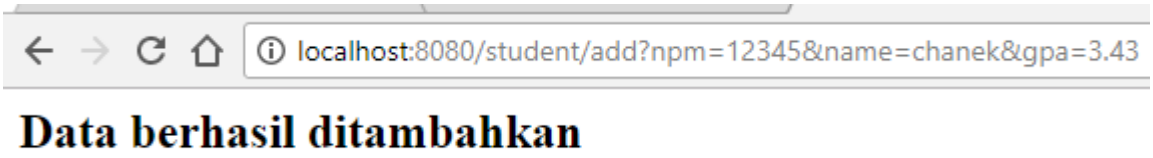
WRITE-UP

Pada tutorial ketiga kali ini, saya diminta untuk membuat model dengan konsep MVC (*model-view-controller*) dalam project Spring Boot serta Membuat service dengan fungsi *create* dan *read* data melalui konsep MVC. Saya diajarkan bagaimana Membuat Class Model pada eclipse, kemudian bagaimana cara membuat service-nya dengan studi kasus sistem informasi mahasiswa yang sederhana. Kemudian dilanjutkan dengan pembuatan Controller dan Fungsi Add untuk menambahkan data mahasiswa (*student*) ke ArrayList yang ada pada *data access object* (DAO). Kemudian untuk melihat apakah data benar-benar tersimpan saya juga diajarkan untuk mengimplementasikan method view by NPM(nomor mahasiswa) serta Method View All untuk melihan seluruh list mahasiswa pada sistem informasi sederhana yang saya buat.

Pada tutorial tersebut terdapat beberapa pertanyaan terkait dengan hal – hal yang baru diajarkan. Saya akan mencoba untuk menjawab pertanyaan pertanyaan tersebut.

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

Jawab:



- Data tersebut berhasil ditambahkan kedalam array List, Request Parameter pada Request Mapping tersebut sudah benar, yaitu npm, name, dan gpa.

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

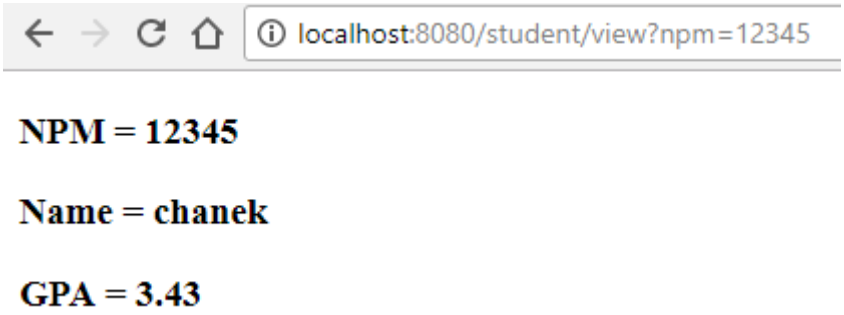
Jawab:

- Akan terjadi Whitelabel Error, dikarenakan Request Parameter pada Request Mapping tersebut tidak benar, hanya terdapat npm dan name saja, tidak terdapat gpa.

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?

Jawab :

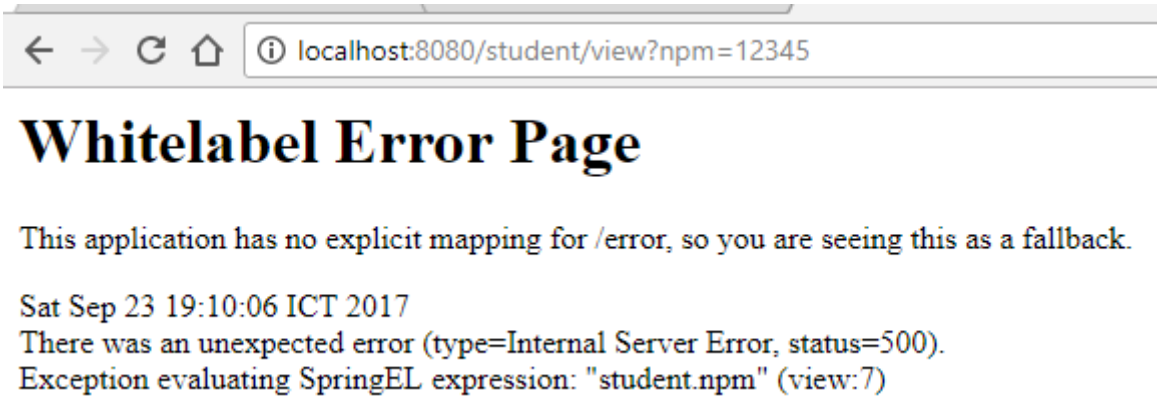


- Data tersebut akan muncul dikarenakan penambahan ke Array List students sebelumnnya telah berhasil. Ketika kita memanggil Request Mapping view npm tersebut, maka data tersebut akan otomatis tampil.

4. 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 :



- Ketika kita mematikan program dan menjalankannya kembali, maka Array List students akan otomatis terhapus. Sehingga, ketika kita langsung memanggil method view maka akan menimbulkan Whitelabel Error. Kita harus menambahkan (add) data mahasiswa tersebut terlebih dahulu.

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

Pertanyaan 5: apakah data Student tersebut muncul?

