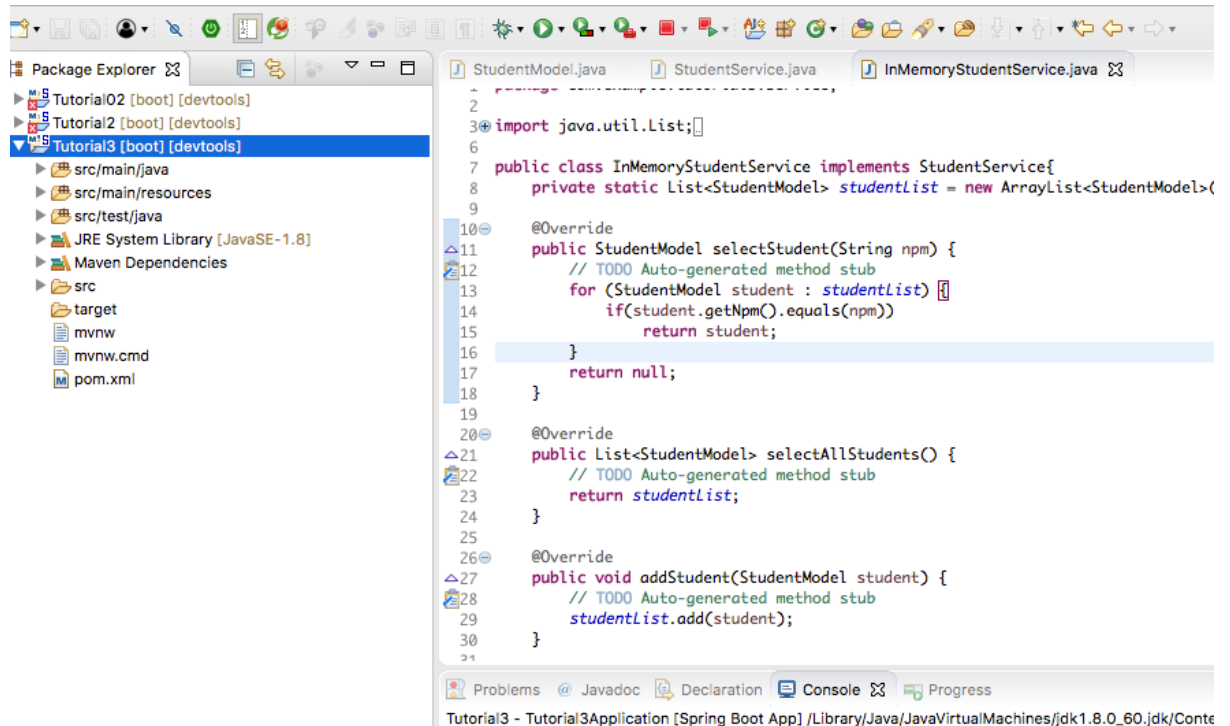


Membuat Class Model

1. Method selectStudent setelah berhasil di implementasikan:

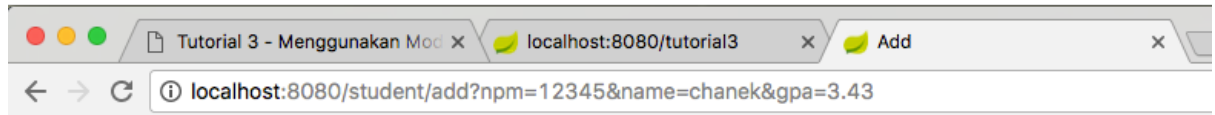


Pada metode berikut, mengiterasikan arraylist untuk menemukan mahasiswa dengan NPM yang sama dengan yang dimasukan dalam parameter, jika ada yang sama kemudian *return* dari mahasiswa yang didapatkan, jika tidak maka NULL menandakan bahwa tidak ada mahasiswa tersebut.

Membuat Service

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

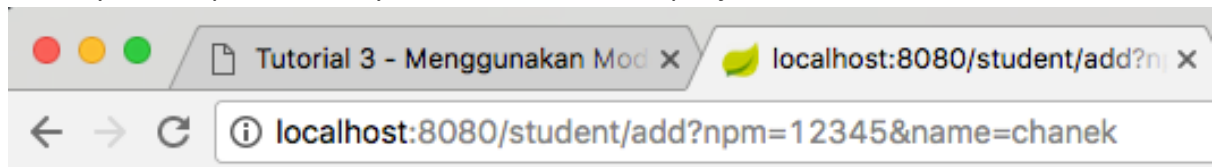
Pertanyaan 1: apakah hasilnya? Jika error, tuliskan penjelasan Anda.



