### 1. Membuat Service Producer

```
Tutorial07Consumer [boot] [devtools

    Tutorial07Producer [boot] [devtools]

                                     Spring Elements
                                       package com.example.rest;
   # src/main/java
     30 import org.springframework.beans.factory.annotation.Autowired;
                                      10
        > 15 Tutorial07ProducerApplica
                                      11 @RestController
     @RequestMapping("/rest")
                                      12
                                      13 public class StudentRestController {
        > 1 CourseController.java
                                      14⊜
                                            @Autowired
        StudentController.java
                                            StudentService studentService;
                                      15
                                      16

\[
\text{\frac{1}{18}} \] com.example.dao

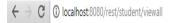
                                      17⊜
                                            @RequestMapping("/student/view/{npm}")
        > II CourseMapper.java
                                            public StudentModel view (@PathVariable(value = "npm") String npm) {
                                      18
                                               StudentModel student = studentService.selectStudent (npm);
                                      19
        20
                                               return student;
     21
                                     22 }
        > / CourseModel.java
        StudentModel.java
     > 5 StudentRestController.java
     > M CourseService.java
        > 🔑 CourseServiceDatabase.jav
        > J StudentService.java
        StudentServiceDatabase.ja
```

### Tampilan ketika membuka localhost:8080/rest/student/view/123

# Tampilan menggunakan JSON View

### **LATIHAN**

1. Viewall students dengan course yang diambil



[{"npm":"123","name":"Chanek","gpa":3.6,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"Chanek Jr","gpa":3.4,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}]},{"npm":"128","name":"hehe","gpa":3.25,"courses":[]}]

```
JS eval
String parse
                                                                ∃ [
□ [
   □ {
                                                                   ⊟ {
      "npm":"123",
                                                                      "npm":"123",
      "name": "Chanek",
                                                                      "name": "Chanek",
      "gpa":3.6,
                                                                      "gpa":3.6,
      "courses": 🖯 [
                                                                      "courses": 🖯 [
         □ {
                                                                          □ {
             "idCourse": null,
                                                                             "idCourse": null,
             "name": "MPKT",
                                                                             "name": "MPKT",
             "credits":6,
                                                                             "credits":6,
             "students": null
                                                                             "students": null
   },
                                                                   },
   □ {
                                                                   □ {
      "npm": "124",
                                                                      "npm": "124",
      "name": "Chanek Jr",
                                                                      "name": "Chanek Jr",
      "gpa":3.4,
                                                                      "gpa":3.4,
      "courses": 🗖 [
                                                                      "courses":□[
                                                                          ⊟ {
         Θ{
             "idCourse": null,
                                                                             "idCourse": null,
            "name":"PSP",
                                                                             "name":"PSP",
            "credits":4,
                                                                             "credits":4,
             "students":null
                                                                             "students":null
         □ {
                                                                          □ {
            "idCourse": null,
                                                                             "idCourse": null,
            "name":"SDA",
                                                                             "name":"SDA",
            "credits":3,
                                                                             "credits":3,
            "students": null
                                                                             "students":null
```

```
1 package com.example.rest;
    3⊕ import java.util.List;
   12
   13 @RestController
   14 @RequestMapping("/rest")
   15 public class StudentRestController {
    16<sup>-</sup>
           @Autowired
    17
           StudentService studentService;
   18
   19⊕
           @RequestMapping("/student/view/{npm}")
    20
           public StudentModel view (@PathVariable(value = "npm") String npm) {
    21
               StudentModel student = studentService.selectStudent (npm);
    22
               return student:
    23
           }
    24
    25⊜
           @Autowired
    26
           @RequestMapping("/student/viewall")
    27
           public List<StudentModel> viewAll () {
   28
               List StudentModel> students = studentService.selectAllStudents();
    29
               return students;
   30
           }
   31 }
StudentRestController.java
                        🖹 viewall.html 💢
 1 <!DOCTYPE html>
 20 <html xmlns:th="http://www.thymeleaf.org">
 3⊜
4
           <title>View All Students</title>
 5
       </head>
       <body>
 6⊕
7
           <h1>All Students</h1>
8
90
           <div th:each="student,iterationStatus: ${students}">
              <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
10
              <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
              <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
12
              <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
13
               <a th:href="'/student/delete/' + ${student.npm}"> Delete Data </a> <br/>
14
15
              <a th:href="'/student/update/' + ${student.npm}"> Update Data </a> <br/> <br/> 
16
17
              <h3>Kuliah yang diambil</h3>
                  18⊜
                      th:text="${course.name} + '-' + ${course.credits} + ' sks'">
19⊜
20
                          Nama Mata Kuliah - Jumlah SKS
21
                      22
```

Dnegan mengguankan method selectStudent yang telah digunakan pada tutorial sebelumnya kemudian kembaliannya berupa arraylist dari mahasiswa itu sendiri.

