

Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792

## 1. Hasil ringkasan materi dalam tutorial kali ini

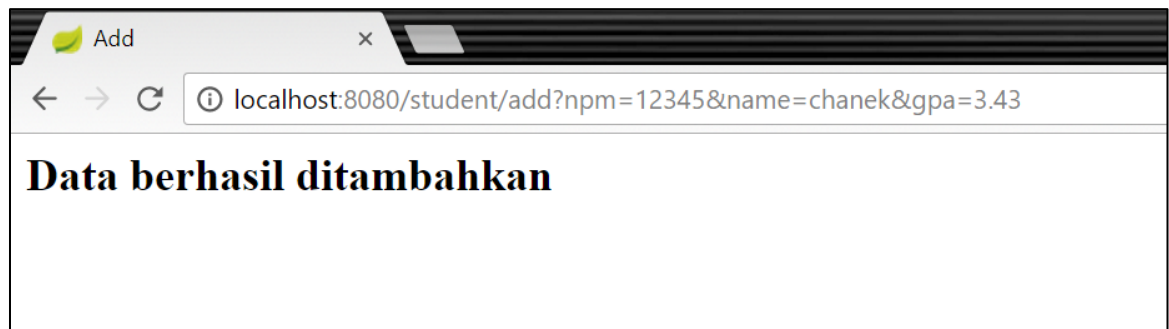
Pada tutorial ini lebih banyak akan mempraktikkan fungsi Service pada permodelan MVC. Service merupakan interface yang berisi method – method yang digunakan untuk menyimpan, memanggil ataupun menghapus atribut student yang ada di dalam List yang ada. Atribut yang disimpan didalam objek student adalah npm, nama, dan gpa.

## 2. Jawaban dari setiap poin pertanyaan beserta screenshot

a. **Pertanyaan 1** : buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43.

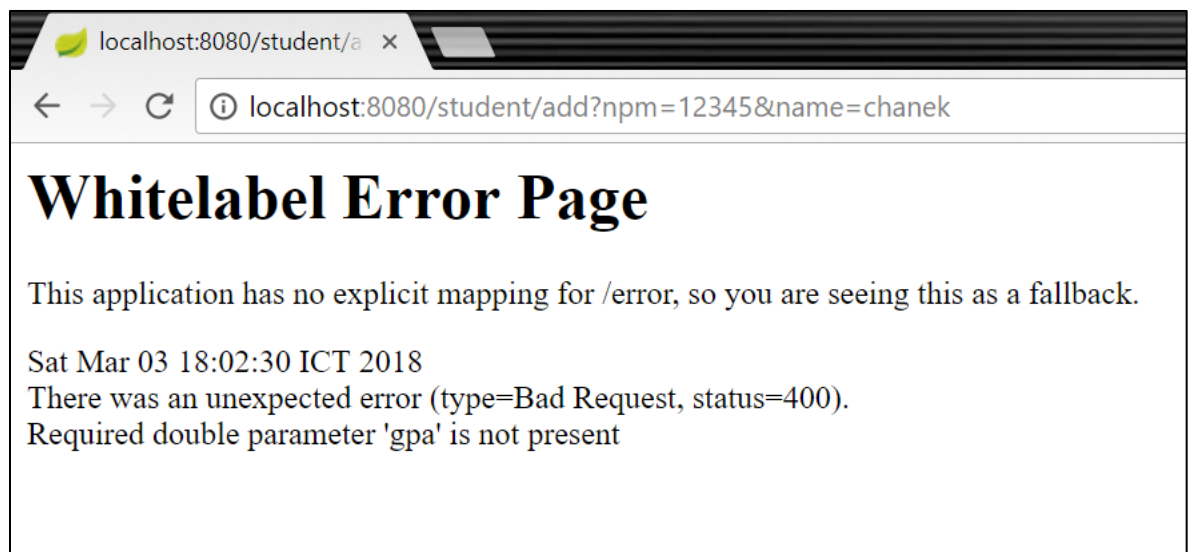
Apakah hasilnya? Jika *error* , tuliskan penjelasan Anda.

→ Hasilnya tidak error, data masuk seperti di gambar di bawah ini



b. **Pertanyaan 2**: buka localhost:8080/student/add?npm=12345&name=chanek. Apakah hasilnya? Jika *error* , tuliskan penjelasan Anda.