Data berhasil ditambahkan

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

Pertanyaan 2: apakah hasilnya? Jika error, tuliskan penjelasan Anda.



Whitelabel Error Page

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

Fri Sep 22 18:08:44 WIB 2017

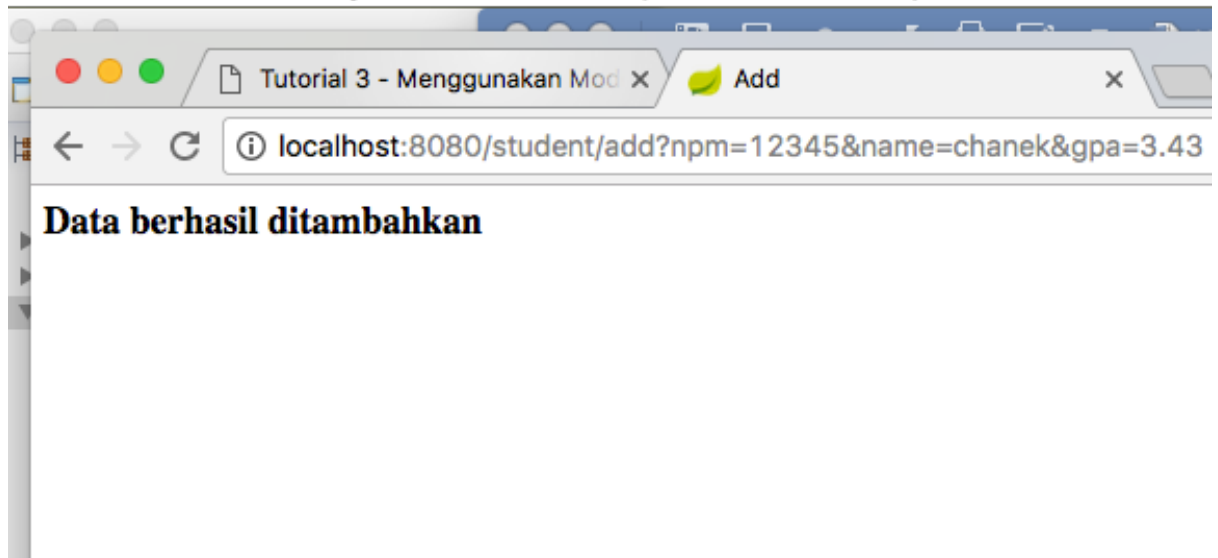
There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

Terjadi Whitelabel Error Page (type=Bad Request, status=400), dibutuhkan parameter 'gpa' untuk RequestParam GET

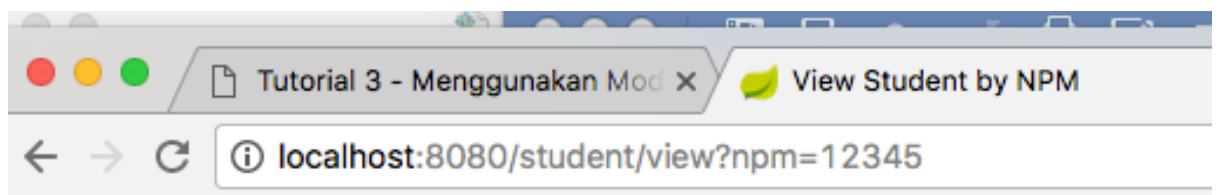
Method View by NPM

1. Jalankan program dan buka
`localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43`



lalu buka

`localhost:8080/student/view?npm=12345`



NPM = 12345

Name = chanek

GPA = 3.43

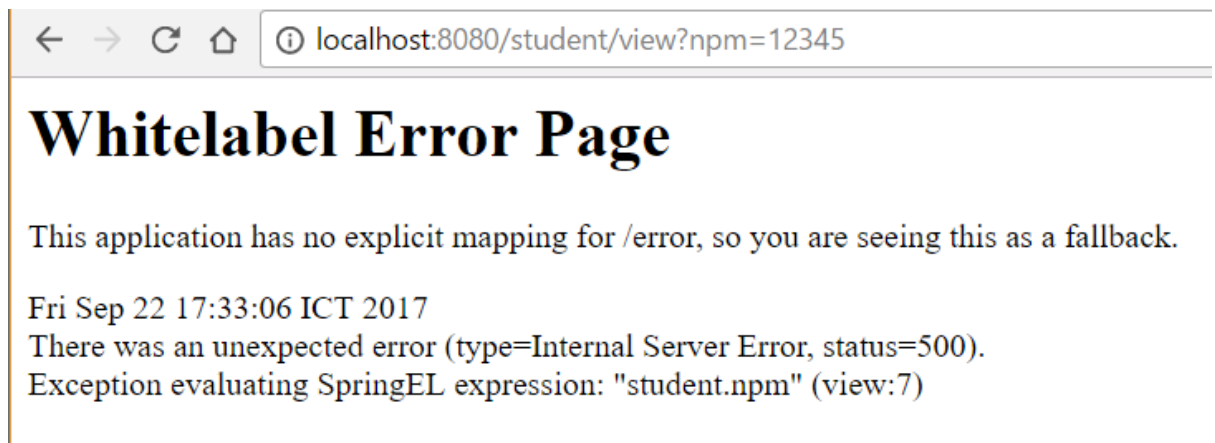
Pertanyaan 3: apakah data Student tersebut muncul? Jika tidak, mengapa?

