

Tutorial 3 – Arsitektur dan Pemograman Aplikasi Perusahaan

NPM: 1506797242

Menggunakan Model dan Service pada Project Spring Boot

Ringkasan Materi

Pada tutorial 3 ini, saya mempelajari tentang implementasi konsep *model-view-controller* (MVC) menggunakan Spring Boot. Fokus tutorial 3 ini adalah model dan *service*. Model sebagai objek untuk menyimpan informasi dari suatu objek misal model *StudentModel* menyimpan informasi *name*, *npm*, dan *gpa*.

Service merupakan suatu perantara antara *controller* dan *database*. Service layer akan mengolah data yang ada pada *database* dan memanipulasi inputan *user* ke dalam database misalnya menghitung IPK pada model mahasiswa. Setelah itu, controller tinggal memanggil method yang ada pada service dengan menggunakan *@RequestMapping*. Tutorial 3 ini belum menggunakan database, data list dari *StudentModel*.

Method Select Student pada Class *InMemoryStudentService*

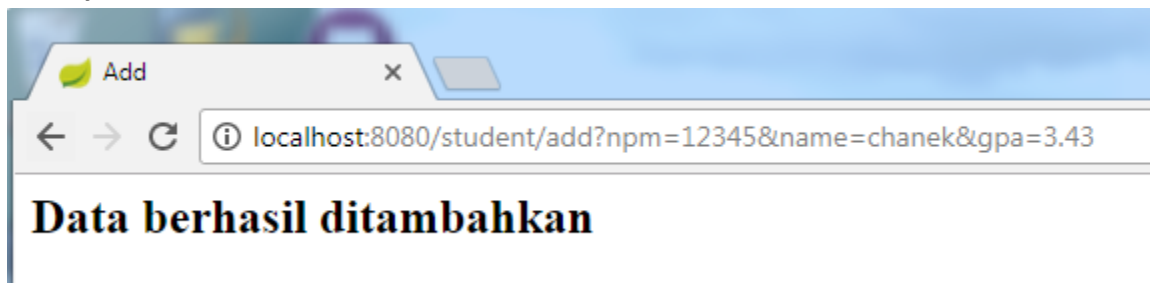
```
public StudentModel selectStudent(String npm) {  
    for (StudentModel student : studentList) {  
        if(student.getNpm().equals(npm)) {  
            return student;  
        }  
    }  
    return null;  
}
```

Jawaban Pertanyaan

Membuat Controller dan Fungsi Add

1. localhost:8080/student/add?npm=12345&name=chanek&gpa=3,43

Hasilnya :

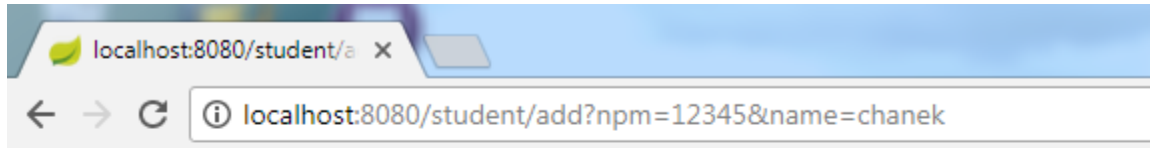


s

Method add akan menerima parameter name, npm dan gpa yang ketiganya merupakan required dengan menggunakan *request* method GET. Method add akan membuat model *StudentModel* dan menambahkan objek tersebut dengan memanggil method *addStudent*.

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

Hasilnya :



Whitelabel Error Page

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

Sat Mar 03 15:27:25 ICT 2018

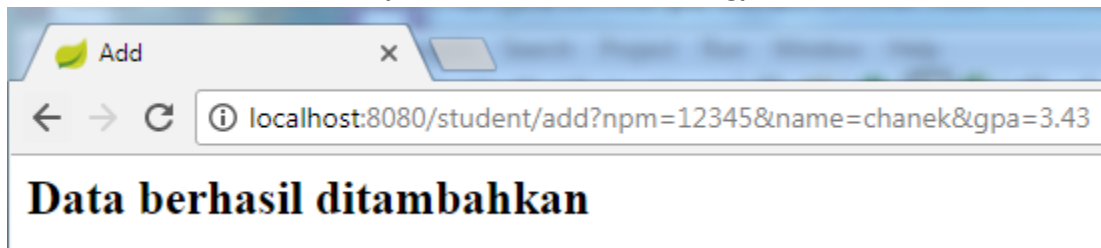
There was an unexpected error (type=Bad Request, status=400).

