

## Wrap-up Tutorial 2

Pada tutorial 2 ini, saya belajar mengenai bagaimana membuat *project from the scratch* dengan membuat proyek Hello World menggunakan *framework* Spring Boot. Kemudian, pada kesempatan ini juga belajar mengenai penggunaan “RequestMapping” yang digunakan untuk memetakan method yang telah dibuat ketika melakukan request melalui url pada browser. Selanjutnya, belajar juga mengenai penggunaan “RequestParam” yang berguna untuk meminta parameter yang harus dikirim ketika melakukan request terhadap halaman suatu web. Kemudian, belajar juga penggunaan “Path Variable” yang memiliki hamper sama dengan “RequestParam” akan tetapi memiliki perbedaan dalam melakukan request-nya, jika tidak ada parameter yang dikirim request web dapat dipetakan kepada “RequestMapping” tertentu, begitupun sebaliknya.

## Latihan Project Hello World

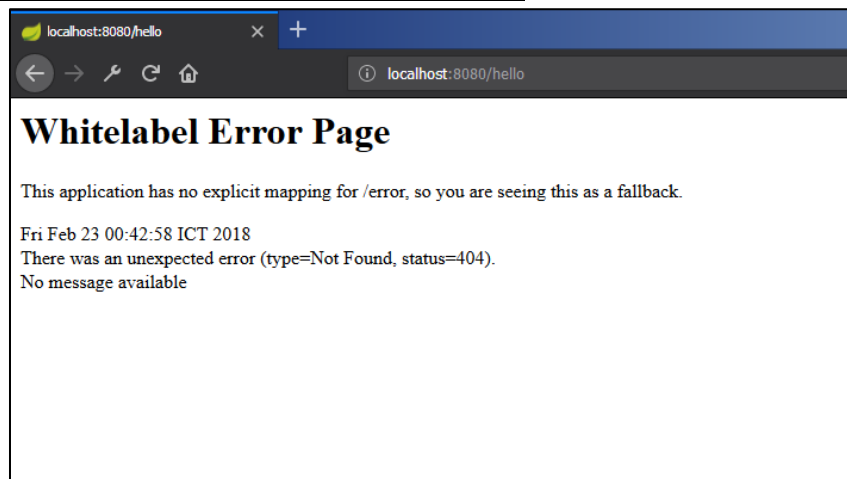
1. Ganti baris tersebut menjadi @RequestMapping("/hello123")

### Pertanyaan:

Apakah compile error? **Tidak ada error.**

Jika tidak, stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apa yang terjadi?

**Tampilan tidak bisa diakses, lihat Gambar 1.**



Gambar 1. Project Hello World Pertanyaan Ke-1

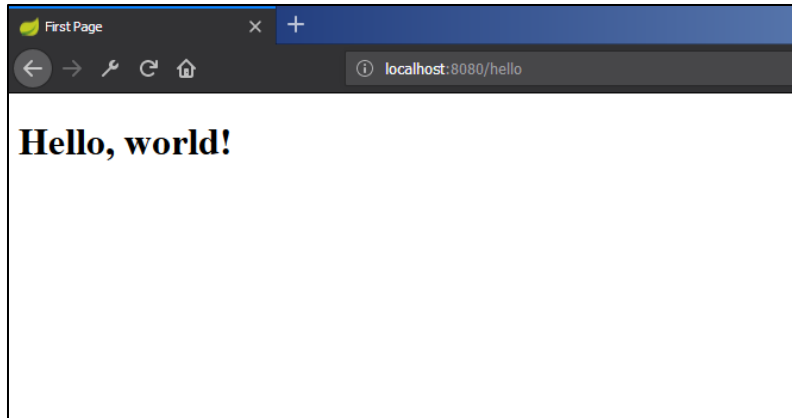
2. Ganti nama method index() dengan nama method hello()

### Pertanyaan:

Apakah compile error? **Tidak ada error.**

Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apakah page hello sebelumnya masih muncul?

