

- **LATIHAN**

- **Latihan 1**

Untuk membuat service yang mengembalikan seluruh student saya membuat method `List<StudentModel>` dimana method ini akan mengembalikan list yang berisi student yang ada pada basis data. Method `selectAllStudents` telah dibuat pada tutorial sebelumnya.

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

Contoh tampilan keluarannya:

localhost:8080/rest/student/viewall

```
[{"npm": "123", "name": "Uhuy", "gpa": 4.0, "courses": [{"id_course": "CSC126", "name": "MPKT", "credits": 6, "students": null}]}, {"npm": "150666060", "name": "maria", "gpa": 3.98, "courses": [{"id_course": "CSC124", "name": "SDA", "credits": 3, "students": null}]}, {"npm": "1506689263", "name": "agnes", "gpa": 3.93, "courses": [{"id_course": "CSC124", "name": "SDA", "credits": 3, "students": null}, {"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students": null}]}, {"npm": "2", "name": "a", "gpa": 3.45, "courses": []}]
```

String parse

```
{
  "npm": "123",
  "name": "Uhuy",
  "gpa": 4.0,
  "courses": [
    {
      "id_course": "CSC126",
      "name": "MPKT",
      "credits": 6,
      "students": null
    }
  ]
},
{
  "npm": "150666060",
  "name": "maria",
  "gpa": 3.98,
  "courses": [
    {
      "id_course": "CSC124",
      "name": "SDA",
      "credits": 3,
      "students": null
    }
  ]
},
{
  "npm": "1506689263",
  "name": "agnes",
  "gpa": 3.93,
  "courses": [
    {
      "id_course": "CSC124",
      "name": "SDA",
      "credits": 3,
      "students": null
    },
    {
      "id_course": "CSC125",
      "name": "DDP 1",
      "credits": 4,
      "students": null
    }
  ]
},
{
  "npm": "2",
  "name": "a",
  "gpa": 3.45,
  "courses": []
}]
```

String parse

```
        "name": "SDA",  
        "credits": 3,  
        "students": null  
    }  
]  
},  
[  
    {  
        "npm": "1506689263",  
        "name": "agnes",  
        "gpa": 3.93,  
        "courses": [  
            {  
                "id_course": "CSC124",  
                "name": "SDA",  
                "credits": 3,  
                "students": null  
            },  
            {  
                "id_course": "CSC125",  
                "name": "DDP 1",  
                "credits": 4,  
                "students": null  
            }  
        ]  
    },  
    {  
        "npm": "2",  
        "name": "a",  
        "gpa": 3.45,  
        "courses": [  
            ]  
        }  
    ]  
}
```