# 2. Service untuk kelas course dengan controller baru

```
← → C ① localhost:8080/rest/course/view/CSC126

{"idCourse":"CSC126", "name":"MPKT", "credits":6, "students":[{"npm":"123", "name":"Chanek", "gpa":3.6, "courses":null}]}
```

```
String parse
                                                                 JS eval
   "idCourse": "CSC126",
                                                                    "idCourse": "CSC126",
   "name": "MPKT",
                                                                     "name": "MPKT",
   "credits":6,
                                                                     "credits":6,
   "students": 🖯 [
                                                                    "students": ☐ [
      □ {
                                                                        □ {
          "npm": "123",
                                                                           "npm": "123",
         "name": "Chanek",
                                                                           "name": "Chanek",
         "gpa":3.6,
                                                                           "gpa":3.6,
         "courses":null
                                                                           "courses":null
  1
                                                                    1
```

```
view-course.html
                    🔎 CourseRestController.java 🔀
  package com.example.rest;
🔏 3⊕ import java.util.List;[.
 14
 15 @RestController
 16 @RequestMapping("/rest")
 17 public class CourseRestController
 18 {
 19⊝
         @Autowired
 20
         CourseService courseService;
 21
 22⊝
         @RequestMapping("/course/view/{id}")
 23
         public CourseModel view (@PathVariable(value = "id") String id) {
 24
            CourseModel course = courseService.selectCourse(id);
 25
             return course;
 26
         }
 27
 28⊜
         @Autowired
 29
         @RequestMapping("/course/viewall")
 30
         public List<CourseModel> viewAll () {
            List<CourseModel> courses = courseService.selectAllCourse();
 31
 32
            return courses;
 33
         }
 34
    B
 35
```

```
🖹 view-course.html 💢
  1 <!DOCTYPE html>
   20 <html xmlns:th="http://www.thymeleaf.org">
   4
             <title> View Course by ID </title>
   5
         </head>
         <body>
            <h3 th:text="'ID = ' + ${course.idCourse}"> Course ID </h3>
             <h3 th:text="'Nama = ' + ${course.name}"> Course Name </h3>
             <h3 th:text="'SKS = ' + ${course.credits}"> Course Credit </h3>
   9
  10
  11
            <h3> Mahasiswa yang mengambil </h3>
             12⊖
                 th:text="${student.npm} + '-' + ${student.name}">
 13⊜
 14
                 NPM - Nama Mahasiswa
 15
                 16
         2 / la = d. . .
🞵 CourseController.java 🔀
 20⊝
        @Autowired
        CourseService courseDAO;
 21
 22
        @RequestMapping("/course/view/{id_course}")
 23⊕
 24
        public String viewCourse (Model model,
                @PathVariable(value = "id_course") String id_course)
 25
 26
 27
            CourseModel course = courseDAO.selectCourse (id_course);
 28
            if (course != null) {
 29
                model.addAttribute ("course", course);
 30
                return "viewCourse";
 31
            } else {
 32
                model.addAttribute ("id_course", id_course);
 33
                return "not-found-course";
 34
            }
 35
        }
 36
        @RequestMapping("/course/viewall")
 37⊜
 38
        public String view (Model model)
 39
        {
 40
            List<CourseModel> courses = courseDAO.selectAllCourse ();
 41
            model.addAttribute ("courses", courses);
 42
 43
            return "view-course";
```

Sama seperti method pada latihan 1, saya menggunakan method dari latihan sebelumnya kemudian untuk mahasiswa yang dimaksud di latihan 1 saya ubah menjadi coursenya.