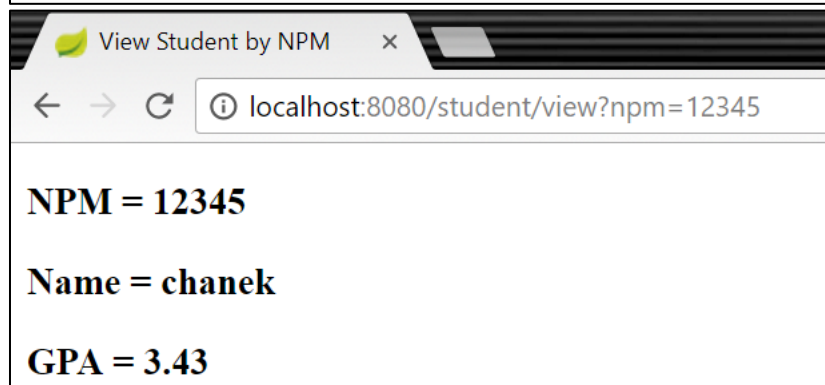
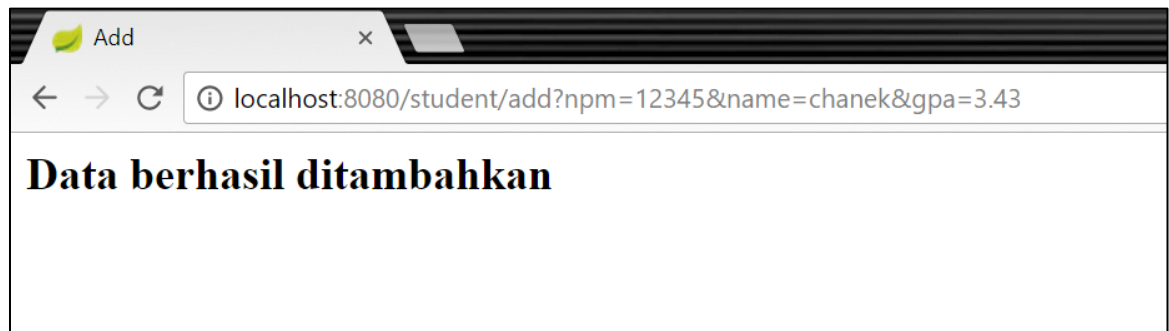
→ Hasilnya error, dikarenakan *variable* gpa tidak dideklarasikan untuk dimasukkan ke dalam konstruktor kelas StudentModel.



c. **Pertanyaan 3** : Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/view?npm=12345. Apakah data Student tersebut muncul? Jika tidak, mengapa?

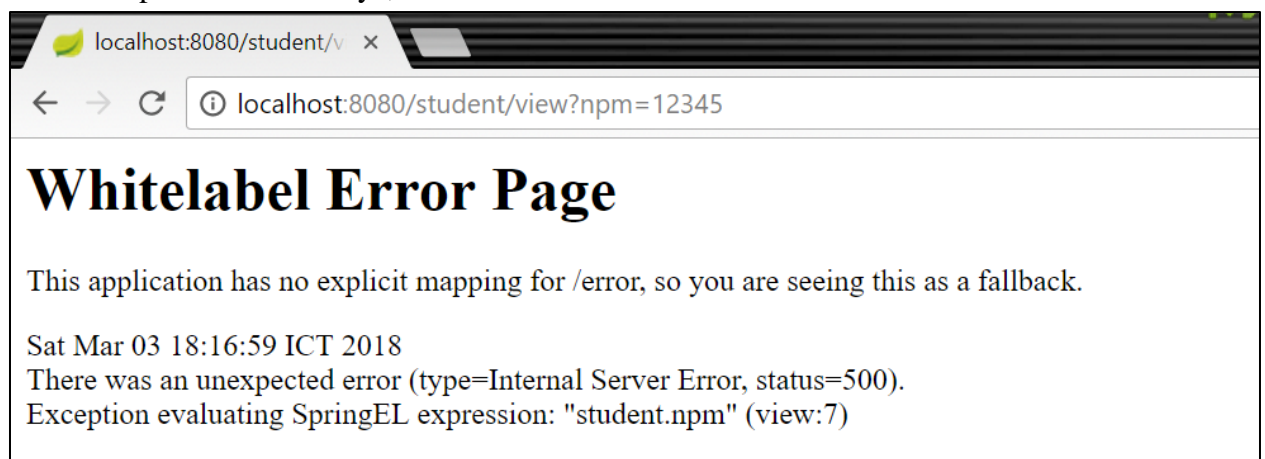
Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792

➔ Data yang ditambahkan muncul dan ketika dipanggil halaman view, muncul data yang baru saja ditambahkan



d. **Pertanyaan 4** : Coba matikan program dan jalankan kembali serta buka localhost:8080/student/view?npm=12345. Apakah data Student tersebut muncul? Jika tidak, mengapa?

