

Tutorial 2

APAP

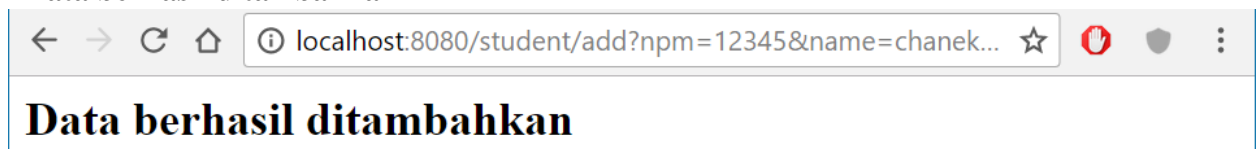
Tutorial

1. Jalankan program dan buka

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

Pertanyaan 1:

- Apakah hasilnya? Jika error, tuliskan penjelasan Anda. **Tidak error. Tampil tulisan “Data berhasil ditambahkan”**



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

Pertanyaan 2:

- Apakah hasilnya? Jika error, tuliskan penjelasan Anda. **Error. Karena parameter gpa nya tidak ada padahal required.**



2.

Method selectStudent():

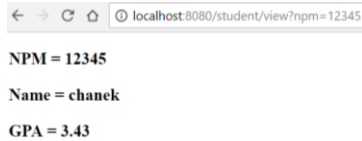
```
@Override
public StudentModel selectStudent(String npm) {
    for(int i = 0; i < studentList.size(); i++){
        if(studentList.get(i).getNpm().equalsIgnoreCase(npm)){
            return studentList.get(i);
        }
    }
    return null;
}
```

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? **Muncul**




NPM = 12345
Name = chanek
GPA = 3.43

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 muncul. Karena saat view request data melalui Model, model tersebut tidak exist Karena list nya kosong.**



Whitelabel Error Page

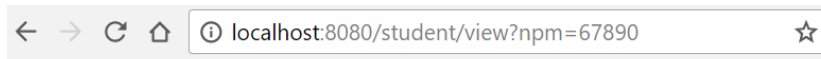
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Sep 23 08:25:57 ICT 2017

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

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

Coba tambahkan data Student lainnya dengan NPM yang berbeda.

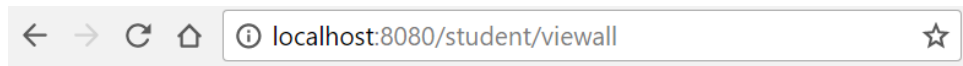


NPM = 67890
Name = tampan
GPA = 4.01

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? **Muncul**



No. 1

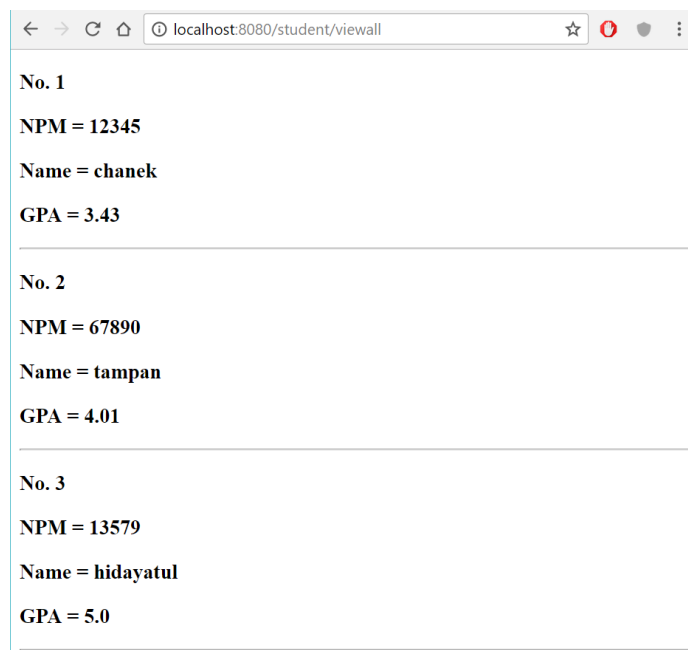
NPM = 12345

Name = chanek

GPA = 3.43

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

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



Latihan

1. Pada **StudentController** tambahkan sebuah method view Student dengan menggunakan **Path Variable**. Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman error yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.

