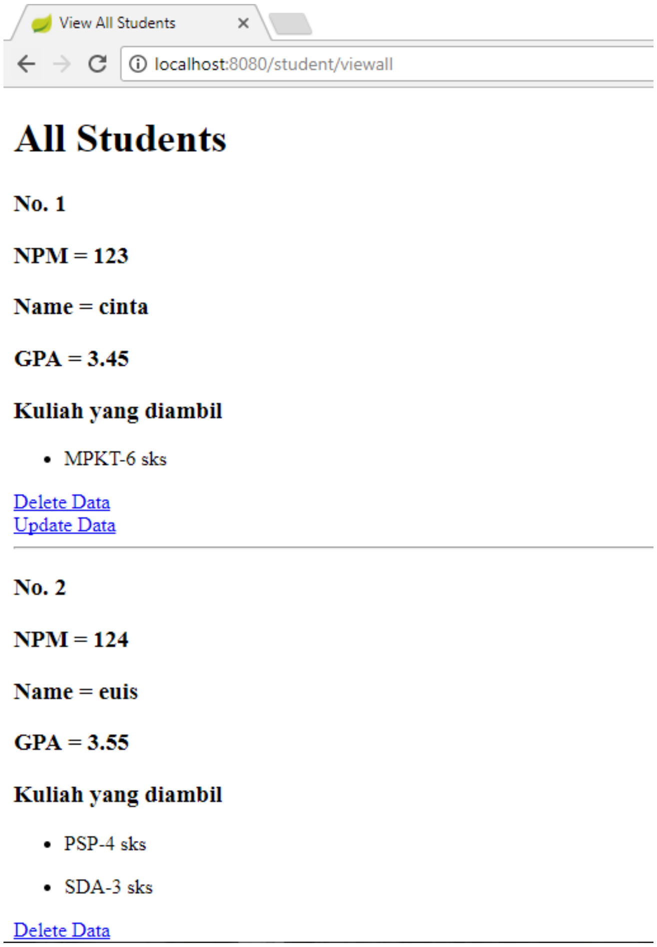
Atikah Luthfiana
1506689250
APAP B
Tutorial 05

## Penjelasan method latihan Select All Students

Hasil :



# All Students

**No. 1**

**NPM = 123**

**Name = cinta**

**GPA = 3.45**

**Kuliah yang diambil**

- MPKT-6 sks

Delete Data
Update Data

**No. 2**

**NPM = 124**

**Name = euis**

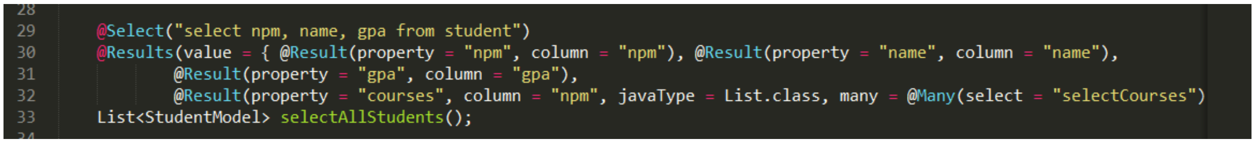
**GPA = 3.55**

**Kuliah yang diambil**

- PSP-4 sks
- SDA-3 sks

Delete Data

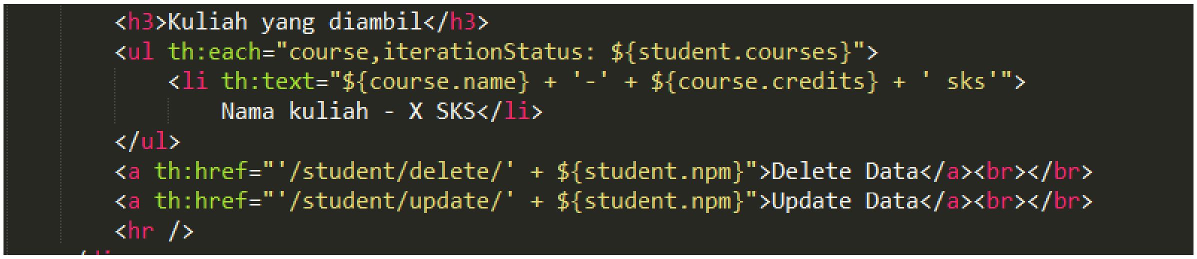
- Mengubah methos selectAllStudent yang ada di StudentMapper



```
    @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();
```

