

Nama : Rinjani Samosir

NPM : 1606954994

Kelas : Ekstensi 2016

## Tutorial 3

### Arsitektur dan Pemrograman Aplikasi Perusahaan Semester Genap 2017/2018

#### Menggunakan Model dan Service pada Project Spring Boot

#### TUTORIAL

##### Membuat *Class Model*

Dalam membuat class StudentModel ditambahkan dengan method constructor, setter dan getter.

```
private String npm;
private String name;
private double gpa;

public StudentModel (String npm, String name, double gpa) {
    this.npm = npm;
    this.name = name;
    this.gpa = gpa;
}

public String getNpm() {
    return npm;
}

public void setNpm(String npm) {
    this.npm = npm;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public double getGpa() {
    return gpa;
}

public void setGpa(double gpa) {
    this.gpa = gpa;
}
```

##### Membuat *Service*

```

public class InMemoryStudentService implements StudentService{
    private static List <StudentModel> studentList = new ArrayList<StudentModel>();

    @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;
    }
}

```

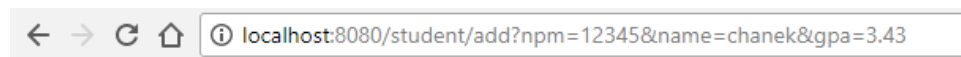
### Membuat *Controller* dan Fungsi *Add*

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.

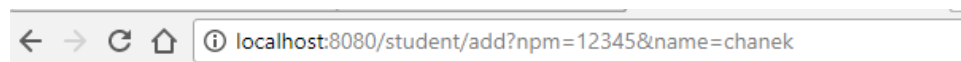


**Data berhasil ditambahkan**

Tidak ada error seperti yang ditampilkan pada screenshot diatas.

**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.

Tue Feb 27 20:55:37 ICT 2018

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

Required double parameter 'gpa' is not present

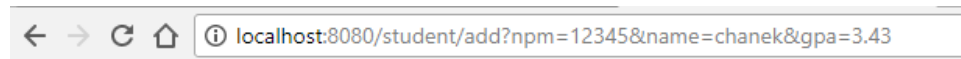
Ya. Ada error. Karena parameter gpa tidak diisi (seharusnya diisi dengan npm, name dan gpa)

### Method View by NPM

3. Jalankan program dan buka localhost:

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

Hasil dari 8080/student/add?npm=12345&name=chanek&gpa=3.43

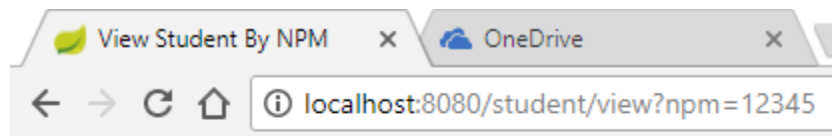


**Data berhasil ditambahkan**

Pertanyaan 3 :

Apakah data Student tersebut muncul? Jika tidak, mengapa?

Ya. Seperti terlihat pada screenshot dibawah ini. Ditampilkan data nama dan GPA sesuai dengan NPM yang dicari.



**NPM = 12345**

**Name = chanek**

**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?

Tidak muncul seperti terlihat pada screenshot dibawah.

← → ↻ 🏠 ⓘ 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 22:40:29 ICT 2018

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

Exception evaluating SpringEL expression: "student.npm" (view:8)

Hal ini karena jika ingin menjalankan view harus melakukan add terlebih dahulu.

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

← → ↻ 🏠 ⓘ localhost:8080/student/add?npm=1606954994&name=Rinjani&gpa=3.90

## Data berhasil ditambahkan

← → ↻ 🏠 ⓘ localhost:8080/student/view?npm=1606954994

**NPM = 1606954994**

**Name = Rinjani**

**GPA = 3.9**

### Method View All

3. Jalankan program dan buka localhost:

**8080/student/add?npm=12345&name=chanek&gpa=3.43** lalu buka

**localhost:8080/student/viewall,**

Pertanyaan 5 : apakah data Student tersebut muncul?

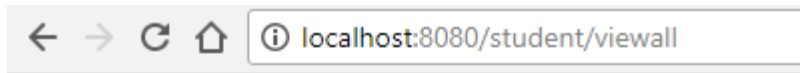
Ya, muncul. Seperti yang terlihat dibawah ini.

