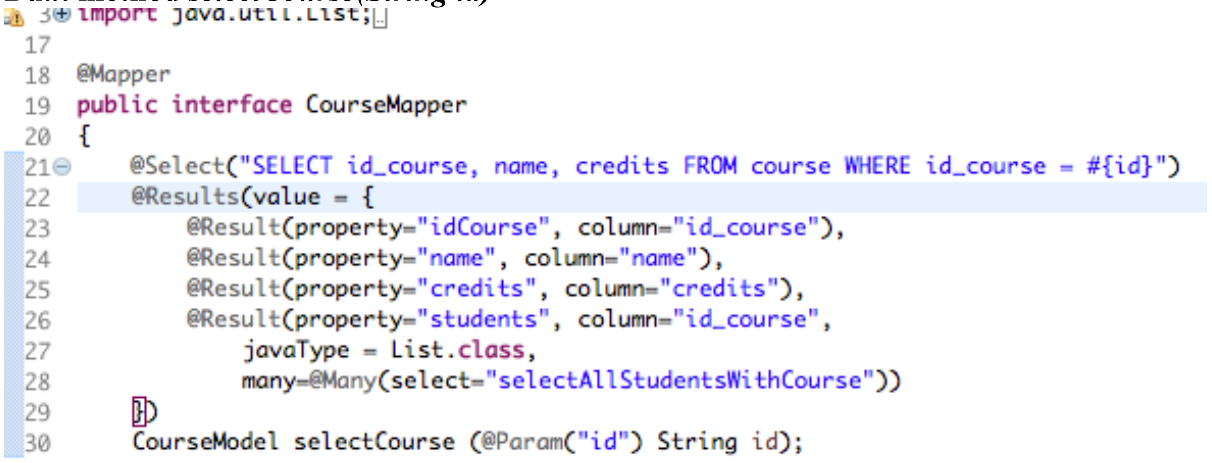
2. Buatlah view pada halaman http://localhost:8080/course/view/{id} untuk Course sehingga dapat menampilkan data course beserta Student yang mengambil

Buat kelas baru yaitu *CourseMapper, CourseService, CourseSerciveDatabase*.



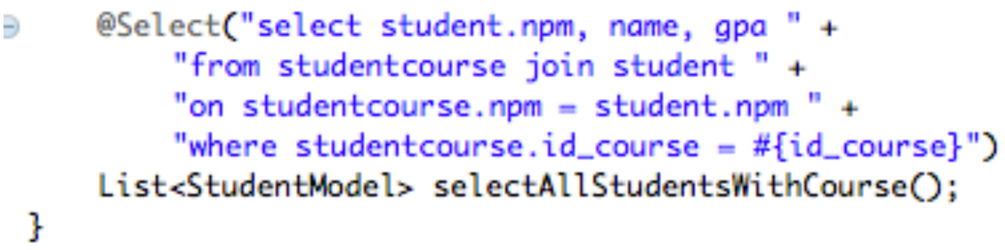
```
com.example.controller
  CourseController.java
  StudentController.java
com.example.dao
  CourseMapper.java
  StudentMapper.java
com.example.model
com.example.service
  CourseService.java
  CourseServiceDatabase.
  StudentService.java
  StudentServiceDatabase
src/main/resources
```

Buat method *selectCourse(String id)*



```
import java.util.List;

@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="selectAllStudentsWithCourse"))
    })
    CourseModel selectCourse (@Param("id") String id);
```

