

Iman Alfathan Yudhanto

1406623524

APAP-A

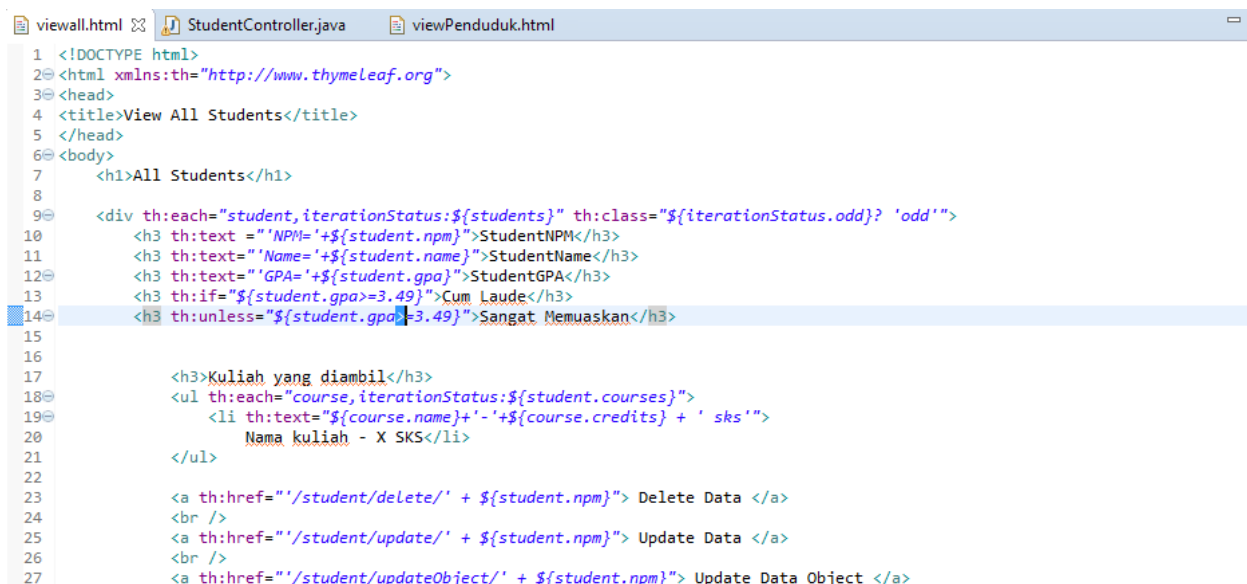
Write-up tutorial 6

Pertanyaan1: Apakah value yang dihasilkan dari `${iterationStatus.odd}`?

Fungsi dari method tersebut adalah me-return Boolean berdasarkan iteration Status. Jadi ketika melakukan looping, maka hasil yang dtampilkan dalam kondisi iterationStatusnya bernilai ganjil.

