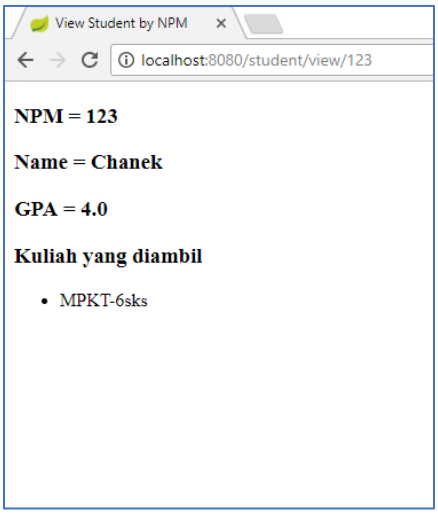
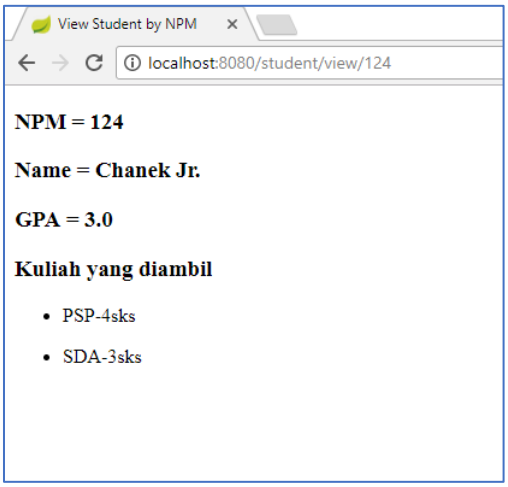
Ervina Puspita Sari | B
1506689414

TUTORIAL

- Menambahkan Model





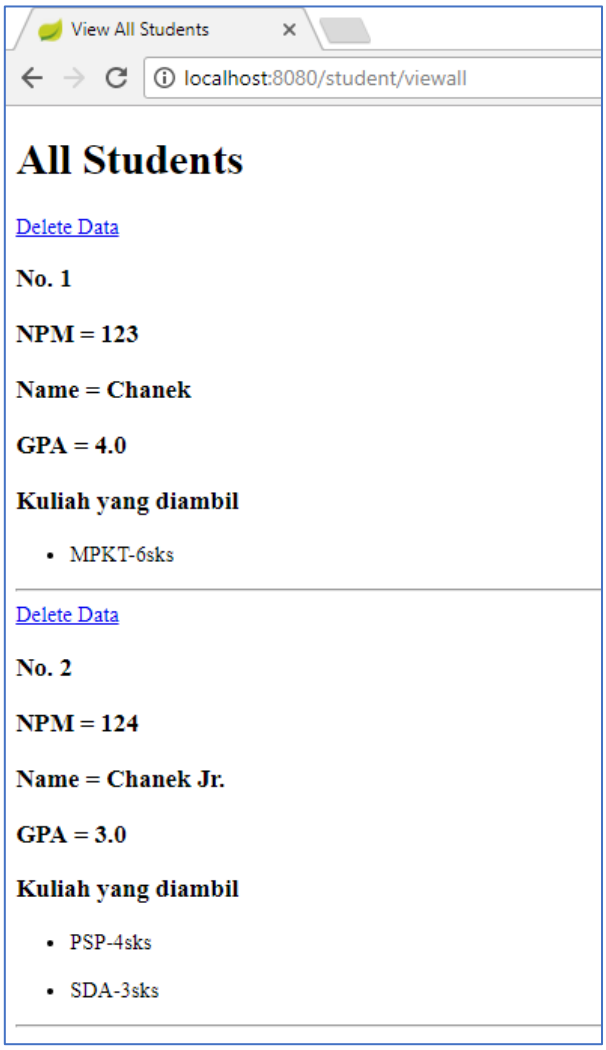
Pada tutorial ini saya mengikuti apa yang dijelaskan di dalam soal. Saya mengubah beberapa kelas, di antaranya menambahkan kelas CourseModel.java pada package com.example.model , menambahkan List<CourseModel> corses pada kelas StudentModel.java , menambah parameter null pada StudentModel student dalam StudentController.java . Selain itu saya juga mengganti beberapa @Select pada StudentMapper yang berguna untuk mengisi variable List<CourseModel> courses, menampilkan seluruh daftar course yang diambil mahasiswa dengan mengembalikan List <CourseModel> dengan menggunakan anotasi @Result yang terdiri dari property dan column. Anotasi @Result membuat kita dapat memilih dan mengambil data dari database untuk ditampilkan atau dipilih.

LATIHAN

1. Viewall pada student
   Untuk menampilkan seluruh data student beserta jadwal course yang diambil, saya merubah method selectAllStudents() pada kelas StudentMapper.java sesuai yang diinstruksikan dalam soal. Saya mengubahnya dengan menyalin @Select yang sebelumnya yaitu pada method selectStudent(), hanya saja parameternya yang berupa npm saya tiadakan, karena method selecAllStudents() tidak membutuhkan paramaeter apapun.

```
@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 ();
```



2. View course/id_course tertentu
   Untuk menampilkan course yang ada dengan daftar mahasiswa yang mengambilnya, saya banyak membuat kelas baru, di antaranya CourseController.java, CourseMapper.java, CourseModel.java, CourseService.java, CourseServiceDatabase.java, hingga viewcoruse.html. Karena yang ingin ditampilkan bukan student, melainkan course, maka saya membuat skenario baru dengan menambahkan Mapping, Controller, hingga Service yang merepresentasikan Course. Isinya pun tidak jauh berbeda dengan yang ada dalam

Student. Hanya saja dalam Course ini sifatnya masih terbatas, yaitu hanya bisa view saja, tanpa fitur-fitur yang lain.

Ada 3 anotasi @Select dalam CourseMapper.java yang di antaranya berfungsi untuk mengisi list of students, view student yang dipilih, dan mengembalikan list dari student yang terpilih tersebut.

```
//untuk mengisi list of students
   @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 id_course);


@Select("select id_course, name, credits from course")
List<CourseModel> selectAllCourses ();


//Untuk view mahasiswa yg ambil
@Select("select name, npm, gpa from student")
@Results (value = {
         @Result (property="name", column="name"),
         @Result (property="npm", column="npm"),
         @Result (property="gpa", column="gpa"),
                 @Result (property="course", column="id_course",
         javaType = List.class, many= @Many(select= "selectCourse"))
})
List<CourseModel> selectStudent();


//Mengembalikan List of Students dari StudentCourse  dengan id_course tertentu
@Select("select student.npm, name, gpa " +
"from student join studentcourse " +
"on student.npm = studentcourse.npm "  +
"where studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudents (@Param("id_course") String id_course);
```

Sementara dalam CourseService.java dan CourseServiceDatabase.java method yang ada hanyalah selectCourse() dan selectAllCourses()

Selain itu, controller yang ada dalam CourseController.java hanya 1 yaitu view berdasarkan id_course yang tertulis pada url

```
@RequestMapping("/course/view/{id_course}")
public String viewCourse (Model modelcourse,
@PathVariable(value = "id_course") String id_course) {
     CourseModel course = courseDAO.selectCourse(id_course);
     if (course != null) {
             modelcourse.addAttribute ("course", course);
             return "viewcourse";
     } else {
             modelcourse.addAttribute ("id_course", id_course);
             return "not-found";
     }
}
```

Ervina Puspita Sari | B
1506689414