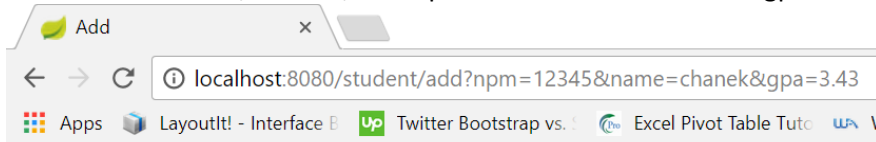


- a. Ringkasan dari materi yang Anda telah pelajari pada tutorial kali ini  
Pada tutorial 3 ini, saya memperelajari tentang penggunaan controller, model, dan service. Model merupakan suatu representasi yang menyimpan informasi data. Service merupakan layer yang menjadi penghubung antara controller dengan model untuk dikembalikan sebagai view.
- b. Hasil jawaban dari setiap poin pada bagian tutorial (dapat didukung dengan screenshot)  
Implementasikan method selectStudent! Method ini menerima NPM mahasiswa dan mengembalikan object Student dengan NPM tersebut. Return null jika tidak ditemukan.

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

Jalankan program dan buka :

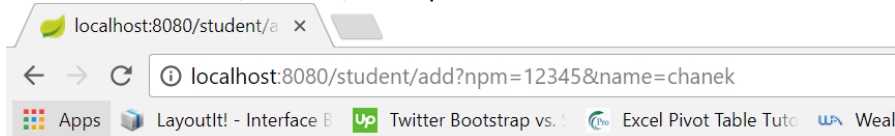
localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43



## Data berhasil ditambahkan

Pertanyaan 1 : apakah hasilnya? Jika error, tuliskan penjelasan Anda.  
- tidak error, karena parameter yang diminta terisi semua

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



## Whitelabel Error Page

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

Fri Sep 22 18:46:29 ICT 2017

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

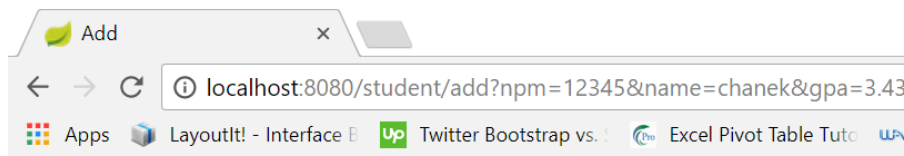
Required double parameter 'gpa' is not present

Pertanyaan 2: apakah hasilnya? Jika error , tuliskan penjelasan Anda.  
- terjadi error, karena parameter gpa yang diminta tidak ada

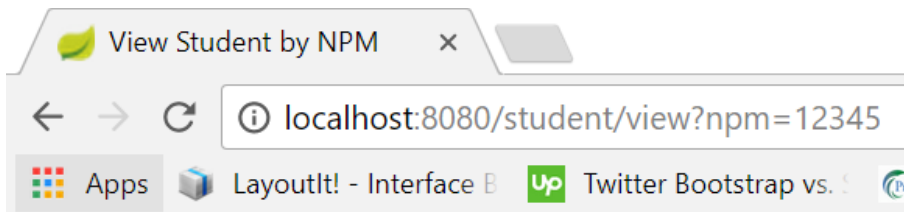
Jalankan program dan buka

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

localhost:8080/student/view?npm=12345



## Data berhasil ditambahkan



**NPM = 12345**

**Name = chanek**

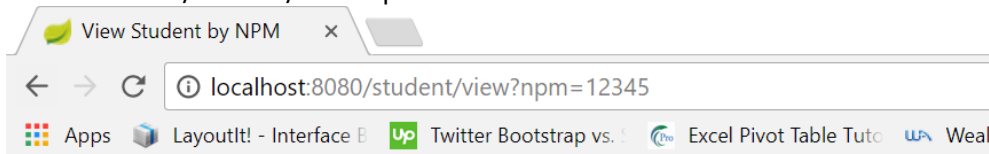
**GPA = 3.43**

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

- Akan 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.

Sat Sep 23 11:32:09 ICT 2017

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

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

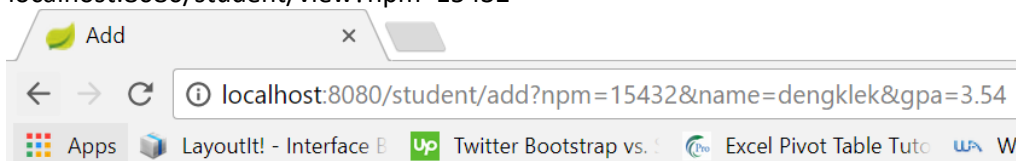
Pertanyaan 4 : apakah data Student tersebut muncul? Jika tidak, mengapa?

