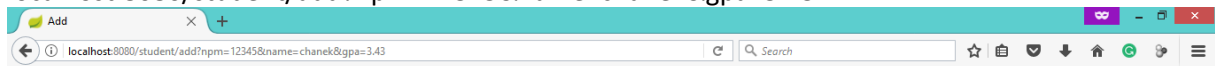


Jalankan program dan buka

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

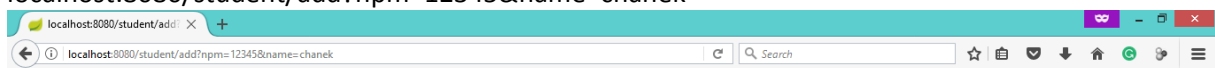


Data berhasil ditambahkan

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

**Jawaban 1 :** Hasilnya adalah akan ke direct ke halaman add lalu muncul tulisan 'Data berhasil ditambahkan'.

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.

Sat Sep 23 15:04:29 ICT 2017

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

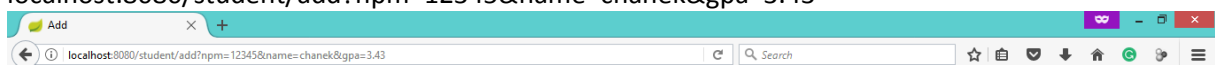
Required double parameter 'gpa' is not present

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

**Jawaban 2 :** Hasilnya error *bad request* dengan kode 400 karena ada salah satu komponen yang tidak ada yaitu gpa, sementara di *code gpa* itu *required*.

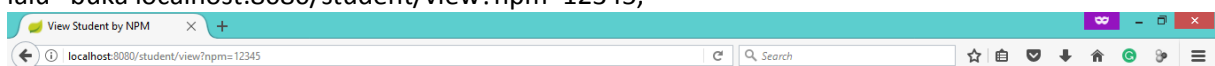
Jalankan program dan buka

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



Data berhasil ditambahkan

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



NPM = 12345

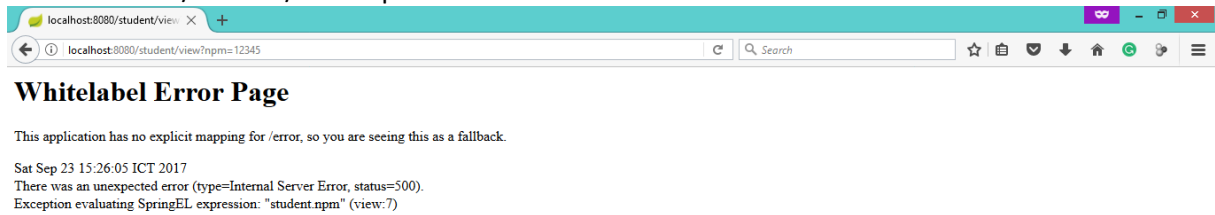
Name = chanek

GPA = 3.43

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

**Jawab 3:** data tersebut muncul karena sebelumnya kita sudah memasukkan data baru ke ArrayList setelah itu kita mengaksesnya.

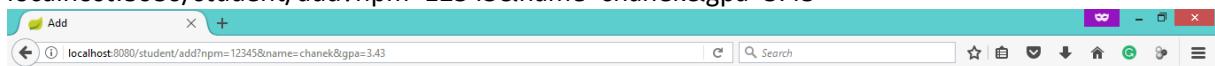
Coba matikan program dan jalankan kembali serta buka  
localhost:8080/student/view?npm=12345



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

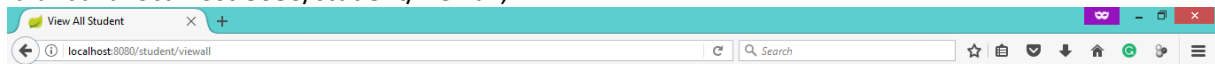
**Jawab 4:** data student tidak muncul karena saat dimatikan dan distart lagi programnya maka data koreset semua jadi belum ada data apa-apa di program. Makanya error.

