

## LESSON LEARNED

Saya mempelajari bahwa bahkan java yang sudah memiliki sistem OOP dapat menjadi lebih OOP jika menggunakan framework spring boot ini. Dalam tutorial ini, banyak sekali hal baru yang saya pelajari dari penggunaan framework spring boot. Yang pertama adalah kegunaan dari interface, dimana isinya berupa method-method kosong yang belum diimplementasi, namun method tersebut adalah method yang kedepannya akan kita gunakan terus menerus. Tidak hanya itu, interface di sini juga membantu mendefinisikan kebutuhan kita akan Model dan data apa saja yang akan kita gunakan. Interface ini lah yang akan diimplement di InMemoryStudent untuk diimplementasikan method-methodnya. Dan terakhir, saya mempelajari kegunaan dari model, dimana model ini adalah tempat kita menyimpan objek dan data yang akan digunakan di dalam method.

## TUTORIAL

### Membuat Service

#### 4. Implementasi kode selectStudents()

```
@Override
public StudentModel selectStudent(String npm) {

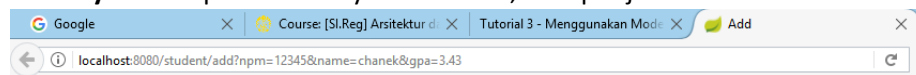
    for (StudentModel s: studentList) {
        if (s.getNpm().equals(npm)) {
            return s;
        }
    }
    return null;
}
```

Di sini saya menggunakan enhanced for loop untuk memeriksa npm ada studentList

### Membuat Controller dan fungsi Add

#### 1. Localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

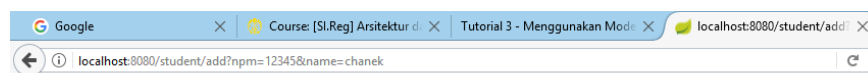
Pertanyaan 1: Apakah hasilnya? Jika error, tulis penjelasan anda



Data berhasil ditambahkan

#### 2. Localhost:8080/student/add?npm=12345&name=chanek

Pertanyaan 2: Apakah hasilnya? Jika error, tulis penjelasan anda



### Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Sep 22 09:03:08 GMT+07:00 2017

There was an unexpected error (type=Bad Request, status=400).

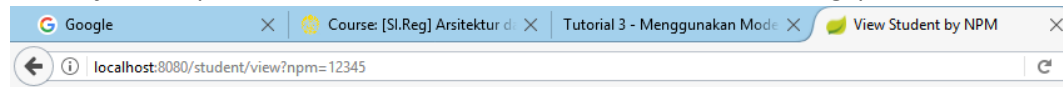
Required double parameter 'gpa' is not present

Terjadi error Karena parameter yang dimasukkan dari route tidak lengkap atau tidak sesuai dengan parameter yang dibutuhkan di method controller

### Method View by NPM

3. Jalankan program dan buka **localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43**, lalu buka **localhost:8080/student/view?npm=12345**

**Pertanyaan 3:**Apakah data Student tersebut muncul? Jika tidak, mengapa?



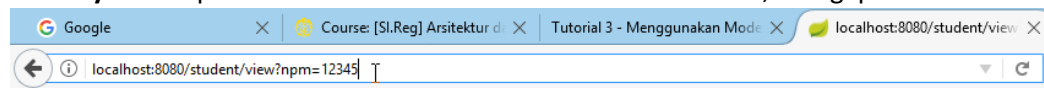
**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

4. Coba matikan program dan jalankan kembali lalu buka **localhost:8080/student/view?npm=12345**

**Pertanyaan 4:**Apakah data Student tersebut muncul? Jika tidak, mengapa?



### Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

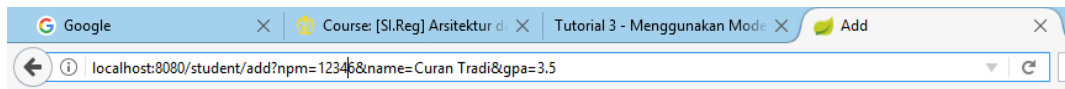
Fri Sep 22 16:43:32 GMT+07:00 2017

There was an unexpected error (type=Internal Server Error, status=500).

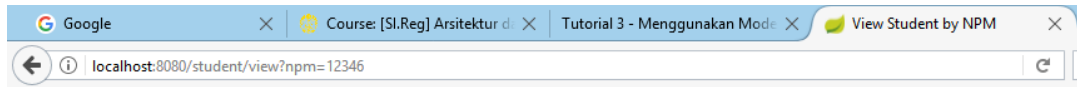
Exception evaluating SpringEL expression: "student.npm" (view:7)

Model hanya menyimpan data saat program dijalankan, sehingga jika program dimatikan lalu dijalankan kembali, data objek student direstart menjadi tidak ada. Jika kita melakukan view sebelum melakukan add pada saat menjalankan program, maka student tidak akan muncul.