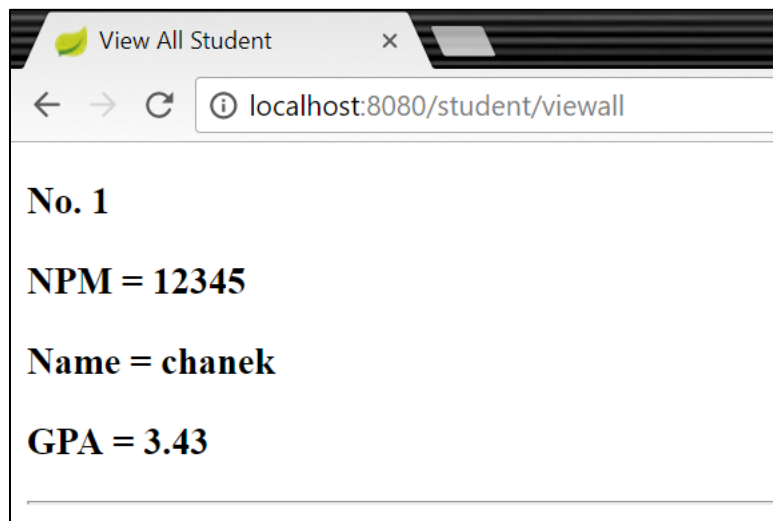
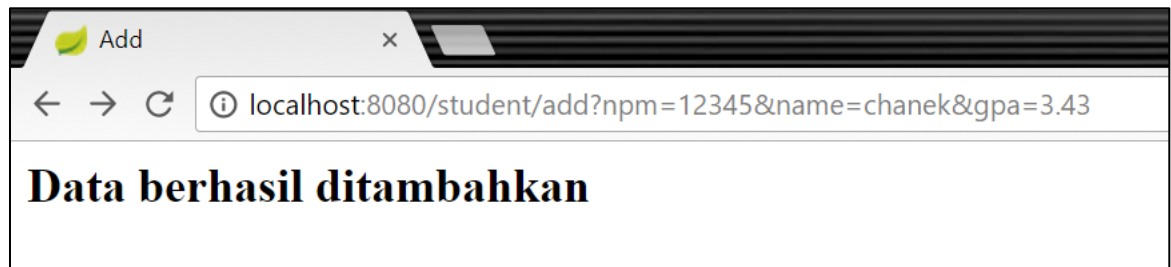
➔ Tidak, dikarenakan setiap service *spring boot* dimatikan, maka data yang telah disimpan ke dalam List akan ter-reset. Sehingga ketika dipanggil kembali NPM yang sudah diinputkan sebelumnya, muncul error.



e. **Pertanyaan 5** : Jalankan program dan buka localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43. Lalu buka localhost:8080/student/viewall. Apakah data Student tersebut muncul?

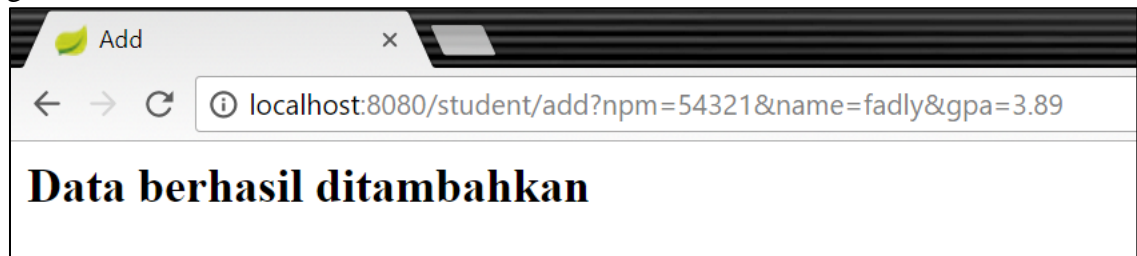
Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792

