Ringkasan Materi

Pada tutorial kali ini membahas dan mempelajari beberapa hal, diantaranya:

1. Membuat data list yang dapat menyimpan data, hapus data, dan view data
2. Membuat service yang berisi method apa saja yang digunakan untuk memanipulasi data list dari student.
3. Membuat object student yang terdiri dari npm,nama dang gpa.
4. Membuat model yang merepresentasikan dan menyimpan informasi atau data.

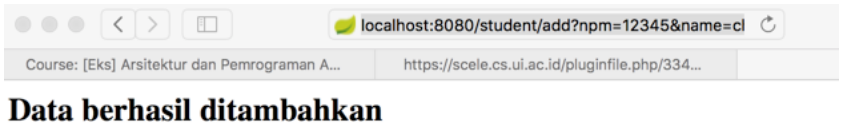
Tutorial

I. Membuat method selectStudent pada class InMemoryService

```
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;
}
```

II. Membuat Fungsi ADD

a. localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43
Pertanyaan 1 : apakah hasilnya? Jika *error* , tuliskan penjelasan Anda.



Penjelasan : Data berhasil ditambah, karena semua parameter terpenuhi.

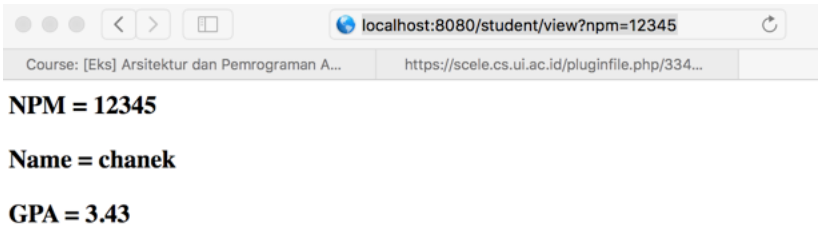
b. localhost:8080/student/add?npm=12345&name=chanek
Pertanyaan 2: apakah hasilnya? Jika *error* , tuliskan penjelasan Anda

Penjelasan: error karena parameter gpa tidak terpenuhi

III. Membuat view

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

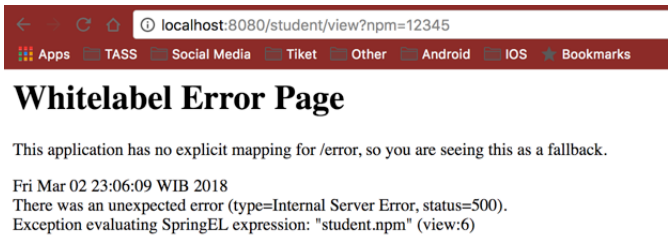
Pertanyaan 3 : apakah data Student tersebut muncul? Jika tidak, mengapa?



Penjelasan : Karena data sudah dimasukkan ke dalam list.

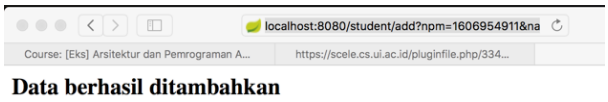
b. Coba matikan program dan jalankan kembali serta buka localhost:8080/student/view?npm=12345

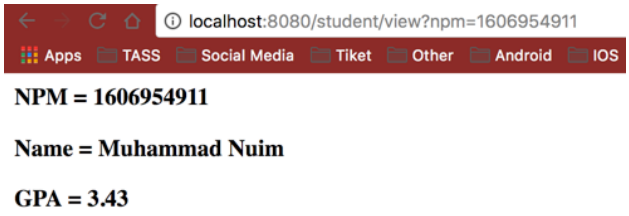
Pertanyaan 4 : apakah data Student tersebut muncul? Jika tidak, mengapa?



Penjelasan : Error karena program dimatikan dan ketika dihidupkan kembali maka list ter-*reset* dan data terhapus sehingga npm tidak ditemukan

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

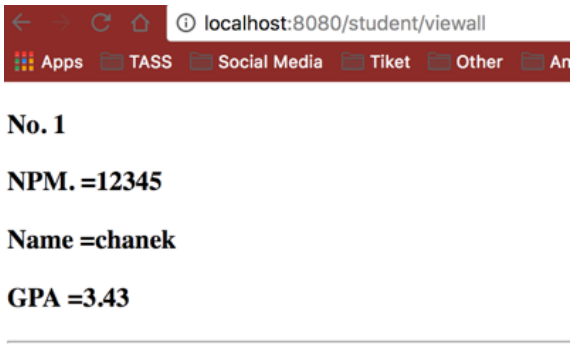




IV. Membuat View All

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

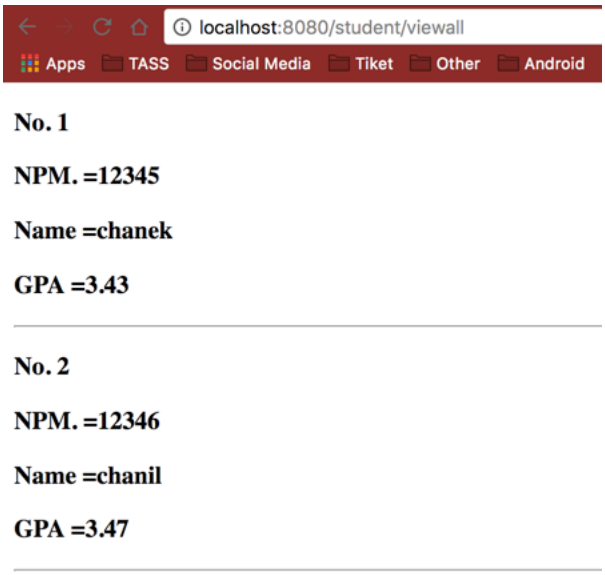
Pertanyaan 5 : apakah data Student tersebut muncul? **Muncul**



b. Tambahkan data student lainnya

localhost:8080/student/add?npm=12346&name=chanil&gpa=3.47

localhost:8080/student/viewall,
Pertanyaan 6 : Apakah semua data Student muncul? **Muncul**



