

1. Ringkasan Materi

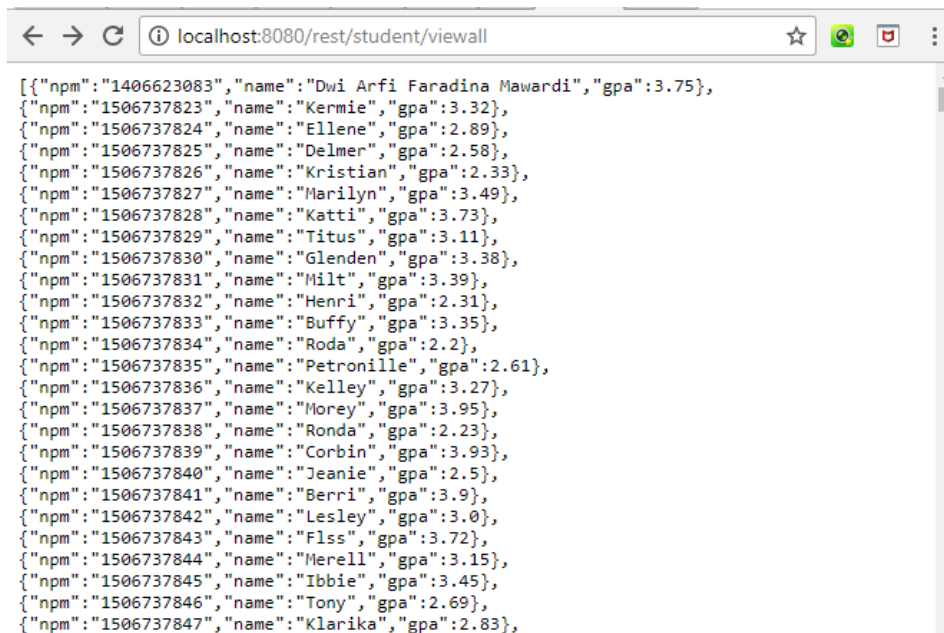
Pada kali tutorial 7 kali ini dipelajari membuat suatu Web Service dengan menggunakan REST dan bagaimana cara mengambil data hasil output dari Web Service yang dihasilkan. Pada kali tutorial ini dibuat 2 buah aplikasi atau project yang mempunyai fungsi yang berbeda. Project yang satu akan berperan sebagai back-end layer sedangkan project yang lain akan berperan sebagai front-end yang akan berkomunikasi dengan user. Untuk dapat mengintegrasikan kedua layer tersebut maka harus dibuatlah suatu service contract yang jelas terkait apa yang dibutuhkan oleh front-end sehingga back-end dapat membuat web service dengan output sesuai yang dibutuhkan oleh front-end tersebut atau sebaliknya. Pada tutorial ini digunakan JSON untuk output dari web service yang nantinya akan digunakan oleh front-end untuk dapat diolah dan ditampilkan ke user.

