

Latihan

1. Ubah method view pilot agar menampilkan daftar flight yang terdaftar untuk setiap pilot.

- File PilotController

```
@RequestMapping(value = "/pilot/view", method = RequestMethod.GET)
private String view(@RequestParam("licenseNumber") String licenseNumber, Model model) {
    PilotModel archive= pilotService.getPilotDetailByLicenseNumber(licenseNumber);

    List<FlightModel> flightList = archive.getPilotFlight();

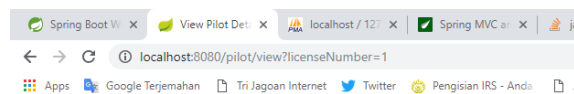
    model.addAttribute("flightList", flightList);
    model.addAttribute("pilot", archive);

    return "view-pilot";
}
```

- File View Pilot

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <title>View Pilot Detail</title>
5 </head>
6 <body>
7 <p>Pilot License Number : </p>
8 <p th:text="${pilot.licenseNumber}"></p>
9 <br>
10 <p>Name : </p>
11 <p th:text="${pilot.name}"></p>
12 <br>
13 <p>Fly Hour : </p>
14 <p th:text="${pilot.flyHour}"></p>
15 <br>
16 <a th:href="@{'/pilot/updatePilot/' + ${pilot.licenseNumber}}">Update Pilot</a>
17
18 <div th:each="flightList : ${flightList}">
19 <p>Flight Number : </p>
20 <h3 th:text="${flightList.flightNumber}"></h3>
21 <p>Origin : </p>
22 <h3 th:text="${flightList.origin}"></h3>
23 <p>Destination : </p>
24 <h3 th:text="${flightList.destination}"></h3>
25 <p>Time : </p>
26 <h3 th:text="${flightList.time}"></h3>
27 <td><a th:href="@{'/flight/updateFlight/' + ${flightList.id}}">Update Flight</a></td>
28 <td><a th:href="@{'/flight/deleteFlight/' + ${flightList.id}}">Delete Flight</a></td>
29 </div>
30
31 <td><a th:href="@{'/flight/add/' + ${pilot.licenseNumber}}">Add Flight</a></td>
32 </body>
33 </html>
```

- Result



Pilot License Number : 1

Name : Rachel Sausan Nabila

Fly Hour : 8

[Update Pilot](#)

All Flights

[Add Flight](#)

Flight Number : 000

Origin : jjjj

Destination : d

Time : 2018-10-26

[Update Flight](#) [Delete Flight](#)

Flight Number : 33

Origin : ori

Destination : gg

Time : 2018-10-04

[Update Flight](#) [Delete Flight](#)

2. Buatlah fitur delete untuk menghapus seorang pilot dan sebuah penerbangan.

Menghapus pilot

- File PilotService

```
PilotServic... PilotModel.java PilotControl... view-pilot.html FlightContr...
1 package com.apap.tutorial4.service;
2
3 import java.util.List;
4
5
6
7 public interface PilotService {
8     void addPilot(PilotModel pilot);
9     PilotModel getPilotDetailByLicenseNumber(String licenseNumber);
10    void deletePilot (PilotModel pilot) ;
11    List <PilotModel> getListPilot();
12    void updatePilot(PilotModel pilot, String licenseNumber);
13
14 }
15
```

- File PilotServiceImpl

```
@Override
public void deletePilot(PilotModel pilot) {
    // TODO Auto-generated method stub
    pilotDb.delete(pilot);
}
}
```

- File PilotController