5. Coba tambahkan data Student lainnya dengan NPM yang berbeda



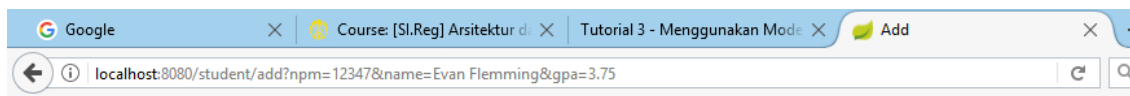
**Data berhasil ditambahkan**



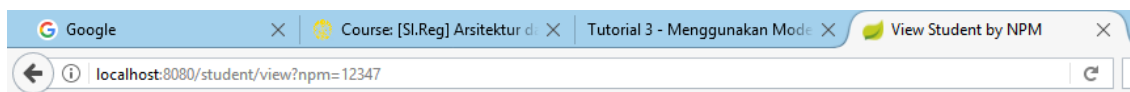
**NPM = 12346**

**Name = Curan Tradi**

**GPA = 3.5**



**Data berhasil ditambahkan**



**NPM = 12347**

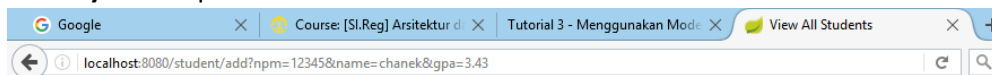
**Name = Evan Flemming**

**GPA = 3.75**

**Method View All**

3. Jalankan program dan buka  
**localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43**, lalu buka  
**localhost:8080/student/viewall**

**Pertanyaan 5: apakah data Student tersebut muncul?**



**No. 1**

**NPM = 12345**

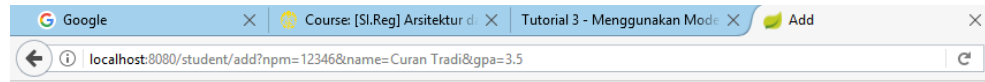
**Name = chanek**

**GPA = 3.43**

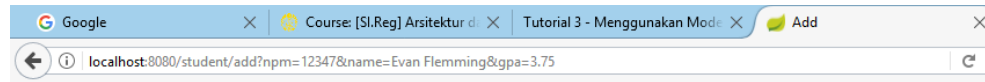
---

4. Coba tambahkan data Student lainnya dengan NPM berbeda, lalu buka **localhost:8080/student/viewall**

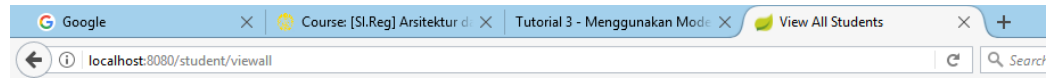
**Pertanyaan 6:** apakah data Student tersebut muncul?



**Data berhasil ditambahkan**



**Data berhasil ditambahkan**



**No. 1**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

---

**No. 2**

**NPM = 12346**

**Name = Curan Tradi**

**GPA = 3.5**

---

**No. 3**

**NPM = 12347**

**Name = Evan Flemming**

**GPA = 3.75**

---

## LATIHAN

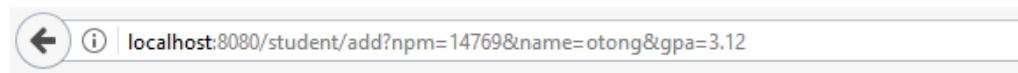
### 1. Tahap-tahap Menyelesaikan Fitur:

- ➔ Pertama-tama saya membuat method baru dengan nama viewPath()
- ➔ Sesuai dengan permintaan soal, saya mengisi parameter dengan parameter path variable, dan model
- ➔ **Untuk request mappingnya, agar tidak bertabrakan dengan method view() sebelumnya, pada method view saya ganti routingnya menjadi student/viewp**
- ➔ Karena saya harus handle jika npm kosong atau tidak ditemukan, dengan kata lain saya harus mereturn halaman selain halaman “view” yaitu halaman error. Oleh Karena itu, saya membuat if yang me-return halaman view jika npm berhasil ditemukan. Kemudian saya membuat if lagi untuk mengecek apakah npm tersebut(dari objek student) tersedia. dan else yang me-return halaman “viewerror” jika npm kosong atau tidak ditemukan.
- ➔ Yang terakhir, saya hanya perlu membuat halaman html baru yang sederhana untuk menjadi halaman viewerror

#### Method viewPath:

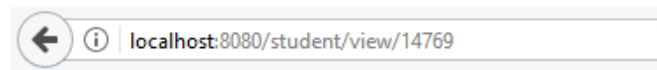
```
@RequestMapping(value = {"/student/view", "/student/view/{npm}"})  
public String viewPath(@PathVariable Optional<String> npm, Model model)  
{  
    if(npm.isPresent()) {  
        StudentModel student = studentService.selectStudent(npm.get());  
        if(student == null) {  
            return "viewerror";  
        }else {  
            model.addAttribute("student", student);  
            return "view";  
        }  
    }else {  
        return "viewerror";  
    }  
}
```

