Bintang Glenn J

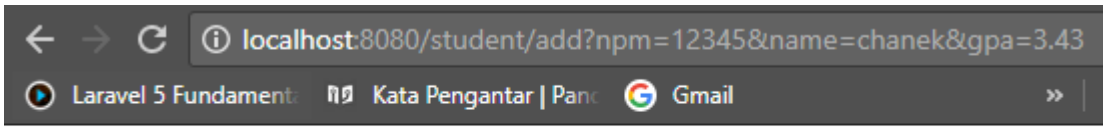
1506757535

Ringkasan Materi

Pada tutorial kali ini saya belajar lebih jauh mengenai *model* dan *service*. Melalui kelas StudentModel, saya menjadi tahu bahwa *model* merupakan representasi sebuah objek yang memiliki atribut-atribut di dalamnya. Konsep ini tidak jauh berbeda dengan konsep objek yang sudah saya ketahui pada *object orienter programming*. Sementara itu, *service* merupakan penghubung antara *database* dan *controller*. Hal-hal seperti operasi penambahan data, pengambilan data, pembaharuan data dan penghapusan data dilakukan pada *service*.

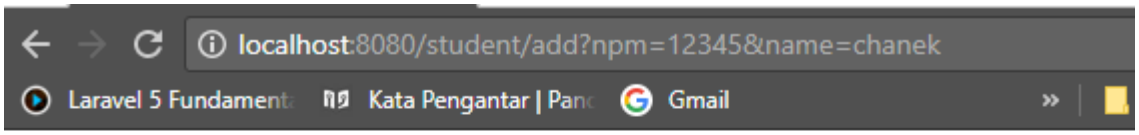
Jawaban Pertanyaan

1. Hasil dari mengakses localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 adalah sebagai berikut:



Data berhasil ditambahkan

2. Hasil dari mengakses localhost:8080/student/add?npm=12345&name=chanek adalah sebagai berikut:



Whitelabel Error Page

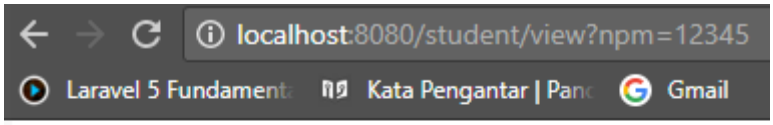
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Sep 22 19:27:48 ICT 2017
There was an unexpected error (type=Bad Request, status=400).
Required double parameter 'gpa' is not present

Error yang terjadi disebabkan oleh tidak adanya parameter gpa yang nilainya diperlukan

```
@RequestParam(value = "npm", required = true)
```

3. Hasil dari mengakses localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 kemudian localhost:8080/student/view?npm=12345 adalah sebagai berikut:

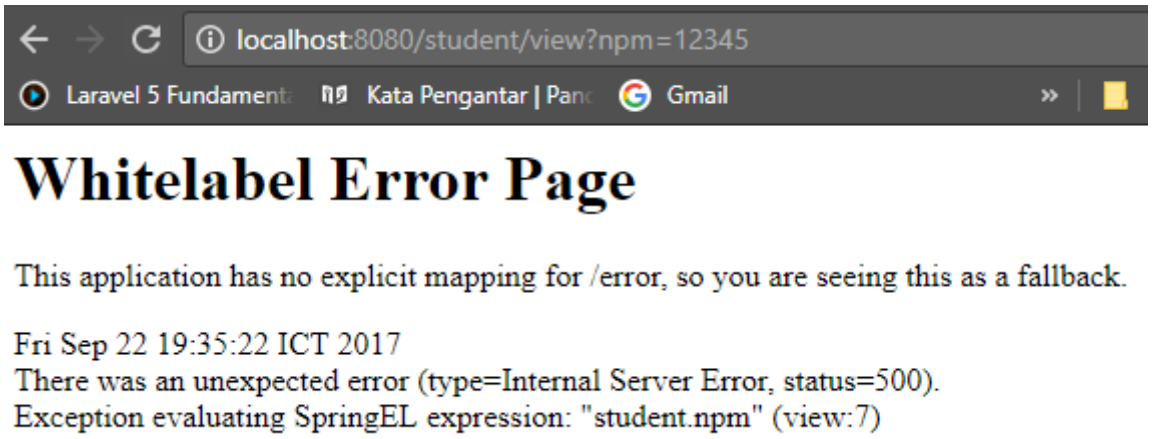


NPM = 12345

Name = chanek

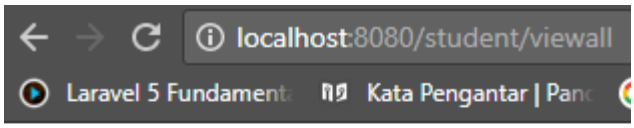
GPA = 3.43

4. Hasil dari mengakses localhost:8080/student/view?npm=12345 setelah mematikan program kemudian menjalankannya kembali adalah sebagai berikut:



Error yang terjadi disebabkan karena npm tersebut belum ada. Seperti yang kita ketahui bersama bahwa pada tutorial kali ini kita menggunakan static list dan bukan database sehingga data tidak tersimpan ketika program dimatikan.

5. Hasil dari mengakses localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 kemudian localhost:8080/student/viewall adalah sebagai berikut:



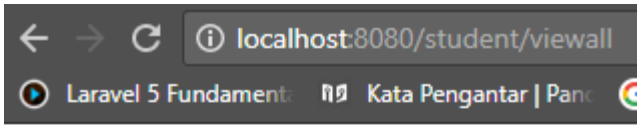
No. 1

NPM = 12345

Name = chanek

GPA = 3.43

6. Hasil dari mengakses localhost:8080/student/viewall setelah menambahkan *student* lain adalah sebagai berikut:



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 67890

Name = nameofstudent

GPA = 3.21

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;
}
```

Method selectStudent yang saya implementasikan hanya sekedar melakukan iterasi di dalam list studentList. Saya hanya mengecek satu per satu apakah terdapat NPM yang sesuai atau tidak. Jika ada, maka akan langsung mengembalikan objek tersebut, namun jika tidak ada sampai akhir iterasi akan mengembalikan null.

Fitur delete

Pertama-tama saya menambahkan baris berikut pada *interface* agar lebih tertata:

```
boolean deleteStudent(String npm);
```

Kemudian saya mengimplementasikan method berikut pada *service* dengan memanfaatkan method selectStudent yang sudah saya buat dan method *remove* dari list.

```java
@Override
public boolean deleteStudent(String npm) {
    return studentList.remove(selectStudent(npm));
}
```

Setelahnya, saya menambahkan method deletePath pada *controller*. Dengan memanfaatkan *container* Optional dan status berupa boolean dari method deleteStudent, saya dapat memisahkan method ini menjadi 3 skenario, yakni NPM kosong, NPM tidak ada dan berhasil.

```java
@RequestMapping(value = {"student/delete", "/student/delete/{npm}"})
public String deletePath(Model model, @PathVariable Optional<String> npm) {
    if (npm.isPresent()) {
        boolean status = studentService.deleteStudent(npm.get());
        if(!status) {
            model.addAttribute("err", "NPM " + npm.get() + " tidak ditemukan");
            model.addAttribute("message", "Proses delete dibatalkan");
            return "error";
        }
        model.addAttribute("npm", npm.get());
        return "delete";
    }
    model.addAttribute("err", "NPM kosong");
    model.addAttribute("message", "Proses delete dibatalkan");
    return "error";
}
```

Saya kemudian menambahkan halaman delete.html pada resources/templates. Halaman yang ditampilkan akan sesuai dengan ekspresi berikut:

```html
<body>
    <h3 th:text="'Student dengan NPM = ' + ${npm} + ' berhasil dihapus'">Delete Student</h3>
</body>
```

Sementara jika *error*, maka halaman *error* yang ditampilkan adalah seperti berikut:

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>Error Page</title>
</head>
<body>
    <h1 th:text="${err}">Error</h1>
    <h2 th:text="${message}"></h2>
</body>
</html>
```