

1. Pelajaran

Pada tutorial 5 kali ini adalah mempelajari tentang *library* Thymeleaf. Thymeleaf ini merupakan sebuah *template engine* yang digunakan untuk memproses *file* XML/XHTML/HTML5 menjadi *file* dengan format tersebut dengan utuh. Thymeleaf ini mempermudah *developer* untuk mengkonstruksi *template* untuk *view* dan dapat menggunakan secara ulang *template* tersebut untuk kebutuhan lain (*reusable*).

2. Tutorial

Pertanyaan 1:

Value yang dihasilkan pada `${iterationStatus.odd}` adalah nilai *boolean* untuk setiap *index array* yang memiliki nilai ganjil misal 1,3,5,... dst (*index array* dimulai dari 0).

Pertanyaan 2:

Pada konteks kondisi yang sama, `th:if` dan `th:unless` merupakan sebuah kondisional yang saling berkebalikan, jika kondisi `th:if` bernilai *true* maka `th:unless` bernilai *false* dan juga sebaliknya. Dapat disebutkan bahwa kondisi pada `th:unless` merupakan negasi dari kondisi `th:if` pada kondisi yang sama. Hal ini layaknya pada *condition statement* pada Java yaitu *if* dan *else*. Kondisi pada `th:unless` dapat juga berbeda dengan `th:if` layaknya *if* dan *else if*.

Pertanyaan 3:

```
The value of attribute "th:if" associated with an element type "h3" must not contain the '<' character.
xerces.internal.util.ErrorHandlerWrapper.createSAXParseException(ErrorHandlerWrapper.java:203) ~[na:1.8.0_
xerces.internal.util.ErrorHandlerWrapper.fatalError(ErrorHandlerWrapper.java:177) ~[na:1.8.0_161]
xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:400) ~[na:1.8.0_161]
xerces.internal.impl.XMLErrorReporter.reportError(XMLErrorReporter.java:327) ~[na:1.8.0_161]
xerces.internal.impl.XMLScanner.reportFatalError(XMLScanner.java:1472) ~[na:1.8.0_161]
```

Terjadi *error* karena pada `th:if` ataupun `th:unless` tidak mendukung karakter “<” untuk *conditional expression*.

Pertanyaan 4:

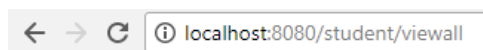
```
<h3 th:text="${student.gpa} >= 3.49 ? 'Cum Laude!' : 'Sangat Memuaskan!'">Status GPA</h3>
```

Cara lain seperti menggunakan *ternary operator* (`?:`). *Ternary operator* ini layaknya *if* dan *else*, “?” merupakan kondisi jika bernilai *true* dan “:” jika kondisi bernilai *false* dan disebut sebagai kondisi *default*. Namun penggunaan operator tersebut tidak digunakan pada `th:if` ataupun `th:unless` melainkan pada `th:text` untuk menentukan secara langsung status yang akan dikeluarkan.

Pertanyaan 5:

`th:replace="fragments/fragment::header"` dan `th:replace="fragments/fragment::footer"` adalah untuk menentukan *header* dan *footer* pada *index.html* dari *file template fragment* pada *folder* “fragments” dan nama *file fragment (layout)* “fragment”.

Pertanyaan 6:



500 Internal Server Error!

```
tutorial5.1606954810 - ApapTutorial04Application [Spring Boot App] C:\Program Files (x86)\Java\jdk1.8.0_161\bin\javaw.exe (Mar 14, 2018, 1:25:
2018-03-14 13:25:58.455 ERROR 15244 --- [nio-8080-exec-6] o.a.c.c.l.[.l./].[dispatcherServlet] : Servlet
org.thymeleaf.exceptions.TemplateProcessingException: Could not parse as expression: "${student.gpa}>=3.49
at org.thymeleaf.standard.expression.StandardExpressionParser.parseExpression(StandardExpressionP
at org.thymeleaf.standard.expression.StandardExpressionParser.parseExpression(StandardExpressionP
at org.thymeleaf.standard.expression.StandardExpressionParser.parseExpression(StandardExpressionP
at org.thymeleaf.standard.processor.attr.AbstractStandardTextChildModifierAttrProcessor.getText(Al
at org.thymeleaf.processor.attr.AbstractTextChildModifierAttrProcessor.getModifiedChildren(Abstra
at org.thymeleaf.processor.attr.AbstractChildrenModifierAttrProcessor.processAttribute(AbstractCh
at org.thymeleaf.processor.attr.AbstractAttrProcessor.doProcess(AbstractAttrProcessor.java:87) ~[
at org.thymeleaf.processor.AbstractProcessor.process(AbstractProcessor.java:212) ~[thymeleaf-2.1.
at org.thymeleaf.dom.Node.applyNextProcessor(Node.java:1017) ~[thymeleaf-2.1.5.RELEASE.jar:2.1.5.1
at org.thymeleaf.dom.Node.processNode(Node.java:972) ~[thymeleaf-2.1.5.RELEASE.jar:2.1.5.RELEASE]
```

