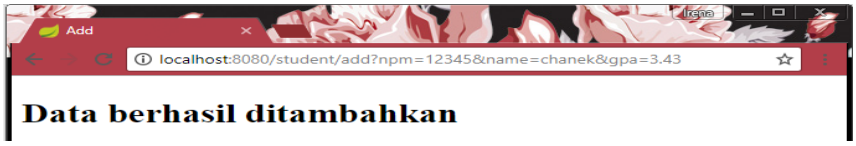
A. RINGKASAN MATERI

- Pada tutorial kali ini mempelajari tentangg implementasi konsep model-view-controller(MVC) menggunakan Spring Boot dengan menggunakan kompenan model(Objek) dan service(layer yang menjadi mediator antara controller dan database) yang akan digunakan saat pembuatan fungsi create dan read data.
- Tahapan yang dilakukan adalah membuat sebuah kelas model, lalu membuat kelas service yang didalamnya memiliki sebuah kelas interface dan kelas yang mengimplementasi interface tersebut. Selanjutnya membuat sebuah controller dan fungsi-fungsi yang ingin dilakukan.

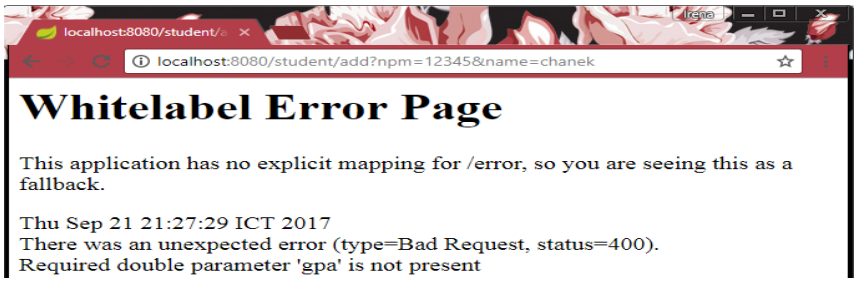
B. HASIL JAWABAN

- MEMBUAT CONTROLLER DAN FUNGSI ADD

  localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43
  Hasilnya adalah akan ditampilkan “Data berhasil ditambahkan”



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



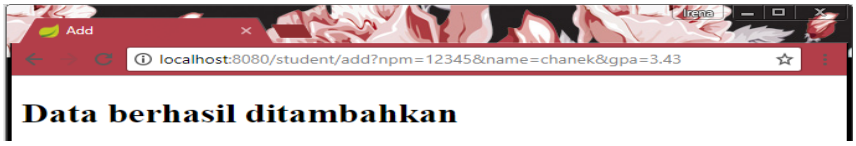
  Karena pada StudentController.java dituliskan kode seperti berikut :

```
RequestParam(value = "gpa", required = true) double gpa
```

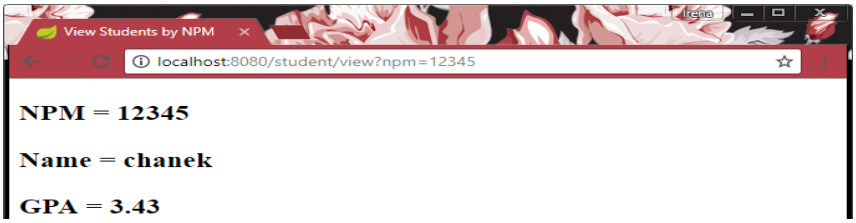
  Terdapat required = true, sehingga nilai dari gpa harus di masukkan/ada (tidak boleh kosong).

- METHOD VIEW BY NPM

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

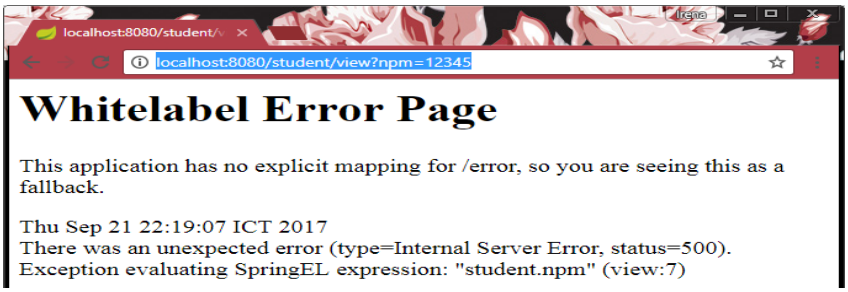


  Lalu buka, **localhost:8080/student/add?npm=12345**



Hasilnya adalah data student dengan npm=12345 dimunculkan.

