

Latihan

1. Pada tutorial, Anda sudah mencoba membuat fragment navigation bar. Tetapi, navigation bar pada tutorial tersebut tidak dinamis. Data HTML pada header akan sama di setiap view yang memakai fragment tersebut. Pada latihan kali ini, Anda diminta untuk membuat sebuah fragment navigation bar yang dinamis. Dimana, aspek dinamis tersebut terdapat pada title navbar yang sebelumnya hanya bertuliskan “APAP” akan berubah sesuai dengan nama page yang sedang diakses. Silahkan implementasi dengan kebutuhan tersebut, dan jika sudah berhasil, pastikan Anda mengimplementasikannya pada setiap view yang ada di project ini.

- a. Pada fragment.html bagian navbar ditambahkan attribute “title”. Sebelumnya navbar bersifat statis yang akan selalu memunculkan “APAP” pada headernya. Namun karena telah menambahkan attribute “title” pada navbar dan menambahkan “th:text=“\${title}”, text APAP akan tertimpa dengan title.

```
15 <body>
16 <nav th:fragment="navbar(title)" class="navbar navbar-expand-lg navbar-dark bg-dark">
17 <a class="navbar-brand" href="#" th:text="${title}">APAP</a>
```

- b. Masing-masing html ditambahkan attribute title yang disesuaikan dengan masing-masing halaman.

add.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Add')"></nav>
```

addFlight.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Add Flight')"></nav>
```

addPilot.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Add Pilot')"></nav>
```

delete.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Delete')"></nav>
```

home.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Home')"></nav>
```

update.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Update')"></nav>
```

updateFlight.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Update Flight')"></nav>
```

updatePilot.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'Update Pilot')"></nav>
```

view-flights.html

```
<body>
<nav th:replace="fragments/fragment :: navbar(title= 'View Flights')"></nav>
```

view-pilot.html

CHANADYA
1606917821
APAP – C

```
<body>  
  <nav th:replace="fragments/fragment :: navbar(title= 'View Pilot')"></nav>
```

- c. Pada saat diakses, akan muncul tampilan seperti berikut:

Add Pilot

The screenshot shows a web browser at localhost:8080/pilot/add. The page has a dark header with 'Add Pilot' and 'Home' links. Below the header, there is a large 'Hello!' greeting followed by the title 'Tambah Pilot'. There are three input fields labeled 'License Number:', 'Name:', and 'Fly Hour:'. At the bottom, there is a 'Submit' button.

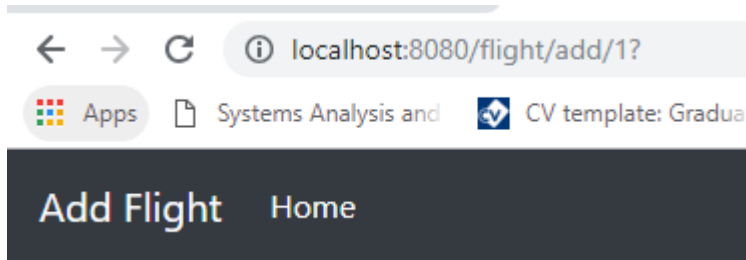
View Pilot

The screenshot shows a web browser at localhost:8080/pilot/view?licenseNumber=1. The page has a dark header with 'View Pilot' and 'Home' links. Below the header, the pilot details are displayed: 'Pilot License Number : 1', 'Name : Mayor Kombi', 'Fly Hour : 100', and 'Status : Pilot Senior'. There are two buttons: 'Update Pilot' and 'Delete Pilot'. Below these is a table with flight data:

No.	Flight Number	Origin	Destination	Time	Action
1	GA123	JKT	DPS	2018-10-16	<input checked="" type="checkbox"/> Hapus
2	GA122	PDG	JKT	2018-10-16	<input checked="" type="checkbox"/> Hapus
3	GA122	JKT	PDG	2018-10-17	<input checked="" type="checkbox"/> Hapus

Below the table, there is a 'Flights' section with a 'Delete Selected Flight' button and an 'Add Flight' button.

Add Flight



2. Pada bagian form handler, anda telah mempelajari cara meng-handle input multiple checkbox untuk dapat melakukan delete banyak flight sekaligus. Pada latihan ini, Anda diminta untuk dapat melakukan insert banyak flight sekaligus.

- a. Pada FlightController tambahkan method addRow

```
@RequestMapping(value = "/flight/add/{licenseNumber}", params= {"addRow"})
public String addRow(@ModelAttribute PilotModel pilot, BindingResult bindingResult ,Model model) {
    if(pilot.getPilotFlight() ==null) {
        pilot.setPilotFlight(new ArrayList<FlightModel>());
    }
    FlightModel newFlight = new FlightModel();
    pilot.getPilotFlight().add(newFlight);
    model.addAttribute("pilot",pilot);
    return "addFlight";
}
```

Untuk pilot yang tidak memiliki flight, akan dibuatkan list kosong.

- b. Dilakukan perubahan pada method add di FlightController.

```
@RequestMapping(value = "/flight/add/{licenseNumber}", method = RequestMethod.GET)
private String add(@PathVariable(value = "licenseNumber") String licenseNumber, Model model) {
    PilotModel pilot = pilotService.getPilotDetailByLicenseNumber(licenseNumber);
    List<FlightModel> flightList = new ArrayList<>();
    pilot.setPilotFlight(flightList);
    flightList.add(new FlightModel());
    model.addAttribute("pilot", pilot);
    return "addFlight";
}
```

- c. Pada flightController ditambahkan method addFlightSubmit

```
@RequestMapping(value = "/flight/add/{licenseNumber}", method = RequestMethod.POST, params={"save"})
private String addFlightSubmit(@ModelAttribute PilotModel pilot, Model model) {
    PilotModel archive = pilotService.getPilotDetailByLicenseNumber(pilot.getLicenseNumber());
    for(FlightModel flight : pilot.getPilotFlight()) {
        flight.setPilot(archive);
        flightService.addFlight(flight);
    }
    return "add";
}
```

Looping disini disesuaikan dengan banyaknya row yang ditambahkan user.

- d. Dilakukan modifikasi pada addFlight.html

```
<h3>Tambah Penerbangan</h3>
<form th:action="@{'/flight/add/' + ${pilot.licenseNumber}}" th:object="${pilot}" method="POST">
  <input type="hidden" th:field="*{licenseNumber}"></input>
  <table>
    <thead>
      <tr>
        <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>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td><input type="text" th:field="*{pilotFlight[__${iterationStatus.index}__].flightNumber}" /></td>
        <td><input type="text" th:field="*{pilotFlight[__${iterationStatus.index}__].origin}" /></td>
        <td><input type="text" th:field="*{pilotFlight[__${iterationStatus.index}__].destination}" /></td>
        <td><input type="date" th:field="*{pilotFlight[__${iterationStatus.index}__].time}" /></td>
      </tr>
    </tbody>
  </table>
  <button type="submit" name="addRow">Add row</button>
</form>
<button type="submit" name="save">Submit</button>
```

e. Tampilan akan muncul seperti ini

localhost:8080/flight/add/1

Apps Systems Analysis and CV template: Gradual add

Add Flight Home

Tambah Penerbangan

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

Submit

Add row