

Nama : Riandani Gita Larasati

NPM : 1606954981

SI Ekstensi

a-b. Ringkasan dan hasil jawaban.

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

import java.util.List;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.example.tutorial3.model.StudentModel;
import com.example.tutorial3.service.InMemoryStudentService;
import com.example.tutorial3.service.StudentService;

@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";
    }
}
```

Setelah mengikuti tahap demi tahap pada tutorial 3 ini, script di atas merupakan controller yang akan terus di-update. Di dalam controller tersebut berisi method-method sesuai dengan kebutuhan program nantinya. Method tersebut akan memanggil view atau file html yang akan ditampilkan pada web browser. Berikut ini merupakan penjelasan beserta *screenshot* code atau hasil setelah dijalankan pada browser.

Membuat Controller dan Fungsi Add

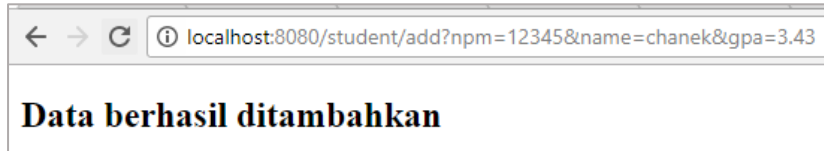
Jalankan program dan buka **localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43**

Pertanyaan 1: apakah hasilnya? Jika error, tuliskan penjelasan Anda.

Tidak error. Karena pada url kita memasukkan npm, nama, dan gpa yang bersifat required pada controller. Seperti yang dapat dilihat pada screenshot dibawah ini, npm, name, dan gpa memiliki required = true

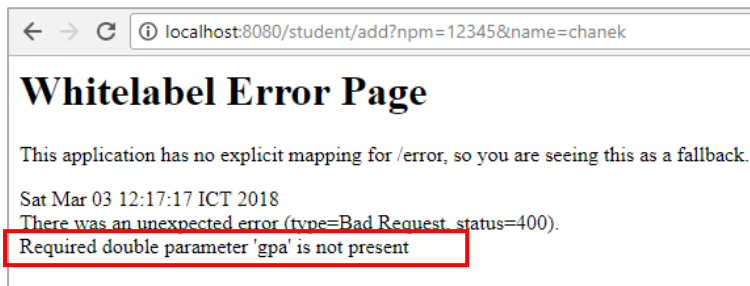
```
public String add(@RequestParam(value = "npm", required = true) String npm,  
                 @RequestParam(value = "name", required = true) String name,  
                 @RequestParam(value = "gpa", required = true) double gpa) {
```

Hasil saat dijalankan pada browser:



Dapat dilihat berdasarkan gambar di atas bahwa data berhasil ditambahkan, yaitu dengan menambahkan npm, nama, dan gpa pada url.

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

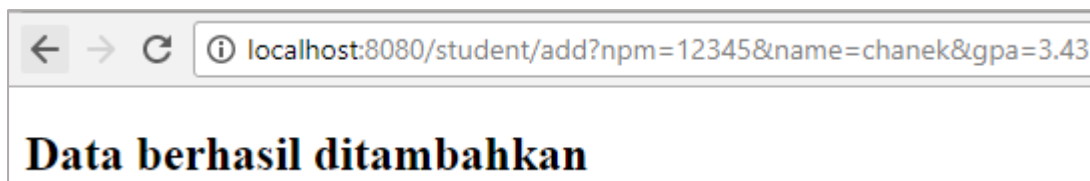


Pertanyaan 2: apakah hasilnya? Jika error, tuliskan penjelasan Anda.

Error, karena tidak ada parameter 'gpa' pada url, sedangkan pada script parameter 'gpa' merupakan required = true yang artinya wajib diinputkan.

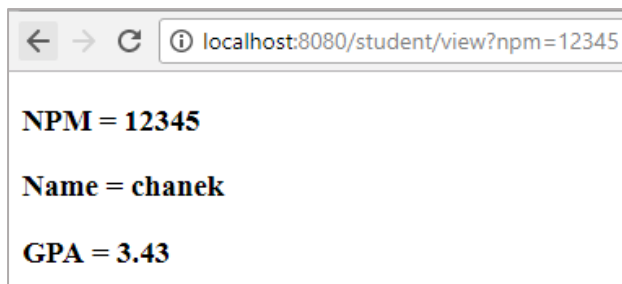
Method View by NPM

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



Dapat dilihat berdasarkan gambar di atas bahwa data berhasil ditambahkan, yaitu dengan menambahkan npm, nama, dan gpa pada url.

