

Tutorial 3: Menggunakan Model dan Service pada Project Spring Boot

Jihan Adibah
ADPAP-C
1506689515

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

Di tutorial kali ini, saya mempelajari lebih dalam mengenai MVC model, dan hal-hal yang telah dipelajari seperti request parameter dan path variable. Di tutorial ini saya belajar cara menggunakan model dan service pada spring boot, dan dalam praktiknya juga ada penggunaan java util seperti List sebagai penyimpanan data. Di tutorial ini saya juga memahami mengenai penggunaan interface, dan bahwa dengan menggunakan interface, akan mempermudah jika data diambil dengan metode lain.

b. Hasil jawaban dari setiap poin pada bagian tutorial

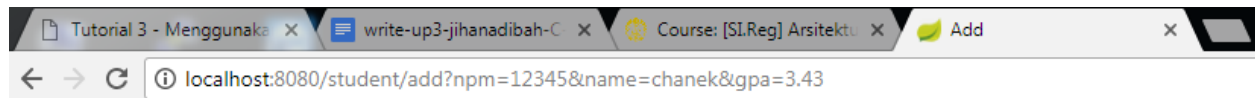
Membuat Controller dan Fungsi Add

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.

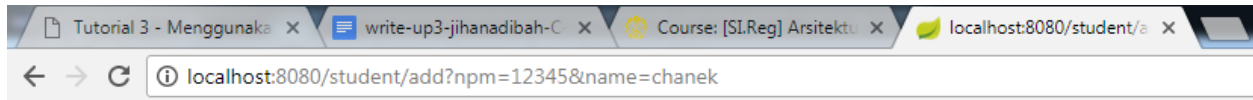


Data berhasil ditambahkan

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

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

Whitelabel error page, karena tidak ada parameter gpa, dan parameter gpa telah didefinisikan required.



Whitelabel Error Page

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

Fri Sep 22 15:29:33 ICT 2017

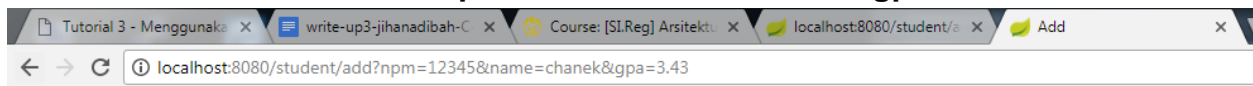
There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

Method View by NPM

Jalankan program dan buka:

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



Data berhasil ditambahkan

lalu buka

localhost:8080/student/view?npm=12345



NPM = 12345

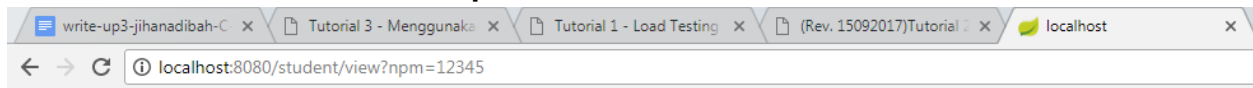
Name = chanek

GPA = 3.43

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

Muncul

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 23:37:31 ICT 2017

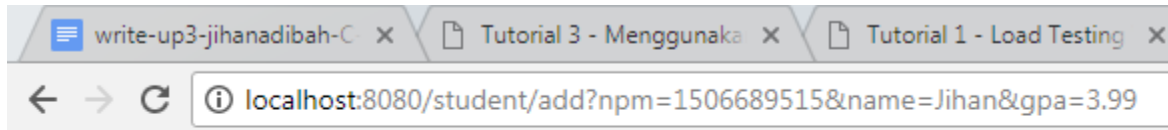
There was an unexpected error (type=Internal Server Error, status=500).

An error happened during template parsing (template: "class path resource [templates/view.html]")

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

Tidak karena belum add setelah di run.

Coba tambahkan data Student lainnya dengan NPM yang berbeda.

