Write-Up Tutorial 5

Arsitektur dan Perancangan Aplikasi Perusahaan

Nama : Lintang Matahari Hasani
NPM : 1506689231

Latihan

Soal 1. Membuat *method* selectAllStudents

Pada latihan ini, dibuat sebuah method selectAllStudents pada StudentController yang mengembalikan sebuah view berisi informasi mengenai semua *student* beserta mata kuliah yang diambilnya.

Langkah pengerjaan :

1) ***Memastikan Model Class untuk Course sudah dibuat dan Class-Class terkait***

   Class-class yang terkait dengan fitur ini adalah Model Class Student, Class StudentServiceDatabase, Interface StudentMapper, Interface StudentService telah dimodifikasi dengan menambahkan method yang diperlukan untuk memproses data course/mata kuliah.

   Catatan : Langkah ini telah dicontohkan dan dilakukan pada tutorial

   Tampilan Model Class untuk Course (Class CourseModel)

```
StudentCont...  StudentMapp...  view-course....  StudentServ...  viewall.html  StudentServ...  CourseModel....
 1 package com.example.model;
 2
 3*import java.util.List;
 8
 9 @Data
10 @AllArgsConstructor
11 @NoArgsConstructor
12 public class CourseModel
13 {
14     private String id_course;
15     private String name;
16     private Integer credits;
17     private List<StudentModel> students;
18 }
```

   Tampilan method selectStudent yang telah dimodifikasi pada Interface StudentMapper

```
21     //Method untuk memilih seorang mahasiswa (beserta informasi mata kuliah yang diambilnya) berdasarkan npm
22     @Select("SELECT npm, name, gpa FROM student WHERE npm = #{npm}")
23     @Results(value = {
24         @Result(property="npm", column="npm"),
25         @Result(property="name", column="name"),
26         @Result(property="gpa", column="gpa"),
27         @Result(property="courses", column="npm",
28         javaType = List.class,
29         many=@Many(select="selectCourses"))
30     })
31     StudentModel selectStudent (@Param("npm") String npm);
32
```

Nama : Lintang Matahari Hasani
NPM : 1506689231

Tampilan method selectCourses pada interface StudentMapper

```
@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);
```

Method view yang mengembalikan semua student tidak perlu dimodifikasi karena data terkait course telah terdapat pada object student dan disimpan dalam List courses dan dapat diakses pada view.

2) *Memodifikasi method selectAllStudents pada Interface StudentMapper*

```
//Method untuk memilih semua mahasiswa beserta informasi mata kuliah yang diambil
@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 ();
```

Method ini mengembalikan List berisi object Student yang mencakup keseluruhan data student termasuk mata kuliah yang diambil. Pada method ini terdapat anotasi many=@Many(select = “selectCourses”) yang akan memanggil method selectCourses untuk memperoleh data course yang diambil student tersebut.

Method ini akan mengambil data terkait student dari database dan kemudian akan dimasukkan ke dalam object Student ketika method ini diimplementasikan.

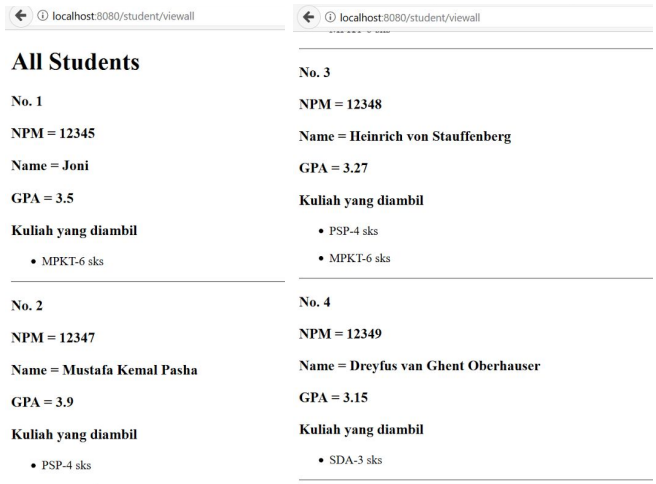
3) *Menambahkan anotasi yang berfungsi menampilkan semua courses yang diambil suatu student pada view viewall.html*

```
<!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="'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>

            <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>
```

Anotasi tersebut diletakkan di dalam iterasi object Students.

Nama : Lintang Matahari Hasani
NPM : 1506689231

Berikut merupakan tampilan fitur menampilkan mata kuliah yang diambil setiap student pada view semua student:



Nama : Lintang Matahari Hasani
NPM : 1506689231

Soal 2. Membuat view untuk Course yang menampilkan data Course beserta mahasiswa yang mengambil

Pada lathihan ini, dibuat sebuah fitur untuk menampilkan data course beserta mahasiswa yang mengambil. Parameter yang digunakan adalah id course.

Langkah Pengerjaan :

1) ***Membuat method selectCourse pada interface StudentService***

```
19
20      CourseModel selectCourse (String idCourse);
21
```

Method ini menerima parameter String dan mengembalikan Object CourseModel.

