

## Write-Up Tutorial 3 (Menggunakan Model dan Service pada Project Spring Boot)

### Membuat Class Model

Setelah mengikuti petunjuk untuk menambahkan *StudentModel* kemudian menambahkan *method constructor, setter, dan getter* untuk setiap *atribut*.

```
public void setName(String name){
    this.name = name;
}

public String getName(){
    return name;
}

public void setNpm(String npm){
    this.npm = npm;
}

public String getNpm(){
    return npm;
}

public void setGpa(double gpa){
    this.gpa = gpa;
}

public double getGpa() {
    return gpa;
}
```

### Membuat Service

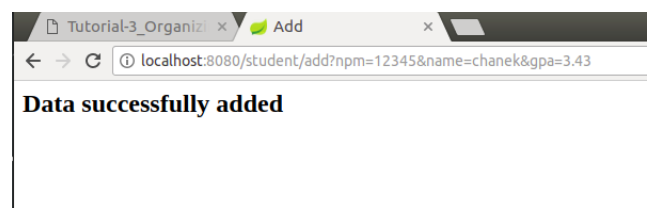
Membuat 2 buah service yaitu *StudentService* dan *InMemoryStudentService*. *StudentService* merupakan interface yang mendefinisikan *method-method* apa saja yang dapat dilakukan untuk memanipulasi kelas *Student*. *InMemoryStudentService* adalah service yang meng-*implements StudentService*. Kenapa di-*implements* ? Karna kali ini saya menggunakan static List untuk menyimpan data *Students*, sehingga jika nanti digganti menjadi *database* hanya perlu menambahi service baru yang meng-*implements StudentService* sehingga tidak perlu memodifikasi *StudentService*.

### Membuat Controller dan Fungsi Add

Membuatclass *StudentController* dan menambahkan *method add* didalamnya. Kemudian menambahkan view pada *directory resources/templates*.

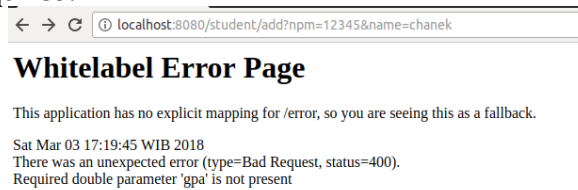
Jalankan program dan membuka localhost:8080/student/add?pm=12345&name=chanek&gpa=3.43

**Jawaban 1:** Hasilnya adalah data berhasil ditambahkan dengan RequestParam



Buka localhost:8080/student/add?npm=12345&name=chanek

**Jawaban 2:** Hasilnya adalah error karena kurang memasukkan parameter gpa RequestParam sedangkan gpa sifatnya required.

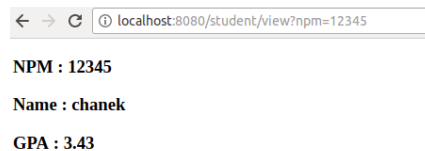


### **Method View by NPM**

Menambahkan *method view* pada *StudentController*. *Method* ini berfungsi untuk mengecek apakah data yang ditambahkan tadi benar tersimpan.

Jalankan program dan membuka localhost:8080/student/add?pm=12345&name=chanek&gpa=3.43 kemudian membuka localhost:8080/student/view?npm=12345

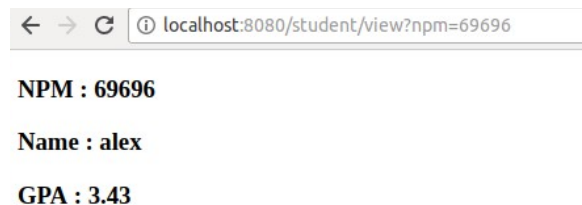
**Jawaban 3:** Hasilnya adalah data *Student* berhasil muncul



Mematikan program dan jalankan kembali serta buka localhost:8080/student/view?npm=12345

**Jawaban 4:** data *Student* tidak muncul karena lprogram sebelumnya sudah mati dan semua data yang sudah disimpan sebelumnya otomatis akan hilang. Sehingga ketika *view* lagi data sudah tidak ada.

Kemudian saya menambahkan data student lain maka data tersebut bisa dilihat.