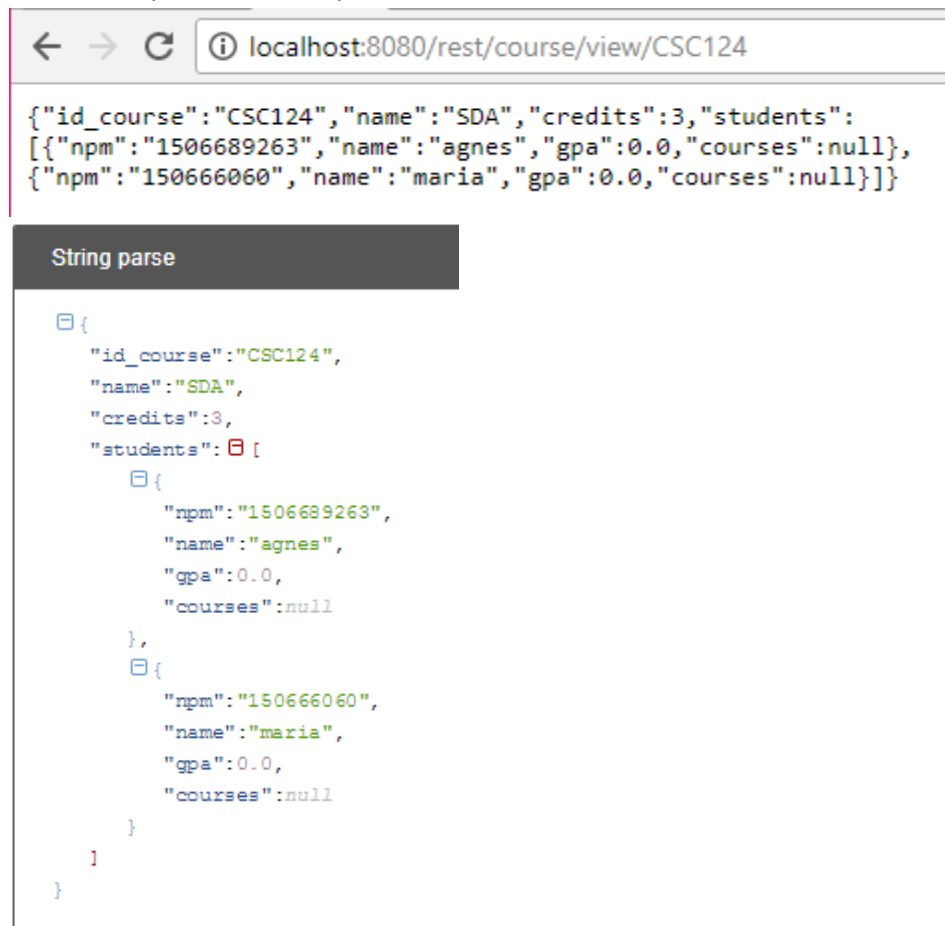
▪ Latihan 2

1. Method View Course By ID

Untuk membuat service yang mengembalikan course yang dicari berdasarkan id saya membuat service baru untuk class Course yaitu CourseRestController. Kemudian saya membuat method viewCourse yang akan mengembalikan course berdasarkan id yang diberikan. Untuk method selectCourse, method tersebut sudah ada pada tutorial sebelumnya, dimana method tersebut akan mengambil data course pada basis data yang sesuai dengan id_course.

```
1 @RestController
2 @RequestMapping("/rest")
3 public class CourseRestController {
4     @Autowired
5     StudentService studentService;
6
7     @RequestMapping("/course/view/{id_course}")
8     public CourseModel viewCourse(@PathVariable(value = "id_course") String id_course) {
9         CourseModel course = studentService.selectCourse(id_course);
10        return course;
11    }
12 }
```

Contoh tampilan keluarannya:



The screenshot shows a web browser window with the address bar displaying `localhost:8080/rest/course/view/CSC124`. Below the address bar, the response body is shown as a JSON object:

```
{
  "id_course": "CSC124",
  "name": "SDA",
  "credits": 3,
  "students": [
    {
      "npm": "1506689263",
      "name": "agnes",
      "gpa": 0.0,
      "courses": null
    },
    {
      "npm": "150666060",
      "name": "maria",
      "gpa": 0.0,
      "courses": null
    }
  ]
}
```

Below the browser window, there is a "String parse" section showing the same JSON object being parsed into a Java object structure:

```
{
  "id_course": "CSC124",
  "name": "SDA",
  "credits": 3,
  "students": [
    {
      "npm": "1506689263",
      "name": "agnes",
      "gpa": 0.0,
      "courses": null
    },
    {
      "npm": "150666060",
      "name": "maria",
      "gpa": 0.0,
      "courses": null
    }
  ]
}
```

2. Method View All Course

Untuk membuat service yang mengembalikan semua course yang ada saya membuat method viewAllCourses pada kelas CourseRestController

```
@RequestMapping("/course/viewall")
public List<CourseModel> viewAllCourses(){
    List<CourseModel> course = studentService.selectAllCourses();
    return course;
}
```

Pada kelas StudentService saya menambahkan method selectAllCourses

```
List<CourseModel> selectAllCourses();
```

```
@Override
public List<CourseModel> selectAllCourses(){
    return studentMapper.selectAllCourses();
}
```

Kemudian pada kelas StudentMapper saya menambahkan method selectAllCourses yang akan mengambil seluruh data course yang ada pada basis data

```
@Select("SELECT student.npm, name " +
        "FROM studentcourse join student " +
        "ON studentcourse.npm = student.npm " +
        "WHERE studentcourse.id_course = #{id_course}")
List<StudentModel> selectStudentCourses();

@Select("SELECT * FROM course")
@Results(value = {
    @Result(property = "id_course", column = "id_course"),
    @Result(property = "name", column = "name"),
    @Result(property = "credits", column = "credits"),
    @Result(property = "students", column = "id_course",
        javaType = List.class,
        many = @Many(select="selectStudentCourses"))
})
List<CourseModel> selectAllCourses();
```

