

Tutorial 3

Nama : Sofyan Hadi

NPM : 1706107011

- a. Ringkasan dari materi.

Pada tutorial kali ini mempelajari mengenai MVC. Model untuk menyimpan data atau mengambil data, di tutorial ini menggunakan List. View untuk menampilkan data yang akan di tampilkan berdasarkan method yang diimplementasikan di Controller. Controller berisi method serta mengatur proses data untuk ditampilkan di view.

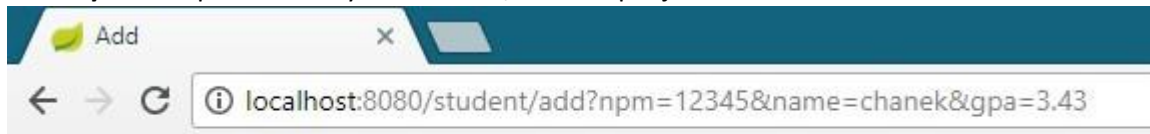
- b. Hasil jawaban dari setiap poin pada bagian tutorial (dapat didukung dengan screenshot)

Membuat Controller dan Fungsi Add

Jalankan program dan buka

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

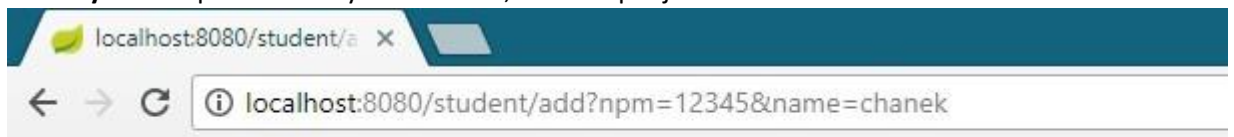
Pertanyaan 1: apakah hasilnya? Jika error, tuliskan penjelasan Anda.



Data berhasil ditambahkan

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.

Fri Mar 02 16:13:13 ICT 2018

There was an unexpected error (type=Bad Request, status=400).

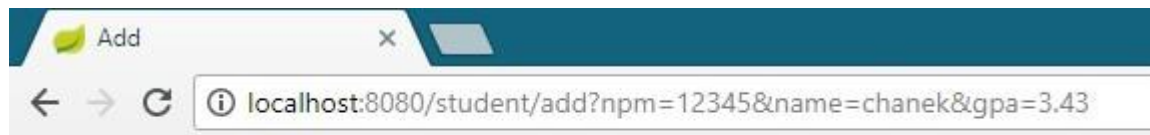
Required double parameter 'gpa' is not present

Error, karena tidak adanya parameter gpa.

Method View by NPM

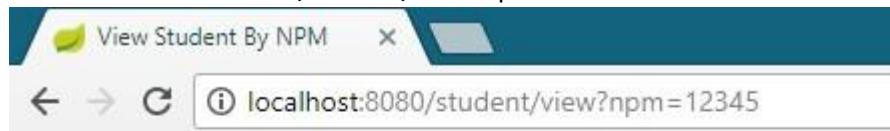
Jalankan program dan buka

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



Data berhasil ditambahkan

lalu buka localhost:8080/student/view?npm=12345



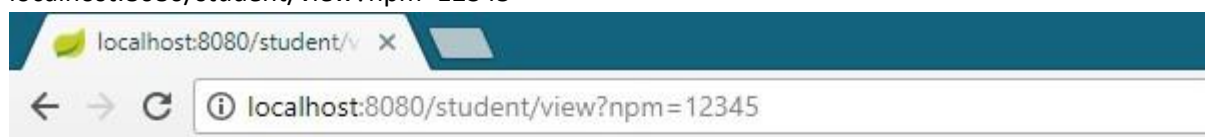
NPM : 12345

Name : chanek

GPA : 3.43

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

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



Whitelabel Error Page

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

Fri Mar 02 17:04:28 ICT 2018

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

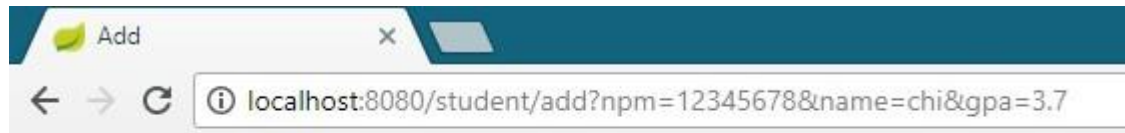
Exception evaluating SpringEL expression: "student.npm" (view:7)

Pertanyaan 4: apakah data Student tersebut muncul? Jika tidak, mengapa? Tidak, karena data dilist akan terhapus ketika program dihentikan.

