

Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

TUTORIAL 3 - MENGGUNAKAN MODEL DAN SERVICE PADA PROJECT SPRING BOOT

Membuat service

- Mengimplementasi method Select Student

```
public StudentModel selectStudent(String npm) {
    for (StudentModel student : studentList) {
        if (student.getNpm().equals(npm)) {
            return student;
        }
    }
    return null;
}

@Override
public StudentModel selectStudent(String npm) {
    for (StudentModel student : studentList) {
        if (student.getNpm().equals(npm)) {
            return student;
        }
    }
    return null;
}
```

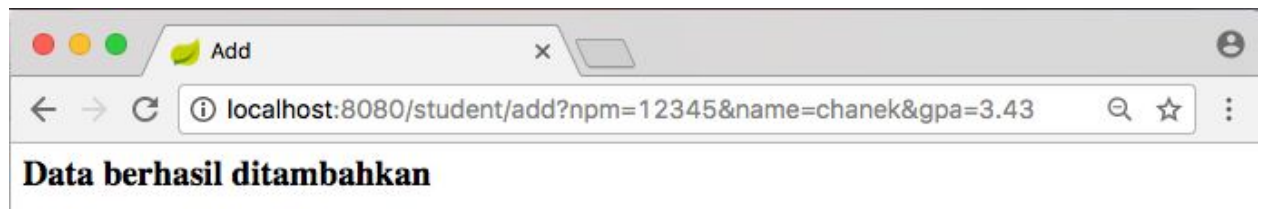
Method selectStudent(String npm) ini melakukan iterasi pada arraylist studentList untuk menemukan mahasiswa yang memiliki NPM yang sama dengan NPM pada parameter, jika terdapat mahasiswa yang NPM-nya sama, maka akan me-return mahasiswa tersebut. Sedangkan apabila tidak terdapat mahasiswa dengan NPM yang sama dengan NPM pada parameter, akan me-return NULL.

Membuat Controller dan Fungsi Add

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

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

Jawab: Tidak terjadi error, muncul page dengan tampilah seperti di bawah ini:



Nama: Mayanggita Rachmaryza Indra

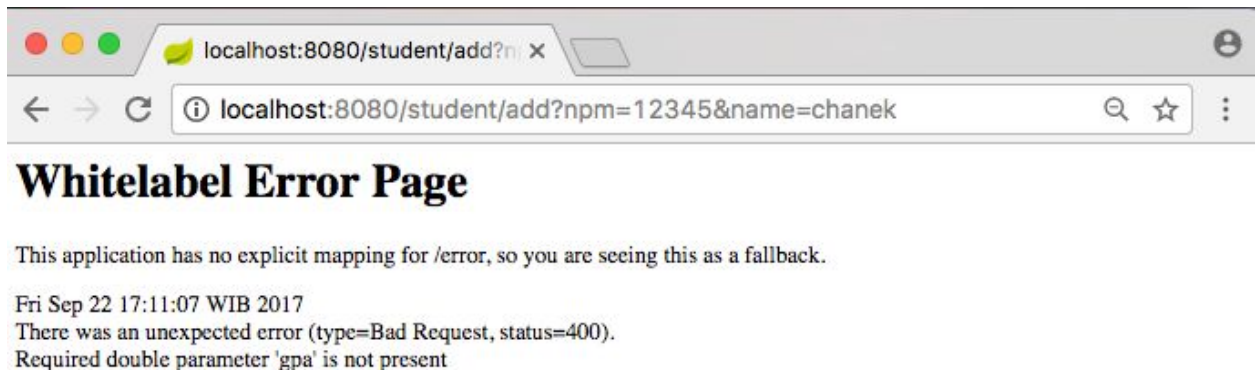
NPM: 1506732974

Kelas: B

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

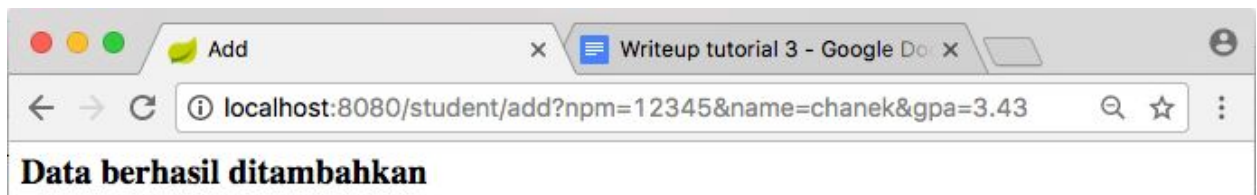
Pertanyaan 2: apakah hasilnya? Jika *error*, tuliskan penjelasan Anda.