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

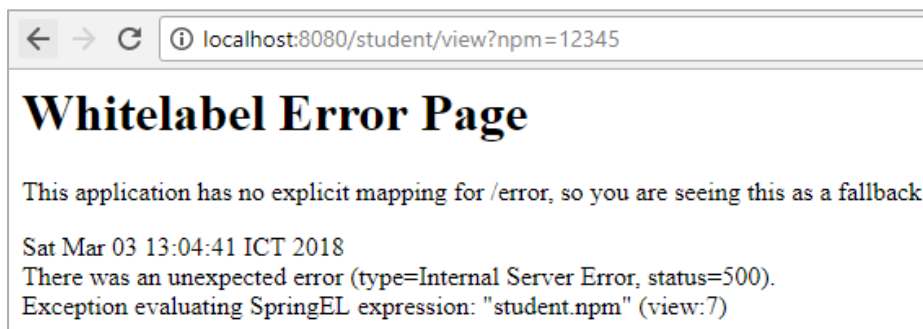


Berdasarkan gambar diatas, dapat dilihat bahwa dengan memasukkan url lengkap dengan npm yang sudah dimasukkan sebelumnya, maka view dapat dipanggil dan menampilkan npm, nama, dan gpa.

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

Ya, karena pada url dipanggil view yang menampilkan npm, nama, gpa berdasarkan npm yang terdapat pada url yaitu 12345.

Coba matikan program dan jalankan kembali serta buka **localhost:8080/student/view?npm=12345**

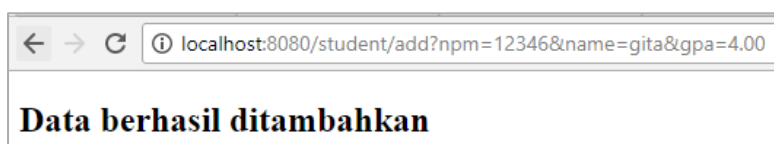


Pertanyaan 4: apakah data Student tersebut muncul? Jika tidak, mengapa?

Tidak, karena sebelumnya program dimatikan sehingga akan reset ulang. Maka jika ingin memunculkan data harus di add kembali.

Coba tambahkan data Student lainnya dengan NPM yang berbeda

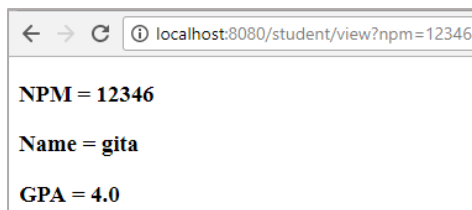
localhost:8080/student/add?npm=12346&name=gita&gpa=4.00



Pada tahapan ini saya menambahkan data dengan menginputkan pada url

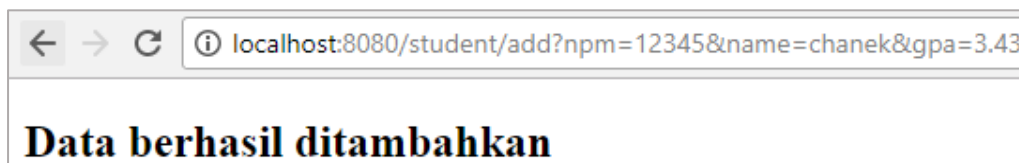
Npm : 12346
Name : gita
Gpa : 4.0

<http://localhost:8080/student/view?npm=12346>



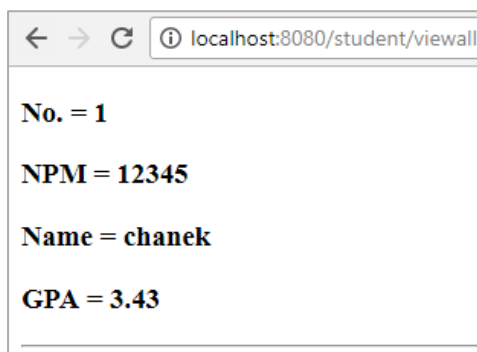
Method View All

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



Dapat dilihat berdasarkan gambar di atas bahwa data berhasil ditambahkan, yaitu dengan menambahkan npm, nama, dan gpa pada url.

lalu buka localhost:8080/student/viewall,

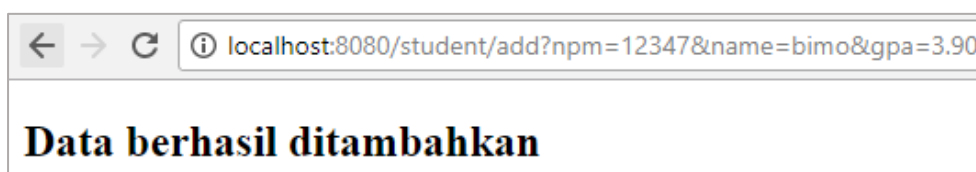


Pertanyaan 5: apakah data Student tersebut muncul?

Ya, muncul. Dengan memanggil viewall maka akan mengembalikan output berupa no, npm, name, dan gpa.

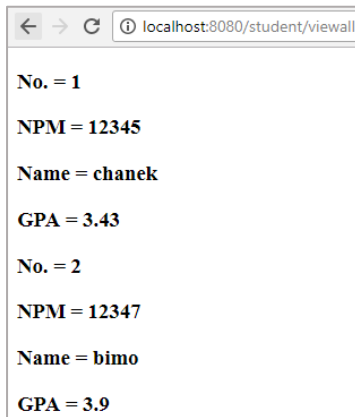
View all sendiri untuk mengembalikan output berupa semua record yang telah diinputkan, akan tetapi pada tahapan ini baru menambahkan satu data.

Coba tambahkan data Student lainnya dengan NPM yang berbeda,



Dapat dilihat berdasarkan gambar di atas bahwa data berhasil ditambahkan, yaitu dengan menambahkan npm, nama, dan gpa pada url.

lalu buka **localhost:8080/student/viewall**,



Pertanyaan 6: Apakah semua data Student muncul?

Ya, muncul. View all mengeluarkan semua record yang telah diinputkan. Yaitu kedua data no 1 dan no 2 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

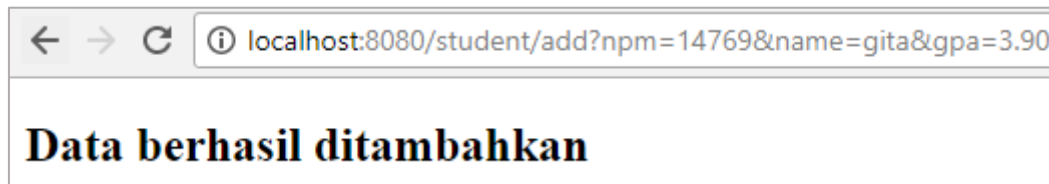
localhost:8080/student/view/14769.

