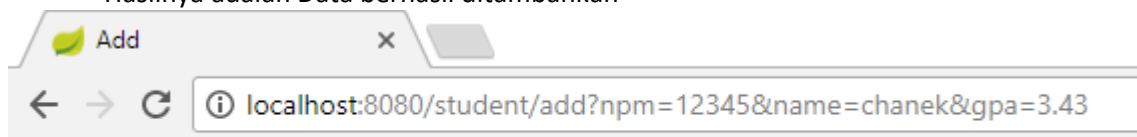


Julio Jaya Handoyo
1506757440
APAP - A

Hal yang saya pelajari dari tutorial ini adalah implementasi konsep *model-view-controller* menggunakan Spring Boot terutama konsep *model* dan *service*. *Model* merupakan representasi sebuah objek yang memiliki informasi-informasi(atribut) tertentu. Berbeda dengan *model*, *service* merupakan penghubung antara *controller* dan *database*. Pada *service layer*, ada *business logic* yang digunakan untuk mengolah data, pengolahan data ini berupa penambahan data, penghapusan data, pengambilan data.

1. Apakah hasilnya? Jika error, tuliskan penjelasan Anda.

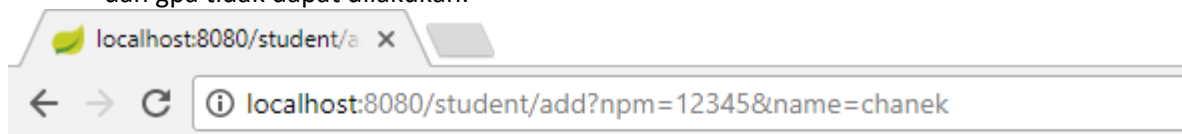
Hasilnya adalah Data berhasil ditambahkan



Data berhasil ditambahkan

2. Apakah hasilnya? Jika error, tuliskan penjelasan Anda.

Hasilnya adalah Whitelabel Error Page. Hal ini terjadi karena tidak ada parameter gpa yang dimasukkan sehingga menyebabkan penyimpanan objek yang membutuhkan npm, nama, dan gpa tidak dapat dilakukan.



Whitelabel Error Page

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

Fri Sep 22 20:31:35 ICT 2017

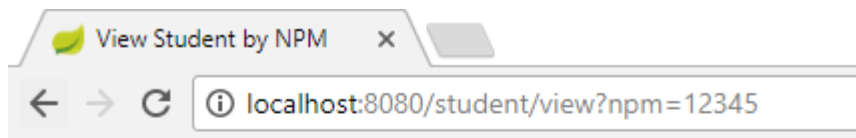
There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

Method View by NPM

3. Apakah data Student tersebut muncul? Jika tidak, mengapa?

Ya data student dengan npm 12345 muncul

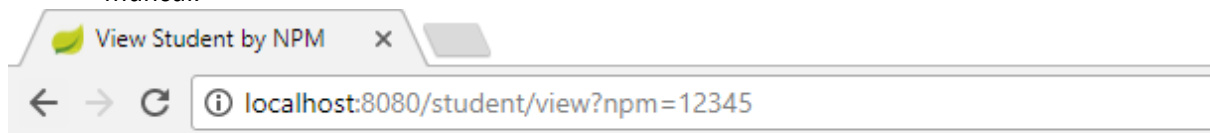


NPM = 12345

Name = chanek

GPA = 3.43

4. Apakah data Student tersebut muncul? Jika tidak, mengapa?
Data student tersebut tidak muncul karena belum ada data yang ditambah ke list sehingga ketika ingin melihat data student dengan npm 12345 pada list, data yang diminta tidak muncul.



Whitelabel Error Page

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

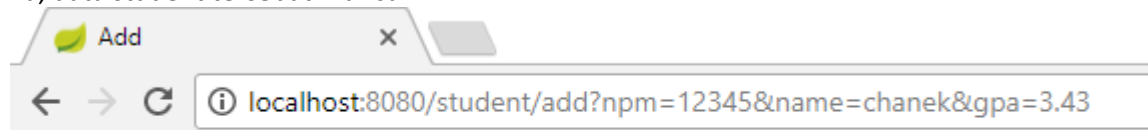
Fri Sep 22 20:50:11 ICT 2017

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

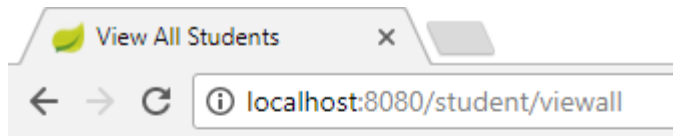
Exception evaluating SpringEL expression: "student.npm" (view:7)

