

All Students

No. 1

NPM = 123

Name = wowow

GPA = 3.0

Sangat Memuaskan!

Kuliah yang diambil

- PSP-4 sks
- PSP-4 sks
- MPKT-6 sks

[Delete Data](#) [Update Data](#)

No. 2

NPM = 124

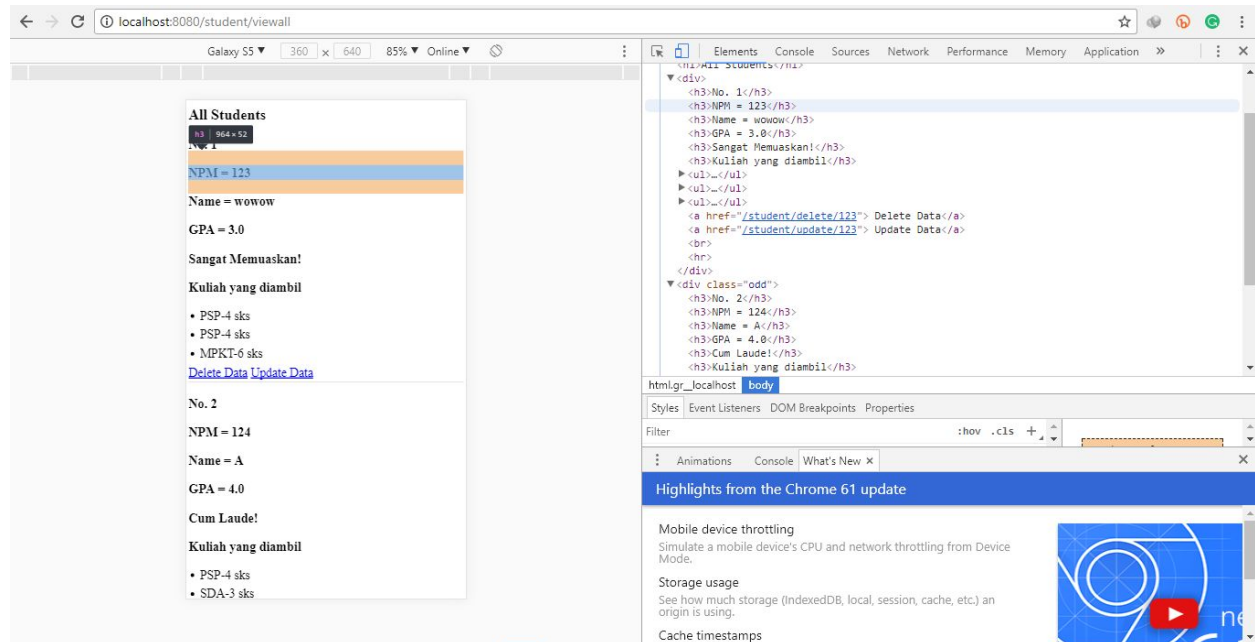
Name = A

GPA = 4.0

Cum Laude!

Pertanyaan 1:

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



Value yang dihasilkan oleh `iterationStatus.odd` adalah "odd" pada index ganjil. Dapat dilihat pada screenshot pada inspect element website, terdapat penambahan kelas `class="odd"` di tiap div dengan iteration ganjil.

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

```
<div th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}? 'odd'">
  <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
  <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
  <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
  <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  <h3 th:if="${student.gpa}>=3.49}">Cum Laude!</h3>
  <h3 th:unless="${student.gpa}>=3.49}">Sangat Memuaskan!</h3>
```

Dapat dilihat, bahwa condition dalam `th:unless` dan `th:if` adalah sama. Hal ini dikarenakan yang dimaksud dengan `th:if` adalah kondisi "Jika `gpa >= 3.49` maka cetak Cum Laude!" sedangkan `th:unless` adalah kondisi "Kecuali `gpa >= 3.49` maka cetak Sangat Memuaskan!" artinya, selain yang `gpa >= 3.49` akan dicetak Sangat Memuaskan.

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?



Whitelabel Error Page

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

Sun Oct 15 13:37:34 ICT 2017

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

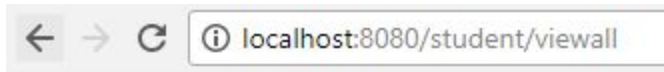
Exception parsing document: template="viewall", line 15 - column 32

Terjadi error pada viewall karena ada perubahan condition dari `th:unless=${student.gpa}>=3.49}` menjadi `th:unless=${student.gpa}<=3.48}`. Hal ini terjadi karena kondisi tersebut secara logika sama dengan `th:if=${student.gpa}>=3.49}`. Karena kesamaan tersebut, program mengeluarkan pesan error pada line `th:unless=${student.gpa}<=3.48}`.