← → ↻ 🏠 ⓘ localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43

## Data berhasil ditambahkan

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

Pertanyaan 6 : Apakah semua data Student muncul

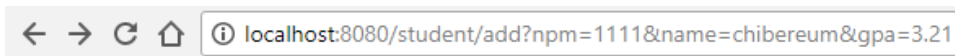


**No. 1**

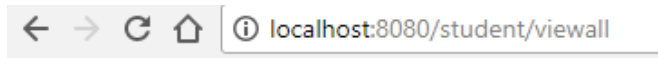
**NPM = 12345**

**Name = chanek**

**GPA = 3.43**



**Data berhasil ditambahkan**



**No. 1**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

**No. 2**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

**No. 3**

**NPM = 1111**

**Name = chibereum**

**GPA = 3.21**

Ya data yang dimasukkan sebelumnya akan ditampilkan semuanya pada viewall seperti yang terlihat pada screenshot diatas.

## **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.

#### **Kode method view**

Ditambahkan 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 "noData";
        }
    }else {
        return "notFindData";
    }
    return "view";
}

```

Ditambahkan file noFindData.html untuk handle error NPM Kosong

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>

<title>Status Data</title>
</head>
<body>
    <h3 th:text="'NPM Kosong'"> Student NPM</h3>
</body>
</html>

```

Ditambahkan file noData.html untuk handle error npm tidak ditemukan

```

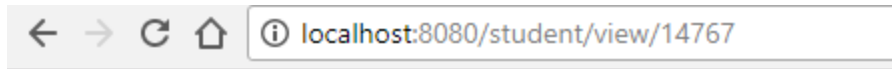
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>

<title>Status Data</title>
</head>
<body>
    <h3 th:text="'Data tidak ditemukan'"> Student NPM</h3>
</body>
</html>

```

← → ↺ ↻ ⓘ localhost:8080/student/add?npm=14767&name=janickun&gpa=3.21

**Data berhasil ditambahkan**



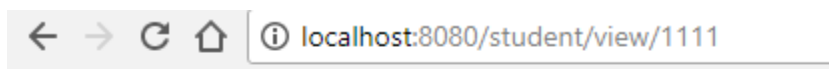
**NPM = 14767**

**Name = janickun**

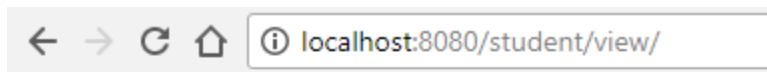
**GPA = 3.21**

Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman error

Yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.



**Data tidak ditemukan**



**NPM Kosong**

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.

#### **Kode method delete**

Ditambahkan pada student controller



```

@RequestMapping (value = {"/student/delete", "/student/delete/{npm}"})
public String deletePath(@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);
        }
        else {
            model.addAttribute("npm", npm.get());
            return "noData";
        }
    }else {
        return "notFindData";
    }
    return "deleteData";
}

```

Ditambahkan pada InMemoryStudent

```

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

```

Ditambahkan pada file deleteData.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>

<title>Status Data</title>
</head>
<body>
    <h3 th:text="'Data berhasil dihapus'"> Student NPM</h3>
</body>
</html>

```

← → ↺ 🏠 ⓘ localhost:8080/student/delete/

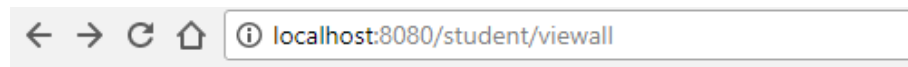
**NPM Kosong**

← → ↺ 🏠 ⓘ localhost:8080/student/delete/2222

**Data tidak ditemukan**

Proses Delete jika berhasil

**Data sebelum dihapus**



**No. 1**

**NPM = 12346**

**Name = chanek**

**GPA = 3.43**

**No. 2**

**NPM = 12345**

**Name = chanek**

**GPA = 3.43**

**No. 3**

**NPM = 12300**

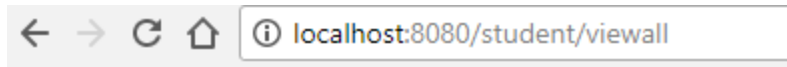
**Name = chanek**

**GPA = 3.43**



**Data berhasil dihapus**

**Data setelah dihapus**



**No. 1**

**NPM = 12346**

**Name = chanek**

**GPA = 3.43**

**No. 2**

**NPM = 12300**

**Name = chanek**

**GPA = 3.43**

Tahapan-tahapan dalam menyelesaikan tutorial 3 adalah sebagai berikut.

Pada tutorial ini dipelajari mengenai model, service dan controller. Service yang dimaksud akan mendefinisikan method-method untuk memanipulasi kelas student. Setelah membuat method-method tersebut, maka data seluruh siswa dapat dilihat berdasarkan NPM. Dalam tutorial ini juga dibuat viewall untuk melihat seluruh data yang sudah ditambahkan sebelumnya. Pembuatan viewall ini menggunakan path variabel sehingga nantinya dapat handle error yang muncul. Selain fitur add dan view dibuat juga akses untuk menghapus data yang sudah dimasukkan dengan membuat method berupa path variabel.