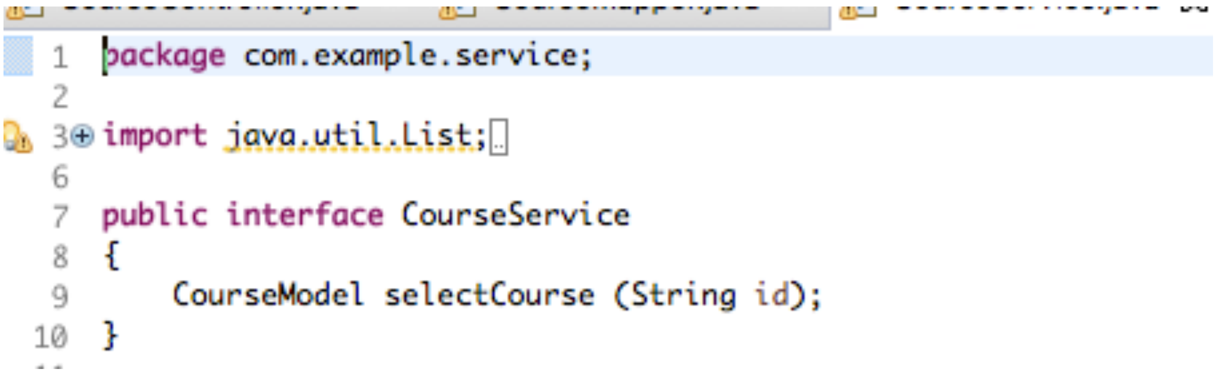
method untuk mengisikan *List<StudentModel> students* yang ada pada *course* yang kita *select.* Untuk mengisi bagian "Mahasiswa yang mengambil:" pada view course



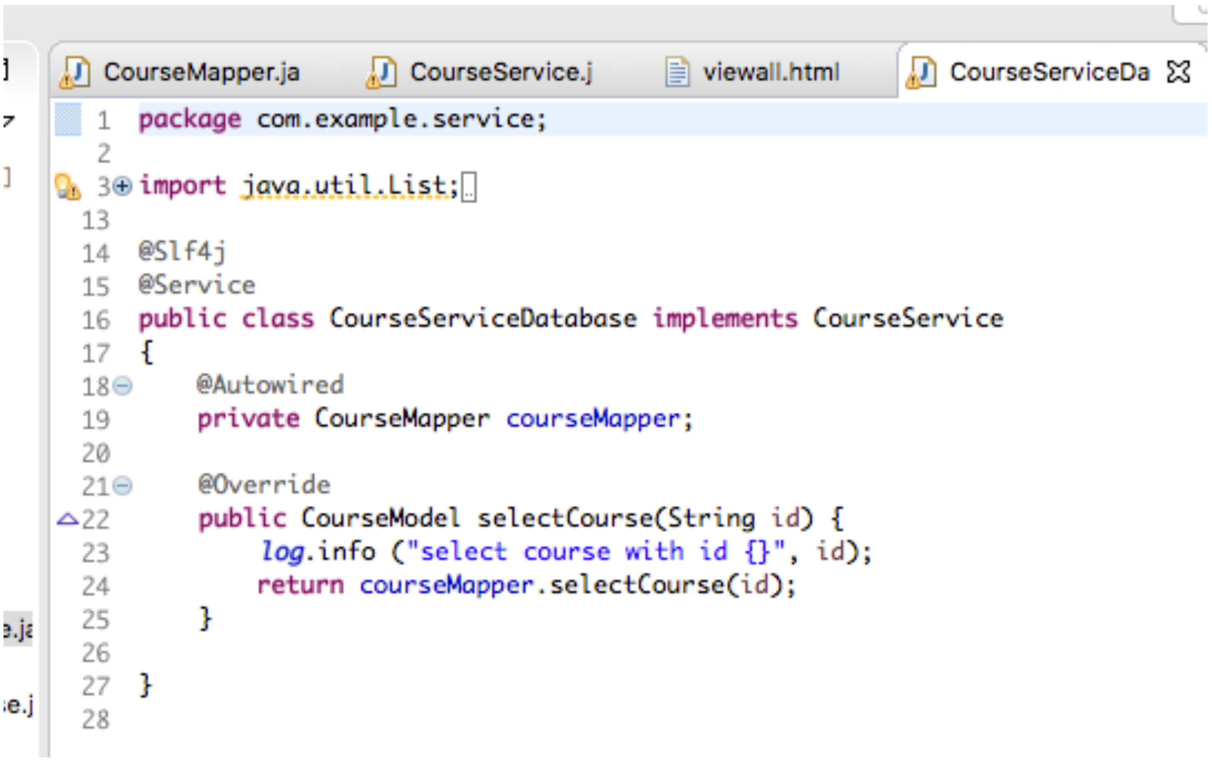
```
    @Select("select student.npm, name, gpa " +
        "from studentcourse join student " +
        "on studentcourse.npm = student.npm " +
        "where studentcourse.id_course = #{id_course}")
    List<StudentModel> selectAllStudentsWithCourse();
}
```