Muncul.

Data student yang dimaksud muncul sesuai dengan yang sebelumnya dimasukkan pada parameter

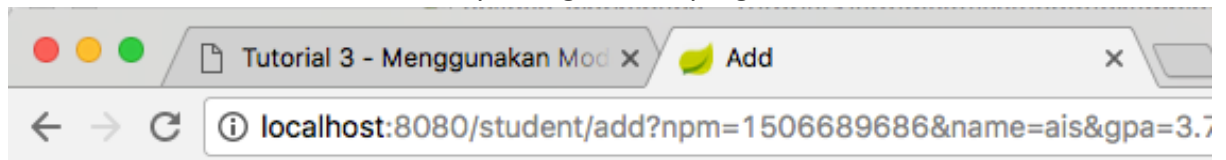
`npm=12345&name=chanek&gpa=3.43`

2. Coba matikan program dan jalankan kembali serta buka
`localhost:8080/student/view?npm=12345`



Pertanyaan 4: apakah data Student tersebut muncul? Jika tidak, mengapa?
Tidak. Data student ada pada session setiap kali springboot dijalankan. Ketika springboot dimatikan, maka session akan diperbaharui sepenuhnya.

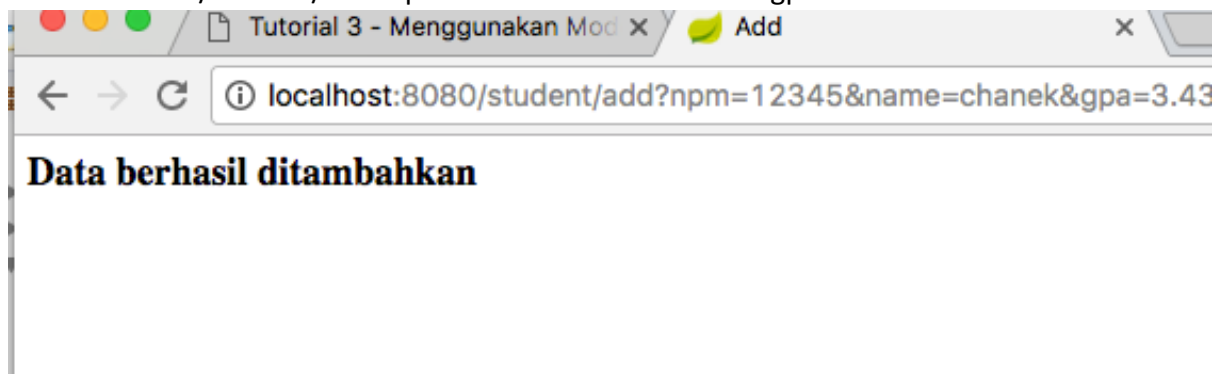
3. Coba tambahkan data Student lainnya dengan NPM yang berbeda.



