

Disa Ainun Y
1506689654
KELAS APAP B

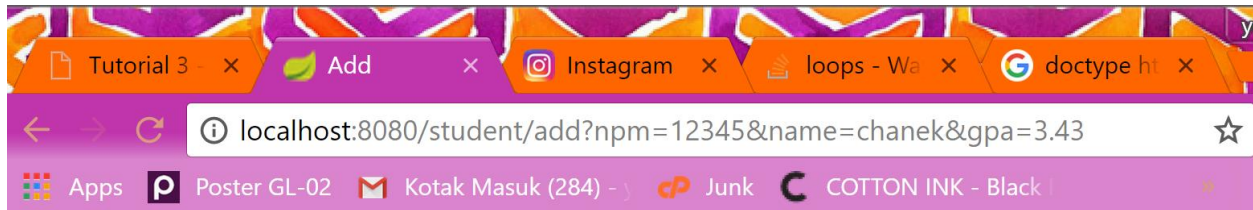
Tutorial

Kode yang diimplementasikan untuk method selectStudent

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

Membuat Controller dan Fungsi Add

1. Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43
Pertanyaan 1: apakah hasilnya?



Data berhasil ditambahkan

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

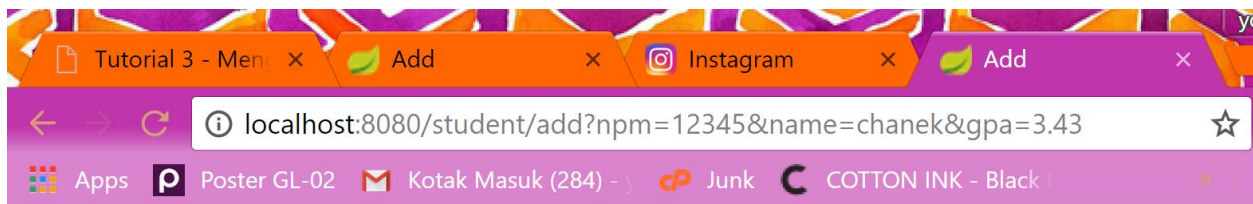
Pertanyaan 2: apakah hasilnya?



Terjadi whitelabel error karena parameter gpa tidak didefinisikan, padahal gpa merupakan field yang required.

Method View by NPM

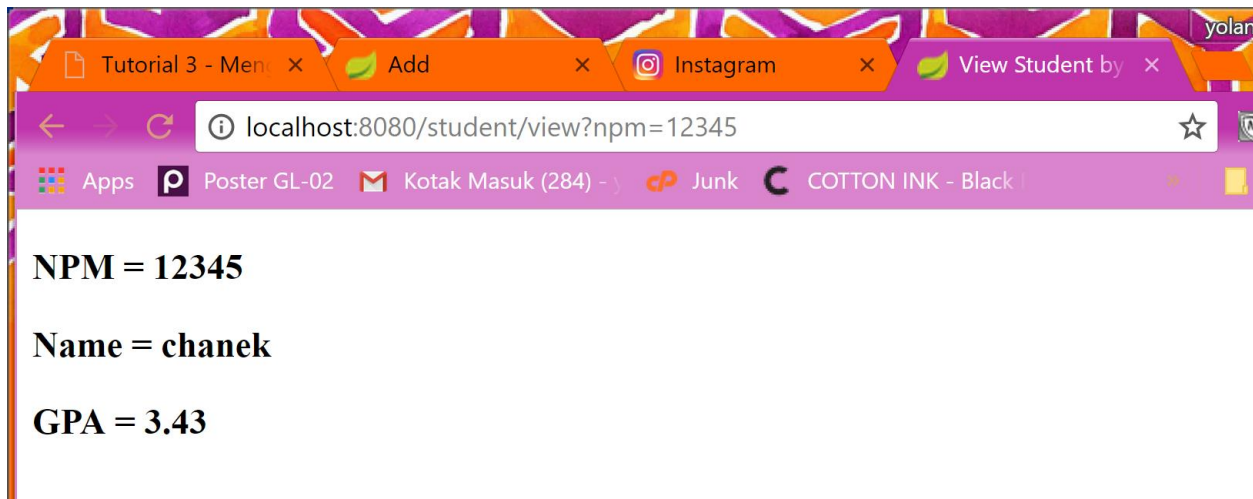
1. Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43



