

Write-up Tutorial 5

a. Ringkasan dari materi yang Anda telah pelajari pada tutorial kali ini

Saya mempelajari bagaimana cara menampilkan hasil query dari 2 tabel yang berbeda (join).

b. Method yang diubah

1. SelectAllStudents

a. Menambahkan result pada parameter di StudentMapper.java

```
@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 ();
```

b. Menambahkan list kuliah yang diambil pada 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}">
<a th:href="/student/delete/" + ${student.npm}>Delete Data</a>
<a th:href="/student/update/" + ${student.npm}>Update Data</a>
<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/>
</div>
</body>
</html>
```

May Iffah Rizki
1506757661
APAP – A

c. Hasil:

View All Students

localhost:8080/student/viewall

Apps gn Home | Bill Gates OpenStax College Student Centered e-L Top 30 OOPS Concep

All Students

[Delete Data](#) [Update Data](#)

No. 1

NPM = 123

Name = may

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

[Delete Data](#) [Update Data](#)

No. 2

NPM = 12345

Name = Iffah

GPA = 4.0

Kuliah yang diambil

[Delete Data](#) [Update Data](#)

No. 3

NPM = 124

Name = mayy

GPA = 4.0

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

2. Menambahkan View pada Course

- a. Menambahkan method selectCourse pada interface StudentService.java

```
CourseModel selectCourse (String id_course);
```

- b. Menambahkan method selectCourse pada StudentServiceDatabase.java untuk memanggil method tersebut

```
@Override  
public CourseModel selectCourse(String id_course) {  
    // TODO Auto-generated method stub  
    Log.info ("select course with id_course {}", id_course);  
    return studentMapper.selectCourse (id_course);  
}
```

- c. Membuat method selectStudents di StudentMapper.java untuk menghasilkan mahasiswa yang mengambil suatu course.

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

- d. Membuat method selectCourse di StudentMapper.java untuk menghasilkan atribut suatu course beserta daftar mahasiswa yang mengambil course tersebut.

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

- e. Membuat page viewcourse.html

May Iffah Rizki
1506757661
APAP – A

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

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

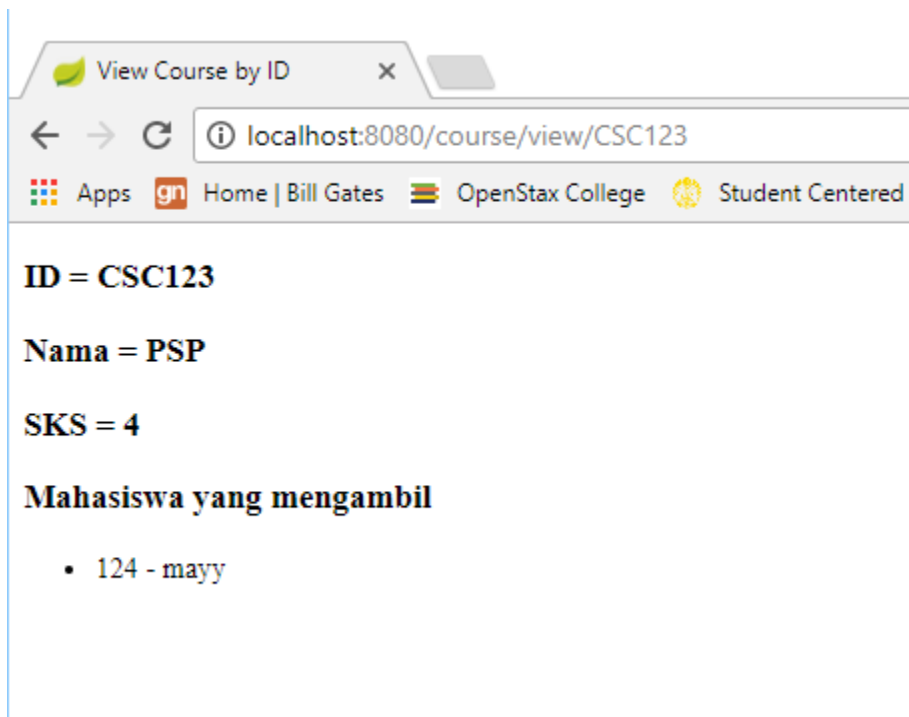
g. Membuat method viewCoursePath untuk menampilkan hasil view courses di viewcourse.html jika id course ditemukan, dan course-not-found.html jika id course tidak

May Iffah Rizki
1506757661
APAP – A

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

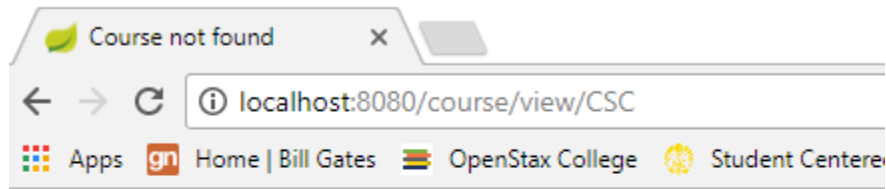
h. Hasil:

i. ID course ditemukan:



May Iffah Rizki
1506757661
APAP – A

ii. ID course tidak ditemukan:



Course not found

ID = CSC