

## LESSON LEARNED

Dalam tutorial kali ini, saya belajar bahwa tidak semua aplikasi yang dikembangkan pasti menggunakan 1 bahasa yang sama, namun terdapat banyak sekali Bahasa pemrograman yang ada di luar sana. Oleh karena itu, web service di sini berguna bagi para developer dan stakeholder untuk melihat data dari sebuah web tanpa mengakses database yang ada, namun mengakses objek yang ada pada halaman dengan bantuan *rest*.

## LATIHAN

### Implementasi Producer

#### 1. Viewall

Berikut adalah kode yang saya buat di rest controller:

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

Karena yang diminta adalah list dari semua students yang ada di database, saya hanya perlu merubah return type menjadi list dengan tipe StudentModel. Di dalam methodnya saya hanya perlu membuat objek list yang memanggil method selectAllStudents pada mapper, lalu me-return objek students untuk melihat hasil. Hasilnya adalah sebagai berikut:

### Raw Data:

```
[{"npm": "123", "name": "Killer Queen", "gpa": 3.1, "courses": [{"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": null}]}, {"npm": "12344", "name": "sing sing", "gpa": 2.0, "courses": []}, {"npm": "12346", "name": "jotaro joestar", "gpa": 3.5, "courses": [{"id_course": "CSC124", "name": "SDA", "credits": 3, "students": null}]}, {"npm": "12351", "name": "endru", "gpa": 3.3, "courses": []}, {"npm": "12352", "name": "aji", "gpa": 2.5, "courses": []}, {"npm": "12357", "name": "jodi", "gpa": 3.2, "courses": []}]
```

### Data yang sudah berbentuk json yang rapih:

```
{
  "npm": "12346",
  "name": "jotaro joestar",
  "gpa": 3.5,
  "courses": [
    {
      "id_course": "CSC124",
      "name": "SDA",
      "credits": 3,
      "students": null
    }
  ]
},
{
  "npm": "123",
  "name": "Killer Queen",
  "gpa": 3.1,
  "courses": [
    {
      "id_course": "CSC126",
      "name": "MPKT",
      "credits": 6,
      "students": null
    }
  ]
},
{
  "npm": "12344",
  "name": "sing sing",
  "gpa": 2.0,
  "courses": []
},
{
  "npm": "12351",
  "name": "endru",
  "gpa": 3.3,
  "courses": []
},
{
  "npm": "12352",
  "name": "aji",
  "gpa": 2.5,
  "courses": []
},
{
  "npm": "12357",
  "name": "jodi",
  "gpa": 3.2,
  "courses": []
}
```

## 2. View dan ViewAll Course

Sama seperti rest controller sebelumnya, saya hanya perlu mengubah beberapa instansiasi model dan mengubah method yang dipanggil menjadi method selectCourse dan selectAllCourse yang ada di mapper course, berikut adalah kode di controller:

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

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

Berikut adalah hasil dari view course "CSC123"

**Raw Data:**

```
{"id_course":"CSC123","name":"PSP","credits":4,"students":[]}
```

**Data yang sudah berbentuk json yang rapih:**

```
{
  "id_course": "CSC123",
  "name": "PSP",
  "credits": 4,
  "students": []
}
```

Berikut adalah hasil dari viewall course

**Raw Data:**

```
[{"id_course":"CSC123","name":"PSP","credits":4,"students":
[]}, {"id_course":"CSC124","name":"SDA","credits":3,"students":
[{"npm":"12346","name":"jotaro joestar","gpa":0.0,"courses":null}]},
{"id_course":"CSC125","name":"DDP 1","credits":4,"students":
[]}, {"id_course":"CSC126","name":"MPKT","credits":6,"students":
[{"npm":"123","name":"Killer Queen","gpa":0.0,"courses":null}]]
```