Sehingga masing-masing student yang di select di hubungkan dengan jenis mata kuliah yang di ambil

- Selanjutnya mengubah viewAll.html



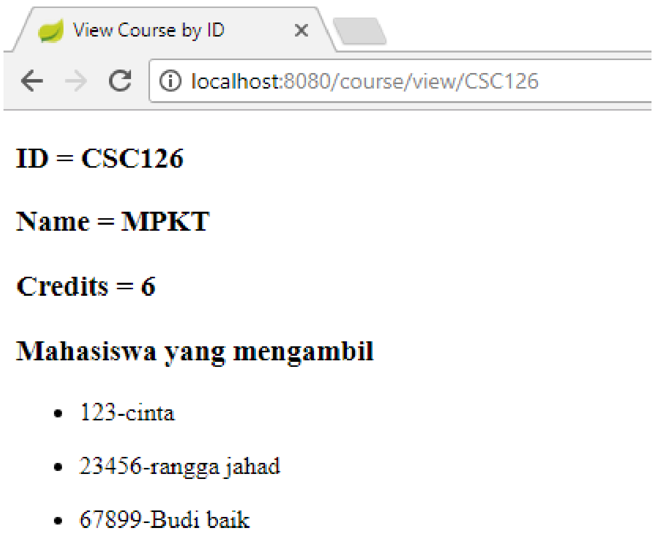
```
        <h3>Kuliah yang diambil</h3>
        <ul th:each="course,iterationStatus: ${student.courses}">
            <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
                Nama kuliah - X SKS</li>
        </ul>
        <a th:href="'/student/delete/' + ${student.npm}">Delete Data</a><br></br>
        <a th:href="'/student/update/' + ${student.npm}">Update Data</a><br></br>
        <hr />
```

Sehingga hasil dari query Results dan Result dapat dipetakan ke dalam html

Atikah Luthfiana
1506689250
APAP B
Tutorial 05

**Penjelasan untuk menambahkan View pada Course**

Hasil :



