

Latihan

1. Latihan 1: Buatlah *service* untuk mengembalikan seluruh student yang ada di basis data. *Service* ini mirip seperti *method* `viewAll` di Web Controller. *Service* tersebut di-*mapping* ke `"/rest/student/viewall"`.

```
@RequestMapping("/student/viewall")
public List<StudentModel> viewAll() {
    List<StudentModel> students = studentService.selectAllStudents();
    return students;
}

@Override
public List<StudentModel> selectAllStudents ()
{
    log.info ("select all students");
    return studentMapper.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="selectAllCourses"))
})
List<StudentModel> selectAllStudents();
```

Hasil keluaran:

Mayanggita Rachmaryza Indra

1506732974

Kelas B



String parse	JS eval
<pre> "students":null }], { "npm":"124", "name":"Chanek Jr.", "gpa":3.4, "courses": [{ "id_course":"CSC123", "name":"PSP", "credits":4, "students":null }, { "id_course":"CSC124", "name":"SDA", "credits":3, "students":null }] }, { "npm":"18181", "name":"mayanggita", "gpa":4.0, "courses": [] }]</pre>	<pre> "students":null }], { "npm":"124", "name":"Chanek Jr.", "gpa":3.4, "courses": [{ "id_course":"CSC123", "name":"PSP", "credits":4, "students":null }, { "id_course":"CSC124", "name":"SDA", "credits":3, "students":null }] }, { "npm":"18181", "name":"mayanggita", "gpa":4, "courses": [] }]</pre>

Penjelasan:

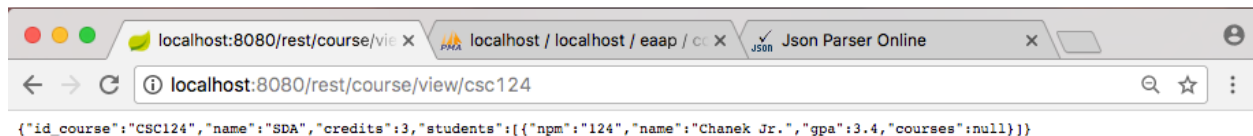
Method viewAll ini akan menginisiasi List of StudentModel yang didapat dari hasil query pada studentMapper.

2. Latihan 2: Buatlah *service* untuk class Course. Buatlah *controller* baru yang terdapat *service* untuk melihat suatu course dengan masukan ID Course (view by ID) dan *service* untuk melihat semua course (view all).

- View course by ID

```
@RestController
@RequestMapping("/rest")
public class CourseRestController {
    @Autowired
    CourseService courseService;

    @RequestMapping("/course/view/{id_course}")
    public CourseModel view(@PathVariable(value = "id_course") String id_course) {
        CourseModel course = courseService.selectCourse(id_course);
        return course;
    }
}
```



Penjelasan: Membuat object CourseModel yang berisi hasil dari selectCourse pada CourseService

- View all course

```

@RequestMapping("/course/viewall")
public List<CourseModel> viewAll() {
    List<CourseModel> courses = courseService.selectAllCourses();
    return courses;
}
}

```





Penjelasan:

Method `viewAllCourse` ini akan menginisiasi List of `CourseModel` yang didapat dari hasil query pada `courseMapper`.

3. Latihan 3: Implementasikan *service consumer* untuk view all Students dengan melengkapi *method* `selectAllStudents` yang ada di kelas `StudentServiceRest`.

```

@Override
public List<StudentModel> selectAllStudents() {
    List<StudentModel> students = restTemplate.getForObject("http://localhost:8080/rest/student/viewall", List.class);
    return students;
}

@Override
public List<StudentModel> selectAllStudents() {
    log.info("REST - select all student");
    return studentDAO.selectAllStudents();
}

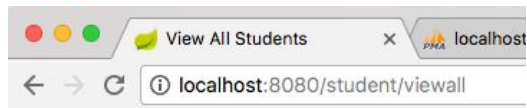
```

Mayanggita Rachmaryza Indra

1506732974

Kelas B

<http://localhost:8080/student/viewall>



All Students

No. 1

NPM = 123

Name = Chanek

GPA = 3.6

Kuliah yang diambil

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

No. 2

NPM = 124

Name = Chanek Jr.