Data berhasil ditambahkan

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

Disa Ainun Y
1506689654
KELAS APAP B

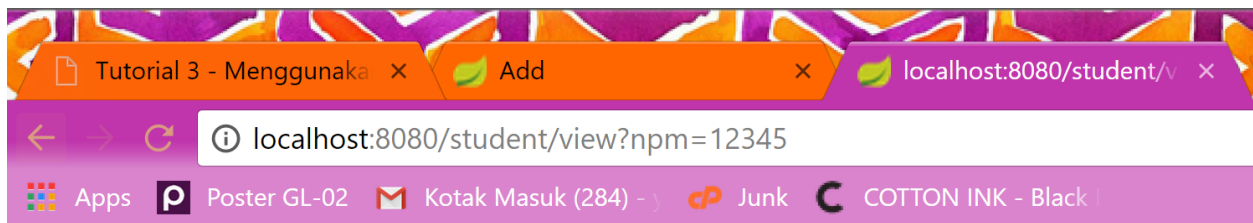


Pertanyaan 3: apakah data Student tersebut muncul? Jika tidak, mengapa?

Muncul.

2. 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.

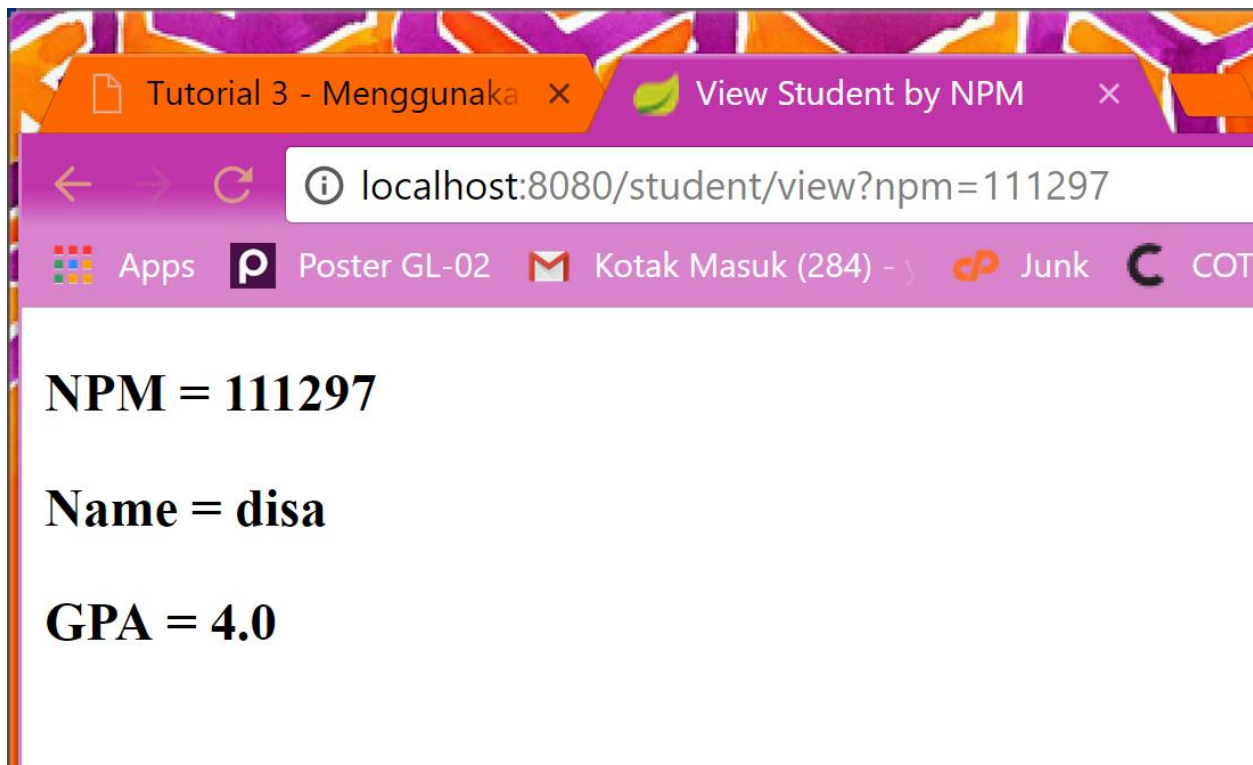
Fri Sep 22 17:37:06 ICT 2017

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

Index: 0, Size: 0

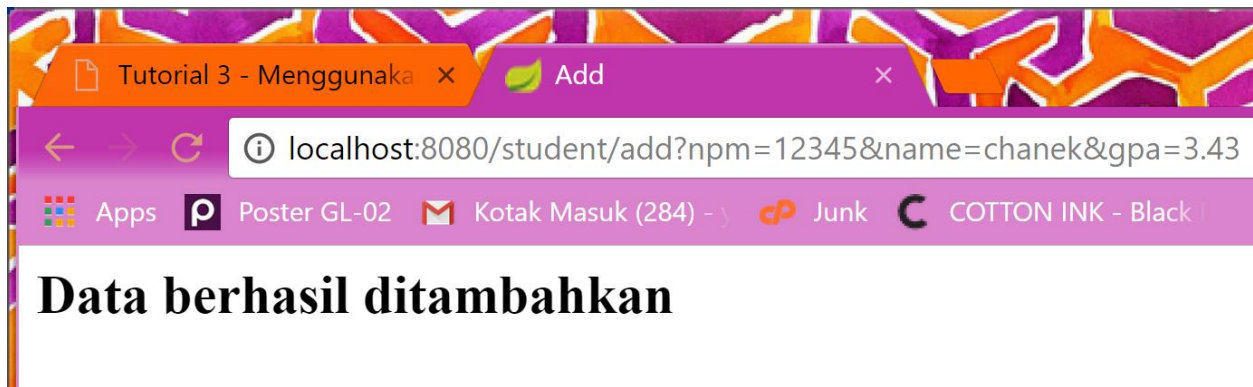
3. Coba tambahkan data Student lainnya dengan NPM yang berbeda