Iya bisa, 500 internal server error muncul ketika suatu program yang sudah berjalan atau sudah di *compile* terjadi suatu *error*.

3. Soal

Soal 1

Pengubahan tampilan daftar *student* pada halaman *viewall* ini saya menggunakan *table* biasa tanpa *library* DataTable dan style yang digunakan menggunakan Bootstrap.

```
<div class="container">..
```

Pertama, pada halaman *viewall* menambahkan sebuah tag `<div>` dan mengimplementasikan atribut *class* "*container*".

```
<div class="container">
  <table class="table table-striped">..
</div>
```

Didalam `<div>` tersebut membuat tag `<table>` untuk menampilkan tabel dan mendefinisikan *class* "*table table-striped*". Definisi *class* tersebut berfungsi untuk mengimplementasikan style CSS dari Bootstrap.

```
<div class="container">
  <table class="table table-striped">
    <thead>
      <tr>
        <th scope="col">No.</th>
        <th scope="col">NPM</th>
        <th scope="col">Name</th>
        <th scope="col">GPA</th>
        <th scope="col">Status</th>
        <th scope="col">Options</th>
      </tr>
    </thead>
    <tbody>..
  </table>
</div>
```

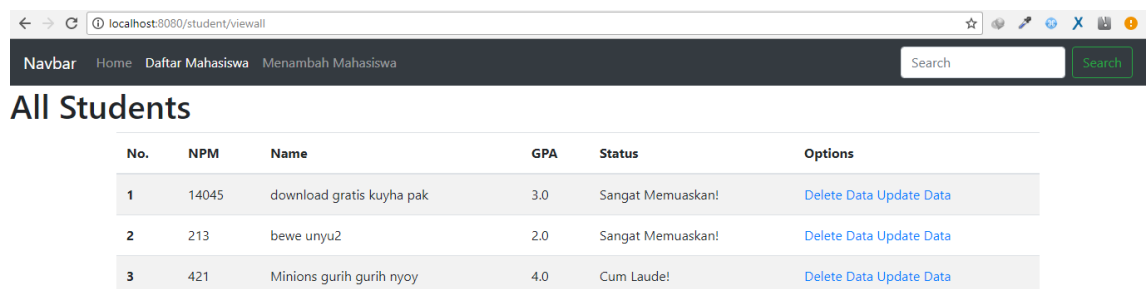
Lalu didalam tag `<table>` tersebut membuat tag `<thead>`, ini untuk mendefinisikan penamaan masing-masing kolom yang akan dibuat (*header column*). Didalam tag `<thead>` membuat kode `<tr>` dan `<th>`, `<tr>` untuk mendefinisikan tabel perbaris, `<th>` untuk mendefinisikan tabel perkolom. Pada tag `<th>` mendefinisikan penamaan dari masing-masing kolom tersebut sesuai data yang akan ditampilkan pada tabel tersebut.

```

<table class="table table-striped">
  <thead>
  <tbody>
    <tr th:each="student, iterationStat: ${students}">
      <th scope="row" th:text="${iterationStat.count}">1</th>
      <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!'">Status
        GPA</td>
      <td><a th:href="/student/delete/" + ${student.npm}">Delete
        Data</a> <a th:href="/student/update/" + ${student.npm}">Update
        Data</a></td>
    </tr>
  </tbody>
</table>

```

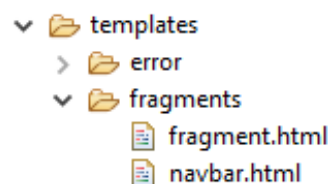
Masih dalam tag <table> menambahkan tag <tbody> untuk mendefinisikan badan dari tabel tersebut. Tag <tbody> ini dimana untuk mendefinisikan isi data dari tabel yang akan ditampilkan. Sama layaknya <tr> dan <td> pada <thead>, <tr> disini akan menggunakan atribut dari Thymeleaf dimana untuk memproses suatu iterasi berdasarkan data yang ada.