```
@RequestMapping (value = {"/student/view", "/student/view/{npm}"})
public String viewPath(@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("message","NPM " + npm.get() + " tidak ditemukan");
            return "notFound";
        }
    }else {
        model.addAttribute("message","NPM kosong.");
        return "notFound";
    }
    return "view";
}
```

Pada screenshot diatas terdapat method viewPath yang akan menerima input npm yang bertipe data String. Adanya pengecekan apabila npm diinputkan maka akan melakukan pengecekan data berdasarkan npm tersebut, namun apabila npm tidak diinputkan maka akan muncul output "NPM kosong." yang dipanggil dari file html yang dinamakan notFound.html. Ada pula kondisi saat npm diinputkan, yaitu pengecekan apakah npm tersebut ada atau tidak. Apabila npm tersebut ada,

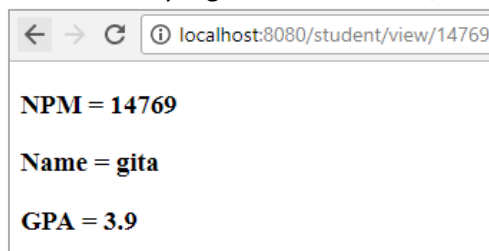
maka program akan memberikan keluaran berupa data student, dan apabila npm tersebut tidak tersedia maka akan muncul output “NPM <NPM yang diinputkan> tidak ditemukan”.

memasukkan data seorang Student yang memiliki NPM 14769:

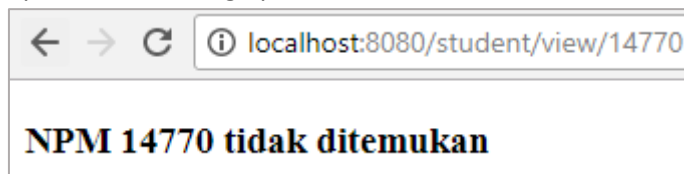


Dapat dilihat berdasarkan gambar di atas bahwa data berhasil ditambahkan, yaitu dengan menambahkan npm, nama, dan gpa pada url.

melihat data yang baru dimasukkan, **localhost:8080/student/view/14769**

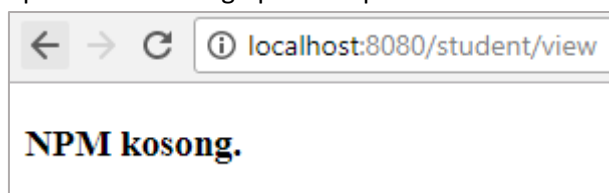


Apabila salah menginputkan:



Pada url, diinputkan npm yang sebenarnya npm itu tidak tersedia atau belum pernah ditambahkan.

Apabila tidak menginputkan npm:



Pada url tidak ditambahkan npm untuk menampilkan data.

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**.

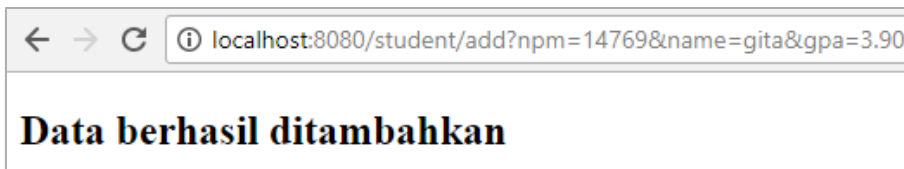
Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus. Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman error yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan dan proses delete dibatalkan.

d. Penjelasan fitur delete yang Anda buat pada bagian latihan.

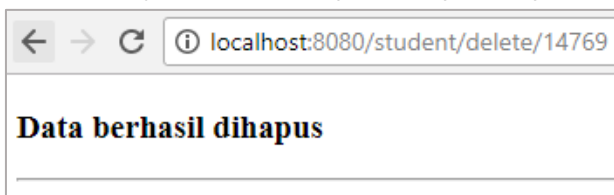
```
@RequestMapping (value = {"/student/delete", "/student/delete/{npm}"})
public String deletePath(@PathVariable Optional <String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student != null) {
            model.addAttribute("student",student);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