- Terjadi error, karena tidak ada data yang ditambahkan

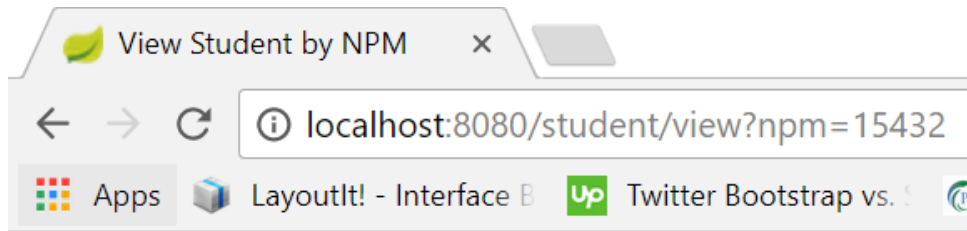
Coba tambahkan data Student lainnya dengan NPM yang berbeda.

localhost:8080/student/add?npm=15432&name=dengklek&gpa=3.54

localhost:8080/student/view?npm=15432



## Data berhasil ditambahkan



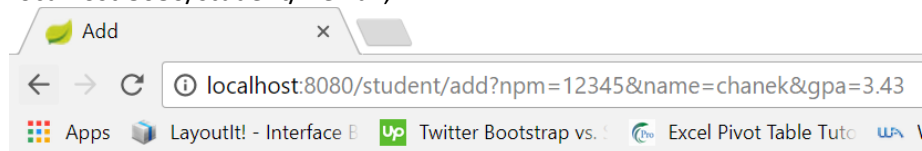
**NPM = 15432**

**Name = dengklek**

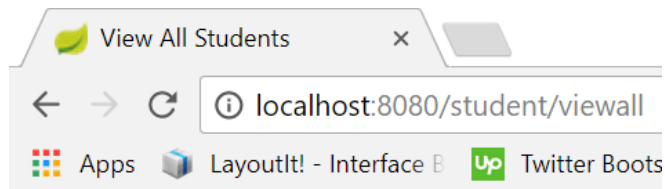
**GPA = 3.54**

Jalankan program dan buka

localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/viewall,



**Data berhasil ditambahkan**



**No. 1**

**NPM = chanek**

**Name = 12345**

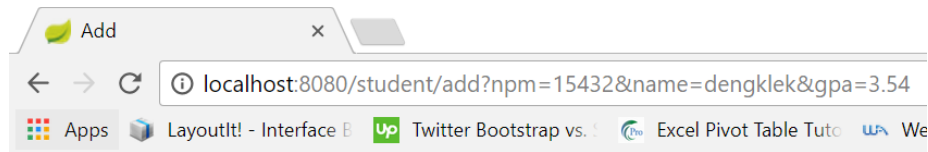
**GPA = 3.43**

---

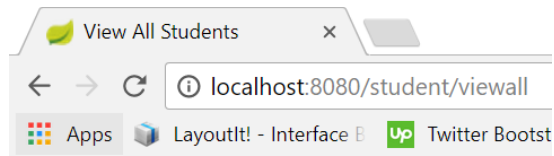
Pertanyaan 5 : apakah data Student tersebut muncul?

- Iya, semua data student tersebut muncul

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



## Data berhasil ditambahkan



**No. 1**

**NPM = chanek**

**Name = 12345**

**GPA = 3.43**

---

**No. 2**

**NPM = dengklek**

**Name = 15432**

**GPA = 3.54**

---

Pertanyaan 6 : Apakah semua data Student muncul?

- Iya, semua data yang telah ditambahkan muncul

c. Method selectStudent yang Anda implementasikan

Method yang diimplementasikan adalah

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

d. Penjelasan fitur delete yang Anda buat pada bagian 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