**Tampilan masih bisa diakses, lihat Gambar 2.**



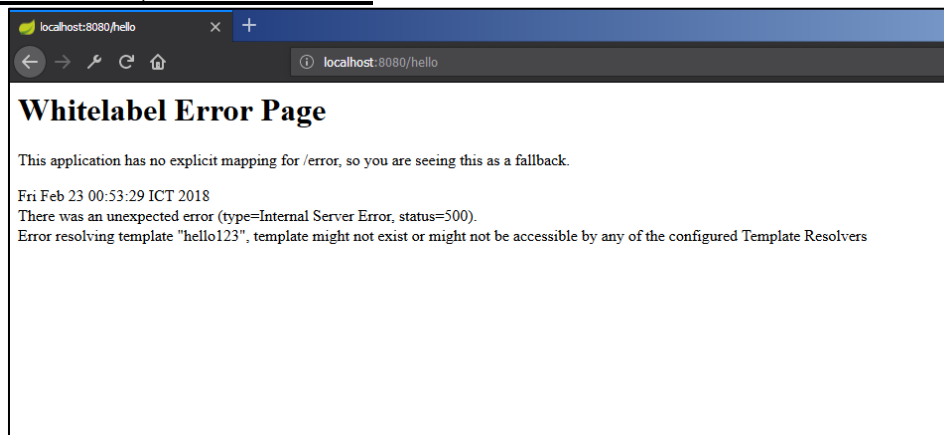
Gambar 2. Project Hello World Pertanyaan Ke-2

3. Ganti string return type menjadi return "hello123";

**Pertanyaan:**

Apakah compile error? **Tidak ada error.**

Jika tidak, Stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apakah page hello sebelumnya masih muncul? **Tampilan page hello tidak muncul, lihat Gambar 3.**



Gambar 3. Project Hello World Pertanyaan Ke-3

**Pertanyaan:**

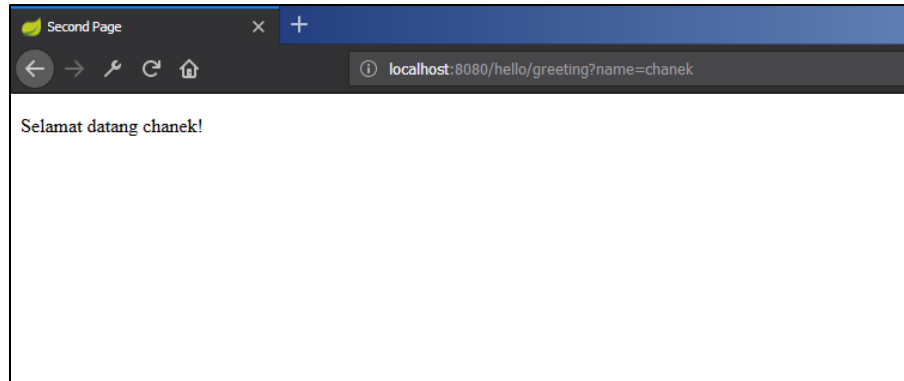
Menandakan apakah String yang di-return tersebut? **String tersebut menandakan file html mana yang ditunjuk oleh sebuah method. Dalam Gambar 3, file html yang ditunjuk bernama "hello123" dan pada project tidak ada file tersebut, sehingga terjadi error seperti pada Gambar 3.**

### Latihan Request Parameter

1. Ubah nilai anotasi RequestMapping dari "/greeting" menjadi "/hello/greeting"

Buka localhost:8080/hello/greeting?name=chanek

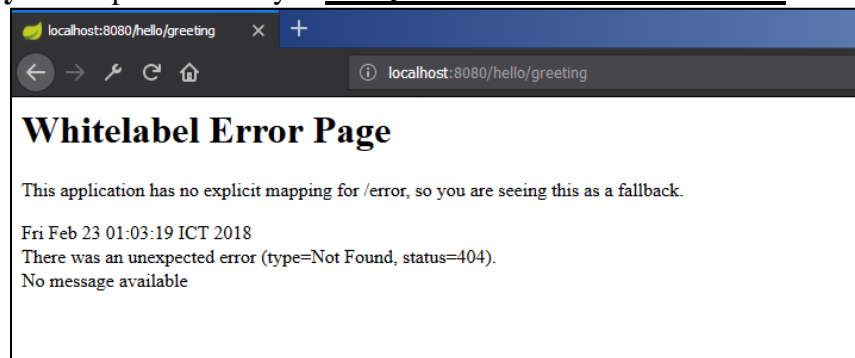
**Pertanyaan:** apakah hasilnya? **Tidak ada error. Lihat Gambar 4.**



