

- CourseModel

```

1 package com.example.model;
2
3+ import java.util.List;
4
5
6
7
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 }

```

- StudentModel

```

1 package com.example.model;
2
3+ import java.util.List;
4
5
6
7
8
9 @Data
10 @AllArgsConstructor
11 @NoArgsConstructor
12 public class StudentModel
13 {
14     private String npm;
15     private String name;
16     private double gpa;
17     private List<CourseModel> courses;
18 }

```

- StudentController

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

- StudentMapper

- selectCourse

```

@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);

```

- selectStudent

```

@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);

```

- selectAllStudents

```
@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 ();
```

- View.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Student by NPM</title>
5 </head>
6 <body>
7 <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
8 <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
9 <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
10
11 <h3>Kuliah yang diambil</h3>
12 <ul th:each="course, iterationStatus: ${student.courses}">
13 <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
14 Nama kuliah-X SKS
15 </li>
16 </ul>
17 </body>
18 </html>
```

- View-all.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Students</title>
5 </head>
6 <body>
7 <h1>All Students</h1>
8
9 <div th:each="student, iterationStatus: ${students}">
10 <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
11 <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
12 <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
13 <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
14 <h3>Kuliah yang diambil</h3>
15 <ul th:each="course, iterationStatus: ${student.courses}">
16 <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
17 Nama kuliah-X SKS
18 </li>
19 </ul>
20 <a th:href="'/student/delete/' + ${student.npm}"> Delete Data </a><br/>
21 <a th:href="'/student/update/' + ${student.npm}"> Update Data </a><br/>
22 <hr/>
23 </div>
24 </body>
25 </html>
```

- CourseController

```

1  package com.example.controller;
2
3  import org.springframework.beans.factory.annotation.Autowired;
11
12  @Controller
13  public class CourseController {
14
15      @Autowired
16      CourseService courseDAO;
17
18      @RequestMapping("/course/view/{id_course}")
19      public String viewCourse (Model model,
20                              @PathVariable(value = "id_course") String id_course)
21      {
22          CourseModel course = courseDAO.selectCourse (id_course);
23
24          if (course != null) {
25              model.addAttribute ("course", course);
26              return "view-course";
27          } else {
28              model.addAttribute ("id_course", id_course);
29              return "not-found";
30          }
31      }
32  }
33

```

- CourseMapper

```

1  package com.example.dao;
2
3  import java.util.List;
14
15  @Mapper
16  public interface CourseMapper {
17
18      @Select("select student.npm, name, gpa " +
19              "from studentcourse join student " +
20              "on studentcourse.npm = student.npm "
21              + "where studentcourse.id_course = #{id_course}")
22      List<StudentModel> selectCourseStudent (@Param("id_course") String id_course);
23
24      @Select("select id_course, name, credits " +
25              "from course " +
26              "where id_course = #{id_course}")
27      @Results(value = {
28          @Result(property="idCourse", column="id_course"),
29          @Result(property="name", column="name"),
30          @Result(property="credits", column="credits"),
31          @Result(property="students", column="id_course",
32                  javaType = List.class,
33                  many=@Many(select="selectCourseStudent"))
34      }) CourseModel selectCourse (@Param("id_course") String id_course);
35  }
36

```



- CourseService

```

1 package com.example.service;
2
3 import com.example.model.CourseModel;
4
5 public interface CourseService {
6     CourseModel selectCourse(String id_course);
7 }
8

```

- CourseServiceDatabase

```

1 package com.example.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Slf4j
6 @Service
7 public class CourseServiceDatabase implements CourseService {
8
9     @Autowired
10     private CourseMapper courseMapper;
11
12     @Override
13     public CourseModel selectCourse(String id_course) {
14         log.info("select course with id course {}", id_course);
15         return courseMapper.selectCourse(id_course);
16     }
17 }
18