```
@RequestMapping(value = "/pilot/deletePilot", method = RequestMethod.GET)
private String deletePilot(@RequestParam("licenseNumber") String licenseNumber, Model model) {
    PilotModel archive= pilotService.getPilotDetailByLicenseNumber(licenseNumber);

    pilotService.deletePilot(archive);
    return "delete";
}
```

- Result (hapus license no = 1)

Hello! This is HOME

[Add Pilot View All Flights](#)

Cari Pilot berdasarkan Nomor Lisensi

License Number:

Cari

Hapus Pilot berdasarkan Nomor Lisensi

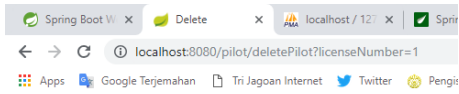
License Number:

Hapus

Update Pilot berdasarkan Nomor Lisensi

License Number:

Update



Data Berhasil Dihapus

[Back to Home](#)

Menghapus penerbangan

- File FlightService

```
1 package com.apap.tutorial4.service;
2
3 import java.util.List;
4
5
6
7 public interface FlightService {
8     void addFlight(FlightModel flight);
9     FlightModel getFlightDetailByFlightNumber(String flightNumber);
10    FlightModel getFlightDetailById(long id);
11    void deleteFlight (FlightModel flight);
12    List <FlightModel> getFlightList();
13    void updateFlight(FlightModel flight, long id);
14
15 }
16
```

- File FlightServiceImpl

```
@Override
public void deleteFlight(FlightModel flight) {
    // TODO Auto-generated method stub
    flightDb.delete(flight);
}
```

- File FlightController

```
@RequestMapping(value = "/flight/deleteFlight/{id}", method = RequestMethod.GET)
private String deleteFlight(@PathVariable(value="id") String id, Model model) {
    long idd = Long.parseLong(id);
    FlightModel archive= flightService.getFlightDetailById(idd);

    flightService.deleteFlight(archive);
    return "deleteFlight";
}
```

- Result (Delete Flight 000)

Pilot License Number : 1

Name : Rachel Sausan Nabila

Fly Hour : 8

[Update Pilot](#)

All Flights

[Add Flight](#)

Flight Number : 000

Origin : jjjj

Destination : d

Time : 2018-10-26

[Update Flight](#) [Delete Flight](#)

Flight Number : 33

Origin : ori

Destination : gg

Time : 2018-10-04

[Update Flight](#) [Delete Flight](#)

Pilot License Number : 1

Name : Rachel Sausan Nabila

Fly Hour : 8

[Update Pilot](#)

All Flights

[Add Flight](#)

Flight Number : 33

Origin : ori

Destination : gg

Time : 2018-10-04

[Update Flight](#) [Delete Flight](#)

3. Buatlah fitur update untuk seorang pilot dan sebuah penerbangan. (Notes: License Number seorang pilot tidak bisa diubah)

Untuk Pilot

- PilotServiceImpl

```
@Override
public void updatePilot(PilotModel pilot, String licenseNumber) {
    // TODO Auto-generated method stub
    PilotModel old = pilotDb.findByLicenseNumber(licenseNumber);
    old.setFlyHour(pilot.getFlyHour());
    old.setName(pilot.getName());
    old.setLicenseNumber(old.getLicenseNumber());
    old.setId(old.getId());
    pilotDb.save(old);
}
```

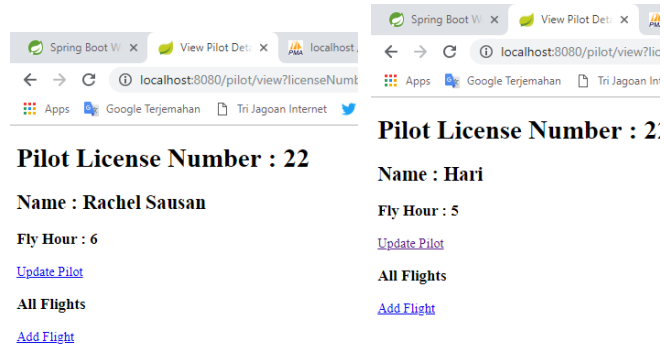
- Pilot Controller