Required double parameter 'gpa' is not present

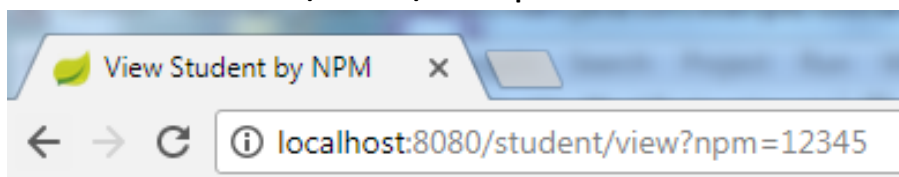
Muncul eror karena *request* parameter tidak sama seharusnya parameter gpa harus ada karena bersifat *required*. Hal tersebut membuat muncul eror dengan HTTP Code 400.

Method View by NPM

3. Localhost:8080/student/add?npm=12345&name=chanek&gpa=3,43



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



NPM = 12345

Name = chanek

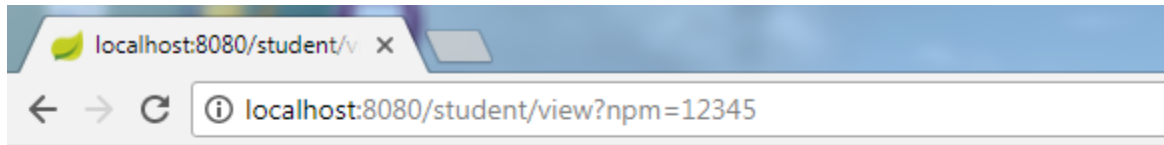
GPA = 3.43

Hasilnya : data student muncul berdasarkan parameter npm yang dimasukkan.

4. **localhost:8080/student/view?npm=12345**

Apakah data Student muncul? Jika tidak, mengapa?

Hasilnya :



Whitelabel Error Page

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

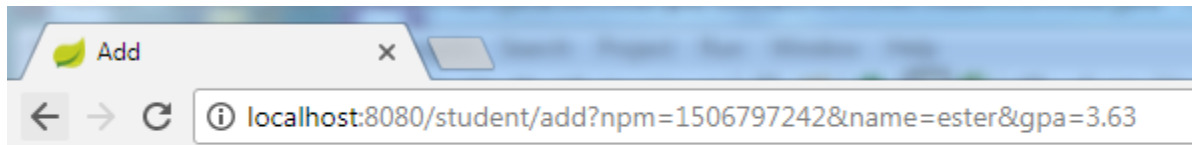
Sat Mar 03 15:53:54 ICT 2018

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

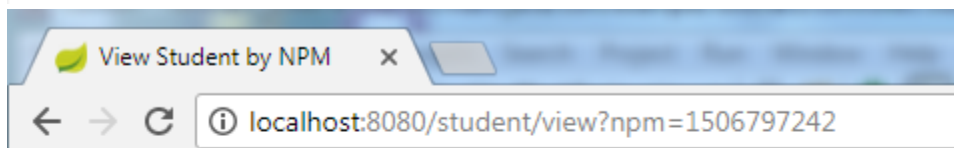
Exception evaluating SpringEL expression: "student.npm" (view:7)

Muncul eror Internal Server Error(500) karena datanya tidak ada dalam List objek Student.

