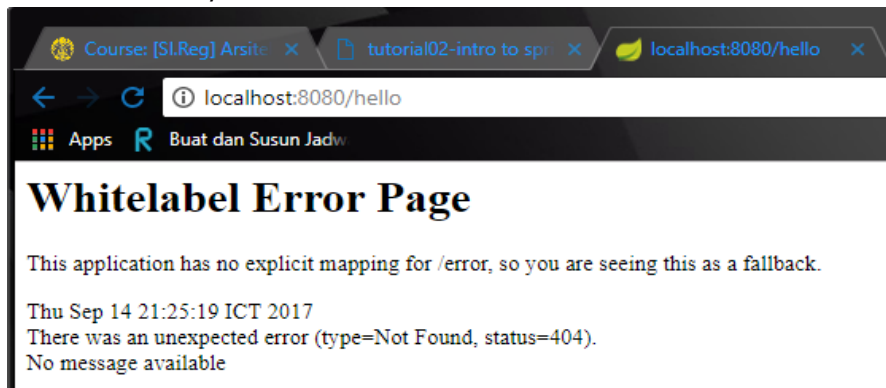


Awalia Mitha (1506689521)

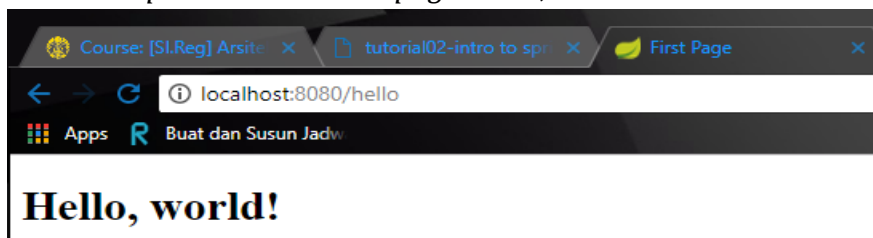
## Latihan *Project* Hello World

1. Tidak *compile error*. Tetapi *message* Hello world tidak muncul ketika diakses dengan localhost:8080/hello



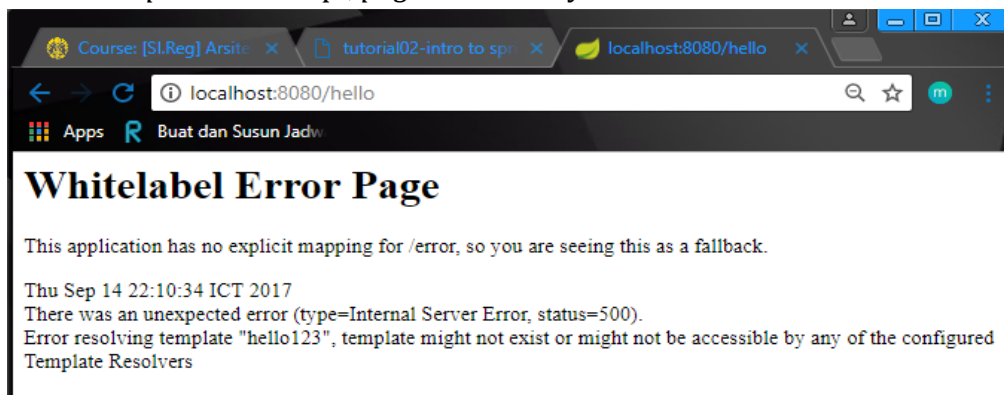
karena *page* yang di *request* berbeda dengan anotasi *RequestMapping*-nya. Tidak ada HTTP dengan *path* /helo123.

2. Tidak *compile error*. Muncul *page* Hello, world!



Karena nama method tidak berpengaruh pada apapun.

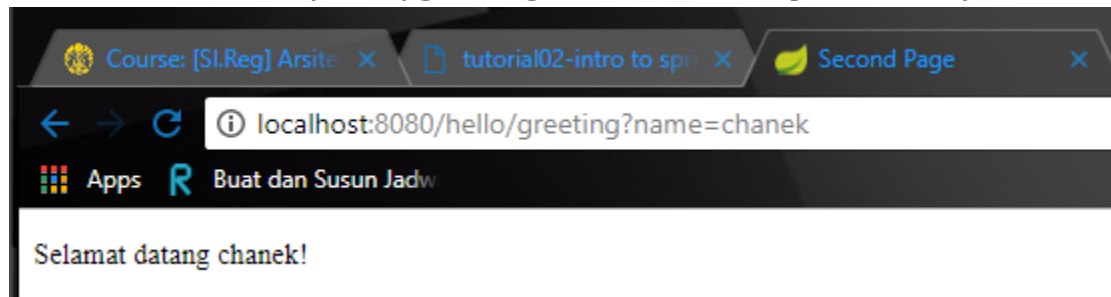
3. Tidak *compile error*. Tapi, *page* sebelumnya tidak muncul.