2) ***Mengimplementasikan method selectCourse pada Class StudentServiceDatabase***

```
58      @Override
59      public CourseModel selectCourse (String idCourse)
60      {
61          log.info ("select course with ID {}", idCourse);
62          return studentMapper.selectCourse (idCourse);
63      }
```

Method ini menerima parameter String idCourse yang ingin dipilih dan mengembalikan Object CourseModel dengan memanggil method selectCourse pada Class StudentMapper.

3) ***Membuat method selectCourse pada interface StudentMapper yang menerima parameter idCourse***

```
72      //Method untuk memilih suatu mata kuliah berdasarkan id mata kuliah
73      @Select("SELECT id_course, name, credits FROM course WHERE id_course = #{id_course}")
74      @Results(value = {
75          @Result(property="id_course", column="id_course"),
76          @Result(property="name", column="name"),
77          @Result(property="credits", column="credits"),
78          @Result(property="students", column="id_course",
79          javaType = List.class,
80          many=@Many(select="selectStudents"))
81      })
82      CourseModel selectCourse (@Param("id_course") String id_course);
83
84
```

Method selectCourse menerima parameter idCourse dari course yang ingin dipilih. Method mengembalikan object CourseModel yang memiliki properti sesuai course yang ingin dipilih berdasarkan idCourse.

Nama : Lintang Matahari Hasani
NPM : 1506689231

4) ***Membuat method selectStudents pada interface StudentMapper yang menerima parameter idCourse***

```
//Method untuk memilih semua mahasiswa yang mengambil suatu mata kuliah
@Select("SELECT student.npm, student.name " +
        "FROM studentcourse JOIN student " +
        "ON studentcourse.npm = student.npm " +
        "WHERE studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudents (@Param("id_course") String id_course);
```

Method ini menerima parameter idCourse dan mengembalikan List berisi StudentModel yang memodelkan student yang mengambil course dengan idCourse yang dipilih.

5) ***Membuat method viewCourse pada Class StudentController***

```
152    @RequestMapping("/course/view/{idCourse}")
153    public String viewCourse (Model model,
154            @PathVariable(value = "idCourse") String idCourse)
155    {
156        CourseModel course = studentDAO.selectCourse(idCourse);
157
158        if (course != null) {
159            model.addAttribute ("course", course);
160            return "view-course";
161        } else {
162            model.addAttribute ("id-course", idCourse);
163            return "course-not-found";
164        }
165    }
```

Method ini menerima parameter idCourse yang ingin dilihat. Kemudian method akan memeriksa apakah Course dengan idCourse tersebut ada pada database. Jika course tidak ditemukan, method mengembalikan view “course-not-found.html”. Jika course ditemukan, method mengembalikan view “view-course”.

6) ***Membuat view untuk menampilkan data Course***

Tampilan view “view-course”

```
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.id_course}">Student NPM</h3>
8         <h3 th:text="'Nama = ' + ${course.name}">Student Name</h3>
9         <h3 th:text="'SKS = ' + ${course.credits}">Student GPA</h3>
10        <h3>Mahasiswa yang mengambil</h3>
11        <ul th:each="student,iterationStatus: ${course.students}">
12            <li th:text="${student.npm} + '-' + ${student.name}" >
13            Nama kuliah-X SKS
14            </li>
15        </ul>
16
17    </body>
18 </html>
```

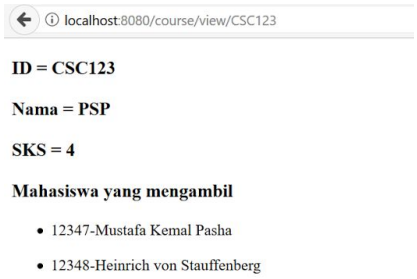
Tampilan view “course-not-found”

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3     <head>
4         <title>Course not found</title>
5     </head>
6     <body>
7         <h1>Course not found</h1>
8         <h3 th:text="'ID = ' + ${idCourse}">Course ID</h3>
9     </body>
10 </html>
```

Nama : Lintang Matahari Hasani
NPM : 1506689231

Berikut merupakan tampilan fitur menampilkan mata kuliah beserta student yang mengambil pada view mata kuliah (view-course.html):

Jika Course ditemukan



Jika idCourse tidak ada pada database

Nama : Lintang Matahari Hasani
NPM : 1506689231

Lesson Learned

Pada tutorial 5 ini, saya banyak belajar mengenai bagaimana menggunakan database (khususnya MySQL) dengan aplikasi Spring Boot. Secara umum, tutorial ini memberikan gambaran umum mengenai penggunaan database untuk aplikasi berbasis Spring Boot.

Pada tutorial ini, saya mempelajari penggunaan anotasi-anotasi yang terkait dengan query database, seperti @Select, @Results, dan @Many pada Interface Java Spring Boot. Anotasi @Select berfungsi membuat query untuk database. Anotasi ini telah digunakan pada tutorial sebelumnya. Anotasi @Result berfungsi meng-”assign” nilai pada kolom hasil query database tertentu ke property tertentu. Anotasi @Many berfungsi menangani pemilihan multiple object dengan memanggil method lain yang mengembalikan object tersebut.

Nama : Lintang Matahari Hasani
NPM : 1506689231