Langkah:

- Buat method baru terlebih dahulu yang menerima Path Variable

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

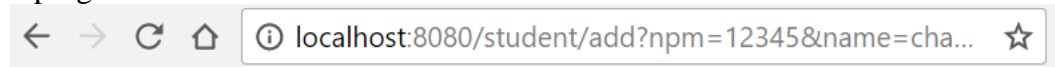
- Buat 2 halaman yang akan di-return jika terjadi kondisi error. Yaitu yang pertama pada saat NPM tidak diberikan atau NPM tidak ditemukan

view.html	npmnull.html	npmnotfound....
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>NPM belum ditulis!</h3> 8 </body> 9 </html>	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>NPM tidak ditemukan!</h3> 8 </body> 9 </html>	

- Implementasi method viewStudentNPMPath

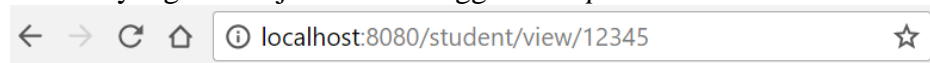
```
@RequestMapping(value={"/student/view", "/student/view/{npm}"})  
public String viewStudentNPMPath (@PathVariable Optional<String> npm, Model model) {  
    if(!npm.isPresent()) return "npmnull";  
    else {  
        StudentModel student = studentService.selectStudent(npm.get());  
        if(student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

- Jalankan program. Tambahkan satu entri terlebih dahulu.



Data berhasil ditambahkan

- Coba lihat data yang baru saja dientri menggunakan *path variable*

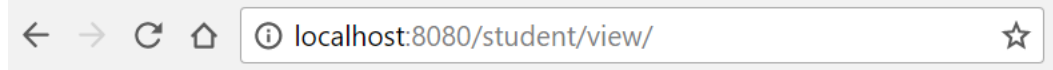


NPM = 12345

Name = chanek

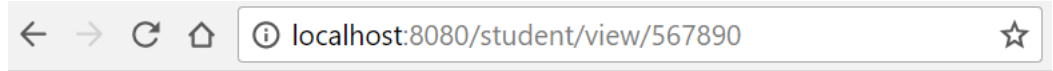
GPA = 3.43

- Jika NPM tidak disertakan, maka akan muncul halaman berikut



NPM belum ditulis!

- - Jika NPM tidak ditemukan, maka akan muncul halaman berikut



NPM tidak ditemukan!

2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. 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.

- Buat method baru terlebih dahulu yang menerima Path Variable

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

- - Buat 2 halaman yang akan di-return jika terjadi kondisi error. Yaitu yang pertama pada saat NPM tidak diberikan atau NPM tidak ditemukan

view.html	npmnull.html	npmnotfound....	view.html	npmnull.html	npmnotfound....
1<!DOCTYPE html>	1<!DOCTYPE html>	1<!DOCTYPE html>	1<!DOCTYPE html>	1<!DOCTYPE html>	1<!DOCTYPE html>
2<html xmlns:th="http://www.thymeleaf.org">	2<html xmlns:th="http://www.thymeleaf.org">	2<html xmlns:th="http://www.thymeleaf.org">	2<html xmlns:th="http://www.thymeleaf.org">	2<html xmlns:th="http://www.thymeleaf.org">	2<html xmlns:th="http://www.thymeleaf.org">
3<head>	3<head>	3<head>	3<head>	3<head>	3<head>
4<title>View Student by NPM</title>	4<title>View Student by NPM</title>	4<title>View Student by NPM</title>	4<title>View Student by NPM</title>	4<title>View Student by NPM</title>	4<title>View Student by NPM</title>
5</head>	5</head>	5</head>	5</head>	5</head>	5</head>
6<body>	6<body>	6<body>	6<body>	6<body>	6<body>
7<h3>NPM belum ditulis!</h3>	7<h3>NPM belum ditulis!</h3>	7<h3>NPM belum ditulis!</h3>	7<h3>NPM tidak ditemukan!</h3>	7<h3>NPM tidak ditemukan!</h3>	7<h3>NPM tidak ditemukan!</h3>
8</body>	8</body>	8</body>	8</body>	8</body>	8</body>
9</html>	9</html>	9</html>	9</html>	9</html>	9</html>

- Saya menambahkan method deleteStudent pada class InMemoryStudentService.java

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

- - Namun, terjadi error karena di interface StudentService belum terimplementasi method tersebut. Jadi, pada interface tersebut akan saya tambahkan.