Jawab: Terjadi Whitelabel Error Page, karena parameter gpa bersifat required untuk RequestParam GET, sehingga harus ada. Sedangkan pada url kita hanya memasukkan npm dan name.

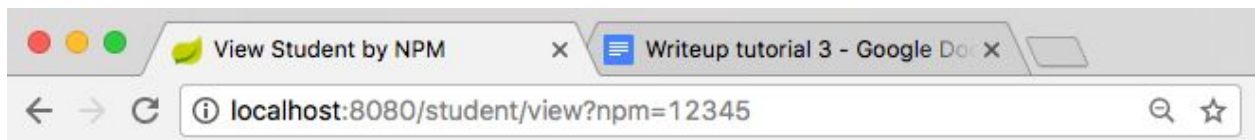


Method View by NPM

- 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?

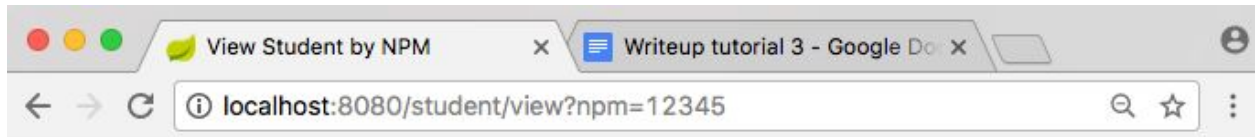
Jawab: Data student tersebut muncul, yaitu NPM = 12345; Name = chanek, GPA = 3.43 sesuai pada url yang kita masukkan yaitu localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

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



Whitelabel Error Page

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

Fri Sep 22 17:37:32 WIB 2017

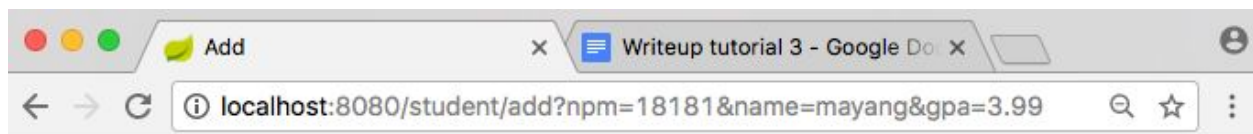
There was an unexpected error (type=Internal Server Error, status=500).

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

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

Jawab: Data student tersebut tidak muncul, karena springboot sudah dimatikan sehingga session akan diperbaharui. Jadi, data student yang ada pada soal ini tersimpan pada session sebelumnya, ketika sudah dimatikan, session akan diperbaharui menjadi kosong kembali sehingga muncul Whitelabel Error Page.

- Coba tambahkan data Student lainnya dengan NPM yang berbeda. Saya mencoba menambahkan student dengan `npm=18181; name=mayang; gpa=3.99` dengan cara membuka `localhost:8080/student/add?npm=18181&name=mayang&gpa=3.99`



Data berhasil ditambahkan

Untuk memeriksa apakah sudah ditambahkan, buka `localhost:8080/student/view?npm=18181`



NPM = 18181

Name = mayang

GPA = 3.99

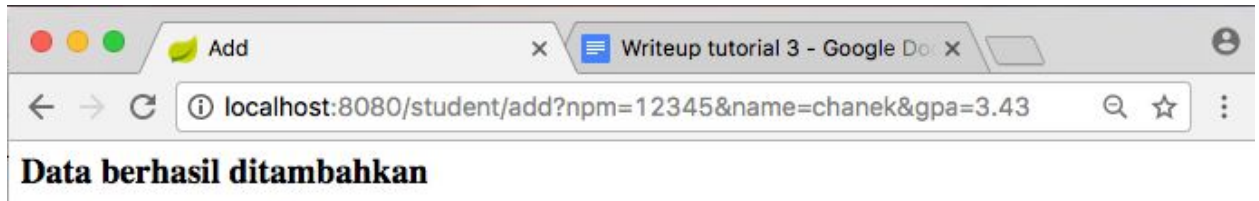
Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

