Muhammad Akhdan Shidqi
1506689295

LESSON LEARNED

Dalam tutorial kali ini, saya telah mempelajari penggunaan database lebih lanjut menggunakan spring boot. Pada nyatanya, kita tidak hanya menggunakan hanya 1 tabel pada database untuk mengimplementasikan arsitektur yang baik, namun kita akan menggunakan lebih dari 1 atau bahkan banyak sekali tabel yang berbeda-beda di database. Ditambah lagi dengan banyaknya yang saling berhubungan akan membutuhkan query yang lebih rumit jika hanya mengandalkan query untuk melakukan mapping. Karena itu spring boot menyediakan fasilitas untuk membuat banyak model dan menghubungkan setiap tabel yang berguna mengambil data yang kita butuhkan.

LATIHAN

1. Melihat daftar student yang ada dengan mata kuliah yang mereka ambil

- Berikut adalah **method selectAllStudent** pada class mapper:

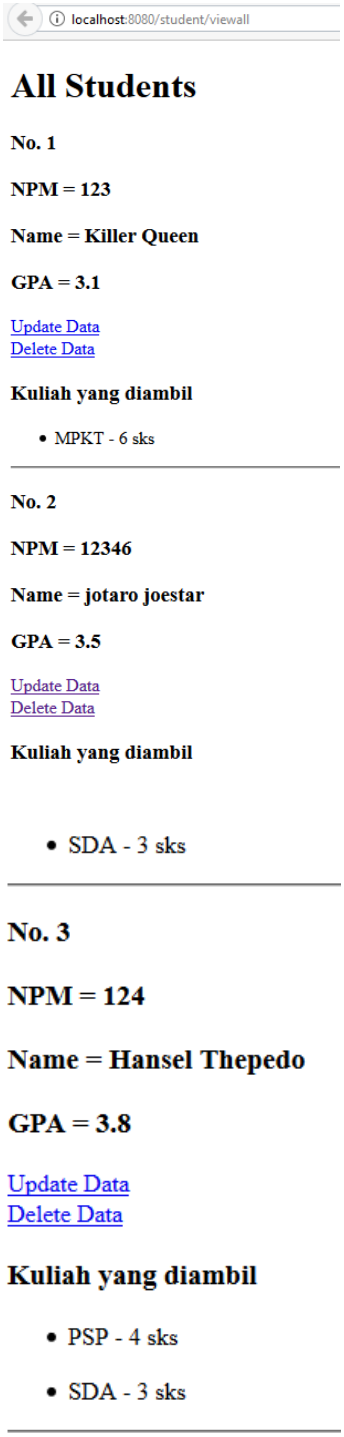
```
@Select("select npm, name, gpa from student")
      @Results(value = {
      @Result(property="npm", column="npm"),
      @Result(property="name", column="name"),
      @Result(property="gpa", column="gpa"),
      @Result(property="courses", column="npm",
            javaType = List.class,
            many=@Many(select="selectCourses"))
})
List<StudentModel> selectAllStudents ();
```

Karena method ini memiliki kemiripan dengan method selectStudent(), saya hanya perlu memasukan anotasi @Result yang sama ke dalam method ini, saya mengambil atribut pada model student, namun karena harus mengambil atribut pada tabel course, saya harus memanggilnya dengan cara yang sama di method selectStudent(). Untuk method selectCourses yang ada di situ bertugas untuk mencari course pada tabel studentcourse lewat npm yang diberikan, dan mengakses query yang ada di method selectCourse untuk mencari course yang diambil oleh npm dari input. karena saya akan memanggil banyak siswa, method yang dipanggil adalah selectAllStudents().

Muhammad Akhdan Shidqi
1506689295

Hasil tampilan

- **Halaman viewAll**



Muhammad Akhdan Shidqi
1506689295

2. Melihat daftar course yang ada dengan student yang mengambil course tersebut

Pada soal sebelumnya, yang dibutuhkan adalah menampilkan semua student yang ada, namun karena sekarang dibutuhkan untuk menampilkan course yang ingin dilihat lewat path variable, saya harus membuat method selectCourse() di kelas mapper, kelas StudentService, kelas StudentServiceDatabase dan kelas StudentController.

- Berikut adalah **method selectCourse** pada class mapper:

```
@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="selectStudents"))
      })
CourseModel selectCourse(@Param("id_course") String idcourse);
```

Method ini bertugas sebagai penghubung ke database dan memanggil course dengan parameter id_course dari input. Anotasi @Result tersebut merupakan kebalikan dari method selectStudent() sebelumnya dimana sekarang property yang dimasukan adalah property dari course karena sekarang kita akan menampilkan data dari model course. Kemudian saya juga perlu mengambil data list dari students karena saya harus siswa mana saja yang mengambil course yang bersangkuatn. Untuk melakukan hal tersebut saya harus memanggil method selectStudents yang saya buat sendiri. Method ini bertugas untuk mencari siswa mana saja yang mengambil kuliah yang ingin ditampilkan. Berikut adalah method selectStudents():

```
@Select("select student.npm, 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);
```

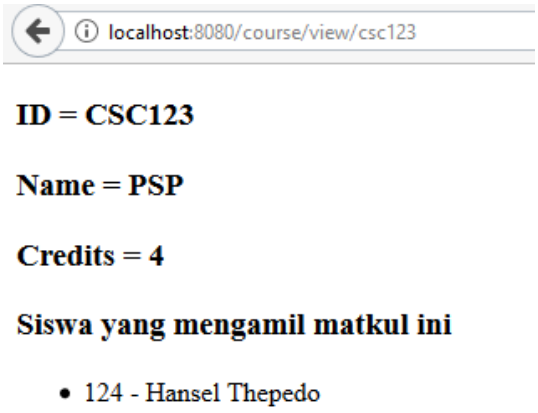
Muhammad Akhdan Shidqi
1506689295

Berikut adalah method baru yang ada di Controller:

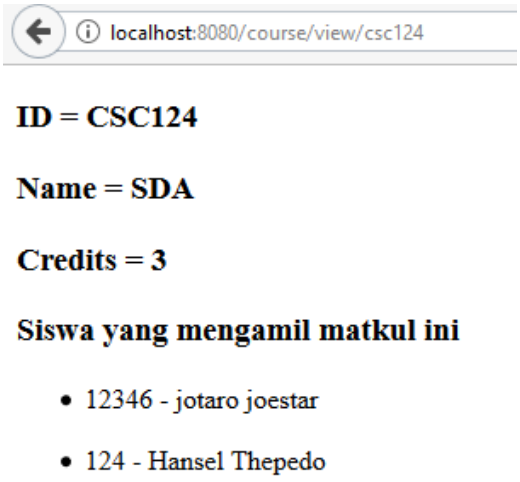
```
@RequestMapping("/course/view/{id}")
public String viewCoursePath(Model model, @PathVariable(value = "id")
String id)
{
	CourseModel course = studentDAO.selectCourse(id);
	model.addAttribute("course", course);
	return "viewCourse";
}
```

Hasil tampilan

- **Course/view/csc123**

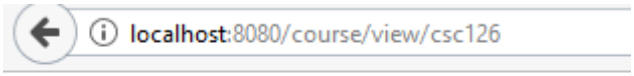


- **Course/view/csc124**



Muhammad Akhdan Shidqi
1506689295

- **Course/view/csc126**



ID = CSC126

Name = MPKT

Credits = 6

Siswa yang mengamil matkul ini

- 123 - Killer Queen