Contoh tampilan keluarannya:



```
[{"id_course": "CSC123", "name": "PSP", "credits": 4, "students": []},
{"id_course": "CSC124", "name": "SDA", "credits": 3, "students":
[{"npm": "1506689263", "name": "agnes", "gpa": 0.0, "courses": null},
{"npm": "150666060", "name": "maria", "gpa": 0.0, "courses": null}]},
{"id_course": "CSC125", "name": "DDP 1", "credits": 4, "students":
[{"npm": "1506689263", "name": "agnes", "gpa": 0.0, "courses": null}]},
{"id_course": "CSC126", "name": "MPKT", "credits": 6, "students":
[{"npm": "123", "name": "Uhuy", "gpa": 0.0, "courses": null}]]
```

```
[
  {
    "id_course": "CSC123",
    "name": "PSP",
    "credits": 4,
    "students": [
    ]
  },
  {
    "id_course": "CSC124",
    "name": "SDA",
    "credits": 3,
    "students": [
      {
        "npm": "1506689263",
        "name": "agnes",
        "gpa": 0.0,
        "courses": null
      },
      {
        "npm": "150666060",
        "name": "maria",
        "gpa": 0.0,
        "courses": null
      }
    ]
  }
]
```

▪ Latihan 3

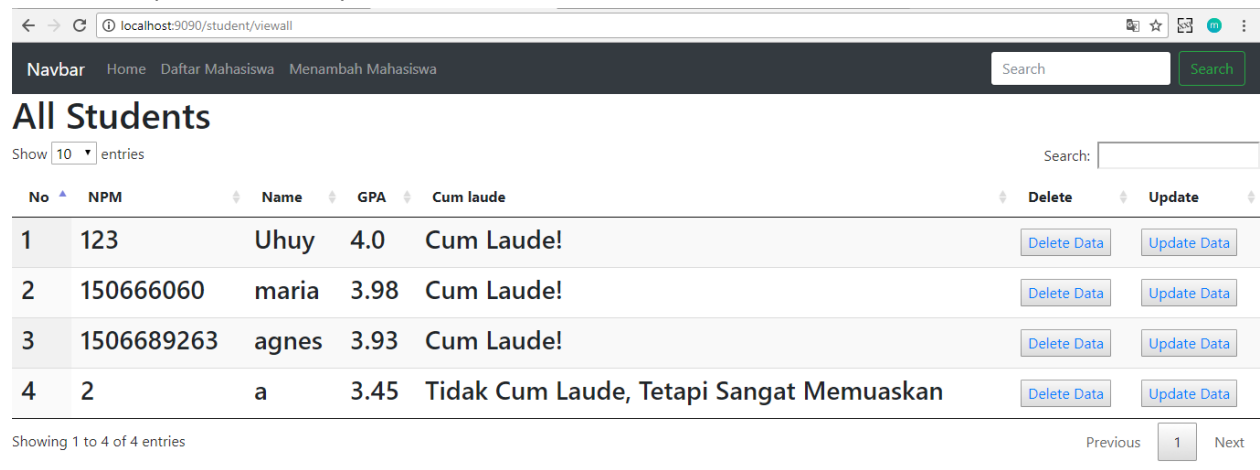
Untuk melengkapi method `selectAllStudents` pada kelas `StudentServiceRest` saya membuat `list<StudentModel>` yang menggunakan `restTemplate` untuk meng-consume object REST web service, dimana url producer web service adalah <http://localhost:8080/rest/student/viewall> dan object yang didapatkan adalah berupa `ArrayList`

```
@Override
public List<StudentModel> selectAllStudents() {
    List<StudentModel> student = restTemplate.getForObject(
        "http://localhost:8080/rest/student/viewall", ArrayList.class);
    return student;
}
```

Lalu saya memanggil method `selectAllStudents` yang ada pada kelas `StudentDAO` di method `selectAllStudents` yang ada di kelas `StudentService Rest`

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

Contoh tampilan keluarannya:



| No | NPM | Name | GPA | Cum laude | Delete | Update |
|----|------------|-------|------|--|-------------|-------------|
| 1 | 123 | Uhuy | 4.0 | Cum Laude! | Delete Data | Update Data |
| 2 | 150666060 | maria | 3.98 | Cum Laude! | Delete Data | Update Data |
| 3 | 1506689263 | agnes | 3.93 | Cum Laude! | Delete Data | Update Data |
| 4 | 2 | a | 3.45 | Tidak Cum Laude, Tetapi Sangat Memuaskan | Delete Data | Update Data |