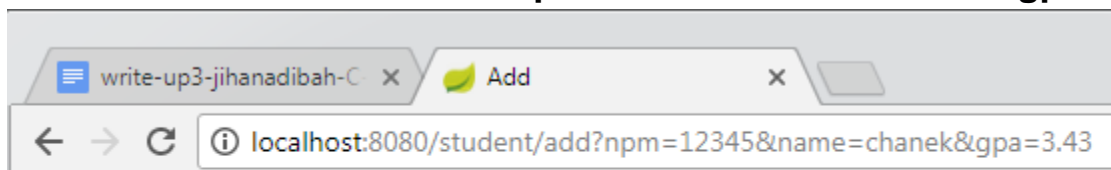


Data berhasil ditambahkan

Method View All

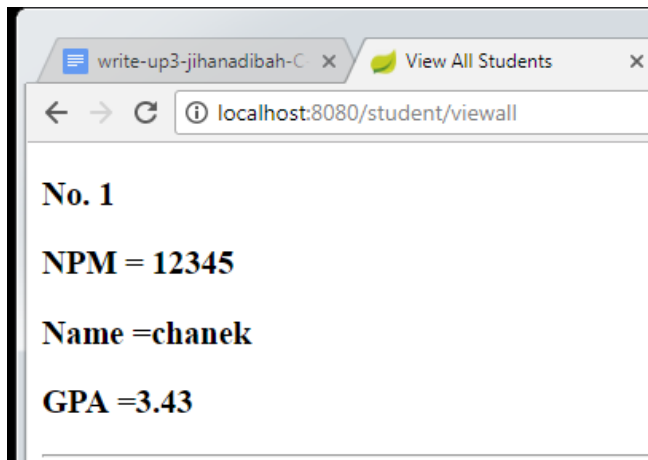
Jalankan program dan buka

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



Data berhasil ditambahkan

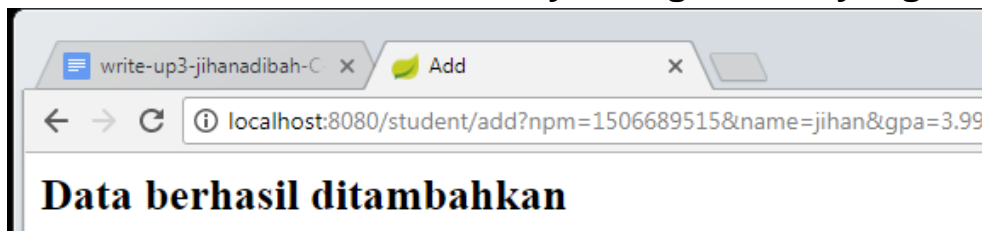
lalu buka localhost:8080/student/viewall,



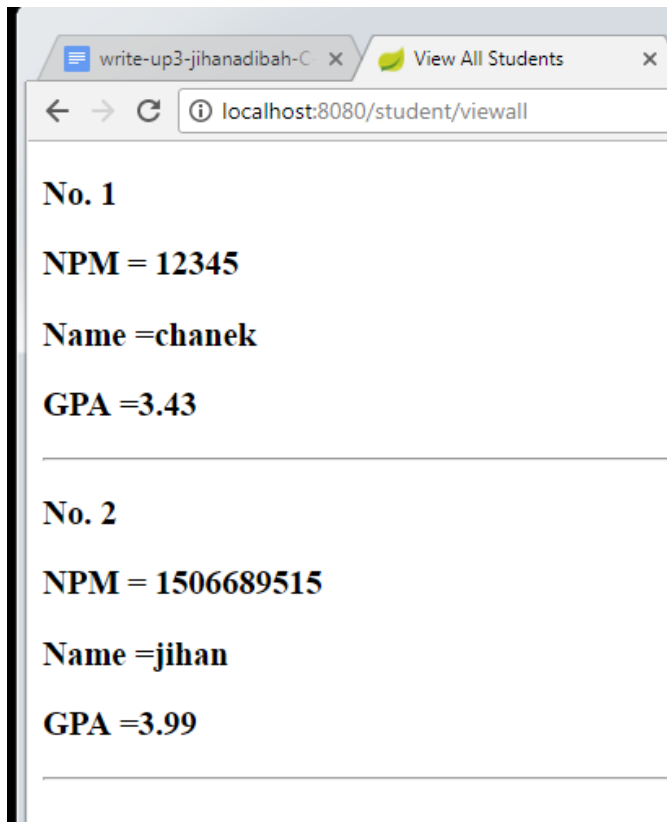
Pertanyaan 5: apakah data Student tersebut muncul?