Disa Ainun Y
1506689654
KELAS APAP B

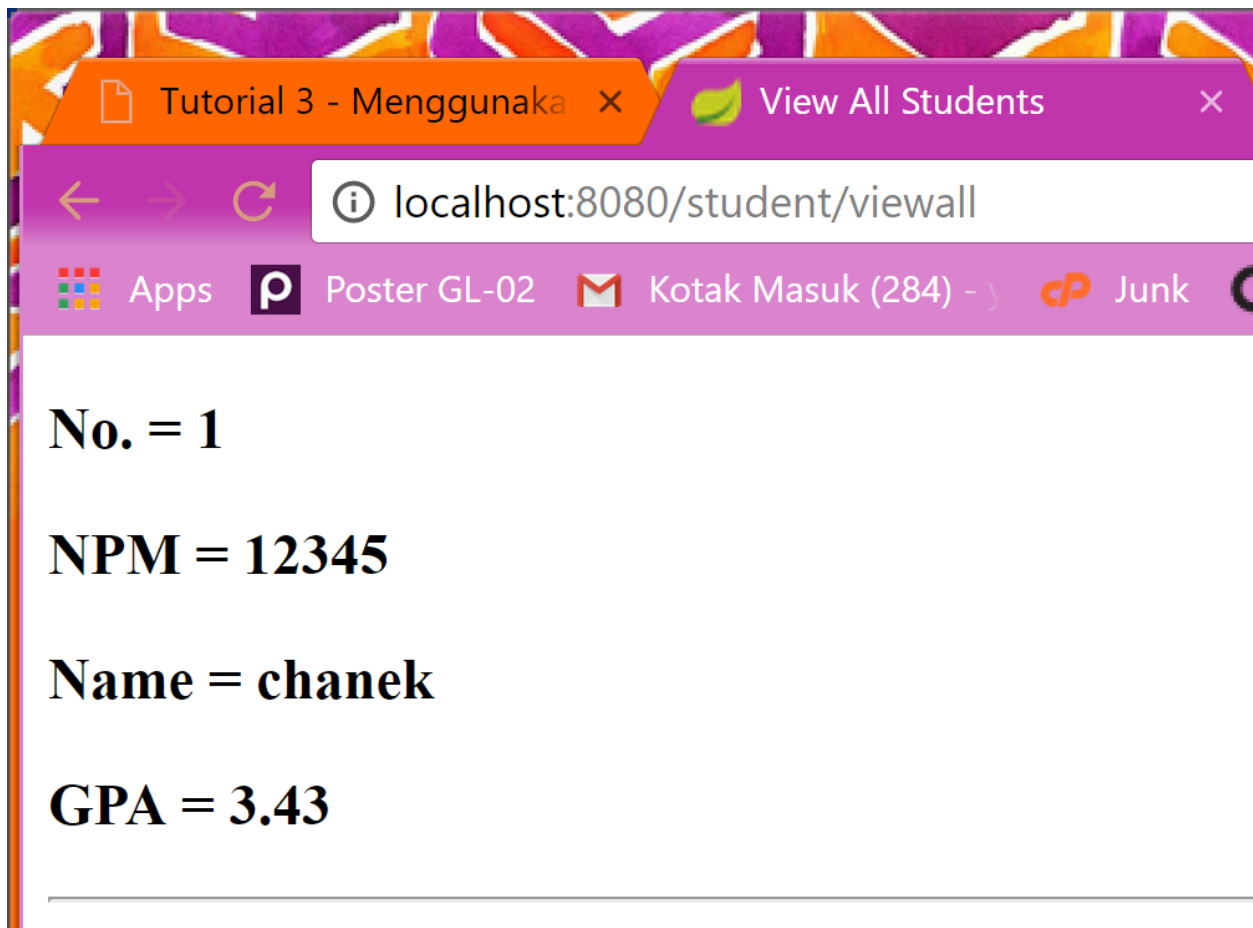


Method View All

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



lalu buka localhost:8080/student/viewall,

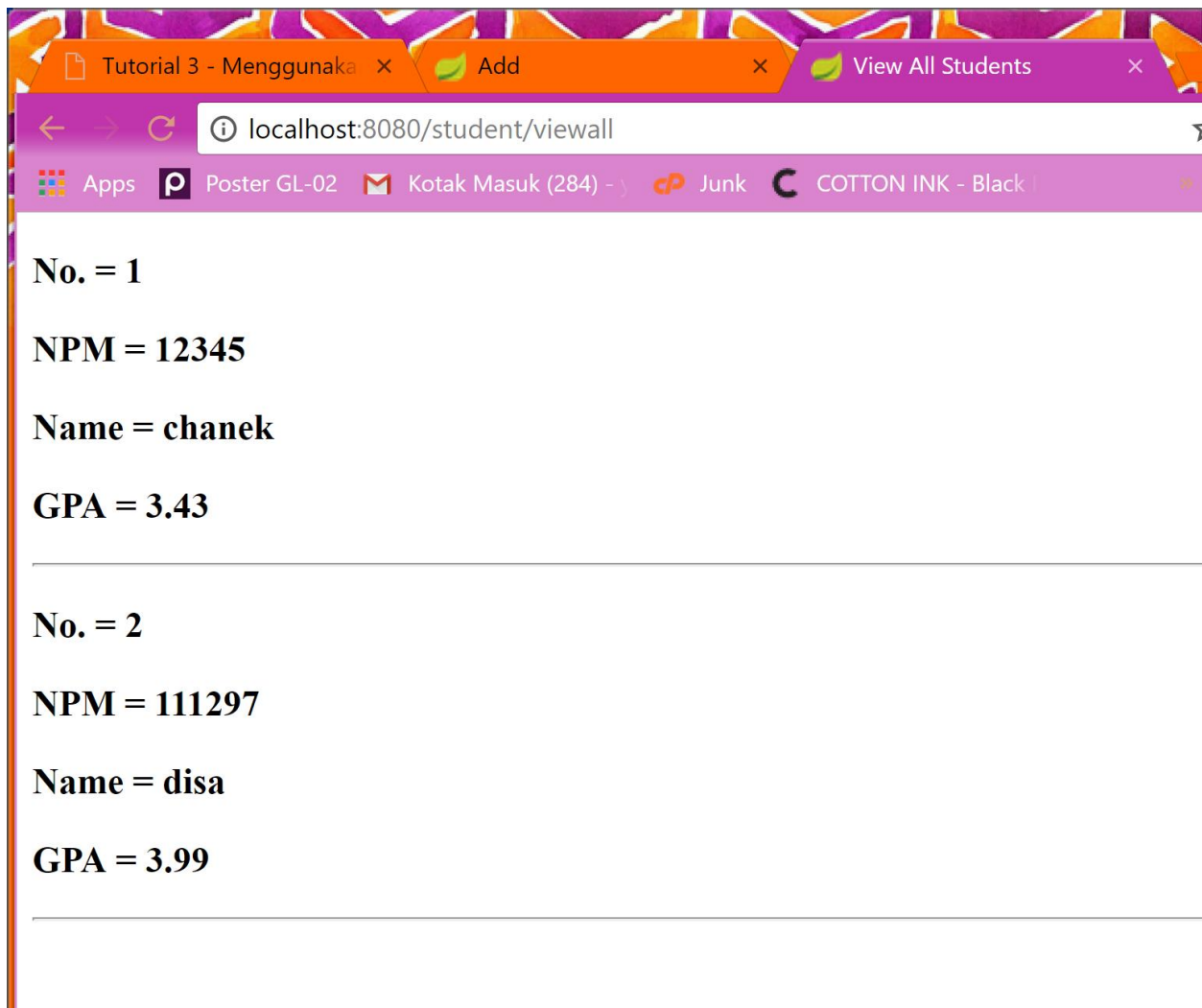