#### **Membuat Service Consumer**

3.

```
StudentServiceRest.java
                       StudentDAOImpl.java 
1 package com.example.service;
  3⊕ import java.util.List;
 13 @Service
 14 public class StudentDAOImpl implements StudentDAO
         @Autowired
 17
         private RestTemplate restTemplate;
 18
 19⊜
        @Override
△20
        public StudentModel selectStudent (String npm)
 21
 22
             StudentModel student =
             restTemplate.getForObject("http://localhost:8080/rest/student/view/"+npm,StudentModel.class);
 23
 24
             return student:
 25
        }
 26
 27⊜
         @Override
△28
         public List<StudentModel> selectAllStudents ()
 29
<u></u>30
             List<StudentModel> students = restTemplate.getForObject("http://localhost:8080/rest/student/viewall", List.c
 31
             return students;
 32
 33 }
```

```
🔎 StudentServiceRest.java 💢
 12
 13
     import lombok.extern.slf4j.Slf4j;
 14
15
     @S1f4j
 16 @Service
 17
     @Primary
 18 public class StudentServiceRest implements StudentService
 19
 20⊝
         @Autowired
 21
         private StudentDAO studentDAO;
 22
         @Override
 23⊕
         public StudentModel selectStudent (String npm)
\triangle 24
 25
              log.info ("REST - select student with npm {}", npm);
 26
              return studentDAO.selectStudent (npm);
 27
 28
 29
         @Override
 30⊝
         public List<StudentModel> selectAllStudents ()
\triangle 31
 32
              log.info ("REST - select all students");
 33
 34
              return studentDAO.selectAllStudents ();
 35
 36
```

```
← → C ① localhost8080/rest/student/viewall

[{"npm":"123","name":"Chanek","gpa":3.6,"courses":[{"idCourse":null,"name":"MPKT","credits":6,"students":null}]},{"npm":"124","name":"Chanek Jr","gpa":3.4,"courses":
[{"idCourse":null,"name":"PSP","credits":4,"students":null},{"idCourse":null,"name":"SDA","credits":3,"students":null}},{"npm":"128","name":"hehe","gpa":3.25,"courses":[]}]
```

Dengan melengkapi method selectAllStudents pada kelas StudentServiceRest kemudian data dapat ditampilkan.

```
CourseService.java

| CourseDAO.java | C
```

4.

```
CourseService.java 
package com.example.service;

import java.util.List;

public interface CourseService

CourseModel selectCourse (String id_course);

List<CourseModel> selectAllCourse();

List<CourseModel> selectAllCourse();
```

```
- 1
🔝 CourseDAOImpl.java 🖂
1 package com.example.service;
 3⊕ import java.util.List;
 12
 13 @Service
 14 public class CourseDAOImpl implements CourseDAO
 15 {
 160
        @Autowired
 17
        private RestTemplate restTemplate;
 18
 19⊜
        @Override
△20
        public CourseModel selectCourse (String id_course)
 21
            CourseModel course =
 22
            restTemplate.getForObject("http://localhost:808
 23
 24
            return course;
 25
         }
 26
27⊝
        @Override
△28
        public List<CourseModel> selectAllCourse()
29
            List<CourseModel> courses = restTemplate.getFor
№30
 31
            return courses;
 32
33 }
```

```
√ CourseDAOImpl.java

                   1 k!DOCTYPE html>
  20 <html xmlns:th="http://www.thymeleaf.org">
 3⊜
       <head>
          <title>View All Courses</title>
 5
          <div th:replace="fragments/fragment :: links"></div>
       </head>
  6
 7⊜
       <body>
 8
           <div th:replace="fragments/fragment :: header"></div>
 9
           <h1>All Courses</h1>
 10⊝
              11⊖
                 <thead>
 120
                     13
                        No
 14
                        ID
 15
                        Name
                        Credits
 16
 17
                     </thead>
 18
 19⊜
                 ≜20⊖
                     <div th:each="course,iterationStatus: ${courses}">

≙21

                        <h3 th:text="${iterationStatus.count}">No. 1</h3>
 22
 23
                           <h3 th:text="${course.idCourse}">Student NPM</h3>
                           <h3 th:text="${course.name}">Student Name</h3>
 24
                           <h3 th:text="${course.credits}">Student GPA</h3>
 25
 26
                        27
                     </div>
                 28
 29
              30
           <div th:replace="fragments/fragment :: footer"></div>
       </body>
 31
 32 </html>
```

Sama seperti latihan pada nomor 3 dan membuat file baru berupa view-al-course.html untuk tampilannya.

### **LESSON LEARNED**

Saya belajar mengenai pembagian 2 layer yaitu consumer dan produser untuk memisahkan front dan back end dengan menggunakan resttemplate yang disediakan springboot.

Anisha Eva Farida 1506689572 ADPAP – B