Data berhasil ditambahkan

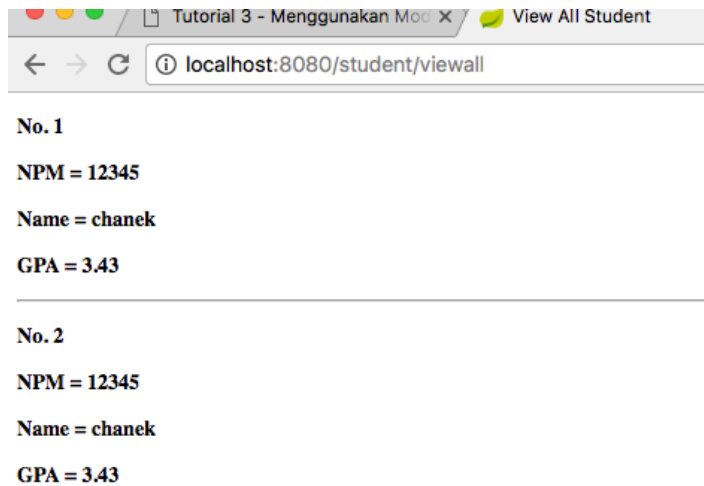
Method View All

1. Jalankan program dan buka
localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

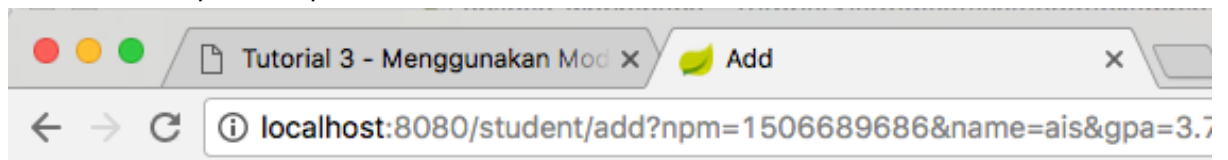


lalu buka localhost:8080/student/viewall

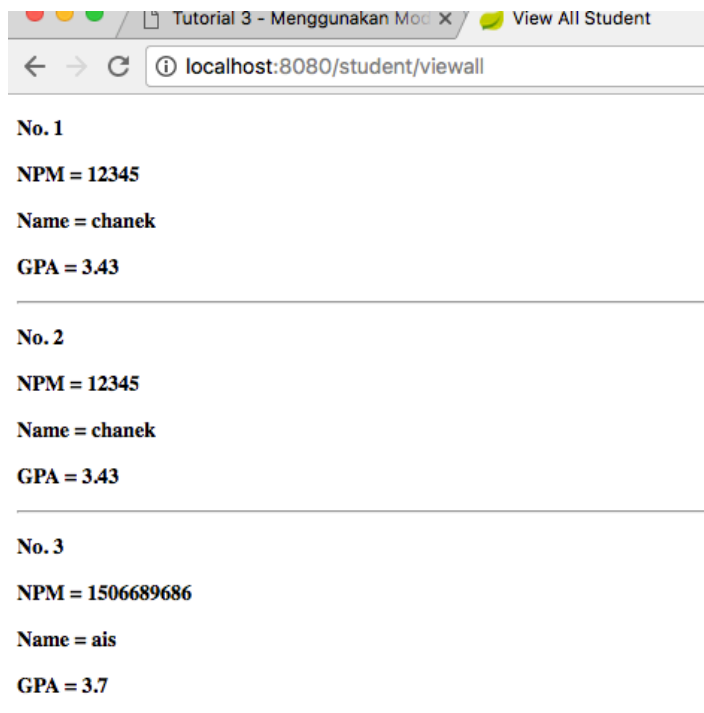
Aisyah Husna
1506689686
ADPAP - C



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



Data berhasil ditambahkan



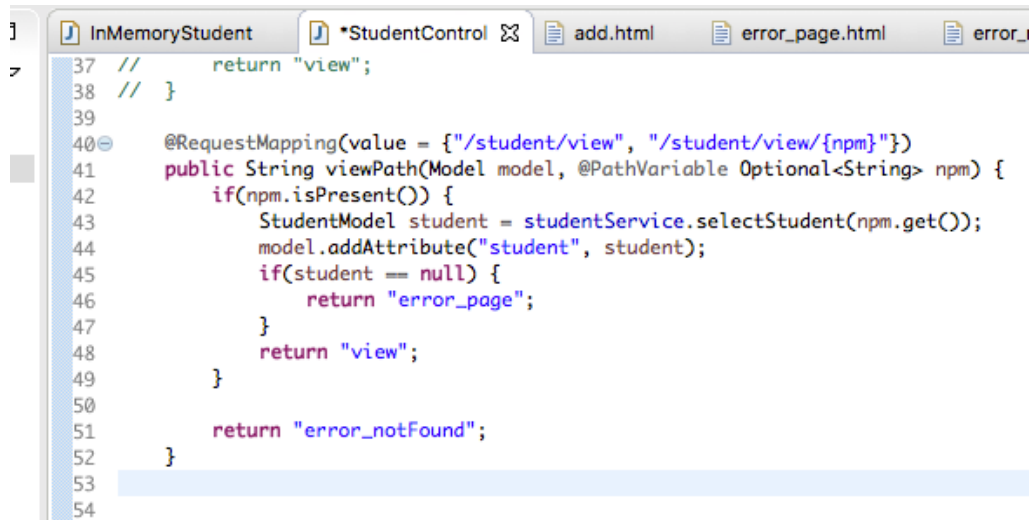
Pertanyaan 6: Apakah semua data Student muncul?

