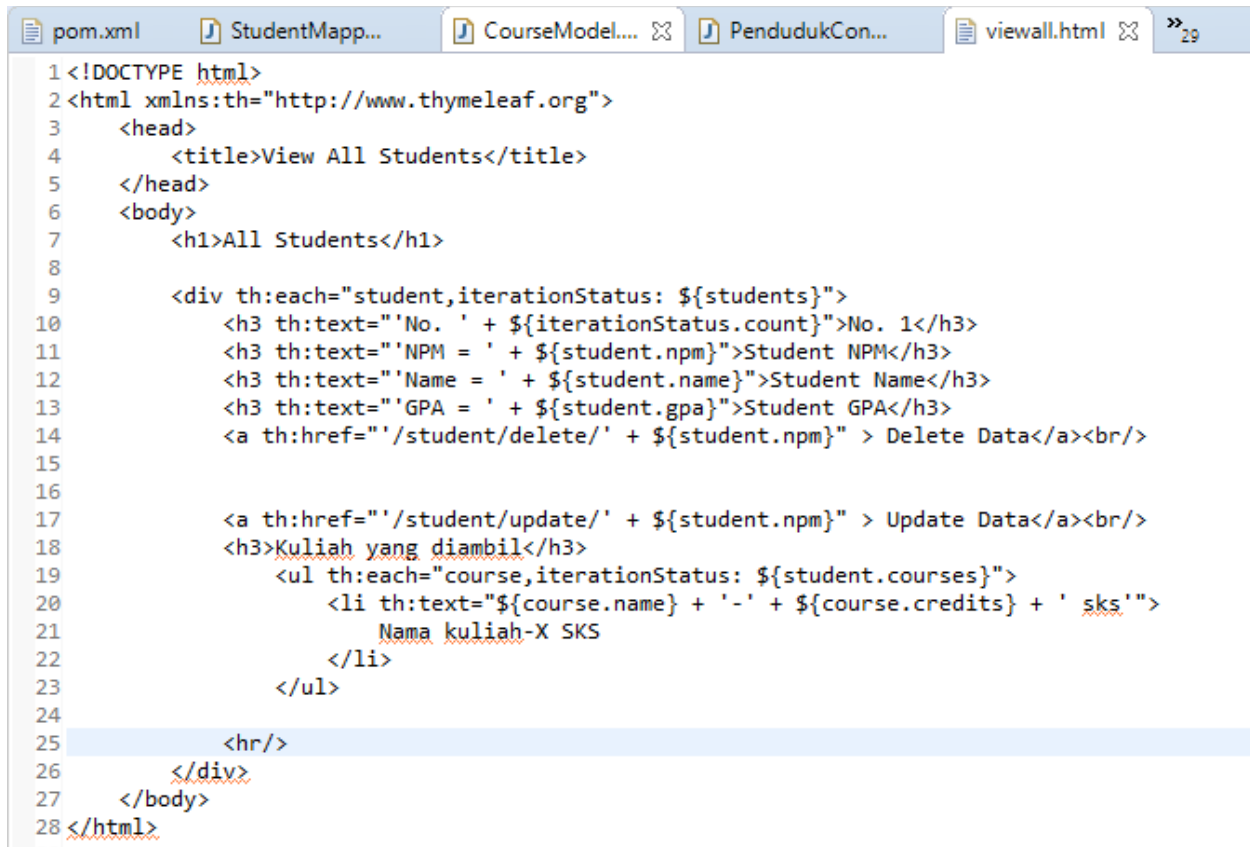


Penggunaan For Looping

Berikut ini contoh penggunaan index ganjil atau genap untuk menandakan suatu class.



