

Write Up Tutorial 5

Latihan

1. Ubah method `selectAllStudents` di kelas `StudentMapper` agar halaman `viewall` menampilkan semua student beserta daftar 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 ();
```

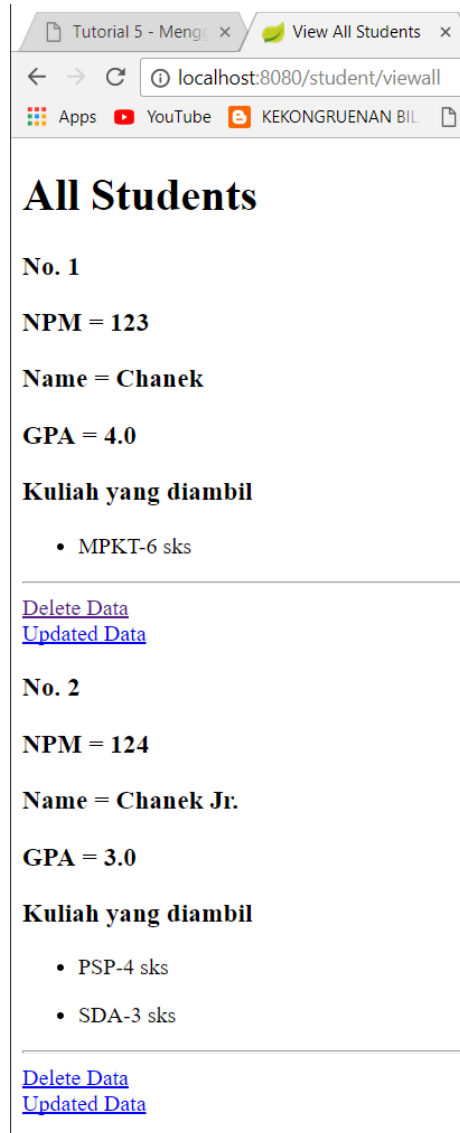
Tahap pertama yang dilakukan adalah mengubah method `selectAllStudent`, method ditambahkan anotasi `@Results` untuk memetakan hasil dari query select ke dalam class model. Dan untuk *variable courses variabel column* diisi dengan `npm` karena kolom `npm` pada database akan dikirim menjadi parameter di method `selectCourses`. Variabel `javaType` menentukan class yang menjadi kembalian. Kemudian variabel `many` diset dengan method yang akan mengisi variabel `courses` yaitu `selectCourses`.

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

Kemudian pada `file view.html` dilakukan iterasi dari setiap student untuk memperlihatkan course apa saja yang diambil.

Sehingga jika mengakses <http://localhost:8080/student/viewall> akan menampilkan tampilan seperti berikut ini.



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

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

Yang pertama dilakukan adalah menambahkan method `selectStudentCourses` pada class `StudentMapper`. Method ini melakukan seleksi untuk mengambil npm, nama dan gpa dari mahasiswa yang mengambil mata kuliah yang sama. Method ini akan mengembalikan list yang berisi mahasiswa.

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

Kemudian pada *class* StudentMapper ditambahkan lagi method selectCourse dengan parameter id_course. Method ini akan mengambil idCourse, nama, dan jumlah sks dari suatu *course* yang kemudian akan dipetakan kedalam *class model*. Dilakukan juga pengambilan data dari list mahasiswa yang mengambil course tersebut. Hal ini dilakukan dengan memanggil method lain, yaitu selectStudentCourse(). Hasil list tersebut kemudian dipetakan pula ke dalam *class model*.

```
public interface StudentService
{
    CourseModel selectCourse(String id_course);
}

public class StudentServiceDatabase implements StudentService
{
    @Autowired
    private StudentMapper studentMapper;

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

Selanjutnya diimplementasikan pula method selectCourse() pada *interface* StudentService dan pada *class* StudentServiceDatabase. Pada *class* StudentServiceDatabase method selectCourse memanggil method selectCourse() dari *class* StudentMapper.

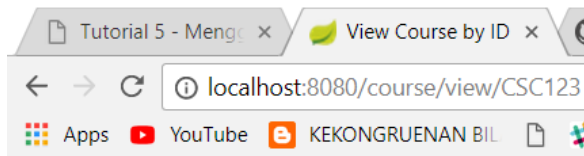
```
@RequestMapping("/course/view/{id_course}")
public String viewCourse (Model model,
    @PathVariable(value = "id_course") String id_course)
{
    CourseModel course = studentDAO.selectCourse (id_course);
    if(course != null){
        model.addAttribute("course", course);
        return "viewCourse";
    }else{
        model.addAttribute("course", course);
        return "courseNotFound";
    }
}
```

Lalu pada *class* StudentController, ditambahkan pula method viewCourse(). Didalam method ini dilakukan pengambilan data menggunakan method selectCourse() yang ada di dalam StudentServiceDatabase. Lalu jika data dari course yang dipilih tidak ada maka akan membuka halaman CourseNotFound, jika ada akan dibukakan halaman dari viewCourse.

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

Yang terakhir ditambahkan pula pada halaman viewCourse agar dapat menampilkan data dari *course* yang dicari yaitu ID, nama, dan npm. Serta dilakukan pula iterasi untuk menampilkan nama dan npm dari mahasiswa yang mengambil course tersebut. Sehingga jika dilakukan pencarian untuk course dengan ID CSC123 akan menampilkan halaman sebagai berikut.



ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.