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

```
<div th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}? 'odd'">
  <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
  <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
  <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
  <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
  <h3 th:text="${student.gpa}>=3.49 ? 'Cum Laude!' : 'Sangat Memuaskan'">Cum Laude!</h3>
```

Maka akan menghasilkan seperti ini pada program



All Students

No. 1

NPM = 123

Name = wowow

GPA = 3.0

Sangat Memuaskan

Kuliah yang diambil

- PSP-4 sks
- PSP-4 sks
- MPKT-6 sks

[Delete Data](#) [Update Data](#)

No. 2

NPM = 124

Name = A

GPA = 4.0

Cum Laude!

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<title>Index</title>
<link rel="stylesheet" href="/css/bootstrap.min.css" />
</head>
<body>
<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>

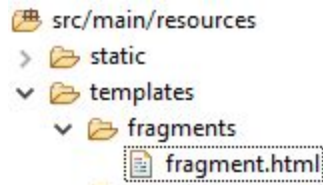
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item">
        <a class="nav-link" href="/">Home</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/student/viewall">Daftar Mahasiswa</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="/student/add">Menambah Mahasiswa</a>
      </li>
    </ul>
  </div>
</nav>
```

- ▼ src/main/resources
 - ▼ static
 - ▼ css
 - bootstrap-grid.css
 - bootstrap-grid.css.map
 - bootstrap-grid.min.css
 - bootstrap-grid.min.css.map
 - bootstrap-reboot.css
 - bootstrap-reboot.css.map
 - bootstrap-reboot.min.css
 - bootstrap-reboot.min.css.map
 - bootstrap.css
 - bootstrap.css.map
 - bootstrap.min.css
 - bootstrap.min.css.map



Welcome To Tutorial 6 APAP

Fragment



Isi pada fragment.html

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

<body>
  <div th:fragment="header">
    <ul>
      <li><a href="/">Home</a></li>
      <li><a href="/student/viewall">Daftar Mahasiswa</a></li>
      <li><a href="/student/add">Menambah Mahasiswa</a></li>
    </ul>
  </div>

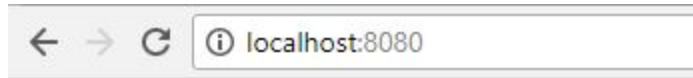
  <div th:fragment="footer">
    <hr/>
    <h4>Mata Kuliah APAP</h4>
  </div>
</body>

</html>
```