Ya, muncul semua data student, dalam kasus ini hanya Chanek.

Tambahkan data Student lainnya dengan NPM yang berbeda,



lalu buka localhost:8080/student/viewall,



Pertanyaan 6: Apakah semua data Student muncul?

Ya, ketika viewall, data semua student muncul.

c. Method selectStudent yang Anda implementasikan

```
@Override
    public StudentModel selectStudent(String npm) {

        //iterasi untuk mencari student dengan npm sesuai
        for(int i=0; i<studentList.size(); i++) {
            if(studentList.get(i).getNpm().equals(npm)) {
                return studentList.get(i);
            }
        }
        return null;
    }
```

Latihan

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

Menambahkan method viewStudent di StudentController

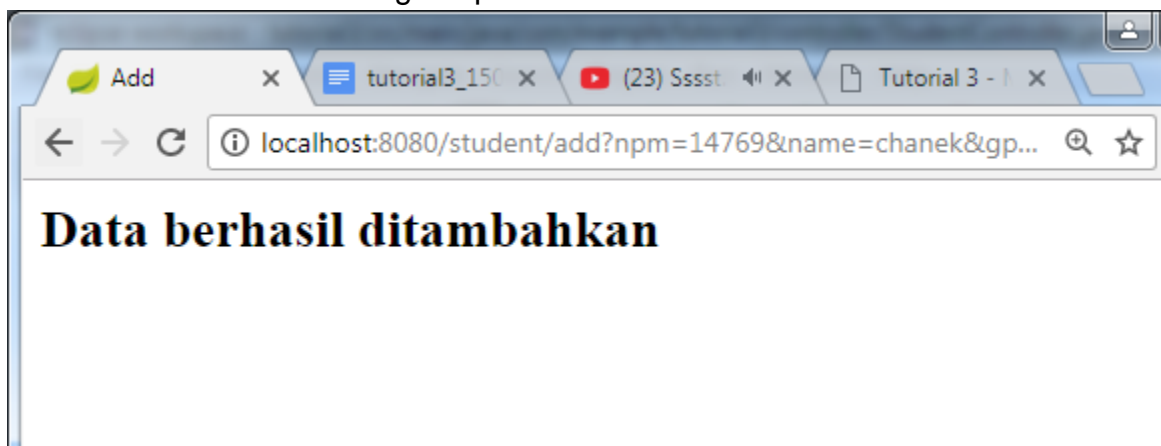
```
@RequestMapping(value = {"/student/view","student/view/{npm}}")
public String viewStudent(@PathVariable Optional<String> npm, Model model) {

    //condition untuk cek apakah npm null
    if (npm.isPresent()) {

        //condition untuk cek apakah ada student dengan npm yang dicari
        if (studentService.selectStudent(npm.get())==null) {

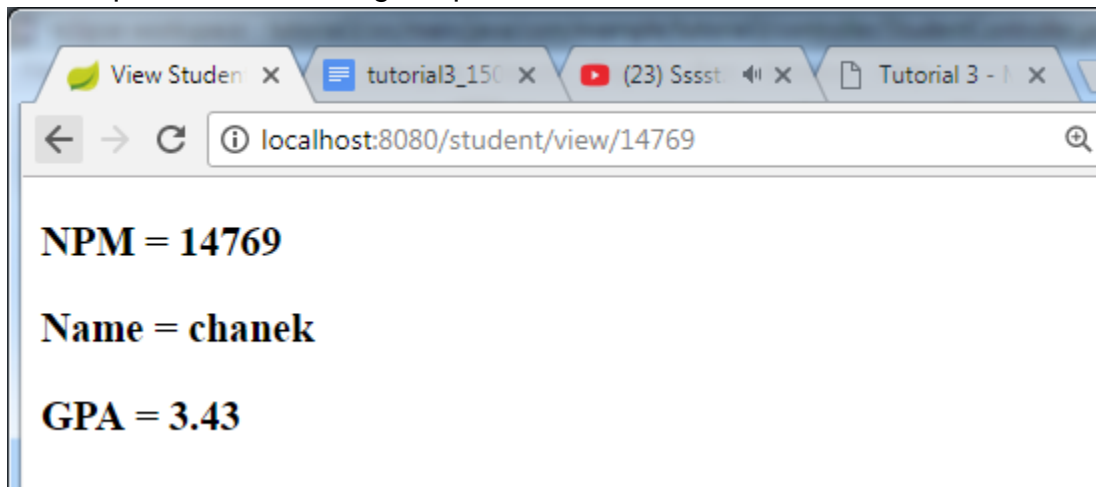
            model.addAttribute("errorMessage","NPM NOT FOUND");
            return "errorPage";
        }
        else {
            StudentModel student = studentService.selectStudent(npm.get());
            model.addAttribute("student" , student);
            return "view";
        }
    }
    else {
        model.addAttribute("errorMessage", "NPM CAN NOT BE EMPTY");
        return "errorPage";
    }
}
```