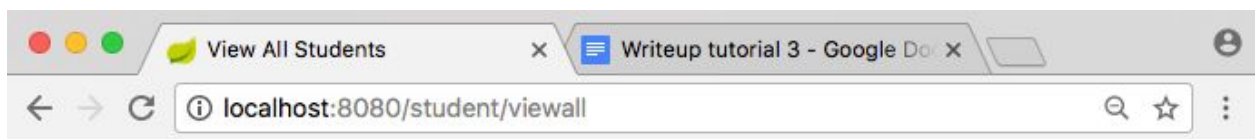
Kelas: B

Method View All

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



lalu buka localhost:8080/student/viewall



No. 1

NPM = 12345

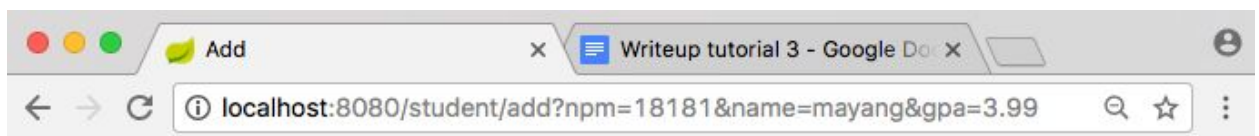
Name = chanek

GPA = 3.43

Pertanyaan 5 : apakah data Student tersebut muncul?

Ya, data student tersebut muncul, dapat dilihat pada gambar di atas

- Coba tambahkan data Student lainnya dengan NPM yang berbeda



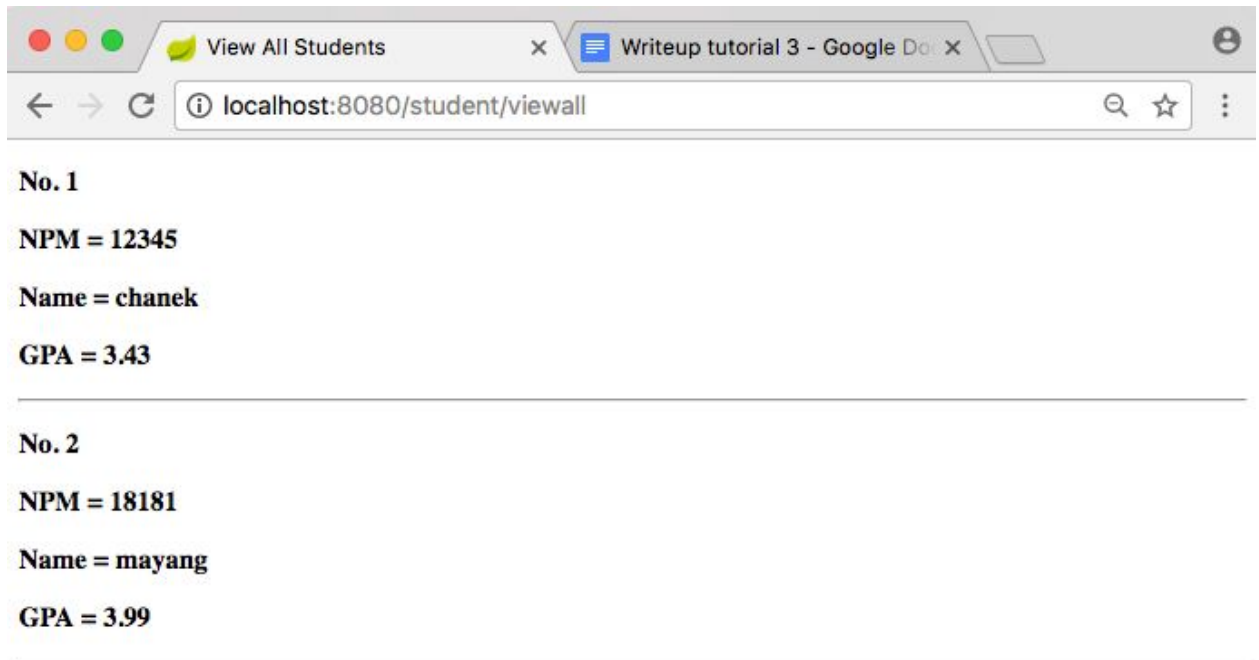
Data berhasil ditambahkan

Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

lalu buka localhost:8080/student/viewall



Pertanyaan 6 : Apakah semua data Student muncul?

