

a. Ringkasan dari materi yang Anda telah pelajari pada tutorial kali ini

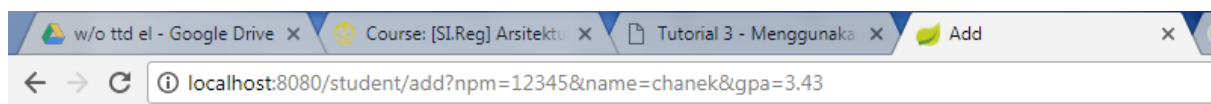
- Mempelajari bagaimana cara memanggil project java yang kita buat melalui web
- Membuat model dengan konsep MVC dalam project Spring Boot
- Membuat service dengan fungsi create dan read data melalui konsep MVC

Mengimplementasikan bagaimana cara untuk menambahkan, menghapus, dan melihat data yang sudah kita input ketika menggunakan SpringBoot. Tapi data yang kita input tersebut belum dimasukkan ke dalam database sehingga ketika kita *stop* dan *start* program dari awal maka data yang kita input tadi hilang jadi seperti *temporarily* saja datanya tidak tersimpan ke dalam sebuah tempat. Mengerti cara input data dengan GET request dan menyelaraskannya dengan parameter di model yang telah kita buat, karena ketika pada saat melakukan tutorial ini saya melakukan kesalahan yaitu saya salah memposisikan parameter sehingga saya mengalami *error*, ternyata ada hikmah dibalik itu yaitu saya jadi mengerti sekali pentingnya ketepatan dan ketelitian kita selagi membuat program sehingga tidak menimbulkan error dengan kesalahan yang kecil namun berakibat fatal.

b. Hasil jawaban dari setiap poin pada bagian tutorial (dapat didukung dengan screenshot)

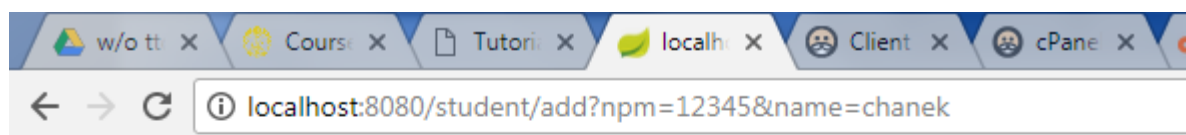
1. localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

Pertanyaan 1: apakah hasilnya? **Tidak**



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

Pertanyaan 2: apakah hasilnya? **Ya, karena value gpa tidak ada (sama dengan null)**



Whitelabel Error Page

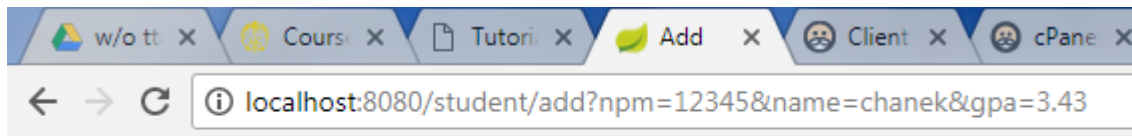
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Sep 22 16:39:43 ICT 2017

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

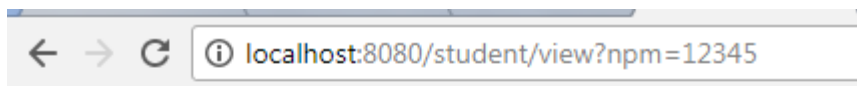
Required double parameter 'gpa' is not present

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



Data berhasil ditambahkan

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



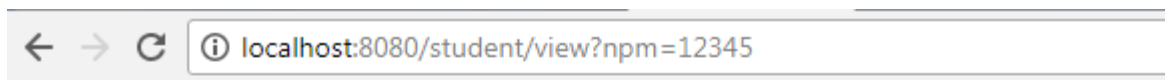
NPM = 12345

Name = chanek

GPA = 3.43

Pertanyaan 3: apakah data Student tersebut muncul? **Muncul**

4. 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.

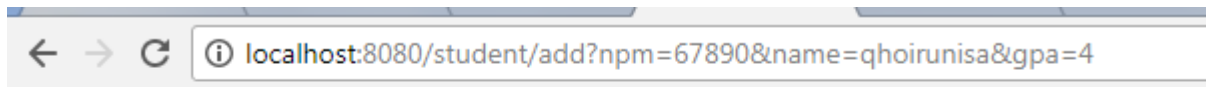
Sat Sep 23 10:45:51 ICT 2017

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? **Tidak, karena ketika kita stop program maka program seperti ter-refresh dan data student yang kita masukkan tadi tidak tersimpan karena kita tidak menyimpan data student tersebut di dalam database.**

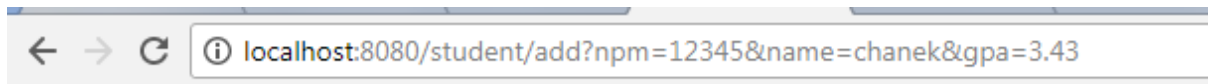
5. Coba tambahkan data Student lainnya dengan NPM yang berbeda.