```
@RequestMapping(value="/pilot/updatePilot/{licenseNumber}", method= RequestMethod.GET)
private String updatePilot(@PathVariable String licenseNumber, Model model) {
    PilotModel pilot = pilotService.getPilotDetailByLicenseNumber(licenseNumber);
    model.addAttribute("pilot", pilot);
    return "updatePilot";
}

@RequestMapping(value="/pilot/updatePilot/{licenseNumber}", method=RequestMethod.POST)
private String updatePilot(@PathVariable String licenseNumber, @ModelAttribute PilotModel pilot) {
    pilotService.updatePilot(pilot, licenseNumber);
    return "update";
}

}
```

- Result



Untuk Penerbangan

- FlightServiceImpl

```
@Override
public void updateFlight(FlightModel flight, long id) {
    FlightModel old = flightDb.findById(id);
    old.setFlightNumber(flight.getFlightNumber());
    old.setOrigin(flight.getOrigin());
    old.setDestination(flight.getDestination());
    old.setTime(flight.getTime());
    flightDb.save(old);
}

}
```

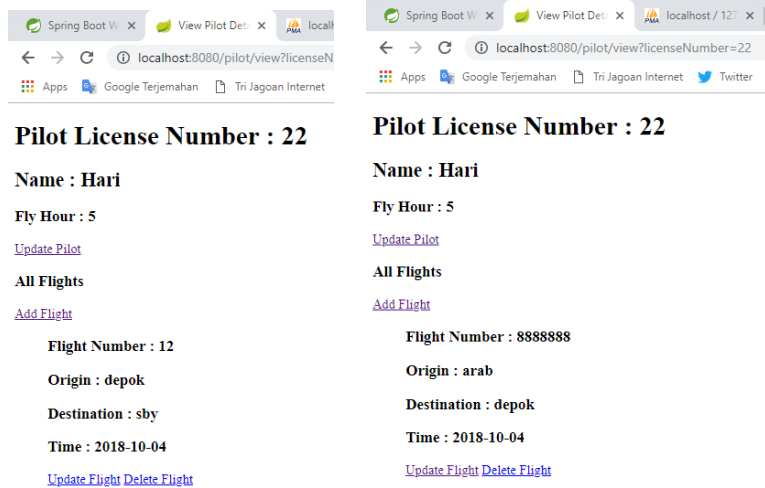
- FlightController

```
@RequestMapping(value="/flight/updateFlight/{id}", method= RequestMethod.GET)
private String updateFlight(@PathVariable (value="id") String id, Model model) {
    long idd = Long.parseLong(id);
    FlightModel flight = flightService.getFlightDetailById(idd);
    model.addAttribute("flight", flight);
    return "updateFlight";
}

@RequestMapping(value="/flight/updateFlight/{id}", method=RequestMethod.POST)
private String updateFlight(@PathVariable (value="id") String id, @ModelAttribute FlightModel flight) {
    long idd = Long.parseLong(id);
    flightService.updateFlight(flight, idd);
    return "update";
}

}
```

- Result



4. Buatlah fitur untuk melihat penerbangan sehingga dapat menampilkan nama dan license number dari pilot yang bertugas.

- PilotController

```
@RequestMapping(value = "/pilot/viewallflight/", method = RequestMethod.GET)
private String viewall(Model model) {
    List<PilotModel> pilotList = pilotService.getListPilot();

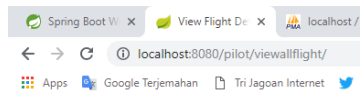
    model.addAttribute("pilotList", pilotList);

    return "viewall-flight";
}
```

- PilotServiceImpl

```
@Override
public List<PilotModel> getListPilot() {
    // TODO Auto-generated method stub
    return pilotDb.findAll();
}
```

- Result



All Flights

Pilot Name : Hari

Pilot licenseNumber : 22

Flight Number : 8888888

Origin : arab

Destination : depok

Time : 2018-10-04

Pilot Name : Hari

Pilot licenseNumber : 22

Flight Number : 122

Origin : gin

Destination : tion

Time : 2018-10-04