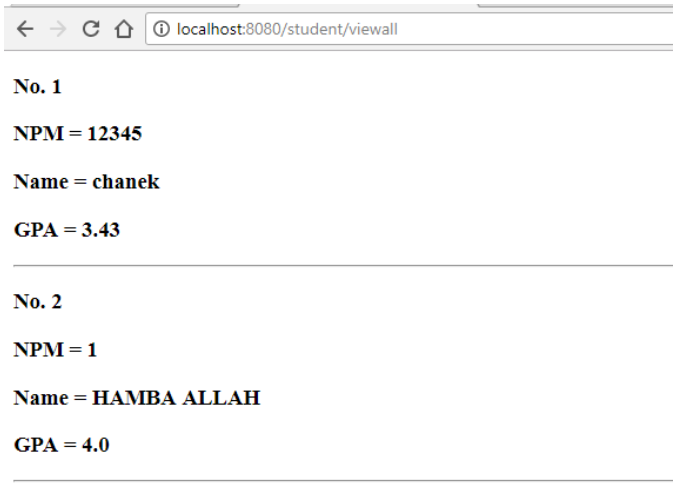
Jawab :

- Ya data student tersebut akan muncul, dikarenakan penambahan ke Array List students sebelumnnya telah berhasil. Jadi ketika kita memanggil Request Mapping viewall, maka data tersebut akan otomatis tampil.

4. Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka localhost:8080/student/viewall,

Pertanyaan 6: Apakah semua data Student muncul?

Jawab :



- Ya semua data student akan muncul, dikarenakan pada method view all, semua data mahasiswa akan diambil dan ditampilkan secara iterasi pada viewall.html.

Penjelasan Method selectStudent

Pada tutorial kali ini saya juga ditugaskan untuk mengimplementasikan method select student pada *class* inMemoryStudentService. Berikut ini kode yang saya buat :

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

Pada method tersebut terlebih dahulu saya melakukan deklarasi tipe data StudentModel yang bernama student. Kemudian saya melakukan iterasi pada studentList untuk mengunjungi masing – masing data pada array List tersebut. Ketika npm pada indek ke-i array List tersebut sama dengan npm pada parameter selectStudent, maka variable student yang telah kita deklarasikan diawal akan menjadi student dengan npm yang sama pada paramater tersebut. Setelah itu return student tersebut. Jika tidak ada akan mengembalikan null.

Penjelasan method view Student dengan menggunakan Path Variable

Pada tutorial kali ini saya juga ditugaskan untuk mengimplementasikan method student dengan menggunakan path variable. Method tersebut berfungsi untuk menampilkan data mahasiswa (NPM,name,dan gpa). Berikut ini kode yang saya buat :

```
@RequestMapping(value = {"/student/view", "student/view/{npm}"})
public String viewPV(@PathVariable (value ="npm", required = false) String npm, Model model) {
    StudentModel student = studentService.selectStudent(npm);
    if(student!=null) {
        model.addAttribute("student",student);
        return "viewPV";
    }else {
        return "salahview";
    }
}
```

Method tersebut hanya menerima Request Mapping "/Student/view" maupun "/Student/view/{npm}". Ketika Request Mapping tersebut sesuai, maka akan terbentuk objek student yang akan mengambil student yang memiliki npm yang sama dengan npm pada Request Mapping tersebut. Kemudian jika terdapat objek student, akan menyimpan student tersebut dan ditampilkannya melalui "viewPV.html". Jika objek student tersebut tidak ada maka akan menampilkan halaman "salahview.html".

Berikut ini implementasi dari “viewPV.html” :

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>View Student by NPM</title>
</head>
<body>
    <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
    <h3 th:text="'Name = ' + ${student.name}">Student name</h3>
    <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
</body>
</html>
```

Akan mencetak NPM, Name, dan GPA dari objek student pada method viewPV.

Berikut ini implementasi dari “salahview.html” :

```
<html>
<head>
<title>ADD</title>
<body>
    <h2>NPM tidak diberikan atau tidak ditemukan </h2>
</body>
</head>
</html>
```

Penjelasan fitur delete pada bagian latihan

Pada tutorial kali ini saya juga ditugaskan untuk mengimplementasikan fitur / method delete pada StudentController.java. Method tersebut berfungsi untuk menghapus data mahasiswa sesuai dengan npm yang diberikan pada Request Mapping. Berikut ini kode yang saya buat :

```
@RequestMapping(value = {"/student/delete", "student/delete/{npm}"})
public String delete(@PathVariable (value ="npm", required = false) String npm, Model model) {
    StudentModel student = studentService.selectStudent(npm);
    if(student!=null) {
        List<StudentModel> students = studentService.selectAllStudents();
        students.remove(student);
        return "delete";
    }else {
        return "salahdelete";
    }
}
```

Method tersebut hanya menerima Request Mapping “/Student/delete” maupun “/Student/delete/{npm}”. Ketika Request Mapping tersebut sesuai, maka akan terbentuk objek student yang akan mengambil student yang memiliki npm yang sama dengan npm pada Request Mapping tersebut. Kemudian jika objek student tersebut ada, maka akan terbentuk array List students yang berisi seluruh student. Setelah itu saya melakukan remove / delete pada array List tersebut kepada objek student yang terbentuk sebelumnya dan akan mengembalikan halaman “delete.html”. Jika objek student tersebut tidak ada maka akan mengembalikan halaman “salahdelete.html”.

Berikut ini implementasi dari “delete.html” :

```
<html>
<head>
<title>Delete</title>
<body>
    <h2>Data berhasil dihapuskan</h2>
</body>
</head>
</html>

```

Berikut ini implementasi dari“salahdelete.html” :

```
<html>
<head>
<title>Delete</title>
<body>
    <h2>NPM kosong atau tidak ditemukan dan proses delete dibatalkan</h2>
</body>
</head>
</html>
```