Data yang sudah berbentuk json yang rapih:

```
[
  {
    "id_course": "CSC123",
    "name": "PSP",
    "credits": 4,
    "students": [
    ]
  },
  {
    "id_course": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": [
      {
        "npm": "12346",
        "name": "jotaro joestar",
        "gpa": 0.0,
        "courses": null
      },
      {
        "id_course": "CSC126",
        "name": "MPKT",
        "credits": 6,
        "students": [
          {
            "npm": "123",
            "name": "Killer Queen",
            "gpa": 0.0,
            "courses": null
          }
        ]
      }
    ]
  },
  {
    "id_course": "CSC125",
    "name": "DDP 1",
    "credits": 4,
    "students": [
    ]
  }
]
```

## Implementasi Consumer

### 1. Service Viewall

Berikut adalah kode yang saya buat di StudentServiceRest:

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

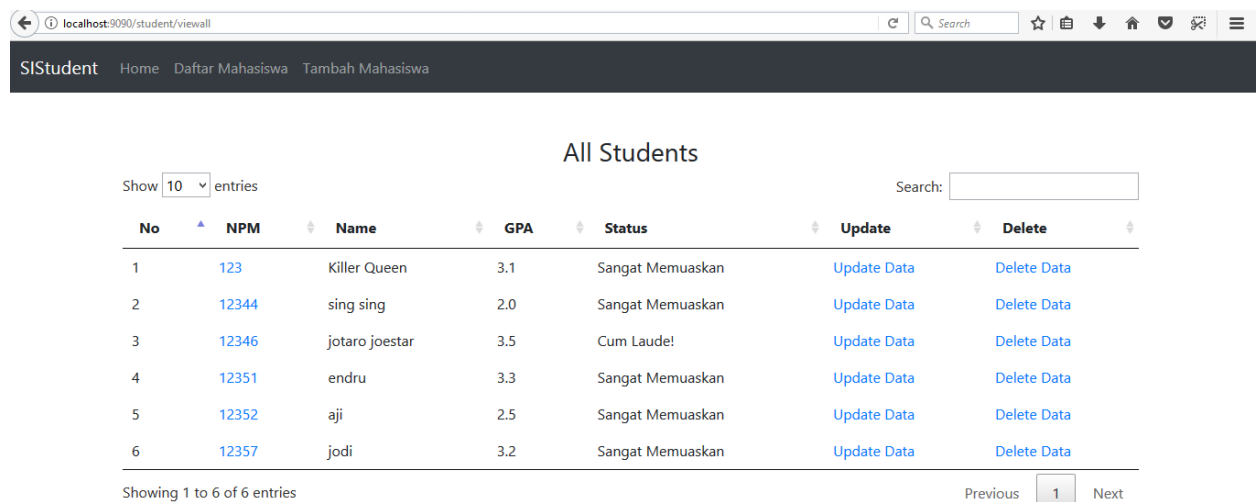
Berikut adalah kode yang saya buat di StudentDAOImpl:

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

    return students;
}
```

Karena sekarang yang kita inginkan adalah menampilkan list students, parameter yang perlu dimasukkan ke method getForObject adalah alamat untuk viewall dan class List. Return method ini adalah List StudentModel, sesuai dengan yang dibutuhkan dari controller dan pemanggilan dari serviceDAO.

Berikut tampilannya:



No	NPM	Name	GPA	Status	Update	Delete
1	123	Killer Queen	3.1	Sangat Memuaskan	<a href="#">Update Data</a>	<a href="#">Delete Data</a>
2	12344	sing sing	2.0	Sangat Memuaskan	<a href="#">Update Data</a>	<a href="#">Delete Data</a>
3	12346	jotaro joestar	3.5	Cum Laude!	<a href="#">Update Data</a>	<a href="#">Delete Data</a>
4	12351	endru	3.3	Sangat Memuaskan	<a href="#">Update Data</a>	<a href="#">Delete Data</a>
5	12352	aji	2.5	Sangat Memuaskan	<a href="#">Update Data</a>	<a href="#">Delete Data</a>
6	12357	jodi	3.2	Sangat Memuaskan	<a href="#">Update Data</a>	<a href="#">Delete Data</a>

## 2. Implementasi Service Course

Sama seperti StudentServiceRest, saya hanya perlu mengganti service yang di-implement, mengganti DAO yang diinstansiasi, dan mengganti method sesuai keperluan

Berikut adalah kode yang ada di CourseServiceRest

