

**a. Ringkasan dari materi yang Anda telah pelajari pada tutorial kali ini**

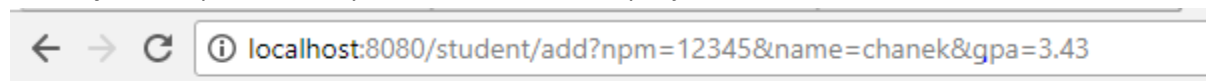
Pada tutorial ini saya belajar menggunakan model dan service pada project spring boot. Dalam mengerjakan tutorial ini saya mengimplementasi konsep model-view-controller. Saya membuat service dengan fungsi read dan create data melalui konsep MVC. Melalui tutorial ini saya menjadi tahu cara kerja penyimpanan data pada springboot, yang datanya berasal dari RequestParam GET dan pathvariable.

**b. Hasil jawaban dari setiap poin pada bagian tutorial (dapat didukung dengan screenshot )**

- **Membuat Controller dan Fungsi Add**

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

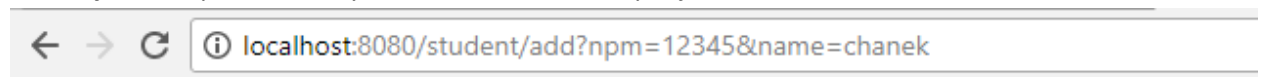
**Pertanyaan 1:** apakah hasilnya? Jika error , tuliskan penjelasan Anda.



## Data berhasil ditambahkan

Jalankan program dan buka 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.

Sat Sep 23 20:21:20 ICT 2017

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

Required double parameter 'gpa' is not present

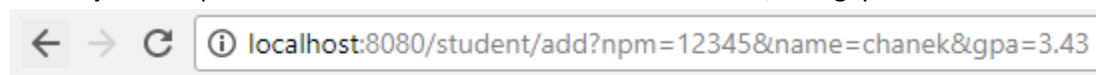
Terjadi error karena pada method add membutuhkan parameter 'gpa' untuk RequestParam GET

- **Method View by NPM**

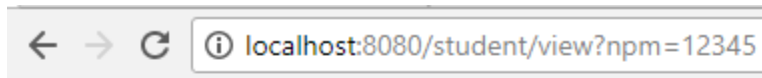
Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

lalu buka localhost:8080/student/view?npm=12345

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



## Data berhasil ditambahkan



**NPM = 12345**

**Name = chanek**

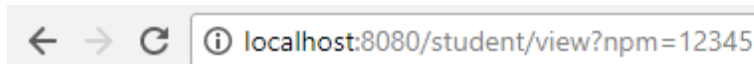
**GPA = 3.43**

Data student muncul

Coba matikan program dan jalankan kembali serta 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.

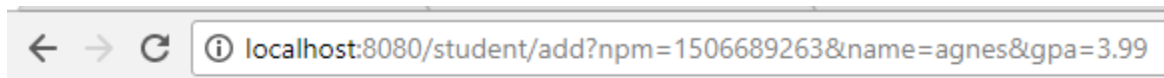
Sat Sep 23 20:31:39 ICT 2017

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

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

Data student tidak muncul karena program baru dijalankan setelah springboot diberhentikan. Sehingga data student yang tadi dimasukkan tidak bisa diakses. Ketika program diberhentikan maka data pada session tersebut akan hilang.

Coba tambahkan data Student lainnya dengan NPM yang berbeda



## Data berhasil ditambahkan

- **Method View All**

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

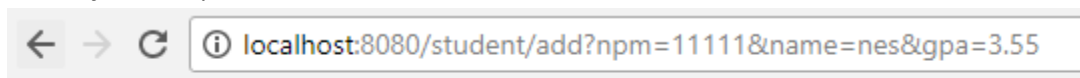
Name = chanek

GPA = 3.43

Ya muncul

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

**Pertanyaan 6:** Apakah semua data Student muncul?



## Data berhasil ditambahkan

No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 11111

Name = nes

GPA = 3.55

Semua data student muncul

### c. Method selectStudent yang Anda implementasikan

Melakukan looping pada studentList untuk mencari data yang sesuai dengan NPM yang ingin dicari

```
public StudentModel selectStudent(String npm) {  
    if(!studentList.isEmpty()){  
        for(int i=0; i<studentList.size(); i++){  
  
            if(studentList.get(i).getNpm().equalsIgnoreCase(npm)){  
                return studentList.get(i);  
            }  
        }  
    }  
    return null;  
}
```