Method View All

5. Apa data Student tersebut muncul?
Ya, data student tersebut muncul.



Data berhasil ditambahkan



No. 1

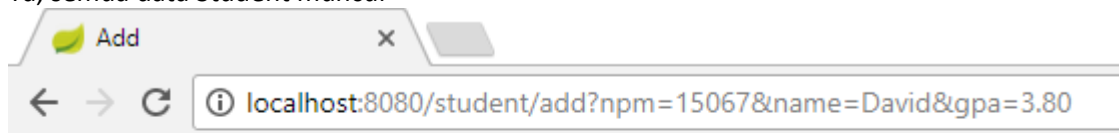
NPM = 12345

Name = chanek

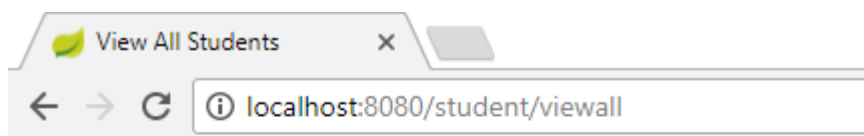
GPA = 3.43

6. Apakah semua data Student muncul?

Ya, semua data Student muncul



Data berhasil ditambahkan



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 15067

Name = David

GPA = 3.8

Method selectStudent

```
public StudentModel selectStudent(String npm) {  
    // TODO Auto-generated method stub  
    for(StudentModel s : studentList) {  
        if(s.getNpm().equals(npm)) {  
            return s;  
        }  
    }  
    return null;  
}
```

Fitur Delete

Langkah pertama yang saya lakukan untuk mengimplementasikan fitur *Delete* adalah menambah method berikut pada interface StudentService.

```
StudentModel deleteStudent(String npm);
```

Selanjutnya, saya implementasikan method deleteStudent yang sebelumnya sudah di override dari interface StudentService pada class InMemoryStudentService.

```
@Override  
public StudentModel deleteStudent(String npm) {  
    // TODO Auto-generated method stub  
    for(StudentModel s : studentList) {  
        if(npm.equals(s.getNpm())) {  
            studentList.remove(s);  
        }  
    }  
    return null;  
}
```

Selanjutnya, saya mengimplementasikan fitur delete pada Controller. Fitur ini berfungsi untuk melakukan penghapusan Student berdasarkan NPM. Jika nomor NPM tidak berikan atau tidak ditemukan, maka akan menampilkan halaman yang berisi informasi bahwa nomor NPM kosong atau ditemukan dan proses delete dibatalkan.

```
@RequestMapping(value = {"/student/delete", "/student/delete/{npm}"})  
public String deletePath(@PathVariable Optional<String> npm, Model model) {  
    if(npm.isPresent()) {  
        StudentModel student = studentService.selectStudent(npm.get());  
        if(student != null) {  
            student = studentService.deleteStudent(npm.get());  
            return "delete";  
        } else {  
            model.addAttribute("error", "NPM " + npm.get() + " tidak ditemukan");  
            model.addAttribute("message", "Proses delete dibatalkan");  
            return "error";  
        }  
    }  
    model.addAttribute("error", "NPM kosong");  
    model.addAttribute("message", "Proses delete dibatalkan");  
    return "error";  
}
```

Saya juga membuat file delete.html pada resources/template. Jika penghapusan berhasil maka akan menampilkan Data berhasil dihapus

```
1 <html>
2   <head>
3     <title>Delete</title>
4   </head>
5   <body>
6     <h2>Data berhasil dihapus</h2>
7   </body>
8 </html>
```

Saya juga membuat file error.html. File ini digunakan ketika ingin melihat data mahasiswa namun npm yang digunakan untuk mencari data tidak ditemukan atau npm kosong. Begitu pula ketika ingin menghapus data. Ketika npm tidak ditemukan atau npm kosong akan mengeluarkan pesan error ini.

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Error Page</title>
5   </head>
6   <body>
7     <h3 th:text="${error}">Error</h3>
8     <h3 th:text="${message}">Message</h3>
9   </body>
10 </html>
```