

TUTORIAL 6

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

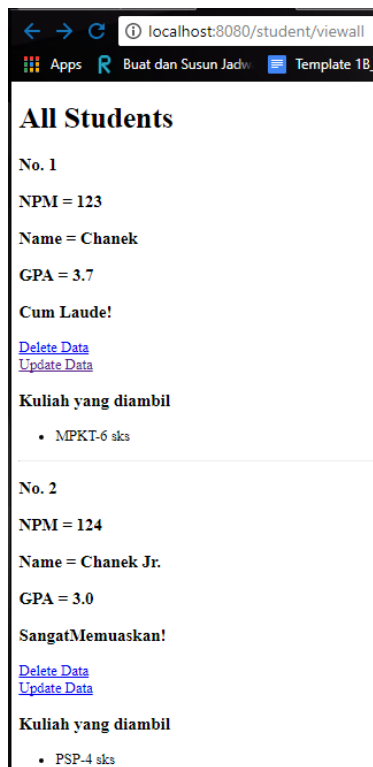
Ketika *true*, hasilnya adalah *evaluation* dari *expression* yang diberikan. The syntax *th:class* is converted into *class* after template processing which is used to load CSS class.

Penggunaan Conditional Expression pada Thymeleaf

1. Tambahkan kode berikut pada viewall.html

```
15 <h3 th:if="${student.gpa}>=3.49}">Cum Laude!</h3>
16 <h3 th:unless="${student.gpa}>=3.49}">SangatMemuaskan!</h3>
```

2. Jalankan kembali program localhost:8080/student/viewall



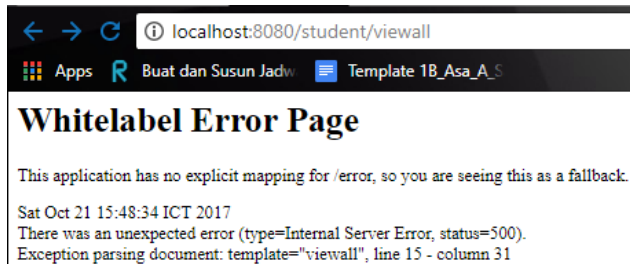
Pertanyaan 2: Mengapa *condition* di dalam *th:unless* sama dengan di dalam *th:if*? Jelaskan jawaban Anda.

Kedua kondisi tersebut sama, karena kondisi di dalam *th:unless* merupakan kondisi negatif atau jika di *java* adalah bagian *else* dari kondisi *th:if*.

Jadi `student.gpa < 3.49` akan masuk ke dalam kondisi *th:unless*

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?

Hasilnya *error*. Berikut *screenshot* nya:



Kondisi seperti itu tidak dapat diterapkan.

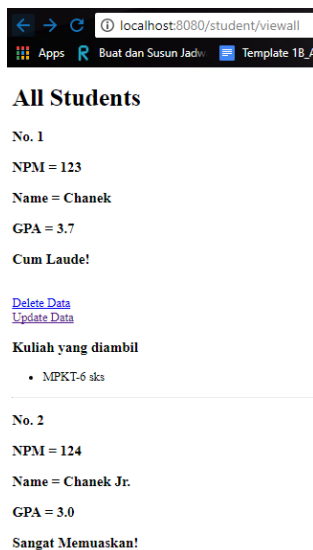
Karena kondisi yang diberikan berbeda dengan standar. Harus ada `th:if` kemudian `th:unless`. Nilai dari atribut `th:unless` juga berasosiasi dengan elemen tipe *null*, sehingga tidak boleh mengandung karakter kurang dari (<).

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.

Seperti tutorial sebelumnya, saya menggunakan cara seperti berikut:

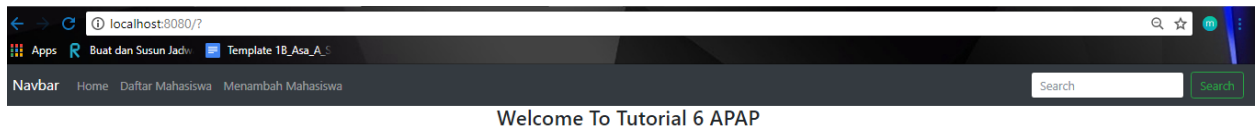
```
<h3 th:text="${student.gpa} >= 3.49 ? 'Cum Laude!' : 'Sangat Memuaskan!'"></h3>
```

