

Tutorial 6

Thymeleaf – Presentation Layer

Write up:

Pada tutorial ini, dibahas mengenai penggunaan thymeleaf pada *view* secara lebih lanjut, antara lain adalah penggunaan index pada iterasi, penggunaan *conditional expression*, *static file*, *fragment*, serta *error handling*. Pada *view* kita bisa menambahkan sebuah index pada iterasi yang sedang berjalan untuk menentukan apakah div yang melakukan iterasi tersebut memiliki suatu class unik atau tidak. Kemudian, ketika kita ingin menampilkan suatu hal dengan kondisi tertentu, kita juga dapat menggunakan *th:unless* dan *th:if* yang bekerja untuk menentukan hal apa yang akan ditampilkan sesuai dengan kondisi yang diberikan. Dijelaskan juga bahwa Spring Boot dan Thymeleaf memiliki fasilitas untuk menggunakan *static file* seperti *css*, *javascript*, dan lain sebagainya. *Static file* tersebut disimpan dan diakses melalui *folder static* yang sudah tersedia secara default dari pembuatan project berbasis Thymeleaf dan kita hanya perlu memasukkan *static file* apa yang ingin ditambahkan pada folder tersebut. Kemudian, terdapat juga fitur *fragment* yang dapat dibentuk dengan membuat folder *fragments* yang berisi file *fragment.html*. Fitur ini berfungsi untuk *me-reuse code* yang dibuat dengan memecah halaman HTML menjadi bagian-bagian kecil yang disimpan di dalam file *fragment.html*. Kemudian, dijelaskan juga mengenai *error handler* di mana kita bisa membuat suatu file *error* pada folder *error* yang ditempatkan di dalam folder *templates*. Folder *error* ini berisi file-file *error handling* seperti *404.html*, *500.html*, dan lain sebagainya sehingga ketika terjadi error misalnya *404 not found*, sistem akan mengembalikan file *html 404.html* dan menampilkan apa yang ada di dalam file *html* tersebut.

Penggunaan For Looping

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

Jawab: Sebuah index yang menandakan apakah div tersebut memiliki class *odd* atau tidak. Ketika iterasi yang sedang di-*run* ganjil, maka div tersebut di-*assign* memiliki class *odd*.

Penggunaan Conditional Expression pada Thymeleaf

Pertanyaan 2: Mengapa condition di dalam *th:unless* sama dengan di dalam *th:if*?

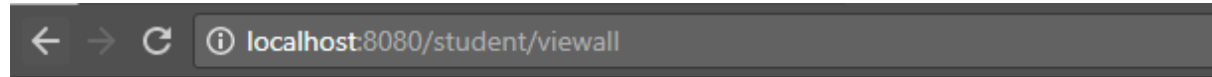
Jawab: Karena *th:unless* merupakan negasi dari *th:if* sehingga *condition* di dalam *th:unless* harus sama dengan *th:if*, karena kita ingin memberikan kondisi negasi dari *th:if* seperti pada kasus ini, kita ingin memberikan *condition* bahwa “*student* yang ipknya bukan ≥ 3.49 maka dinyatakan lulus dengan predikat sangat memuaskan” di mana untuk kasus ipk yang ≥ 3.49 sudah di-*handle* oleh *th:if*.

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?

Ricca Fitriani
1506689616
APAP – B

Jawab: Terjadi error. Dapat dilihat pada console bahwa th:unless tidak dapat memiliki condition < (kurang dari).