➔ Data yang sudah diinputkan sebelumnya muncul seperti gambar di bawah ini

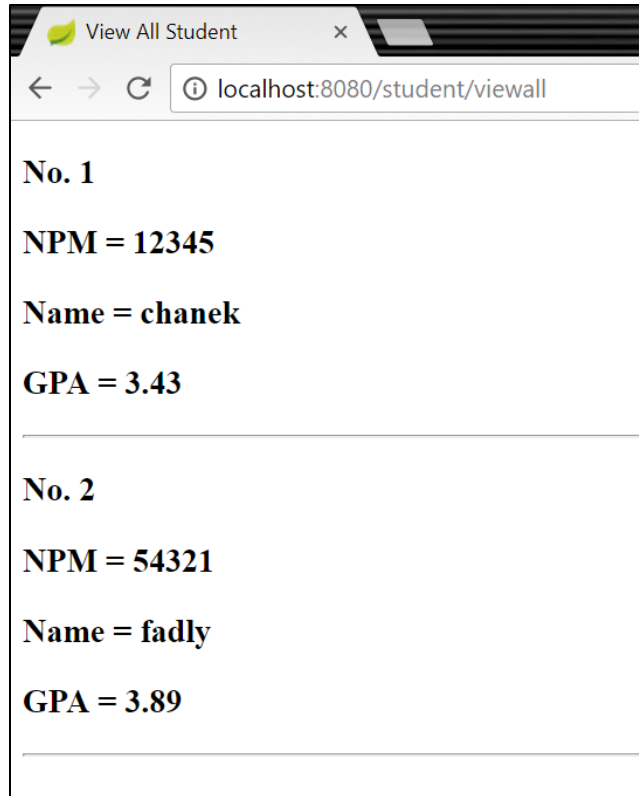


f. **Pertanyaan 6** : Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka localhost:8080/student/viewall. Apakah semua data Student muncul?

➔ Iya, semua data student yang ditambahkan muncul teratur dan looping seperti pada gambar dibawah ini



Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792



### 3. Penjelasan method Select Student

Method Select Student digunakan untuk melakukan pengambilan value dari instance student yang sudah disimpan ke dalam konstruktor kelas StudentModel. Ada 2 mekanisme yang harus dilakukan, antara lain :

- Penambahan method SelectStudent pada kelas InMemoryStudentService

Pada kelas InMemoryStudentService, ditambahkan method SelectStudent untuk melakukan pemanggilan student di dalam List static studentList yang digunakan untuk menyimpan atribut dari student yang datanya ber-tipe instance kelas StudentModel.

Method yang saya buat adalah :

```
@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;
}
```

Method ini akan menerima npm yang sudah ditangkap oleh link yang diinputkan pada browser kemudian dilakukan cek dengan List student yang sudah ada di dalam static list studentList dengan melakukan looping agar semua npm dapat dibandingkan. Jika npm nya ada yang sama maka akan dilakukan return npm tersebut. Jika tidak, maka akan return null.

Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792

b. Implementasi method selectStudent pada kelas StudentController

Method selectStudent digunakan pada instance variable student maupun instance lain untuk menampilkan atribut student sesuai dengan npm yang diinputkan untuk kemudian dilakukan pemanggilan method tsb di kelas studentService yang berguna untuk mengirim atribut ke view seperti baris kode berikut,

```
@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";
}
```

#### 4. Penjelasan fitur delete

Fitur delete berguna untuk melakukan penghapusan List Student yang sudah tersimpan di dalam list. Beberapa kelas yang harus ditambahkan untuk menjalankan method delete antara lain :

a. Kelas StudentControl

```
@RequestMapping(value = {"/student/delete/", "/student/delete/{npm}"})
public String delete(@PathVariable Optional<String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student==null) {
            model.addAttribute("student", "NPM Tidak Ditemukan");
            return "viewpathgagal";
        } else {
            model.addAttribute("student", student);
            studentService.deleteStudent(student);
            return "delete";
        }
    } else {
        return "viewnpmkosong";
    }
}
```

Kelas ini melakukan pengecekan pada path delete dengan inputan npm. Skema nya sama seperti viewStudent dimana akan dilakukan pengecekan npm yang sudah disimpan ke dalam list static, jika ada maka akan dilakukan pemanggilan method deleteStudent pada kelas InMemoryStudentService dan memanggil view delete.html

b. Kelas StudentService

```
public interface StudentService{
    StudentModel selectStudent(String npm);
    List<StudentModel> selectAllStudent();
    void addStudent(StudentModel student);
    void deleteStudent(StudentModel student);
}
```

Penambahan method pada interface StudentService

c. Kelas InMemoryStudentService

Nama : Fadly Muhammad Ridho  
Kelas : Ekstensi  
NPM : 1606954792

```
@Override  
public void deleteStudent(StudentModel student) {  
    // TODO Auto-generated method stub  
    studentList.remove(student);  
}
```

Penambahan method deleteStudent pada InMemoryStudentService

d. View delete.html

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

View Delete.html