Menambahkan student dengan npm 14769

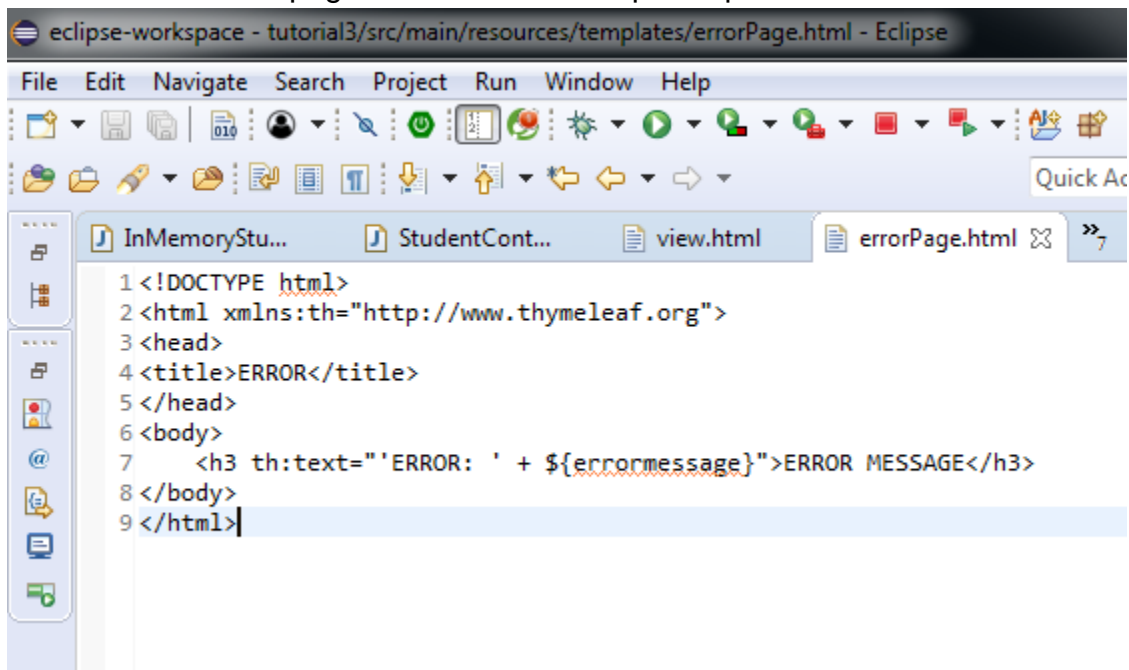


localhost:8080/student/view/14769

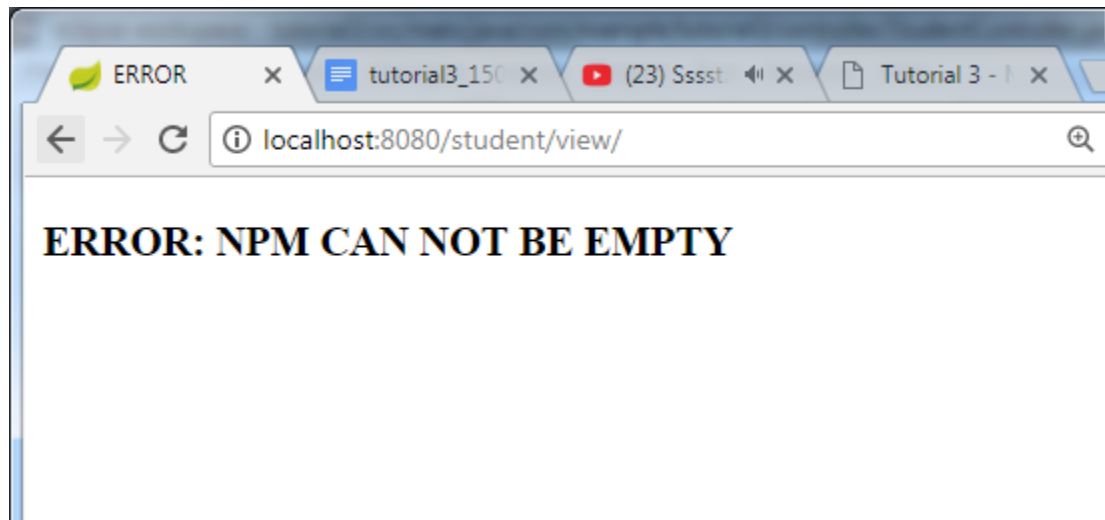
Menampilkan student dengan npm 14769



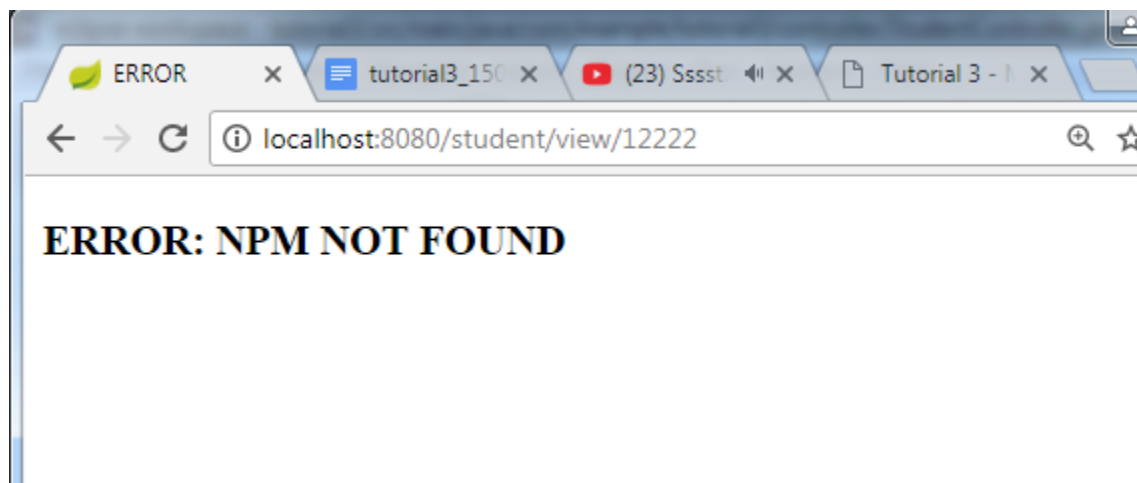
