

Nama : Adil Krisnadi Pradana
NPM : 1406623631
Kelas : APAP-C

TUTORIAL 7

Lesson Learned

Pada tutorial 7 kali ini, kita mempelajari tentang penggunaan *Web Service*. *Web Service* sendiri adalah merupakan sebuah URL yang akan mengembalikan data dalam representasi seperti JSON atau XML, di mana pada tutorial 7 kali ini data yang dikembalikan berbentuk JSON. Implementasi *Web Service* pada tutorial 7 kali ini dilakukan dengan melakukan pemisahan *layer back-end* pada *service producer* (yang membutuhkan data-data dari tutorial 5) dan *layer front-end* pada *service consumer* (yang membutuhkan data-data dari tutorial 6). Di mana *service consumer* merupakan aplikasi yang berinteraksi dengan *user*, sementara *service producer* merupakan aplikasi yang memberikan data kepada *service consumer* berdasarkan apa yang *service consumer* minta. Dengan demikian, *service consumer* dapat menyediakan dan mengolah data ke *user* tanpa perlu memiliki *database*. Sementara *service producer* hanya menyediakan data dari *database* dan biasanya tidak memiliki *view* yang dapat dilihat *user*.

Latihan 1

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

Method ini berfungsi untuk mengembalikan daftar dari objek *student* yang ada di *database*, dengan cara memanggil *method* `selectAllStudents`, lalu akan dikembalikan dalam bentuk objek `List`. Saat ditampilkan, objek tersebut akan dikembalikan dalam format JSON.

Latihan 2

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

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

Method yang pertama berfungsi untuk mengembalikan objek *course* dengan ID *course* tertentu yang berada di *database*, dengan cara memanggil *method* `selectCourse` dan dikembalikan dalam bentuk objek `CourseModel`, sebelum ditampilkan dengan format JSON. Sedangkan *method* yang kedua berfungsi untuk mengembalikan seluruh objek *course* yang terdapat di *database*, dengan cara

memanggil *method* `selectAllCourse`, lalu objek tersebut akan dikembalikan dalam bentuk objek `List` sebelum ditampilkan dengan format JSON.

Latihan 3

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

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

Method yang pertama berada di *class* `StudentServiceRest.java`, di mana *method* tersebut berfungsi untuk mengembalikan daftar dari objek *student* yang akan diambil dari *class* `StudentDAO` dengan cara memanggil *method* `selectAllStudents`. Sementara *method* yang kedua berada di *class* `StudentDAOImpl.java`, di mana *method* tersebut berfungsi untuk mengambil daftar objek *student* yang berada di halaman *service producer* yakni pada halaman `"/rest/student/viewall"`, sebelum nantinya objek tersebut akan ditampilkan di halaman *service consumer*.

Latihan 4

```
@Override
public CourseModel selectCourse(String id_course) {
    Log.info("REST - select course with id_course {}", id_course);
    return courseDAO.selectCourse(id_course);
}

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

Method yang pertama berada di *class* `CourseServiceRest.java`, di mana *method* tersebut berfungsi untuk mengembalikan daftar dari objek *course* dengan ID *course* tertentu yang akan diambil dari *class* `CourseDAO` dengan cara memanggil *method* `selectCourse`. Sementara *method* yang kedua berada di *class* `CourseDAOImpl.java`, di mana *method* tersebut berfungsi untuk mengambil daftar objek *course* yang berada di halaman *service producer* yakni pada halaman `"/rest/course/view"`, sebelum nantinya objek tersebut akan ditampilkan di halaman *service consumer*.