

## RINGKASAN MATERI

Pada latihan tutorial3 ini mempelajari tentang Model, model digunakan untuk merepresentasikan dan menyimpan informasi data student dalam tutorial ini adalah objek student yang menyimpan data Nama, NPM, dan GPA. Selain itu dipelajari pula service yang merupakan interface digunakan untuk mendefinisikan method apa saja yang dapat digunakan untuk manipulasi data dari student.

### Membuat *Class Model*

- Student Model

```
package com.example.tutorial3.model;

public class StudentModel {
    private String name;
    private String npm;
    private double gpa;

    public StudentModel (String npm, String name, double gpa) {
        this.npm = npm;
        this.name = name;
        this.gpa = gpa;
    }
}
```

- Menambahkan method constructor, setter dan getter untuk setiap variabel

```
package com.example.tutorial3.model;

public class StudentModel {
    private String name;
    private String npm;
    private double gpa;

    public StudentModel (String npm, String name, double gpa) {
        this.npm = npm;
        this.name = name;
        this.gpa = gpa;
    }

    public String getNpm() {
        return npm;
    }

    public void setNpm(String npm) {
        this.npm = npm;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public double getGpa() {
        return gpa;
    }

    public void setGpa(double gpa) {
        this.gpa = gpa;
    }
}
```

## Membuat Service

- StudentService.java (Interface)

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

import java.util.List;
import java.util.Optional;

import com.example.tutorial3.model.StudentModel;

public interface StudentService {
    StudentModel selectStudent (String npm) ;

    List<StudentModel> selectAllStudents();

    void addStudent (StudentModel student);

    StudentModel selectStudent(Optional<String> npm);
}
```

- InMemoryStudentService dengan implementasi static list (studentList) untuk menyimpan data mahasiswa.

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

import java.util.ArrayList;

public class InMemoryStudentService implements StudentService{
    private static List<StudentModel> studentList = new ArrayList<StudentModel>();

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

    @Override
    public List<StudentModel> selectAllStudents(){
        return studentList;
    }

    @Override
    public void addStudent (StudentModel student) {
        studentList.add(student);
    }
}
```

## Membuat Controller dan Fungsi Add

- StudentController (method add)

Menerima parameter name, npm, dan gpa dengan menggunakan request method GET.

```
package com.example.tutorial3.controller;

import java.util.List;

@Controller
public class StudentController {
    private final StudentService studentService;

    public StudentController() {
        studentService = new InMemoryStudentService ();
    }

    @RequestMapping("/student/add")
    public String add(@RequestParam(value="npm",required= true)String npm,
        @RequestParam(value = "name",required = true)String name,
        @RequestParam(value = "gpa",required = true) double gpa){
        StudentModel student = new StudentModel(npm,name,gpa);
        studentService.addStudent(student);
        return "add";
    }
}
```

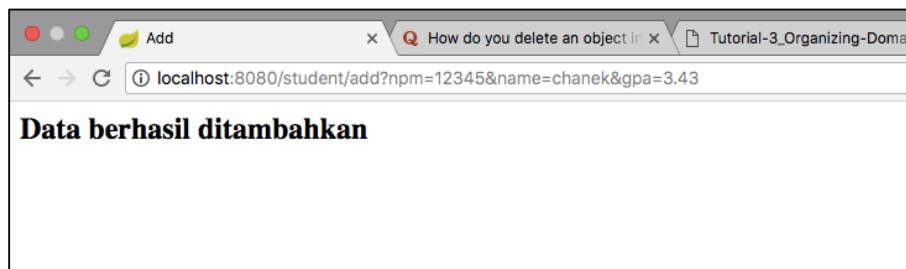
- Pada directory resources/ templates tambahkan add.html

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

- Jalankan program dan buka

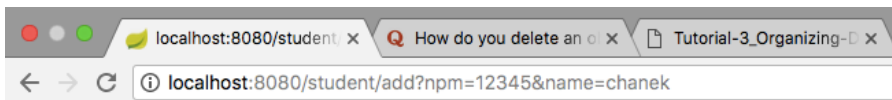
- a. **localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43**

Hasilnya seperti pada gambar berikut.



- b. **localhost:8080/student/add?npm=12345&name=chanek**

Akan terjadi error dikarenakan parameter gpa tidak diberikan nilai.



