

Pertanyaan 1: Apakah value yang dihasilkan dari `${iterationStatus.odd}`?

Jawab: Value yang dihasilkan adalah boolean apakah dia itu ganjil atau bukan

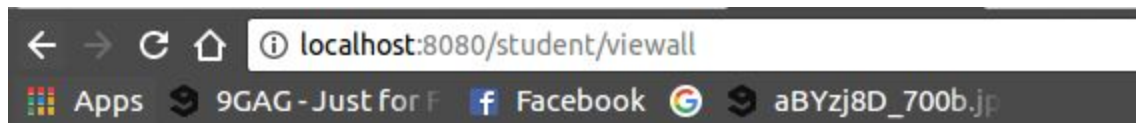
Pertanyaan 2: Mengapa condition di dalam `th:unless` sama dengan di dalam `th:if`? Jelaskan jawaban Anda!

Jawab:

Karena `unless` adalah negatif dari `if` jadi jika `gpa` tidak lebih besar sama dengan 3.49 maka teks akan ditampilkan.

Pertanyaan 3: Coba ubah condition pada `th:unless=${student.gpa}>=3.49}` menjadi `th:unless=${student.gpa}<=3.48}`, lalu run kembali aplikasi Anda dan mengakses halaman yang sama. Apakah terjadi error? Jika iya, kenapa? Jelaskan alasan Anda. Jika tidak error, apakah hasilnya sesuai?

Jawab: Error, karena `if` dan `unless`nya mengeluarkan hasil yang sama.



Whitelabel Error Page

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

Wed Oct 11 12:48:59 WIB 2017

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

Exception parsing document: template="viewall", line 35 - column 47

Pertanyaan 4: Tuliskan cara lain Anda dengan penulisan conditional expression yang berbeda untuk mendapatkan hasil yang sama seperti gambar di atas. Hint: Gunakan ternary operator.

Jawab:

```
<th th:text="${student.gpa}>=3.49}? 'Ya' : 'Tidak'></th>
```

Pertanyaan 5: Apa yang dimaksud dengan `th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file `index.html` yang Anda buat?

Jawab:

Artinya bagian itu di replace dengan fragment tersebut.

Pertanyaan 6: Apa handler dengan metode ini juga dapat berlaku bagi error lain seperti error 500: internal server error?

Jawab: Tidak bisa, karena itu hanya bisa untuk Not Found.

Latihan

1. DataTable

Untuk mengimplementasikan DataTable pertama-tama harus memanggil CDN dari DataTable, JQuery, dan Javascript dari DataTable

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<div th:replace="fragments/fragment :: title"></div>

<link rel="stylesheet" type="text/css" th:href="@{https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css}"></link>
<script th:src="@{https://ajax.googleapis.com/ajax/libs/jquery/3.1.0/jquery.min.js}"></script>
<script type="text/javascript" charset="utf-8" th:src="@{http://cdn.datatables.net/1.10.16/js/jquery.dataTables.js}"></script>
```

Setelah itu akan dipanggil fungsi dari DataTable dengan cara

```
<script>
$(document).ready( function () {
    $('#example').DataTable();
} );
</script>
```

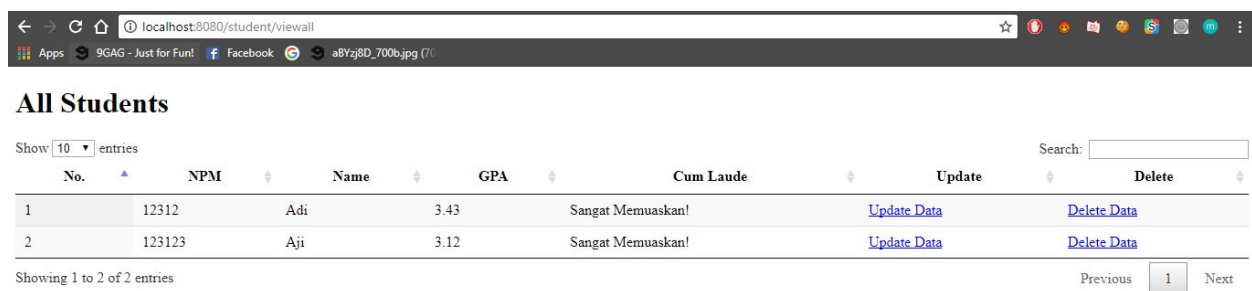
Id tersebut disesuaikan dengan id yang ada pada table, dalam kasus ini id table adalah example

