

LESSON LEARNED

Dalam tutorial kali ini, saya baru mengetahui bahkan di dalam html pun kita bisa melakukan conditional statement untuk menghasilkan output yang kita inginkan dengan menggunakan thymeleaf. Saya juga mempelajari penggunaan bootstrap dan js pada framework spring boot ini. Hal yang sudah pernah saya pelajari adalah penggunaan “template”, dalam case ini menggunakan fragment, dimana istilah yang saya kenali dari Laravel adalah include. Penggunaan fragment ini akan sangat bermanfaat untuk mengurangi redundansi dari kode yang dibuat di html, sehingga memudahkan developer dalam membangun websitenya dengan menggunakan spring boot. Hal terakhir yang saya pelajari adalah perlunya ada error handler di dalam suatu website, karena itu perlu di buat template yang berisi statement error pada html untuk nantinya ditampilkan jikalau user memasukan alamat routing yang tidak ada.

PERTANYAAN

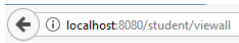
- **Pertanyaan 1:**

Value yang dihasilkan oleh `${iterationStatus.odd}` adalah mencetak semua class yang memiliki nilai yang ganjil.

- **Pertanyaan 2:**

Condition yang ada di dalam **th:if** dan **th:unless** memang sama, namun yang membedakannya adalah **If** dan **unless** itu sendiri, dimana **unless** adalah *negative counterpart* dari **if**. Misal jika mahasiswa memiliki gpa di atas 3.49, maka dia mendapatkan cumlaude. Pernyataan tersebut jika diungkapkan dalam unless, maka akan menjadi “siswa yang memiliki gpa di bawah 3.49”

Berikut adalah tampilannya:



No. 4

NPM = 12346

Name = jotaro joestar

GPA = 3.5

Cum Laude!

[Update Data](#)
[Delete Data](#)

Kuliah yang diambil

- SDA - 3 sks

No. 5

NPM = 12348

Name = josuke

GPA = 2.4

Sangat Memuaskan

[Update Data](#)
[Delete Data](#)

- Pertanyaan 3:

Terjadi error, karena kedua kondisi menunjuk pada output yang sama, karena unless diubah menjadi `th:unless="{student.gpa}<=3.48}` Yang berarti bahwa kondisi tersebut mencari mahasiswa yang nilainya di atas 3.48, dengan kata lain sama dengan kondisi if yang ada. Hal ini membuat terjadinya tabrakan antara kedua kondisi.



Whitelabel Error Page

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

Sat Oct 14 00:29:52 GMT+07:00 2017

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

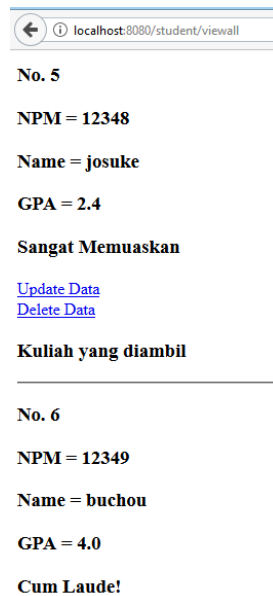
Exception parsing document: template="viewall", line 15 - column 38

- Pertanyaan 4:

Cara yang saya lakukan adalah:

```
<div th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}?'odd'">
  <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
  <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
  <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
  <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  <h3 th:text="${student.gpa}>3.49) ? 'Cum Laude!' : 'Sangat Memuaskan'"></h3>
  <a th:href="/student/update/' + ${student.npm}">Update Data</a><br/>
  <a th:href="/student/delete/' + ${student.npm}">Delete Data</a><br/>
```

Dengan tampilan yang sama:



← ⓘ localhost:8080/student/viewall

No. 5

NPM = 12348

Name = josuke

GPA = 2.4

Sangat Memuaskan

[Update Data](#)

[Delete Data](#)

Kuliah yang diambil

No. 6

NPM = 12349

Name = buchou

GPA = 4.0

Cum Laude!

