

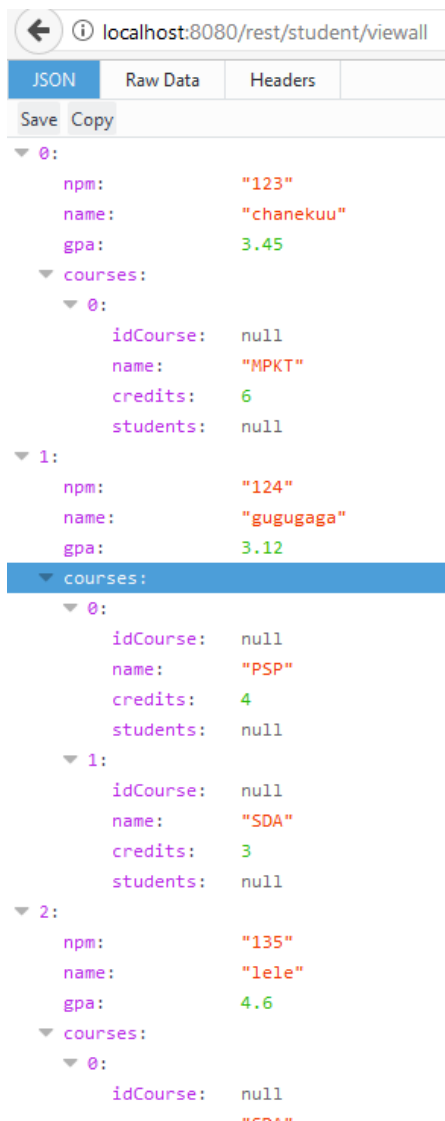
- Latihan 1.

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

Saya menambahkan method viewAll pada StudentRestController, method ini memanggil method selectAllStudent pada kelas StudentServiceDataBase, dan memanggil method pada mapper

```
@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();
```

Kemudian berikut adalah query akan mengembalikan list student dan akan muncul tampilan daftar student berupa JSON



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/rest/student/viewall'. Below the address bar, there are tabs for 'JSON', 'Raw Data', and 'Headers'. The 'JSON' tab is selected, and the response is displayed in a collapsible tree view. The response is a list of three student objects, each containing 'npm', 'name', 'gpa', and a list of 'courses'.

```
{
  "0": {
    "npm": "123",
    "name": "chanekuu",
    "gpa": 3.45,
    "courses": [
      {
        "idCourse": null,
        "name": "MPKT",
        "credits": 6,
        "students": null
      }
    ]
  },
  "1": {
    "npm": "124",
    "name": "gugugaga",
    "gpa": 3.12,
    "courses": [
      {
        "idCourse": null,
        "name": "PSP",
        "credits": 4,
        "students": null
      },
      {
        "idCourse": null,
        "name": "SDA",
        "credits": 3,
        "students": null
      }
    ]
  },
  "2": {
    "npm": "135",
    "name": "lele",
    "gpa": 4.6,
    "courses": [
      {
        "idCourse": null,
        "name": "PSP",
        "credits": 4,
        "students": null
      }
    ]
  }
}
```

- Latihan 2.

Berikut adalah CourseRestController, berisikan dua method, yakni viewCourse yang akan menampilkan detail course dan viewAllCourse yang akan menampilkan daftar course.

```
import java.util.List;

@RestController
@RequestMapping("/rest")
public class CourseRestController {

    @Autowired
    StudentService studentService;

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

    @RequestMapping("/course/viewallcourses")
    public List<CourseModel> viewAllCourses() {
        List<CourseModel> courses = studentService.selectAllCourses();
        return courses;
    }
}
```

Masing-masing method tersebut akan memanggil method pada kelas StudentServiceDatabase, dan pada kelas mapper akan menjalankan query berikut.

```
@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 = "selectStudentClass")) })
CourseModel selectCourse(@Param("id") String id);

@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")})
List<CourseModel> selectAllCourses();
```

