

Write-Up Tutorial 3 APAP

Nama : Lintang Matahari Hasani Kelas : A
NPM : 1506689231

Tutorial

Implementasi method selectStudent()

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

Penjelasan method selectStudent() pada Class InMemoryStudentService

Method ini dimaksudkan untuk mencari StudentModel dengan npm yang diminta dan mengembalikan object StudentModel yang dicari. Implementasi yang dilakukan untuk memenuhi requirement ini adalah dengan melakukan looping di ArrayList studentList yang menyimpan object StudentModel dan mencocokkan npm object di setiap index dengan npm yang diberikan. Jika npm sama, method akan mengembalikan StudentModel tersebut. Jika npm yang sama tidak ada pada ArrayList, method mengembalikan null yang selanjutnya akan ditangani di Class StudentController.

Membuat Controller dan Fungsi Add

Jalankan program dan buka:

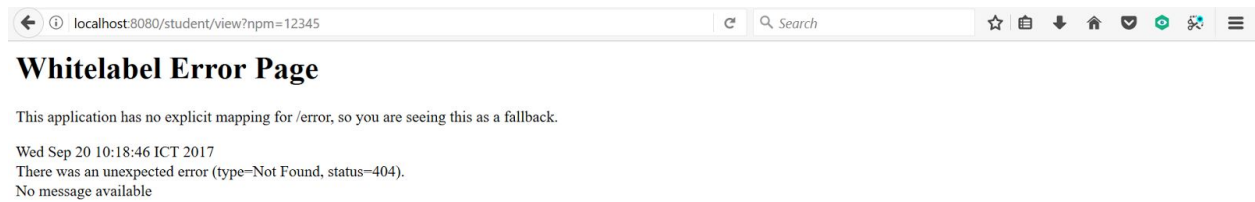
localhost:8080/student/add?npm=12345&name=chanek&gpa=3.43



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

Tidak terjadi error. Pada browser ditampilkan template view.html yang berisi tulisan “Data berhasil ditambahkan.” Pada saat path tersebut dibuka melalui browser, Controller menjalankan method add dan data student baru ditambahkan ke ArrayList studentList.

localhost:8080/student/view?npm=12345

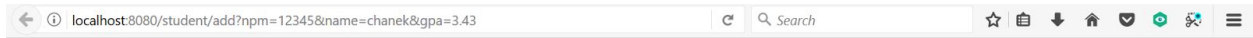


Pertanyaan 2 : *apakah hasilnya? Jika error , tuliskan penjelasan Anda.*

Data student tidak muncul meski penambahan data student berhasil. Pada browser ditampilkan pesan error. Pada class StudentController belum terdapat method view yang mengembalikan templates “view”. Templates “view.html” juga belum dibuat sehingga controller tidak dapat mengakses view tersebut.

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, data student tersebut berhasil 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.

Wed Sep 20 14:32:46 ICT 2017

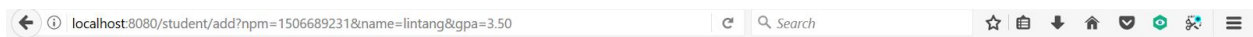
There was an unexpected error (type=Internal Server Error, status=500).

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

Pertanyaan 4 : *apakah data Student tersebut muncul? Jika tidak, mengapa?*

Data student tersebut tidak muncul. Hal ini dikarenakan data student yang disimpan dalam ArrayList studentList dihapus oleh Java GarbageCollector saat program dihentikan.

Coba tambahkan data Student lainnya dengan NPM yang berbeda.



Data berhasil ditambahkan



NPM = 1506689231

Name = lintang

GPA = 3.5



Data berhasil ditambahkan



NPM = 1506681234

Name = fulan

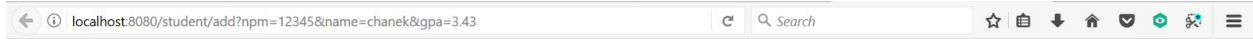
GPA = 3.5

Nama : Lintang Matahari Hasani Kelas : A
NPM : 1506689231

Method View All

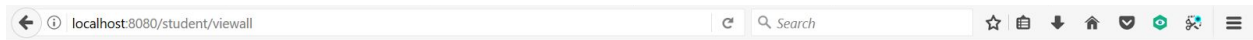
Jalankan program dan buka

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



Data berhasil ditambahkan

`localhost:8080/student/viewall`



No. 1

NPM = 12345

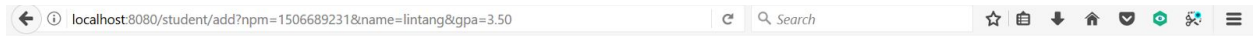
Name = chanek

GPA = 3.43

Pertanyaan 5 : apakah data Student tersebut muncul?