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



Data berhasil ditambahkan



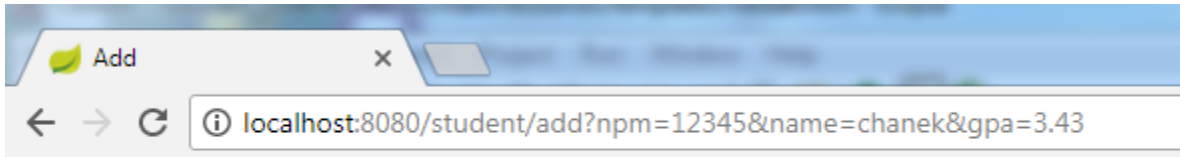
NPM = 1506797242

Name = ester

GPA = 3.63

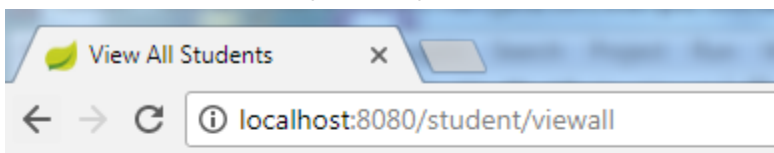
Method View All

6. 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

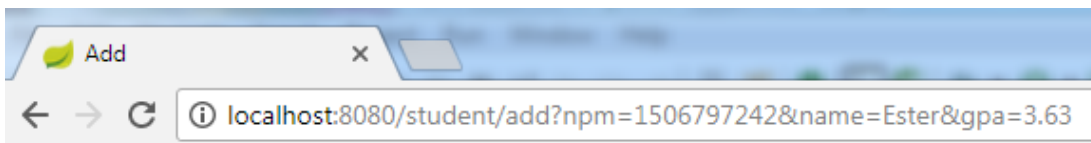
Apakah data Student tersebut muncul?

Hasilnya : Data student muncul dengan nama chanek

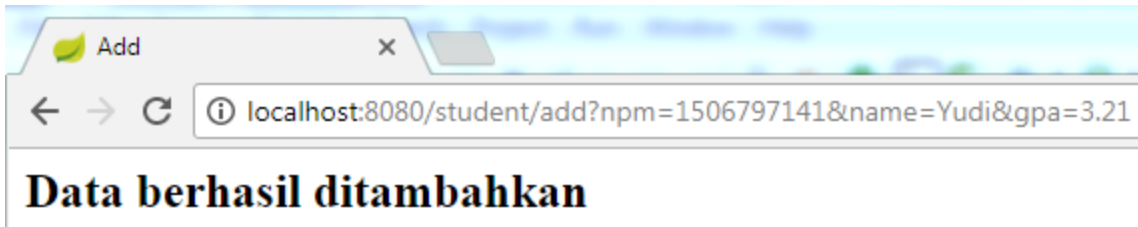
7. Coba tambahkan data Student lainnya dengan NPM yang berbeda,

Apakah semua data Student muncul?

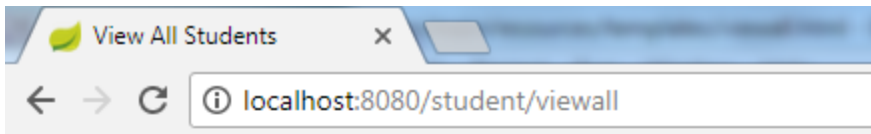
Hasilnya : semua data student yang dimasukkan muncul pada halaman.



Data berhasil ditambahkan



lalu buka `Localhost:8080/student/viewall`



No. 1

NPM = 12345

Name = chanek

GPA = 3.43

No. 2

NPM = 1506797242

Name = Ester

GPA = 3.63

No. 3

NPM = 1506797141

Name = Yudi

GPA = 3.21

Latihan

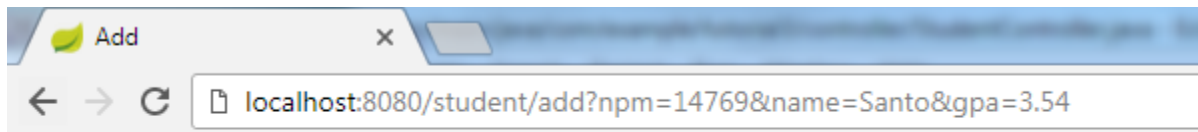
1. Pada **StudentController** tambahkan sebuah method **view Student** dengan menggunakan **Path Variable** misalnya, Anda ingin memasukkan data seorang Student yang memiliki NPM 14769, untuk melihat data yang baru dimasukkan tersebut dapat mengakses halaman **localhost:8080/student/view/14769**.

