Nama : Windiany Lestari Sitorus

Kelas : Ekstensi

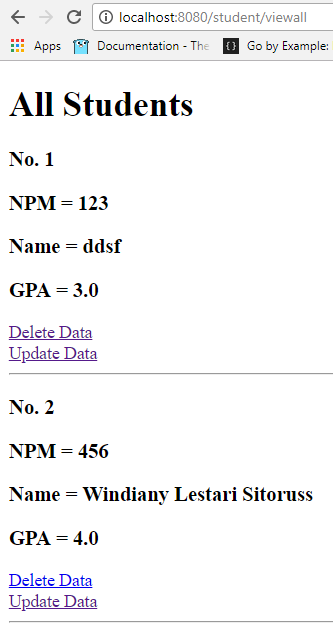
NPM : 1606955063

**Penggunaan For Looping**

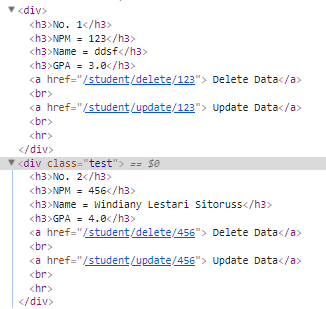
**Pertanyaan 1**

1. Value yang dihasilkan ${iterationStatus.odd} adalah menghasilkan class=‘odd’ dengan iterationStatus yang ganjil.

Hasil tampilan pada browser akan sama saja, tidak ada perubahan.



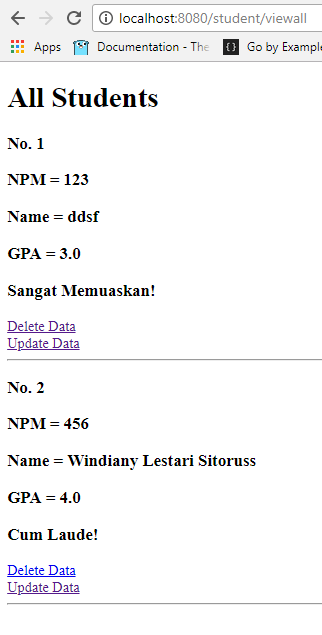
Akan tetapi apabila di inspect element akan ditemukan saat iterationStatus ke 2 (akan tetapi indexing dimulai dari 0) dengan index 1 = ganjil akan dicreate class = ‘odd’ sesuai code.



**Penggunaan Conditional Expression pada Thymeleaf**

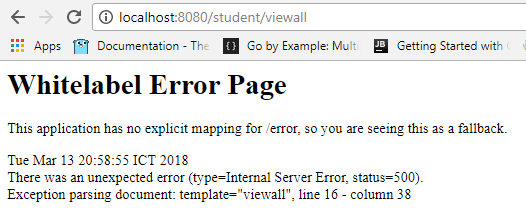
**Pertanyaan 2**

1. th:unless merupakan negative counterpart dari th:if, seperti not (!). Sehingga condition dalam th:if dan th:unless sama, karena sudah dibedakan dengan adanya th:unless. Apabila condition dalam th:unless diubah dengan kebalikan condition th:if, maka conditionnya akan sama dengan th:if dan akan menghasilkan error. Th:unless dapat berdiri sendiri tanpa th:if. Condition dalam th:unless juga tidak harus selalu sama dengan th:if, asalkan tidak memiliki makna yang sama dengan th:if.



**Pertanyaan 3**

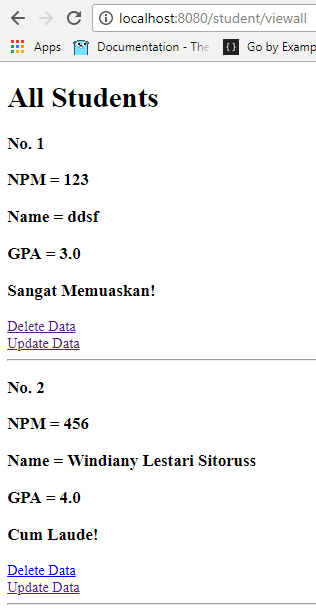
1. Terjadi error karena condition dalam th:unless menggambarkan hal yang sama dengan th:if. Pada th:unless=${student.gpa<=3.48} memiliki arti apabila student.gpa tidak lebih kecil sama dengan 3.48 sehingga target value nya adalah student.gpa lebih besar sama dengan 3.49. Sementara hal yang sama juga terjadi akan di th:if=${student.gpa>=3.49}. Hal ini menyebabkan error karena condition yang dicover memiliki arti yang sama.