Aisyah Husna
1506689686
ADPAP - C

Implementasikan *interface* apa saja yang dibutuhkan pada *CourseService* dan *CourseServiceDatabase:*



```java
package com.example.service;

import java.util.List;

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

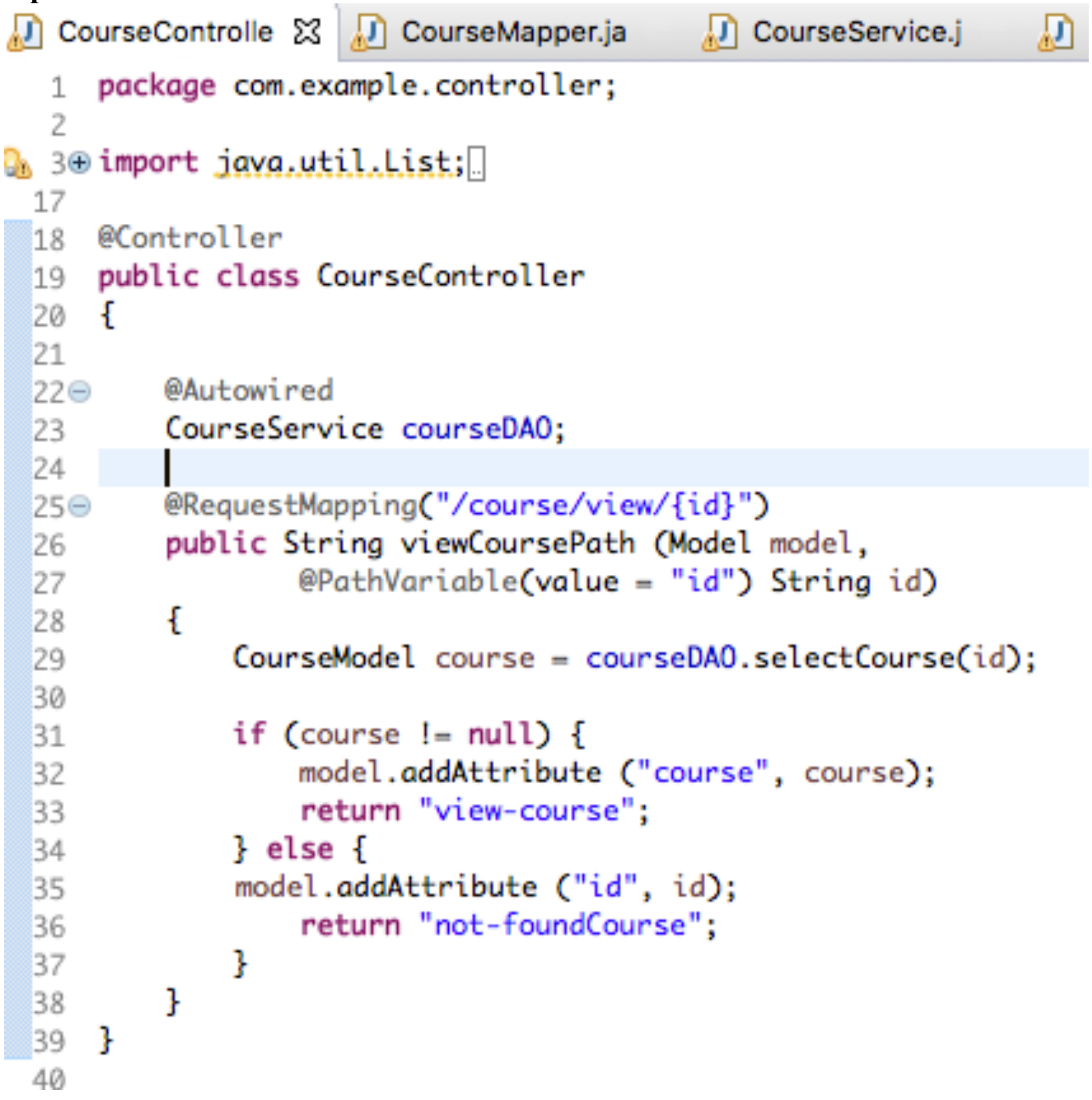