String yang di *return* menandakan *page html* mana yang akan dipanggil dalam *request mapping*.

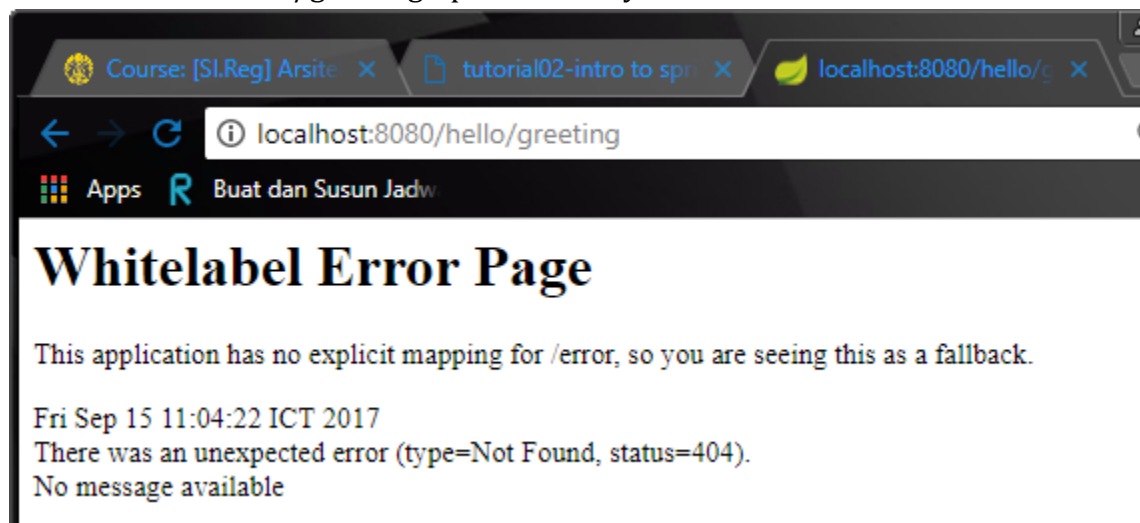
## Request Parameter (Query String)

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



Karena *url* sudah sama dengan anotasi pada *RequestMapping*. Jadi message tetap muncul.

2. Setelah *request map* dikembalikan menjadi `@RequestMapping("/greeting")`. Lalu akses **localhost:8080/greeting** Apakah hasilnya?

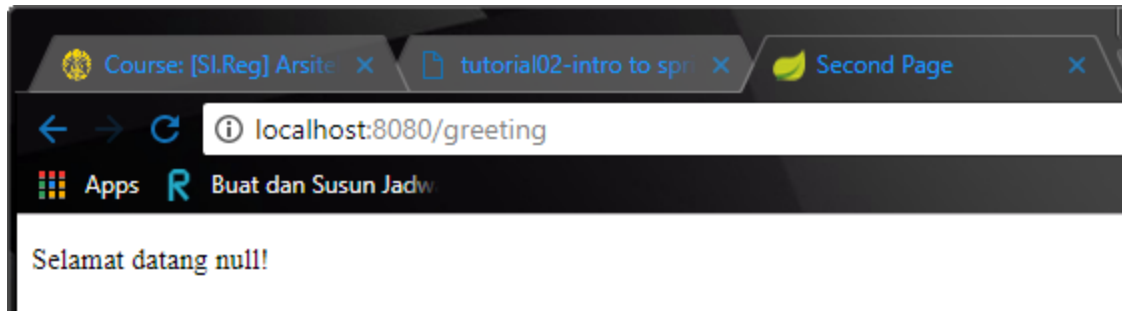


*Whitelabel Error Page* terjadi karena method *greeting* meminta parameter => ada *RequestParam(value = "name")*. Namun, di *url* tidak ada parameter, maka *error*.

Ubah *header* method *greeting* menjadi :

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

Stop *Spring Boot* yang sedang berjalan, run kembali, buka **localhost:8080/greeting** Apakah hasilnya?