Langkah-langkah menambahkan method view Student :

Menambahkan method **viewStudent** dengan menggunakan **Path Variable** dan method **viewStudent** menerima parameter **NPM** dan tanpa parameter.

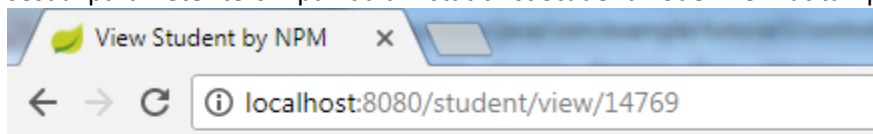
```
@RequestMapping(value = {"/student/view/", "student/view/{npm}"})
public String viewStudent (@PathVariable Optional<String> npm, Model model) {
    if(!npm.isPresent()) {
        model.addAttribute("error", "Tambahkan npm pada request URL");
    } else {
        StudentModel student = studentService.selectStudent(npm.get());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

Untuk menampilkan data mahasiswa terlebih dahulu tambahkan mahasiswa dengan NPM 14769. Dapat dilihat seperti gambar berikut ini :



Data berhasil ditambahkan

Setelah itu, NPM yang telah dibuat pada object **StudentModel** dan ditemukan dalam list kemudian memanggil **selectStudent** pada service **inMemoryStudent** dengan mencari objek student dengan npm sesuai parameter tersimpan dalam static list **StudentModel**. Berikut tampilannya :



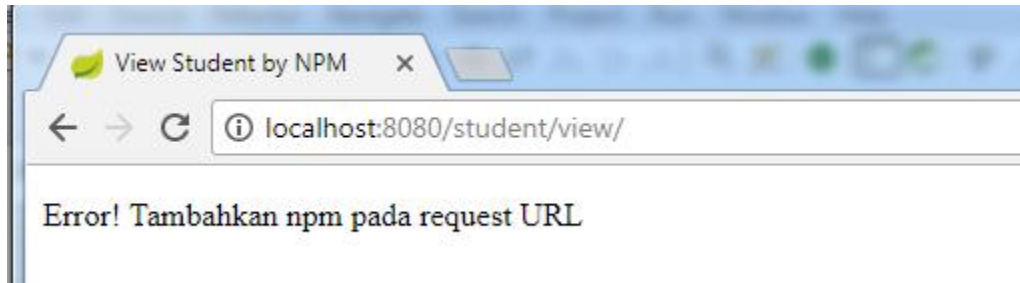
NPM = 14769

Name = Santo

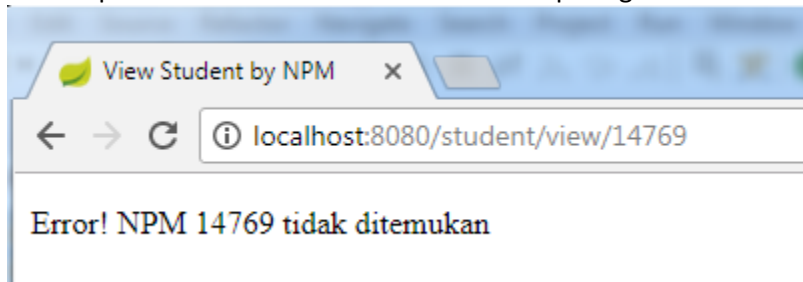
GPA = 3.54

Jika nomor NPM tidak diberikan atau tidak ditemukan kembalikan halaman *error* yang berisi informasi bahwa nomor NPM kosong atau tidak ditemukan.

Jika parameter NPM tidak diset dalam request URL maka akan menampilkan halaman sebagai berikut :



Jika parameter NPM tidak ditemukan maka akan mengakses ke error pada view.html dan menampilkan bahwa NPM tidak ditemukan seperti gambar berikut ini :



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.

Menambahkan method *deleteStudent* untuk menghapus objek student pada *listStudent* dan method *deleteStudent* menerima parameter NPM dan tanpa parameter.

- Tambahkan method *deleteStudent* pada *InMemoryStudent*

