Nama : Haryo Parigroho
NPM : 1506689603

## Membuat *Service Producer*

Latihan 1 : Membuat *method* ***viewAllStudent()*** pada *Class StudentRestController*

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

Method ini memanggil method selectAllStudent() pada studentService.

Latihan 2 : Membuat method view(String id) dan viewAllCourse() pada Class CourseRestController

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

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

Method view menangani mapping (/rest/course/view/{id}) dan mengembalikan objek courseModel dengan id yang ditentukan.

Method viewAllCourse menangani mapping (/rest/course/viewall) dan mengembalikan objek list of courseModel.

## Membuat *Service Consumer*

Latihan 3:  Implementasi method selectAllStudent pada StudentServiceRest

Method selectAllStudent pada StudentServiceRest hanya memanggil selectAllStudent() pada studentDAO, sedangkan pada method selectAllStudent() pada class StudentDAOImpl adalah seperti berikut;

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

Method ini memanggil method exchange dari RestTemplate dengan url http://localhost:8080/rest/student/viewall/ dan kembaliannya adalah list<StudentModel> lalu untuk mendapatkan list<StudentModel> menggunakan method getBody().

Latihan 4: Implementasi service consumer untuk class CourseModel

- Membuat CourseDAO pada package dao.
- Membuat CourseDAOImpl pada package service untuk implementasi DAO, class ini memiliki 2 method;

```
@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> selectAllCourse() {
    ResponseEntity<List<CourseModel>> response = restTemplate.exchange("http://localhost:8080/rest/course/viewall/", HttpMethod.
            new ParameterizedTypeReference<List<CourseModel>>() {
    });
    List<CourseModel> courses = response.getBody();
    return courses;
}
```

  Method selectCourse berfungsi untuk mendapatkan sebuah object CourseModel dari service dengan link /rest/course/view/id.
  Method selectAllCourse berfungsi untuk mendapatkan list of object CourseModel dari service dengan link /rest/course/viewall.
- Membuat CourseServiceRest, class ini mengimplementasi CoureService dan memiliki 2 method;

```
@Override
public CourseModel selectCourse(String id) {
    return courseDAO.selectCourse(id);
}

@Override
public List<CourseModel> selectAllCourse() {
    return courseDAO.selectAllCourse();
}
```

  Method selectCourse(id) akan memanggil fungsi selectCourse pada CourseDAO yang sudah diimplementasi pada CourseDAOImpl untuk mendapatkan Course dengan id tertentu.
  Method selectAllCourse() akan memanggil fungsi selectAllCourse() pada CourseDAO untuk mendapatkan List of Object CourseModel.

## Hal yang Saya Pelajari dari Tutorial 7

- Membuat RESTful Web Service menggunakan Spring dengan annotation @RestController dan @RequestMapping("/rest") untuk mendapatkan JSON nya.
- Memanggil / menggunakan REST Web Service dengan RestTemplate.
- Menjalankan aplikasi pada port yang berbeda.

- Menggunakan method getForObject untuk mendapatkan satu object dari service dan method exchange untuk mendapatkan List of Object.