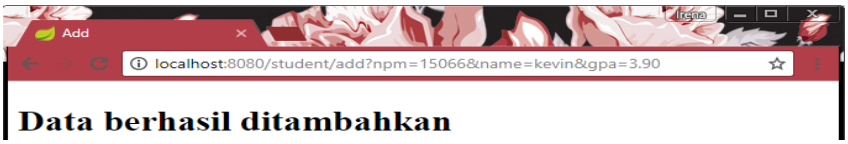
Mematikan program dan menjalankan kembali dengan membuka
Localhost:8080/student/view?npm=12345
Hasilnya adalah error



Muncul pesan error karena kita telah menghentikan jalannya program yang telah berisi input peserta dan menjalankan kembali program dari awal saat belum ada input data NPM, nama dan GPA peserta yang disimpan, sehingga saat kita ingin melihat peserta dengan npm=12345 maka data yang dicari tidak ada dan memunculkan error.

Menambahkan data student lainnya dengan NPM berbeda

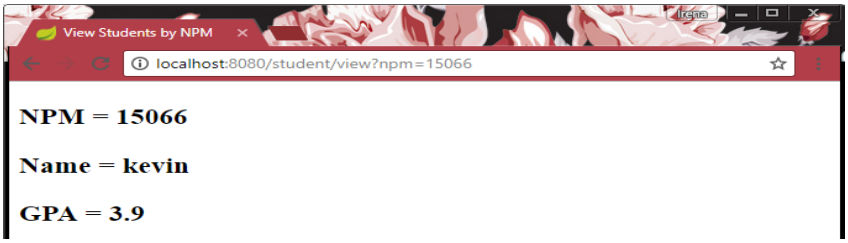
Student 1



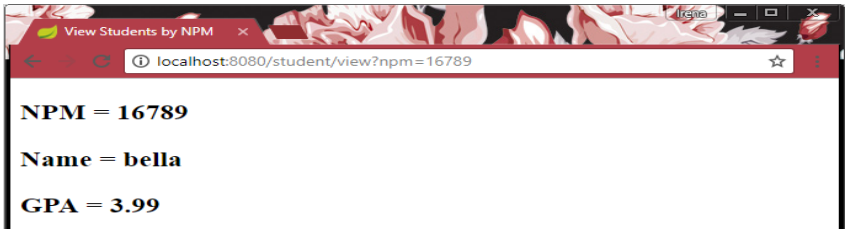
Student 2

Menampilkan data student yang telah berhasil dimasukkan

Student 1

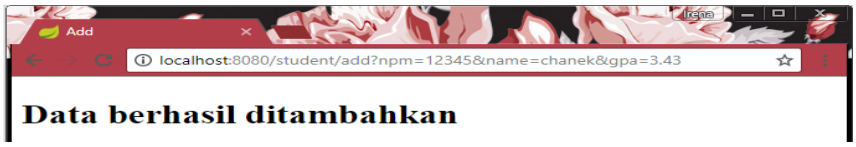


Student 2

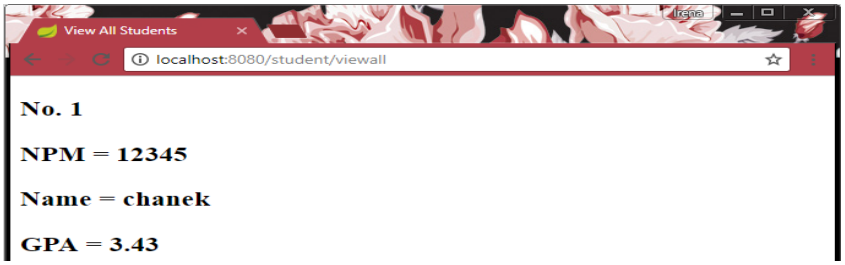


- METHOD VIEW ALL

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



Lalu buka, **localhost:8080/student/viewall;**



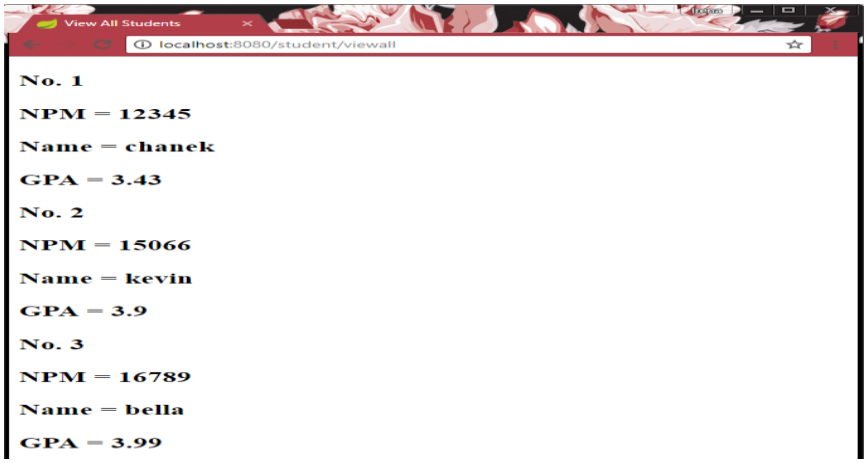
Ya, data student yang dimasukkan muncul dan ditampilkan urutan nomornya seperti pada gambar diatas.