## Whitelabel Error Page

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

Fri Mar 02 23:01:43 WIB 2018

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

Required double parameter 'gpa' is not present

## Method View by NPM

- Student Controller (method view)

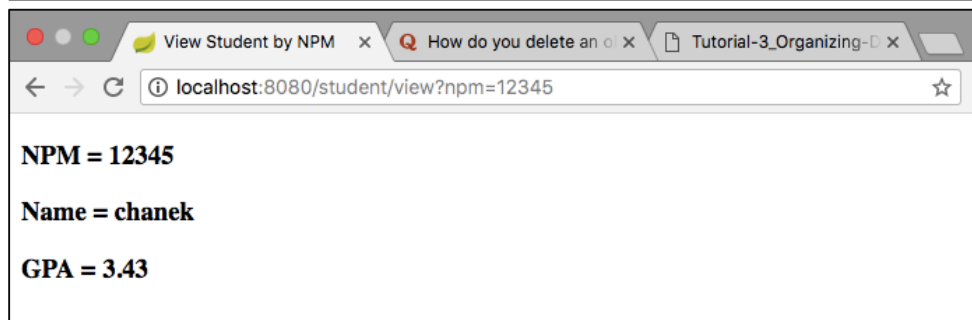
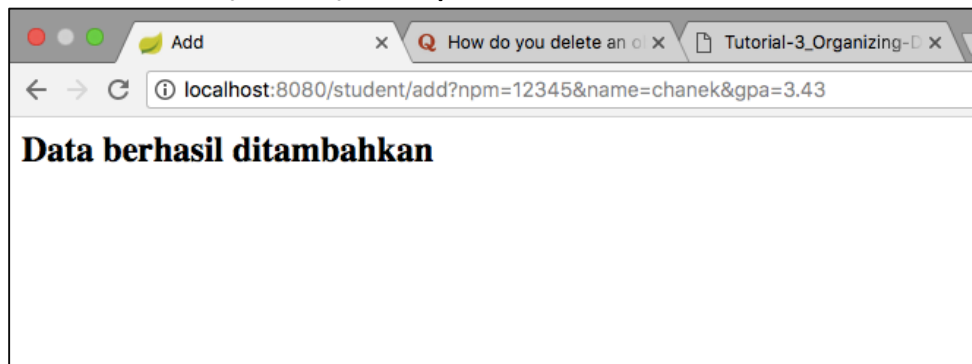
```
@RequestMapping("/student/view")
public String view (Model model, @RequestParam(value="npm", required = true)String npm) {
    StudentModel student = studentService.selectStudent(npm);
    model.addAttribute("student", student);
    return "view";
}
```

- Pada directory/ templates tambahkan view.html

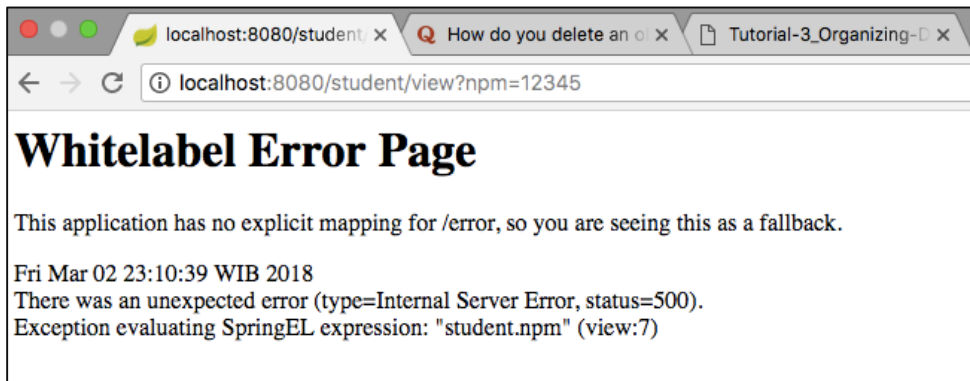
```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Student by NPM</title>
5 </head>
6 <body>
7     <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
8     <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
9     <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
10 </body>
11 </html>
```

Jalankan program dan buka

- a. **localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43** dan buka **localhost:8080/student/view?npm=12345**.



- b. **Setelah dimatikan dan dijalankan kembali localhost:8080/student/view?npm=12345**  
Akan muncul tampilan error seperti pada gambar berikut dikarenakan data mahasiswa belum dimasukkan, dan harus menjalankan method add terlebih dahulu.