```

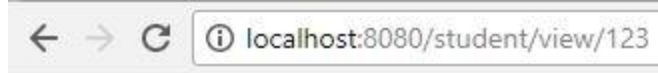
- View-course.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <head>
4         <title>View Course</title>
5     </head>
6     <body>
7         <h3 th:text="'ID = ' + ${course.idCourse}">Course ID</h3>
8         <h3 th:text="'Nama = ' + ${course.name}">Course Name</h3>
9         <h3 th:text="'SKS = ' + ${course.credits}">Course Credits</h3>
10
11         <h3>Mahasiswa yang mengambil</h3>
12         <ul th:each="student, iterationStatus: ${course.students}">
13             <li th:text="${student.npm} + '-' + ${student.name}">
14                 Npm mahasiswa-X
15             </li>
16         </ul>
17     </body>
18 </html>

```

- <http://localhost:8080/student/view/123>



**NPM = 123**

**Name = Namnung**

**GPA = 4.0**

**Kuliah yang diambil**

- MPKT-6 sks

- <http://localhost:8080/student/view/124>



**NPM = 124**

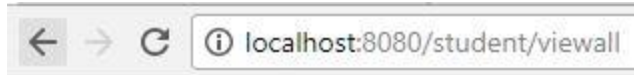
**Name = Budi**

**GPA = 3.5**

**Kuliah yang diambil**

- PSP-4 sks
- SDA-3 sks

○ http://localhost:8080/student/viewall



## All Students

No. 1

NPM = 123

Name = Namnung

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

---

No. 2

NPM = 124

Name = Budi

GPA = 3.5

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

[Delete Data](#)

[Update Data](#)

---

---

No. 3

NPM = 125

Name = Ari

GPA = 3.0

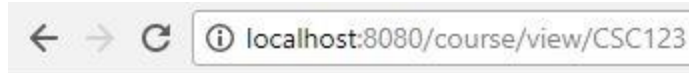
Kuliah yang diambil

[Delete Data](#)

[Update Data](#)

---

- <http://localhost:8080/course/view/CSC123>



**ID = CSC123**

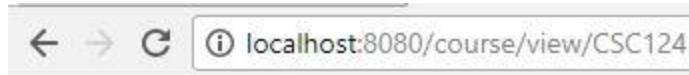
**Nama = PSP**

**SKS = 4**

**Mahasiswa yang mengambil**

- 124-Budi

- <http://localhost:8080/course/view/CSC124>



**ID = CSC124**

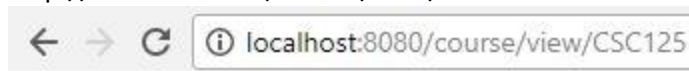
**Nama = SDA**

**SKS = 3**

**Mahasiswa yang mengambil**

- 124-Budi

- <http://localhost:8080/course/view/CSC125>



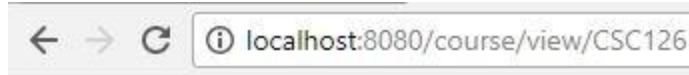
**ID = CSC125**

**Nama = DDP 1**

**SKS = 4**

**Mahasiswa yang mengambil**

- <http://localhost:8080/course/view/CSC126>



**ID = CSC126**

**Nama = MPKT**

**SKS = 6**

**Mahasiswa yang mengambil**

- 123-Namnung

1. Pada latihan pertama diminta agar pada halaman viewall menampilkan semua student beserta daftar kuliah yang diambil. Maka pada kelas StudentMapper ditambahkan kode pada method selectAllStudents seperti terlampir di atas, sementara pada templates viewall.html juga ditambahkan kode seperti terlampir di atas. Sehingga terdapat sebuah list pada method selectAllStudents yang nantinya akan diiterasi pada viewall.html
2. Pada latihan kedua diminta agar terdapat halaman view course berdasarkan id course. Maka perlu ditambahkan CourseController, CourseMapper, CourseService, CourseServiceDatabase, CourseModel. Satu paket ini dibutuhkan untuk membuat “/course/” agar terdapat controller, mapper, service, service database, dan model yang berbeda antara course dan student sehingga tidak terdapat ambiguitas pada program. Kode-kode dari masing-masing kelas terlampir di atas yang isinya kurang lebih sama seperti student.