

Tutorial 05 – Thymeleaf (Presentation Layer)

Tutorial ini mengasumsikan penggunaan pada *Windows*, tanpa melarang penggunaan OS lain, silakan menyesuaikan.

1. Pendahuluan

- Jalankan STS
- Buat *project* baru:
 - o Nama : tutorial-05
 - o Package: com.apap.tu05
 - o *Dependencies*: DevTools, Thymeleaf, Web, JPA, dan MySQL
- Menyelesaikan (sebagian) tutorial 04
Pada tutorial ini Anda akan diminta melakukan berbagai perubahan terutama pada bagian view dari tutorial 04. Silakan *copy* seluruh kode yang sudah ada kedalam project tutorial-05.

2. Looping, iterationStatus

- Pada tutorial 04 Anda sudah menggunakan ekspresi *iteration* pada salah satu halaman, misalnya view-pilot.html. Pada halaman tersebut misalnya terdapat baris kode sbb.:

```
<div th:each="flight : ${listOfFlight}">
  <h3 th:text="${flight.id}"></h3>
  <h3 th:text="${flight.flightNumber}"></h3>
  <h3 th:text="${flight.origin}"></h3>
  <h3 th:text="${flight.destination}"></h3>
  <h3 th:text="${flight.time}"></h3>
</div>
```

- Ubah baris kode tersebut menjadi sbb.:

```
<table>
  <tbody>
    <tr th:each="flight, iterationStatus : ${listOfFlight}" th:style="${iterationStatus.even} ? 'display:none;' ">
      <td th:text="${iterationStatus.count}"></td>
      <td th:text="${flight.flightNumber}"></td>
      <td th:text="${flight.origin}"></td>
      <td th:text="${flight.destination}"></td>
      <td><a th:href="@{'/flight/view/' + ${flight.flightNumber}}">See Flight Detail</a></td>
    </tr>
  </tbody>
</table>
```

- Jalankan program lalu akses pada browser. Amati keluaran pada browser.

Attribute/field pada *iterationStatus* antara lain:

- o Index : index iterasi, dimulai dari 0 (nol)
- o count : banyaknya element yang telah diproses
- o size : jumlah total element yang ada pada list
- o even/odd: cek index, apakah genap atau ganjil
- o first : cek apakah element yang sedang diproses merupakan element pertama pada list
- o last : cek apakah element yang sedang diproses merupakan element terakhir pada list

3. Conditional Expression

- Pada halaman view-pilot tambahkan baris kode sbb.:

```
<h3 th:if="${pilot.flyHour} >= 100">Status: Pilot Senior</h3>
<h3 th:unless="${pilot.flyHour} >= 100">Status: Pilot Junior</h3>
```

- Atau dapat pula dilakukan dengan cara sbb.:

```
<h3 th:text="${pilot.flyHour} >= 100 ? 'Status: Pilot Senior' : 'Status: Pilot Junior'"></h3>
```

- Jalankan program lalu akses pada browser. Amati keluaran pada browser.

4. Static File (css, javascript, image, dll.)

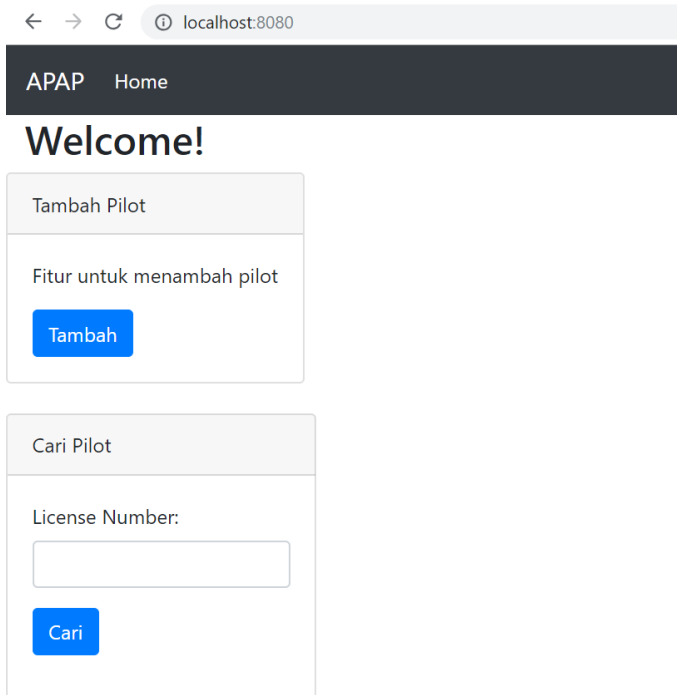
Pada bagian ini Anda akan mempelajari implementasi css bootstrap sebagai *static file* yang dapat digunakan pada Thymeleaf.

