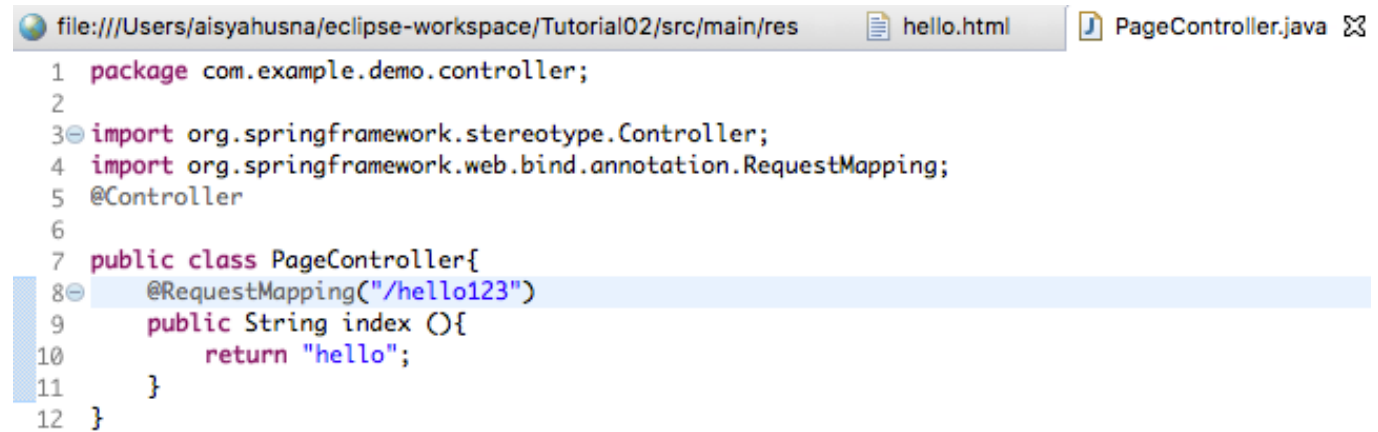


## Tutorial 2

### Latihan Project Hello World

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



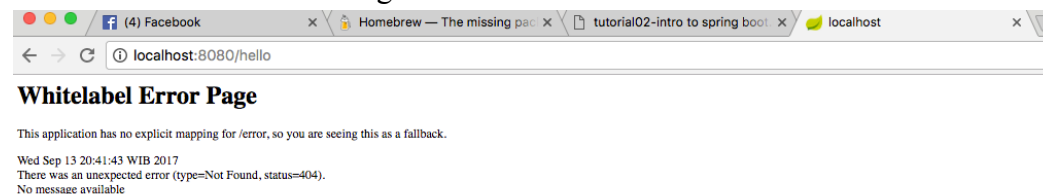
```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 @Controller
6
7 public class PageController{
8     @RequestMapping("/hello123")
9     public String index (){
10         return "hello";
11     }
12 }
```

Pertanyaan: Apakah compile error?

