Kevin Noel Anugrah
1506689370

Thymeleaf – Presentation Layer

Pada tutorial kali ini diajarkan bagaimana menggunakan Thymeleaf dengan looping dan conditional expressions, static files, fragments, dan error pages.

I. Looping

Tutorial sebelumnya sudah memperkenalkan looping tetapi penjelasan lebih lanjut ada pada tutorial ini. Looping pada thymeleaf dilakukan melalui atribut th:each dengan menspesifikasikan iteration variable, iteration status, dan iterated variable.

```
th:each="iterationVariable,iterationStatus: iteratedVariable"
```

Jawaban 1: Value yang dihasilkan dari `${iterationStatus.odd}` adalah boolean yang bernilai true jika iteration index (bukan count) bernilai ganjil.

II. Conditional

Conditional expression pada thymeleaf dilakukan melalui atribut th:if dan th:unless yang menerima variable expression yang memiliki nilai boolean. Jika th:if bernilai true, maka children dari element tersebut akan ditampilkan. Sedangkan th:unless menampilkan childrennya jika bernilai false. Variable expression dapat berisi operasi boolean misalnya `${2<1}`.

Jawaban 2: Condition dalam th:unless sama dengan th:if karena keduanya bekerja sebagai if-else statement, di mana th:if menangkap statement bernilai true sedangkan th:unless menangkap statement bernilai false.

Jawaban 3: Jika th:if tidak memiliki condition yang sama dengan th:unless pasangannya, atau bahkan tidak ada pasangannya, tidak masalah karena keduanya tidak merupakan satu kesatuan expression. Jika tidak sama, bisa saja ada statement yang tidak ditangkap oleh pasangan expression karena probabilitas keduanya tidak komplemen.

Jawaban 4: Selain menggunakan th:if dan th:unless, dapat dilakukan ternary operator untuk mengevaluasi statement dan menampilkan text yang sesuai:

```
<h1 th:text="${student.gpa >= 3.49}?'Cum laude!' : 'Sangat Memuaskan!'"></h1>
```

III. Static File

File css , js, dan resources lainnya yang bersifat static dapat diakses dalam folder /resources/static.

IV. Fragment

Untuk memasukkan potongan halaman lainnya ke halaman yang ditampilkan dapat digunakan fitur fragment. Element yang dijadikan potongan akan memiliki atribut th:fragment, yang dapat diselipkan pada halaman lainnya pada elemen dengan atribut th:replace.

Jawaban 5: `th:replace = "fragments/fragment :: header"` berarti mengganti elemen yang memiliki atribut itu sendiri dengan elemen beserta children dengan atribut th:fragment bernilai "header" pada template fragment.html di folder fragments. `th:replace ="fragments/fragment :: footer"` berarti mengganti elemen yang memiliki atribut itu sendiri dengan elemen beserta children dengan atribut th:fragment bernilai "footer" pada template fragment.html di folder fragments.

V. Error Handler

Untuk menghandle berbagai http status code dapat dibuat halaman yang memiliki nama yang sama dengan status codenya pada /templates/error, misalnya 404.html.

Jawaban 6: Untuk menghandle error lainnya, misalkan 500 internal server error, dapat dibuat /templates/error/500.html.

VI. Datatables

Untuk mengubah tampilan viewall dengan datatables dapat diunduh pluginnya dan diletakkan di folder static, lalu ubah viewall.html menjadi:

```
<table id="table" class="display">
<thead>
        <tr>
                <th>No</th>
                <th>NPM</th>
                <th>Name</th>
                <th>GPA</th>
                <th>Cum laude</th>
                <th></th>
                <th></th>
        </tr>
</thead>
<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="${student.gpa}">Student GPA</td>
                <td th:if="${student.gpa >= 3.49}">Ya</td>
                <td th:unless = "${student.gpa>=3.49}">Tidak</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>
```

Lalu tambahkan script untuk menginisiasi datatable sesuai petunjuk di websitenya.

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

VII. Dynamic Header Fragment

Untuk membuat tampilan header paling tidak berbeda antara template – template yang menggunakannya, dapat dimasukkan expression yang mengambil url dan menampilkannya sebagai judul besar:

```
<div th:text="${#strings.toUpperCase(#strings.replace(#httpServletRequest.requestURI,'/',' '))}"
class="display-3"></div>
```

(URI dari halaman diambil, lalu separatornya ditukar menjadi spasi, lalu diubah menjadi huruf kapital.)