Coba tambahkan data Student lainnya dengan NPM yang berbeda.



Data berhasil ditambahkan

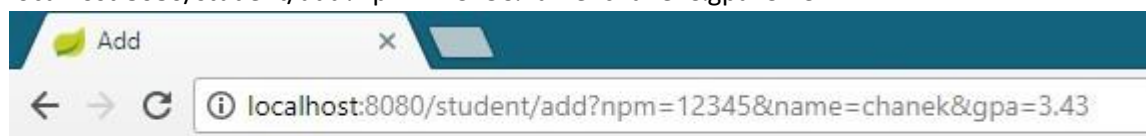


Data berhasil ditambahkan

Method View All

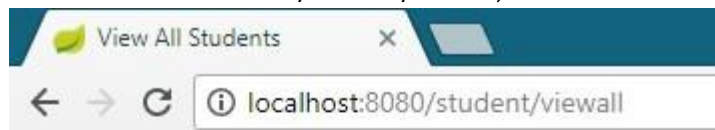
Jalankan program dan buka

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



Data berhasil ditambahkan

lalu buka localhost:8080/student/viewall,



No : 1

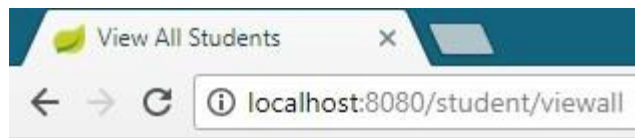
NPM : 12345

Name : chanek

GPA : 3.43

Pertanyaan 5: apakah data Student tersebut muncul? Ya muncul

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



No : 1

NPM : 12345

Name : chanek

GPA : 3.43

No : 2

NPM : 12345

Name : chanek

GPA : 3.43

No : 3

NPM : 123456

Name : chal

GPA : 3.5

No : 4

NPM : 1234567

Name : chak

GPA : 3.6

Pertanyaan 6: Apakah semua data Student muncul? Ya muncul

Latihan

1. Menambahkan method view dengan path variable pada StudentController

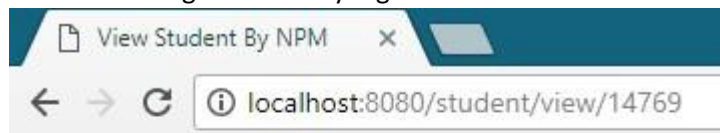
```
@RequestMapping(value = {"student/view", "student/view/{npm}"})
public String viewPath (@PathVariable Optional<String> npm, Model model)
{
    if (npm.isPresent()) {
        StudentModel student = studentService.selectStudent(npm.get());
        if(student != null) {
            model.addAttribute("student",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

    } else {
        model.addAttribute("npm", "Kosong");
        return "error_nf";
    }
    return "view";
}
```

Lalu menambahkan file error_nf.html untuk menampilkan hasil dari method tersebut

```
<!DOCTYPE html>
<html>
<head>
<title>Error Not Found</title>
</head>
<body>
<h3 th:text="'Error NIM '+${npm}+' tidak ditemukan'">Error</h3>
</body>
</html>
```

Mencoba mengakses NPM yang terdaftar.

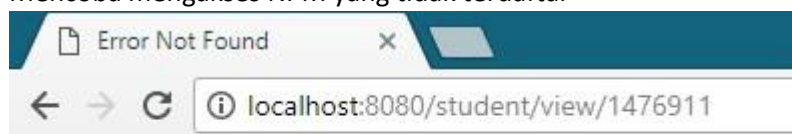


NPM : 14769

Name : Clay

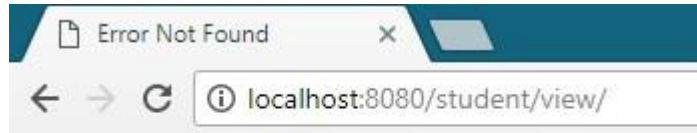
GPA : 3.22

Mencoba mengakses NPM yang tidak terdaftar



Error NIM 1476911 tidak ditemukan

Mencoba mengakses tanpa memasukan NPM



Error NIM Kosong tidak ditemukan

2. Menambahkan fitur delete untuk menghapus berdasarkan NPM pertama adalah menambahkan method pada interface

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

dan mengimplement method tersebut pada class inMemoryStudentService

```
@Override
public void deleteStudent(StudentModel student) {
    // TODO Auto-generated method stub
    studentList.remove(student);
}
```