Membuat view errorpage.html untuk menampilkan pesan error



Jika nomor NPM tidak diberikan

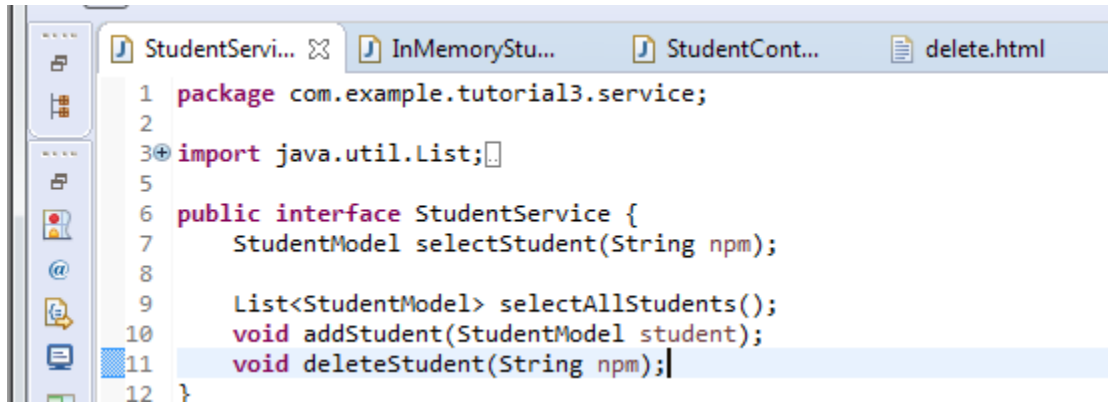


Jika nomor NPM tidak ditemukan



2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM.

