

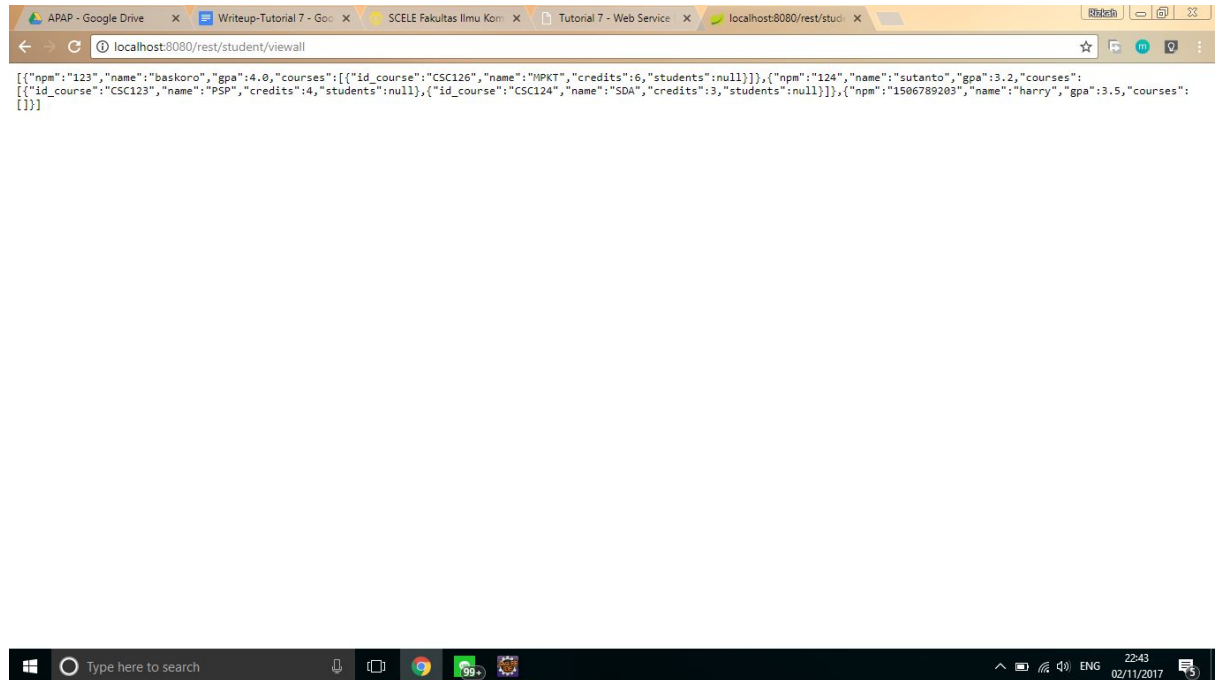
Rizkah Shalihah  
1506689641  
Kelas - B

## WRITEUP TUTORIAL 7

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”.

Jawaban:

**Output**



Setelah di parsing:

```
String parse

[
  {
    {
      "npm": "123",
      "name": "baskoro",
      "gpa": 4.0,
      "courses": [
        {
          "id_course": "CSC126",
          "name": "MPKT",
          "credits": 6,
          "students": null
        }
      ]
    },
    {
      "npm": "124",
      "name": "sutanto",
      "gpa": 3.2,
      "courses": [
        {
          "id_course": "CSC123",
          "name": "PSP",
          "credits": 4,
          "students": null
        },
        {
          "id_course": "CSC124",
          "name": "SDA",
          "credits": 3,
          "students": null
        }
      ]
    }
  ],
  {
    "npm": "1506789203",
    "name": "harry",
    "gpa": 3.5,
    "courses": [
    ]
  }
]

students : null
}
```

### Method di RestController:

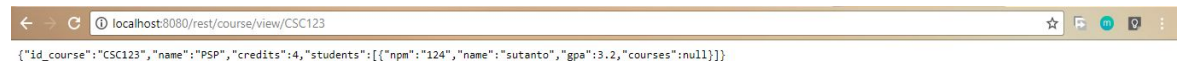
```
@RequestMapping("/student/viewall")
public List<StudentModel> viewall () {
    List<StudentModel> student = studentService.selectAllStudents();
    return student;
}
```

Penjelasan: saya mengambil semua database pada tabel student dan menyimpannya dalam bentuk list karena data yang ada lebih dari satu. Sehingga method yang saya buat juga dalam bentuk list dan return list object student.