[x.SAXParseException](#): The value of attribute "th:unless" associated with an element type "h3" must not contain the '<' character.
com.sun.org.apache.xerces.internal.util.ErrorHandlerWrapper.createSAXParseException(Unknown Source) ~[na:1.8.0_144]



Whitelabel Error Page

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

Fri Oct 13 18:54:03 ICT 2017

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

Exception parsing document: template="viewall", line 14 - column 32

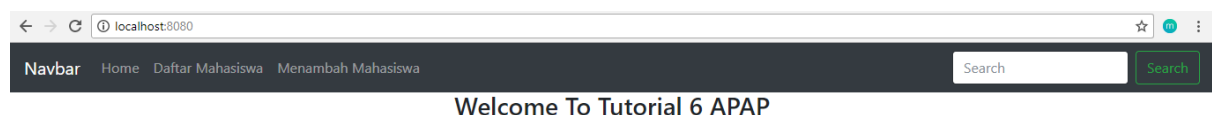
Pertanyaan 4: Tuliskan cara lain Anda dengan penulisan conditional expression yang berbeda untuk mendapatkan hasil yang sama seperti gambar di atas. Hint: Gunakan ternary operator

Jawab 4: Cara lain yang dapat dilakukan adalah sebagai berikut:

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

Static file

Menggunakan css bootstrap sebagai *static file* yang dapat digunakan pada Thymeleaf, kita dapat menampilkan *file* dengan css bootstrap sebagai berikut:



Fragment

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: th:replace tersebut memasukan fragment header dan footer yang ada pada folder fragments, file fragment sehingga pada tag div dengan th:replace tersebut memiliki nilai yang ada pada tag div dengan fragment header dan footer.

Tampilan yang muncul dengan fragment tersebut adalah sebagai berikut:



- [Home](#)
- [Daftar Mahasiswa](#)
- [Menambah Mahasiswa](#)

Welcome To Tutorial 6 APAP

Mata Kuliah APAP

Error 404 Not Found Handler

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

Jawab: Ya. Dapat dicoba ketika file yang diakses menghasilkan error 500 internal server error akan ditampilkan error sebagai berikut jika belum memiliki *handler* pada project yang dikerjakan:



Whitelabel Error Page

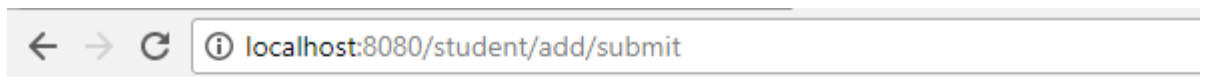
This application has no explicit mapping for /error, so you are seeing this as a fallback.

Sat Oct 14 19:23:51 ICT 2017

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

Error updating database. Cause: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Unknown column 'npm' in 'field list' ### The error may involve com.example.dao.StudentMapper.addStudent-Inline ### The error occurred while setting parameters ### SQL: INSERT INTO course (npm, name, gpa) VALUES (?, ?, ?) ### Cause: com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Unknown column 'npm' in 'field list' ; bad SQL grammar []; nested exception is com.mysql.jdbc.exceptions.jdbc4.MySQLSyntaxErrorException: Unknown column 'npm' in 'field list'

Kemudian, ketika sudah terdapat file 500.html, file tersebut akan ditampilkan jika sistem terjadi error 500 internal server sebagai berikut:



Oops! Ada Kesalahan Pada Server

Latihan

1. Mengubah view pada localhost:8080/student/viewall menggunakan DataTables, dengan kolom-kolom 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)

Dengan DataTables, maka pertama harus membuat dan mengimport JavaScript untuk DataTables tersebut sebagai berikut:

```
<script src="https://code.jquery.com/jquery-1.12.4.js"></script>
<script src="https://cdn.datatables.net/1.10.16/js/jquery.dataTables.min.js"></script>
<script>
    $(document).ready(function() {
        $('#data').DataTable( {
            "ordering": false
        });
    });
</script>
```

Ordering false berarti pada DataTables yang akan dimunculkan nanti tidak terdapat fitur sorting data per kolom tabel.