Jika gpa seorang *student* ≥ 3.49 maka cetak *Cum Laude!* selain itu ($\text{gpa} < 3.49$) cetak Sangat Memuaskan



Static file

1. Unduh file *bootstrap*
 2. Ubah kode pada index.html
 3. Jalankan kembali program
- Berikut *screenshot* nya:



Fragment

Berikut ini adalah cara penggunaan fragment:

1. Buat sebuah folder dalam folder templates dengan nama fragments
2. Buat sebuah halaman HTML bernama fragment.html dan ubah isi index.html
3. Jalankan kembali program. Berikut *screenshot* nya



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

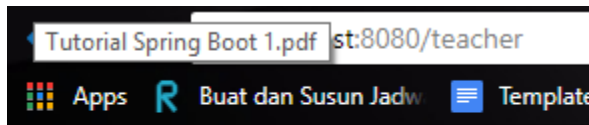
`th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file index.html yakni semua bagian mulai dari `<div>` sampai `</div>` pada file fragment.html memiliki *header* dan *footer* yang akan ditampilkan melalui file index.html

Error 404 Not Found Handler

Berikut ini adalah cara untuk membuat *error 404 not found handler*:

1. Buat sebuah folder dalam folder *templates* dengan nama *error*
2. Buat sebuah halaman HTML bernama 404.html

3. Akses suatu halaman yang tidak tersedia dari aplikasi Anda. Misalnya halaman dengan mapping /teacher. Contoh tampilan yang keluar adalah sebagai berikut



Halaman tidak ditemukan

Pertanyaan 6: Apa *handler* dengan metode ini juga dapat berlaku bagi *error* lain seperti *error 500: internal server error*? Ya, tentu dapat berlaku.

Latihan

1. Pada tutorial, anda sudah mencoba untuk memperlihatkan apakah student tersebut cum laude atau tidak sampai ke bagian view. Pada latihan kali ini, anda diminta untuk mengubah view tersebut menggunakan DataTables. DataTables adalah plugin jquery library, yang dapat membuat sebuah tabel lebih interaktif dan lebih baik (<https://datatables.net/>). Oleh karena itu, gunakan data tabel untuk mengubah tampilan localhost:8080/student/viewall. Dengan kolom-kolom yang harus ada sebagai berikut: 1. No 2. NPM 3. Name 4. GPA 5. Cum laude 6. Delete (jika diklik, maka dapat menghapus student,) 7. Update (jika diklik, yang dapat mengupdate student) Anda tidak harus menggunakan datatables versi server side (https://datatables.net/examples/server_side/), yang mana lebih baik digunakan untuk data yang sangat banyak. Walaupun demikian, jika anda anda ingin mengimplementasi hal ini sangat diperbolehkan, tetapi tidak ada penambahan nilai untuk hal tersebut.