```
@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService
{
    @Autowired
    private CourseDAO courseDAO;

    @Override
    public CourseModel selectCourse(String id_course)
    {
        log.info("REST - select course with id {}", id_course);
        return courseDAO.selectCourse(id_course);
    }

    @Override
    public List<CourseModel> selectAllCourses()
    {
        log.info("REST - select all courses");
        return courseDAO.selectAllCourses();
    }
}
```

Berikut adalah kode yang ada di CourseDAOImpl

```
@Slf4j
@Service
public class CourseDAOImpl implements CourseDAO
{
    @Autowired
    private RestTemplate restTemplate;

    @Bean
    public RestTemplate restTemplate(RestTemplateBuilder builder)
    {
        return builder.build();
    }

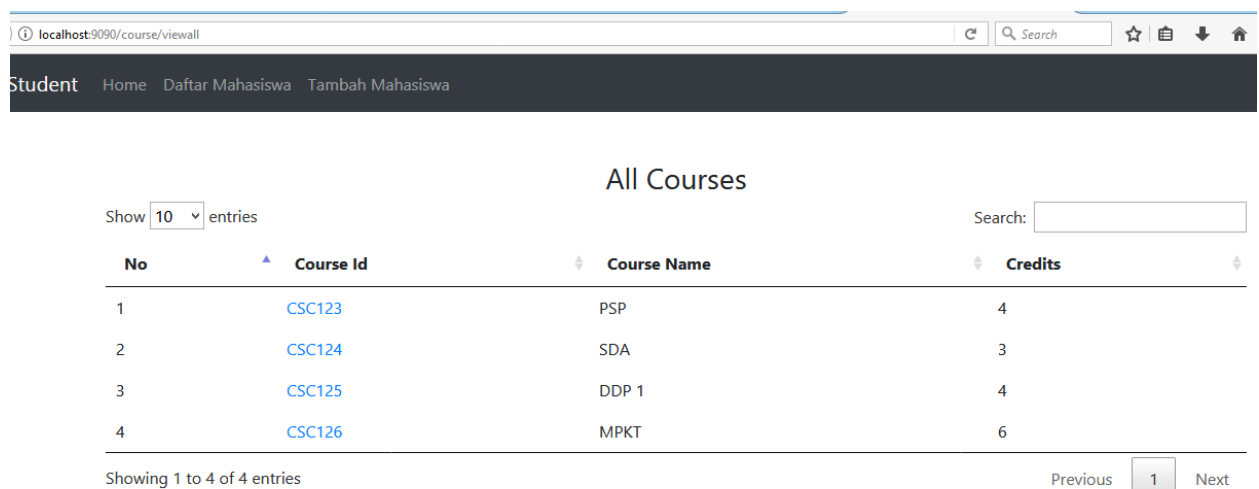
    @Override
    public CourseModel selectCourse (String id_course)
    {
        CourseModel course = restTemplate.getForObject(
            "http://localhost:8080/rest/course/view/"+ id_course,
            CourseModel.class);
    }
}
```

```
        return course;
    }

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

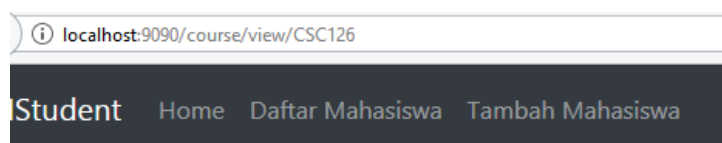
        return courses;
    }
}
```

### Tampilan View All Course



No	Course Id	Course Name	Credits
1	CSC123	PSP	4
2	CSC124	SDA	3
3	CSC125	DDP 1	4
4	CSC126	MPKT	6

### Tampilan View Course by Id



Student	Home	Daftar Mahasiswa	Tambah Mahasiswa
---------	------	------------------	------------------

ID = CSC126

Name = MPKT

Credits = 6

Siswa yang mengambil matkul ini

- 123 - Killer Queen