Menambahkan data student lainnya dengan NPM berbeda

Student 1

Student 2

Setelah membuka **localhost:8080/student/viewall**



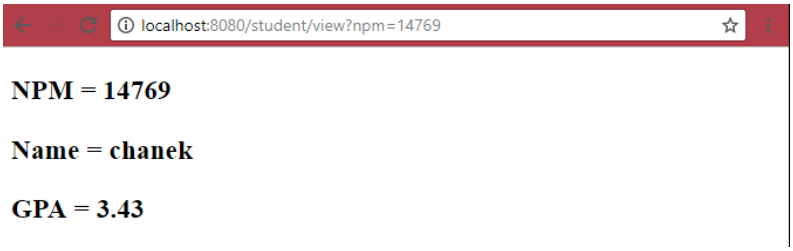
Ya, semua data student muncul dengan nomor urutan data dimasukkan

- 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.**

     Memasukan data student dengan NPM = 14769

     Mengakses halaman localhost:8080/student/view/14769

     Menghentikan program dan menjalankan kembali localhost:8080/student/view/14769

  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.

C. METHOD SELECT STUDENT

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

D. PENJELASAN FITUR DELETE

- StudentService.java

```
public interface StudentService {

    StudentModel selectStudent(String npm);
    boolean deleteStudent(String npm);
    List<StudentModel> selectAllStudents();
    void addStudent(StudentModel student);
}
```

Dalam kelas public interface StudentService.java ditambahkan interface untuk mendefinisikan method untuk menghapus data student berdasarkan npm.

- InMemoryStudentService.java

```
@Override
public boolean deleteStudent(String npm) {
    for(int i = 0; i < studentList.size(); i++) {
        StudentModel student = studentList.get(i);
        if(student.getNpm().equals(npm)) {
            studentList.remove(i);
            return true;
        }
    }
    return false;
}
```

Dalam kelas InMemoryStudentService.java ditambahkan sebuah method untuk menghapus data student dengan NPM. Dalam method ini saya menggunakan public Boolean untuk mereturn hasil true jika data student yang akan dihapus sama dengan NPM data student yang disimpan telah berhasil dihapus. Jika data student dengan NPM tidak sama atau data yang disimpan tidak ada, maka return false

- StudentController.java

```
@RequestMapping(value = {"/student/delete/{npm}"})
public String deletePath(@PathVariable Optional<String> npm, Model model) {
boolean isDeleted = studentService.deleteStudent(npm.get());
    if(!isDeleted) {
        return "deleteBatal";
    }else{
        return"delete";
    }
}
```

Dalam kelas StudentController.java dibuat anotasi @RequestMapping dengan menggunakan path variable untuk menghapus data student hanya berdasarkan NPM. Dikatakan pada soal jika nomor NPM tidak diberikan, maka itu kita menggunakan Optional<String> npm. Lalu method deleteStudent dengan mengambil npm akan dijalankan dan hasilnya akan disimpan di dalam variable isDeleted yang bertipe Boolean, dimana pada method delete student akan di cek apakah NPM yang ingin dihapus sama dengan data NPM student yang telah disimpan, jika sama maka akan direturn true dan data NPM yang ingin dihapus akan dihapus, lalu jika berhasil dihapus maka akan return ”delete” di kelas StudentController.java yang akan menandakan bahwa ada request HTTP ke delete.html, sehingga akan ditampilkan isi dari delete.html yaitu “Data berhasil dihapus”

```
1  <!DOCTYPE html>
2  <html xmlns="http://www.thymeleaf.org">
3  <head>
4  <title>Delete Student by NPM</title>
5  </head>
6  <body>
7      <h2>"Data berhasil dihapus"</h2>
8  </body>
9  </html>
```

Apabila saat dicek NPM tidak sama dengan NPM yang disimpan atau belum ada data yang dimasukkan maka akan mereturn false yang artinya method deleteStudent tidak dijalankan sehingga tidak ada data yang dihapus. Oleh karena itu pada kelas

StudentController.java akan return “deleteBatal” yang akan menandakan ada request HTTP ke deleteBatal.html, sehingga akan ditampilkan isi dari deleteBatal.html yaitu “NPM kosong atau tidak ditemukan dan proses delete dibatalkan”

```
<!DOCTYPE html>
<html xmlns="http://www.thymeleaf.org">
<head>
<title>Delete Students by NPM</title>
</head>
<body>
    <h2>"NPM kosong atau tidak ditemukan dan proses delete dibatalkan"</h2>
</body>
</html>
```