Menambahkan method di interface studentservice



```
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    void addStudent(StudentModel student);
11    void deleteStudent(String npm);
12 }
```

Menambahkan method deleteStudent di service inmemorystudentservice

```
30 }
31
32 @Override
33 public void deleteStudent(String npm) {
34     StudentModel student = selectStudent(npm);
35     studentList.remove(student);
36 }
37 }
```

Menambahkan method delete di controller StudentController

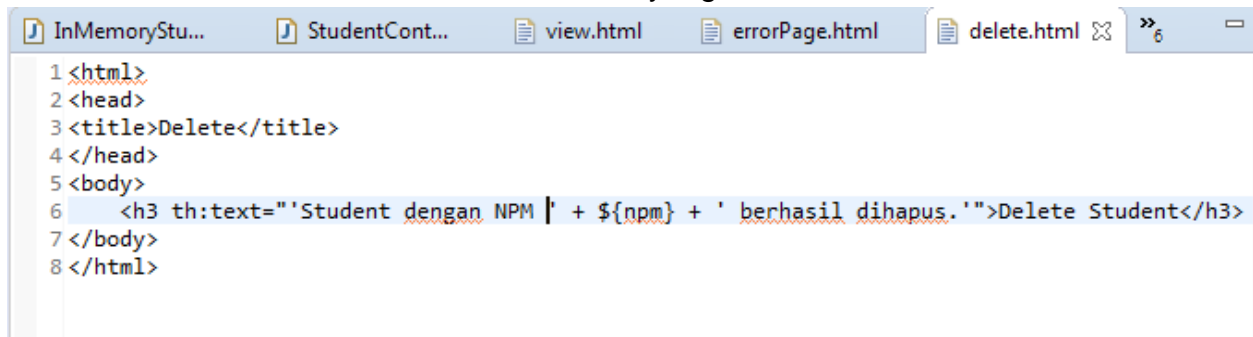
```
@RequestMapping(value = {"/student/delete","student/delete/{npm}"})
public String delete(@PathVariable Optional<String> npm, Model model) {
    //condition untuk cek npm tidak null
    if (npm.isPresent()) {

        //condition untuk cek npm yang dicari ada
        if (studentService.selectStudent(npm.get())==null) {
            model.addAttribute("errorMessage","NPM NOT FOUND, DELETE CANCELLED");
            return "errorPage";
        }

        //jika npm ditemukan, hapus student dengan npm yang sesuai
        else {
            studentService.deleteStudent(npm.get());
            model.addAttribute("npm" , npm.get());
            return "delete";
        }

    } else {
        model.addAttribute("errorMessage", "NPM CAN NOT BE EMPTY, DELETE CANCELLED");
        return "errorPage";
    }
}
```

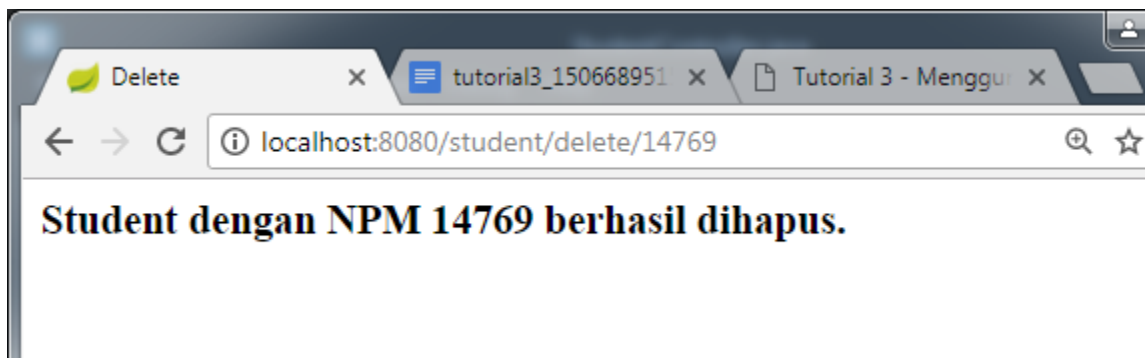
Menambahkan view delete.html untuk delete yang berhasil