GPA = 3.4

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

[Delete Data](#)

[Update Data](#)

No. 3

NPM = 18181

No. 3

NPM = 18181

Name = mayanggita

GPA = 4.0

Kuliah yang diambil

[Delete Data](#)

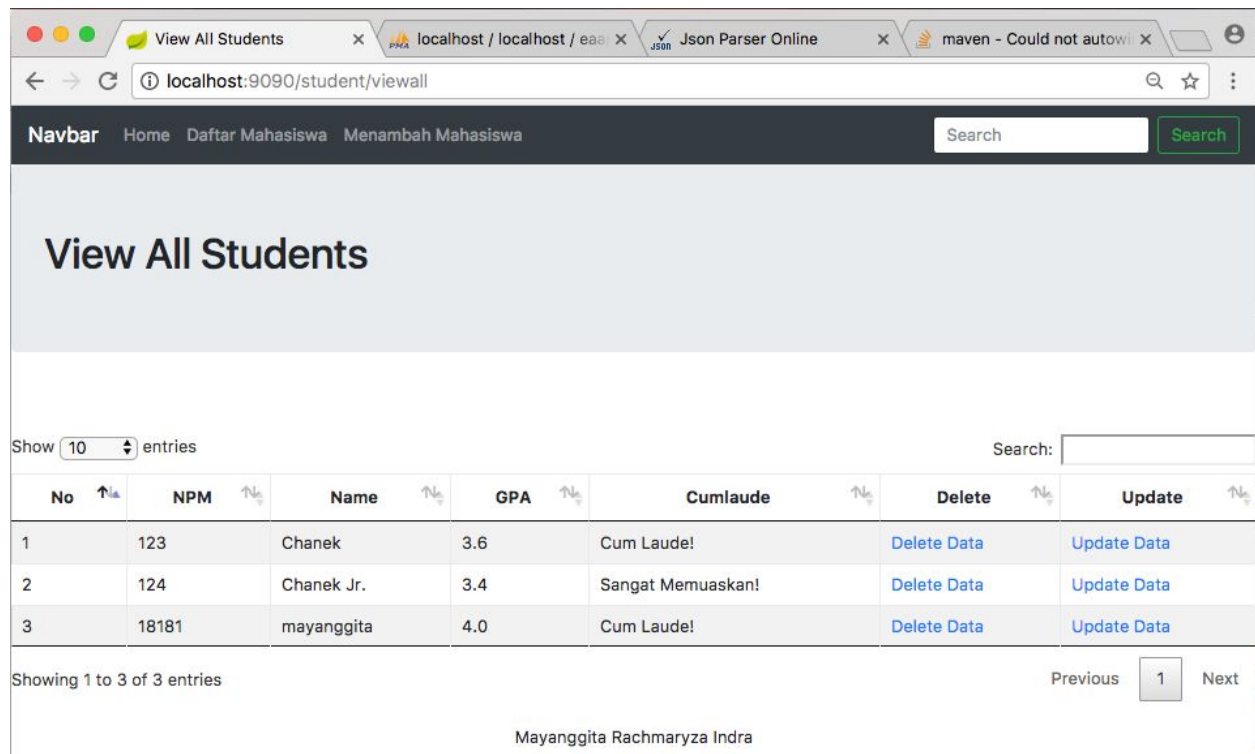
[Update Data](#)

Mayanggita Rachmaryza Indra

1506732974

Kelas B

<http://localhost:9090/student/viewall>



View All Students

Show 10 entries Search:

No	NPM	Name	GPA	Cumlaude	Delete	Update
1	123	Chanek	3.6	Cum Laude!	Delete Data	Update Data
2	124	Chanek Jr.	3.4	Sangat Memuaskan!	Delete Data	Update Data
3	18181	mayanggita	4.0	Cum Laude!	Delete Data	Update Data

Showing 1 to 3 of 3 entries Previous 1 Next

Mayanggita Rachmaryza Indra

Penjelasan: Method SelectAllStudent pada StudentDAOImpl akan menginisiasi list of StudentModel dengan nama students yang berisi semua student yang ada yang didapat dari <http://localhost:8080/rest/student/viewall>

4. Latihan 4: Implementasikan *service consumer* untuk *class* CourseModel dengan membuat *class-class* DAO dan *service* baru.