```
<table id="example" class="display">
<thead>
<tr>
<th>No. </th>
<th>NPM </th>
<th>Name </th>
<th>GPA </th>
<th>Cum Laude </th>
<th>Update </th>
<th>Delete</th>
```

Setelah itu akan dimasukkan kode untuk mengenerate data yang akan ditampilkan oleh table seperti berikut

```
<table id="example" class="display">
  <thead>
    <tr>
      <th>No. </th>
      <th>NPM </th>
      <th>Name </th>
      <th>GPA </th>
      <th>Cum Laude </th>
      <th>Update </th>
      <th>Delete</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="student, iterationStatus: ${students}">
      <td th:text="${iterationStatus.count}">No. </td>
      <td th:text="${student.npm}">Student NPM</td>
      <td th:text="${student.name}">Student Name</td>
      <td th:text="${student.gpa}">Student GPA</td>
      <td th:text="${student.gpa}>=3.49}? 'Cum Laude!' : 'Sangat Memuaskan!'">Cum Laude</td>
      <td><a th:href = "/student/update/" + ${student.npm}" > Update Data </a></td>
      <td><a th:href = "/student/delete/" + ${student.npm}" > Delete Data </a></td>
    </tr>
  </tbody>
</table>
```

Testing:



No.	NPM	Name	GPA	Cum Laude	Update	Delete
1	12312	Adi	3.43	Sangat Memuaskan!	Update Data	Delete Data
2	123123	Aji	3.12	Sangat Memuaskan!	Update Data	Delete Data

2. Fragment

Untuk membuat *fragment* yang dinamis, langkah-langkahnya adalah sebagai berikut:

- Mengganti kode title pada fragment.html sehingga bisa menerima input model title dari controller.



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <div th:fragment="title">
5 <title th:text="${title}">Title</title>
6 </div>
7 </head>
```

- Selanjutnya, pada controller akan ditambahkan atribut title yang berisi String yang nantinya akan ditampilkan di setiap view

```

24
25 @RequestMapping("/")
26 public String index (Model model)
27 {
28     model.addAttribute("title", "Index");
29     return "index";
30 }
31
32
33 @RequestMapping("/student/add")
34 public String add (Model model)
35 {
36     model.addAttribute("title", "Add Student");
37     return "form-add";
38 }
39
40
41 @RequestMapping("/student/add/submit")
42 public String addSubmit (
43     @RequestParam(value = "npm", required = false) String npm,
44     @RequestParam(value = "name", required = false) String name,
45     @RequestParam(value = "gpa", required = false) double gpa,
46     Model model)

```

- Selanjutnya, pada setiap view akan ditambahkan div untuk mereplace title dengan title yang sesuai dimasukkan pada controller dengan kode seperti berikut:

```

<head>
<div th:replace="fragments/fragment :: title"></div>

```

LESSON LEARNED

Pada tutorial kali ini, saya belajar tentang For Looping dan Conditional pada Thymeleaf. Untuk menggunakan for loop kita dapat menggunakan th:each, sedangkan untuk conditional kita dapat menggunakan th:if.

Selain itu, saya juga belajar tentang DataTable dan Fragment. DataTable berguna untuk memudahkan kita mengambil data pada table sekaligus juga memperbagus tampilan dari table. Sementara, fragment berguna untuk memudahkan layouting serta membuat tampilan menjadi lebih dinamis.