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>View All Students</title>
5   </head>
6   <body>
7     <h1>All Students</h1>
8
9     <div th:each="student,iterationStatus: ${students}">
10      <h3 th:text="'No. ' + ${iterationStatus.count}">No. 1</h3>
11      <h3 th:text="'NPM = ' + ${student.npm}">Student NPM</h3>
12      <h3 th:text="'Name = ' + ${student.name}">Student Name</h3>
13      <h3 th:text="'GPA = ' + ${student.gpa}">Student GPA</h3>
14      <a th:href="'/student/delete/' + ${student.npm}" > Delete Data</a><br/>
15
16      <a th:href="'/student/update/' + ${student.npm}" > Update Data</a><br/>
17      <h3>Kuliah yang diambil</h3>
18      <ul th:each="course,iterationStatus: ${student.courses}">
19        <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
20          Nama kuliah-X SKS
21        </li>
22      </ul>
23
24    </div>
25    <hr/>
26  </body>
27 </html>
```

Pertanyaan 1:

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

➔ Value yang dihasilkan dari `${iterationStatus.odd}` adalah angka-angka ganjil.

Penggunaan Conditional Expression pada Thymeleaf

Ketika ingin menampilkan sesuatu hal dengan kondisi tertentu, Thymeleaf menyediakan conditional expression, salah satunya berupa `th:if=${...}` dan `th:unless=${...}`. Hal ini dapat memudahkan kita dalam menampilkan suatu tampilan dengan kondisi tertentu. Berikut inicara yang dapat dilakukan.

1. Tambahkan kode baris berikut ini pada file `viewall.html`.

Nabilah Zhafira Viderisa

1506721806



ADPAP – C





```
<h3>Kuliah yang diambil</h3>
<ul th:each="course, iterationStatus: ${student.courses}">
  <li th:text="${course.name} + '-' + ${course.credits} + ' sks'">
    Nama kuliah-X SKS
  </li>
</ul>

<h3> th:if="${student.gpa}>=3.49}">Cum Laude!</h3>
<h3> th:unless="${student.gpa}>=3.49}">Sangat Memuaskan!</h3>
```

2. Coba run aplikasi Anda, lalu akses localhost:8080/student/viewall. Tampilan akan seperti gambar di bawah ini.

Nabilah Zhafira Viderisa
1506721806
ADPAP – C

 View All Students 

    localhost:8080/student/viewall

All Students

No. 1

NPM = 123

Name = Nabilah

GPA = 3.9

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

Kuliah yang diambil

- MPKT-6 sks

Cum Laude!

No. 2

NPM = 124

Name = bila

GPA = 3.2

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

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

Sangat Memuaskan!

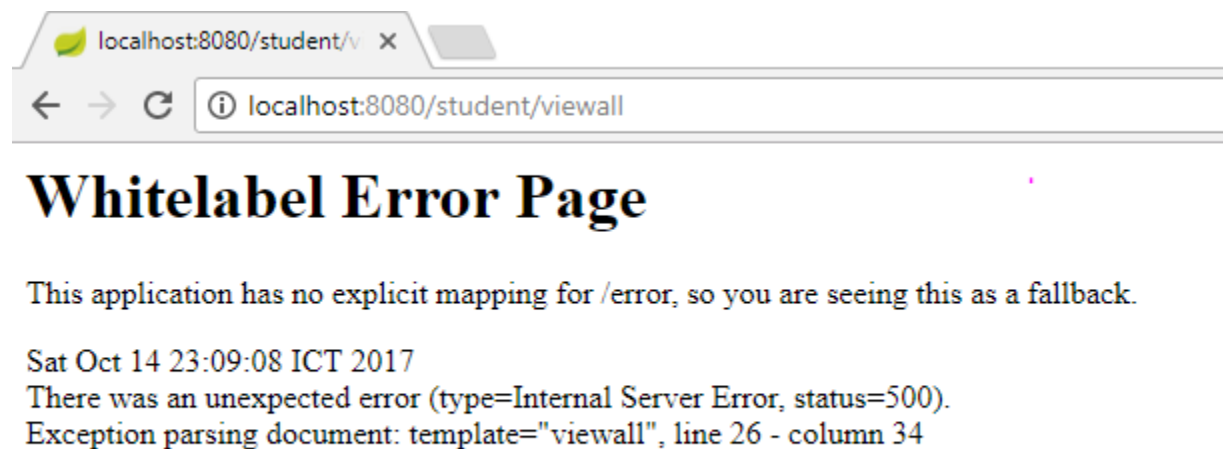
Pertanyaan 2: Mengapa condition di dalam th:unless sama dengan di dalam th:if? Jelaskan jawaban Anda.

- ➔ Karena th:unless memiliki fungsi yang hampir sama seperti *if else*. Pada kode tersebut th:unless berarti jika kondisi bukan seperti yang tertera, maka akan keluar “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?