2. Penjelasan Method Latihan

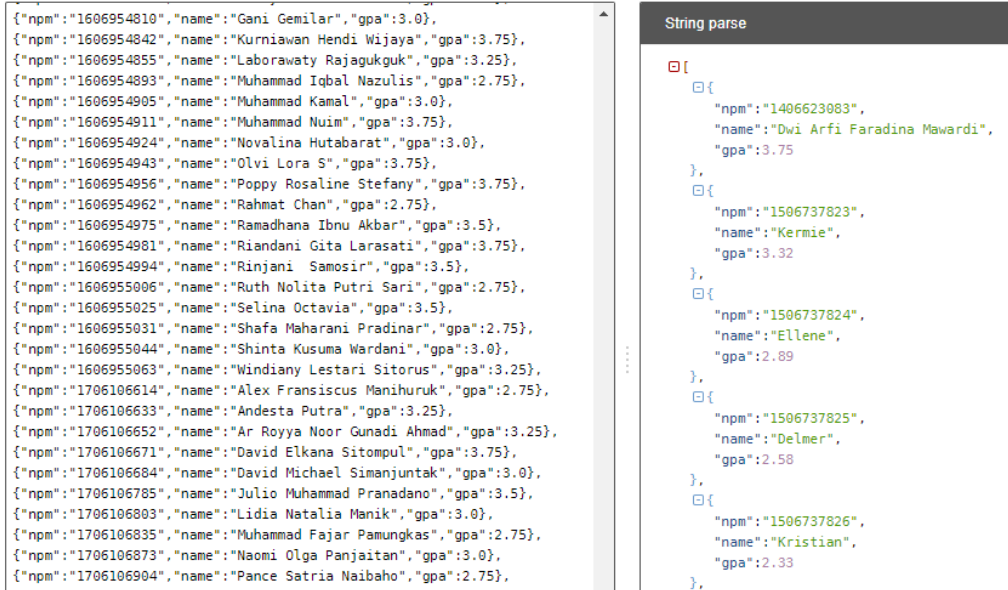
1) Latihan 1

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

Membuat RestController, pada awal penulisan nama class terdapat anotasi @RestController yang mendeskripsikan tipe controller tersebut. Kemudian untuk dapat membuat suatu web service untuk kebutuhan view all maka perlu dibuat controller method viewall seperti yang tertampil pada gambar di atas. Method ini akan mengembalikan List untuk object Student Model. List object Student Model didapatkan dari studentService yang memanggil mapper untuk dapat mendapatkan data dari database.



Gambar di atas merupakan hasil dari pemanggilan Web Service untuk method viewall dimana berupa JSON. Seluruh data student tertampil dengan format JSON.



Untuk dapat memudahkan JSON maka dapat dilihat pada gambar atas kanan.

2) Tutorial 2

```
@Override
public List<StudentModel> selectAllStudents ()
{
    ResponseEntity<List<StudentModel>> studentResponse =
        restTemplate.exchange("http://localhost:8080/rest/student/viewall",
                               HttpMethod.GET, null, new ParameterizedTypeReference<List<StudentModel>>()) {
    });
    List<StudentModel> student = studentResponse.getBody();
    return student;
}
```

Membuat implementasi dari interface StudentDAO pada StudentImpl untuk method selectAllStudent. Pada method ini akan mengembalikan List dari object Student Model dimana list tersebut di dapatkan dari web service yang telah dibuat pada latihan 1. Method ini akan mengakses url yang telah dibuat pada latihan 1. Pertama tama dibuat sebuah variable yang akan menerima output dari web service terlebih dahulu. Response Entity digunakan untuk menerima return dari RestTemplate dengan method getForEntity() dan exchange(). Pada latihan ini digunakan method exchange() dimana parameter yang dibutuhkan untuk dapat menerima output dari web service yaitu url (<http://localhost:8080/rest/student/viewall> → url untuk latihan 1 yang mengembalikan list seluruh student dengan format JSON), http merhod (method yang digunakan untuk mengakses url → digunakan mehod GET), type untuk return value (null), uriVariables (variable pada templete). Untuk uriVariables diguanakan class PrameterizedTypeReference digunakan untuk bisa mendapatkan generic ResponseEntity<T> dimana T adalah sebuah generic class. Disini T adalah sebuah List dengan object Student Model sesuai dengan tipe variable studentResponse. Untuk bisa mendapatkan tipe data List dengan object Student Mahasiswa yang akan digunakan untuk return method selectStudent maka perlu dipanggil body dari studentResponse.

```

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

```

Pada StudentServiceRest yang akan dipanggil oleh controller dibuat method untuk memanggil seluruh student yaitu selectAllStudent. Pada method ini cukup memanggil method selectAllStudent pada Interface StudentDAO yang telah dibuat sebelumnya pada StudentDAOImpl.

The screenshot shows a web browser at localhost:9090/student/viewall. The page has a dark navbar with the title 'All Students Table'. Below the navbar, there is a search bar and a 'Show 10 entries' dropdown. The table displays 6 students with their NPM, Name, GPA, and Cum Laude status. Each row has a 'Delete' button and an 'Update' button.

No	NPM	Name	GPA	Cum Laude		
1	1406623083	Dwi Arfi Faradina Mawardi	3.75	Cum Laude!	Delete	Update
2	1506737823	Kermie	3.32	Sangat Memuaskan!	Delete	Update
3	1506737824	Ellene	2.89	Sangat Memuaskan!	Delete	Update
4	1506737825	Delmer	2.58	Sangat Memuaskan!	Delete	Update
5	1506737826	Kristian	2.33	Sangat Memuaskan!	Delete	Update
6	1506737827	Marilyn	3.49	Cum Laude!	Delete	Update

Berikut hasil dari pemanggilan StudentServiceRest method selectAllStudent.