- **Pertanyaan 5:**

Atribut tersebut berguna sebagai “include” yang bertugas untuk mengganti tag div tersebut menjadi tag div header atau footer yang berada pada alamat fragments/fragment, ini mirip sekali dengan include blade pada Laravel. Dengan ini kita bisa melakukan reuse pada div header dan footer untuk digunakan di banyak halaman selain index.html sehingga dapat memudahkan kita melakukan layouting.

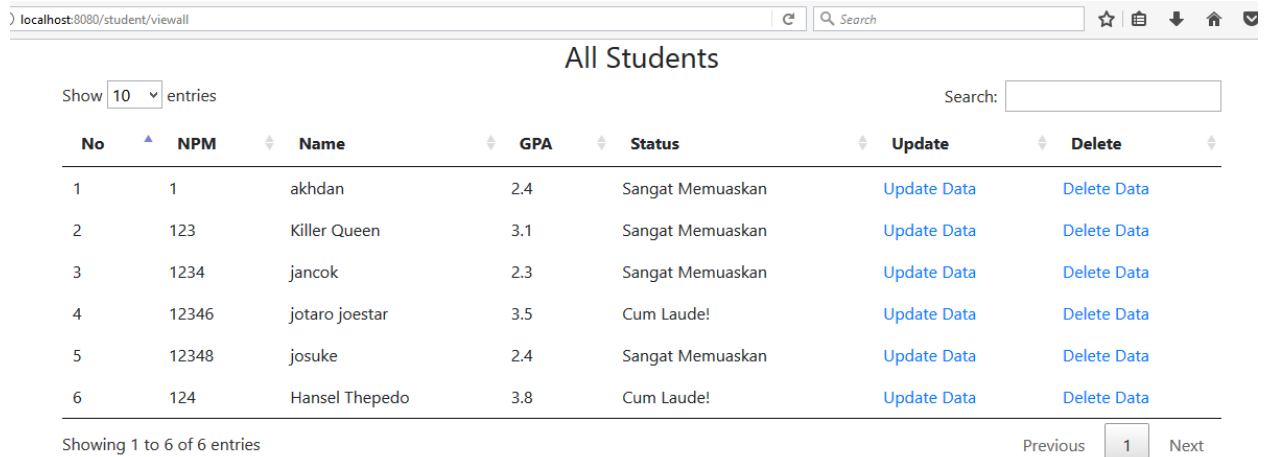
- **Pertanyaan 6:**

Error handling untuk *500: internal server error* bisa di handle jika dipisah dari html error handler *404: not found*.

LATIHAN

1. Implementasi DataTables

Tampilan:



No	NPM	Name	GPA	Status	Update	Delete
1	1	akhdan	2.4	Sangat Memuaskan	Update Data	Delete Data
2	123	Killer Queen	3.1	Sangat Memuaskan	Update Data	Delete Data
3	1234	jancok	2.3	Sangat Memuaskan	Update Data	Delete Data
4	12346	jotaro joestar	3.5	Cum Laude!	Update Data	Delete Data
5	12348	josuke	2.4	Sangat Memuaskan	Update Data	Delete Data
6	124	Hansel Thepedo	3.8	Cum Laude!	Update Data	Delete Data

2. Header Dinamis

Saya sudah dapat mengimplementasikan header dinamis dengan menggunakan fragment header. Ide yang saya dapatkan adalah dengan memanggil atribut model pada fragment header tersebut. Atribut model header ini berasal dari method controller, dimana di setiap method controller saya melakukan addAttribute ke model yang baru, kemudian menamainya "(page_title, XX)" dimana XX adalah halaman yang akan dikembalikan oleh method controller. Misalkan jika method controller akan mengarahkan ke halaman index, maka XX akan menjadi "Home". Untuk lebih lanjutnya bisa dilihat di kode yang telah saya buat.