Muncul

Latihan

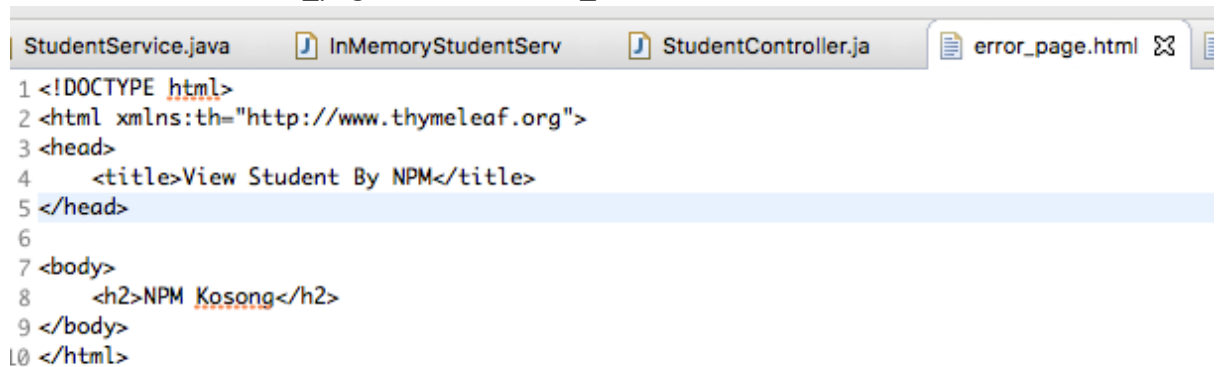
1. Pada StudentController tambahkan sebuah method view Student dengan menggunakan Path Variable.

Method yang ditambahkan



```
37 // return "view";
38 // }
39
40 @RequestMapping(value = {"/student/view", "/student/view/{npm}"})
41 public String viewPath(Model model, @PathVariable Optional<String> npm) {
42     if(npm.isPresent()) {
43         StudentModel student = studentService.selectStudent(npm.get());
44         model.addAttribute("student", student);
45         if(student == null) {
46             return "error_page";
47         }
48         return "view";
49     }
50
51     return "error_notFound";
52 }
53
54
```

buatlah halaman error_page.html dan error_notFound.html



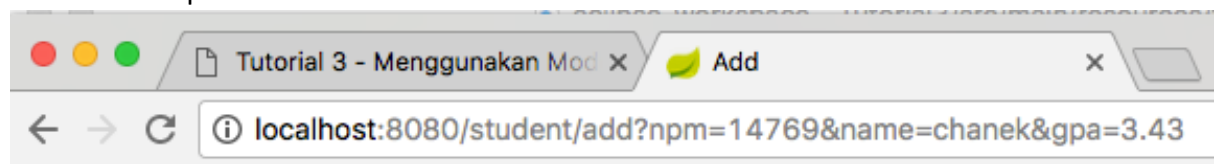
```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title>View Student By NPM</title>
5 </head>
6
7 <body>
8     <h2>NPM Kosong</h2>
9 </body>
10 </html>
```

Aisyah Husna
1506689686
ADPAP - C

```
tudentService.java  InMemoryStudentServ  StudentController.ja  error_page.html  error_notFound.html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>View Student By NPM</title>
</head>

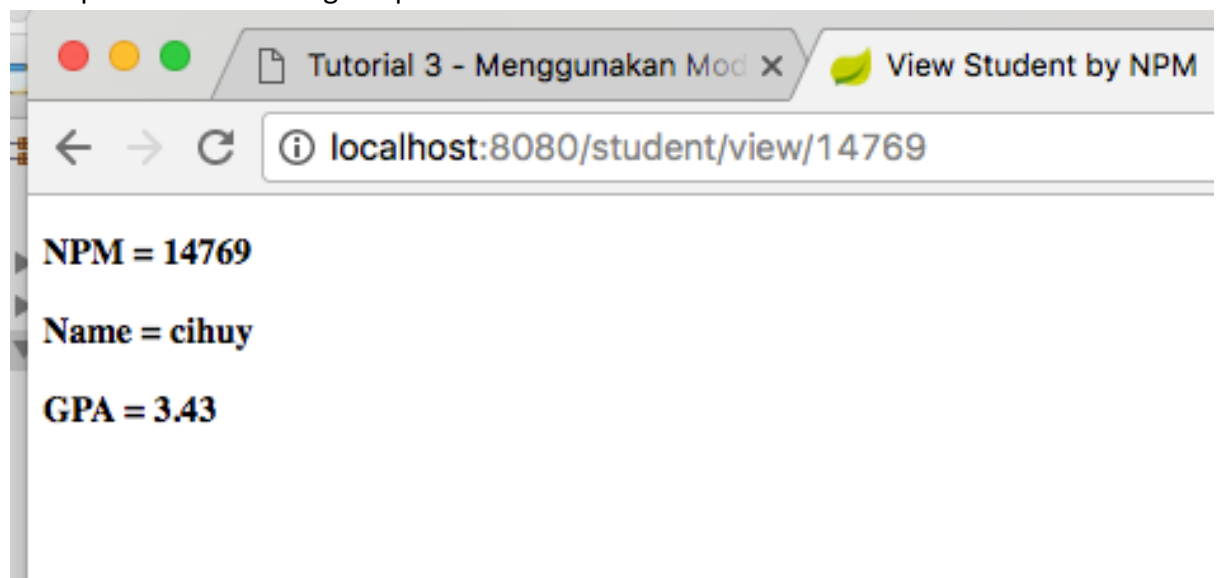
<body>
  <h2>NPM tidak ditemukan.</h2>
</body>
</html>
```