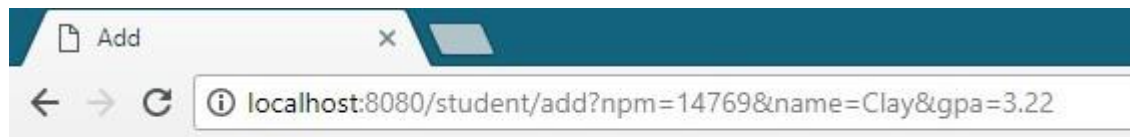
Menambahkan method delete pada StudentController

```
@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) {
            studentService.deleteStudent(student);
            model.addAttribute("msg", "Data "+npm.get()+" berhasil dihapus");
        } else {
            model.addAttribute("msg", "Data tidak ada");
            return "delete";
        }
    } else {
        model.addAttribute("msg", "Data NPM kosong");
        return "delete";
    }
    return "delete";
}
```

Membuat file baru delete.html untuk melihat hasil dari method delete

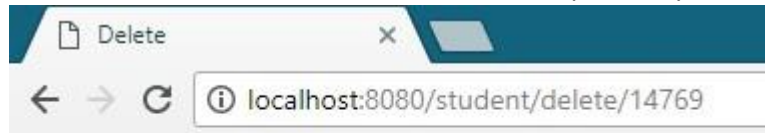
```
<!DOCTYPE HTML>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Delete</title>
</head>
<body>
<h3 th:text="${msg}">msg</h3>
</body>
</html>
```


Menambahkan data mahasiswa baru



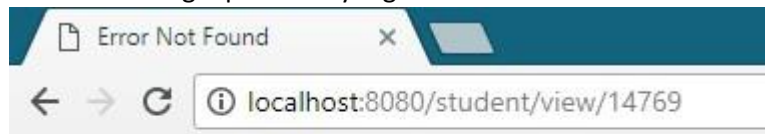
Data berhasil ditambahkan

Lalu mencoba untuk melakukan delete terhadap NPM nya



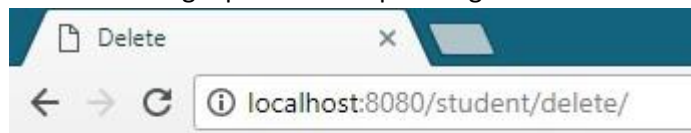
Data 14769 berhasil dihapus

Mencoba menghapus NPM yang tidak terdaftar



Error NIM 14769 tidak ditemukan

Mencoba menghapus NPM tanpa mengisi NPM



Data NPM kosong

- c. Method selectStudent yang diimplementasikan

```
@Override
public StudentModel selectStudent(String npm) {
    StudentModel student=null;
    for (StudentModel st:studentList ) {
        if (st.getNpm().equals(npm)) {
            return student=st;
        }
    }
    return null;
}
```

Method selectStudent digunakan untuk mencari data student berdasarkan NIM yang dicari dari keseluruhan data student.

d. Penjelasan fitur delete.

```
@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) {
            studentService.deleteStudent(student);
            model.addAttribute("msg", "Data "+npm.get()+" berhasil dihapus");

        } else {
            model.addAttribute("msg", "Data tidak ada");
            return "delete";
        }
    } else {
        model.addAttribute("msg","Data NPM kosong");
        return "delete";
    }
    return "delete";
}
```

Method ini digunakan untuk menghapus data berdasarkan NPM yang dimasukan. Ketika NPM terdaftar dalam data(List) maka akan terhapus, jika memasukan NPM yang tidak terdapat dalam data maka akan menampilkan "Data tidak ada" dan jika tidak memasukan NPM maka akan menampilkan "Data NPM kosong".