### ID = CSC126

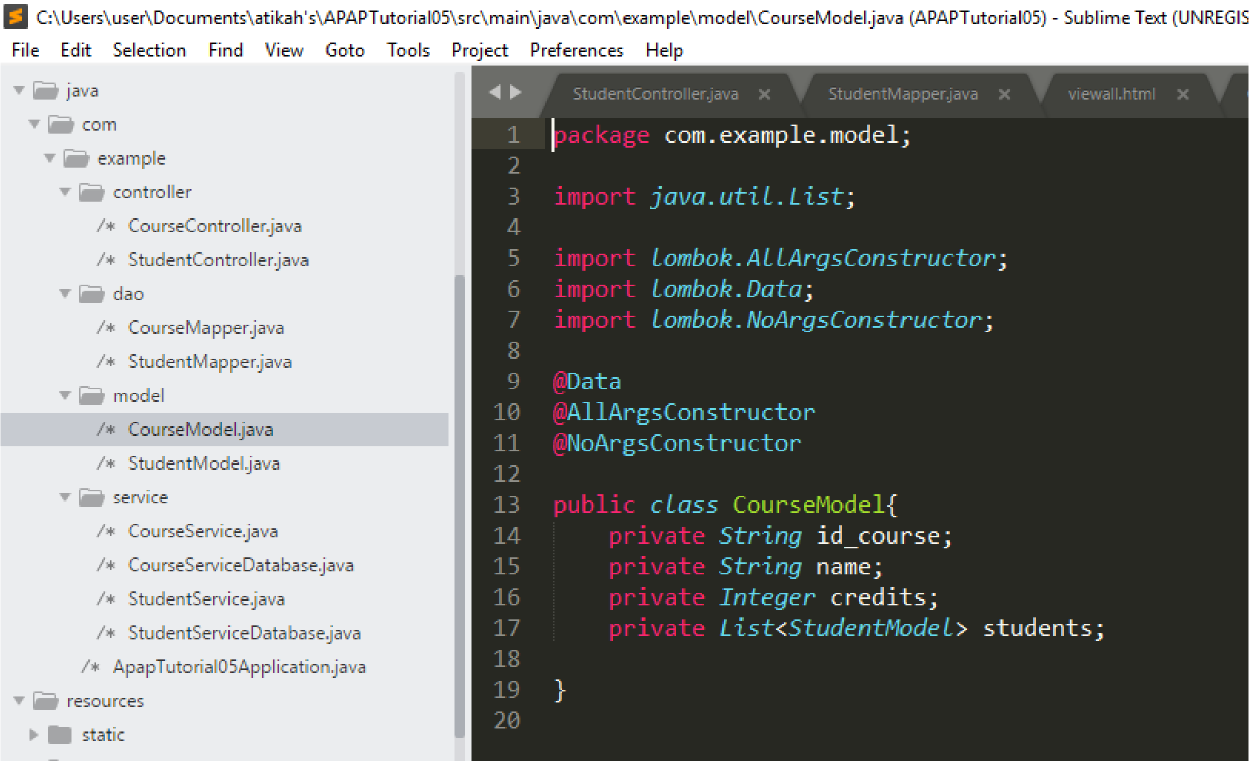
### Name = MPKT

### Credits = 6

### Mahasiswa yang mengambil

- 123-cinta
- 23456-rangga jahad
- 67899-Budi baik

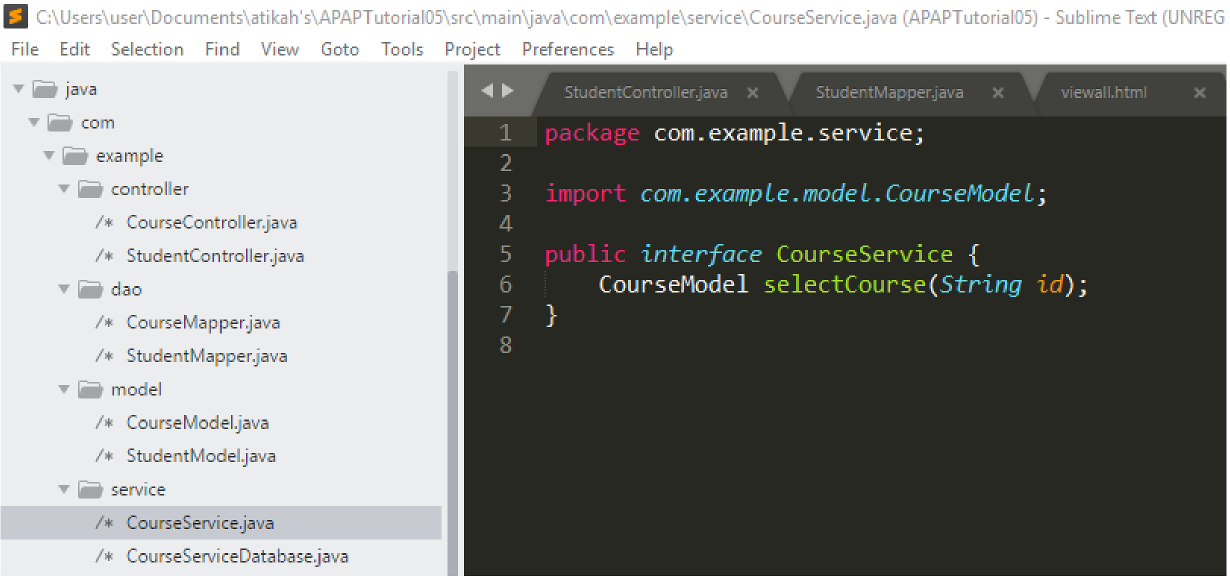
- Pertama diperlukan sebuah class yang memodelkan sebuah course dalam database dengan menggunakan CourseModel yang telah dibuat sebelumnya. Class inimenggambarkan object course yang berisi atribut-atribut yang ada pada tabel course di database yaitu id_course, name, dan credits serta List yang berisi npm mahasiswa yang mengambil course tersebut



```java
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 id_course;
    private String name;
    private Integer credits;
    private List<StudentModel> students;

}
```