Data berhasil ditambahkan

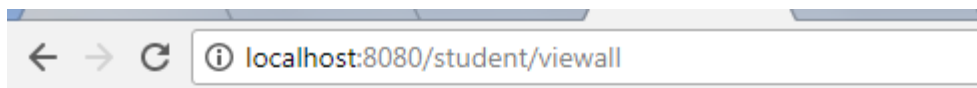
Method View All

1. 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

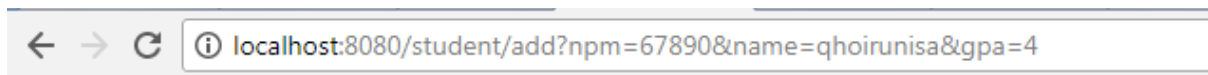
NPM = 12345

Name = chanek

GPA = 3.43

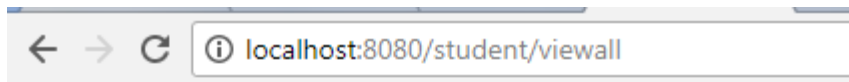
Pertanyaan 5: apakah data Student tersebut muncul? **Ya**

2. Coba tambahkan data Student lainnya dengan NPM yang berbeda



Data berhasil ditambahkan

lalu buka localhost:8080/student/viewall



No. = 1

NPM = 12345

Name = chanek

GPA = 3.43

No. = 2

NPM = 67890

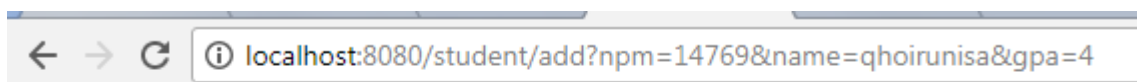
Name = qhoirunisa

GPA = 4.0

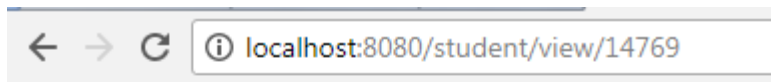
Pertanyaan 6: Apakah semua data Student muncul? **Ya**

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.



Data berhasil ditambahkan



NPM = 14769

Name = qhoirunisa

GPA = 4.0

Jika nomor NPM tidak diberikan



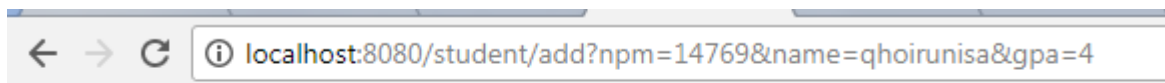
page not found, you must enter your NPM

atau tidak ditemukan



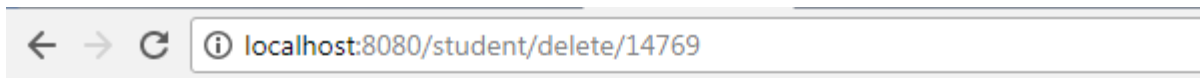
NPM not found, you enter a wrong NPM

2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. Misalnya, setelah melakukan
 - add Student pada soal nomor 1



Data berhasil ditambahkan

- 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.



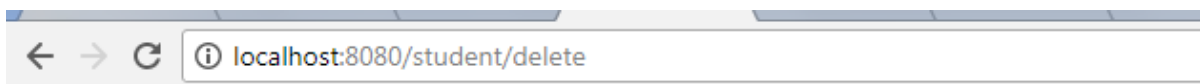
Data student dengan data berikut berhasil dihapus

NPM = 14769

Name = qhoirunisa

GPA = 4.0

- Jika nomor NPM tidak diberikan atau kosong



page not found, you must enter your NPM

- Jika salah memasukkan nomor NPM dan nomor NPM tidak ditemukan di dalam database



NPM not found, you enter a wrong NPM

c. Method selectStudent yang Anda implementasikan

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

```
    }
}
```

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

- Pertama saya membuat method deleteStudent di kelas InMemoryStudentService

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

- Lalu saya implementasikan deleteStudent pada interface StudentService

```
void deleteStudent(StudentModel student);
```

- Lalu saya buat controller untuk deleteStudent

```
@RequestMapping(value = { "/student/delete", "student/delete/{npm}" })
public String deletePath(@PathVariable Optional<String> npm, Model model) {

    //cek apakah parameter npm ada
    if (npm.isPresent()) {
        //select student yang ada pada List dengan npm yang sesuai
        dengan parameter
        StudentModel student =studentService.selectStudent(npm.get());

        //cek misalkan nilainya null (tidak ketemu)
        if(student == null) {
            // maka munculkan halaman error
            model.addAttribute("student", "NPM not found, you enter
a wrong NPM");
            return "error";
        } else {
            //jika nilainya tidak null yang berarti ketemu maka panggil
            method deleteStudent lalu munculkan page delete
            studentService.deleteStudent(student);
            model.addAttribute("student", student);
            return "delete";
        }
    } else {
        //jika kita tidak memasukkan nilai NPM maka akan dimunculkan
        page error
        model.addAttribute("student", "page not found, you must enter
your NPM");
        return "error";
    }
}
```