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



**Data berhasil ditambahkan**

lalu buka localhost:8080/student/viewall,



**No. 1**

**NPM = 12345**

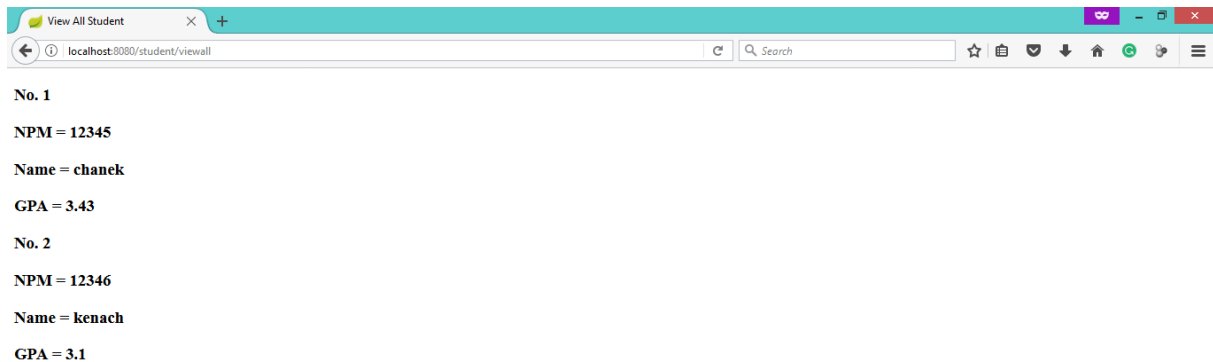
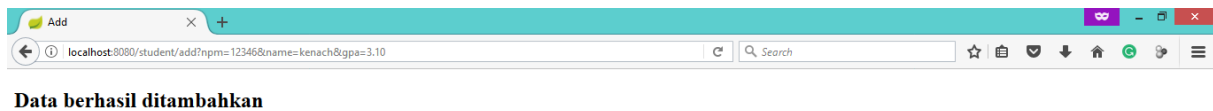
**Name = chanek**

**GPA = 3.43**

**Pertanyaan 5 :** apakah data Student tersebut muncul?

**Jawab 5 :** Ya, datanya muncul

Coba tambahkan data Student lainnya dengan NPM yang berbeda,



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

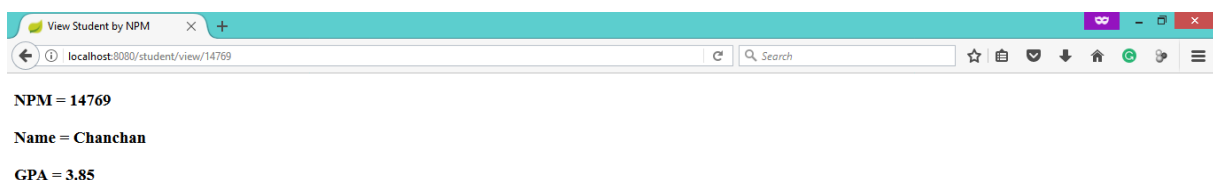
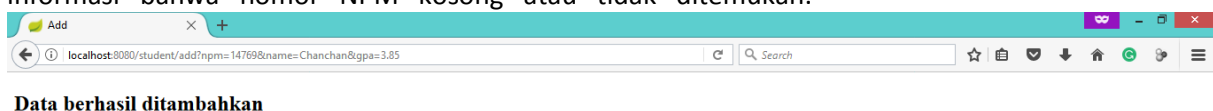
**Pertanyaan 6 : Apakah semua data Student muncul?**

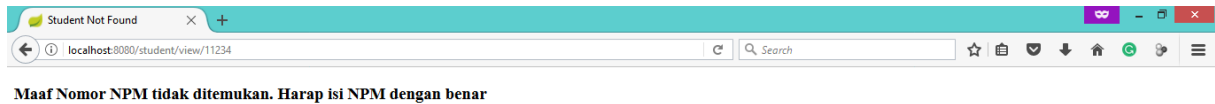
**Jawab 6:** Ya, semua data student muncul

### 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`**.

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

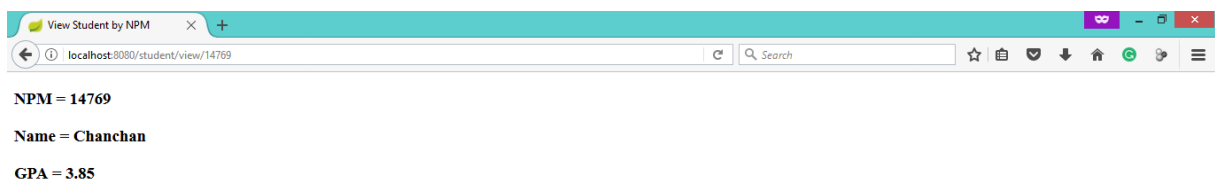
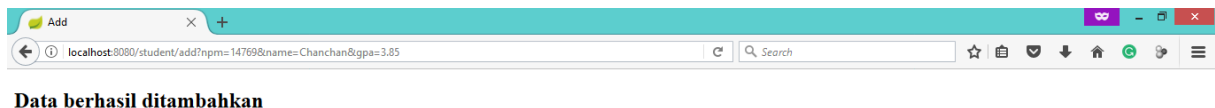