Atikah Luthfiana
1506689250
APAP B
Tutorial 05

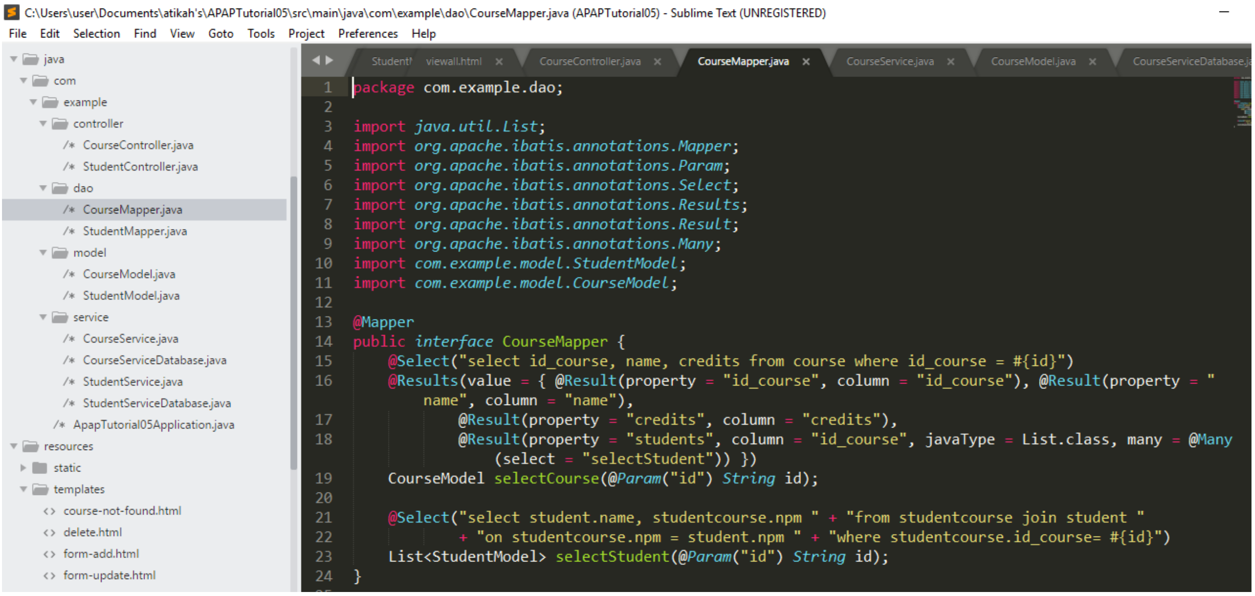
- Menambahkan membuat interface CourceService


C:\Users\user\Documents\atikah's\APAPTutorial05\src\main\java\com\example\service\CourseService.java (APAPTutorial05) - Sublime Text (UNREG
File Edit Selection Find View Goto Tools Project Preferences Help
java
com
example
controller
CourseController.java
StudentController.java
dao
CourseMapper.java
StudentMapper.java
model
CourseModel.java
StudentModel.java
service
CourseService.java
CourseServiceDatabase.java
StudentController.java
StudentMapper.java
viewall.html

```java
package com.example.service;

import com.example.model.CourseModel;

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

```

Di dalam interface StudentService dibuat abstract method untuk melakukan select course berdasarkan id course