```
<h3 th:if="${student.gpa}>=3.49}">Cum Laude!</h3>  
<h3 th:unless="${student.gpa}<=3.48}">Sangat Memuaskan!</h3>
```

- ➔ Terjadi error. Hal ini terjadi karena *condition* yang ada sama dengan *condition* sebelumnya. Sedangkan fungsi `th:unless` sendiri sama seperti fungsi *elseif* yaitu mengeluarkan output tersebut jika *condition* selain yang sebelumnya. *Condition* yang sama itulah yang menyebabkan error.



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.

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

Nabilah Zhafira Viderisa
1506721806
ADPAP – C

View All Students

localhost:8080/student/viewall

All Students

No. 1

NPM = 123

Name = Nabilah

GPA = 3.9

[Delete Data](#)

[Update Data](#)

Kuliah yang diambil

- MPKT-6 sks

Cum Laude!

No. 2

NPM = 124

Name = bila

GPA = 3.2

[Delete Data](#)

[Update Data](#)

Kuliah yang diambil

- PSP-4 sks
- SDA-3 sks

Sangat Memuaskan!

Static file

Spring Boot dan Thymeleaf memberikan fasilitas untuk menggunakan static file seperti file css, javascript, image, dan sebagainya. Static file tersebut disimpan dan diakses melalui folder static yang sudah tersedia secara default dari pembuatan project berbasis Thymeleaf. Pada bagian ini, kita akan mempelajari implementasi css bootstrap sebagai static file yang dapat digunakan pada Thymeleaf

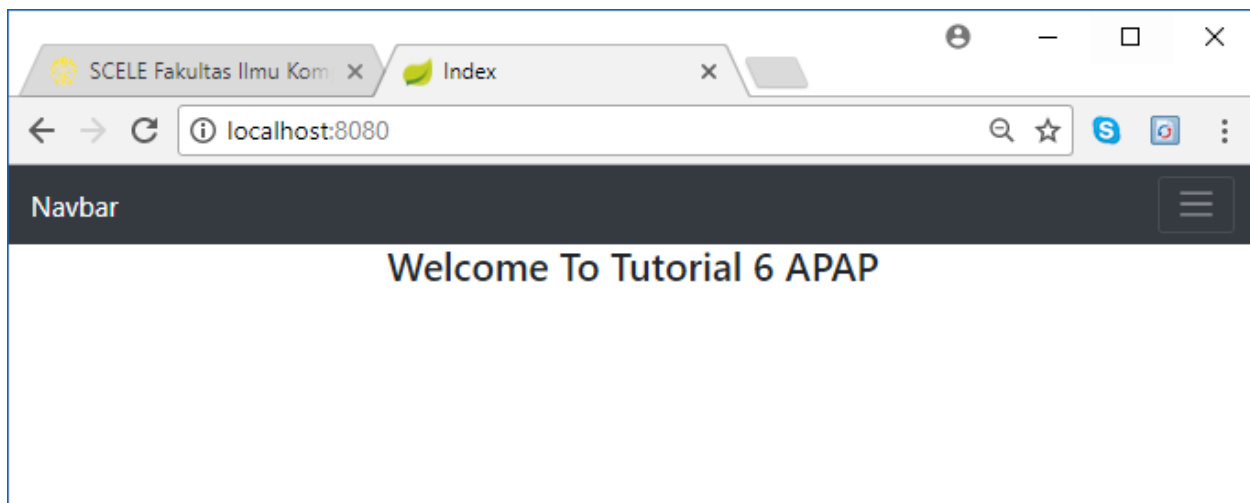
Berikut ini adalah cara pemakaian bootstrap:

1. Unduh bootstrap pada <http://getbootstrap.com/docs/4.0/getting-started/download/>
2. Extract folder zip yang Anda unduh ke dalam folder static yang terdapat pada folder src/main/resources.
3. Buat halaman index.html menjadi source code dibawah ini

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>Index</title>
5 <link rel="stylesheet" href="/css/bootstrap.min.css" />
6 </head>
7 <body>
8 <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
9 <a class="navbar-brand" href="#">Navbar</a>
10 <button class="navbar-toggler" type="button" data-toggle="collapse"
11 data-target="#navbarSupportedContent"
12 aria-controls="navbarSupportedContent" aria-expanded="false"
13 aria-label="Toggle navigation">
14 <span class="navbar-toggler-icon"></span>
15 </button>
16
17 <div class="collapse navbar-collapse" id="navbarSupportedContent">
18 <ul class="navbar-nav mr-auto">
19 <li class="nav-item">
20 <a class="nav-link" href="/">Home</a>
21 </li>
22 <li class="nav-item">
23 <a class="nav-link" href="/student/viewall">Daftar Mahasiswa</a>
24 </li>
25 <li class="nav-item">
26 <a class="nav-link" href="/student/add">Menambah Mahasiswa</a>
27 </li>
28 </ul>
29
30 <form class="form-inline my-2 my-lg-0">
31 <input class="form-control mr-sm-2"
32 type="text"
33 placeholder="Search"
34 aria-label="Search" />
```

```
35         <button class="btn btn-outline-success my-2 my-sm-0" type="submit">
36             Search
37         </button>
38     </form>
39 </div>
40 </nav>
41 <center>
42     <h3>Welcome To Tutorial 6 APAP</h3>
43 </center>
44 </body>
45 </html>
```