```java
package com.example.service;

import java.util.List;

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

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

}
```

Buat *controller* baru untuk *CourseService,* menginout *CourseService* dengan anotasi @Autowired. Kemudian buat method yang dibutuhkan untuk *view*

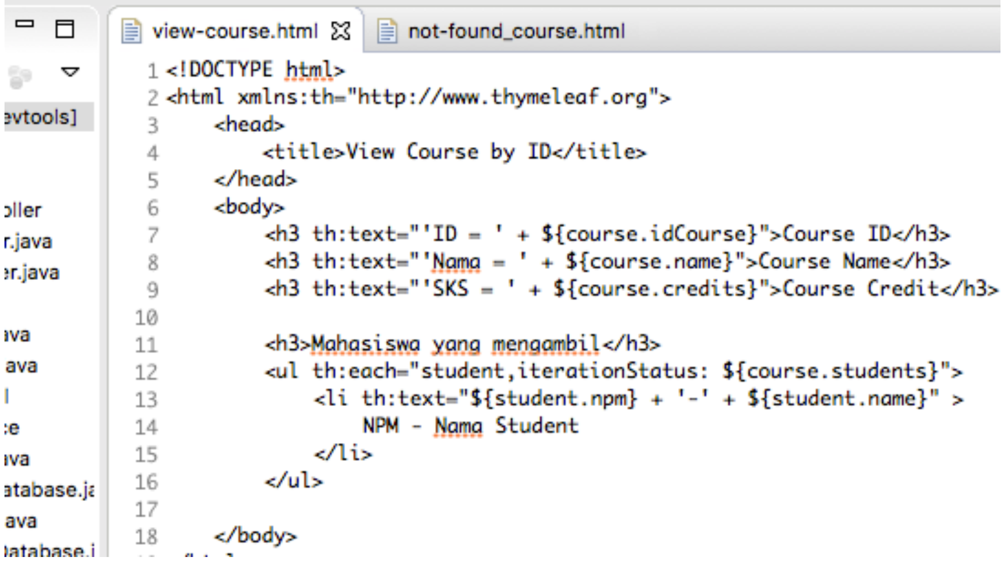
Aisyah Husna
1506689686
ADPAP - C

seperti berikut:



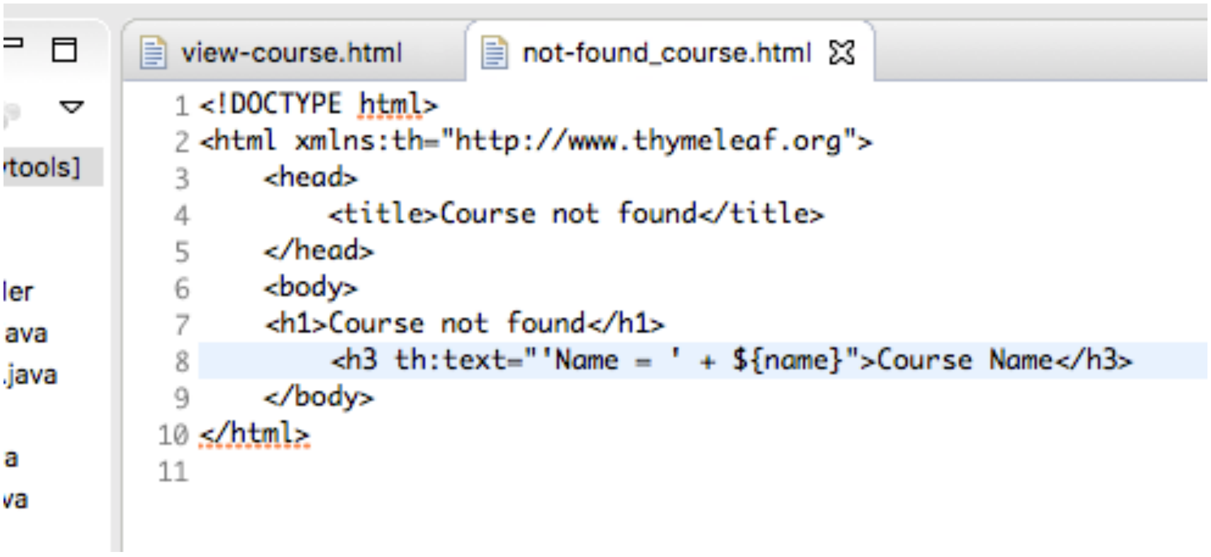
```java
package com.example.controller;