A screenshot of a code editor with several tabs open: 'InMemoryStu...', 'StudentCont...', 'view.html', 'errorPage.html', and 'delete.html'. The 'delete.html' tab is active, showing the following HTML code:

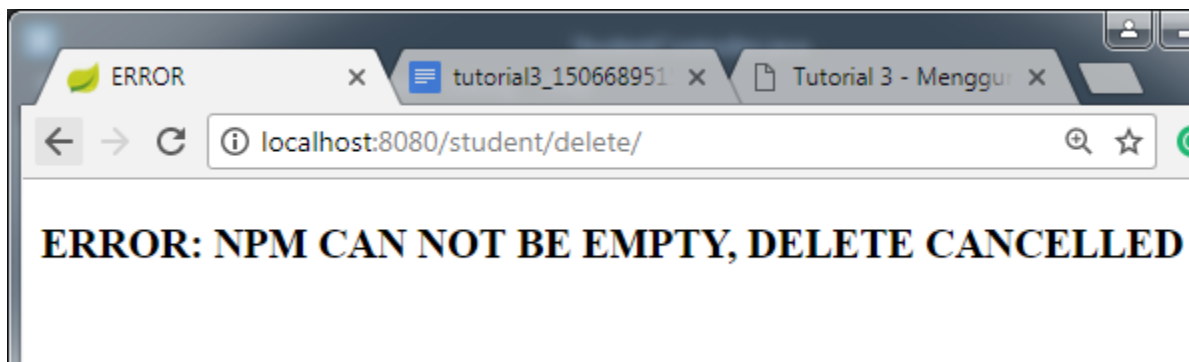
```
1 <html>
2 <head>
3 <title>Delete</title>
4 </head>
5 <body>
6   <h3 th:text="'Student dengan NPM ' + ${npm} + ' berhasil dihapus.'">Delete Student</h3>
7 </body>
8 </html>
```

Untuk npm tidak ditemukan atau tidak ada isinya, controller akan return errorPage.

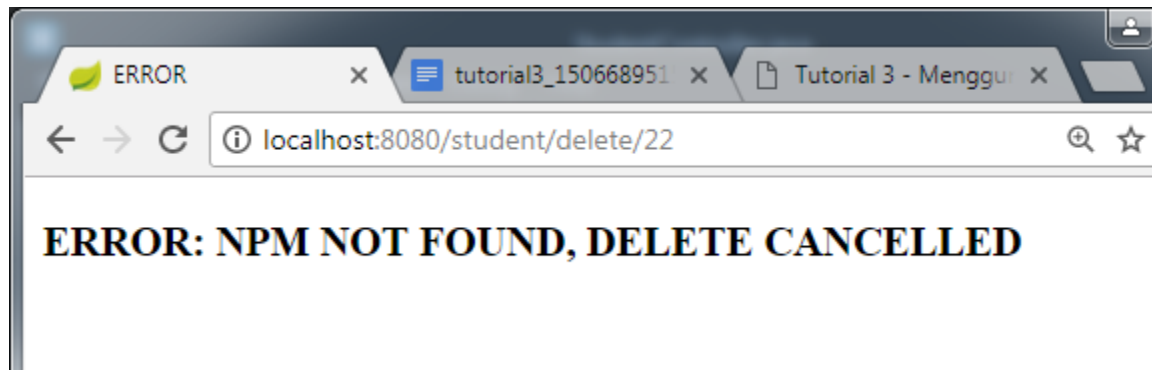
localhost:8080/student/delete/14769.



Jika nomor NPM tidak diberikan



Jika nomor NPM tidak ditemukan



d. Penjelasan fitur delete yang Anda buat pada bagian latihan

Hal pertama yang dilakukan adalah menambahkan method deleteStudent di interface studentservice. Lalu mendefinisikan method deleteStudent di service inmemorystudentservice. Method deleteStudent memiliki parameter npm, di dalamnya akan memanggil selectStudent dengan npm tersebut, lalu menghapus student yang telah di select dari list. Setelah itu, menambahkan method delete di controller StudentController, dan menambahkan view delete.html untuk delete yang berhasil, dan errorpage.html untuk npm tidak ditemukan atau tidak ada isinya.