

1. Ubah method selectAllStudents pada kelas StudentMapper agar halaman viewall menampilkan semua student beserta daftar kuliah yang diambil.

**All Students**

No. 1

NPM = 123

Name = aduhai

GPA = 2.9

Kuliah yang diambil

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

No. 2

NPM = 124

Name = hoho

GPA = 3.6

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

[Delete Data](#)

[Update Data](#)

No. 3

NPM = 125

Name = rei

GPA = 4.0

Kuliah yang diambil

- DDP 1-4 sks

[Delete Data](#)

[Update Data](#)

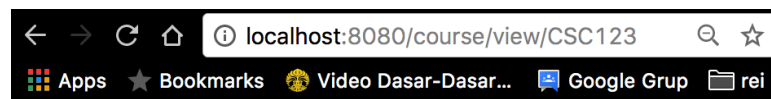
Pada awalnya saya membuat CourseModel selanjutnya saya menambahkan code sbg berikut di StudentMapper class, hal ini guna untuk membggail selectAllStudents:

```
@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 ();
```

Dan menambahkan code seperti dibawah pada viewall.html. Hal ini guna dilakukannya iterasi th:each untuk semua course yang diambil oleh student

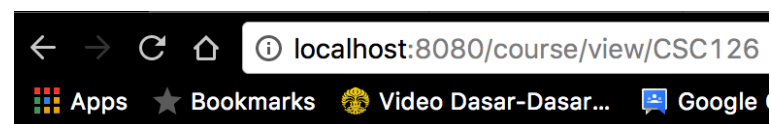
```
<div th:each="student, iterationStatus: ${students}">
  <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
  <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
  <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
  <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  <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>
  <a th:href="/student/delete/" + ${student.npm}"> Delete Data</a><br/>
  <a th:href="/student/update/" + ${student.npm}"> Update Data</a><br/>
  <hr/>
</div>
```

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



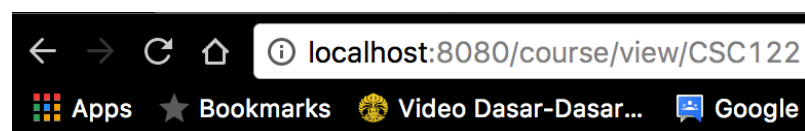
ID = CSC123  
Nama = PSP  
SKS = 4  
Mahasiswa yang mengambil

- 124-hoho



ID = CSC126  
Nama = MPKT  
SKS = 6  
Mahasiswa yang mengambil

- 123-aduhai



Course not found  
ID = null

Saya membuat sebuah class dan hmtl baru diantaranya:

- a. Class CourseController

Class ini berguna untuk memanggil method sesuai dengan address input yang diberikan.

- Jika user memasukkan addres <http://localhost:8080/course/view/CSC122> maka akan berjalan code dibawah

```
@RequestMapping("/course/view/{id_course}")
public String viewPath (Model model,
    @PathVariable(value = "id_course") String id_course)
{
    CourseModel course = courseDAO.selectCourse (id_course);

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

- Sedangkan jika user memasukkan <http://localhost:8080/course/viewall>, maka akan berjalan code dibawah ini

```
@RequestMapping("/course/viewall")
public String view (Model model)
{
    List<CourseModel> courses = courseDAO.selectAllCourses();
    model.addAttribute ("courses", courses);

    return "viewall-course";
}
```

## b. Interface CourseMapper

Untuk mendefine property dan mangambil data yang ada didatabase

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

@Select("select id_course, name, credits from 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="selectStudents"))
})
List<CourseModel> selectAllCourses ();
```

## c. Interface CourseService

Penamabahn code seperti dibawah ini:

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

```
List<CourseModel> selectAllCourses ();
```

d. Class CourseServiceDatabase

- Penambahan selectCourse untuk menjalankan query dari selectcourse

```
public CourseModel selectCourse (String id_course)
{
    log.info ("select cours with id_course {}", id_course);
    return courseMapper.selectCourse (id_course);
}
```

- Penambahan selectAllCourses untuk menjalankan query dari selectAllcourses

```
@Override
public List<CourseModel> selectAllCourses ()
{
    log.info ("select all course");
    return courseMapper.selectAllCourses ();
}
```

e. File html view-course

Untuk menampilkan course sesuai inputan beserta student yang mngambilnya

f. File html viewall-course

Untuk menampilkan semua courses yang ada beserta student yang mengambilnya

g. File html not-found-course

Untuk menampilkan not found course jika idcourse yang dimasukkan tidak terdaftar.