### Method View All

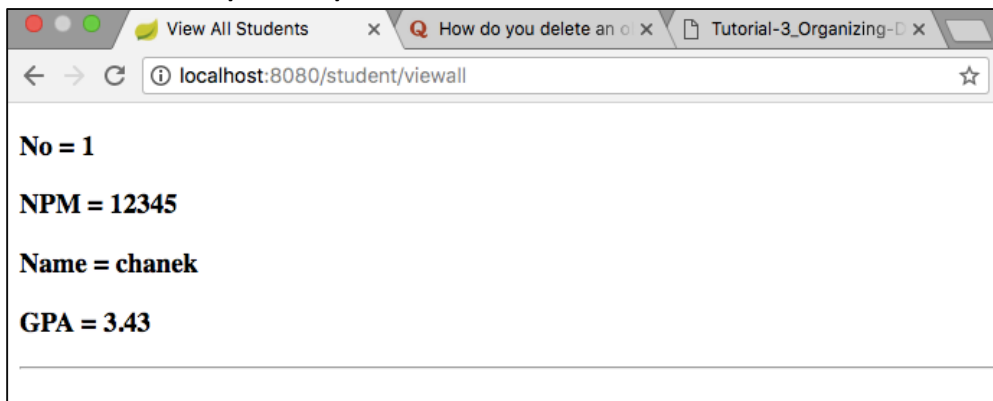
- StudentController (method viewall)

```
@RequestMapping("/student/viewall")
public String viewAll (Model model) {
    List<StudentModel> students = studentService.selectAllStudents();
    model.addAttribute("students", students);
    return "viewall";
}
```