1. Tambahkan code baris berikut ini pada file **viewall.html**.

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



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View All Students</title>
5 </head>
6 <body>
7 <h1>All Students</h1>
8
9 <div th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}? 'odd'">
10 <h3 th:text="'NPM=' + ${student.npm}">StudentNPM</h3>
11 <h3 th:text="'Name=' + ${student.name}">StudentName</h3>
12 <h3 th:text="'GPA=' + ${student.gpa}">StudentGPA</h3>
13 <h3 th:if="${student.gpa}>=3.49}">Cum Laude</h3>
14 <h3 th:unless="${student.gpa}>=3.49}">Sangat Memuaskan</h3>
15
16 <h3>Kuliah yang diambil</h3>
17 <ul th:each="course, iterationStatus: ${student.courses}">
18 <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
19 Nama kuliah - X SKS</li>
20 </ul>
21
22 <a th:href="'/student/delete/' + ${student.npm}"> Delete Data </a>
23 <br />
24 <a th:href="'/student/update/' + ${student.npm}"> Update Data </a>
25 <br />
26 <a th:href="'/student/updateObject/' + ${student.npm}"> Update Data Object </a>
27
```

2. Coba run aplikasi Anda, lalu akses **localhost:8080/student/viewall**. Tampilan akan seperti gambar di bawah ini.

View All Students - Mozilla Firefox

Sistem Akun UL... Course: [SI.Reg... tutorial-06-thyme... View All Stu... Thymeleaf Con... iterationstatus... Tutorial: Using ... Counters in Lo... java - How to d... New Tab

localhost:8080/student/viewall

All Students

NPM=1

Name=Chanek Jr.

GPA=3.41

Sangat Memuaskan

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

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

NPM=11

Name=Dudung

GPA=3.0

Sangat Memuaskan

Find in page Highlight All Match Case

EN 10:53 AM

View All Students - Mozilla Firefox

localhost / 127... Course: [SI.Reg... tutorial-06-thyme... View All Stu... Thymeleaf Con... iterationstatus... Tutorial: Using ... Counters in Lo... java - How to d... New Tab

localhost:8080/student/viewall

Name=balrog

GPA=2.3

Sangat Memuaskan

Kuliah yang diambil

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

NPM=29

Name=Munaroh

GPA=3.55

Cum Laude

Kuliah yang diambil

- MPKT-6 sks
- PSP-4 sks

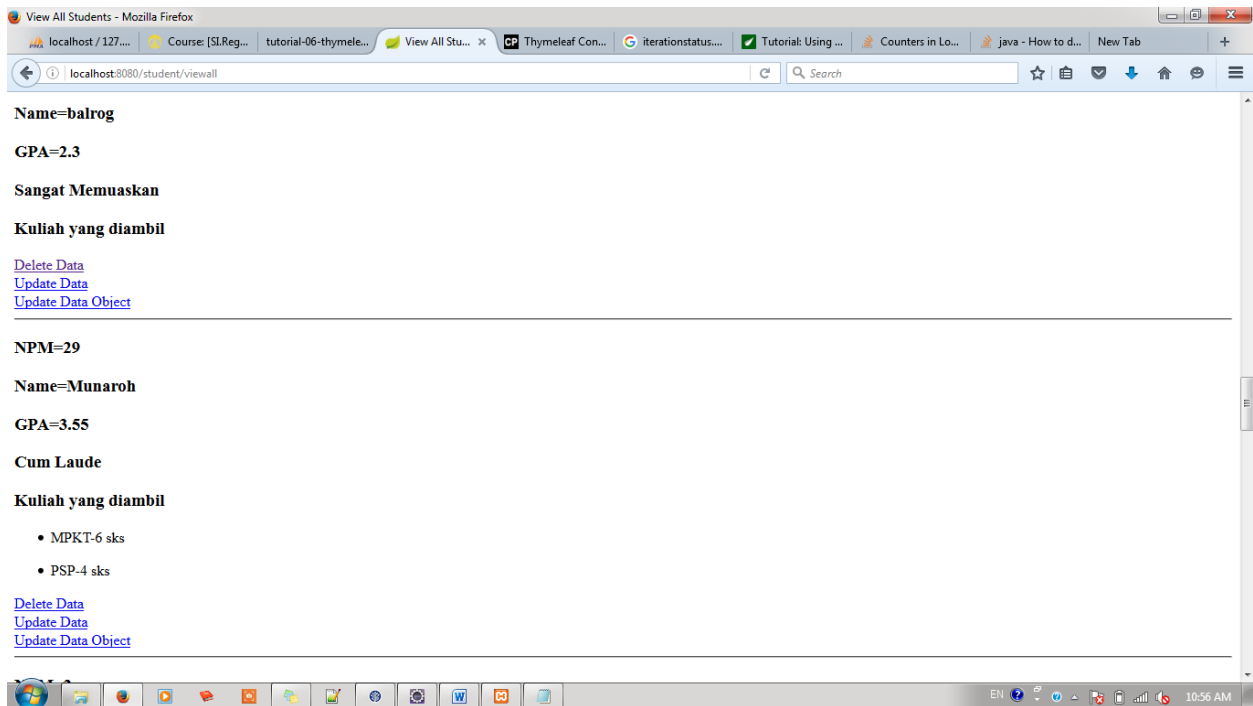
[Delete Data](#)
[Update Data](#)
[Update Data Object](#)

EN 10:56 AM

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

Karena th:unless berfungsi untuk mengembalikan negasi dari th:if. Sehingga nilai dari th:unless sama dengan nilai th:if.

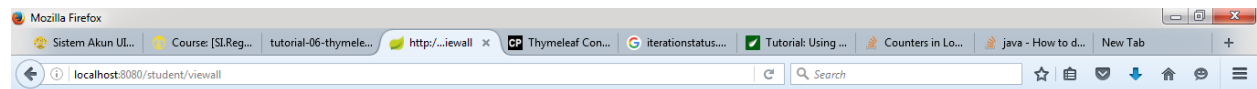
Hasil:



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

Terjadi error pada program. Penyebabnya adalah *condition* dari th:unless yang mengharuskan sama dengan *condition* pada th:if. Th:unless mengembalikan nilai negasi dari th:if, maka *condition* harus sama.

Hasil:



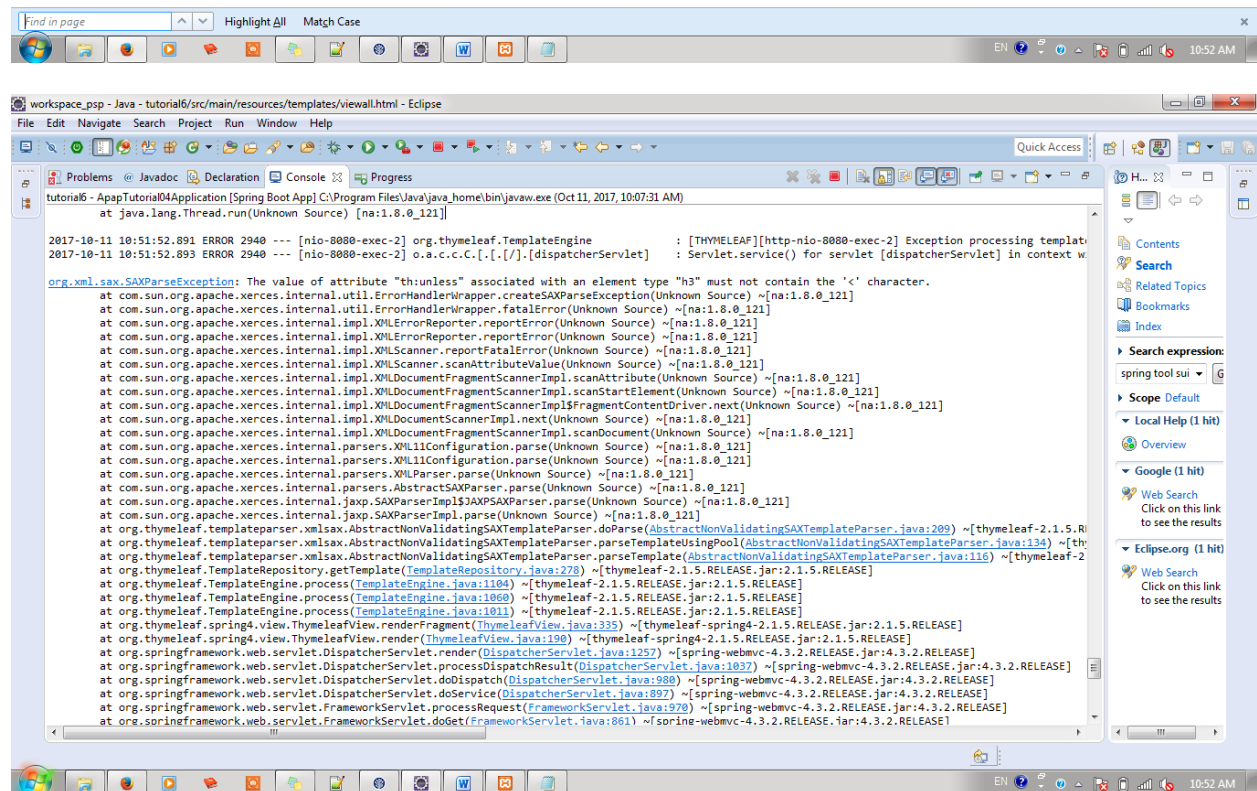
Whitelabel Error Page

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

Wed Oct 11 10:51:52 ICT 2017

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

Exception parsing document: template="viewall", line 14 - column 31



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.

Untuk melakukan penulisan *conditional expression* yang berbeda adalah dengan menggunakan th:switch dan th:case. Th:switch dan Th:case berguna jika kita memiliki banyak *condition* dengan hasil yang berbeda.

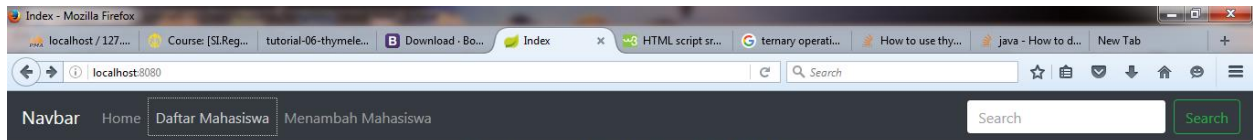
Berikut contoh penggunaan th:case yang penulis gunakan:

```
58 <tr>
59   <td>Kewarganegaraan</td>
60   <td th:switch="{penduduk.isWNI}">
61     <div th:case="1">WNI</div>
62     <div th:case="0">Warga Asing</div>
63   </td>
64 </tr>
65 <tr>
66   <td>Status Kematian</td>
67   <td th:switch="{penduduk.isWafat}">
68     <div th:case="1">Wafat</div>
69     <div th:case="0">Hidup</div>
70   </td>
71 </tr>
```

Pada program di atas, th:switch berfungsi sebagai input yang akan diterima. Th:case berfungsi sebagai *conditional expression* dari th:switch. Jika nilai th:switch sesuai dengan th:case yang dituju, maka akan keluar hasil sesuai dengan th:case.

Berikut ini adalah cara pemakaian bootstrap:

1. Unduh bootstrap pada <http://getbootstrap.com/docs/4.0/getting-started/download/>
2. *Extract folder* zip yang Anda unduh ke dalam folder **static** yang terdapat pada folder **src/main/resources**.
3. Buat halaman **index.html** menjadi *source code* dibawah ini
4. Akses halaman index dari aplikasi Anda. Contoh tampilan adalah sebagai berikut.



Welcome To Tutorial 6 APAP



Berikut ini adalah cara penggunaan *fragment*:

1. Buat sebuah *folder* dalam *folder templates* dengan nama **fragments**
2. Buat sebuah halaman HTML bernama **fragment.html** yang berisi *source code* seperti dibawah ini. File HTML tersebut dibuat pada folder **fragments** yang baru Anda buat sebelumnya
3. Pada *source code* diatas, **th:fragment** merepresentasikan nama *fragment* dari bagian yang Anda ingin jadikan sebuah *fragment*.
4. Buat halaman **index.html** menjadi *source code* dibawah ini
5. Coba jalankan halaman index yang Anda buat. Contoh tampilan yang keluar adalah sebagai berikut



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

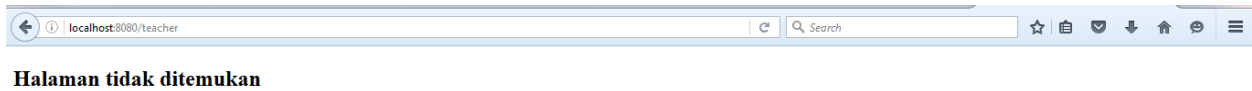
`Th:replace` berfungsi untuk menggantikan bagian yang ditandai dengan sebuah fragment yang telah dibuat. Selain itu, bagian yang ditandai dapat diganti dengan nama komponen pada halaman fragment. Pada contoh/tutorial ditunjukkan bahwa ada bagian yang diganti dengan nama komponen pada halaman fragment.

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** yang berisi *source code* seperti dibawah ini. File HTML tersebut dibuat pada folder **error** yang baru Anda buat sebelumnya

