- Create Table Course

	#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan
☐	1	**id_course** 🔑	varchar(20)			Tidak	*Tidak ada*
☐	2	**name**	varchar(45)			Ya	*NULL*
☐	3	**credits**	double			Ya	*NULL*

- Insert into table course

id_course	name	credits
CSC123	PSP	4
CSC124	SDA	3
CSC125	DDP 1	4
CSC126	MPKT	6

- Table studentcourse

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan
1	**npm**	varchar(20)			Tidak	*Tidak ada*
2	**id_course**	varchar(30)			Tidak	*Tidak ada*

- Isi table studentcourse

npm	id_course
123	CSC126
124	CSC123
124	CSC124

MENAMBAHKAN MODEL

1. Membuka kode tutorial 04 yang sudah dikerjakan minggu lalu.

2. Karena kita menambahkan Model baru maka kita perlu menambahkan class

CourseModel pada package com.example.model

```
package com.example.model;
import java.util.List;
import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

@Data
@AllArgsConstructor
@NoArgsConstructor

public class CourseModel {
    private String idCourse;
    private String name;
    private Integer credits;
    private List <StudentModel> students;


}
```

3. Karena Student memiliki daftar Course yang dia ambil, maka pada class StudentModel perlu ditambahkan list of CourseModel pada variabel:

```
package com.example.model;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
import java.util.List;

@Data
@AllArgsConstructor
@NoArgsConstructor
public class StudentModel
{
    private String npm;
    private String name;
    private double gpa;
    private List <CourseModel> courses;
}
```

4. Selanjutnya kita akan membuat method pada StudentMapper yang mengembalikan list of course pada Student dengan NPM tertentu. Methodnya adalah sebagai berikut:

```
@Select("select course.id_course, name, credits " +
"from studentcourse join course " + "on studentcourse.id_course = course.id_course "
+ "where studentcourse.npm = #{npm}")
List<CourseModel> selectCourses (@Param("npm") String npm);
```

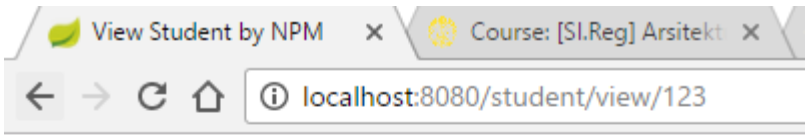
5. Selanjutnya kita akan menghubungkan method selectStudent dengan method selectCourses agar saat melakukan select maka variabel List juga akan terisi. Pada class StudentMapper terdapat method selectStudent diubah menjadi

```
@Select("select npm, name, gpa from student where npm = #{npm}")
@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"))


})
StudentModel selectStudent (@Param("npm") String npm);
```

6. Selanjutnya kita akan menambahkan view pada view.html untuk menampilkan list kuliah yang diambil oleh mahasiswa seperti berikut:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Student by NPM</title>
    </head>
    <body>
        <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-X SKS
            </li>
        </ul>
    </body>
</html>
```

7. Jalankan program tersebut dan buka http://localhost:8080/student/view/123



NPM = 123

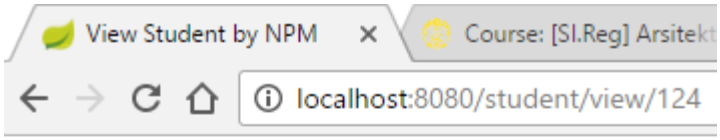
Name = Chanek

GPA = 4.0

Kuliah yang diambil

- MPKT-6 sks

Jalankan program tersebut dan buka http://localhost:8080/student/view/124



NPM = 124

Name = Chanek Jr.

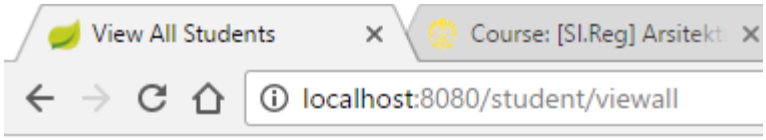
GPA = 3.0

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

LATIHAN 1.

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



All Students

No. 1

NPM = 123

Name = Chanek

GPA = 4.0

Delete Data Update Data

Kuliah yang diambil

- MPKT-6 sks

No. 2

NPM = 124

Name = Chanek Jr.

GPA = 3.0

Delete Data Update Data

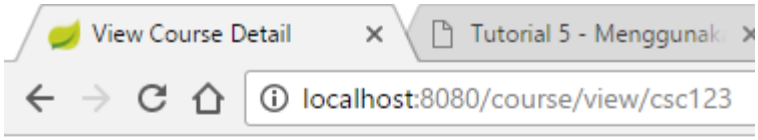
Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

---

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

http://localhost:8080/course/view/csc123



ID = CSC123

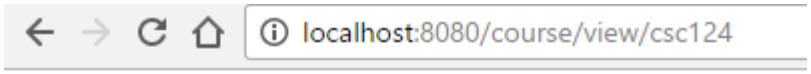
Nama = PSP

SKS = 4

Mahasiswa yang mengambil

- 124 - Chanek Jr.

http://localhost:8080/course/view/csc124



ID = CSC124

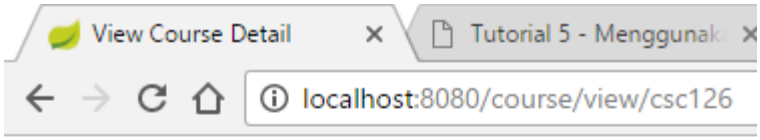
Nama = SDA

SKS = 3

Mahasiswa yang mengambil

- 124 - Chanek Jr.

http://localhost:8080/course/view/csc126



ID = CSC126

Nama = MPKT

SKS = 6

Mahasiswa yang mengambil

- 123 - Chanek

- Lesson Learned

- Application of @Result annotation in MyBatis. This annotation is used to pass parameter along in the beginning of the of the result mapping, stating that parameter should get the value from the table's certain column . Use this annotation to map table column with java property that helps to get result in @Select annotation.
- @Many annotation: A mapping to a collection property of a complex type. We can pass multiple parameters using @Many

- Method merubah 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 ();
```

Select semua student yang ada untuk viewAll. Lalu many diisi dengan courses yang diambil oleh student.

Lalu di viewAll.html

```
<h1>All Students</h1>

<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>
    <a th:href="'/student/delete/' + ${student.npm}" > Delete Data</a>
    <a th:href="'/student/update/' + ${student.npm}" > Update Data</a><br/>
     <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/>
```

Setiap mahasiswa akan dilakukan iterasi pada List<CourseModel> courses yang dimiliki oleh mahasiswa tersebut. Lalu ditampilkan nama dan jumlah sks nya yang didapat dari course dari List Course yang dimiliki mahasiswa.

- Method menambah View pada Course

  Di StudentMapper tambahkan method SelectCourse

```
@Select("select name, id_course, credits from course where id_course = #{idCourse}")
@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("idCourse") String idCourse);
```

Method ini untuk mengambil informasi-informasi dari sutatu course dan list student yang mengambil course tersebut. Method ini memanggil method lain yaitu select Students.

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

Method SelectStduents ini akan mengambil mahasiswa yang mengambil course dengan id yang diberikan dari parameter.

Kemudian buat view bernama viewCourse.html

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

Akan menampilkan detail seperti nama, id, dan jumlah sks course. View akan menampilkan juga student yang mengambil course ini yang didapat dari model di controller.