- Membuat courseMapper yang berfungsi untuk memetakan hasil select database

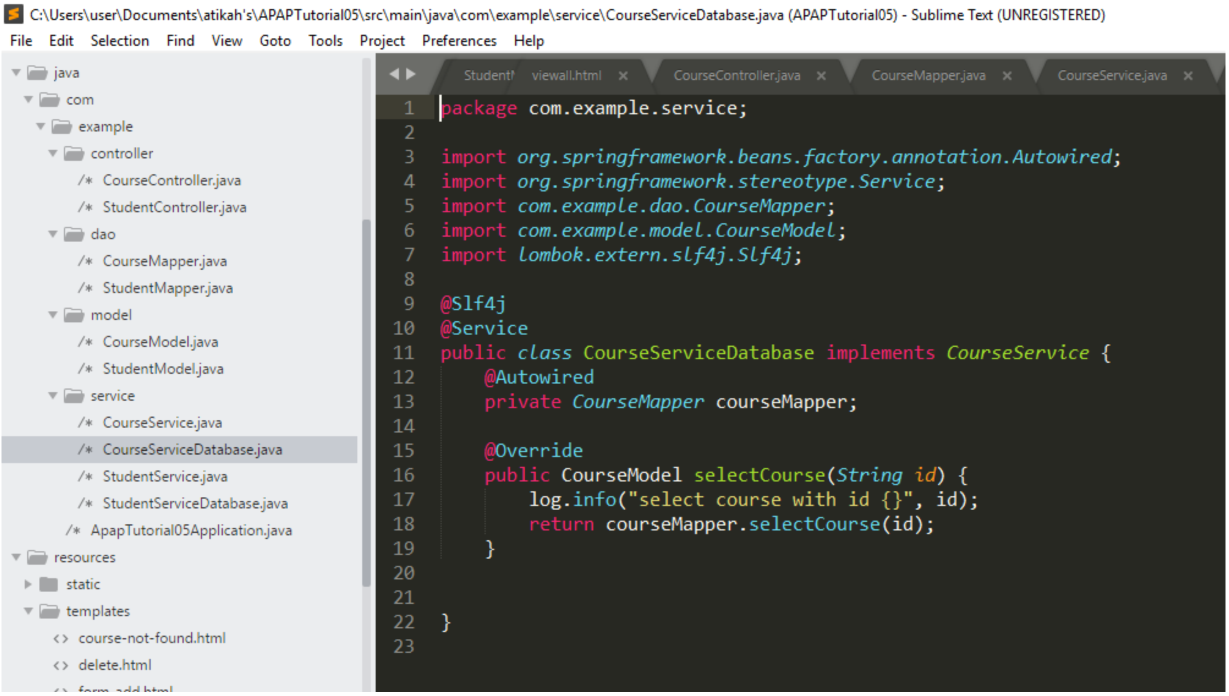


```java
package com.example.dao;

import java.util.List;
import org.apache.ibatis.annotations.Mapper;
import org.apache.ibatis.annotations.Param;
import org.apache.ibatis.annotations.Select;
import org.apache.ibatis.annotations.Results;
import org.apache.ibatis.annotations.Result;
import org.apache.ibatis.annotations.Many;
import com.example.model.StudentModel;
import com.example.model.CourseModel;

@Mapper
public interface CourseMapper {
    @Select("select id_course, name, credits from course where id_course = #{id}")
    @Results(value = { @Result(property = "id_course", 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.name, studentcourse.npm " + "from studentcourse join student "
            + "on studentcourse.npm = student.npm " + "where studentcourse.id_course= #{id}")
    List<StudentModel> selectStudent(@Param("id") String id);
}
```

Pada studentMapper kita membutuhkan query selectCourse untuk mendapatkan suatu course dengan id_course tertentu. Pada selectStudent, query yang dihasilkan merupakan nama-nama mahasiswa yang mengambil mata kuliah yang telah di-select pada method selectCourse

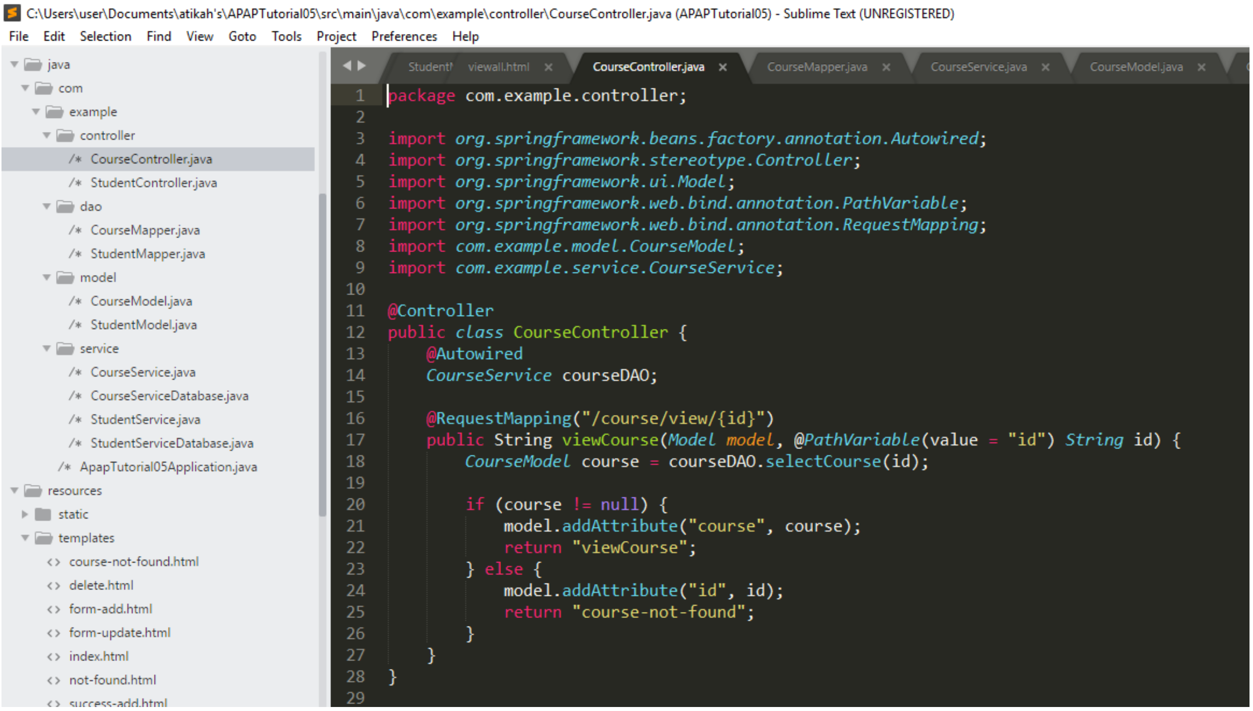
- Selanjutnya dibuat StudentServiceDatabase