### Hasil Tampilan:



## Data berhasil ditambahkan

Jika npm berhasil ditemukan



**NPM = 14769**

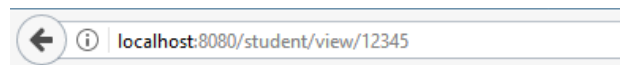
**Name = otong**

**GPA = 3.12**

Jika npm kosong dan tidak ditemukan



## NPM kosong atau tidak ditemukan



## NPM kosong atau tidak ditemukan

### 2. Tahap-tahap Menyelesaikan Fitur dan penjelasan fitur delete:

- ➔ Pertama saya membuat method deleteStudent() pada class InMemoryStudent() yang bertugas untuk menghapus objek student dari array list studentList
- ➔ Tidak lupa untuk menambahkan `void deleteStudent(StudentModel student);`  
Pada class StudentService sebagai interface
- ➔ Sesuai dengan soal, saya membuat method delete pada controller dengan parameter path variable
- ➔ Di dalamnya saya membuat 1 if dimana if pertama mengecek apakah ada input npm yang ingin di delete, sementara elsenya langsung mereturn halaman error yang berarti tidak ada input
- ➔ Di dalam if tersebut ada if lagi yang mengecek apakah objek student yang memiliki npm masukan dari input tersedia di dalam studentList.
- ➔ Jika tersedia, maka program akan mendelete student dari list dan mereturn ke halaman yang memberikan informasi sukses mendelete student, jika tidak tersedia maka akan mereturn ke halaman error

#### Method deleteStudent:

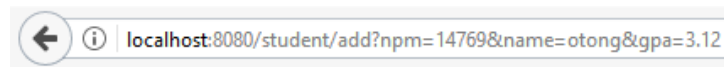
```
@Override
public void deleteStudent(StudentModel student)
{
    studentList.remove(student);
}
```

#### Method delete:

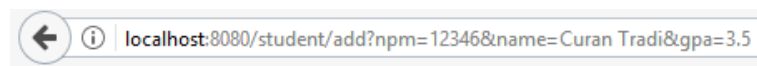
```
@RequestMapping(value = {"/student/delete", "/student/delete/{npm}"})
public String delete(@PathVariable Optional<String> npm, Model model)
{
    if(npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student == null) {
            return "viewerror";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

#### Hasil Tampilan:

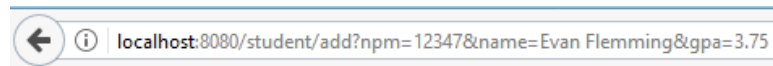
Menambah 3 student:



**Data berhasil ditambahkan**



**Data berhasil ditambahkan**



**Data berhasil ditambahkan**

## Viewall

← ⓘ localhost:8080/student/viewall
<b>No. 1</b>
<b>NPM = 14769</b>
<b>Name = otong</b>
<b>GPA = 3.12</b>
<hr/>
<b>No. 2</b>
<b>NPM = 12346</b>
<b>Name = Curan Tradi</b>
<b>GPA = 3.5</b>
<hr/>
<b>No. 3</b>
<b>NPM = 12347</b>
<b>Name = Evan Flemming</b>
<b>GPA = 3.75</b>

## Delete Curan Tradi (12346)

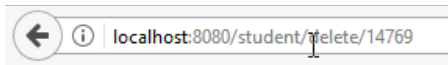
← ⓘ localhost:8080/student/delete/12346
---

## Student berhasil didelete!

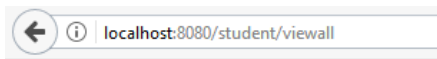
← ⓘ localhost:8080/student/viewall
<b>No. 1</b>
<b>NPM = 14769</b>
<b>Name = otong</b>
<b>GPA = 3.12</b>
<hr/>
<b>No. 2</b>
<b>NPM = 12347</b>
<b>Name = Evan Flemming</b>
<b>GPA = 3.75</b>



### Delete Otong (14769)



**Student berhasil didelete!**



**No. 1**

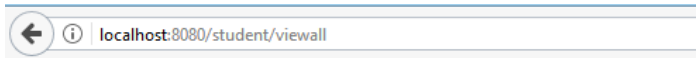
**NPM = 12347**

**Name = Evan Flemming**

**GPA = 3.75**

---

### Delete Gagal



**No. 1**

**NPM = 14769**

**Name = otong**

**GPA = 3.12**

---

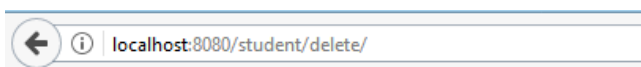
**No. 2**

**NPM = 12345**

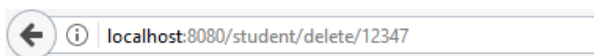
**Name = chanek**

**GPA = 3.43**

---



**NPM kosong atau tidak ditemukan**



**NPM kosong atau tidak ditemukan**