

1. Jelaskan apa saja hal yang Anda pelajari dari tutorial ini.
2. Mengubah method `selectAllStudents` pada kelas `StudentMapper` agar halaman `viewall` menampilkan semua student beserta daftar kuliah yang diambil.

Berikut method `selectAllStudents` pada class `studentMapper`

```
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 ();
```

Berikut file `viewall.html` yang juga dirubah agar dapat menampilkan daftar kuliah yang diambil mahasiswa dengan mengiterasi daftar `course`.

```
<body>
<h1>All Students</h1>

<div th:each="student, iterationStatus: ${students}">
    <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
    <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>
    <a th:href="'/student/delete/' + ${student.npm}" > Delete Data</a><br/>
    <a th:href="'/student/update/' + ${student.npm}" > Update Data</a><br/>

    <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>
    <hr/>
</div>
</body>
```

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

Pertama-tama saya membuat `CourseMapper` untuk mendefinisikan method yang akan digunakan. Kemudian mengimplementasikan method yang dibuat sebelumnya di class `StudentServiceDatabase` dan membuat `CourseMapper` yang berisi query untuk mengambil data dari tabel `course` yang terdapat di database.

```
public interface CourseMapper {

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

    @Select ("select studentcourse.id_course, student.name, student.npm, course.credits " +
        "from studentcourse, student, course " +
        "where studentcourse.id_course=#{id_course} " +
        "AND student.npm=studentcourse.npm AND course.id_course=studentcourse.id_course")
    List <StudentModel> selectCourseById (@Param("id_course") String id_course );
}

public class CourseService {

    @Autowired
    private CourseMapper courseMapper;

    public CourseModel selectCourse (String id_course) {
        log.info ("select course by ID {}", id_course);
        return courseMapper.selectCourse (id_course);
    }
}

public class CourseModel {
    private String id_course;
    private String name;
    private Integer credits;
    private List <StudentModel> students;
}
```

Setelah itu, saya membuat controller baru yaitu CourseController untuk mencari data dalam database lalu kemudian ditampilkan data tersebut. Jika data tersebut ditemukan akan merefer file view-course.html.

```
@RequestMapping("/course/view/{id_course}")
public String viewCoursePath (Model model,
    @PathVariable(value = "id_course") 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";
    }
}
```

Berikut isi file view-course.html

Dayana Sheilla Agathy  
1406572630  
Write-up Tutorial 5

```
<!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 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>
```