Atikah Luthfiana
1506689250
APAP B
Tutorial 05



```java
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) {
        log.info("select course with id {}", id);
        return courseMapper.selectCourse(id);
    }


}

```

- Selanjutnya membuat RequestMapping dari url html ke dalam CourseController

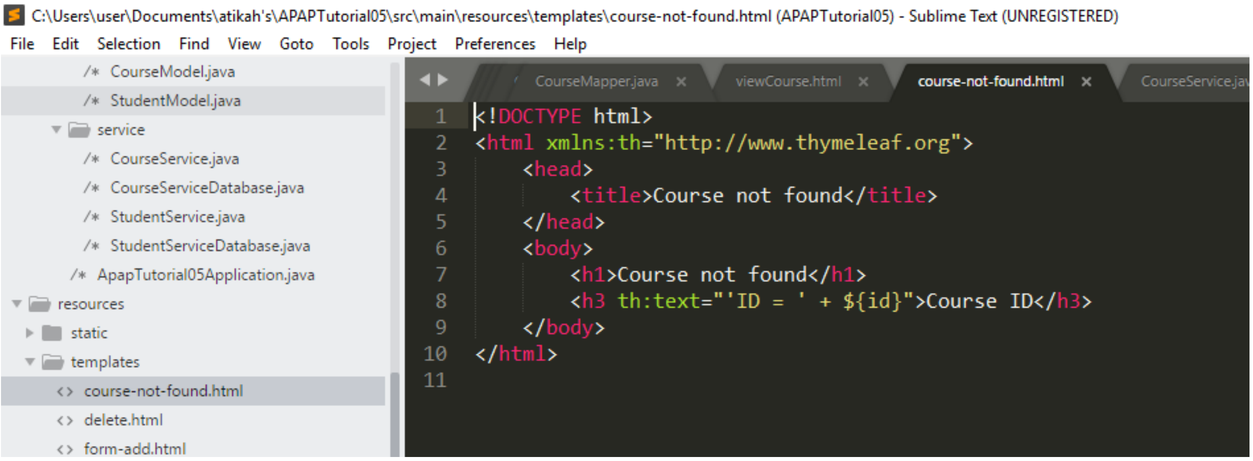


```java
package com.example.controller;

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 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 "course-not-found";
        }
    }
}

```

Method viewCourse akan mencari course berdasarkan id course yang ada di url ke dalam database dengan bantuan courseMapper. Jika course ditemukan, maka hasil request mapping akan di transfer ke viewCourse.html. Namun, jika course tidak ditemukan akan di arahkan ke course-not-found yang berisi informasi bahwa course yang dicari tidak ditemukan.

