

Write Up Tutorial 3 – Menggunakan Model dan Service pada Project Spring Boot

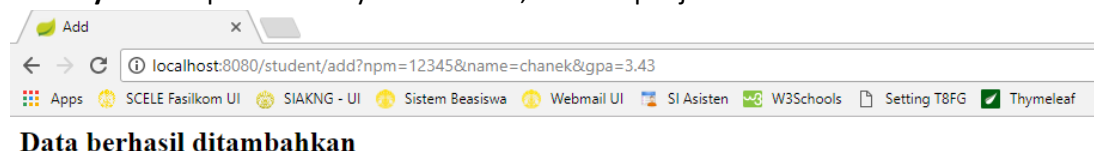
A. Ringkasan

Materi yang telah saya pelajari pada tutorial kali ini antara lain konsep MVC menggunakan spring boot. Konsep MVC sendiri adalah kepanjangan dari konsep model-view-controller. Model yang dimaksud pada tutorial kali ini adalah class `StudentModel.java`. Untuk controller, terdapat class `StudentController.java`, sedangkan untuk view ada beberapa file html bergantung kepada output yang diinginkan. Selain itu ada juga service yang berfungsi untuk berkoneksi ke database pada tutorial selanjutnya. Untuk service sendiri, terdapat interface `StudentService` yang diimplement oleh class `InMemoryStudentService`. Melalui tutorial kali ini, saya dapat lebih mengerti bagaimana konsep MVC digunakan secara baik dan benar.

B. Tutorial

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

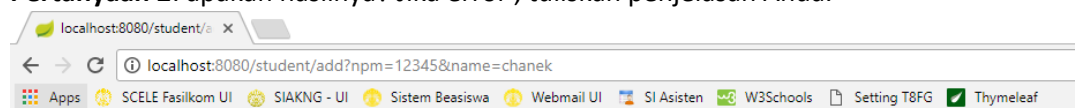
Pertanyaan 1 : apakah hasilnya? Jika error , tuliskan penjelasan Anda.



Tidak muncul error, dan muncul tampilan bahwa data berhasil ditambahkan.

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

Pertanyaan 2: apakah hasilnya? Jika error , tuliskan penjelasan Anda.



Whitelabel Error Page

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

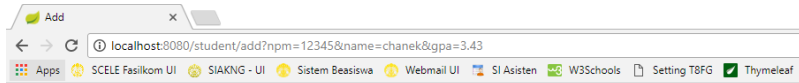
Sat Sep 23 18:03:12 ICT 2017

There was an unexpected error (type=Bad Request, status=400).
Required double parameter 'gpa' is not present

Error tersebut muncul karena parameter gpa telah diset `required = true` sehingga string query gpa harus ada jika ingin menggunakan method add.

3. 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?



Data berhasil ditambahkan



NPM = 12345

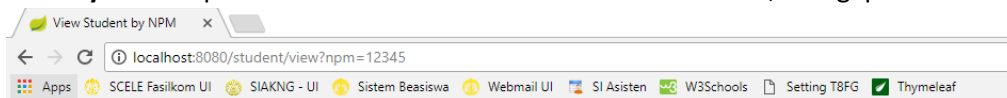
Name = chanek

GPA = 3.43

Data Student telah muncul tanpa adanya error

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?



Whitelabel Error Page

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

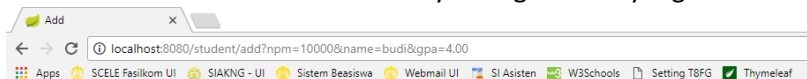
Sat Sep 23 20:45:29 ICT 2017

There was an unexpected error (type=Internal Server Error, status=500).

Exception evaluating SpringEL expression: "student.npm" (view:7)

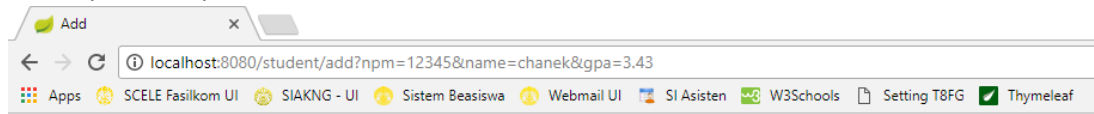
Data Student tidak muncul karena isi dari studentList sudah tidak ada lagi jika program telah dimatikan.