Ya, data tersebut ditampilkan.

Coba tambahkan data Student lainnya dengan NPM yang berbeda



Data berhasil ditambahkan



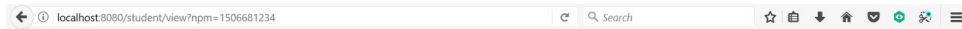
NPM = 1506689231

Name = lintang

GPA = 3.5



Data berhasil ditambahkan

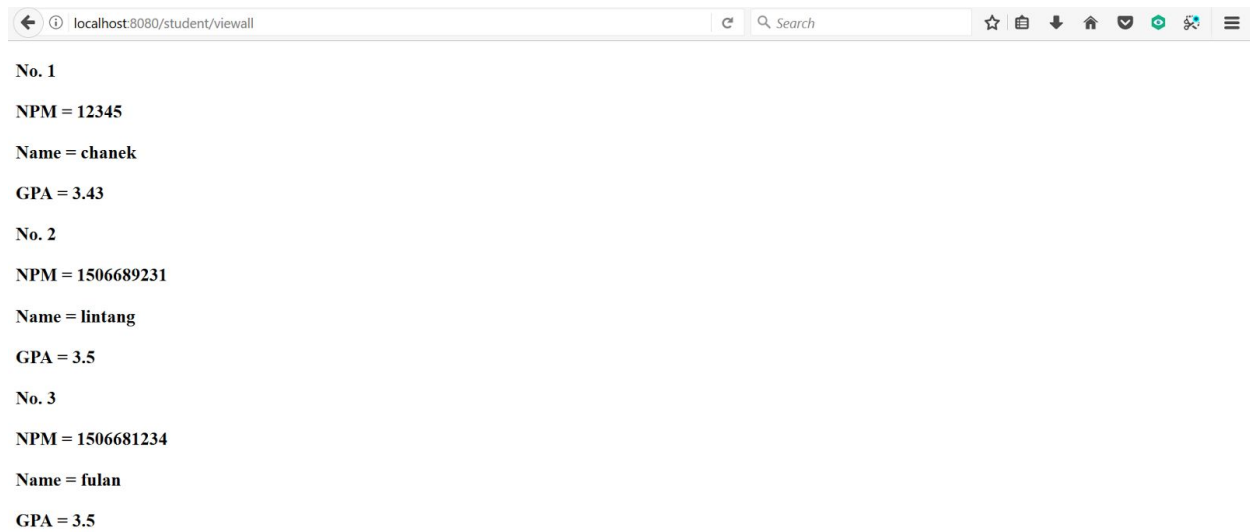


NPM = 1506681234

Name = fulan

GPA = 3.5

lalu buka *localhost:8080/student/viewall*



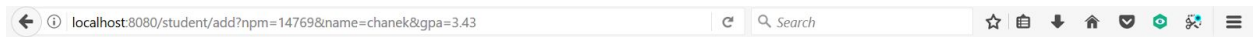
Pertanyaan 6 : Apakah semua data Student muncul?

Ya, semua data student yang dimasukkan berhasil ditampilkan.

Latihan

1. Pada StudentController tambahkan sebuah method viewStudent 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*.

localhost:8080/student/add?npm=14769&name=chanek&gpa=3.43



Data berhasil ditambahkan

localhost:8080/student/view/14769

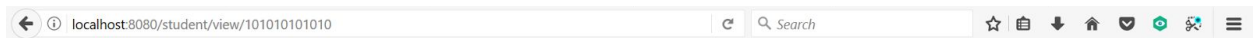


NPM = 14769

Name = chanek

GPA = 3.43

Jika data student tidak ditemukan: *localhost:8080/student/view/101010101010*



Maaf, NPM kosong atau tidak ditemukan

Jika data student kosong



Maaf, NPM kosong atau tidak ditemukan

Langkah Penyelesaian Soal Latihan No.1 :