Gambar 4. Latihan Request Parameter Pertanyaan Ke-1

2. Akses **localhost:8080/greeting**

**Pertanyaan:** Apakah hasilnya? Tampilan error. Lihat Gambar 5.



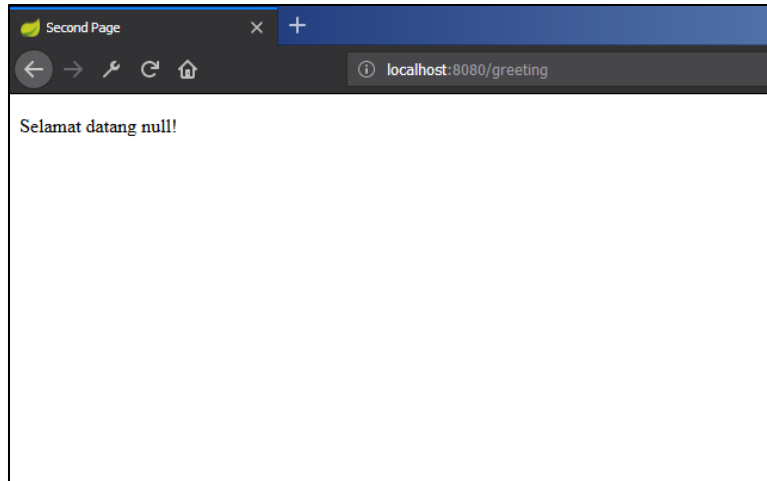
Gambar 5. Latihan Request Parameter Pertanyaan Ke-2

3. Ubah header method greeting seperti Gambar 6.

```
public String greeting (@RequestParam(value = "name", required = false) String  
name, Model model)
```

Gambar 6. Header method greeting

**Pertanyaan:** Apakah hasilnya? Tampilan tidak error. Lihat Gambar 7.



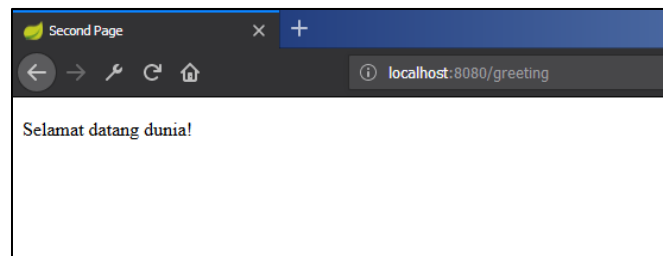
Gambar 7. Latihan Request Parameter Pertanyaan Ke-3

4. Ubah header method greeting seperti Gambar 8.

```
public String greeting (@RequestParam(value = "name", required = false,  
defaultvalue = "dunia") String name, Model model)
```

Gambar 8. Header method greeting Ke-4

**Pertanyaan:** Apakah hasilnya (localhost:8080/greeting)? **Tampilan tidak error. Lihat Gambar 9.**



Gambar 9. Latihan Request Parameter Pertanyaan Ke-4

5. Perhatikan bahwa pada berkas greeting.html, tag paragraf yang kita tambahkan adalah sebagai berikut (lihat Gambar 10):

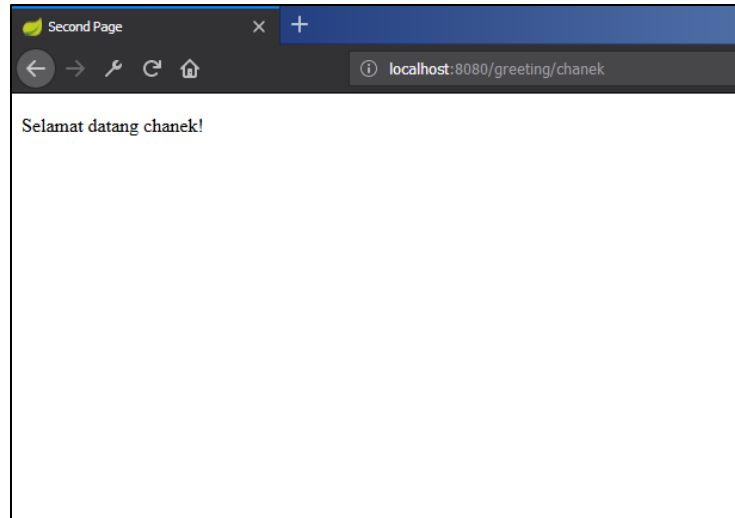
```
<p th:text="'Selamat datang ' + ${name} + '!'">Sapaan untuk user</p>
```

Gambar 10. greeting.html

**Pertanyaan:** Mengapa tulisan “Sapaan untuk user” tidak pernah muncul? **Tulisan tersebut tidak muncul sebab digantikan oleh value pada attribute “th:text”. Sehingga, text apapun yang ada dalam attribute html akan tergantikan textnya jika menggunakan attribute thymeleaf “th:text”.**