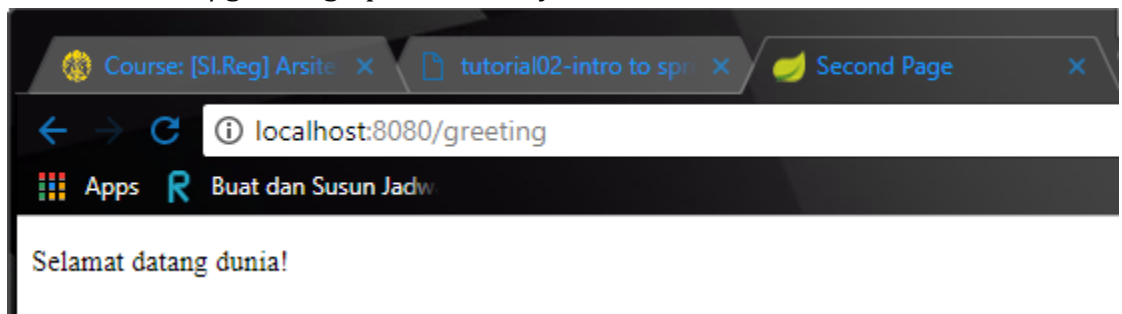


Message bisa ditampilkan, namun karena parameter tidak ada, jadi null. Itu karena sudah ditambahkan *required = false*. Jadi, bisa gak ada parameter yang diberikan.

3. Ubah *header method* :

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

Stop Spring Boot yang sedang berjalan, run kembali, buka localhost:8080/greeting Apakah hasilnya?



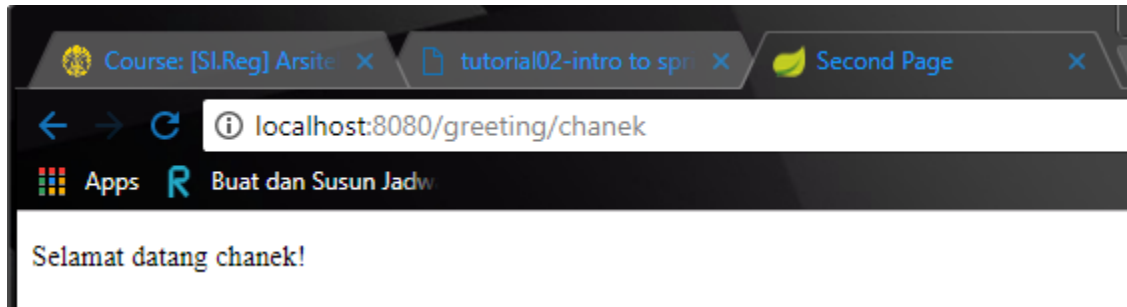
Message nya muncul, namun karena tidak ada parameter dan kita set *defaultValue = dunia*, maka *name* nya menjadi dunia. *DefaultValue* dapat digunakan ketika parameter yang diminta tidak ada, maka akan ada nilai pengganti nya.

4. Mengapa tulisan "Sapaan untuk user" tidak pernah muncul?

Sebenarnya tulisan tersebut muncul, hanya saja tertimpa oleh kalimat yang ada di `th:text`.