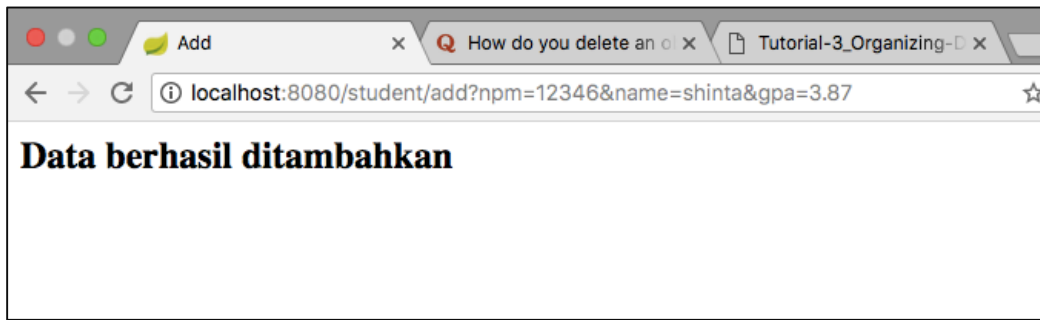
- Pada directory resources tambahkan viewall.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Students</title>
5 </head>
6 <body>
7 <div th:each="student, iterationStatus: ${students}">
8     <h3 th:text="No = ' + ${iterationStatus.count}">No. 1</h3>
9     <h3 th:text="NPM = ' + ${student.npm}">Student NPM</h3>
10    <h3 th:text="Name = ' + ${student.name}">Student Name</h3>
11    <h3 th:text="GPA = ' + ${student.gpa}">Student GPA</h3>
12 </div>
13 </div>
14 </body>
15 </html>
```

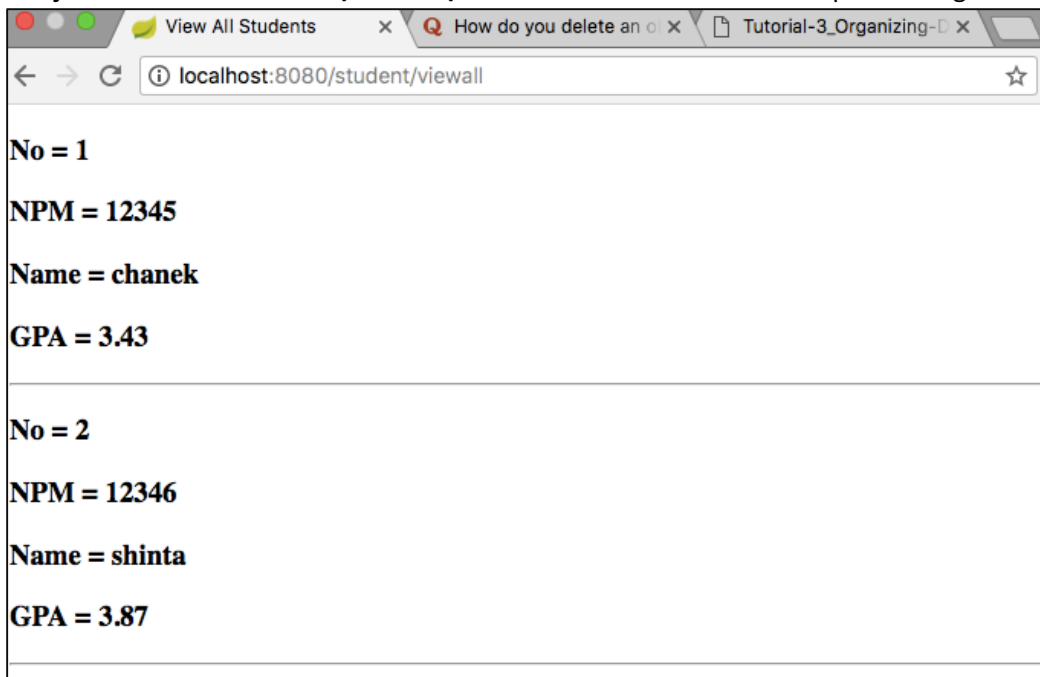
- a. jalankan localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/viewall



- b. Tambahkan Student lain dengan NPM yang berbeda dan buka kembali



lalu jalankan `localhost:8080/student/viewall` maka akan muncul tampilan sebagai berikut.



## LATIHAN

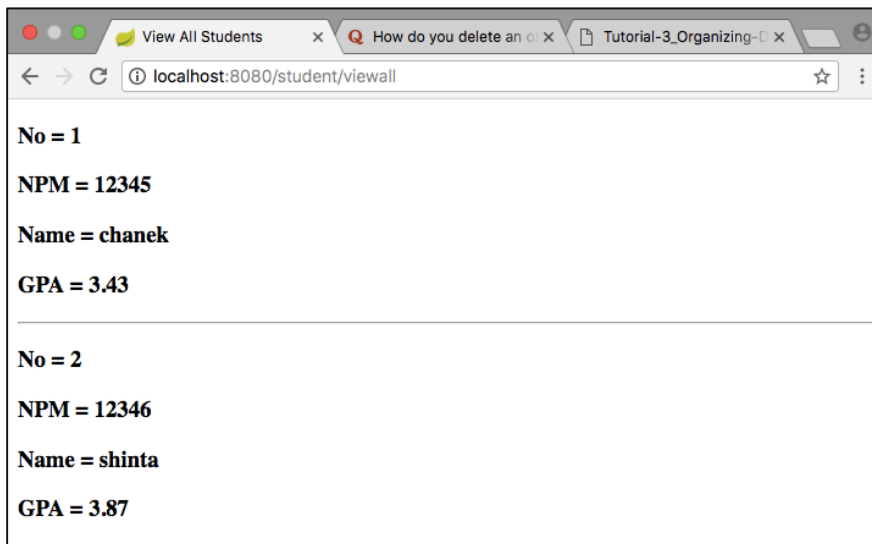
1. Pada StudentController tambahkan method view Student dengan menggunakan Path Variable, dan masukkan data mahasiswa.
- Student Controller method view dengan menggunakan Path Variable.

```
@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) {
            model.addAttribute("student", student);
        }
        else {
            model.addAttribute("npm", npm.get());
            return "notfound";
        }
    } else {
        return "notfound";
    }
    return "view";
}
```

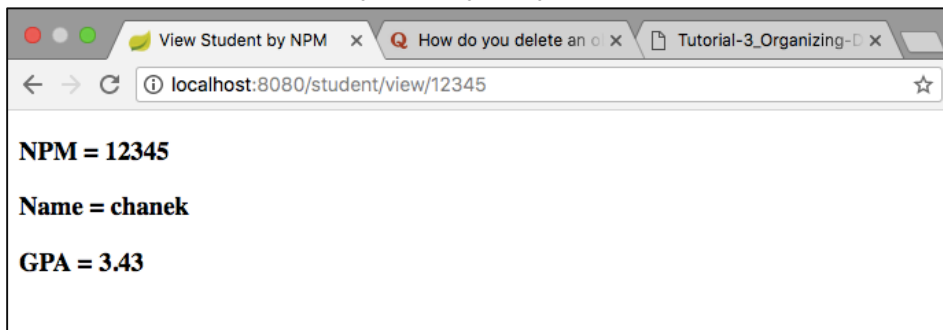
Buat file baru pada directory/ resources dengan nama notfound.html. File ini digunakan untuk menampilkan jika NPM yang dimasukkan dan jika tidak memasukkan NPM akan muncul tampilan "Data tidak Ditemukan".

