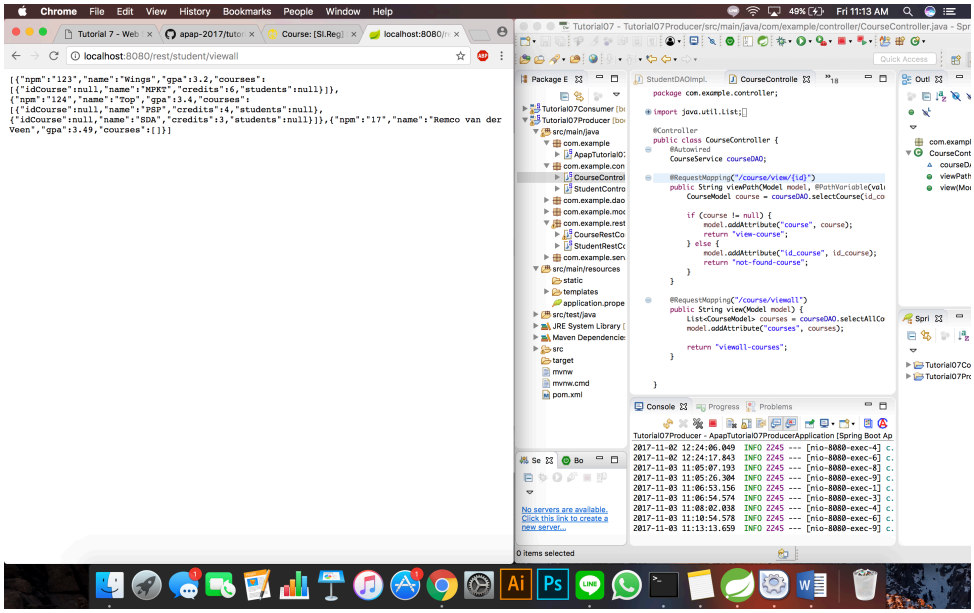
Sum Up yang dipelajari pada Tutorial 7: Hal yang saya pelajari pada Tutorial 7 adalah beberapa hal sebagai berikut – yang pertama yaitu pengimplementasian pemisahan *layer backend* (*service producer*) dan *layer frontend* (*service consumer*). Dengan adanya pemisahan tanggung jawab ini, sebuah aplikasi *service consumer* dapat fokus untuk menyediakan dan mengolah data saja ke pengguna tanpa perlu memiliki database. Sementara *service producer* hanya bertanggung jawab untuk menyediakan data dari *database* dan biasanya tidak memiliki view yang dapat dilihat pengguna.

Latihan 1: Buatlah service untuk mengembalikan seluruh student yang ada di basis data. Service ini mirip seperti method viewAll di Web Controller. Service tersebut di-mapping ke "/rest/student/viewall".

```
@RequestMapping("/student/viewall")
	public String view(Model model) {
		List<StudentModel> students = studentDAO.selectAllStudents();
		model.addAttribute("students", students);

		return "viewall";
	}
```

Contoh keluaran http://localhost:8080/rest/student/viewall



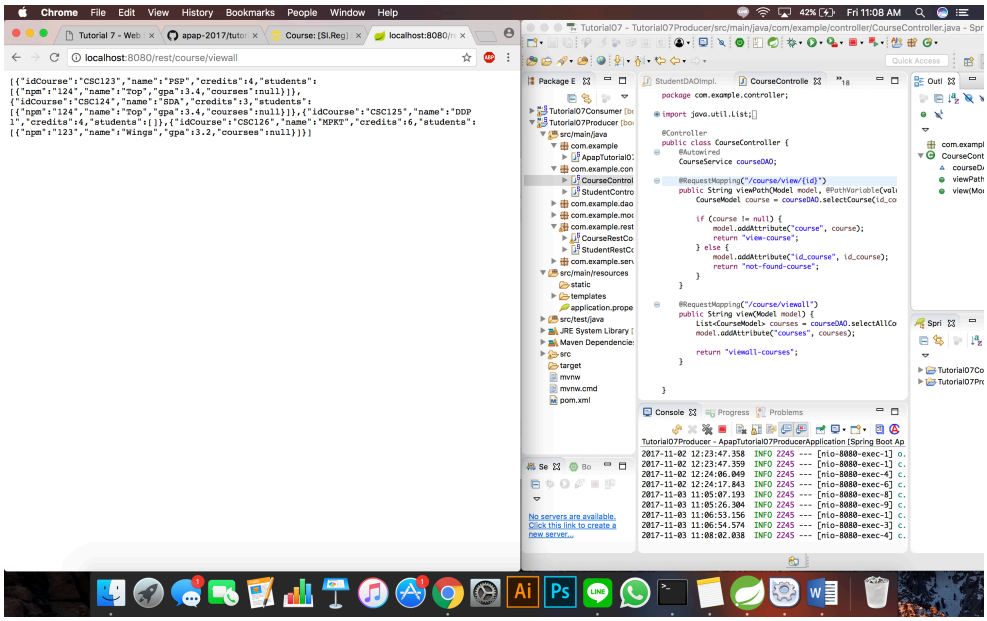
Latihan 2: Buatlah service untuk class Course. Buatlah controller baru yang terdapat service untuk melihat suatu course dengan masukan ID Course (view by ID) dan service untuk melihat semua course (view all).

```
@RequestMapping("/course/view/{id}")
public String viewPath(Model model, @PathVariable(value = "id") String id_course) {
		CourseModel course = courseDAO.selectCourse(id_course);

		if (course != null) {
			model.addAttribute("course", course);
```