1) Membuat method view() pada Class StudentController

```
@RequestMapping("/student/view/{npm}")
public String view(
    Model model,
    @PathVariable String npm
)
{
    StudentModel student = studentService.selectStudent(npm);
    if(student == null) {
        return "errorpage";
    }
    model.addAttribute("student", student);
    return "view";
}
```

Method ini menerima string npm dan akan memanggil function selectStudent untuk memilih StudentModel berdasarkan data npm. Jika npm tersebut tidak ditemukan, yakni ketika method selectStudent mengembalikan null, method ini akan mengembalikan view errorpage.html

2) Membuat method viewEmpty() pada Class StudentController

```
//Jika user tidak memasukkan npm ketika mengakses path view
@RequestMapping("/student/view")
public String viewEmpty()
{
    return "errorpage";
}
```

Method ini menerima path dengan path variable kosong dan mengembalikan view errorpage.html

3) Membuat view errorpage.html

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Error Page</title>
5   </head>
6   <body>
7     <h1>Maaf, NPM kosong atau tidak ditemukan</h1>
8   </body>
9 </html>
```

Laman error ini berisi informasi bahwa npm pada path kosong atau tidak ditemukan.

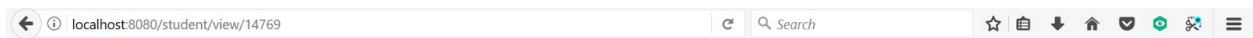
2. Tambahkan fitur untuk melakukan delete Student berdasarkan NPM. Misalnya, setelah melakukan add Student pada soal nomor 1, cobalah untuk melakukan delete data tersebut

dengan mengakses halaman *localhost:8080/student/delete/14769*. Tampilkan sebuah halaman yang memberikan informasi bahwa data tersebut telah berhasil dihapus.



Student berhasil dihapus dari database

Informasi student tersebut berhasil dihapus dan dapat diketahui dari tidak adanya data mengenai student tersebut ketika mengakses *localhost:8080/student/view/14769*



Maaf, NPM kosong atau tidak ditemukan

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.

Jika NPM yang diberikan tidak ditemukan: *http://localhost:8080/student/delete/14760*



Maaf, NPM kosong atau tidak ditemukan

Jika NPM kosong: *http://localhost:8080/student/delete/*



Maaf, NPM kosong atau tidak ditemukan

Langkah Penyelesaian Soal Latihan No. 2 :

1) Membuat void method deleteStudent pada interface StudentService.

```
1 package com.example.tutorial3.service;
2
3 import java.util.List;
4
5 public interface StudentService {
6     StudentModel selectStudent(String npm);
7     List<StudentModel> selectAllStudents();
8     void addStudent(StudentModel student);
9     void deleteStudent(String npm);
10 }
11
12
```

Method ini akan diimplementasikan di Class InMemoryStudentService. Method ini menerima String npm.

2) Mengimplementasikan void method deleteStudent pada Class InMemoryStudentService.

```
public void deleteStudent(String npm) {
    for(int i = 0; i < studentList.size(); i++) {
        if(studentList.get(i).getNpm().equals(npm)) {
            studentList.remove(i);
        }
    }
}
```

Method ini menerima String npm student yang ingin dihapus. Pertama, method akan melakukan looping pada ArrayList untuk mencari object StudentModel dengan npm yang dicari. Ketika npm ditemukan, object tersebut akan dihapus dari ArrayList.

3) Membuat method delete() pada Class StudentController

```
@RequestMapping("/student/delete/{npm}")
public String delete(
    @PathVariable String npm
) {
    StudentModel student = studentService.selectStudent(npm);
    if(student == null) {
        return "errorpage";
    }
    studentService.deleteStudent(npm);
    return "delete";
}
```

Method ini menerima String npm student yang ingin dihapus. Jika npm terdapat pada object yang tersimpan pada ArrayList, function deleteStudent akan dipanggil dan StudentModel yang diinginkan akan dihapus dan method ini mengembalikan view delete.html. Jika npm tidak ditemukan method akan mengembalikan view errorpage.html.

4) Membuat method deleteEmpty() pada Class StudentController

```
89 @RequestMapping("/student/delete")
90 public String deleteEmpty()
91 {
92     return "errorpage";
93 }
```

Method ini menerima path dengan path variable kosong. Method ini mengembalikan view errorpage.html.