

Thymeleaf – Presentation Layer Design



Ringkasan Materi

Pada tutorial kali ini, mempelajari bagaimana mendesign user interface dari tampilan web dengan menggunakan thymeleaf, bootstrap dan datatables. Selain itu pembuatan looping dengan menggunakan for, Pembuatan kondisi dengan menggunakan thymeleaf, pembuatan *fragment* dan pembuatan handler untuk kode error 404 dan 500.



Pertanyaan Tutorial

I. Pertanyaan 1

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

Jawab:

Ketika halaman viewall diakses maka akan menghasilkan class="odd" pada nomer iterasi yang ganjil.

II. Pertanyaan 2

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

Jawab:

Karena **th:unless** berfungsi untuk mengeksekusi kondisi diluar **th:if** yaitu ketika `gpa < 3.49`, oleh karena itu perbandingan yang digunakan sama jika menggunakan perbandingan yang berbeda maka **th:unless** akan menjalankan kondisi yang sama dengan **th:if**.

III. 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?

Jawab:

Error karena karakter "<" tidak boleh digunakan pada *format* th:text, dan lebih disarankan untuk menggunakan "<".

```
Attribute "th:unless" associated with an element of type "h3" must not contain the '<' character.
org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.createSAXParseException(ErrorHandlerWrapper.java:203) ~[na:1.8.0_101]
org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.fatalError(ErrorHandlerWrapper.java:177) ~[na:1.8.0_101]
org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.mpl.XMLErrorReporter.reportError(XMLErrorReporter.java:400) ~[na:1.8.0_101]
org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandlerAdapter.mpl.XMLErrorReporter.reportError(XMLErrorReporter.java:327) ~[na:1.8.0_101]
```

IV. 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:

Ternary Operator

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

```
<tr th:each="student, iterationStatus: ${students}">
  <td th:text=" " + ${iterationStatus.count}"></td>
  <td th:text=" " + ${student.npm}"></td>
  <td th:text=" " + ${student.name}"></td>
  <td th:text=" " + ${student.gpa}"></td>
  <td th:text= "${student.gpa}>=3.49} ? 'Cum Laude!' : 'Sangat Memuaskan!' "></td>
</tr>
```

V. 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:

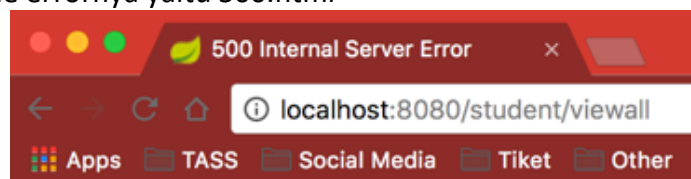
Maksudnya adalah untuk mereplace div yang ada pada `index.html` dengan kode yang ada pada `fragment.html` dimana seolah-olah meng-include file `fragment.html`

VI. Pertanyaan 6

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

Jawab:

Error 500 juga berlaku dengan membuat file html baru dengan nama sesuai dengan kode errornya yaitu `500.html`



Internal Server Error

Latihan

1) Penerapan DataTables

- Source code pada viewall.html

```
<table id="students" class="table table-hover">

  <thead class="table-dark">
    <tr>
      <th>No</th>
      <th>NPM</th>
      <th>Name</th>
      <th>GPA</th>
      <th>Status</th>
      <th>Delete</th>
      <th>Update</th>
      <th>View</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <th:each="student, iterationStatus: ${students}">
        <td th:text="' ' + ${iterationStatus.count}"></td>
        <td th:text="' ' + ${student.npm}"></td>
        <td th:text="' ' + ${student.name}"></td>
        <td th:text="' ' + ${student.gpa}"></td>
        <td th:text="${student.gpa}>=3.49 ? 'Cum Laude' : 'Sangat Memuaskan' "></td>
        <td>
          <a th:href="/student/delete/" + ${student.npm}">
            <button type="button" class="btn btn-danger">Delete</button>
          </a>
        </td>
        <td>
          <a th:href="/student/update/" + ${student.npm}">
            <button type="button" class="btn btn-primary">Update</button>
          </a>
        </td>
        <td>
          <a th:href="/student/view/" + ${student.npm}">
            <button type="button" class="btn btn-primary">View</button>
          </a>
        </td>
      </th:each>
    </tr>
  </tbody>
</table>
```

```
<script>
  $(document).ready( function () {
    $('#students').DataTable({
      "info": false
    });
  } );
</script>
```

