Bintang Glenn J

1506757535

Hal yang dipelajari

Pada tutorial kali ini saya belajar lebih jauh terkait thymeleaf. Saya menjadi lebih mengerti bagaimana caranya melakukan *looping*, *conditional expression*, mengelola *static file*, menggunakan *fragment* hingga melakukan *error handling*. Saya juga kembali mengingat penggunaan DataTables untuk mengelola *pagination.*

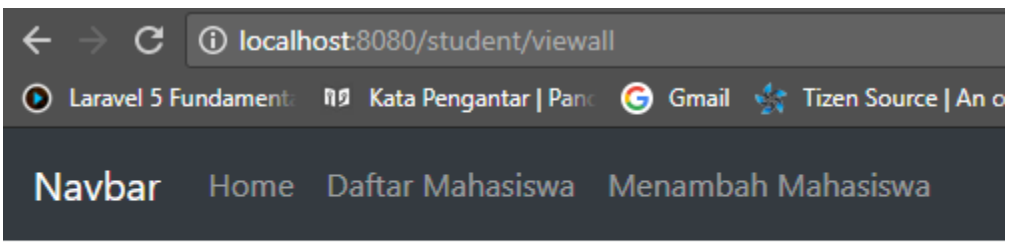
Pertanyaan

1. Value yang dihasilkan oleh ${iterationStatus.odd} adalah true/false, yang berarti tipe datanya adalah boolean.

Hasil tersebut saya dapatkan dengan mencetak hasil ekspresi itu sendiri, seperti berikut:

```
<div th:each="student,iterationStatus: ${students}">
    <h3 th:text="'No. ' + ${iterationStatus.odd}">No. 1</h3>
```

2. Ya, kondisi di dalam th:unless dan th:if memang sama, namun keduanya memiliki fungsi yang berbeda. th:unless akan menampilkan teks jika kondisi yang ada di dalamnya bernilai false, sedangkan th:if akan menampilkan teks jika kondisi yang ada di dalamnya bernilai true.
3. Ya, terjadi error seperti berikut:



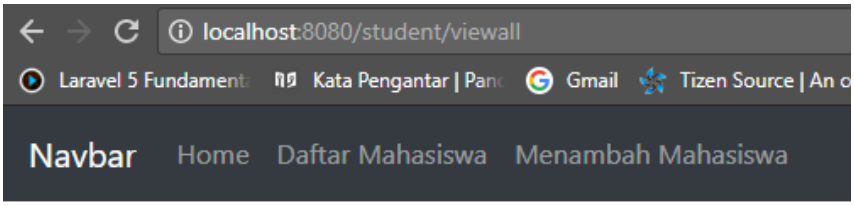
Error tersebut terjadi karena kondisi di dalamnya tidak boleh mengandung karakter “<” seperti yang terlihat jelas di dalam *console.*

```
sociated with an element type "h3" must not contain the '<' character.
```

4. Seperti petunjuk yang diberikan, cukup gunakan *ternary operator* dalam th:text untuk mendapatkan hasil yang sama.

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

5. th:replace berfungsi mengganti isi dari tag tempatnya berada dengan fragment yang diberikan. fragments/fragment menunjukkan lokasi file yang berisi fragment tersebut dan header serta footer menunjukkan nama dari fragment yang akan digunakan.
6. Ya, metode seperti ini juga dapat digunakan untuk melakukan *error handling* pada *error* yang lain. Saya telah mencobanya dengan membuat *error handling* bagi *error* 500 *internal service error*.



Internal Server Error

Latihan Menggunakan DataTables

Pertama, saya menambahkan tag table dengan id student_table:

```
<table id="student_table" class="display">
```

Kemudian saya menambahkan *table head* sesuai dengan kolom yang diminta:

```
<thead>
    <tr>
        <th>No</th>
        <th>NPM</th>
        <th>Name</th>
        <th>GPA</th>
        <th>Cum Laude</th>
        <th>Delete</th>
        <th>Update</th>
    </tr>
</thead>
```

Pada bagian *table body*, saya melakukan *looping* untuk setiap student dalam list students:

```
<tbody>
    <tr th:each="student,iterationStatus: ${students}">
        <td th:text="${iterationStatus.count}">No. 1</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!'">Graduation Status</td>
        <td> <a th:href="'/student/delete/' + ${student.npm}">Delete Data</a></td>
        <td> <a th:href="'/student/update/' + ${student.npm}">Update Data</a></td>
    </tr>
</tbody>
```

Pada setiap kolom dalam baris akan berisi sesuai dengan info yang diminta. Saya kemudian menambahkan hubungan ke DataTables dengan menambahkan link ke css dari DataTables dan script ke javascript dari DataTables.

```
<link rel="stylesheet" type="text/css" href="/DataTables/datatables.css"/>
<script type="text/javascript" charset="utf8" src="/DataTables/datatables.js"></script>
```

Terakhir, saya menambahkan script untuk memanggil DataTables pada tabel sebelumnya:

```
<script type="text/javascript">
    $(document).ready( function () {
        $('#student_table').DataTable();
    } );
</script>
```

Latihan Membuat Fragment Dinamis

Pertama-tama saya cukup kesulitan karena mengira saya menggunakan thymeleaf 3 sedangkan ternyata thymeleaf yang ada pada *project* merupakan versi 2.1.5. Untuk bisa mengganti *title* pada setiap halaman, saya membuat fragment *header* pada tag *head* dengan parameter *title*:

```
<head th:fragment="header(title)">
    <title th:text="${title}">Page Title</title>
    <link rel="stylesheet" href="/css/bootstrap.min.css" />
    <link rel="stylesheet" type="text/css" href="/DataTables/datatables.css"/>
    <script type="text/javascript" charset="utf8" src="/DataTables/datatables.js"></script>
</head>
```

Saya menambahkan DataTables juga karena tidak menemukan cara untuk menambahkan link melalui parameter di thymeleaf 2.1 seperti pada thymeleaf 3. Selanjutnya tag *title* pada fragment saya berikan th:text dengan isi ${title} yang akan diganti sesuai dengan *title* yang diberikan. Terakhir, saya hanya mengubah setiap tag *head* dengan menambahkan th:include untuk mengambil fragment *header* yang telah didefinisikan. Setiap halaman hanya perlu mengisikan parameter yang berbeda untuk *title*.

```
<head th:include="fragments/fragment :: header(title='View All Students')">
</head>
```