

Nama : Rinjani Samosir  
NPM : 1606954994  
Kelas : Ekstensi

## Tutorial 7

### Membuat Web Service Menggunakan Spring Boot Framework

#### Yang Dipelajari Pada Tutorial 7

Pada tutorial 7 ini dipelajari mengenai web service dengan cara pemisahan layer back-end (service producer) dan front-end (service consumer). Web service adalah URL yang akan mengambil data dalam representasi yang sudah disepakati seperti JSON dan XML. Untuk membuatnya pertama kali dibuat dua folder untuk producer dan consumer. Service producer akan diberikan RequestMapping dengan /rest yang artinya pemanggilannya selalu diawali dengan /rest. Sedangkan pada service consumer akan diberikan port 9090 pada application.properties yang berarti akan dijalankan di localhost:9090. Dalam pengimplementasiannya ditambahkan interface StudentDAO dan kelas StudentDAOImpl untuk mengimplementasikan interface tersebut. Pada kelas StudentDAOImpl digunakan kelas RestTemplate untuk meng-consume object REST web service. Dalam hal ini tidak lagi dibutuhkan studentMapper karena sudah digantikan oleh StudentDAO. Untuk menghandle-nya digunakan anotasi @Primary. Autowired secara otomatis akan menginisiasi StudentServiceRest karena sudah primary.

#### 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;
}
```

Method ini dibuat di StudentRestController.java. Pemanggilan viewall dapat diakses melalui /rest/student/viewall. Method viewall akan mengembalikan object yang ada pada service yaitu selectAllStudents().

2. **Latihan 2: Implementasikan service consumer untuk view all Students dengan melengkapi method selectAllStudents yang ada di kelas StudentServiceRest.**

Di StudentDaoImp ditambahkan method SelectAllStudents seperti dibawah ini.

```
@Override
public List<StudentModel> selectAllStudents ()
{
    ResponseEntity<StudentModel[]> students =
        restTemplate.getForEntity("http://localhost:8080/rest/student/viewall",
            StudentModel[].class);
    return Arrays.asList(students.getBody());
}
```

Method tersebut akan menampilkan seluruh mahasiswa. ResponseEntity akan merepresentasikan respon HTTP. RestTemplate akan consume object Rest pada StudentService. Method yang digunakan adalah getForEntity untuk merespon header agar mengembalikan status yang diinginkan. Method tersebut akan mengembalikan data seluruh mahasiswa berupa array.