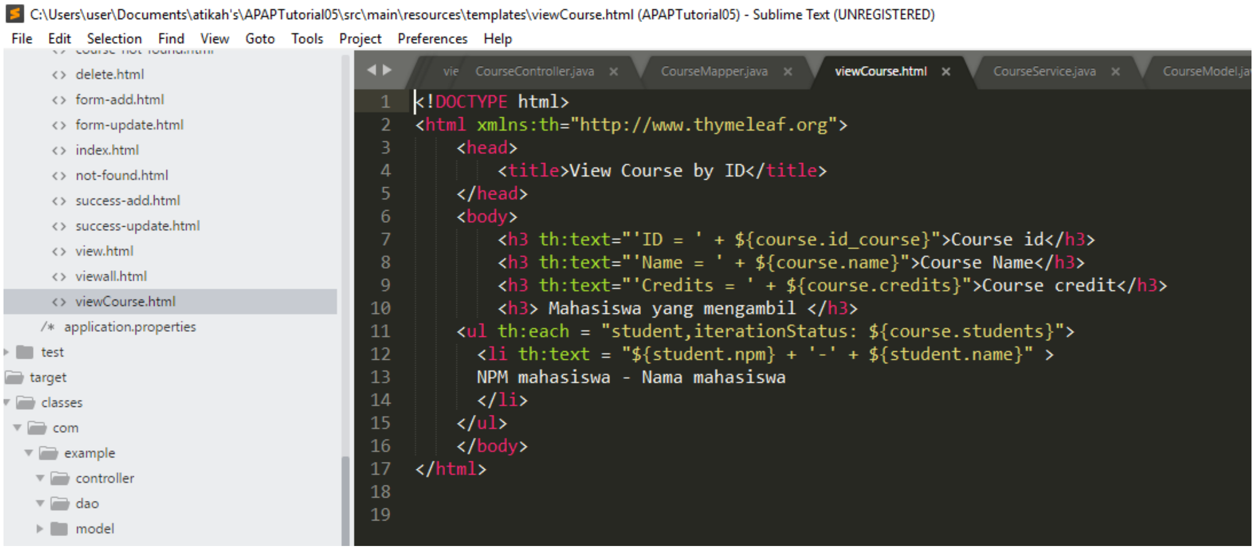
Atikah Luthfiana
1506689250
APAP B
Tutorial 05

- Membuat course-not-found.html untuk memberikan informasi kepada user bahwa course yang dicari dari query ke database tidak ditemukan



```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</h3>
    </body>
</html>
```

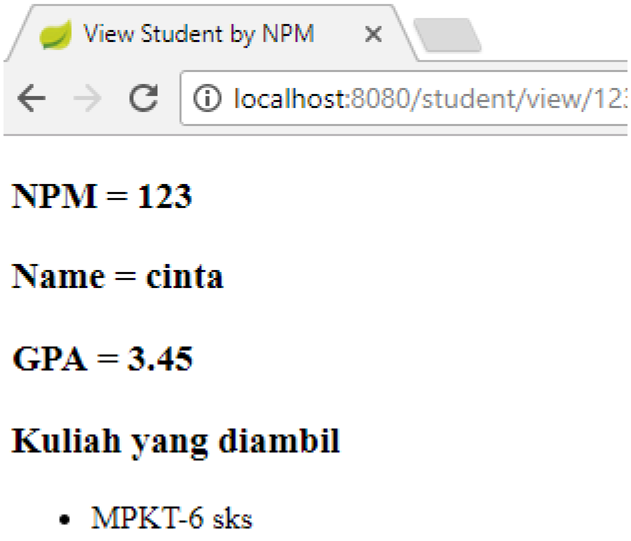
- Terakhir, membuat viewCourse.html untuk memetakan hasil query yang didapatkan untuk ditampilkan ke user



```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Course by ID</title>
    </head>
    <body>
        <h3 th:text="'ID = ' + ${course.id_course}">Course id</h3>
        <h3 th:text="'Name = ' + ${course.name}">Course Name</h3>
        <h3 th:text="'Credits = ' + ${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 mahasiswa - Nama mahasiswa
      </li>
    </ul>
    </body>
</html>
```

Atikah Luthfiana
1506689250
APAP B
Tutorial 05

**Hal yang Dipelajari dari Tutorial 05**



NPM = 123

Name = cinta

GPA = 3.45

Kuliah yang diambil

- MPKT-6 sks

Pada tutorial kali ini saya belajar mengenai anotasi Results dan Result. Anotasi @Result digunakan untuk memetakan hasil query select ke class model dari hasi query yang dilakukan oleh anotasi @Results. Sehingga kita dapat memetakan hasil query dari suatu method ke dalam suatu list tertentu.