## Path Variable

**Berikut hasil dari tutorial Path Variable. Lihat Gambar 11.**



**Gambar 11. Path Variable**

### **Latihan Path Variable**

1. Akses **localhost:8080/greeting/**

**Pertanyaan:** Apa hasilnya?

Ubah method `greetingPath` menjadi seperti Gambar 12.

```
@RequestMapping(value = {"/greeting", "greeting/{name}"})  
public String greetingPath(@PathVariable Optional<String> name, Model model) {  
    if (name.isPresent()) {  
        model.addAttribute("name", name.get());  
    } else {  
        model.addAttribute("name", "apap");  
    }  
    return "greeting";  
}
```

**Gambar 12. Method `greetingPath`**

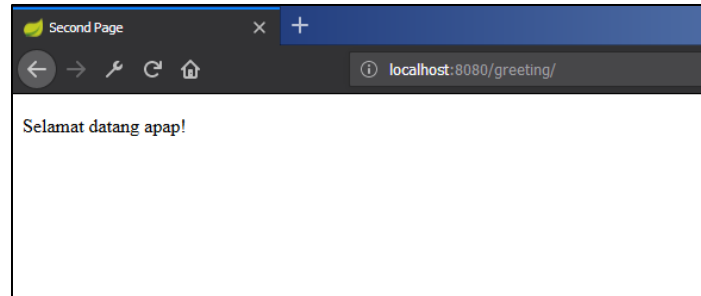
dan tambahkan import (lihat Gambar 13).

```
import java.util.Optional;
```

**Gambar 13. Import `java.util.Optional`**

Akses **localhost:8080/greeting/**

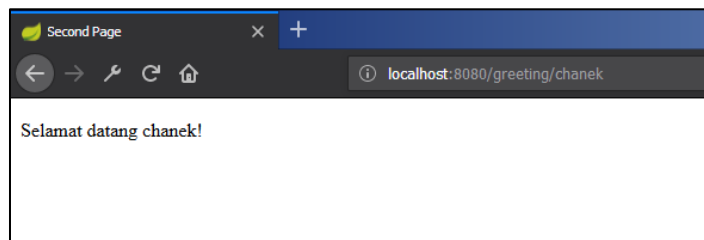
**Pertanyaan:** Apa hasilnya? **Menampilkan tulisan "Selamat data apap!". Lihat Gambar 14.**



Gambar 14. Hasil Path Variable Ke-1

Akses **localhost:8080/greeting/chanek**

**Pertanyaan:** Apa hasilnya? Menampilkan tulisan “Selamat data chanek!”. Lihat **Gambar 15.**



Gambar 15. Hasil Path Variable Ke-2

## Latihan Membuat Method Perkalian

### 1. Method Perkalian (Lihat Gambar 16)

```
@RequestMapping("/perkalian")
public String perkalian (@RequestParam(value = "a", required = false, defaultValue = "0") String a
    , @RequestParam(value = "b", required = false, defaultValue = "0") String b, Model model) {
    model.addAttribute("a", a);
    model.addAttribute("b", b);
    model.addAttribute("hasil", Integer.parseInt(a) * Integer.parseInt(b));
    return "perkalian";
}
```

Gambar 16. Method Perkalian

### 2. View Perkalian (Lihat Gambar 17)

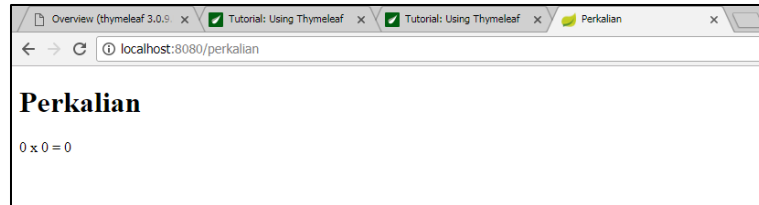
```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>Perkalian</title>
5   </head>
6   <body>
7     <h1>Perkalian</h1>
8     <p th:text="${a} + ' x ' + ${b} + ' = ' + ${hasil}">Hasil Perkalian</p>
9   </body>
10 </html>
```

Gambar 17. perkalian.html

### 3. Tampilan (Lihat Gambar 18)

Nama: Arri Kurniawan  
NPM: 1606954735

APAP 2018  
Tutorial 2



**Gambar 18. Tampilan Perkalian**