```
public boolean deleteStudent(String npm) {  
    boolean isFound = false;  
    if(this.selectStudent(npm) != null) {  
        ListIterator<StudentModel> iter = studentList.listIterator();  
        while(iter.hasNext()){  
            if(iter.next().getNpm().equals(npm)){  
                iter.remove();  
                isFound = true;  
                break;  
            }  
        }  
    }  
}
```

- Setelah itu tambahkan method deleteStudent pada interface StudentService

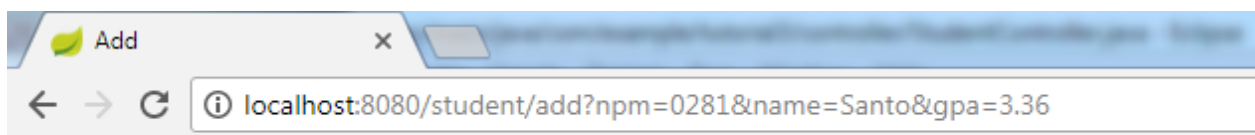
```
public interface StudentService {
    StudentModel selectStudent(String npm);
    List<StudentModel> selectAllStudent();
    void addStudent (StudentModel student);
    boolean deleteStudent(String npm);
}
```

- Tambahkan delete.html pada template\

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <meta charset="UTF-8"/>
5 <title>Delete</title>
6 </head>
7 <body>
8 <p th:text="${read}">Message</p>
9 </body>
10 </html>
```

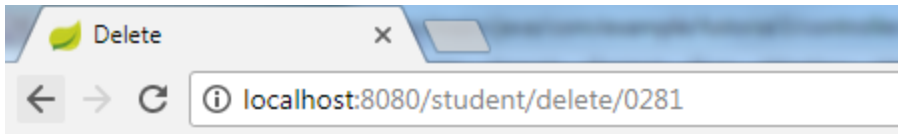
```
@RequestMapping(value = {"/student/delete", "student/delete/{npm}"})
public String deleteStudent(@PathVariable Optional<String> npm, Model model) {
    if(!npm.isPresent()) {
        model.addAttribute("read", "Masukkan npm Anda");
    } else {
        Boolean isDeleted = studentService.deleteStudent(npm.get());
        if(isDeleted) {
            model.addAttribute("read", "Berhasil menghapus mahasiswa dengan npm "+npm.get());
        } else {
            model.addAttribute("read", "Gagal menghapus mahasiswa karena npm "+npm.get()+" tidak ditemukan");
        }
    }
    return "delete";
}
```

Pertama, tambahkan objek student pada list objek



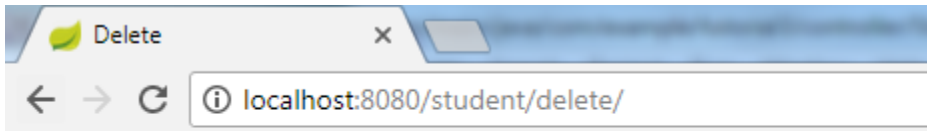
Data berhasil ditambahkan

Jika ditemukan data objek student maka akan dihapus dari list dan jika tidak ditemukan maka akan return false. Pada StudentController akan dicek jika return pada deleteStudent adalah true maka akan diakses read dan delete berhasil.



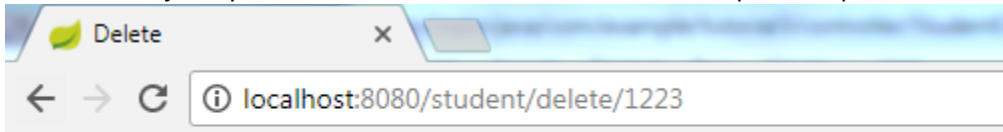
Berhasil menghapus mahasiswa dengan npm 0281

Jika parameter NPM tidak diset dalam request URL maka akan menampilkan halaman sebagai berikut :



Masukkan npm Anda

Kondisi error jika npm student tidak ditemukan maka menampilkan seperti eror berikut ini :



Gagal menghapus mahasiswa karena npm 1223 tidak ditemukan