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

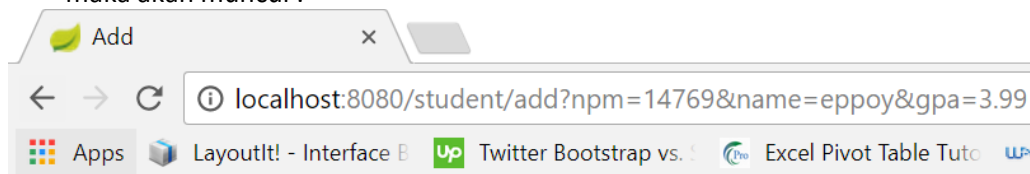
- melakukan perubahan pada method view, menjadi :

```

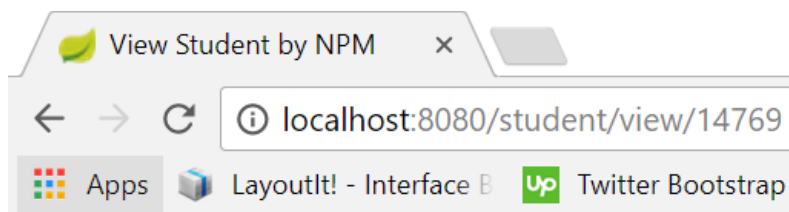
@RequestMapping(value = { "/student/view/{npm}", "/student/view" })
public String viewPath(@PathVariable Optional<String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if (student == null) {
            model.addAttribute("error", "Sorry, NPM not found!");
            return "error";
        } else {
            model.addAttribute("student", student);
            return "view";
        }
    } else {
        model.addAttribute("error", "Sorry, NPM is Empty!");
        return "error";
    }
}

```

- kemudian, menjalankan  
localhost:8080/student/add?npm=14769&name=eppoy&gpa=3.99 lalu buka  
localhost:8080/student/view/14769  
maka akan muncul :



## Data berhasil ditambahkan



**NPM = 14769**

**Name = eppoy**

**GPA = 3.99**

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.
  - menambahkan request mapping untuk halaman delete pada class StudentController

```

@RequestMapping(value = { "/student/delete/{npm}", "/student/delete" })
public String deletePath(@PathVariable Optional<String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.deleteStudent(npm.get());
        if (student == null) {
            model.addAttribute("error", "Sorry, NPM not found!");
            return "error";
        } else {
            model.addAttribute("student", student);
            return "delete";
        }
    } else {
        model.addAttribute("error", "Sorry, NPM is Empty!");
        return "error";
    }
}
}

```

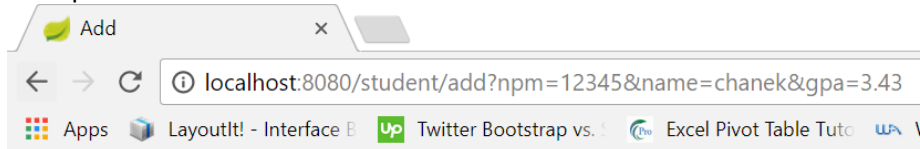
- kemudian menambahkan method delete pada class StudentService.java
- kemudian menambahkan method deleteStudent pada class InMemoryStudentService.java

```

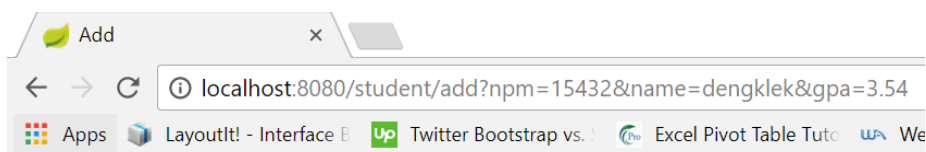
@Override
public StudentModel deleteStudent(String npm) {
    for (int i = 0; i < studentList.size(); i++) {
        if (npm.equalsIgnoreCase(studentList.get(i).getNpm())) {
            return studentList.remove(i);
        }
    }
    return null;
}

```

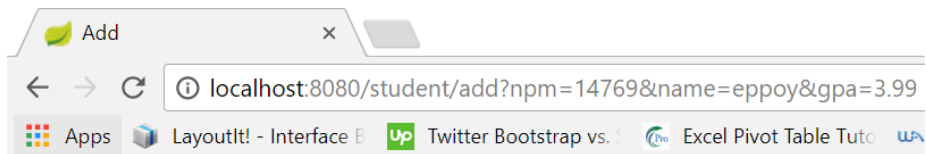
- kemudian menambahkan file html delete.html
- Hasil percobaan :



