Mohammad Izzat Raihan
1506689446

Tutorial 3

Method selectStudent

```
@Override
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;
}
```

Karena berada dalam sebuah arraylist, maka kita perlu untuk meloop arraylist tersebut untuk mencari student dengan npm yang sesuai. Setelah didapatkan, maka akan return student dengan index kesekian tersebut.

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

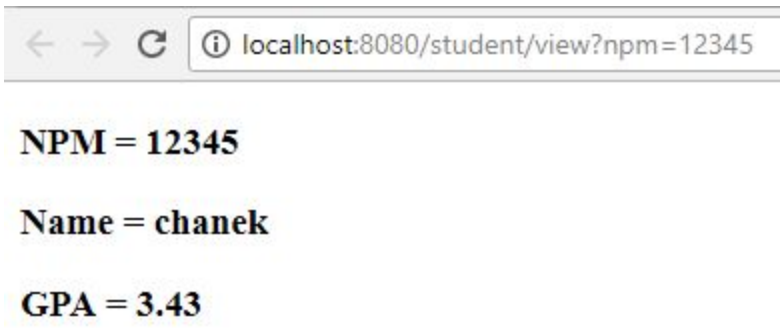
Tidak terjadi error

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

Mohammad Izzat Raihan
1506689446

Terjadi error dengan type Bad Request, yaitu error karena tidak ada gpa yang ditambahkan. Pada method add, diperlukan data gpa yang harus diinput.

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



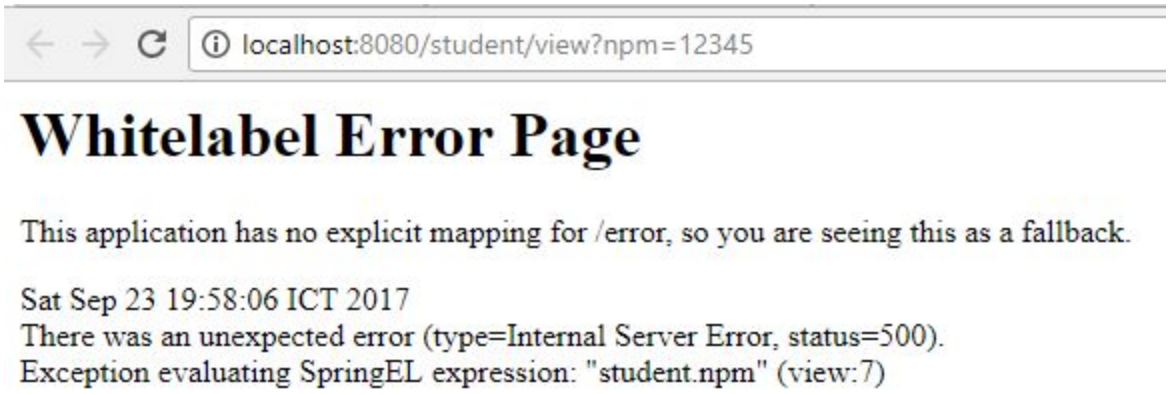
Pertanyaan 3: apakah data Student tersebut muncul? Jika tidak, mengapa?

Iya, muncul. Hal ini dikarenakan pada tahap sebelumnya sudah ditambahkan student dengan npm 12345, name chanek dan GPA 3.43

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

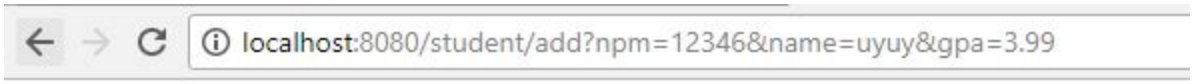
Pertanyaan 4: apakah data Student tersebut muncul? Jika tidak, mengapa?

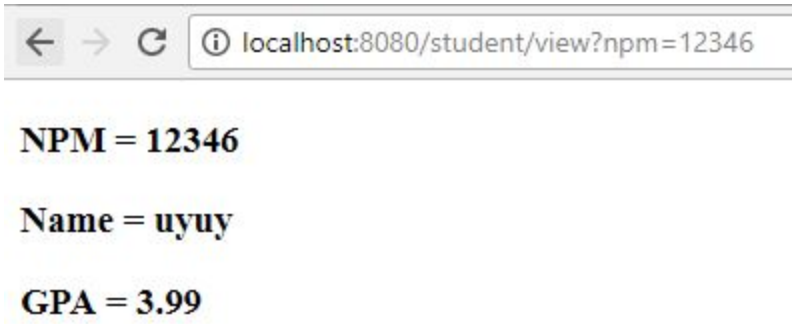
Mohammad Izzat Raihan
1506689446



Tidak muncul, hal ini dikarenakan program tidak menemukan student dengan npm 12345.

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





Data mahasiswa dengan NPM lain tetap ditemukan.

3. Jalankan program dan buka

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

Mohammad Izzat Raihan
1506689446

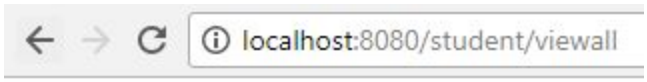
Pertanyaan 5: apakah data Student tersebut muncul?

Iya, data student itu muncul karena telah ditambahkan pada perintah pertama

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

Pertanyaan 6: Apakah semua data Student muncul?

Mohammad Izzat Raihan
1506689446



No. = 1

NPM = 12345

Name = chanek

GPA = 3.43

No. = 2

NPM = 12346

Name = uyuy

GPA = 3.99

Ya, semua data muncul pada halaman viewall

Mohammad Izzat Raihan
1506689446

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.