4. Akses halaman index dari aplikasi Anda. Contoh tampilan adalah sebagai berikut.



Fragment

Fragment merupakan sebuah fitur dalam Thymeleaf yang memungkinkan developer untuk me-reuse source code dari halaman HTML-nya. Fitur ini memungkinkan Anda untuk memecah halaman HTML menjadi bagian-bagian kecil dan bagian tersebut dapat digunakan oleh halaman HTML lain. Code reuse ini berguna ketika developer menggunakan komponen-komponen yang sering dipakai di berbagai bagian dalam aplikasi yang dibuatnya. Contohnya adalah pembuatan header dan footer. Thymeleaf juga sudah menyediakan dokumentasi mengenai fragment yang dapat diakses pada <http://www.thymeleaf.org/doc/articles/layouts.html>

Berikut ini adalah cara penggunaan fragment:

1. Buat sebuah folder dalam folder templates dengan nama fragments
2. Buat sebuah halaman HTML bernama fragment.html yang berisi source code seperti dibawah ini. File HTML tersebut dibuat pada folder fragments yang baru Anda buat sebelumnya

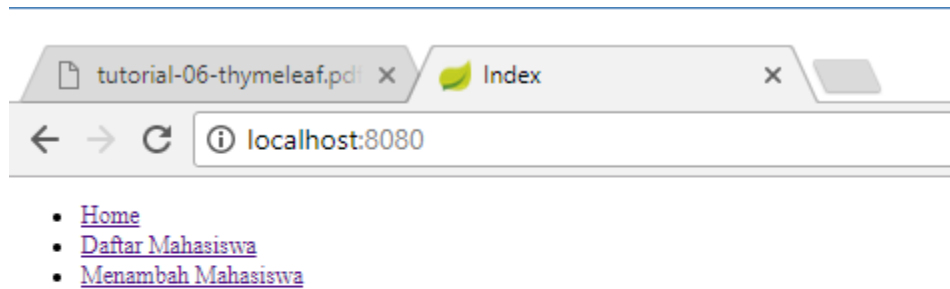
```
index.html StudentCont... form-add.html fragment.html 65
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>Fragment</title>
5 </head>
6
7 <body>
8   <div th:fragment="header">
9     <ul>
10      <li><a href="/">Home</a></li>
11      <li><a href="/student/viewall">Daftar Mahasiswa</a></li>
12      <li><a href="/student/add">Menambah Mahasiswa</a></li>
13    </ul>
14  </div>
15
16  <div th:fragment="footer">
17    <hr/>
18    <h4>Mata Kuliah APAP</h4>
19  </div>
20 </body>
21
22 </html>
```

3. Pada source code diatas, th:fragment merepresentasikan nama fragment dari bagian yang Anda ingin jadikan sebuah fragment.