Tidak

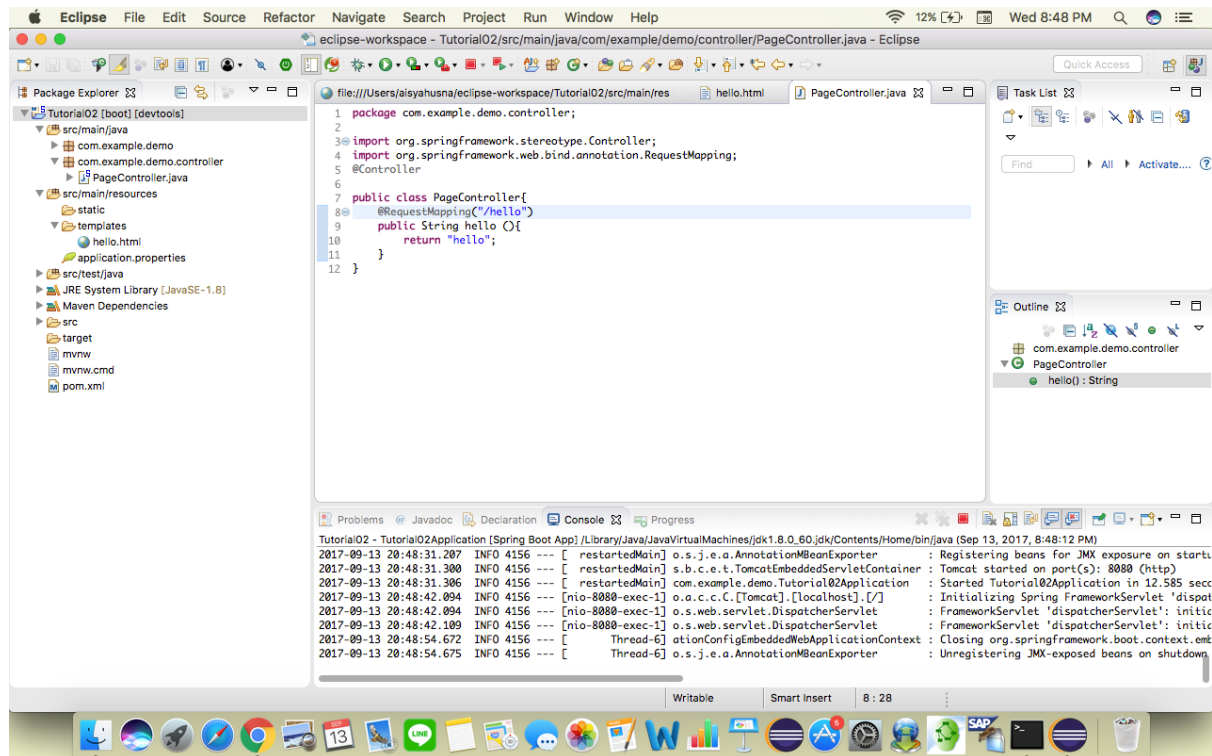
Jika tidak, stop Spring Boot yang sedang berjalan, run kembali dan buka localhost:8080/hello apa yang terjadi?

Muncul Whitelabel Error Page



Kembalikan RequestMapping menjadi /hello

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

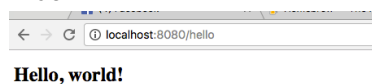


Pertanyaan: Apakah compile error?

Tidak

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

Tidak



Kembalikan nama method menjadi index()

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

Pertanyaan: Apakah compile error?

Tidak

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



## Whitelabel Error Page

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

Wed Sep 13 19:24:13 ICT 2017

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

Error resolving template "hello123", template might not exist or might not be accessible by any of the configured Template Resolvers

Kembalikan return type menjadi "hello"

Pertanyaan: Menandakan apakah String yang di-return tersebut?

Jawab: Menandakan file html tersebut akan dieksekusi.

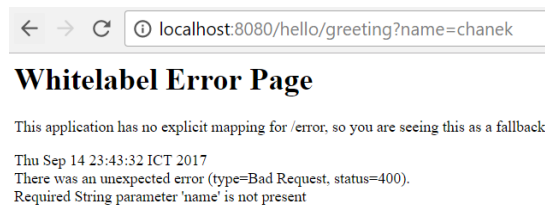
## Latihan Request Parameter

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

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

Pertanyaan: apakah hasilnya?

Muncul Whitelabel Error Page

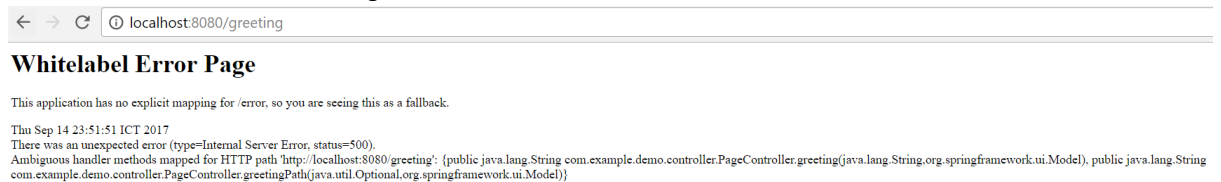


Kembalikan request map menjadi @RequestMapping("/greeting")

2. Akses localhost: 8080/greeting

Pertanyaan: Apakah hasilnya?

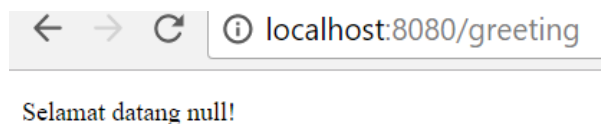
Muncul Whitelabel Error Page



Ubah header method greeting. Stop Spring Boot yang sedang berjalan, run kembali, buka localhost: 8080/greeting

Pertanyaan: Apakah hasilnya?