Pertanyaan 4

1. Cara lain penulisan conditional expression menggunakan ternary operator adalah:





**Static File**

Pada bagian ini diajarkan menggunakan bootstrap.

**Fragment**

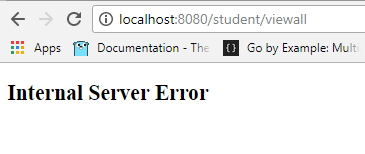
Pertanyaan 5

1. th:replace=”fragments/fragment::header” memiliki arti menggantikan <div dengan fragment header yang telah didefine pada fragments/fragment.html. Berlaku juga dengan yang footer.

**Error 404 Not Found Handler**

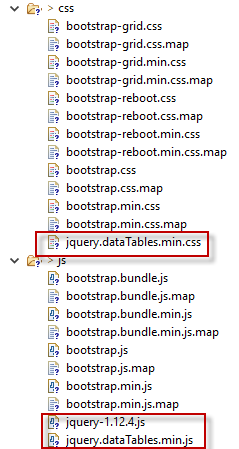
Pertanyaan 6

1. Handler ini juga berlaku bagi error lain, terbukti ketika saya mencoba file 500.html di folder error dapat handle Internal server error yang terjadi. Jadi setiap file html yang dicretae akan spesifik menghandle sesuai dengan HTML error code.



**Latihan**

1. Setelah membuka <https://datatables.net/examples/basic_init/zero_configuration.html> menggunakan bagian Basic Initialisation, saya mengunduh file css dan js yang telah disediakan, kemudian menambahkan pada folder css dan js yang telah ditambahkan pada tutorial sebelumnya.

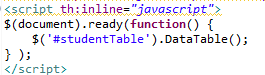


Kemudian menambahkan script css dan js pada viewall.html.





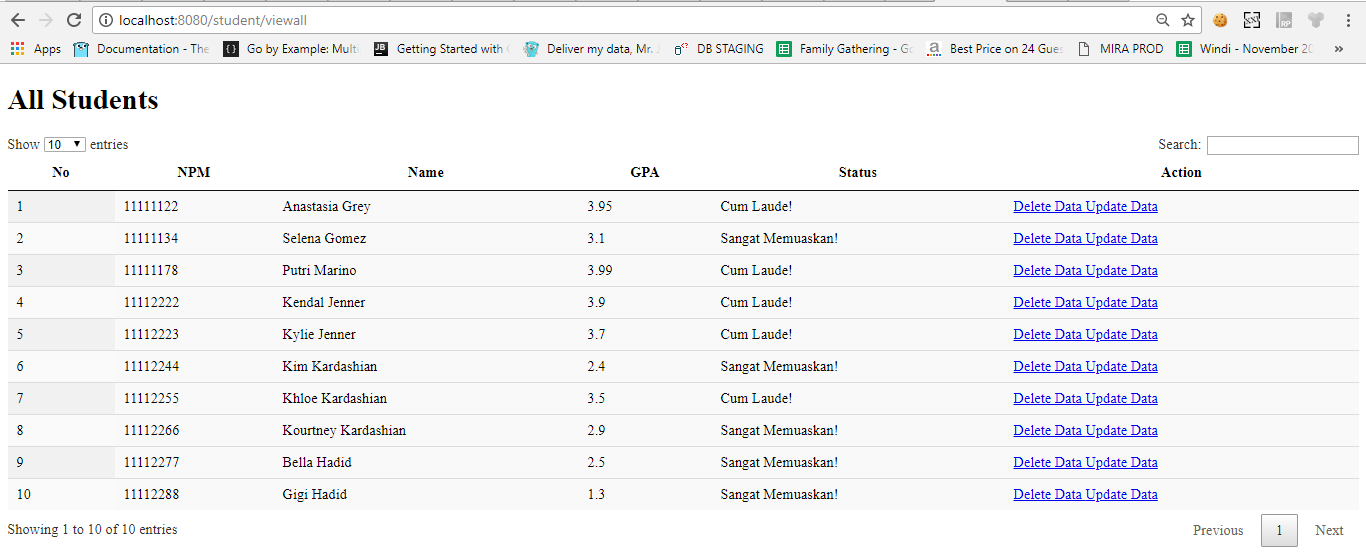
Kemudian menambahkan function data table.