```
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>Data Tidak Ditemukan</h3>
8 </body>
9 </html>
10
```

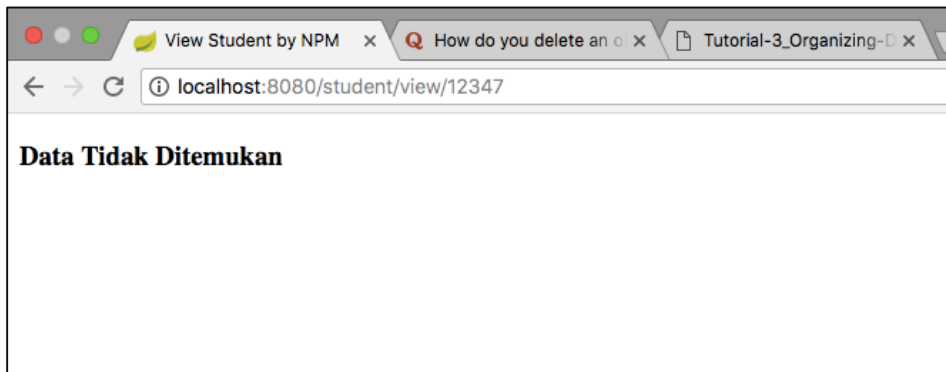
a. Jalankan localhost:8080/student/viewall maka akan muncul tampilan sebagai berikut.



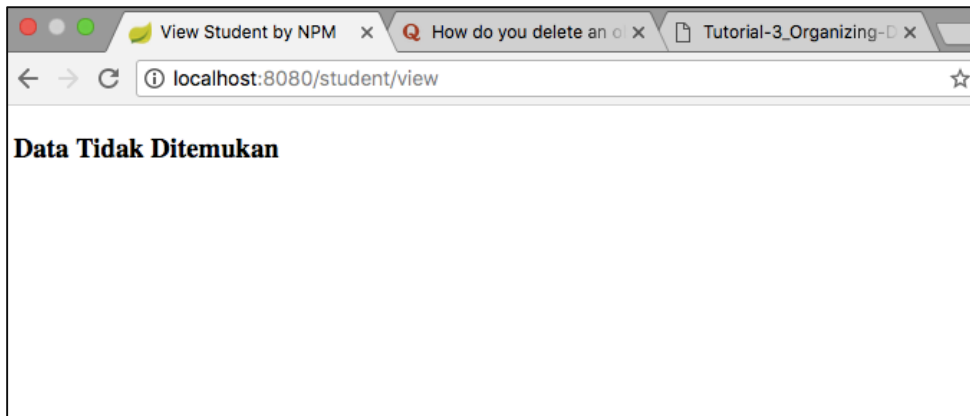
b. Masukkan localhost:8080/student/view/12345



c. Jika NPM tidak ada pada list student maka akan muncul tampilan sebagai berikut



d. Jika tidak menginputkan npm maka akan muncul tampilan sebagai berikut



2. Method selectStudent yang di implementasikan dengan parameter yang memiliki tipe data yang berbeda. Dalam method tersebut akan dilakukan pengecekan apakah npm yang diinputkan ada didalam list student tersebut. Jika npm yang diinputkan tidak ada dalam list maka akan mengembalikan nilai null.

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

```
}
@Override
public StudentModel selectStudent(Optional<String> npm) {
    // TODO Auto-generated method stub
    for (int i=0;i<studentList.size();i++) {
        if (studentList.get(i).getNpm().equals(npm)) {
            return studentList.get(i);
        }
    }
    return null;
}
```

3. Tambahkan Fitur Delete Student berdasarkan NPM.
  - Dengan menambahkan method delete pada StudentCotroller (method delete)



