

1. Ringkasan Materi

Pada tutorial 5 ini lebih berfokus pada front end atau tampilan dari aplikasi. Menggunakan `th:if` dan `th:unless` untuk menampilkan status berdasarkan data pada database. Kemudian digunakan pula `css` dan `jquery` untuk bisa membuat tampilan lebih menarik. Fungsi fragmentasi untuk membagi halaman menjadi beberapa bagian seperti `header`, `body` dan `footer`.

2. Jawaban Pertanyaan Tutorial

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

Jawab : Value yang dihasilkan adalah `true` and `false`. Mengecek apakah index dari `IterationStatus` merupakan index ganjil. Jika `true` (index ganjil) maka akan hasilnya akan ditempatkan menjadi `class`.

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

Jawab : Kondisi dalam `th:unless` sama dengan kondisi dalam `th:if` karena untuk menandakan bahwa action yang dilakukan `me-refers` pada kondisi tersebut. Jadi memang fungsi dari `th:unless` adalah action yang dilakukan jika kondisi di dalamnya bernilai `false`.

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 : Iya terjadi error karena terdapat dua action untuk kondisi yang sama sehingga server akan bingung untuk mengeksekusi action yang mana. Jadi ketika `gpa` bernilai `>= 3.49` maka akan melakukan action `Cum Laude`. Dan ketika `gpa` **tidak** bernilai `<=3.48` (misal bernilai `>=49`) maka juga akan melakukan action `Sangat Memuaskan`. Kondisi dalam `th:if` dan `th:unless` haruslah sama.

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

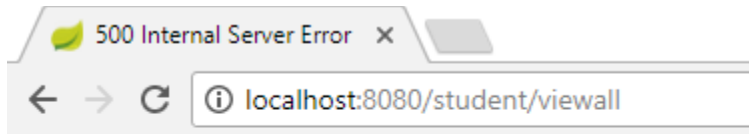
Pertanyaan 5: Apa yang dimaksud dengan `th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file `index.html` yang Anda buat?

Jawab : `th:replace="fragments/fragment :: header"` berarti akan menggantikan tag `html` yang meng-cover-nya dengan code di dalam file `fragment` dalam tag `th:fragment = "header"`.

th:replace="fragments/fragment :: footer" berarti akan menggantikan tag html yang meng-cover-nya dengan code di dalam file fragment dalam tag th:fragment = "footer".

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

Jawab : Ya dapat berlaku untuk error lain karena handle error mendeteksi status html yang dihasilkan. Berikut contoh halaman yang dihasilkan ketika terjadi error 500:internal server error

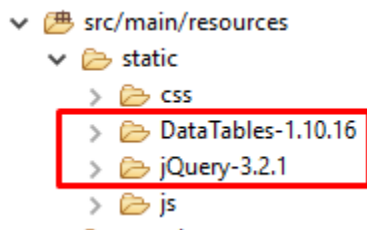


Internal Server Error

3. Penjelasan Fitur Latihan

a) Tampilan viewall dengan menggunakan datatables

Untuk dapat menggunakan fungsi dari datatables maka perlu di download library datatables-nya dan masukkan ke folder resource static.



Untuk dapat menggunakan library dari datatables maka perlu dituliskan link dan referensi script-nya.

```
<link rel="stylesheet" type="text/css" href="/DataTables-1.10.16/css/jquery.dataTables.css" />
<script type="text/javascript" charset="utf8" src="/DataTables-1.10.16/js/jquery.dataTables.js"></script>
```

Sesuaikan iterasi yang dilakukan pada body table agar sesuai dengan datatables yang digunakan.

```

<tbody>
  <tr th:each="student, iterationStatus: ${students}">
    <td th:text="' ' + ${iterationStatus.count}"></td>
    <td>
      <a th:href="'/student/view/' + ${student.npm}" >
        <div th:text="' ' + ${student.npm}"></div>
      </a>
    </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="submit" class="btn btn-danger">Delete</button>
      </a>
    </td>
    <td>
      <a th:href="'/student/update/' + ${student.npm}">
        <button type="submit" class="btn btn-primary">Update</button>
      </a>
    </td>
  </tr>
</tbody>

```

Memanggil fungsi datatables dengan menggunakan script jquery.

```

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

```

Berikut hasil akhir dari viewall dengan datatables.

View All Students

localhost:8080/student/viewall

Navbar

All Students Table

Show entries

Search:

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

Previous 1 2 3 4 5 Next

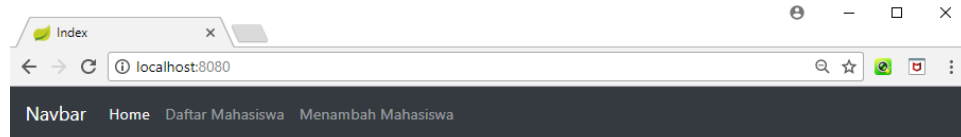
b) Membuat fragment header yang dinamis

Ubah code pada fragment.html bagian fragment selector header. Tambahkan `th:classappend` pada setiap link href navbar dimana hal ini akan membuat navbar aktif berdasarkan halaman yang sedang diakses.

```
<body>
<div th:fragment = "header" >
<link rel="stylesheet" href="/css/bootstrap.min.css"/>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="/">Navbar</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<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>
</div>
</div>
```

Berikut adalah hasil dari penambahan header yang dinamis.



The screenshot shows a web browser window with the address bar at localhost:8080/student/viewall. The page has a dark blue header with a 'Navbar' and links to 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa'. The main content area displays 'All Students Table'. Below the title, there is a 'Show 10 entries' dropdown and a 'Search:' input field. The table lists 5 students with columns for No, NPM, Name, GPA, and Cum Laude. Each row has 'Delete' and 'Update' buttons.

No	NPM	Name	GPA	Cum Laude		
1	1406623083	Dwi Arfi Faradina Mawardi	3.75	Cum Laude!	Delete	Update
2	1506797242	Esther Theresia Sihotang	3.0	Sangat Memuaskan!	Delete	Update
3	1606954703	Aisyah Sadiyah	2.75	Sangat Memuaskan!	Delete	Update
4	1606954716	Akhda Khairul Zadid	2.75	Sangat Memuaskan!	Delete	Update
5	1606954735	Arri Kurniawan	3.75	Cum Laude!	Delete	Update

The screenshot shows a web browser window with the address bar at localhost:8080/student/add. The page has a dark blue header with a 'Navbar' and links to 'Home', 'Daftar Mahasiswa', and 'Menambah Mahasiswa'. The main content area displays 'Problem Editor'. Below the title, there are input fields for 'NPM', 'Name', and 'GPA', and a 'Save' button.

NPM

Name

GPA