

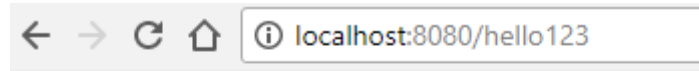
WRITE-UP

Pada tutorial kedua kali ini, saya diminta untuk membuat dan menggunakan *Controller* pada Spring Boot serta pengenalan dasar *routing* pada Spring Boot. Saya diminta untuk melakukan modifikasi kode pada tutorial 0 sebelumnya serta menjawab beberapa pertanyaan. Kemudian pada latihan saya diminta untuk membuat program kalkulator perkalian sederhana dengan menggunakan Request Parameter yang telah diajarkan. Saya akan mencoba untuk menjawab pertanyaan pertanyaan pada tutorial tersebut.

Latihan Project Hello World

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

- Apakah compile error?
Iya, akan terjadi error, dikarenakan `@RequestMapping` berubah, maka link dari localhost-nya pun juga harus diubah menjadi "hello123" agar dapat diakses.



Hello world

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

- Apakah compile error?
Tidak, dikarenakan hanya penamaan method saja.

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

- Apakah compile error?
Iya, akan terjadi error, dikarenakan tidak terdapat halaman html bernama hello123 yang dituju.
- Menandakan apakah String yang di-return tersebut?
Menandakan halaman yang dituju (html).

Latihan Request Parameter

1. Ubah nilai anotasi `RequestMapping` dari `"/greeting"` menjadi `"/hello/greeting"` Buka localhost: 8080/hello/greeting?name=chanek

- Apakah hasilnya?



Selamat datang chanek!

Chanek merupakan value name yang akan digunakan pada greeting.html

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

2. Akses localhost: 8080/greeting

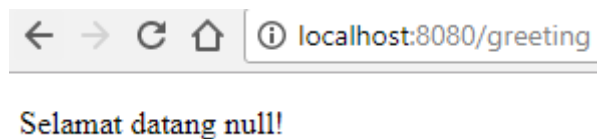
- Apakah hasilnya?
Akan terjadi error, dikarenakan tidak ada kasus yang meng-*handle* @RequestParam kosong.

Ubah header method greeting menjadi seperti berikut

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

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

- Apakah hasilnya?



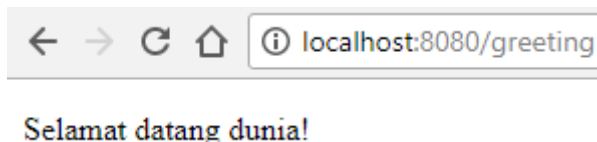
Nama akan menjadi "null", dikarenakan tidak terdapat *default value*.

3. Ubah header method greeting menjadi seperti berikut

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

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

- Apakah hasilnya?



Nilai "null" akan berubah menjadi dunia yang merupakan *default value*.

4. Perhatikan bahwa pada berkas greeting.html

- Mengapa tulisan "Sapaan untuk user" tidak pernah muncul?
Dikarenakan p (*paragraph*) telah diisi dengan text yang berasal dari file java.

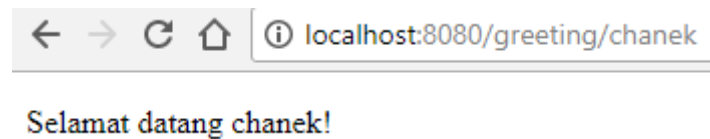
Path Variable:

1. Tambahkan method greetingPath pada PageController

```
@RequestMapping("/greeting/{name}")
public String greetingPath (@PathVariable String name, Model model)
{
    model.addAttribute("name", name);
    return "greeting";
}
```

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

- apakah hasilnya?

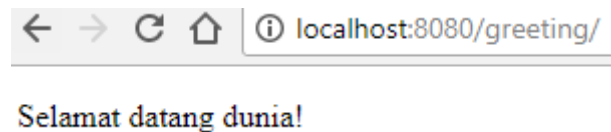


Chanek merupakan *path variable* name.

Latihan Path Variable:

1. Akses `localhost:8080/greeting/`

- Apa hasilnya?



Dikarenakan input kosong, maka muncul “dunia” yang merupakan *default value*.

Ubah method `greetingPath` menjadi seperti berikut

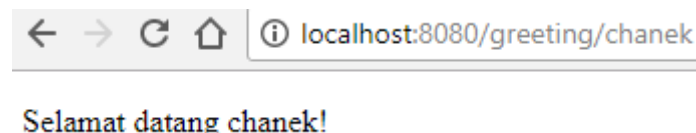
```
@RequestMapping(value = {"/greeting", "greeting/{name}"})
public String greetingPath(@PathVariable Optional<String> name, Model model) {
    if (name.isPresent()) {
        model.addAttribute("name", name.get());
    } else {
        model.addAttribute("name", "dengklek");
    }
    return "greeting";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

Akses `localhost:8080/greeting/`

- Apa hasilnya?
Akan terjadi error, dikarenakan Request mapping `localhost:8080/greeting/`
Terdapat pada dua method yaitu `greeting` dan `greeting path`. Pada `greeting` berisi “dunia”
dan pada `greetingPath` berisi “dangklek”.

Akses `localhost:8080/greeting/chanek`

- Apa hasilnya?



Chanek merupakan *path variable* name.