4. Buat halaman index.html menjadi source code dibawah ini

```
<!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>
```

5. Coba jalankan halaman index yang Anda buat. Contoh tampilan yang keluar adalah sebagai berikut.



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

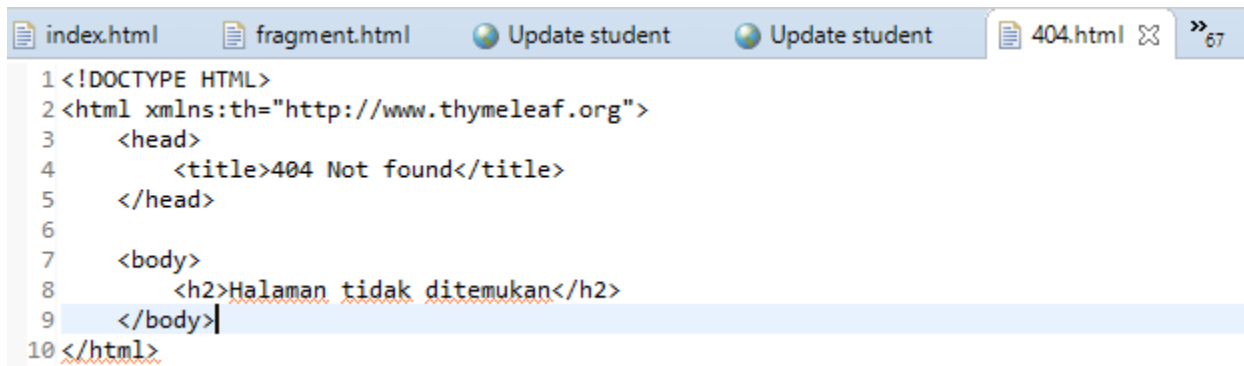
- ➔ `th:replace="fragments/fragment :: header"` dan `th:replace="fragments/fragment :: footer"` pada file `index.html` memiliki maksud bahwa semua `div` dari `fragments.html` yang memiliki `header` dan `footer` akan diimport pada `index.html`

Error 404 Not Found Handler

Dalam membuat aplikasi berbasis web, terkadang halaman web yang dituju oleh pengguna tidak dapat ditemukan. Pengguna dapat kebingungan jika diberikan halaman error yang tidak umum atau tidak jelas. Untuk mengatasi hal tersebut, kita dapat membuat handler melalui Thymeleaf. Pembuatan handler ini berguna agar pengguna mengetahui mengapa suatu error terjadi. Misalnya jika terjadi error 404 not found, kita dapat mengembalikan halaman HTML yang mengatakan bahwa halaman yang pengguna akses tersebut tidak dapat ditemukan. Hal ini akan meningkatkan UX value yang diberikan oleh aplikasi yang kita kembangkan kepada pengguna.

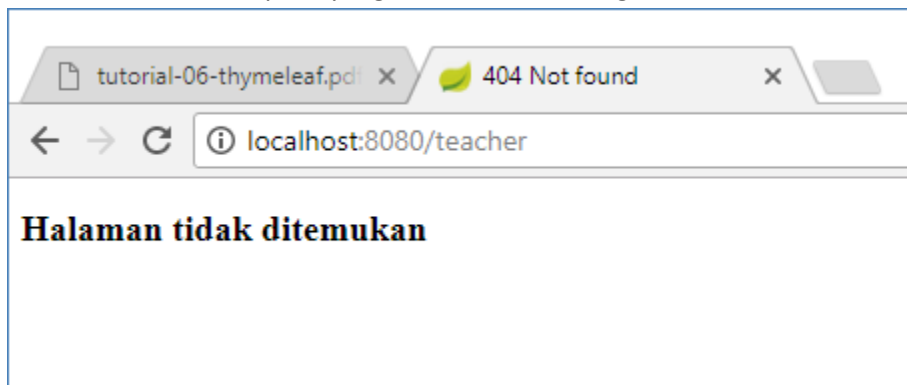
Berikut ini adalah cara untuk membuat error 404 not found handler:

1. Buat sebuah folder dalam folder `templates` dengan nama `error`
2. Buat sebuah halaman HTML bernama `404.html` yang berisi source code seperti dibawah ini. File HTML tersebut dibuat pada folder `error` yang baru Anda buat sebelumnya



```
1 <!DOCTYPE HTML>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <title>404 Not found</title>
5   </head>
6
7   <body>
8     <h2>Halaman tidak ditemukan</h2>
9   </body>
10 </html>
```