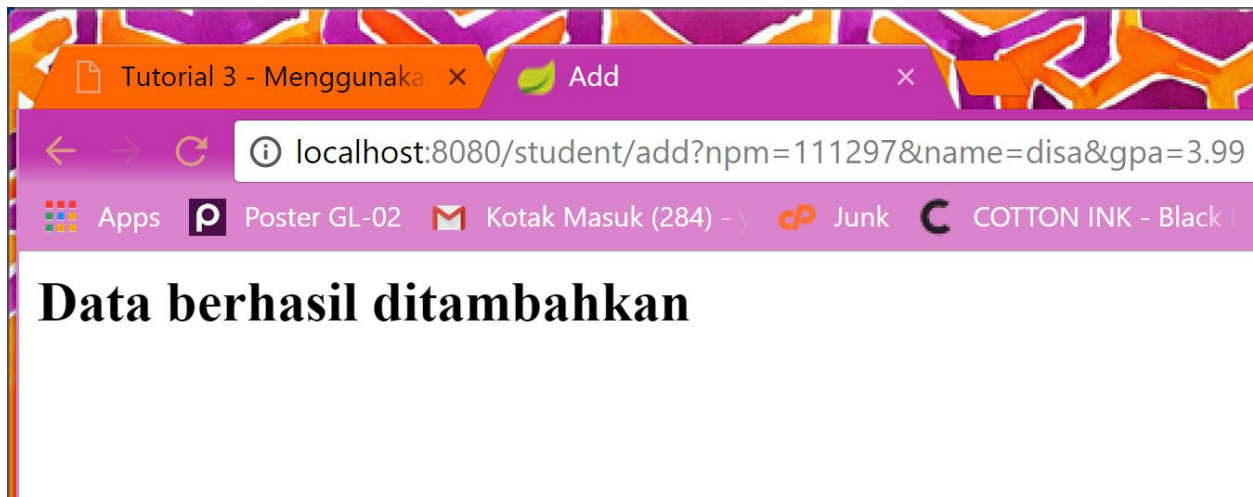


Pertanyaan 5: apakah data Student tersebut muncul?

Muncul

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

Disa Ainun Y
1506689654
KELAS APAP B



Pertanyaan 6: Apakah semua data Student muncul?