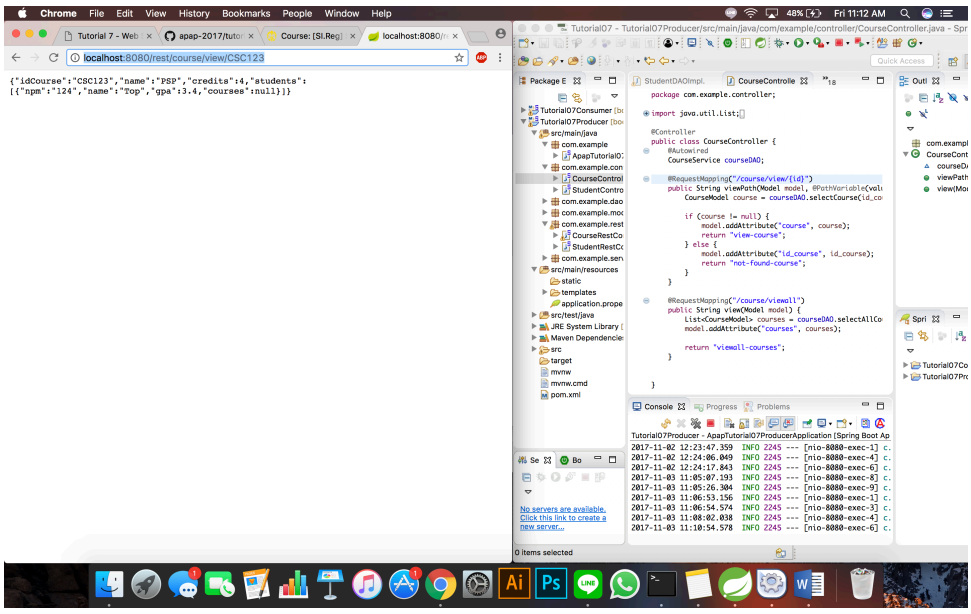
Hera Mentari Putri | 1506689420
Kelas APAP A

```
                return "view-course";
        } else {
                model.addAttribute("id_course", id_course);
                return "not-found-course";
        }
}

@RequestMapping("/course/viewall")
public String view(Model model) {
        List<CourseModel> courses = courseDAO.selectAllCourses();
        model.addAttribute("courses", courses);

        return "viewall-courses";
}
```

Contoh keluaran http://localhost:8080/rest/course/viewall



Contoh keluaran http://localhost:8080/rest/course/view/CSC123



Hera Mentari Putri | 1506689420
Kelas APAP A

Latihan 3: Implementasikan service consumer untuk view all Students dengan melengkapi method selectAllStudents yang ada di kelas StudentServiceRest.

Pada file StudentDAOImpl.java:

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

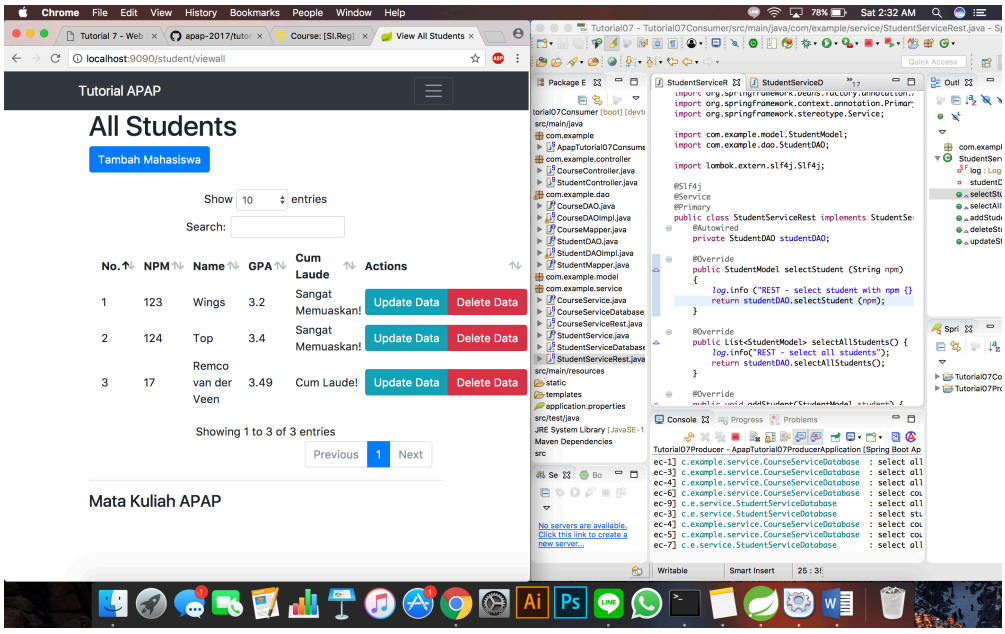
Pada file StudentServiceRest.java:

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

Penambahan pada file ApapTutorial07ConsumerApplication.java:

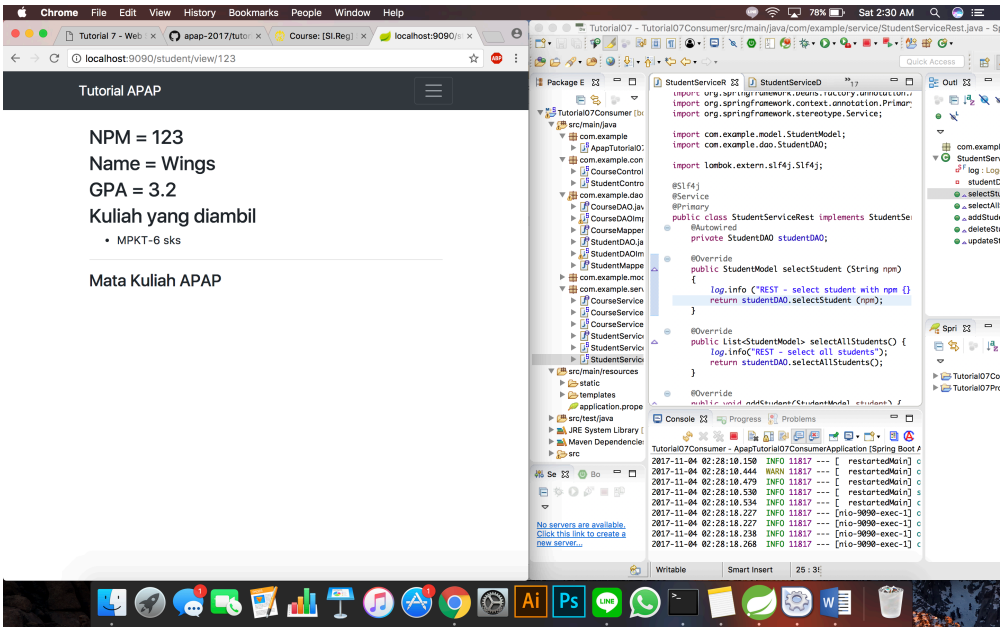
```
// create new bean for restTemplate
// https://stackoverflow.com/questions/28024942/how-to-autowire-resttemplate-using-
annotations
	@Bean
	public RestTemplate restTemplate() {
	    return new RestTemplate();
	}
```

Contoh keluaran http://localhost:9090/student/viewall



Hera Mentari Putri | 1506689420
Kelas APAP A

Contoh keluaran http://localhost:9090/student/view/123



Latihan 4: Implementasikan service consumer untuk class CourseModel dengan membuat class-class DAO dan service baru.

Membuat file baru yaitu CourseDAO.java, dan CourseDAOImpl.java. Caranya sama dengan latihan no.3 namun untuk file course.

Pada file CourseDAOImpl.java:

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

	@Override
	public List<CourseModel> selectAllCourses() {
		List<CourseModel> courses =
restTemplate.getForObject("http://localhost:8080/rest/course/viewall",
				List.class);
		return courses;
	}
```

Pada file CourseServiceRest.java:

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

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

Contoh keluaran http://localhost:9090/course/view/CSC123