Coba tambahkan data Student lainnya dengan NPM yang berbeda.

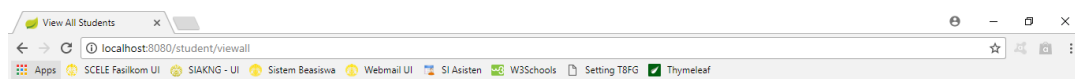


Data berhasil ditambahkan

5. localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43 lalu buka localhost:8080/student/viewall,
Pertanyaan 5 : apakah data Student tersebut muncul?



Data berhasil ditambahkan



No. 1

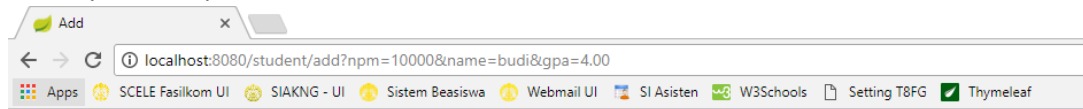
NPM = 12345

Name = chanek

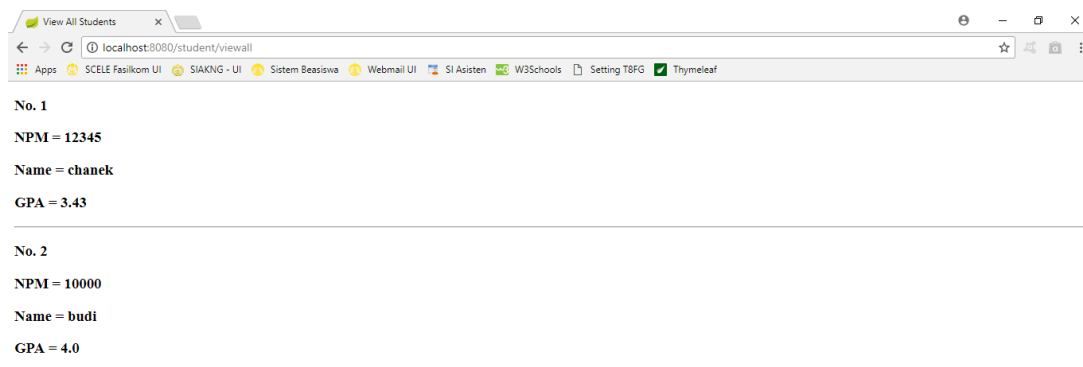
GPA = 3,43

Data Student berhasil muncul.

6. Coba tambahkan data Student lainnya dengan NPM yang berbeda, lalu buka localhost:8080/student/viewall,
Pertanyaan 6 : Apakah semua data Student muncul?



Data berhasil ditambahkan



Data Student berhasil muncul

C. Implementasi method selectStudent.

```
@Override
public StudentModel selectStudent(String npm) {
    // TODO Auto-generated method stub
    for(int i = 0; i < studentList.size(); i++) {
        if(npm.equals(studentList.get(i).getNpm())){
            return studentList.get(i);
        }
    }
    return null;
}
```

D. Penjelasan fitur Delete.

Fitur delete yang saya buat pada latihan menggunakan method delete pada StudentController.java . Berikut adalah methodnya.

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

Pada method ini mirip dengan jawaban latihan nomor 1 yaitu saya menggunakan Path Variable agar dapat diakses dengan format seperti **localhost:8080/student/delete/14769**. Proses nya juga mirip dengan latihan nomer 1 yaitu pertama-tama, method ini mengecek apakah string query nya ada atau tidak. Jika tidak ada, akan mengembalikan halaman error. Sedangkan jika ada akan dicek lagi apakah npm tersebut ada di studentList atau tidak. Jika tidak, akan mengembalikan halaman error. Sedangkan jika terdapat di studentList maka akan menggunakan method deleteStudent yang ada di class InMemoryStudentService yang mengimplement interface StudentService.

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

Proses method ini mirip dengan method selectStudent namun bedanya ia akan menghapus object Student di StudentList namun akan mengembalikan object Student ke pemanggilnya. Setelah itu akan ditampilkan informasi Data Student yang dimaksud oleh view delete.