Tambahkan kode berikut pada StudentController.java

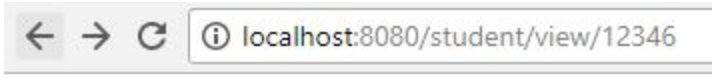
```
@RequestMapping(value = {"/student/view", "/student/view/{npm}"})
public String view(@PathVariable Optional<String> npm, Model model) {
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if (student != null) {
            model.addAttribute("student",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

    } else {
        return "error";
    }
}
```

Lalu buat view error.html dengan code seperti berikut

```
<!DOCTYPE html>
<html>
<head>
<title>ERROR!</title>
</head>
<body>
    <h3>ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!</h3>
</body>
</html>
```

Ketika dijalankan, maka akan seperti berikut

Mohammad Izzat Raihan
1506689446

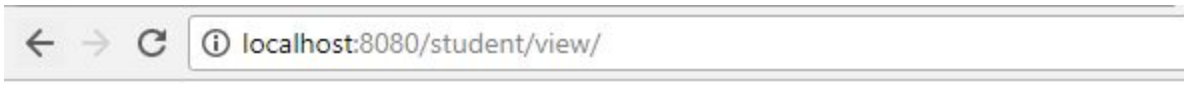


NPM = 12346

Name = uyuy

GPA = 3.99

ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!



ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!

Data student berhasil ditambahkan dengan npm 12346, name uyuy, dan GPA 3.99. Kita dapat melihat hasilnya dengan cara buka halaman localhost:8080/student/view/12346. Setelah dibuka dapat dilihat bahwa data ditemukan. Lalu, ketika membuka halaman localhost:8080/student/view/12347 (tidak ada data dengan npm 12347), akan membuka halaman error. Terakhir, apabila membuka halaman localhost:8080/student/view/ maka akan menampilkan pesan 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.

Mohammad Izzat Raihan
1506689446

Tambahkan kode berikut pada StudentController.java

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

    } else {
        return "error";
    }
}
```

Lalu juga tambahkan kode berikut pada InMemoryStudentService.java

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

Dan tambahkan pula pada StudentService.java

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

Saya menambahkan deleteStudent pada interface agar dapat diimplementasi pada class InMemoryStudentService.java. Pada class tersebut, saya jadi dapat meremove tepat student tersebut bukan lagi index ke sekian.

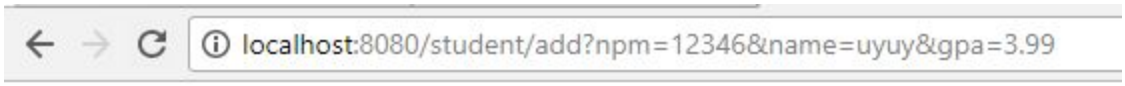
Controller akan menjalankan method deleteStudent tersebut setelah dicari menggunakan method selectStudent yang telah diimplementasi sebelumnya. Ketika menemukan student yang tepat (npmnya sesuai), maka akan diremove. Kalau tidak, maka akan menampilkan pesan error.

Tidak lupa untuk menambahkan view pada delete.html dengan kode seperti berikut

Mohammad Izzat Raihan
1506689446

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>View Student by NPM</title>
</head>
<body>
    <h3>Data berikut telah dihapus</h3>
    <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>
```

Maka akan menghasilkan seperti berikut ini

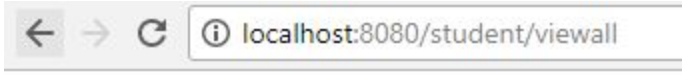


Data berhasil ditambahkan

NPM = 12346

Name = uyuy

GPA = 3.99



No. = 1

NPM = 12346

Name = uyuy

GPA = 3.99

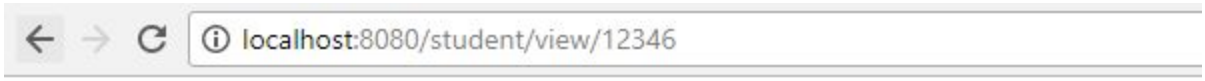
Mohammad Izzat Raihan
1506689446

Data berikut telah dihapus

NPM = 12346

Name = uyuy

GPA = 3.99



ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!

ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!

ERROOOOORRRRR. NPM-nya tidak terisi atau tidak ditemukan!!!!!

Dapat dilihat bahwa data tersebut berhasil dihapus. Dan ketika kita ingin mendelete dengan npm yang sama lagi (data tidak ditemukan) maka akan menghasilkan pesan error dari error.html. Begitu pula jika tidak diisikan npm.

Ringkasan pembelajaran

Pada tutorial 3 kali ini, saya belajar untuk membuat model dengan konsep MVC (Model, View, Controller) dalam project Spring Boot seperti minggu lalu. Ada hal yang berbeda pada tutorial kali ini, yaitu dijalankannya service dengan fungsi create dan read data melalui konsep MVC. Seperti yang dijalankan di atas, terdapat package Service yang berisi interface StudentService.java dan juga class InMemoryStudentService.java.

Dimana interface tersebut terdapat service berupa select, add, dan delete yang akan dijalankan pada class lainnya. Class InMemoryStudentService akan membuat list of student dalam arraylist. Class ini juga memiliki method yang dapat men-select student yang disimpan dalam sebuah list (arraylist). Setelah itu juga terdapat method untuk memilih seluruh student, menambahkan, dan menghapus.

Adanya service tersebut memungkinkan user untuk menambahkan data selama program masih berjalan (ketika program dihentikan, akan menghapus data yang ada). Hal ini akan membantu ketika kita sudah harus memanipulasi data yang berada dalam database.