- Unduh bootstrap pada link berikut: <https://getbootstrap.com/>
- Ekstrak folder zip yang anda unduh ke dalam folder static (src\main\resources\static).
- Ubahlah halaman home.html menjadi sbb.:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Home</title>
    <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
    <script th:src="@{/js/bootstrap.min.js}"></script>
  </head>
  <body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a class="navbar-brand" href="#">APAP</a>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="#">Home</a>
          </li>
        </ul>
      </div>
    </nav>

    <div class="container-fluid">
      <h2>Welcome!</h2>
      <div class="row">
        <div class="col-md-6">
          <div class="card">
            <div class="card-header">Tambah Pilot</div>
            <div class="card-body">
              <p class="card-text">Fitur untuk menambah pilot</p>
              <a class="btn btn-primary" th:href="@{/pilot/add}">Tambah</a>
            </div>
          </div>
        </div>
      </div>
      <br>
      <div class="row">
        <div class="col-md-6">
          <div class="card">
            <div class="card-header">Cari Pilot</div>
            <form th:action="@{/pilot/view}" method="GET">
              <div class="card-body">
                <label for="licenseNumber">License Number: </label>
                <input type="text" name="licenseNumber" class="form-control"/>
                <p></p>
                <button type="submit" class="btn btn-primary">Cari</button>
              </div>
            </form>
          </div>
        </div>
      </div>
    </div>
  </body>
</html>
```

- Akses pada browser. Contoh tampilannya sbb.:



The screenshot shows a web browser window with the address bar set to 'localhost:8080'. The page has a dark header with 'APAP' and 'Home' links. Below the header, there's a large 'Welcome!' heading. The main content area contains two white boxes. The first box, titled 'Tambah Pilot', has a subtitle 'Fitur untuk menambah pilot' and a blue 'Tambah' button. The second box, titled 'Cari Pilot', has a subtitle 'License Number:', a text input field, and a blue 'Cari' button.

5. Fragment

Fragment merupakan sebuah fitur dalam Thymeleaf yang memungkinkan developer untuk *re-use source code* dari halaman HTML-nya. Fitur ini memungkinkan Anda untuk memecah halaman HTML menjadi bagian-bagian kecil. Bagian tersebut lalu dapat digunakan kembali di 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*.

- Buat sebuah folder dengan nama **fragments** pada folder templates.
- Pada folder tersebut buatlah sebuah halaman HTML dengan nama `fragment.html` dengan spesifikasi sbb.:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
  <head>
    <title>Home</title>
    <head th:fragment="css">
      <link rel="stylesheet" th:href="@{/css/bootstrap.min.css}">
    </head>
    <head th:fragment="js">
      <script th:src="@{/js/bootstrap.min.js}"></script>
    </head>
  </head>

  <body>
    <nav th:fragment="navbar" class="navbar navbar-expand-lg navbar-dark bg-dark">
      <a class="navbar-brand" href="#">APAP</a>
      <div class="collapse navbar-collapse" id="navbarSupportedContent">
        <ul class="navbar-nav mr-auto">
          <li class="nav-item active">
            <a class="nav-link" href="#">Home</a>
          </li>
        </ul>
      </div>
    </nav>
  </body>
</html>
```

- Ubahlah sebagian kode home.html menjadi seperti dibawah ini:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Home</title>
        <object th:include="fragments/fragment :: css" th:remove="tag"></object>
        <object th:include="fragments/fragment :: js" th:remove="tag"></object>
    </head>
    <body>
        <nav th:replace="fragments/fragment :: navbar"></nav>

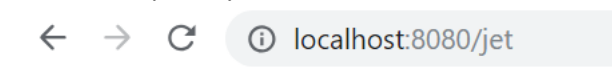
        <div class="container-fluid">
            <h2>Welcome!</h2>
```

6. Error 404 Not Found Handler

- Buat sebuah folder dengan nama **error** pada folder templates.
- Pada folder tersebut buatlah sebuah halaman HTML dengan nama 404.html dengan spesifikasi sbb.:

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>404 Not Found</title>
    </head>
    <body>
        <h2>Halaman tidak ditemukan!</h2>
    </body>
</html>
```

- Akses halaman yang tidak tersedia pada aplikasi Anda, misalnya <http://localhost:8080/jet>. Contoh tampilannya sbb.:



Halaman tidak ditemukan!

- Buatlah sebuah halaman error untuk Error 500 Internal Server Error.

7. Form Handler

Pada bagian ini, anda akan mempelajari bagaimana Thymeleaf meng-handle checkbox.