Muncul "Selamat datang null!"



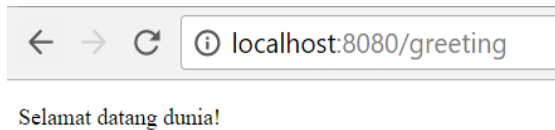
3. Ubah header method greeting

```
1 package com.example.demo.controller;
2
3 import org.springframework.stereotype.Controller;
4 import org.springframework.ui.Model;
5 import org.springframework.web.bind.annotation.RequestMapping;
6 import org.springframework.web.bind.annotation.RequestParam;
7
8 @Controller
9 public class PageController {
10     @RequestMapping("/hello")
11     public String index() {
12         return "hello";
13     }
14
15     @RequestMapping("/greeting")
16     public String greeting (@RequestParam(value = "name", required = false,
17         defaultValue = "dunia") String name, Model model)
18     {
19         model.addAttribute("name", name);
20         return "greeting";
21     }
22 }
```

Stop Spring Boot yang sedang berjalan, run kembali, dan buka localhost:8080/greeting.

Pertanyaan: apakah hasilnya?

Muncul "Selamat datang dunia!"



4. Perhatikan bahwa pada berkas greeting.html, tag paragraf yang kita tambahkan adalah sebagai berikut:

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

Pertanyaan: Mengapa tulisan "Sapaan untuk user" tidak pernah muncul?

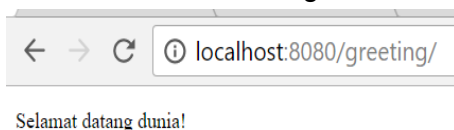
Karena teroverwrite dengan tag th:text milik Thymeleaf. Th:text tersebut mengganti teks yang ada di tag p. Hal tersebut membuat semua teks yang ada di tag atribut tersebut terganti.

### Latihan Path Variable:

1. Akses localhost:8080/greeting/

Pertanyaan: Apa hasilnya?

Muncul "selamat datang dunia"



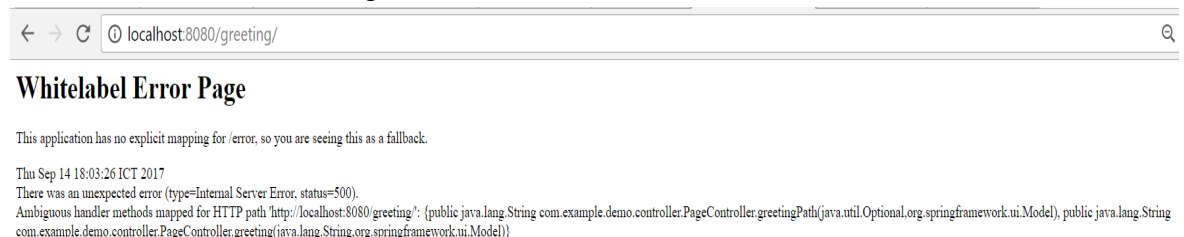
Ubah method greetingPath menjadi seperti berikut

```
@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", "dengklek");  
    }  
    return "greeting";  
}
```

Akses localhost:8080/greeting/

Pertanyaan: Apa hasilnya?