```
1 package com.example.tutorial3.service;
2
3 import java.util.List;
4
5
6
7 public interface StudentService {
8     StudentModel selectStudent(String npm);
9
10    List<StudentModel> selectAllStudents();
11
12    void addStudent(StudentModel student);
13
14    void deleteStudent(StudentModel student);
15
16 }
17
```

- Kemudian, saya membuat halaman untuk menunjukkan bahwa proses delete berhasil.

```
InMemoryStudentService.java StudentController.java view.html StudentService.java deletesuccess.html
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>Delete Student by NPM</title>
5 </head>
6 <body>
7 <h3>Student dengan detail:</h3>
8 <h3 th:text="'NPM = ' + ${studentnpm}">Student NPM</h3>
9 <h3 th:text="'Name = ' + ${studentnama}">Student Name</h3>
10 <h3 th:text="'GPA = ' + ${studentgpa}">Student GPA</h3>
11 <h3>Telah berhasil dihapus!</h3>
12 </body>
13 </html>
```

-
- Lalu implementasi method pada StudentController

```
59 @RequestMapping(value={"/student/delete", "/student/delete/{npm}"})
60 public String deleteStudentNPMPath (@PathVariable Optional<String> npm, Model model) {
61     if(!npm.isPresent()) return "npmnull";
62     else {
63         StudentModel student = studentService.selectStudent(npm.get());
64         if(student == null) return "npmnotfound";
65         else {
66             model.addAttribute("studentnama", student.getName());
67             model.addAttribute("studentnpm", student.getNpm());
68             model.addAttribute("studentgpa", student.getGpa());
69             studentService.deleteStudent(student);
70             return "deletesuccess";
71         }
72     }
73 }
74 }
```

-
- Jalankan program. Tambahkan beberapa student

← → ↻ 🏠 ⓘ localhost:8080/student/add?npm=12345&name=cha... ☆

Data berhasil ditambahkan

← → ↻ 🏠 ⓘ localhost:8080/student/add?npm=67890&name=tam... ☆

Data berhasil ditambahkan

-
- Lihat seluruh students

← → ↻ 🏠 ⓘ localhost:8080/student/viewall ☆

No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

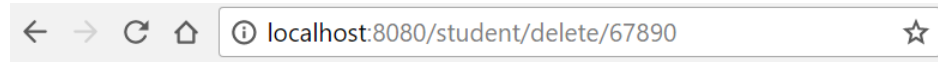
NPM = 67890

Name = tampan

GPA = 4.01

○

- Delete salah satu student. Contohnya saya akan delete student tampan melalui NPM nya (67890)



Student dengan detail:

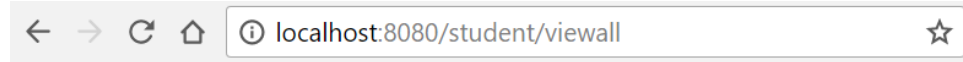
NPM = 67890

Name = tampan

GPA = 4.01

Telah berhasil dihapus!

- ➤ Saat melihat kembali daftar students, student tampan tidak ada lagi karena sudah terhapus



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

- ➤ Penjelasan Fitur: Method delete yang saya implementasi yaitu menggunakan fitur remove() pada interface List, dengan gabungan implementasi Spring agar bisa men-delete entri melalui URL.

Ringkasan:

Yang telah saya pelajari melalui tutorial 3 APAP adalah saya telah mengetahui bagaimana class Model dalam MVC. Selain itu, saya juga mempelajari bagaimana menggunakan interface dengan MVC. Selain itu, saya juga mempelajari bagaimana data bisa tersimpan pada Springboot yang bisa mengkolaborasikan List dengan Springboot agar bisa mengakses melalui URL.