

LATIHAN

1. Jelaskan apa saja hal yang Anda pelajari dari tutorial ini
Pada tutorial kali ini saya belajar untuk menggunakan dua library eksternal seperti pada tutorial sebelumnya, yaitu MyBatis dan Lombok. Saya mempelajari bagaimana membuat table pada phpMyAdmin yang nantinya akan digunakan untuk mengakses database pada program ini. Saya juga mempelajari database dengan Spring Boot menggunakan MyBatis tetapi menggabungkan 2 entity. Saya mempelajari anotasi baru yaitu @Results yang digunakan untuk memetakan hasil dari query select ke class model.
2. Ubah method selectAllStudents pada 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 ();
```

← → ↺ ⬆ ⓘ localhost:8080/student/viewall

All Students

No. 1

NPM = 123

Name = Zahrah

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

No. 2

NPM = 12345

Name = atikah

GPA = 3.0

Kuliah yang diambil

[Delete Data](#)

[Update Data](#)

- Penjelasan method `selectAllStudents()`;
Anotasi `Results` digunakan untuk memetakan hasil dari query select ke class model. Variabel `property` diisi dengan nama variabel di class `StudentModel`, sedangkan variabel `column` diisi dengan nama kolom hasil kueri di database (parameter yang dibutuhkan). `Property` diset dengan variabel `courses` untuk mengisi variabel `courses` di class `StudentModel`. 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`. Setelah itu dimasukkan kedalam `list<StudentModel>` dan disimpan dalam method `selectAllStudents()`
- 3. Buatlah view pada halaman `http://localhost:8080/course/view/{id}` untuk Course sehingga dapat menampilkan data course beserta Student yang mengambil.

- Pada `StudentMapper.java`

```
@Select("select * 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="selectStudentCourse"))
})
CourseModel selectCourse (@Param("id_course") String id_course);

@Select("select * from student join StudentCourse on student.npm = studentcourse.npm where studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudentCourse(@Param("id_course") String id_course);
```

Penjelasan :

Anotasi `Select` mengambil dari table `course` dimana `id_course` sama dengan `id_course` yang dimasukkan pada URL. Anotasi `Results` digunakan untuk memetakan hasil dari query select ke class model. Variabel `property` diisi dengan nama variabel di class `CourseModel`, sedangkan variabel `column` diisi dengan nama kolom hasil kueri di database (parameter yang dibutuhkan). `Property` diset dengan variabel `students` untuk mengisi variabel `students` di class `CourseModel`. Variabel `column` diisi dengan `id_course` karena kolom `id_course` pada database akan dikirim menjadi parameter di method `selectStudentCourse`. Anotasi `Select` yang kedua mengambil dari table `student` dan men-join antara `npm` pada table `student` dan `npm` pada table `studentcourse` dimana `id_course` pada table `studentcourse` sama dengan `id_course` yang dimasukkan pada URL. Lalu dimasukkan ke list bernama `selectStudentCourse`

- Pada `StudentService.java`

```
CourseModel viewCourse(String idCourse);
```

Penjelasan :

Mendeclare viewCourse untuk nantinya berpengaruh pada studentController

- Pada StudentServiceDatabase.java

Penjelasan :

Melakukan override pada method viewCourse. viewCourse akan melakukan selectCourse

```
@Override
public CourseModel viewCourse(String idCourse) {
    log.info("Course " + idCourse);
    return studentMapper.selectCourse(idCourse);
}
```

- Pada StudentController.java

```
@RequestMapping("/course/view/{id}")
public String viewCourse (Model model,
    @PathVariable(value = "id") String idCourse)
{
    CourseModel course = studentDAO.viewCourse(idCourse);

    if (course != null) {
        model.addAttribute ("course", course);
        return "view-course";
    } else {
        model.addAttribute ("id_course", idCourse);
        return "not-found-course";
    }
}
```

Penjelasan :

Anotasi RequestMapping berisi /course/view/{id} yang nantinya id diisi oleh id_course yang ingin ditampilkan. Apabila course ada pada database maka menampilkan view-course.html. dan apabila course tidak ada pada database akan menampilkan not-found-course.html

- Membuat view-course.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View Student by NPM</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}">SKS</h3>

    <h3>Mahasiswa yang mengambil</h3>
    <ul th:each="student, iterationStatus: ${course.students}">
      <li th:text="${student.npm} + ' - ' + ${student.name} + ' sks'">
    </li>
    </ul>
  </body>
</html>

```

Penjelasan :

Mengambil idCourse, name dan credits pada table course untuk ditampilkan. Lalu menampilkan mahasiswa yang mengambil dengan menyertakan npm dan nama yang diambil dari table student.

- Membuat not-found-course.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="'Course = ' + ${id}">Course ID</h3>
  </body>
</html>

```

Penjelasan :

Menampilkan Course not found beserta ID Course yang dimasukkan pada URL

← → ↻ 🏠 ⓘ localhost:8080/course/view/CSC130

Course not found

Course = CSC130

← → ↻ 🏠 ⓘ localhost:8080/course/view/CSC123

ID = CSC123

Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124 - Halim