- Implementasi datatables.js

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>View All Students</title>

    <script src="/jQuery-3.2.1/jquery-3.2.1.js"></script>

    <link href="/css/bootstrap.css" rel="stylesheet" type="text/css" />
    <link href="/DataTables-1.10.16/css/jquery.dataTables.css" rel="stylesheet" type="text/css" />
    <script src="/DataTables-1.10.16/js/jquery.dataTables.js"></script>

  </head>
```

Tampilan viewall

StudentHomeDaftar MahasiswaMenambah Mahasiswa

Search

Search

Data Mahasiswa

Show10entries

Search:

No	NPM	Name	GPA	Status	Delete	Update	View
1	1406623083	Dwi Arfi Faradina Mawardi	2.75	Sangat Memuaskan!	Delete	Update	View
2	1506797242	Esther Theresia Siotang	3.75	Cum Laude!	Delete	Update	View
3	1606954703	Aisyah Sadiyah	2.75	Sangat Memuaskan!	Delete	Update	View
4	1606954716	Akhda Khairul Zaidid	2.75	Sangat Memuaskan!	Delete	Update	View
5	1606954735	Arri Kurniawan	2.75	Sangat Memuaskan!	Delete	Update	View
6	1606954741	Ayu Musfita Nainggolan	3.5	Cum Laude!	Delete	Update	View
7	1606954754	Budi Indrawan	3.25	Sangat Memuaskan!	Delete	Update	View
8	1606954760	Budi Wahyu Herlen Adita	3.0	Sangat Memuaskan!	Delete	Update	View
9	1606954792	Fadiy Muhammad Ridho	2.75	Sangat Memuaskan!	Delete	Update	View
10	1606954810	Gani Gemilar	3.25	Sangat Memuaskan!	Delete	Update	View

Previous

12345

Next

Mata Kuliah APAP

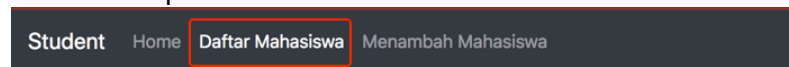
2) Penggunaan Fragments Dinamis

a. Active navbar

○ Source code pada fragment.html

```
<div class="collapse navbar-collapse" id="navbarSupportedContent">
  <ul class="navbar-nav mr-auto">
    <li class="nav-item" th:classappend="${#httpServletRequest.getRequestURI().equals('/' ? 'active':'')}">
      <a class="nav-link" th:href="@{/}">Home</a>
    </li>
    <li class="nav-item" th:classappend="${#httpServletRequest.getRequestURI().equals('/student/viewall' ? 'active':'')}">
      <a class="nav-link" th:href="@{/student/viewall}">Daftar Mahasiswa</a>
    </li>
    <li class="nav-item" th:classappend="${#httpServletRequest.getRequestURI().equals('/student/add' ? 'active':'')}">
      <a class="nav-link" th:href="@{/student/add}">Menambah Mahasiswa</a>
    </li>
  </ul>
  <form class="form-inline my-2 my-lg-0">
    <input class="form-control mr-sm-2"
      type="text"
      placeholder="Search"
      aria-label="Search" />
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">
      Search
    </button>
  </form>
</div>
```

○ Contoh tampilan navbar



Data Mahasiswa

Show 10 entries

b. Perubahan title dinamis

Pada fragment di tag title ditambah “\$title” yang nilainya akan diperoleh dari StudentController yang telah di set untuk masing-masing file html.

• Source code fragment.html

```
<head>
  <title th:text="${title}"></title>
```

• Menambahkan model.addAttribute title untuk mengganti nilai dari \$title yang ada di file html.

```
@RequestMapping("/student/viewall")
public String view (Model model)
{
  List<StudentModel> students = studentDAO.selectAllStudents ();
  model.addAttribute ("students", students);
  model.addAttribute("title", "Daftar Mahasiswa");
  return "viewall";
}
```

• Contoh Tampilan