Dan mengubah script html menjadi seperti berikut:



Maka saat dijalankan di browser akan menjadi:



1. Untuk membuat fragment header di bagian title, maka pada fragment.html ditambahkan variable $title yang akan menampung nama title untuk setiap page yang akan ditampilkan.



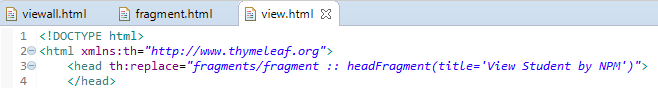
Kemudian pada file html lainnya tambahkan, untuk menggantikan tag head dengan head yang telah didefine pada fragment.html. Disini variable title akan diganti dengan value title yang anda inginkan.

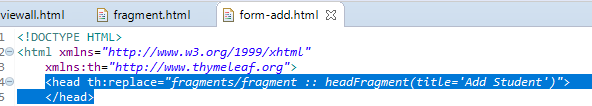


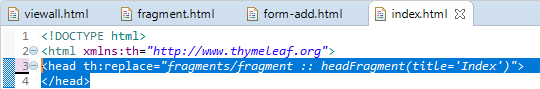


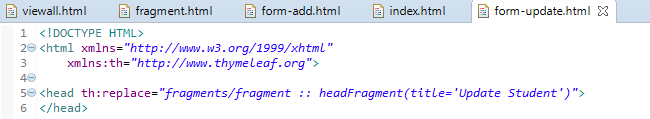
Menurut saya, latihan fragment ini menunjukkan bahwa code dalam fragment dapat di reuse di halaman html yang kita inginkan, sehingga code lebih efisien dan reusable.

Kemudian semua tag head pada file html akan menggunakan fragment header sehingga:









**Ringkasan Materi Tutorial 5 APAP**

1. Pada tutorial 5 ini bagian yang paling banyak dibahas adalah mengenai presentation layer atau view. Pada Java Spring Boot biasanya menggunakan template engine Thymeleaf.
2. Belajar penggunaan looping untuk menampilkan data berulang ulang menggunakan th:each.
3. Mempelajari conditional expression pada Thymeleaf menggunakan th:if da nth:unless juga conditional expression. Dan pada bagian ini juga dibahas mengenai condition di dalam if dan unless.
4. Pada tutorial ini juga diajarkan cara menggunakan boostrap dalam project.
5. Belajar mengenai fragment, yang dapat digunakan untuk reuse code. Didefinisikan di fragment menggunakan th:fragment kemudian akan digunakan di halaman html yang kita inginkan menggunakan th:replace. Fragment ini sangat berguna apabila kita sering melakukan hal berulang ulang dalam file html kita.
6. Belajar mengenai error handler, dengan create file dengan nama sesuai error code HTTP, maka akan mampu handle saat error tersebut terjadi. Seperti yang telah dilakukan di tutorial, file 404.html akan mampu handle error saat kita mengakses halaman yang tidak ada. Sama juga dengan internal server error akan mampu dihandle oleh 500.html.
7. Pada tutorial ini juga dipelajari mengenai penggunaan data table untuk mengelola data yang ditampilkan.