Ya, data semua student muncul (lihat gambarnya di atas)

Latihan

1. Pada StudentController tambahkan sebuah *method* *view* Student dengan menggunakan Path Variable. Misalnya, Anda ingin memasukkan data seorang Student yang memiliki NPM 14769, untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman localhost:8080/student/view/14769.

Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.

Langkah:

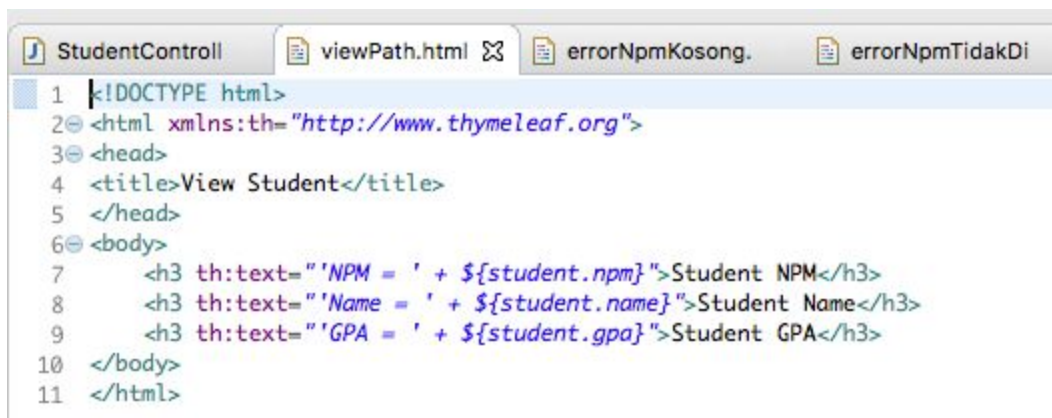
1. Buat method *viewPath* pada StudentController.java

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

2. Buat hml page untuk

- a. *viewPath.html*

→ Page untuk menampilkan NPM, *Name* dan GPA dari mahasiswa



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Student</title>
5 </head>
6 <body>
7 <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
8 <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
9 <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
10 </body>
11 </html>
```


Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

b. errorNpmKosongl.html

→ Page error apabila dalam parameter tidak ada npm atau npmnya kosong



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Student</title>
5 </head>
6 <body>
7 <h3>NPM kosong!</h3>
8 </body>
9 </html>
```

c. errorNpmTidakDitemukan.html

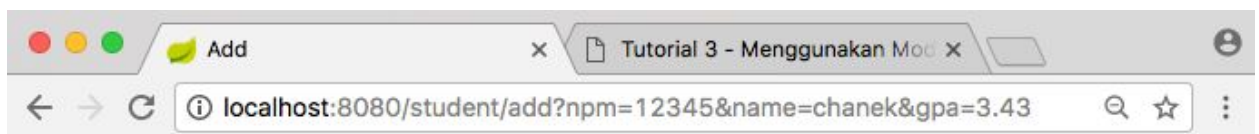
→ Page error apabila tidak ada mahasiswa yang memiliki NPM sama dengan parameter atau dengan kata lain NPM tidak ditemukan pada arraylist