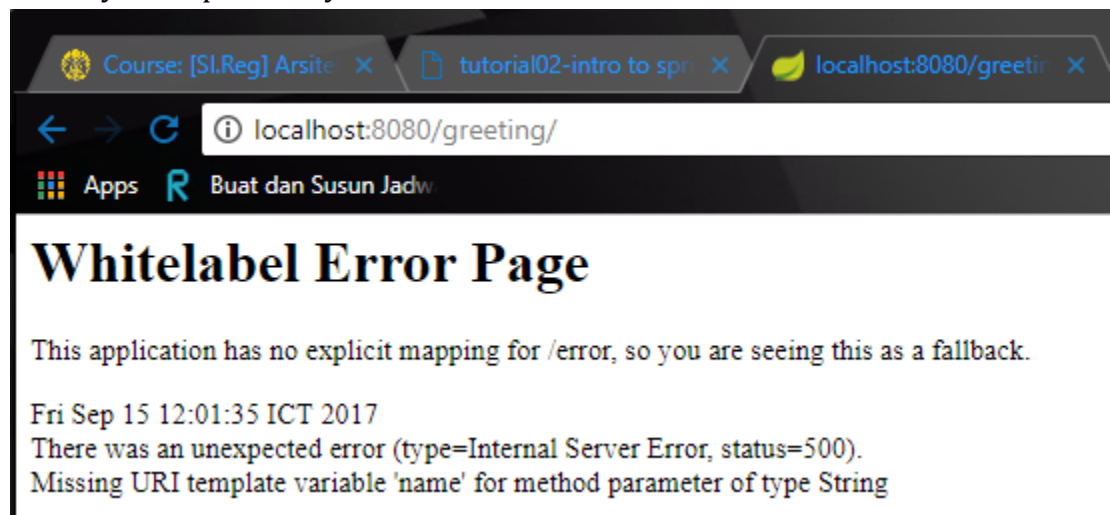
## Path Variable

Jawaban langkah no.2



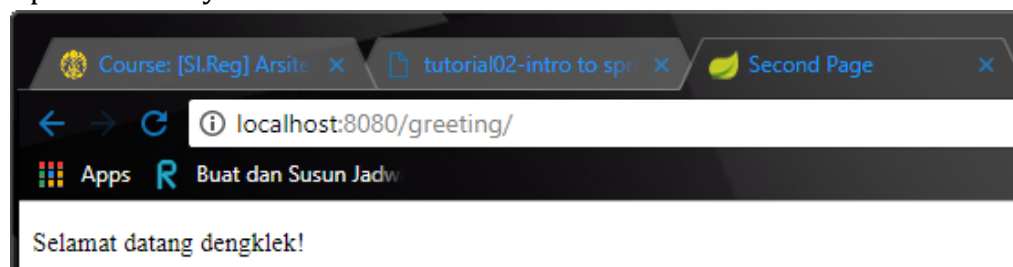
Latihan

1. Akses localhost:8080/greeting/  
Pertanyaan: Apa hasilnya?



Karena *path variable* nya tidak ada.

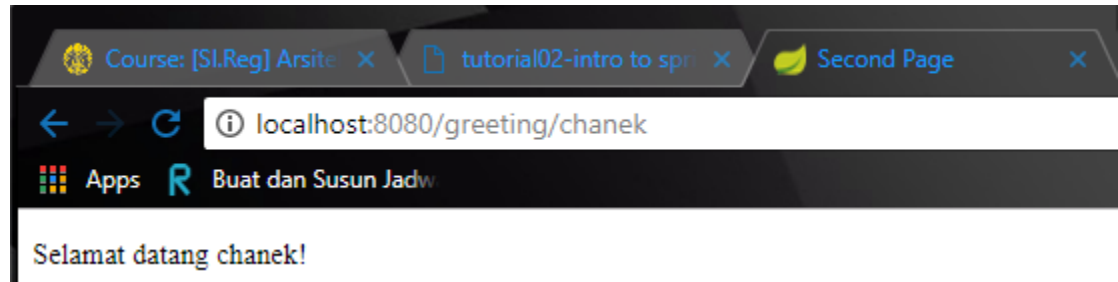
Ubah *method greetingPath*. Lalu akses localhost:8080/greeting/  
Apakah hasilnya?



Karena ada 2 kondisi yang dibuat dalam kodingan. Jika tidak ada *path variable* nya, maka *name* yang ditampilkan adalah "dengklek". Selain itu (jika ada *path variable*) maka yang ditampilkan adalah nama yang menjadi *path variable* tersebut.

Akses localhost:8080/greeting/chanek

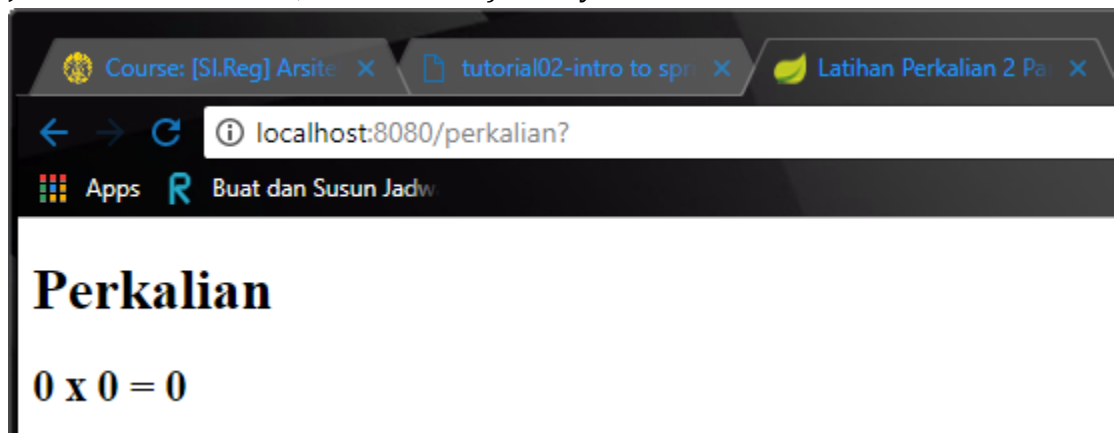
Apa hasilnya?