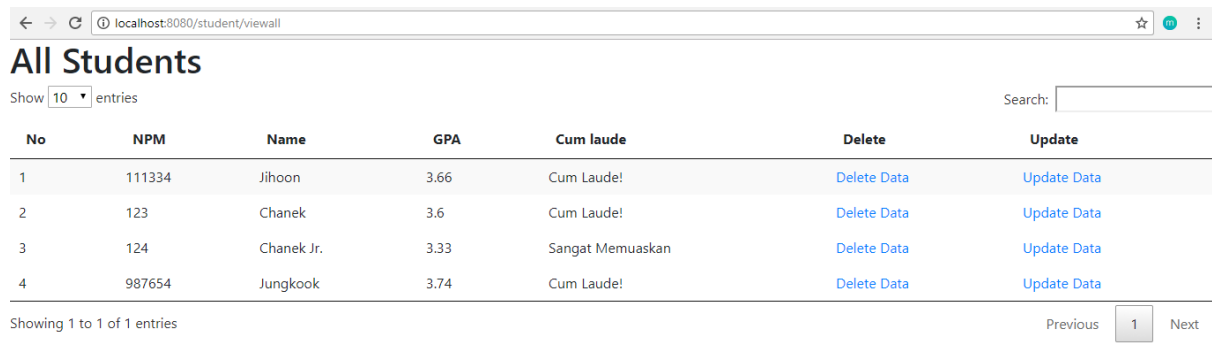
Kemudian, untuk html tabel dapat dibuat sebagai berikut:

```
<table id="data" class="display" cellspacing="0" width="100%">
  <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>
  <tbody th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}? 'odd'">
    <tr>
      <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'"></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>
</table>
```

Kemudian, untuk css DataTables tersebut harus diimport css sebagai berikut:

```
<link rel="stylesheet" href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css"/>
```

Finally, ketika program di-run dengan memanggil localhost:8080/student/viewall akan dihasilkan tampilan sebagai berikut:



No	NPM	Name	GPA	Cum laude	Delete	Update
1	111334	Jihoon	3.66	Cum Laude!	Delete Data	Update Data
2	123	Chanek	3.6	Cum Laude!	Delete Data	Update Data
3	124	Chanek Jr.	3.33	Sangat Memuaskan	Delete Data	Update Data
4	987654	Jungkook	3.74	Cum Laude!	Delete Data	Update Data

2. Membuat sebuah fragment header yang dinamis.

Pertama, dapat dibuat sebuah fragment *header* pada file *fragment.html* sebagai berikut:

```
<head th:fragment="header">
  <title th:text="${title}">Fragment</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <link rel="stylesheet" href="/css/bootstrap.min.css"/>
  <link rel="stylesheet" href="https://cdn.datatables.net/1.10.16/css/jquery.dataTables.min.css"/>
</head>
```

`th:text="${title}"` akan menampilkan *value* yang diassign untuk title tersebut oleh controller sebagai berikut:

```
@RequestMapping("/student/add")
public String add (Model model)
{
    model.addAttribute("title", "Add Student");
    return "form-add";
}
```

Misalnya pada contoh controller tersebut, controller memanggil *form-add* untuk halaman menambah *student* dengan title pada halaman tersebut “Add Student”. Maka “Add Student” tersebut akan diassign valuenya ke halaman *form-add* sebagai title menggunakan model, sehingga halaman tersebut akan ditampilkan dengan title “Add Student”. Untuk halaman lain pun dilakukan hal yang sama dengan menyesuaikan dengan title setiap halaman yang diinginkan.

Kemudian, fragment header tersebut diletakan pada setiap head halaman sebagai berikut:

```
<head th:replace="fragments/fragment :: header"></head>
```

Sehingga, ketika program dijalankan, misalnya dengan `localhost:8080/student/add`, akan ditampilkan title yang sesuai dengan halaman tersebut sebagai berikut:

Ricca Fitriani
1506689616
APAP – B

tutorial-06-thymeleaf.pdf x

Add Student x

localhost:8080/student/add

Problem Editor

NPM

Name

GPA

Save