Tambahkan npm 14769

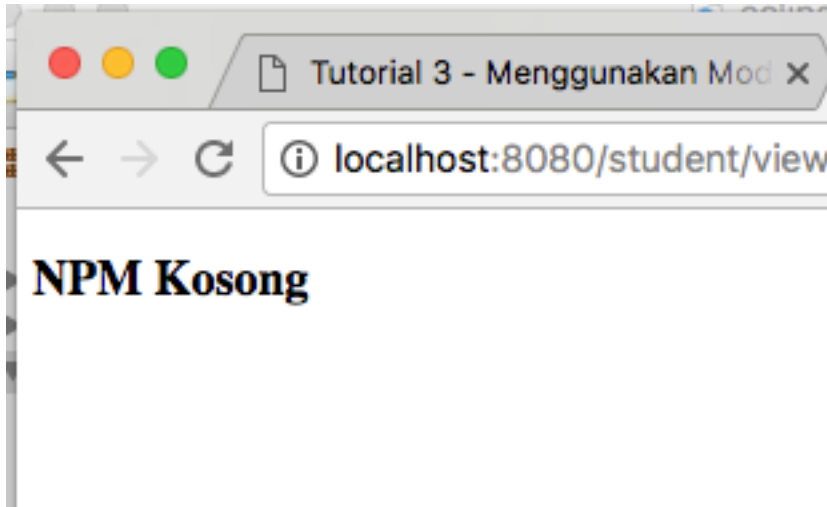


Data berhasil ditambahkan

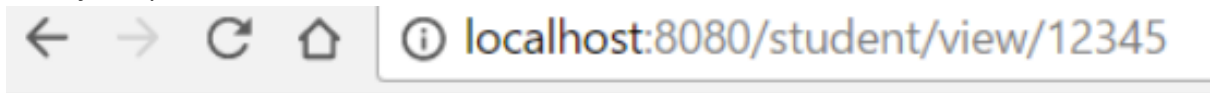
Cek apakah student dengan npm tersebut sudah masuk