```

    @RequestMapping(value = {"/student/delete","student/delete/{npm}"})
    public String deleteStudent (@PathVariable Optional<String> npm, Model model) {
        if (npm.isPresent()) {
            StudentModel student = studentService.selectStudent(npm.get());
            if (student != null) {
                model.addAttribute("student", student);
                studentService.deleteStudent(student);
            }
            else {
                model.addAttribute("npm", npm.get());
                return "notfound";
            }
        } else {
            return "notfound";
        }
        return "del";
    }

```

- Tambahkan file pada directories/ templates dengan nama del.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 th:text="'Student with NPM ' + ${student.npm} + ' deleted'">Student NPM</h3>
8
9 </body>
10 </html>
11

```

- Tambahkan method pada InMemoryStudent

```

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

```

- StudentService

```

package com.example.tutorial3.service;

import java.util.List;

public interface StudentService {
    StudentModel selectStudent (String npm) ;

    List<StudentModel> selectAllStudents();

    void addStudent (StudentModel student);

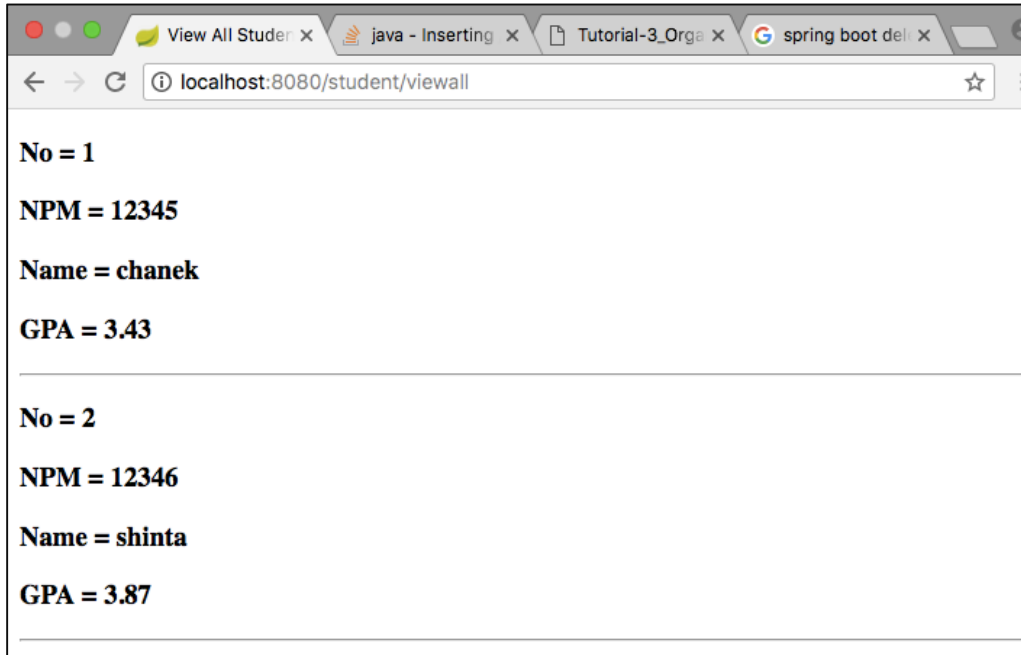
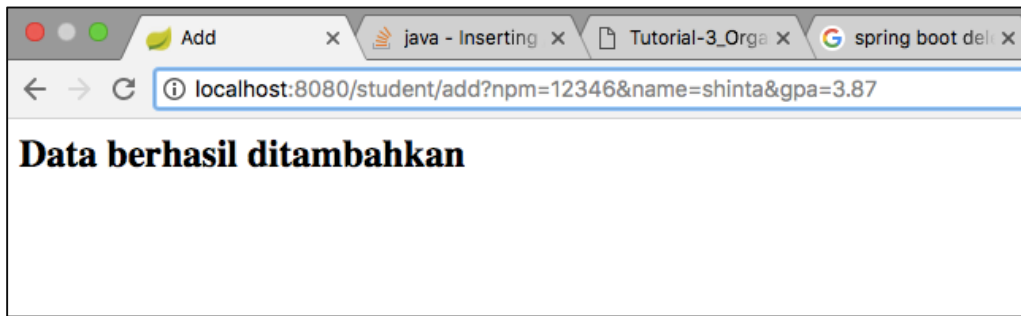
    StudentModel selectStudent(Optional<String> npm);

    void deleteStudent(StudentModel student);
}

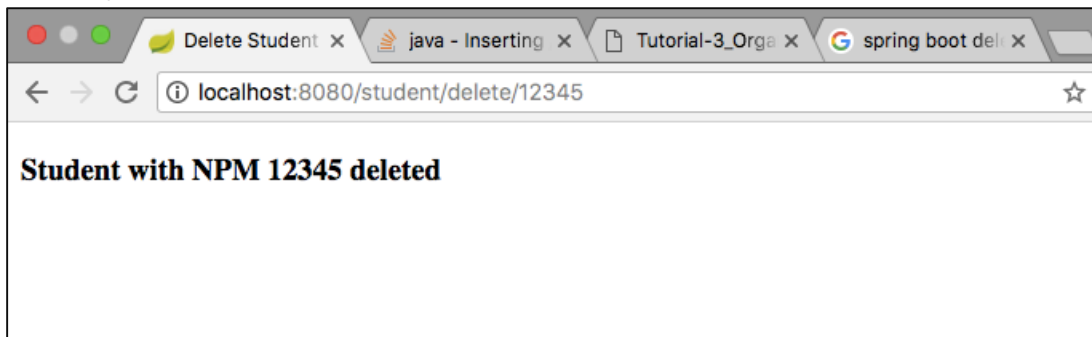
```

