

## Rangkuman Tutorial 7

Pelajaran yang dapat dipelajari dari tutorial 7 ini adalah pengertian tentang *web service* dan bagaimana membangun *web service* dengan Spring. *Web service* memungkinkan integrasi antar aplikasi yang tersebar di banyak *server*. Dengan adanya *web service* antar aplikasi satu dengan yang lainnya dapat saling komunikasi atau bertukar data. Dalam pengembangan aplikasi *web service* ada dua konsep pemisahan level aplikasi yaitu *service consumer* dan *service producer*. Hal ini seperti pengembangan pada aplikasi biasa dimana ada *backend* dan *frontend* untuk memudahkan bagi pengembang dalam mengembangkan aplikasi dan fokus pada layanan masing-masing level. *Service consumer (frontend)* ini merupakan aplikasi yang berinteraksi dengan pengguna. *Service producer (backend)* ini merupakan sebuah aplikasi yang menyediakan data untuk *service consumer*.

### Latihan 1

Membuat REST method yang mengeluarkan semua data *student*.

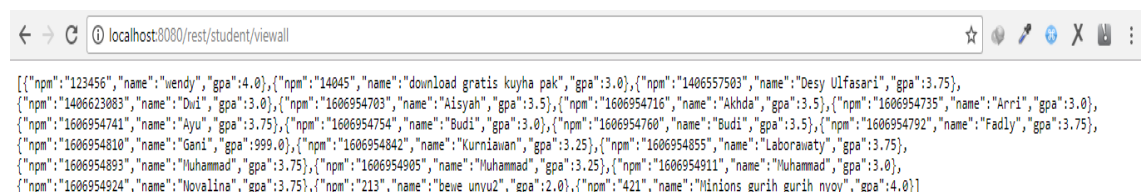
```
@RestController
@RequestMapping("/rest")
public class StudentRestController {

    @Autowired
    StudentService studentService;

    public StudentModel view(@PathVariable(value = "npm") String npm) {

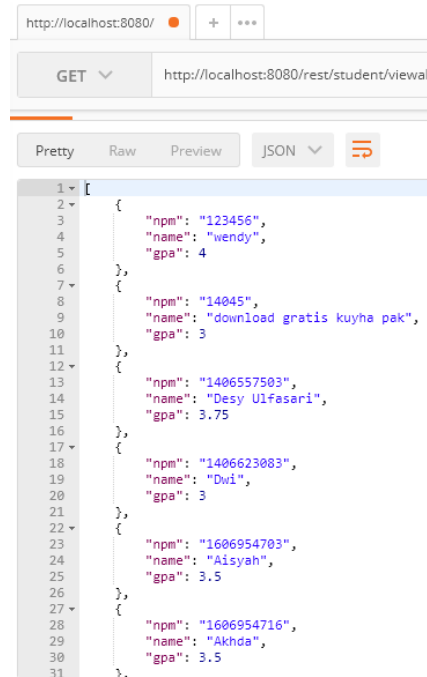
        @RequestMapping("/student/viewall")
        public List<StudentModel> viewall() {
            List<StudentModel> students = studentService.selectAllStudents();
            return students;
        }
    }
}
```

Pembuatan method `viewAll()` ini sama dengan yang ada pada *class* `StudentController`, bedanya hanya pada *method* ini me-*return* data objek `List<StudentModel>` dan bukan me-*return* `String` seperti yang ada di *class* `StudentController`. Di *class* `StudentRestController` juga memiliki perbedaan dengan yang ada di *class* `StudentController` yaitu dengan mengganti anotasi `@Controller` dengan `@RestController` dan penambahan anotasi `@RequestMapping("/rest")`. `@RestController` ini menandakan bahwa *class* tersebut untuk menjalankan REST, dan `@RequestMapping("/rest")` ini menandakan bahwa *class* tersebut diakses pada URL `"/rest"`.



```
[{"npm": "123456", "name": "wendy", "gpa": 4.0}, {"npm": "14045", "name": "download gratis kuyha pak", "gpa": 3.0}, {"npm": "1406557503", "name": "Desy Ulfasari", "gpa": 3.75}, {"npm": "1406623083", "name": "Dwi", "gpa": 3.0}, {"npm": "1606954703", "name": "Aisyah", "gpa": 3.5}, {"npm": "1606954716", "name": "Akhda", "gpa": 3.5}, {"npm": "1606954735", "name": "Arri", "gpa": 3.0}, {"npm": "1606954741", "name": "Ayu", "gpa": 3.75}, {"npm": "1606954754", "name": "Budi", "gpa": 3.0}, {"npm": "1606954760", "name": "Budi", "gpa": 3.5}, {"npm": "1606954792", "name": "Fadly", "gpa": 3.75}, {"npm": "1606954810", "name": "Gani", "gpa": 999.0}, {"npm": "1606954842", "name": "Kurniawan", "gpa": 3.25}, {"npm": "1606954855", "name": "Laborawaty", "gpa": 3.75}, {"npm": "1606954893", "name": "Muhammad", "gpa": 3.75}, {"npm": "1606954905", "name": "Muhammad", "gpa": 3.25}, {"npm": "1606954911", "name": "Muhammad", "gpa": 3.0}, {"npm": "1606954924", "name": "Novalina", "gpa": 3.75}, {"npm": "213", "name": "bewe unyu2", "gpa": 2.0}, {"npm": "421", "name": "Minions gurih gurih nyoy", "gpa": 4.0}]
```