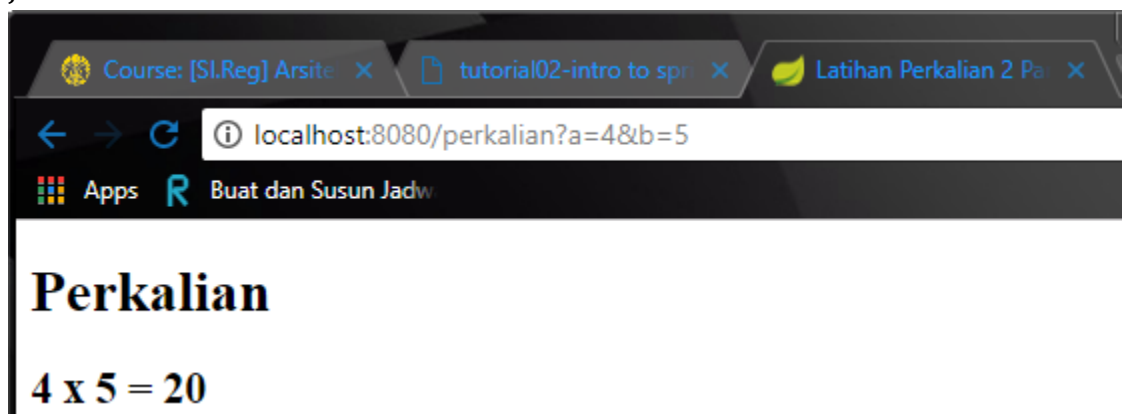
Karena ada *path variable*, maka nama nya akan di *get*.

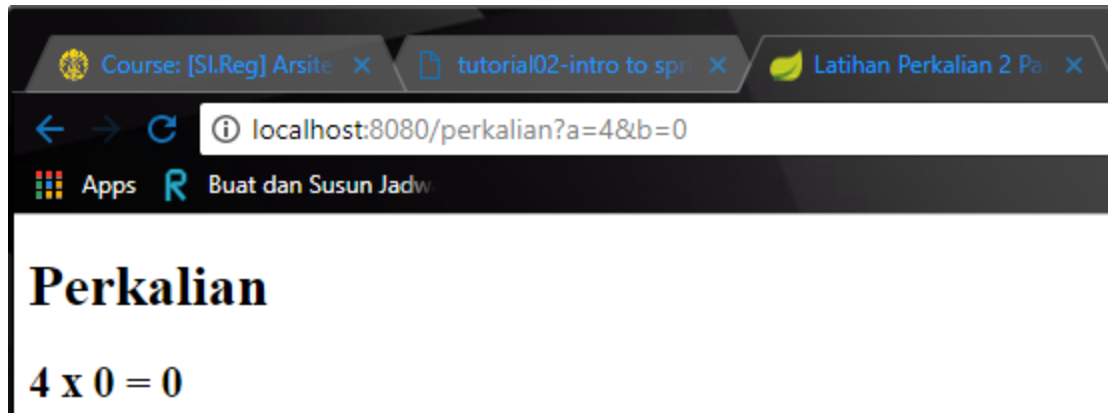
Latihan Perkalian

1. Jika a dan b tidak ada, maka nilai *default* nya 0



2. Jika a dan b ada





### **Write up**

Tutorial 2 ini membahas tentang *routing* dan *controller* dalam *Project Spring Boot*. Jadi, di *Spring Boot* akan selalu memakai *controller*. Bertugas sebagai *back-end*, dan file html sebagai *front-end*.

- anotasi *RequestMapping* digunakan untuk me-request berdasarkan url yang sedang dijalankan, yang kemudian akan menjalankan *method* di bawahnya. Jadi, harus sama anotasi yang di *request* dengan url yang sedang diakses.
- *return String* dalam *method* menandakan *page* html mana yang akan ditampilkan ketika halaman akan diakses.
- *RequestParam* digunakan ketika kita ingin mengambil nilai (*get*) dari url. Model harus ada, untuk menampilkan ke *front-end* atau html nya.
- *RequestParam* harus diberikan *defaultValue* ketika parameter tidak ada, menghindari error.
- *Path variable* dapat menggantikan misal url tanpa *path variable* harus mengikutsertakan *...?nama=chanek*. Ketika menggunakan *path variable* bisa langsung akses *id variable* nya menjadi *.../chanek*.