- Jalankan localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 dan localhost:8080/student/add?npm=12346&name=shinta&gpa=3.87 lalu buka localhost:8080/student/viewall

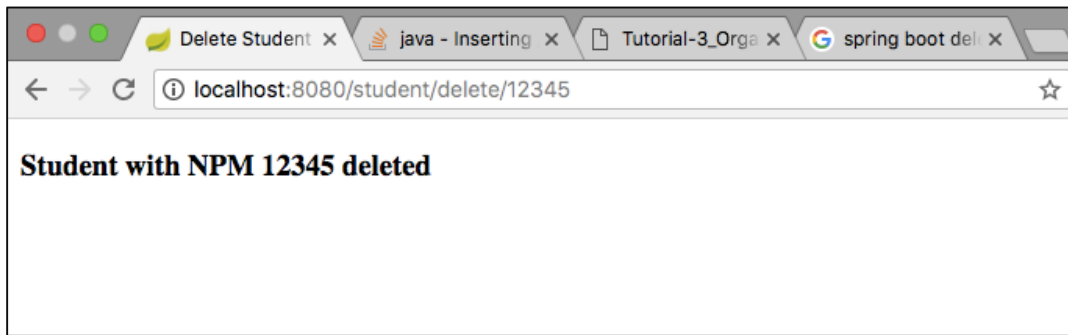
Shinta Kusuma Wardhani  
1606955044  
Tutorial 3



b. Lalu jalankan `localhost:8080/student/delete/12345`



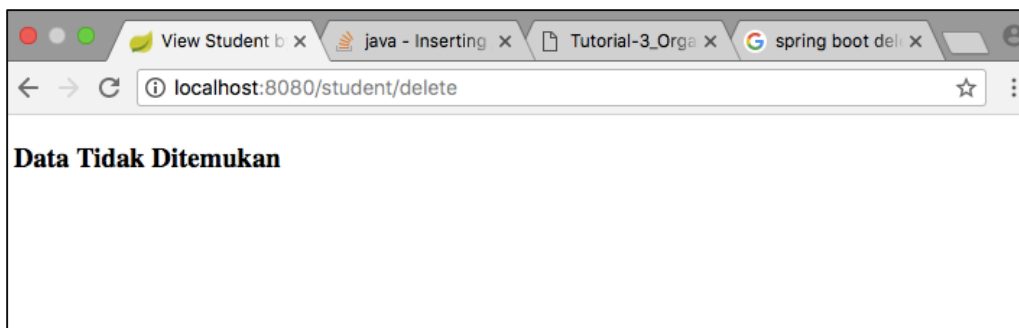
c. Lalu jalankan `localhost:8080/student/delete/12345` maka akan muncul tampilan sebagai berikut



d. Lalu jalankan `localhost:8080/student/viewall`



e. Jika dijalankan `localhost:8080/student/delete` maka akan muncul tampilan sebagai berikut



f. Jika NPM yang dimasukkan tidak terdapat dalam list student maka akan muncul tampilan sebagai berikut. Pertama jalankan `localhost:8080/student/delete/12354`