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Student</title>
5 </head>
6 <body>
7 <h3>NPM Tidak Ditemukan!</h3>
8 </body>
9 </html>
```

3. Hasil

- a. Untuk menguji fitur ini, pertama coba menambahkan mahasiswa. Misalnya saya menambah mahasiswa dengan npm 12345, name chaneK dan gpa 3.43 dengan cara membuka <http://localhost:8080/student/add?npm=12345&name=chaneK&gpa=3.43>



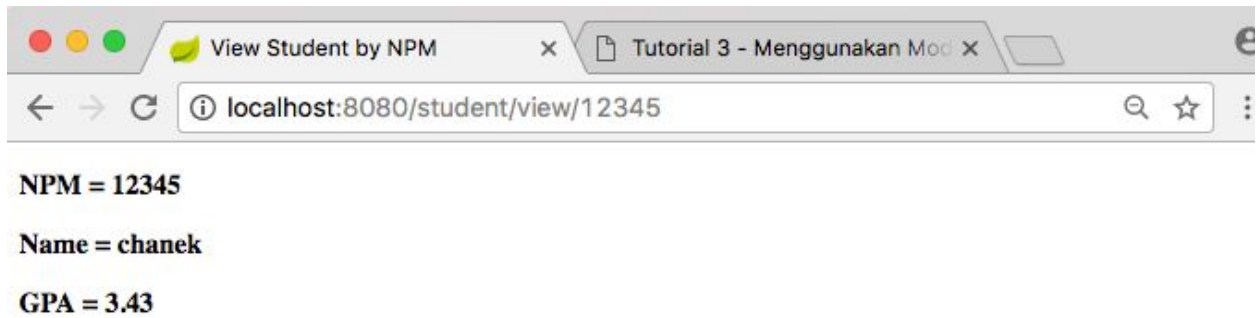
Data berhasil ditambahkan

Nama: Mayanggita Rachmaryza Indra

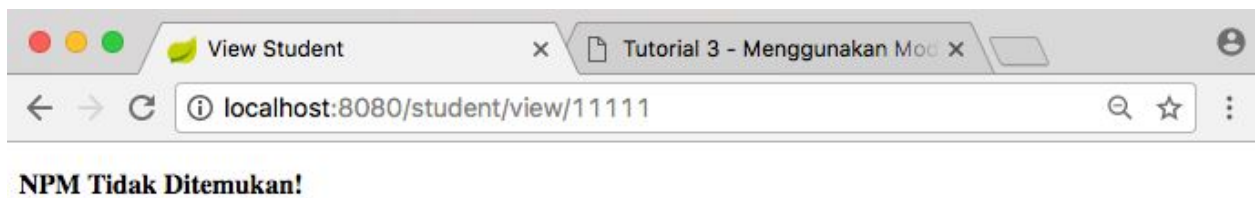
NPM: 1506732974

Kelas: B

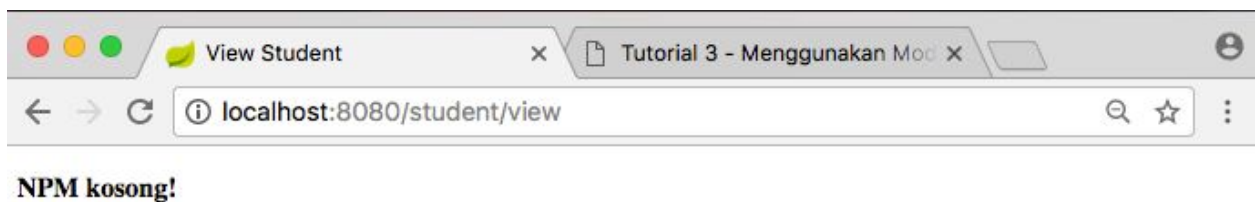
- b. Jika kita mencari npm dan npmnya ditemukan, maka akan muncul seperti di bawah ini



- c. Jika npm tidak ditemukan akan muncul seperti di bawah ini



- d. Jika tidak ada parameter npm, maka akan muncul seperti di bawah ini



Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

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. Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan dan proses *delete* dibatalkan.

Langkah:

1. Membuat method delete pada StudentController.java

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

2. Pada direktori `resources/templates`, buat file baru `delete.html` seperti berikut



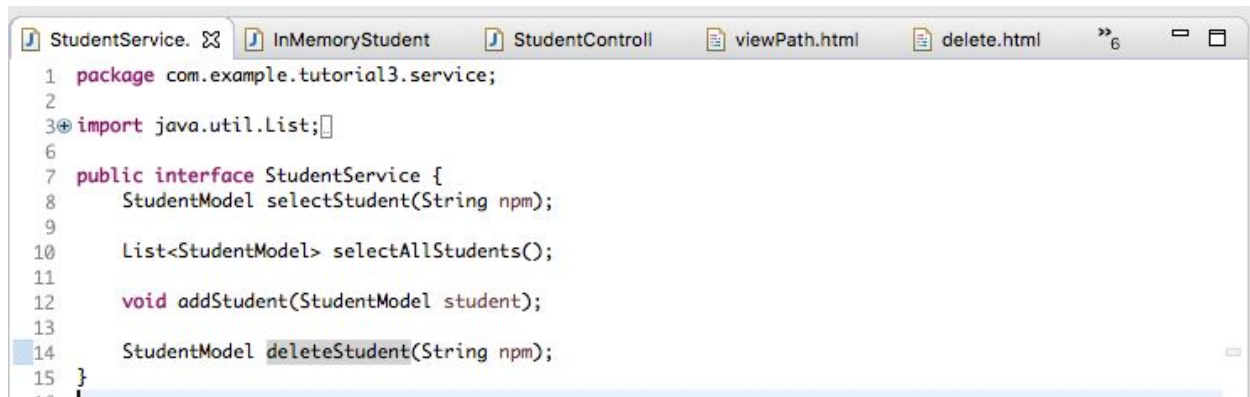
```
1 <html xmlns:th="http://www.thymeleaf.org">
2 <head>
3 <title>Delete</title>
4 </head>
5 <body>
6 <h2>Data berhasil dihapus</h2>
7 <h3 th:text="'Student dengan NPM ' + ${student.npm} + ' berhasil dihapus'">Berhasil dihapus</h3>
8 </body>
9 </html>
```

Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

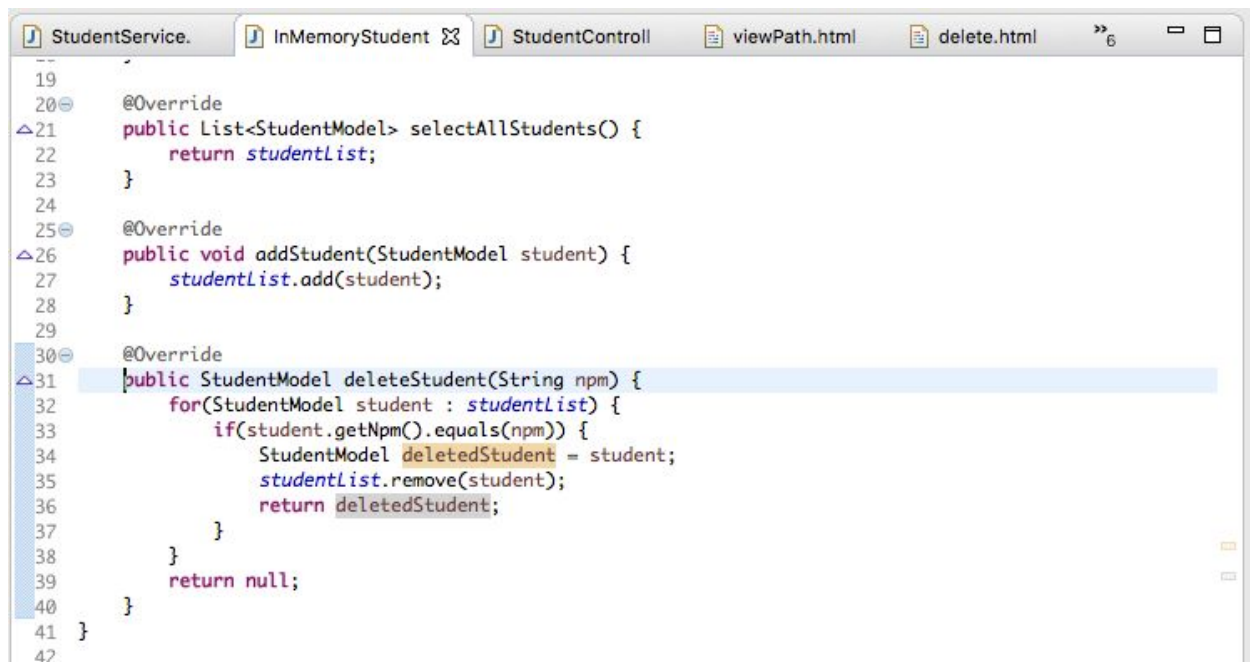
3. Pada StudentService.Java tambahkan seperti di bawah:



```
1 package com.example.tutorial3.service;
2
3 import java.util.List;
4
5
6 public interface StudentService {
7     StudentModel selectStudent(String npm);
8
9     List<StudentModel> selectAllStudents();
10
11     void addStudent(StudentModel student);
12
13     StudentModel deleteStudent(String npm);
14 }
15
```

4. Pada InMemoryStudent tambahkan method deleteStudent

Method ini pertama melakukan iterasi pada studentList untuk mencari student yang memiliki NPM yang sama dengan NPM pada parameter. Sebelum dihapus,, student tersebut disimpan ke deletedStudent. Kemudian, pada studentList saya me-remove student yang memiliki npm sama tersebut. Setelah dihapus, baru me-return deletedStudent tersebut.



```
19
20 @Override
21 public List<StudentModel> selectAllStudents() {
22     return studentList;
23 }
24
25 @Override
26 public void addStudent(StudentModel student) {
27     studentList.add(student);
28 }
29
30 @Override
31 public StudentModel deleteStudent(String npm) {
32     for(StudentModel student : studentList) {
33         if(student.getNpm().equals(npm)) {
34             StudentModel deletedStudent = student;
35             studentList.remove(student);
36             return deletedStudent;
37         }
38     }
39     return null;
40 }
41 }
42
```

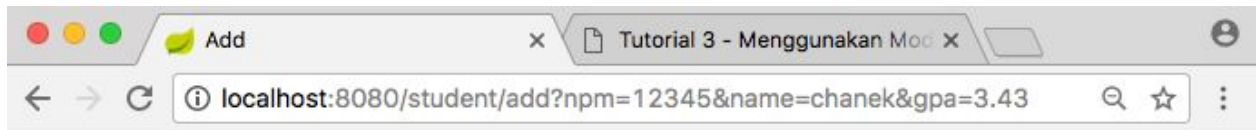
Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

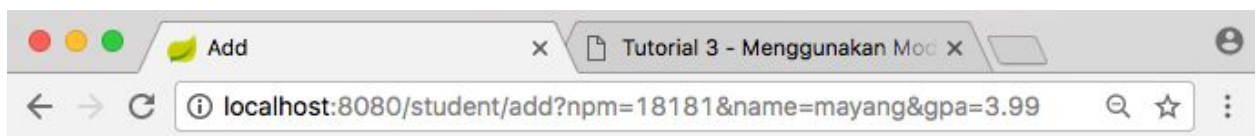
5. Hasil

- a. Untuk menguji fitur ini, pertama add student dengan membuka `localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43`



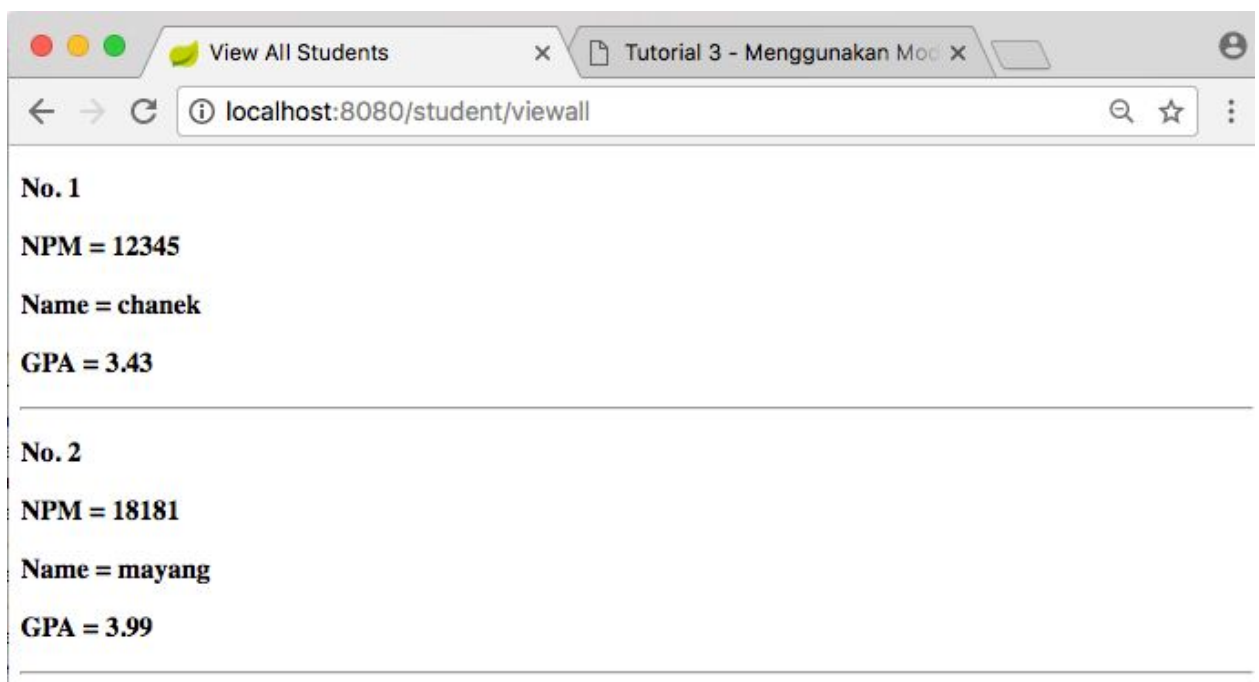
Data berhasil ditambahkan

- b. Kemudian tambahkan satu student lagi dengan cara `localhost:8080/student/add?npm=18181&name=mayang&gpa=3.99`