- View course by Id

Membuat interface CourseDAO

```
public interface CourseDAO {  
  
    CourseModel selectCourse (String id_course);  
  
    List<CourseModel> selectAllCourses();  
}
```

Membuat kelas CourseDAOImpl, implementasikan method selectCourses

```
@Service  
public class CourseDAOImpl implements CourseDAO {  
    @Autowired  
    private RestTemplate restTemplate;  
  
    @Override  
    public CourseModel selectCourse(String id_course) {  
        CourseModel course = restTemplate.getForObject("http://localhost:8080/rest/course/view/"+id_course, CourseModel.class);  
        return course;  
    }  
  
    @Override  
    public List<CourseModel> selectAllCourses() {
```

Hasil keluaran:

The image shows two browser screenshots. The top screenshot displays a REST client (likely Swagger UI) at the URL `localhost:8080/rest/course/view/csc124`. The response is a JSON object: `{ "id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{ "npm": "124", "name": "Chanek Jr.", "gpa": 3.4, "courses": null }] }`. The bottom screenshot shows a web application at `localhost:9090/course/view/csc124`. The page has a dark navbar with links 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa', and a search bar. The main content area is titled 'View Course' and displays the following information: 'ID = CSC124', 'Nama = SDA', 'SKS = 3', 'Mahasiswa yang mengambil' followed by a list containing '124-Chanek Jr.'. The footer of the page reads 'Mayanggita Rachmaryza Indra'.

Penjelasan: Method `SelectCourse` pada `CourseDAOImpl` akan menginisiasi `CourseModel` dengan mengambil objek dari <http://localhost:8080/rest/course/view/csc124> dimana 'CSC124' adalah ID dari course yang ingin dilihat

- View all course

Menambahkan method selectAllCourses() pada interface CourseDAO

```
public interface CourseDAO {

    CourseModel selectCourse (String id_course);

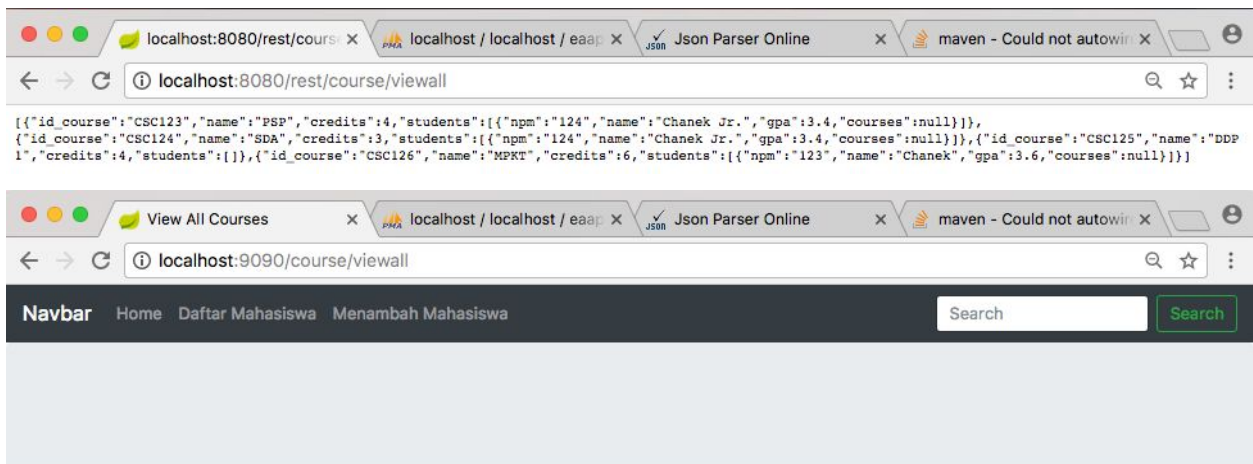
    List<CourseModel> selectAllCourses();

}
```

Mengimplementasi method selectAllCourses() pada kelas CourseDAOImpl

```
@Override
public List<CourseModel> selectAllCourses() {
    List<CourseModel> courses = restTemplate.getForObject("http://localhost:8080/rest/course/viewall", List.class);
    return courses;
}
```

Hasil keluaran:



Show entries Search:

No	ID	Name	Credit
1	CSC123	PSP	4
2	CSC124	SDA	3
3	CSC125	DDP 1	4
4	CSC126	MPKT	6

Showing 1 to 4 of 4 entries Previous 1 Next

Penjelasan: Method SelectAllCourses pada CourseDAOImpl akan menginisiasi List of CourseModel dengan mengambil objek dari <http://localhost:8080/rest/course/viewall>

Mayanggita Rachmaryza Indra

1506732974

Kelas B

Lesson Learned:

Pada tutorial ke-7 ini, saya mempelajari tentang cara membuat dan menggunakan REST pada SpringBoot, dan juga penggunaan Service Producer (*backend*) dan Service Consumer (*frontend*) dimana service producer yang akan memberikan data kepada service consumer berdasarkan apa yang diminta oleh service consumer. Jadi service producer hanya bertanggung jawab untuk menyediakan data dari database, sedangkan service consumer akan menyediakan dan mengolah datanya saja ke pengguna (tidak memiliki database)