No.	NPM	Name	GPA	Status	Options
1	14045	download gratis kuyha pak	3.0	Sangat Memuaskan!	Delete Data Update Data
2	213	bewe unyu2	2.0	Sangat Memuaskan!	Delete Data Update Data
3	421	Minions gurih gurih nyoy	4.0	Cum Laude!	Delete Data Update Data

Berikut adalah hasil tampilan implementasi kode diatas.

Soal 2



Pada pembuatan *fragment header* dan *footer*, langkah pertama adalah membuat sebuah halaman HTML dengan nama “navbar.html” (penamaan halaman HTML bebas). Pada “navbar.html” ini terdapat atribut *header* dan *footer* untuk dapat diimplementasikan pada halaman lain.

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <link th:replace="index :: script" />
</head>
<body>
  <div th:fragment="navbar(selection)">
  <div th:fragment="footer">
    <hr />
    <h4>Mata Kuliah APAP</h4>
  </div>
</body>
</html>

```

Pada “navbar.html” mendefinisikan sebuah kode HTML dan atribut th:fragment untuk melakukan pendefinisian nama dari *fragment* tersebut. Membuat dua atribut th:fragment pada halaman tersebut, dimana th:fragment=“navbar(selection)” merupakan sebuah *fragment header* untuk menampilkan *header* halaman berupa *navbar*, th:fragment=“footer” merupakan sebuah *fragment footer* untuk menampilkan *footer* halaman.

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<link th:replace="index :: script" />
</head>
<body>
<div th:fragment="navbar(selection)">
<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">
</div>
</nav>
</div>
<div th:fragment="footer">
</div>
</body>
</html>
```

```
<div th:fragment="navbar(selection)">
<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">
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
<li th:class="${selection == 'home'} ? 'active' : null"
class="nav-item"><a class="nav-link" href="/">Home</a></li>
<li th:class="${selection == 'viewall'} ? 'active' : null"
class="nav-item"><a class="nav-link" href="/student/viewall">Daftar
Mahasiswa</a></li>
<li th:class="${selection == 'add'} ? 'active' : null"
class="nav-item"><a class="nav-link" 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</butt
</form>
</div>
</nav>
</div>
```

Didalam tag <div th:fragment=“navbar(selection)”> ini menambahkan sebuah tag <nav>, ini berfungsi untuk menampilkan sebuah *navbar*. Dan didalam tag <nav> tersebut menambahkan sebuah menu sesuai dengan soal yang diberikan.

```
<!doctype HTML>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head th:fragment="common_header(content)">
<title>Index</title>
<link th:fragment="script" rel="stylesheet"
href="/bootstrap-4.0.0-dist/css/bootstrap.min.css" />
</head>
<body>
<div th:replace="fragments/navbar :: navbar('home')"></div>
<center>
<h3>Welcome To Tutorial 05 APAP</h3>
</center>
<div th:replace="fragments/navbar :: footer"></div>
</body>
</html>
```

Setelah membuat “navbar.html”, selanjutnya mengimplementasikan *fragment* yang telah dibuat pada halaman tersebut. Dalam menggunakan *fragment* yang telah dibuat dengan

menggunakan atribut `th:replace`. Atribut `th:replace="fragments/navbar :: navbar('home')"` ini untuk mengimplementasikan *fragment* dengan nama "navbar" pada halaman navbar, atribut `th:replace="fragments/navbar :: footer"` untuk mengimplementasikan *fragment* dengan nama "footer" pada halaman navbar. Implementasi diatas pada halaman `index.html`. *Fragment* tersebut diimplementasikan juga pada halaman lainnya seperti halaman *viewall*, *add*, dll.



Mata Kuliah APAP



All Students

No.	NPM	Name	GPA	Status	Options
1	14045	download gratis kuyha pak	3.0	Sangat Memuaskan!	Delete Data Update Data
2	213	bewe unyu2	2.0	Sangat Memuaskan!	Delete Data Update Data
3	421	Minions gurih gurih nyoy	4.0	Cum Laude!	Delete Data Update Data

Mata Kuliah APAP



Problem Editor

NPM

Name

GPA

Mata Kuliah APAP

Berikut adalah hasil implementasi *fragment* pada halaman `index.html`, `viewall.html` dan `form-add.html`.