- Pada PilotController ubahlah route “/pilot/view” menjadi sbb.:

```
@RequestMapping(value = "/pilot/view", method = RequestMethod.GET)
public String viewPilot(@RequestParam("licenseNumber") String licenseNumber, Model model) {
    PilotModel pilot = pilotService.getPilotDetailByLicenseNumber(licenseNumber);
    model.addAttribute("pilot", pilot);
    return "view-pilot";
}
```

- Lalu pada FlightController ubahlah route “/flight/delete” menjadi sbb.:

```
@RequestMapping(value = "/flight/delete", method = RequestMethod.POST)
private String deleteFlight(@ModelAttribute PilotModel pilot, Model model) {
    for(FlightModel flight : pilot.getPilotFlight()) {
        flightService.deleteFlightById(flight.getId());
    }
    return "delete";
}
```

- Ubahlah view-pilot.html menjadi sbb.:

```
<body>
  <h3 th:text="'Pilot License Number: ' + ${pilot.licenseNumber}"></h3>
  <h3 th:text="'Name: ' + ${pilot.name}"></h3>
  <h3 th:text="'Fly Hour: ' + ${pilot.flyHour}"></h3>
  <h3 th:text="'${pilot.flyHour} >= 100 ? 'Status: Pilot Senior' : 'Status: Pilot Junior'"></h3>

  <form th:action="@{/pilot/update/} + ${pilot.id}">
    <button type="submit">Update Pilot</button>
  </form>
  <form th:action="@{/pilot/delete/} + ${pilot.id}">
    <button type="submit">Delete Pilot</button>
  </form>

  <form th:action="@{/flight/delete}" th:object="${pilot}" method="POST">
    <table border="1">
      <caption>Flights</caption>
      <thead>
        <tr>
          <th style="text-align: center">No.</th>
          <th style="text-align: center">Flight Number</th>
          <th style="text-align: center">Origin</th>
          <th style="text-align: center">Destination</th>
          <th style="text-align: center">Time</th>
          <th style="text-align: center">Action</th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="flight, iterationStatus : ${pilotFlight}">
          <td th:text="${iterationStatus.count}"></td>
          <td th:text="${flight.flightNumber}"></td>
          <td th:text="${flight.origin}"></td>
          <td th:text="${flight.destination}"></td>
          <td th:text="${flight.time}"></td>
          <td>
            <input type="checkbox" th:field="*{pilotFlight}" th:value="${flight.id}" />
            <label th:for="${#ids.prev('pilotFlight')}">Hapus</label>
          </td>
        </tr>
      </tbody>
    </table>
    <button type="submit">Delete Selected Flight</button>
  </form>
  <br>
  <form th:action="@{/flight/add/} + ${pilot.licenseNumber}">
    <button value="submit">Add Flight</button>
  </form>
</body>
```

- Lakukanlah penghapusan dengan menggunakan checkbox.

Pilot License Number: 001

Name: Abka

Fly Hour: 100

Status: Pilot Senior

Update Pilot
Delete Pilot

Flights

No.	Flight Number	Origin	Destination	Time	Action
1	4	Bali	Bandung	2019-04-20	<input checked="" type="checkbox"/> Hapus
2	1	Bandung	Bali	2019-04-24	<input checked="" type="checkbox"/> Hapus

Delete Selected Flight

Add Flight

Latihan

1. Buatlah *fragment navigation bar* sehingga yang semula hanya bertuliskan “APAP” menjadi berubah-ubah (dinamis) sesuai dengan nama halaman yang sedang diakses. Pastikan Anda mengimplementasikannya pada setiap *view* yang ada pada *project*.
2. Anda telah mempelajari cara meng-*handle multiple* checkbox untuk dapat melakukan banyak *delete* sekaligus. Pada latihan ini anda diminta untuk mengimplementasi banyak *insert* untuk flight. Contoh implementasinya sbb.:
 - Terdapat tombol untuk menambah field/form untuk *object* flight lainnya.

Flight Number	Origin	Destination	Time	Add Flight
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="dd/mm/yyyy"/>	
<input type="button" value="Submit"/>				

- Jika tombol tersebut (Add Flight) ditekan, maka muncul field/form untuk insert flight lainnya.

Flight Number	Origin	Destination	Time	Add Flight
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="dd/mm/yyyy"/>	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="dd/mm/yyyy"/>	
<input type="button" value="Submit"/>				

Hint: Anda dapat mencontoh implementasi *dynamic field* pada alamat berikut:

<https://www.thymeleaf.org/doc/tutorials/3.0/thymeleafspring.html#dynamic-fields>

Pengumpulan

1. *Screen capture* pengerjaan latihan yang menunjukkan bahwa fungsi berhasil diimplementasi (output pada browser, tulisan navbar sesuai dengan alamat yang diakses untuk semua *view*). Pengumpulan dituliskan dalam satu file dengan format **npm_nama-lengkap-anda.pdf** dan unggah ke submission slot yang disediakan di Scele.
2. Folder tutorial-05 dikumpulkan dengan cara push ke GitHub (<https://github.com/achmad-f-abka/apap>).