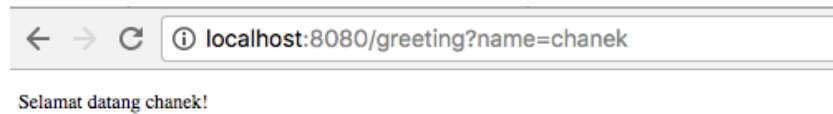
Muncul Whitelabel Error Page



Akses localhost:8080/greeting/chanek

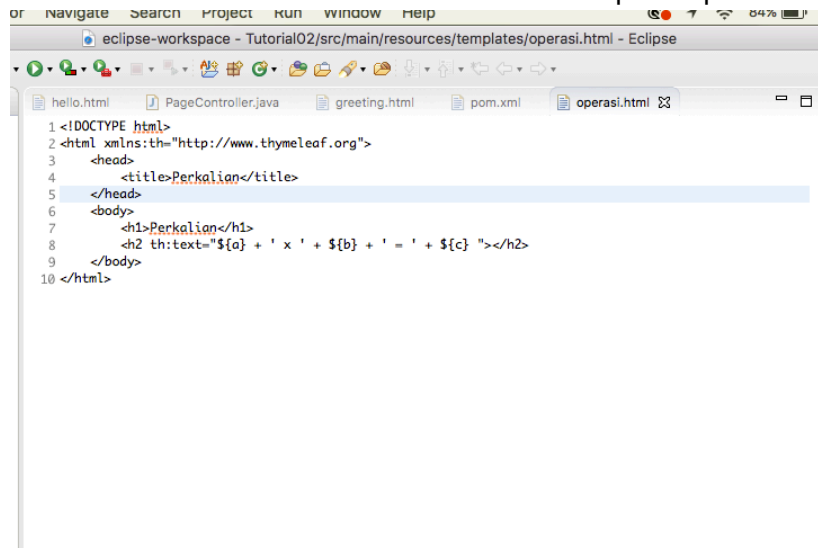
Pertanyaan: Apa hasilnya?

Muncul “Selamat datang chanek!”



## Latihan

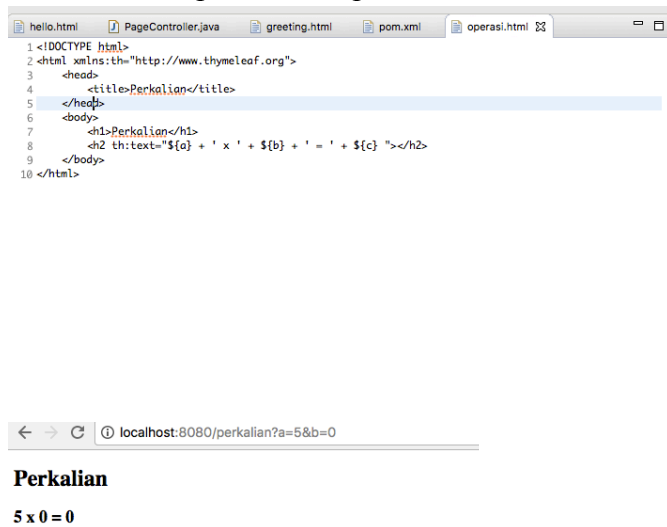
1. Buatlah satu halaman baru untuk melakukan operasi perkalian.



2. Halaman ini menerima 2 buah parameter dari query string yaitu a dan b.



3. Pada view tampilkan hasil perkalian a dan b.



```
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     <h2 th:text="${a} + ' x ' + ${b} + ' = ' + ${c}"></h2>
9   </body>
10 </html>
```

← → ↻ localhost:8080/perkalian?a=5&b=0

**Perkalian**

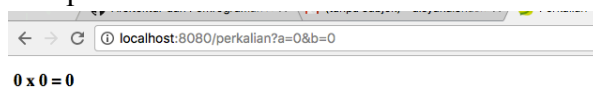
**5 x 0 = 0**

4. Jika a atau b tidak ada, nilai defaultnya 0.

```
    return "greeting";
  }
  @RequestMapping("/perkalian")
  public String greeting (@RequestParam(value = "a", required = false, defaultValue = "0") int a,
    @RequestParam(value = "b", required = false, defaultValue = "0") int b, Model model){

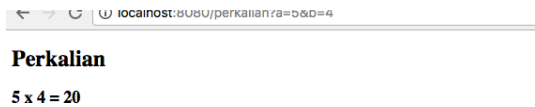
    int c = a * b;
    model.addAttribute("a", a);
    model.addAttribute("b", b);
    model.addAttribute("c", c);
    return "operasi";
  }
}
```

5. Tampilan



← → ↻ localhost:8080/perkalian?a=0&b=0

**0 x 0 = 0**



## 6. Lesson Learned

Menurut saya, yang saya pelajari pada tutorial kali ini adalah saya belajar cara kerja MVC pada Spring Boot. Saya mempelajari mengenai behavior router dari spring. Saya mempelajari bagaimana cara kerja controller sera passing-variable menggunakan metode request. Dengan mengetahui cara kerja Spring Boot, saya harap kedepannya ilmu yang saya dapat pada tutorial kali ini dapat bermanfaat.