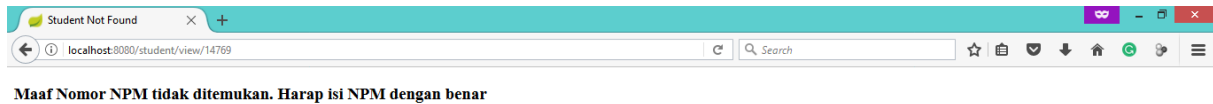
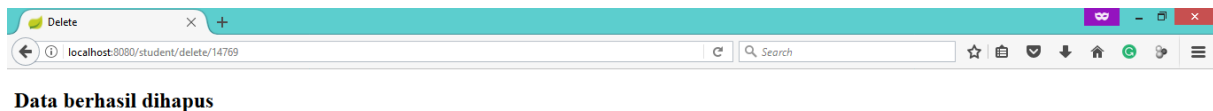


Jadi untuk fitur view ini menggunakan Path Variable yang menerima NPM dari mahasiswa, setelah itu akan muncul datanya yang sesuai. Jadi setelah menerima variable npm, maka variable tersebut akan menjadi index untuk mencari siswa yang npm-nya sesuai dengan menggunakan method `selectStudent()` di `StudentModel`. Setelah itu akan di `addAttribute` di modelnya. Setelahnya akan di return ke halaman view untuk menampilkan data. Kalau npm tidak ada akan di direct ke halaman error.

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.





Jadi untuk fitur delete ini pertama dibuat dulu method untuk remove-nya di `InMemoryStudent` lalu ditulis di `StudentInterfacenya`. Disini nama methodnya adalah `removeStudent()`. Setelah itu dibuat `RequestMapping` ke halaman delete, lalu methodnya menggunakan `Path Variable npm`. Variable `npm` akan digunakan sebagai index pencari data yang akan dihapus. Setelah studentnya ketemu nanti akan memanggil method `removeStudent()`. Untuk page delete menampilkan kalau data berhasil dihapus. Kalau data tidak ada atau error maka akan ke direct ke halaman error. Misalkan setelah kita menambahkan data (Gambar 1), kita cek dulu apakah datanya masuk (Gambar 2). Setelah memastikan datanya ada, kita coba hapus (Gambar 3). Setelah itu kita cek lagi dengan `npm` tersebut apakah berhasil di delete. Ternyata datanya sudah hilang (Gambar 4).

---Method yang dibuat pada tutorial kali ini---

#### `selectStudent()`

```
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;  
}
```

#### `removeStudent`

```
public void removeStudent(StudentModel student) {  
    studentList.remove(student);  
}
```

```
}
```

## deleteStudent

```
@RequestMapping("student/delete/{npm}")
    public String delete(@PathVariable String npm) {

        StudentModel student = studentService.selectStudent(npm);
        studentService.removeStudent(student);

        return "delete";
    }
```

## View

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

## ---Lesson Learned dan Ringkasan---

Jadi pada tutorial kali ini saya belajar banyak. Yang pertama adalah ternyata kelas-kelas bisa dimasukkan ke package yang berbeda-beda asalkan tetap inherit nama package dari Main Classnya. Package controller untuk yang mendirect ke halaman, model untuk objek-objeknya dan service untuk method-method yang akan digunakan. Selain itu ternyata iterasi di html yang pakai Spring bisa pakai th:each. Selain itu tiap bikin project baru mavenya harus update, bisa pakai update project dan kalau waktu di run masih error, hapus repository di .m2 dan lakukan Maven Clean dan Maven Install. Pastikan internet stabil karena jika internet tidak stabil kemungkinan programnya masih error meskipun sudah ditungguin lama dan dalam hitungan jam. Katanya karena jar-nya mungkin ada yang *corrupt*.