Data berhasil ditambahkan

- c. Setelah menambahkan 2 students, cek apakah berhasil ditambahkan atau tidak, untuk melihat semua student yang terdaftar, dapat dilakukan dengan membuka `localhost:8080/student/viewall`

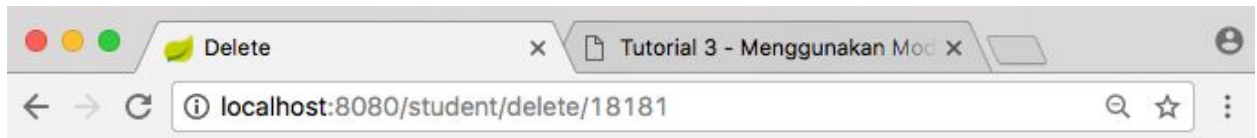


Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B

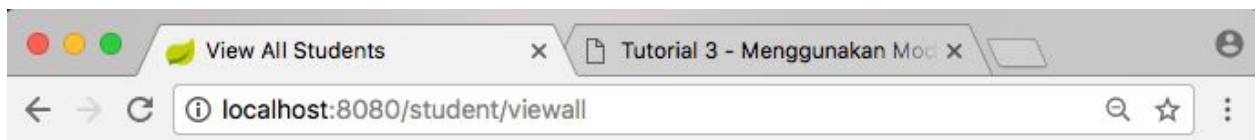
- d. Setelah itu, coba menghapus student dengan npm 1818, caranya adalah dengan membuka localhost:8080/student/delete/1818



Data berhasil dihapus

Student dengan NPM 18181 berhasil dihapus

- e. Untuk mengecek apakah sudah benar-benar terhapus, maka buka viewall student lagi, localhost:8080/student/viewall



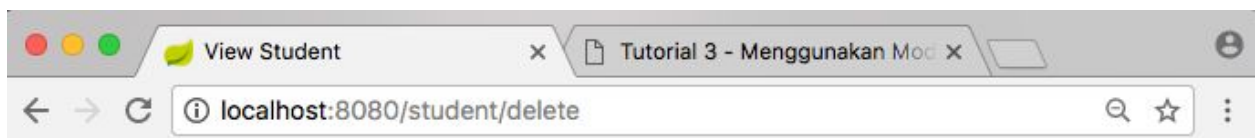
No. 1

NPM = 12345

Name = chanek

GPA = 3.43

- f. Dari gambar di atas, dapat dilihat bahwa student dengan npm 18181 sudah berhasil terhapus sehingga sudah tidak ada lagi di daftar student
- g. Apabila akan men-delete tapi tidak menuliskan parameter npm, maka akan tampil seperti di bawah ini



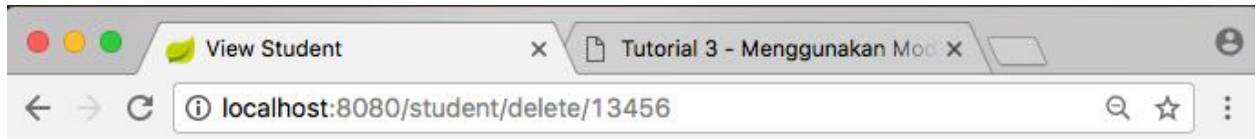
NPM kosong!

- h. Apabila akan men-delete tapi npmnya tidak ada pada daftar students, maka akan muncul seperti ini:

Nama: Mayanggita Rachmaryza Indra

NPM: 1506732974

Kelas: B



Ringkasan

Pada tutorial 3 kali ini, saya mempelajari lebih dalam tentang implementasi konsep model-view-controller (MVC) pada Spring Boot. Saya mempelajari lebih dalam tentang model (objek yang merepresentasikan dan menyimpan informasi terhadap suatu hal) dan service (berperan untuk menghubungkan antara controller dan database). Selain itu, saya juga mempelajari bagaimana cara menyimpan data untuk Spring Boot (melalui RequestParam dan PathVariable) dan juga bagaimana membuat serta menggunakan interface pada MVC.