Muncul

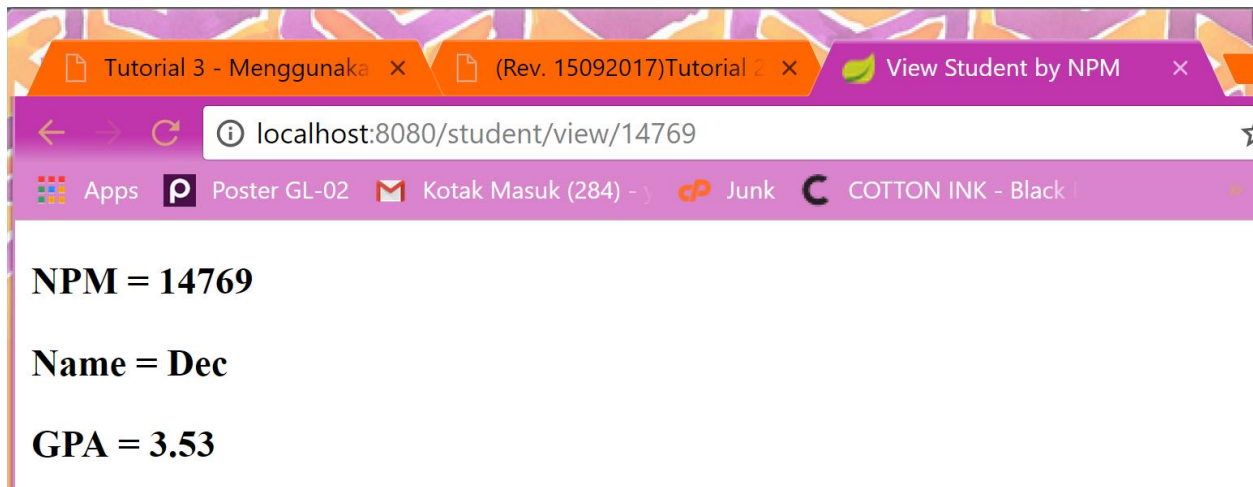
Latihan

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.

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



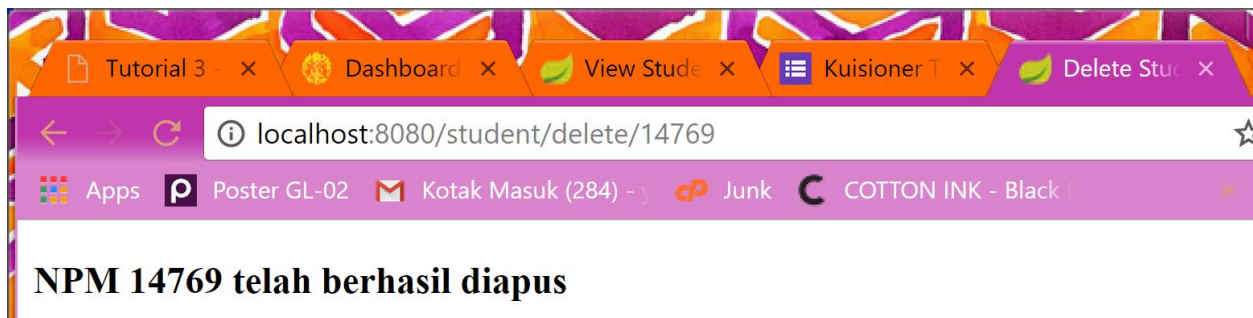
Disa Ainun Y
1506689654
KELAS APAP B



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



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.



Disa Ainun Y
1506689654
KELAS APAP B

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



Ringkasan Materi

Pada tutorial kali ini saya mempelajari mengenai pemanggilan sebuah project java melalui localhost. saya juga mempelajari pembuatan model dengan konsep MVC dalam project Spring Boot. Saya mempelajari keterhubungan antar template seperti kapan kita harus menambahkan method pada controller, dan sebagainya.

Penjelasan Fitur Delete

Pada pembuatan fitur delete, saya menambahkan method deleteStudents pada InMemoryStudent dengan kembalian void yang memerintahkan untuk menghapus object student dari List of Student.