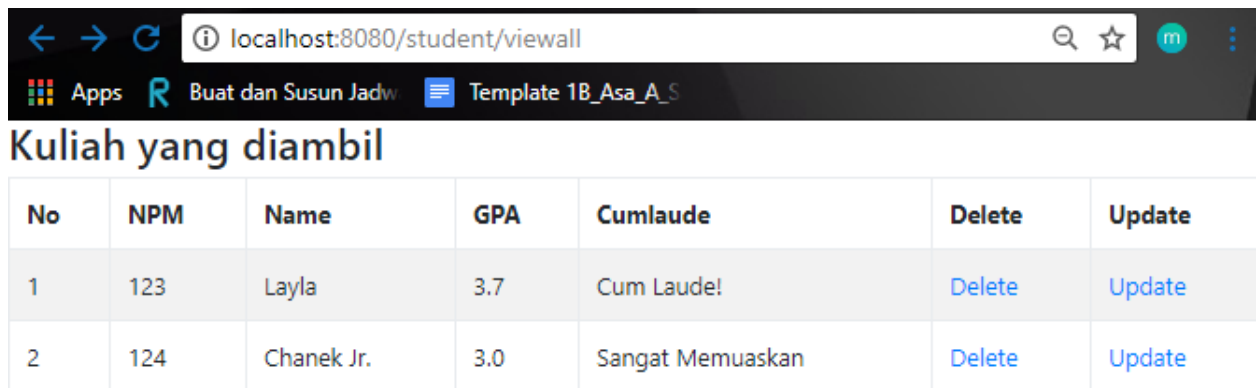
Ubah kode pada file viewall.html menjadi seperti berikut:

```

viewall.html
APAPTutorial6/src/main/resources/templates/viewall.html
2 <html xmlns:th="http://www.thymeleaf.org">
3
4 <head>
5 <title th:replace="fragments/fragment :: head"></title>
6 <link rel="stylesheet" href="/js/jquery.dataTables.min.js"/>
7 <link rel="stylesheet" href="/css/jquery.dataTables.min.css"/>
8 <link rel="stylesheet" href="/js/dataTables.bootstrap.min.js"/>
9 <link rel="stylesheet" href="/css/bootstrap.min.css"/>
10 <script>
11 $(document).ready(function() {
12     $('#students').DataTable();
13 });
14 </script>
15
16 </head>
17 <body>
18 <h3>Kuliah yang diambil</h3>
19
20 <table id="students" class="table table-striped table-bordered" cellspacing="0" width="100%">
21 <thead>
22 <tr>
23 <th>No</th>
24 <th>NPM</th>
25 <th>Name</th>
26 <th>GPA</th>
27 <th>Cumlaude</th>
28 <th>Delete</th>
29 <th>Update</th>
30 </tr>
31 </thead>
32 <tbody>
33 <tr th:each="student, iterationStatus : ${students}">
34 <td th:text="${iterationStatus.count}"></td>
35
36 <td th:text="${student.npm}"></td>
37 <td th:text="${student.name}"></td>
38 <td th:text="${student.gpa}"></td>
39 <td th:text="${student.gpa >= 3.49 ? 'Cum Laude!' : 'Sangat Memuaskan'}"></td>
40 <td><a th:href="/student/delete/" + ${student.npm}" >Delete</a></td>
41 <td><a th:href="/student/update/" + ${student.npm}" >Update</a></td>
42 </tr>
43 </tbody>
44 </table>
45 <hr/>
46 </body>
47 </html>

```

Jalankan kembali program:



No	NPM	Name	GPA	Cumlaude	Delete	Update
1	123	Layla	3.7	Cum Laude!	Delete	Update
2	124	Chanek Jr.	3.0	Sangat Memuaskan	Delete	Update

Referensi : https://datatables.net/examples/server_side/

2. Pada tutorial, Anda sudah mencoba membuat fragment header dan footer. Tetapi, header dan footer pada tutorial tersebut tidak dinamis. Data HTML pada header atau footer akan sama di setiap view yang memakai fragment tersebut. Pada latihan kali ini, Anda diminta untuk membuat sebuah fragment header yang dinamis. Dimana, aspek dinamis tersebut terdapat pada title yang diletakkan pada fragment header, bukan setiap page HTML yang ada. Lebih jauh, title tersebut akan berubah-ubah sesuai dengan page yang ada. Silahkan implementasi dengan kebutuhan tersebut, dan jika sudah berhasil, pastikan header tersebut dimuat pada setiap view yang ada di project ini.

Tambahkan kode berikut pada dile fragment.html

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/'}">Index</title>
5 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/course/view/${id_course}'}">View Course by Id</title>
6 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/delete'}">Delete</title>
7 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/delete/${npm}'}">Delete Student by NPM</title>
8 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/add'}">Form Add Student</title>
9 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/add/submit'}">Add Student</title>
10 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/view'}">View Student</title>
11 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/view/${npm}'}">View Student by NPM</title>
12 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/viewall'}">View All Students</title>
13 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/update/${npm}'}">Update Student by NPM</title>
14 <title th:fragment="head" th:if="${httpServletRequest.requestURI == '/student/update/submit'}">Update</title>
15 </head>
16 <body>
17 <div th:fragment="header">
18 <ul>
19 <li><a href="/">Home</a></li>
20 <li><a href="/student/viewall">Daftar Mahasiswa</a></li>
21 <li><a href="/student/add">Menambah Mahasiswa</a></li>
22 </ul>
23 </div>
24 <div th:fragment="footer">
25 <hr/>
26 <h4>Mata Kuliah APAP</h4>
27 </div>
28 </body>
29 </html>
30

```

Lalu, selaraskan kode pada setiap file html lainnya. Contoh pada file view course

```
1 |<!DOCTYPE html>
2 |<html xmlns:th="http://www.thymeleaf.org">
3 |   <head>
4 |     <title th:replace="fragments/fragment :: head"></title>
```

Write-Up

Tutorial 6 ini membahas tentang bagaimana menggunakan bootstrap serta membuat header dan footer dalam file fragments. Agak rumit, namun berhasil selesai...