Berikut adalah hasil ketika membuka halaman `localhost:8080/rest/student/viewall` dan halaman tersebut mengembalikan data semua *student* dalam bentuk JSON.



Ini hasil JSON yang dirapihkan dengan aplikasi PostMan.

## Latihan 2

Melengkapi method `selectAllStudents()` yang ada di `class StudentServiceRest`.

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

Sebelum mengimplementasikan method `selectAllStudents()` di `class StudentServiceRest`, pertama adalah implementasi kode pada method `selectAllStudents()` tersebut yang ada pada `class StudentDAOImpl`. Di dalam method tersebut dimana akan me-*return* objek berupa `List<StudentModel>` hal ini berbeda dengan implementasi kode pada method `selectStudent()`. Untuk mendapatkan objek berupa *list* maka disini akan menggunakan `class ResponseEntity` dengan tipe data generiknya `StudentModel[]` (Array). `Class` ini berfungsi untuk membuat dan menampung respon data dari rest template. Method pada `restTemplate` yang digunakan adalah `getForEntity()` dimana method tersebut akan me-*return* objek `ResponseEntity`. Setelah itu `re` dari objek `ResponseEntity` akan menyimpan array dengan data `StudentModel`. Variable `re` akan diiterasikan untuk dimasukkan ke dalam `List` untuk dijadikan *return* method `selectAllStudents()`.

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

Setelah kode method `selectAllStudents()` yang ada di *class* `StudentDAOImpl` sudah di implementasikan, selanjutnya ada mengimplementasikan method tersebut pada *class* `StudentServiceRest` di method `selectAllStudents()` (dapat dilihat di gambar).

```

Tutorial07Consumer - Tutorial7Consumer [Spring Boot App] C:\Program Files (x86)\Java\jdk1.8.0_161\bin\javaw.exe (Apr 14, 2018, 5:57:17 PM)
s.j.e.a.AnnotationMethodParameter : registering beans for java exposure on startup
b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 9090 (http)
m.example.Tutorial7Consumer : Started Tutorial7Consumer in 6.732 seconds (JVM running for 1368
a.c.c.c.[Tomcat].[localhost].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
s.web.servlet.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed i
m.example.service.StudentServiceRest : REST - select all students
  
```

Pada hasil di *console* berikut adalah hasil dimana “REST – select all student” merupakan log yang mepresentasikan pengambilan data semua *student* dari aplikasi *service producer* dengan URL `localhost:8080/rest/student/viewall`.

No.	NPM	Name	GPA	Status	Options
1	123456	wendy	4.0	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
2	14045	download gratis kuyha pak	3.0	Sangat Memuaskan!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
3	1406557503	Desy Ulfasari	3.75	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
4	1406623083	Dwi	3.0	Sangat Memuaskan!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
5	1606954703	Aisyah	3.5	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
6	1606954716	Akhda	3.5	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
7	1606954735	Arri	3.0	Sangat Memuaskan!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
8	1606954741	Ayu	3.75	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
9	1606954754	Budi	3.0	Sangat Memuaskan!	<a href="#">Delete Data</a> <a href="#">Update Data</a>
10	1606954760	Budi	3.5	Cum Laude!	<a href="#">Delete Data</a> <a href="#">Update Data</a>

Berikut merupakan hasil tampilan yang berhasil mengambil data semua *student* di aplikasi *service producer*.