import java.util.List;

@Controller
public class CourseController
{

    @Autowired
    CourseService courseDAO;

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

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

Buat ***view-course.html*** **dan** ***not-foundCourse.html*** **untuk** ***ViewCoursePath:***



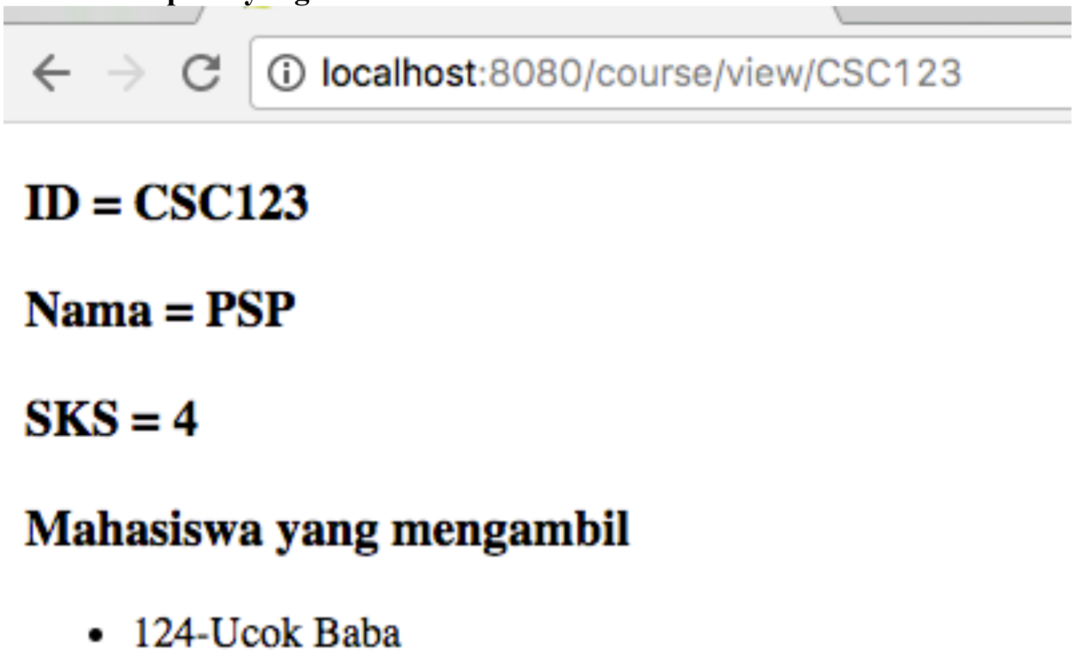
```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Course by ID</title>
    </head>
    <body>
        <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
        <h3 th:text="'Nama = ' + ${course.name}">Course Name</h3>
        <h3 th:text="'SKS =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

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

    </body>
```

Aisyah Husna
1506689686
ADPAP - C



```
<!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="'Name = ' + ${name}">Course Name</h3>
    </body>
</html>
```

Contoh tampilan yang muncul:



ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124-Ucok Baba

3. *Lesson Learned*
   Saya belajar bagaimana cara mengimplementasi relasi pada database pada *MVC Programming*, salah satunya yaitu menggunakan *join*. Saya juga belajar bagaimana cara memetakkan hasil *query* pada *class Model* yang ada pada MVC. Langkah diantaranya yaitu mengisi nama variable yang dituju pada *property*, dan mengisi hasil *query* dari database pada *column*.