#### d. Penjelasan fitur delete yang Anda buat pada bagian latihan

Langkah-langkah yang saya lakukan dalam membuat fitur delete ada pada bagan latihan. Untuk penjelasan membuat fitur ini saya membuat method delete student pada interfacenya terlebih dahulu. Setelah itu saya mengimplementasi bagaimana method delete student yang saya buat pada interface. Pada kelas InMemoryStudentService saya akan melakukan looping untuk mencari data student yang sesuai dengan NPM yang dicari, jika sudah ditemukan saya akan menghapus data student tersebut dari studentList dan akan mengreturn data student tersebut. Setelah itu saya membuat method delete student pada StudentController. Kemudian saya membuat view untuk tampilan ketika data student yang dicari berhasil dihapus. Jika npm yang dicari berupa null akan ditampilkan view\_empty, jika npm tidak ditemukan akan ditampilkan view\_not\_found, dan jika ditemukan akan ditampilkan view\_delete.

#### 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:

- Menambah method view pada StudentController

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

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

- Membuat view error\_not\_found

```
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>ERROR!</h3>
8     <h3>NPM is Not Found</h3>
9   </body>
10 </html>
```

- Membuat view error\_empty

```
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>ERROR!</h3>
8     <h3>NPM is Empty</h3>
9   </body>
10 </html>
```

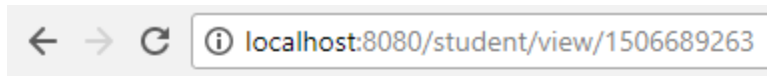
#### Hasil Fitur:

- Menambahkan student dengan npm 1506689263

← → ↻ ⓘ localhost:8080/student/add?npm=1506689263&name=agnes&gpa=3.99

## Data berhasil ditambahkan

- Melihat data yang baru dimasukkan

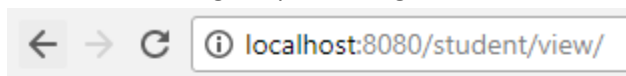


**NPM = 1506689263**

**Name = agnes**

**GPA = 3.99**

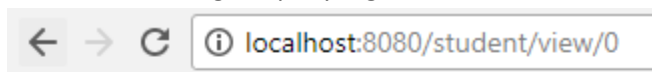
- Melihat data dengan npm kosong



**ERROR!**

**NPM is Empty**

- Melihat data dengan npm yang belum ditambahkan



**ERROR!**

**NPM is Not Found**

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:**

- Menambahkan method delete student pada interface StudentService

```
package com.example.tutorial3.service;

import java.util.List;

import com.example.tutorial3.model.StudentModel;

public interface StudentService {
    StudentModel selectStudent(String npm);

    List<StudentModel> selectAllStudents();

    void addStudent(StudentModel student);

    StudentModel deleteStudent (String npm);
}
```

- Mengimplementasi method delete student di kelas InMemoryStudentService

```
@Override
public StudentModel deleteStudent(String npm) {
    for(int i=0; i<studentList.size(); i++){
        if(studentList.get(i).getNpm().equalsIgnoreCase(npm)){
            StudentModel student = studentList.get(i);
            studentList.remove(i);
            return student;
        }
    }
    return null;
}
```

- Membuat method delete student pada StudentController

```

@RequestMapping(value={"/student/delete", "/student/delete/{npm}"})
public String studentDeletePath (@PathVariable Optional<String> npm, Model model)
{
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());

        if(student == null){
            return "error_not_found";
        } else{
            StudentModel deletedStudent = studentService.deleteStudent(npm.get());
            model.addAttribute("student", deletedStudent);
            return "view_delete";
        }
    } else {
        return "error_empty";
    }
}

```

- Membuat view untuk tampilan jika data student berhasil dihapus

```

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>This student's data is deleted</h3>
8 <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
9 <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
10 <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
11 </body>
12 </html>

```

#### Hasil Fitur:

- Mencoba menghapus data student yang ditambahkan pada nomor 1

← → ↻ ⓘ localhost:8080/student/delete/1506689263

**This student's data is deleted**

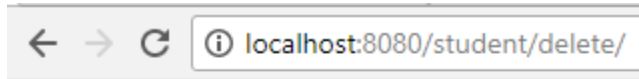
**NPM = 1506689263**

**Name = agnes**

**GPA = 3.99**



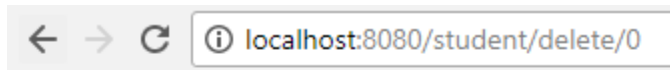
- Menghapus data student dengan npm kosong



**ERROR!**

**NPM is Empty**

- Menghapus data student dengan npm yang tidak ditemukan



**ERROR!**

**NPM is Not Found**