Berikut adalah tampilan untuk detail course, berupa JSON



The screenshot shows a web browser window with the address bar displaying 'localhost:8080/rest/course/view/CSC125'. Below the address bar, there are tabs for 'JSON', 'Raw Data', and 'Headers'. The 'JSON' tab is selected, and the response is displayed as a JSON object. The response contains the following data:

```
{
  "idCourse": "CSC125",
  "name": "DDP 1",
  "credits": 4,
  "students": []
}
```

Berikut adalah tampilan untuk daftar seluruh course, berupa JSON



The screenshot shows a REST client interface with the URL `localhost:8080/rest/course/viewallcourses`. The response is displayed in JSON format, showing an array of four course objects. Each object contains the following fields: `idCourse`, `name`, `credits`, and `students`. The `students` field is `null` for all courses.

```
{
  "0": {
    "idCourse": "CSC123",
    "name": "PSP",
    "credits": 4,
    "students": null
  },
  "1": {
    "idCourse": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": null
  },
  "2": {
    "idCourse": "CSC125",
    "name": "DDP 1",
    "credits": 4,
    "students": null
  },
  "3": {
    "idCourse": "CSC126",
    "name": "MPKT",
    "credits": 6,
    "students": null
  }
}
```

- Latihan 3

Pada StudentDAO, tambahkan method selectAllStudents

```
StudentDAO.java
1 package com.example.demo.dao;
2
3 import java.util.List;
4
5
6
7 public interface StudentDAO {
8     StudentModel selectStudent(String npm);
9     List<StudentModel> selectAllStudents();
10 }
11
```

Kemudian lengkapi method selectAllStudents di StudentDAOImpl.java, method tersebut akan mengambil data dari Tutorial07Producer.java

```
package com.example.demo.service;

import java.util.List;

@Service
public class StudentDAOImpl implements StudentDAO {

    @Autowired
    private RestTemplate restTemplate;

    @Override
    public StudentModel selectStudent(String npm) {
        StudentModel student = restTemplate.getForObject("http://localhost:8080/rest/student/view/" + npm,
            StudentModel.class);
        return student;
    }

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

Kemudian, tambahkan method selectAllStudents pada kelas StudentServiceRest.java, yang mengambil data melalui StudentDAOImpl.java

```
import java.util.List;

@Slf4j
@Service
@Primary
public class StudentServiceRest implements StudentService {

    @Autowired
    private StudentDAO studentDAO;

    @Autowired
    private CourseDAO courseDAO;

    @Override
    public StudentModel selectStudent(String npm) {
        log.info("REST - select student with npm {}", npm);
        return studentDAO.selectStudent(npm);
    }

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

Berikut adalah view untuk detail student



NPM = 123

Name = chanek

GPA = 3.33

Kuliah yang diambil

- PSP-4 sks
- MPKT-6 sks

Dan berikut adalah tampilan untuk daftar seluruh student



All Students

No.	NPM	Nama	GPA	Cumlaude	Delete	Update
1	123	chanek	3.33	Sangat Memuaskan	Delete Data	Update Data
2	124	Chanek Jr.	3.9	Cumlaude	Delete Data	Update Data
3	125	adi	2.9	Sangat Memuaskan	Delete Data	Update Data

- Latihan 4

Buat interface CourseDAO

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

Buat CourseDAOImpl yang mengimplementasikan CourseDAO, method tersebut akan mengambil data dari Tutorial07Producer.java

```
package com.example.demo.service;

import java.util.List;

@Service
public class CourseDAOImpl implements CourseDAO{

    @Autowired
    private RestTemplate restTemplate;

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

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

Pada kelas StudentServiceRest tambahkan method berikut

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

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

Tambahkan method viewAllCourse di StudentController

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

    return "viewallcourses";
}
```

Berikut adalah tampilan detail course

  localhost:9090/course/view/CSC125

All Students

No. 1

Name = DDP 1

Credits = 4

Daftar peserta course

Berikut adalah tampilan daftar seluruh course

  localhost:9090/course/viewallcourses

All Courses

No. 1

PSP-4 sks

No. 2

SDA-3 sks

No. 3

DDP 1-4 sks

No. 4

MPKT-6 sks

Hal yang dipelajari:

- Consumer akan mengambil data dari producer
- Penggunaan REST akan mengembalikan data berbentuk JSON.