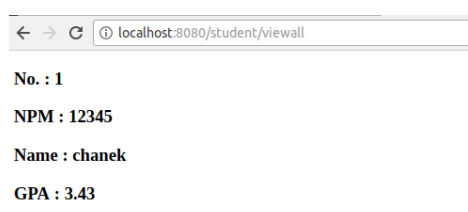


### **Method View ALL**

Menambahkan *method viewALL* pada *StudentController*. *Method* ini berfungsi untuk melihat semua data *Student*

Jalankan program dan membuka localhost:8080/student/add?pm=12345&name=chanek&gpa=3.43 kemudian membuka localhost:8080/student/viewall

**Jawaban 5:** Hasilnya adalah data *Student* berhasil muncul



Menambahkan data *Student* lain kemudian membuka localhost:8080/student/viewall

**Jawaban 6:** Hasilnya adalah Semua data *Student* berhasil muncul

← → ↻ 🔍 localhost:8080/student/viewall

No. : 1
NPM : 12345
Name : chaneK
GPA : 3.43

---

No. : 2
NPM : 69696
Name : alex
GPA : 3.43

## Latihan

### 1. Menambahkan *method view* dengan menggunakan *path variable*

Berikut potongan *code* yang saya tambahkan pada *StudentController*

```
@RequestMapping(value = {"student/view/", "student/view/{npm}"})
public String view(@PathVariable Optional<String> npm, Model model){

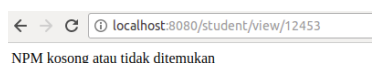
    if (npm.isPresent()){
        StudentModel student=studentService.selectStudent(npm.get());
        if (student != null){
            model.addAttribute("student", student);
            return "view";
        }
        else{
            return "viewnotfound";
        }
    }
    else{
        return "viewnotfound";
    }
}
```

Kemudian menambahkan *viewnotfound* pada *directory resources/templates* yang berisi seperti berikut.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Delete Student by NPM</title>
</head>
<body>

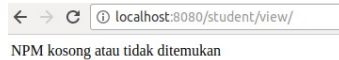
NPM kosong atau tidak ditemukan
</body>
</html>
```

Berikut hasilnya ketika membuka localhost:8080/student/view/12453



NPM kosong atau tidak ditemukan

Dan berikut hasilnya ketika membuka localhost:8080/student/view/



## 2. Menambahkan *feature delete* berdasarkan NPM

- Menambahkan *method deleteStudent* pada *studentService* sehingga menjadi seperti berikut

```
public interface StudentService {  
    StudentModel selectStudent(String npm);  
    List<StudentModel> selectAllStudents();  
    void addStudent(StudentModel student);  
    void deleteStudent(String npm);  
}
```

- Kemudian meng-*override* method tersebut di *InMemoryStudentService* menjadi seperti berikut

```
@Override  
public void deleteStudent(String npm){  
    StudentModel entityStudent = selectStudent(npm);  
    if (entityStudent != null) {  
        studentList.remove(entityStudent);  
    }  
}
```

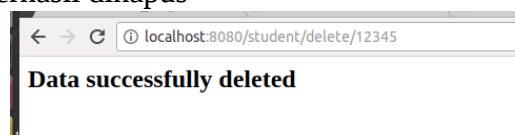
- Menambahkan *method delete* pada *StudentController* seperti berikut

```
@RequestMapping(value = {"/student/delete", "student/delete/{npm}"})  
public String delete(@PathVariable Optional<String> npm, Model model) {  
    if (npm.isPresent()){  
        StudentModel student=studentService.selectStudent(npm.get());  
        if (student != null){  
            System.out.println(student);  
            studentService.deleteStudent(npm.get());  
            return "delete";  
        }  
        else{  
            return "notdelete";  
        }  
    }  
    else {  
        return "notdelete";  
    }  
}
```

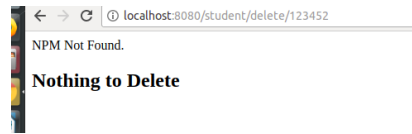
- Menambahkan *delete* dan *notdelete* pada *directory resources/templates*

Berikut hasilnya

- ✓ Ketika data berhasil dihapus



- ✓ Ketika data tidak ditemukan



- ✓ Ketika parameter tidak lengkap