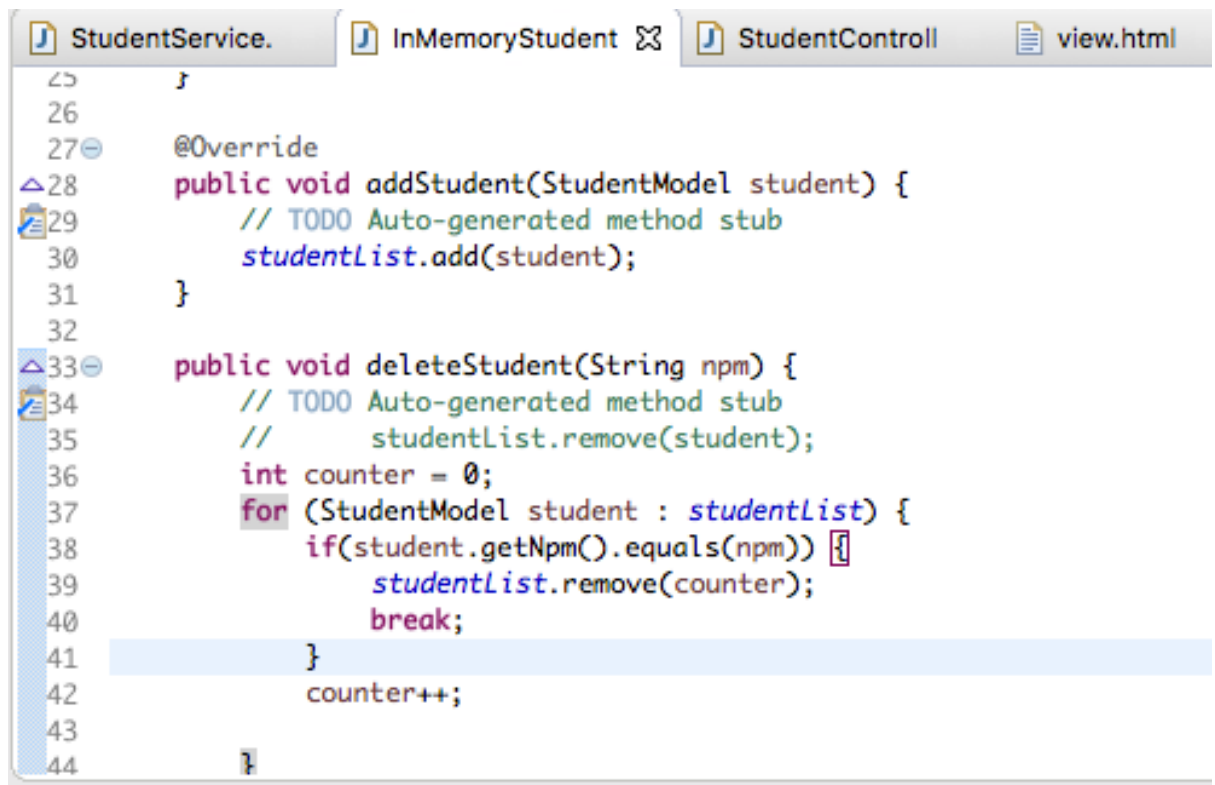
Ketika menulis npm tidak diberikan parameter



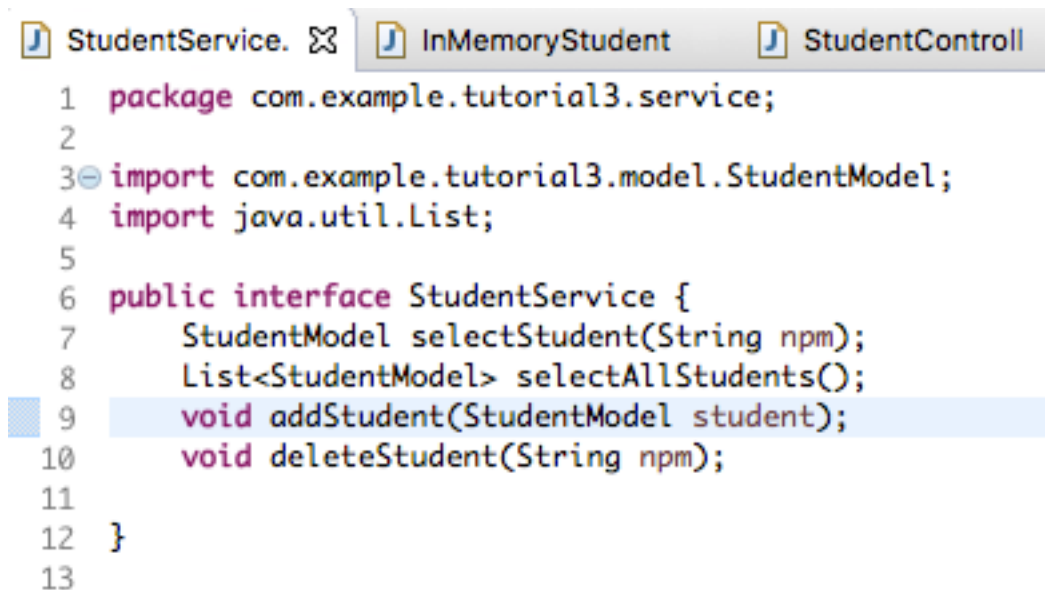
Error jika npm tidak ditemukan



2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. Misalnya, setelah melakukan add Student pada soal nomor 1, cobalah untuk melakukan delete data tersebut dengan mengakses halaman localhost:8080/student/delete/14769. Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus.
Buat method baru deleteStudent



```
StudentService. InMemoryStudent StudentControll view.html
25 }
26
27 @Override
28 public void addStudent(StudentModel student) {
29     // TODO Auto-generated method stub
30     studentList.add(student);
31 }
32
33 public void deleteStudent(String npm) {
34     // TODO Auto-generated method stub
35     // studentList.remove(student);
36     int counter = 0;
37     for (StudentModel student : studentList) {
38         if(student.getNpm().equals(npm)) {
39             studentList.remove(counter);
40             break;
41         }
42         counter++;
43     }
44 }
```



```
StudentService. InMemoryStudent StudentControll
1 package com.example.tutorial3.service;
2
3 import com.example.tutorial3.model.StudentModel;
4 import java.util.List;
5
6 public interface StudentService {
7     StudentModel selectStudent(String npm);
8     List<StudentModel> selectAllStudents();
9     void addStudent(StudentModel student);
10    void deleteStudent(String npm);
11
12 }
13
```