```
error.html
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>404 Not found</title>
5 </head>
6 <body>
7 <h2>Halaman tidak ditemukan</h2>
8 </body>
9 </html>
```

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



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

Tidak bisa di-*handle* karena error 500 merupakan error pada konfigurasi ataupun jaringan. Tidak ada hubungan dengan mapping pada controller.

Hal yang dipelajari:

Hal yang dipelajari di tutorial ini adalah menambahkan javascript dan css ke dalam tampilan *front-end* dan penggunaan static file. Cara meng-*import* css dan javascript ke bagian *front-end* hampir sama seperti yang telah penulis pelajari ketika mengambil mata kuliah PPW. Yang membedakannya adalah adanya beberapa syntax pada thymeleaf yang baru diketahui penulis. Penulis juga mempelajari penggunaan static file. File yang bersifat statis seperti gambar, javascript, dan css dapat disimpan di suatu folder static dan juga dapat diakses.

Hal kedua yang dipelajari adalah memodifikasi ekspresi iterationStatus. Di tutorial ini, penulis diajarkan bagaimana memodifikasi iteration status. Penulis dapat melakukan iterasi dengan membedakan index yang bernilai ganjil/genap.

Hal terakhir yang dipelajari di tutorial ini adalah terdapat beberapa syntax seperti `th:switch` dan `th:case`, `th:if` yang sepaket dengan `th:unless`, `th:fragment`, dan `th:replace`. Fungsi `th:if` adalah untuk memberikan sebuah condition pada front-end. Jika condition sesuai, maka akan muncul tampilan sesuai condition. Sedangkan `th:unless` memberikan hasil negasi dari condition. Jadi yang ditampilkan adalah negasinya. `Fragment` merupakan potongan-potongan kode yang sengaja dipisahkan agar nantiya dapat digunakan kembali di halaman html yang lain. `Th:fragment` berguna untuk menentukan bagian, biasanya bagian tersebut akan terus digunakan pada suatu program. Tiap fragment diberikan sebuah nama yang merepresentasikan wujud dan perannya. Sementara `th:replace` berfungsi untuk mengganti suatu bagian dengan bagian yang ada di bagian fragment.

Sumber:

- <http://www.thymeleaf.org/doc/tutorials/2.1/usingthymeleaf.html#iteration>
- <https://www.concretepage.com/thymeleaf/thymeleaf-conditionals-if-unless-and-switch-statements-example>
- <https://stackoverflow.com/questions/15904756/counters-in-loops-in-thymeleaf>
-