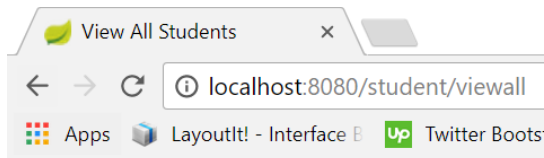
## Data berhasil ditambahkan



## Data berhasil ditambahkan



## Data berhasil ditambahkan



**No. 1**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

---

**No. 2**

**NPM = 15432**

**Name = dengklek**

**GPA = 3.54**

---

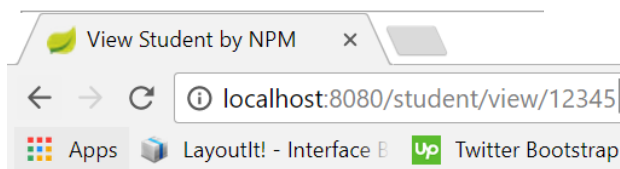
**No. 3**

**NPM = 14769**

**Name = eppoy**

**GPA = 3.99**

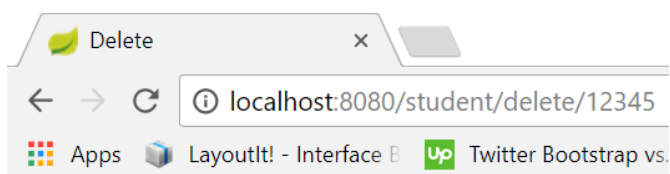
---



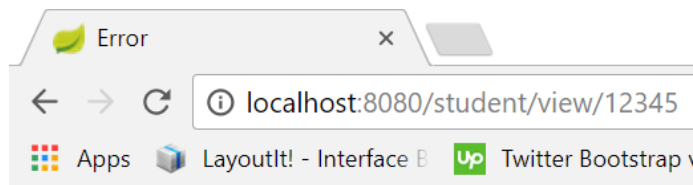
**NPM = 12345**

**Name = chanek**

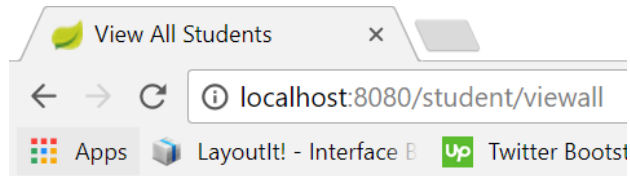
**GPA = 3.43**



**Data berhasil dihapus**



**Sorry, NPM not found!**



**No. 1**

**NPM = 15432**

**Name = dengklek**

**GPA = 3.54**

---

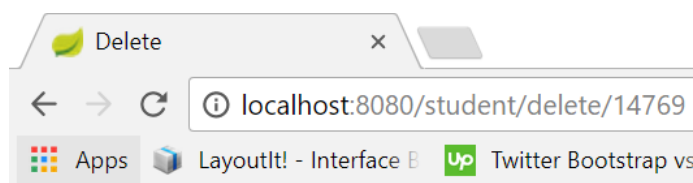
**No. 2**

**NPM = 14769**

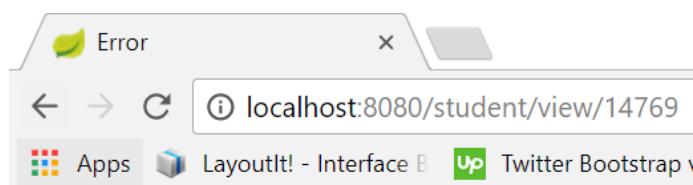
**Name = eppoy**

**GPA = 3.99**

---

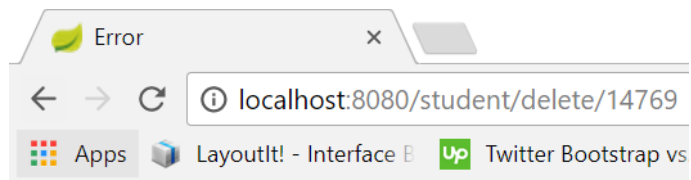


**Data berhasil dihapus**

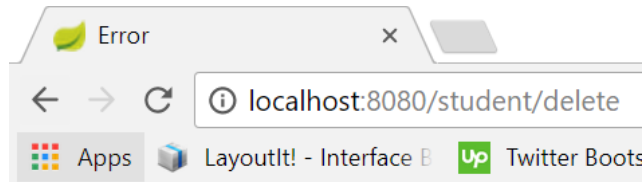


**Sorry, NPM not found!**

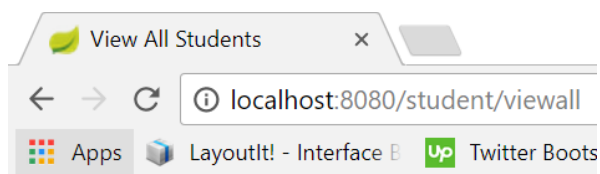




**Sorry, NPM not found!**



**Sorry, NPM is Empty!**



**No. 1**

**NPM = 15432**

**Name = dengklek**

**GPA = 3.54**

---