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).

- a. View Course by id

### Output:



```
{\"id_course\": \"CSC123\", \"name\": \"PSP\", \"credits\": 4, \"students\": [{\"npm\": \"124\", \"name\": \"sutanto\", \"gpa\": 3.2, \"courses\": null}]}
```

### Hasil Parsing:

String parse

```
{
  \"id_course\": \"CSC123\",
  \"name\": \"PSP\",
  \"credits\": 4,
  \"students\": [
    {
      \"npm\": \"124\",
      \"name\": \"sutanto\",
      \"gpa\": 3.2,
      \"courses\": null
    }
  ]
}
```

### Method RestController:

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

Penjelasan: idnya sama seperti mengambil student pada database, jadi studentService memanggil method selectCourseStudent di service dari database sesuai id yang dimasukkan. Lalu return object course.

- b. View AllCourse

### Output:

```
localhost:8080/rest/course/viewall
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "sutanto", "gpa": 3.2, "courses": null}], {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "sutanto", "gpa": 3.2, "courses": null}], {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": [{"npm": "123", "name": "baskoro", "gpa": 4.0, "courses": null}]}
```

### Hasil Parsing:

String parse

```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": [{"npm": "124", "name": "sutanto", "gpa": 3.2, "courses": null}], {"id_course": "CSC124", "name": "SDA", "credits": 3, "students": [{"npm": "124", "name": "sutanto", "gpa": 3.2, "courses": null}], {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": [{"npm": "123", "name": "baskoro", "gpa": 4.0, "courses": null}]}
```

### Method RestController:

```
@RequestMapping("/course/viewall")
public List<CourseModel> viewAllCourse (Model model) {
    List<CourseModel> course = studentService.selectAllCourse();
    return course;
}
```

Penjelasan: saya mengambil semua database pada tabel course dan menyimpannya dalam bentuk list karena data yang ada lebih dari satu. Sehingga method yang saya buat juga dalam bentuk list dan return list object course.

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

### Output:



## All Students

**No. 1**

**NPM = 123**

**Name = baskoro**

**GPA = 4.0**

**Kuliah yang diambil**

- MPKT-6 sks

[Delete Data](#)

[Update Data](#)

---

**No. 2**

**NPM = 124**

**Name = sutanto**

**GPA = 3.2**

**Kuliah yang diambil**

- PSP-4 sks
- SDA-3 sks

[Delete Data](#)

---

**Method pada StudentDAOImpl:**

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

Penjelasan: menggunakan kelas RestTemplate untuk meng-consume object REST web service. Method yang digunakan adalah getForObject yang menerima parameter berupa URL producer web service dan tipe class dari object yang didapatkan. Pada kasus ini URL yang digunakan adalah <http://localhost:8080/rest/student/viewall>, yang telah saya buat di latihan sebelumnya.

Sedangkan tipe class object yang saya gunakan adalah List karena saya akan mengambil hasil json tersebut dalam bentuk list. Dan kembali me-return object student dalam bentuk list.

4. **Latihan 4:** Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.
- a. select course by id

**Output:**



**Method pada CourseDAOImpl:**

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

Penjelasan:

Mengambil json yang dihasilkan di url yang dimasukkan. Dan method selectCourseStudent akan dipanggil di controller dan seperti biasa menampilkannya dalam html.

- b. select all course

**Output:**



## All Course

**No. 1**

**Id = CSC123**

**Name = PSP**

**Credit = 4**

**Mahasiswa yang Mengambil**

- 124-sutanto

**No. 2**

**Id = CSC124**

**Name = SDA**

**Credit = 3**

**Mahasiswa yang Mengambil**

- 124-sutanto

**Method pada CourseDAOImpl:**

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

Penjelasan: menggunakan kelas RestTemplate untuk meng-consume object REST web service. Method yang digunakan adalah getForObject yang menerima parameter berupa URL producer web service dan tipe class dari object yang didapatkan. Pada kasus ini URL yang digunakan adalah <http://localhost:8080/rest/course/viewall>, yang telah saya buat di latihan sebelumnya. Sedangkang tipe class object yang saya gunakan adalah List karena saya akan mengambil hasil json tersebut dalam bentuk list. Dan

Rizkah Shalihah

1506689641

Kelas - B

kembali me-return object student dalam bentuk list. Setelah itu seperti biasa method ini akan dipanggil di controller dan menampilkannya ke html.