3. Akses suatu halaman yang tidak tersedia dari aplikasi Anda. Misalnya halaman dengan mapping /teacher. Contoh tampilan yang keluar adalah sebagai berikut



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 dapat terhandle oleh 404.html. error tersebut dapat dihandle dengan cara membuat 500.html

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 kolum-kolum yang harus ada sebagai berikut:

1. No
2. NPM
3. Name

Nabilah Zhafira Viderisa
1506721806
ADPAP – C

4. GPA

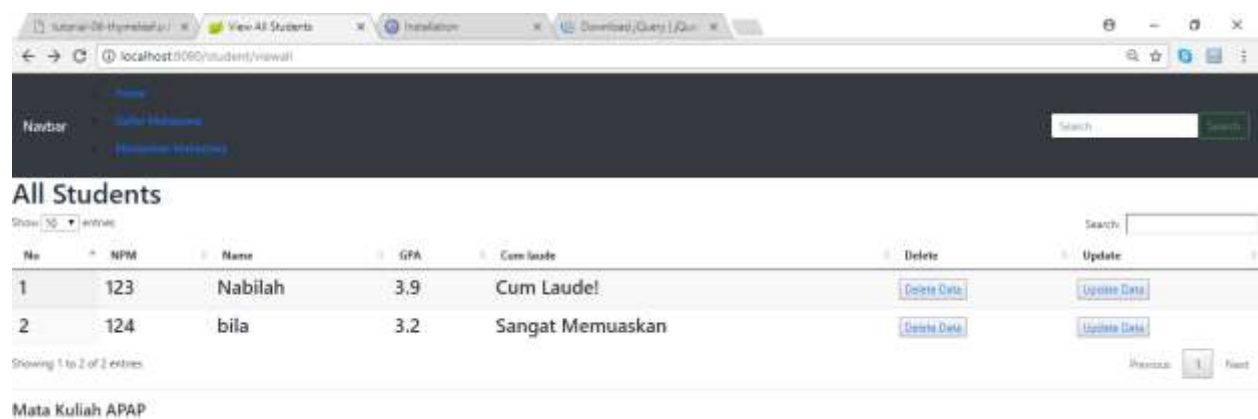
5. Cum laude

6. Delete (jika diklik, maka dapat menghapus student,)

7. Update (jika diklik, yang dapat mengupdate student)

Anda tidak harus menggunakan datatables versi server side

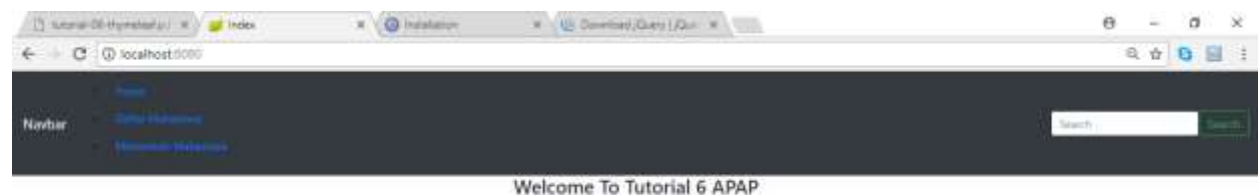
(https://datatables.net/examples/server_side/), yang mana lebih baik digunakan untuk data yang sangat banyak. Walaupun demikian, jika anda anda ingin mengimplementasi hal ini sangat diperbolehkan, tetapi tidak ada penambahan nilai untuk hal tersebut.



No	NPM	Name	GPA	Cum laude	Delete	Update
1	123	Nabilah	3.9	Cum Laude!	Delete Data	Update Data
2	124	bila	3.2	Sangat Memuaskan	Delete Data	Update Data

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.



Nabilah Zhafira Viderisa
1506721806
ADPAP – C



LESSON LEARNED:

Pada tutorial 6 kali ini, saya belajar untuk membuat *error page*. Selain itu saya juga belajar untuk membuat tabel dengan menggunakan *datatables*. Pada tutorial 6 ini saya juga belajar membuat *presentation layer*, dimana tiap layer yang dibentuk pada template kemudian ditambahkan navbar secara dinamis dengan menggunakan *fragments* untuk setiap page nya. Pada tutorial ini saya juga belajar menggunakan untuk menggunakan *ternary operator*.