Showing 1 to 4 of 4 entries

Previous 1 Next

Mata Kuliah APAP

Pada console tertulis:

```
INFO 5488 --- [nio-9090-exec-1] c.e.t.service.StudentServiceRest : REST - select all students
```

▪ Latihan 4

Untuk mengimplementasi service consumer pada class CourseModel saya membuat interface CourseDAO dan kelas CourseDAOImpl pada package dao. Kemudian saya juga membuat interface CourseService dan kelas CourseServiceRest pada package service. Saya menambahkan method selectCourse dan selectAllCourses pada interface CourseDAO

```
public interface CourseDAO {  
  
    CourseModel selectCourse(String id_course);  
    List<CourseModel> selectAllCourses();  
}
```

Saya menambahkan method selectCourse dan selectAllCourses pada kelas CourseDAOImpl

Pada method selectCourse saya membuat object CourseModel yang menggunakan restTemplate untuk meng-consume object REST web service, dimana url producer web service adalah <http://localhost:8080/rest/course/view/CSC124> dan object yang didapatkan adalah berupa kelas CourseModel.

Pada Method selectAllCourses saya membuat List<CourseModel> yang menggunakan restTemplate untuk meng-consume object REST web service, dimana url producer web service adalah <http://localhost:8080/rest/course/viewall> dan object yang didapatkan adalah berupa ArrayList

```
public class CourseDAOImpl implements CourseDAO{  
  
    @Autowired  
    private RestTemplate restTemplate;  
  
    @Override  
    public CourseModel selectCourse( String id_course){  
        CourseModel course = restTemplate.getForObject(  
            "http://localhost:8080/rest/course/view/"+id_course, CourseModel.class);  
        return course;  
    }  
  
    @Override  
    public List<CourseModel> selectAllCourses(){  
        List<CourseModel> course = restTemplate.getForObject(  
            "http://localhost:8080/rest/course/viewall", ArrayList.class);  
        return course;  
    }  
}
```

Saya membuat method selectCourse dan selectAllCourses pada interface CourseService

```
public interface CourseService {  
  
    CourseModel selectCourse(String id_course);  
    List<CourseModel> selectAllCourses();  
}
```

Saya membuat method selectCourse dan selectAllCourses pad akelas CourseServiceRest

Pada method selectCourse saya memanggil method selectCourse yang ada pada CourseDAO. Sedangkan, pada method selectAllCourses saya memanggil method selectAllCourses yang ada pada CourseDAO

```
@Slf4j
@Service
@Primary
public class CourseServiceRest implements CourseService{

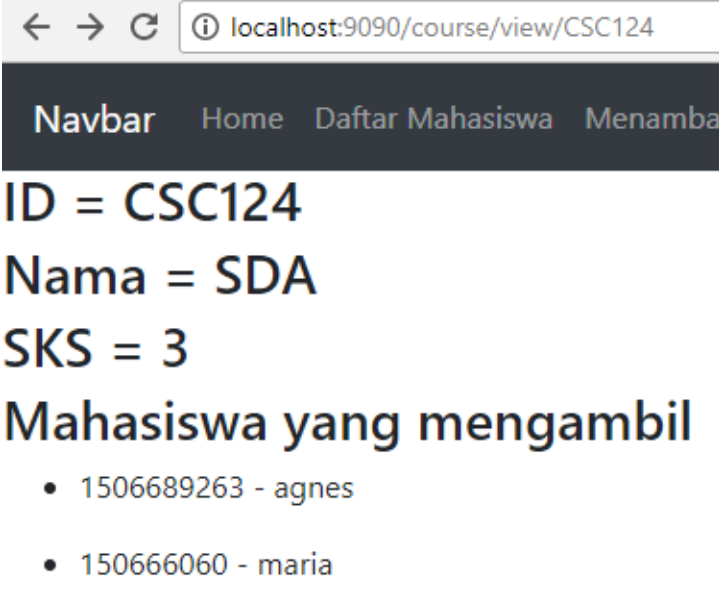
    @Autowired
    private CourseDAO courseDAO;

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

    @Override
    public List<CourseModel> selectAllCourses(){
        log.info("REST - select all courses");
        return courseDAO.selectAllCourses();
    }
}
```

1. Method View Course by ID

Contoh tampilan keluarannya:



The screenshot shows a web browser at the URL `localhost:9090/course/view/CSC124`. The page has a dark navigation bar with links: **Navbar**, Home, Daftar Mahasiswa, and Menamba. The main content area displays the following information:

- ID = CSC124**
- Nama = SDA**
- SKS = 3**
- Mahasiswa yang mengambil**
 - 1506689263 - agnes
 - 150666060 - maria

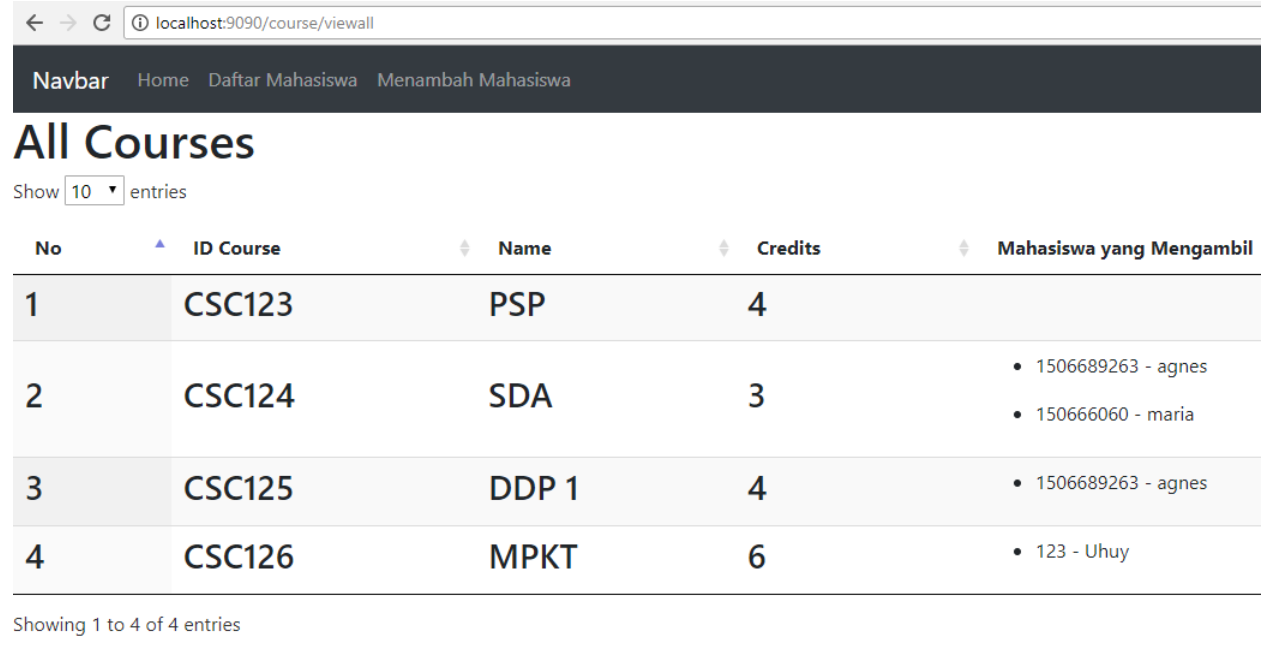
Below this content, the text **Mata Kuliah APAP** is visible.

Contoh keluaran pada console:

```
[nio-9090-exec-8] c.e.tutorial6.service.CourseServiceRest : REST - select course with id CSC124
```

2. Method View All Course

Contoh tampilan keluarannya:



| No | ID Course | Name | Credits | Mahasiswa yang Mengambil |
|----|-----------|-------|---------|--|
| 1 | CSC123 | PSP | 4 | |
| 2 | CSC124 | SDA | 3 | <ul style="list-style-type: none">1506689263 - agnes150666060 - maria |
| 3 | CSC125 | DDP 1 | 4 | <ul style="list-style-type: none">1506689263 - agnes |
| 4 | CSC126 | MPKT | 6 | <ul style="list-style-type: none">123 - Uhuy |

Showing 1 to 4 of 4 entries

Mata Kuliah APAP

Contoh keluaran pada console:

```
[io-9090-exec-10] c.e.tutorial6.service.CourseServiceRest : REST - select all courses
```

• LESSONS LEARNED

- Mengetahui bagaimana cara membuat web service
- Mengetahui cara membuat dan penggunaan service producer dan service consumer
- Mengetahui apa itu RestTemplate dan cara penggunaannya
- Mengetahui kegunaan anotasi @Primary