Pada screenshot diatas terdapat method deletePath yang akan menerima input npm yang bertipe data String. Adanya pengecekan apabila npm diinputkan maka akan melakukan pengecekan data berdasarkan npm tersebut, namun apabila npm tidak diinputkan maka akan muncul output “NPM kosong.” yang dipanggil dari file html yang dinamakan notFound.html. Ada pula kondisi saat npm diinputkan, yaitu pengecekan apakah npm tersebut ada atau tidak. Apabila npm tersebut ada, maka program akan menghapus data student, dan apabila npm tersebut tidak tersedia maka akan muncul output “NPM <NPM yang diinputkan> tidak ditemukan”.

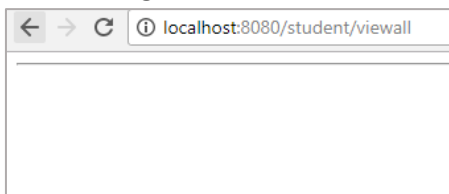
Sebelumnya:



Delete data, localhost:8080/student/delete/14769

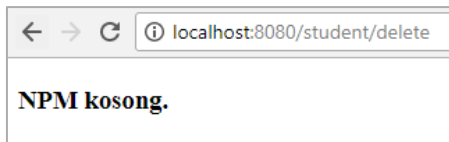


Data kosong

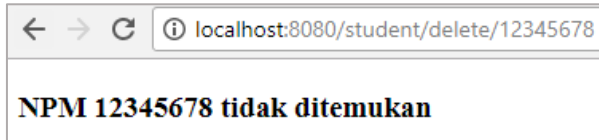


Kosong dikarenakan belum ada data yang ditambahkan sehingga belum ada data yang dapat ditampilkan.

Apabila akan delete data akan tetapi tidak menginputkan NPM



Apabila akan delete akan tetapi npm tidak terdaftar



c. Method selectStudent yang Anda implementasikan

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

    for (int i=0;i<studentList.size();i++) {
        System.out.println(npm + " " + studentList.get(i).getNpm());
        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);
}

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

public StudentModel selectStudent(Optional<String> npm) {
```