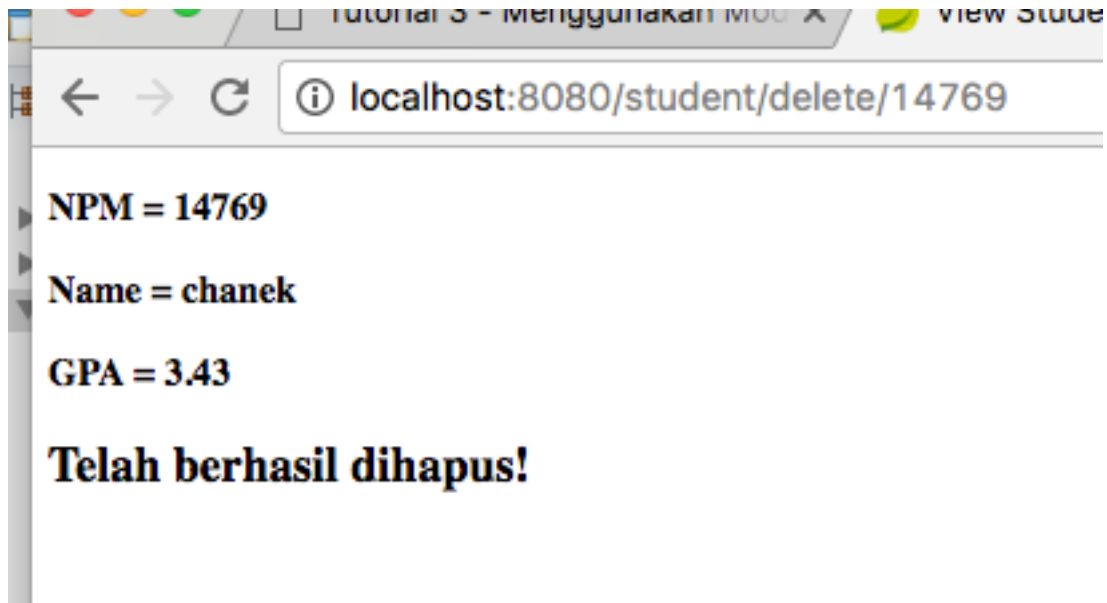
Buat method deletePath pada StudentController

```
StudentService. InMemoryStudent StudentControll view.html delete.l
54 @RequestMapping(value = {"/student/delete", "/student/delete/{npm}"})
55 public String deletePath(Model model, @PathVariable Optional<String> npm) {
56     if(npm.isPresent()) {
57         StudentModel student = studentService.selectStudent(npm.get());
58         if(student == null) {
59             return "error_page";
60         }else {
61             studentService.deleteStudent(npm.get());
62             model.addAttribute("studentName", student.getName());
63             model.addAttribute("studentNpm", student.getNpm());
64             model.addAttribute("studentGpa", student.getGpa());
65         }
66         return "delete";
67     }
68     return "error_page";
69 }
70
71
```

Halaman html yang akan ditampilkan:

```
StudentService. InMemoryStudent StudentControll view.html delete.htm
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <title>View Student by NPM</title>
5 </head>
6 <body>
7     <h3 th:text="'NPM = ' + ${studentNpm}">Student NPM</h3>
8     <h3 th:text="'Name = ' + ${studentName}">Student Name</h3>
9     <h3 th:text="'GPA = ' + ${studentGpa}">Student GPA</h3>
10    <h2>Telah berhasil dihapus!</h2>
11 </body>
12 </html>
```

Contoh halaman html yang akan ditampilkan



Lesson Learned

Dalam tutorial 3 kali ini, saya belajar cara pengimplementasian interface yang baik menggunakan MVC. Saya belajar mengenai class Model apa saja yang terdapat dalam MVC. Selain itu, saya mengimplementasikan cara menggunakan java util. Saya belajar cara penggunaan looping pada view. Selain itu, Saya melakukan requestparam dan pathvariable untuk penyimpanan data untuk sprinboot