Pada index.html menjadi seperti berikut

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title>Index</title>
</head>
<body>
  <div th:replace="fragments/fragment :: header"></div>
  <h2>Welcome To Tutorial 6 APAP</h2>
  <div th:replace="fragments/fragment :: footer"></div>
</body>
</html>
```

Tampilan akan menjadi seperti ini:



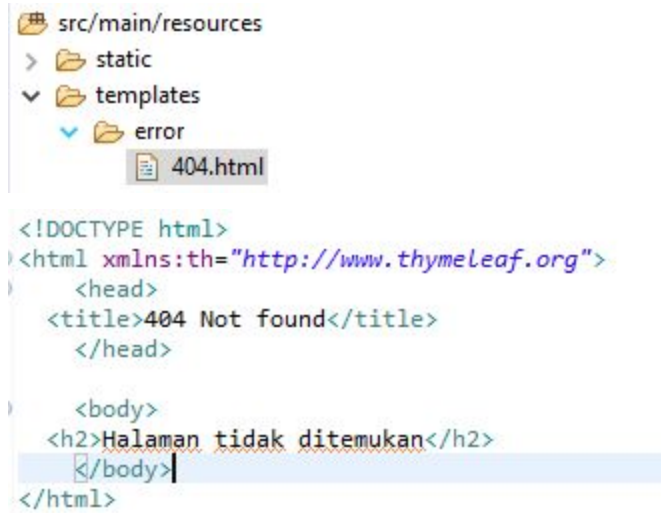
- [Home](#)
- [Daftar Mahasiswa](#)
- [Menambah Mahasiswa](#)

Welcome To Tutorial 6 APAP

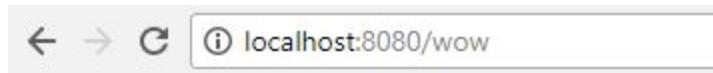
Mata Kuliah APAP

Pertanyaan 5: Apa yang dimaksud dengan `th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file `index.html` yang Anda buat? Hal itu berarti div tersebut akan diisi oleh div yang ada dalam `fragment.html` dengan `th:fragment="header"` untuk header, dan `th:fragment="footer"` untuk footer.

Error 404 Not Found Handler

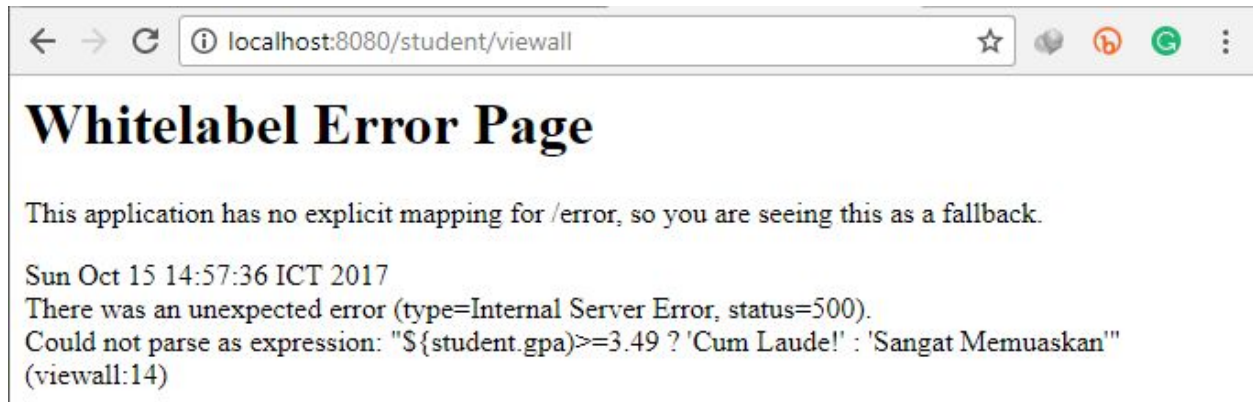


Ketika mengakses halaman yang tidak ada, akan menghasilkan



Halaman tidak ditemukan

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



Error 500: internal server error tidak terhandle dengan adanya halaman error 404 karena jenis error yang berbeda.

LATIHAN

1. Pada tutorial, anda sudah mencoba untuk memperlihatkan apakah student tersebut *cum laude* atau tidak sampai ke bagian view. Pada latihan kali ini, anda diminta untuk mengubah view tersebut menggunakan DataTables. DataTables adalah plugin jquery *library*, yang dapat membuat sebuah tabel lebih interaktif dan lebih baik (<https://datatables.net/>). Oleh karena itu, gunakan data tabel untuk mengubah tampilan **localhost:8080/student/viewall**. Dengan kolom-kolom yang harus ada 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)

Anda tidak harus menggunakan datatables versi *server side*

(https://datatables.net/examples/server_side/), yang mana lebih baik digunakan untuk data yang sangat banyak. Walaupun demikian, jika anda anda ingin mengimplementasi hal ini sangat diperbolehkan, tetapi tidak ada penambahan nilai untuk hal tersebut.

Import terlebih dahulu css dan js datatable ke dalam folder static serta sertakan pada halaman viewall.html. Jangan lupa untuk menyertakan jquery. Lalu buat table seperti:


```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
<link rel="stylesheet" type="text/css" href="/css/datatables.min.css"/>
<link rel="stylesheet" type="text/css" href="/css/bootstrap.min.css"/>
<script type="text/javascript" src="/js/bootstrap.min.js"></script>
</head>
<body>
<div th:replace="fragments/fragment :: header"></div>
<h1>All Students</h1>
<table id="allStudents" class="display" style="text-align:center;">
  <thead>
    <tr>
      <th>No.</th>
      <th>NPM</th>
      <th>Name</th>
      <th>GPA</th>
      <th>Status</th>
      <th>Delete</th>
      <th>Update</th>
    </tr>
  </thead>
  <tbody>
    <tr th:each="student, iterationStatus: ${students}" th:class="${iterationStatus.odd}? 'odd'">
      <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'">Cum Laude!</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>
<script type="text/javascript" src="/js/datatables.min.js"></script>
<script>
<script type="text/javascript" src="/js/datatables.min.js"></script>
<script>
$(document).ready( function () {
  $('#allStudents').DataTable();
} );
</script>
</body>
</html>
```

Maka akan menghasilkan table seperti ini:

← → ↻

localhost:8080/student/viewall

☆ 🔊 🔍 🌐

View All Students

Home

Daftar Mahasiswa

Menambah Mahasiswa

All Students

Showing 10 entries

Search:

No.	NPM	Name	GPA	Status	Delete	Update
1	123	wowow	3.0	Sangat Memuaskan	Delete Data	Update Data
2	124	hahahah	4.0	Cum Laude!	Delete Data	Update Data
3	125	Ab	4.0	Cum Laude!	Delete Data	Update Data

Showing 1 to 3 of 3 entries

Previous

1

Next

2. Pada tutorial, Anda sudah mencoba membuat *fragment header* dan *footer*. Tetapi, *header* dan *footer* pada tutorial tersebut tidak dinamis. Data HTML pada *header* atau *footer* akan sama di setiap *view* yang memakai *fragment* tersebut.

Pada latihan kali ini, Anda diminta untuk membuat sebuah *fragment header* yang dinamis. Dimana, aspek dinamis tersebut terdapat pada *title* yang diletakkan pada *fragment header*, bukan setiap page HTML yang ada. Lebih jauh, *title* tersebut akan berubah-ubah sesuai dengan *page* yang ada. Silahkan implementasi dengan kebutuhan tersebut, dan jika sudah berhasil, pastikan *header* tersebut dimuat pada setiap *view* yang ada di *project* ini.

Lakukan perubahan pada fragment

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/'}">Index</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/course/view/${id_course}'}">View Course</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/view/${npm}'}">View Student by NPM</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/viewall'}">View All Students</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/delete/${npm}'}">Delete Student</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/delete'}">Delete</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/add'}">Add Student</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/add/submit'}">Add</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/update/${npm}'}">Update Student</title>
  <title th:fragment="title" th:if="${HttpServletRequest.requestURI == '/student/update/submit'}">Update</title>
</head>

<body>
  <div th:fragment="header">
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a th:if="${HttpServletRequest.requestURI == '/'}" class="navbar-brand">Index</a>
      <a th:if="${HttpServletRequest.requestURI == '/course/view/${id_course}'}" class="navbar-brand">View Course</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/view/${npm}'}" class="navbar-brand">View Student by NPM</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/viewall'}" class="navbar-brand">View All Students</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/delete/${npm}'}" class="navbar-brand">Delete Student</a>
      <a th:if="${HttpServletRequest.requestURI == '/delete'}" class="navbar-brand">Delete</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/add'}" class="navbar-brand">Add Student</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/add/submit'}" class="navbar-brand">Add</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/update/${npm}'}" class="navbar-brand">Update Student</a>
      <a th:if="${HttpServletRequest.requestURI == '/student/update/submit'}" class="navbar-brand">Update</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">
            <a class="nav-link" href="/">Home</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/student/viewall">Daftar Mahasiswa</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/student/add">Menambah Mahasiswa</a>
          </li>
        </ul>
      </div>
    </nav>
  </div>

  <div th:fragment="footer">
    <hr/>
    <h4>Mata Kuliah APAP</h4>
  </div>
</body>
</html>
```

Lalu lakukan perubahan pada tiap halamannya

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <title th:replace="fragments/fragment :: title"></title>
  <link rel="stylesheet" type="text/css" href="/css/bootstrap.min.css"/>
  <script type="text/javascript" src="/js/bootstrap.min.js"></script>
</head>
<body>
  <div th:replace="fragments/fragment :: header"></div>
  <h2>Welcome To Tutorial 6 APAP</h2>
  <div th:replace="fragments/fragment :: footer"></div>
</body>
</html>
```

Kurang lebih seperti itu pada tiap halaman
Lalu, ketika ditampilkan akan seperti:



Mata Kuliah APAP

Begitu juga pada halaman-halaman yang lain, akan menampilkan title dan title-navbar yang berbeda

Lesson Learned:

Pada tutorial 6, saya belajar bagaimana memanfaatkan static pada template. Static berguna untuk menyimpan file css, js, dsb untuk mendukung bagian view dari program. Saya juga belajar bagaimana memanfaatkan thymeleaf untuk keperluan mengaitkan view dengan java spring. Thymeleaf dapat membantu kita untuk memanipulasi view sesuai dengan data yang ada dalam program maupun database. Selanjutnya, saya juga belajar menggunakan fragment. Fragment merupakan salah satu fasilitas yang disediakan oleh thymeleaf yang berguna untuk memberikan "template" atau sesuatu yang sering digunakan seperti header atau footer untuk digunakan di berbagai bagian aplikasi lainnya. Saya juga belajar untuk membuat halaman error, hal itu dapat dilakukan dengan cara membuat halaman dengan namaerror.html, lalu ketika error tersebut muncul, akan ditampilkan halaman error. Hal itu berguna untuk meningkatkan ux dari aplikasi kita. Terakhir, saya juga belajar menggunakan datatables agar dapat menampilkan data secara rapi dengan fitur-fitur yang disediakan.