

Ringkasan materi:

Pada tutorial 3 ini, terdapat beberapa materi tambahan dari tutorial sebelumnya yang diberikan diantaranya:

Service adalah sebuah layer komponen yang melayani controller, model dan database yang mencangkup ke dalam objek sehingga kita bisa mengolah database dalam bentuk objek dan memanggilnya ke view.

Mapper adalah sebuah pemanggil untuk mendapatkan database dan kemudian diolah oleh service.

Database adalah kumpulan data yang akan kita simpan.

Method selectStudent:

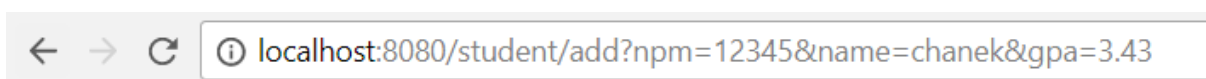
```
@Override
public StudentModel selectStudent(String npm) {
    for(int i=0; i<studentList.size(); i++) {
        if(studentList.get(i).getNpm().equals(npm)) {
            return studentList.get(i);
        }
    }
    return null;
}
```

Tutorial:

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.

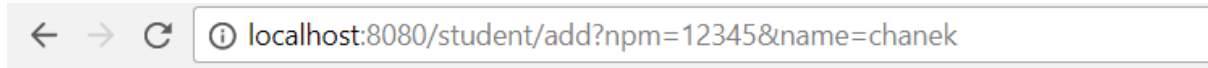


Data berhasil ditambahkan

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

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

Terjadi error, informasi GPA bersifat wajib dikarenakan @RequestParam-nya “required true”



Whitelabel Error Page

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

Sat Sep 23 16:11:43 ICT 2017

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

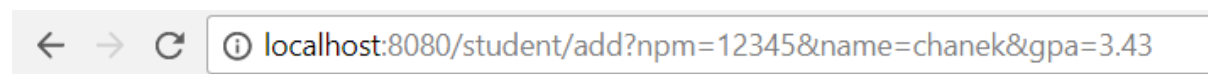
Required double parameter 'gpa' is not present

3. Jalankan program dan buka

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

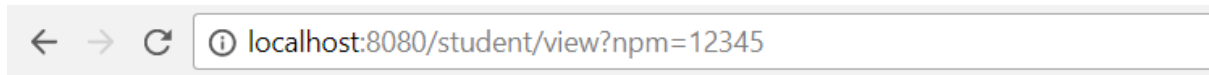
GPA = 3.43

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?

Karena setiap kali memulai program, database di List belum ada (belum dimasukan)



Whitelabel Error Page

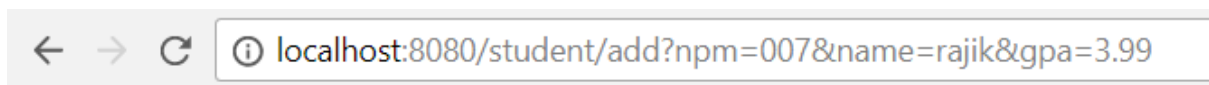
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Sep 23 16:16:27 ICT 2017

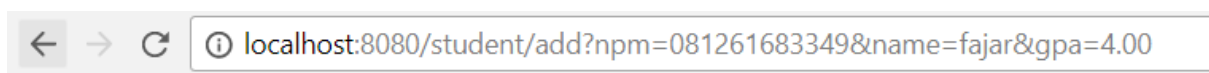
There was an unexpected error (type=Internal Server Error, status=500).

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

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



Data berhasil ditambahkan

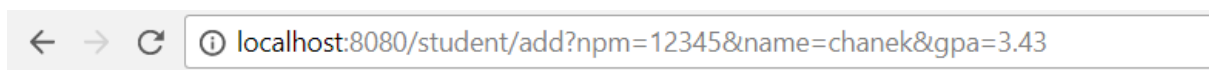


Data berhasil ditambahkan

3. Jalankan program dan buka

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

localhost:8080/student/viewall,



Data berhasil ditambahkan

A screenshot of a web browser window. The address bar shows the URL 'localhost:8080/student/viewall'. The page displays a table with student data.

No. 1
NPM = 12345
Name = chanek
GPA = 3.43
No. 2
NPM = 081261683349
Name = fajar
GPA = 4.0
No. 3
NPM = 007
Name = rajik
GPA = 3.99

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

Pertanyaan 6: Apakah semua data Student muncul?

← → ↻ localhost:8080/student/viewall 🔍 ☆ ⋮	
No. 1	
NPM = 12345	
Name = chanek	
GPA = 3.43	
No. 2	
NPM = 081261683349	
Name = fajar	
GPA = 4.0	
No. 3	
NPM = 007	
Name = rajik	
GPA = 3.99	
No. 4	
NPM = 081261683349	
Name = fajar	
GPA = 4.0	
No. 5	
NPM = 12345	
Name = chanek	
GPA = 3.43	

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**.

Masukan data:

npm=14769 nama=marseto gpa=3.5

← → ↻	localhost:8080/student/add?npm=14769&name=marseto&gpa=3.5
-------	---

Data berhasil ditambahkan

Melihat npm=14769 menggunakan Path Variable: localhost:8080/student/view/14769

← → ↻	localhost:8080/student/view/14769
-------	-----------------------------------

NPM = 14769

Name = marseto

GPA = 3.5

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

Melihat npm=8888888(tidak ada data dengan npm=8888888) menggunakan Path Variable:
localhost:8080/student/view/8888888

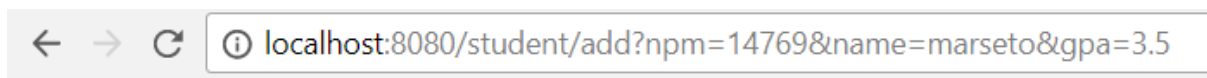


Nomor NPM kosong atau tidak ditemukan

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.

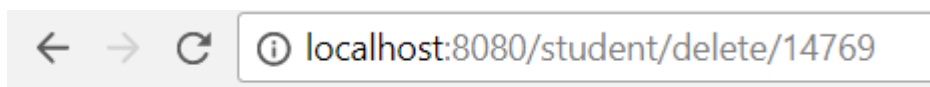
Masukan data:

npm=14769 nama=marseto gpa=3.5



Data berhasil ditambahkan

Menghapus data dengan npm=14769:



Data tersebut telah berhasil dihapus

Menghapus npm=8888888(tidak ada data dengan npm=8888888)

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.



Nomor NPM kosong atau tidak ditemukan dan proses delete dibatalkan

Penjelasan fitur delete:

Pada interface StudentService.java, membuat method deleteStudent yang dapat melakukan manipulasi kelas Student di InMemoryStudentService.java

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

Membuat method deleteStudent di InMemoryStudentService.java guna untuk menghapus data sesuai object yang di select

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

Membuat method dengan @RequestMapping di StudentController.java, dengan meminta npm yang kemudian di select di sebuah object, setelah mendapatkan objectnya, baru object tersebut di delete.

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

Membuat view dengan delete.html yang berisikan keluaran dari @RequestMapping

```
1 <!DOCTYPE html>  
2 <html xmlns:th="http://www.thymeleaf.org">  
3 <head>  
4 <title>Delete Student by NPM</title>  
5 </head>  
6 <body>  
7 <h3 th:text="${student} != null ? 'Data tersebut telah berhasil dihapus' : 'Nomor NPM kosong atau tidak ditemukan dan proses delete dibatalkan'">  
8     Student NPM</h3>  
9 </body>  
10 </html>  
11  
12
```