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

Lalu saya menambahkan deleteStudents juga pada interface StudentService.

```
void deleteStudents(StudentModel student);
```

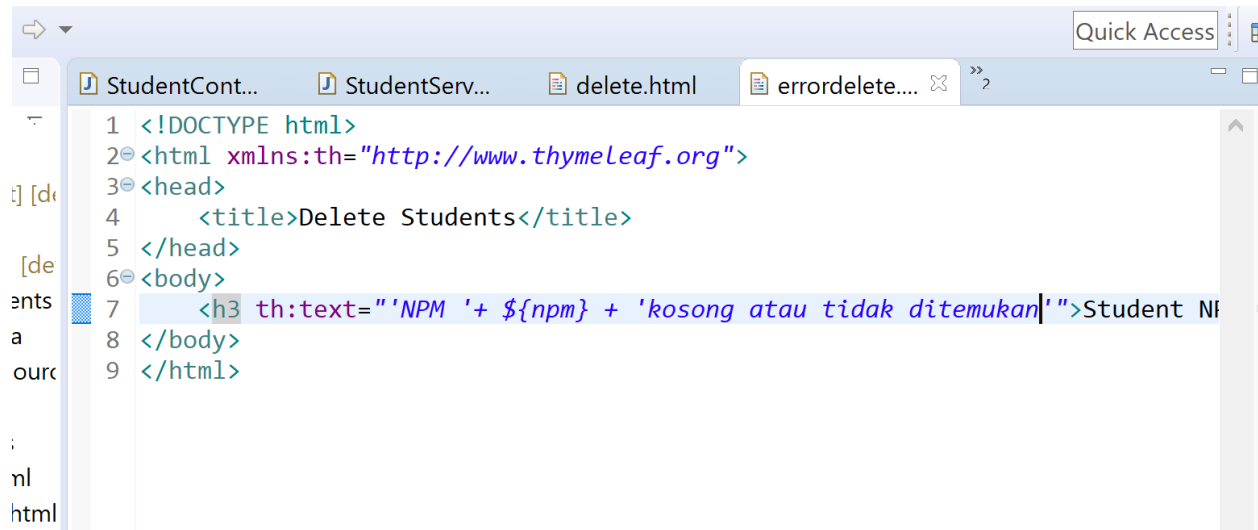
Untuk dapat menampilkan pesan notifikasi terhapusnya object students, saya menambahkan file delete.html dan errordelate.html pada resource/templates. delete.html berisikan pesan bahwa suatu object student berhasil dihapus.



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'resources' and 'templates'. The code editor is open to a file named 'delete.html'. The code in the editor is as follows:

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

Errordelate.html menampilkan pesan mengenai tidak berhasilnya object student dihapus karena npm tersebut tidak ditemukan atau kosong.



The screenshot shows the same IDE with the file explorer and code editor. The file explorer now shows a file named 'errordelate.html' selected. The code editor is open to this file. The code in the editor is as follows:

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4   <title>Delete Students</title>
5 </head>
6 <body>
7   <h3 th:text="'NPM ' + ${npm} + 'kosong atau tidak ditemukan'">Student NI
8 </body>
9 </html>
```

Disa Ainun Y
1506689654
KELAS APAP B

Setelah menambahkan file untuk menampilkan pesan, saya menambahkan method delete pada StudentController.

```
@RequestMapping(value = {"/student/delete", "/student/delete/{npm}"})
public String delete(Model model, @PathVariable Optional<String> npm) {
    String npmTmp = "";
    //Cek apakah terdapat npm dalam path jika ada masuk ke if
    if(npm.isPresent()) {
        npmTmp = npm.get();
        StudentModel student = studentService.selectStudent(npmTmp);
        //cek apakah terdapat student dengan npm yang dicantumkan di path
        if(student != null) {
            studentService.deleteStudents(student);
            model.addAttribute("npm", npmTmp + " ");
            return "delete";
        }
        model.addAttribute("npm", npmTmp + " ");
        return "errordelete";
    }

    npmTmp= npmTmp + " ";
    return "errordelete";
}
```