V. Latihan

a. 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.

Tahapan:

1. Membuat request mapping baru pada StudentControll
2. Membuat method baru dengan menggunakan pathVariable

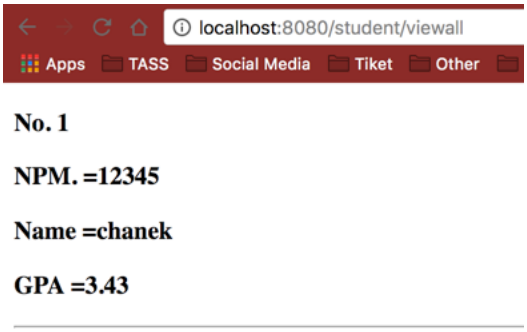
```
@RequestMapping(value = {"/student/view","/student/view/{npm}"})
public String studentPathView (@PathVariable Optional<String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if (student != null) {
            model.addAttribute("student", student);
        }else {
            model.addAttribute("npm", npm.get());
            return "notfound";
        }
    }else {
        return "notfound";
    }
    return "view";
}
```

3. Membuat file html jika npm tidak ditemukan atau parameter kosong

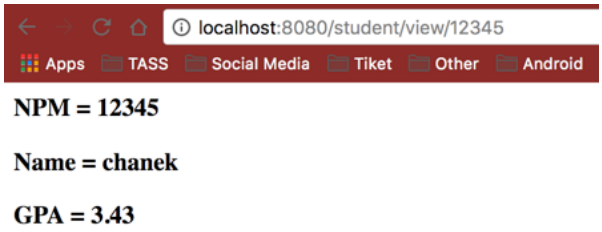
```
<html>
<head>
<title>View Student by NPM</title>
</head>
<body>
    <h3 th:text="'Data Tidak Ditemukan ' ">Data Tidak ada</h3>

</body>
</html>
```

Output:

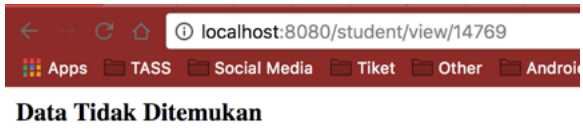
Data Awal

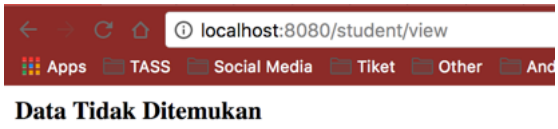


a. Jika data tersedia



b. Jika data tidak tersedia atau kosong





b. 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.

Tahapan:

1. Membuat service delete pada StudentService

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

2. Membuat method delete pada **InMemoryStudentService** pada method ini akan menghapus *object* student.

```
@Override
public void delStudent(StudentModel student) {
    studentList.remove(student);

}
```

3. Membuat method delete pada StudentController dengan menggunakan pathVariable. Pada method ini pertama kali akan mencari object student berdasarkan npm menggunaan method selectStudent kemudian setelah menemukan object yang dicari maka object tersebut akan dihapus dari list.

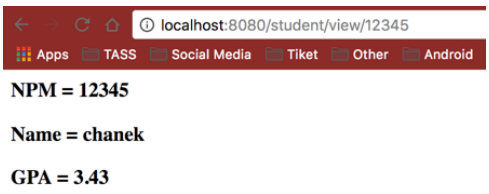
```
@RequestMapping(value = {"/student/delete","/student/delete/{npm}"})
public String studentPathDel (@PathVariable Optional<String> npm, Model model) {
    if(npm.isPresent()) {
        StudentModel st = studentService.selectStudent(npm.get());
        if (st != null) {
            model.addAttribute("student", st);
            studentService.delStudent(st);
        }else {
            model.addAttribute("npm", npm.get());
            return "notfound";
        }
    }else {
        return "notfound";
    }

    return "delete";
}
```

4. Membuat file delete.html

```
<html>
<head>
<title>Delete Student by NPM</title>
</head>
<body>
    <h3 th:text="'Data Mahasiswa dengan NPM = ' + ${student.npm} + ' Telah dihapus'"></h3>
</body>
</html>
```

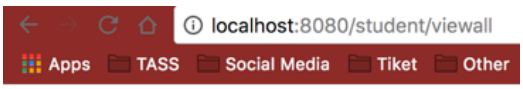
Output:

Data Awal



1. Jika data tersedia dan dapat di delete.

Dan pada saat mengakses viewall maka data kosong



2. Jika data yang ingin di delete tidak tersedia atau npm tidak diberikan