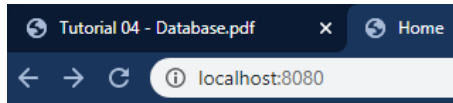


Nama : Izzatunnisa Arwadi

NPM : 1806269726

Kelas : Ekstensi

- Tampilan awal



**Welcome!**

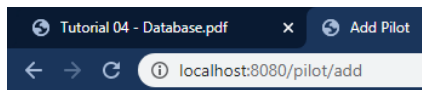
**Tambah Pilot**

[Tambah](#)

**Cari Pilot**

License Number:

- Tambah Pilot



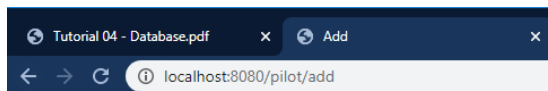
**Welcome!**

**Tambah Pilot**

License Number:

Name:

Fly Hour:



**Data Berhasil Ditambahkan!**

## Latihan

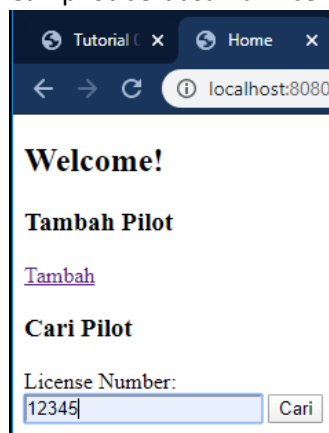
1. Implementasikan method di PilotController untuk menampilkan seorang pilot berdasarkan license number. Method ini menerima parameter licenseNumber dan mengembalikan view (**view-pilot.html**). Buatlah tampilan view-pilot.html minimal seperti gambar di bawah ini. Tambahkan juga link/tombol untuk menambahkan penerbangan pada bagian bawah view. Lalu lakukan tambah flight untuk pilot yang tampil.

Untuk melakukan implementasi di atas dilakukan penambahan method baru pada PilotController, method ini akan menampilkan data pilot berdasarkan license number yang dicari. Selain itu, dilakukan pembuatan file view-pilot.html

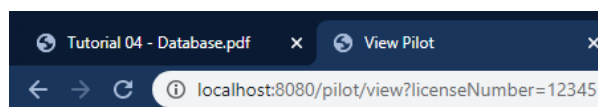
```
@RequestMapping("/pilot/view")
private String view(@RequestParam("licenseNumber") String licenseNumber, Model model) {
    PilotModel archive = pilotService.getPilotDetailByLicenseNumber(licenseNumber);
    if(archive==null) {
        return "no-data";
    }else {
        model.addAttribute("pilot", archive);
        return "view-pilot";
    }
}
```

Method di atas bertujuan untuk menampilkan data pilot dari license number yang dicari.

- Cari pilot berdasarkan license number



Dilakukan input berdasarkan license number



**Pilot License Number : 12345**

**Name : Caramia**

**Fly Hour : 3**

[Add Flight](#)

Akan ditampilkan data pilot berdasarkan license number yang dicari.

Untuk operation add flight, menggunakan method pada FlightController dan file addPilot.html yang telah dituliskan codenya pada tutorial.

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

@RequestMapping(value = "/flight/add", method = RequestMethod.POST)
private String addFlightSubmit(@ModelAttribute FlightModel flight) {
    flightService.addFlight(flight);
    return "add";
}
```

Code di atas digunakan untuk menambahkan data flight. Method pertama ditujukan untuk menampilkan form add serta terdapat pengesetan object dan addAttribute untuk flight. Method kedua digunakan untuk menambahkan data yang diinputkan ke dalam database

- Add Flight

**Welcome!**

### Tambah Penerbangan

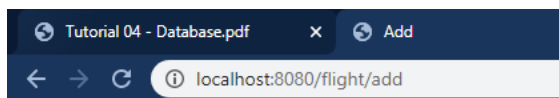
Flight Number:

Origin:

Destination:

Time:

Dilakukan penginputan data penerbangan berdasarkan license number.



### Data Berhasil Ditambahkan!

Ketika penambahan data penerbangan berhasil, ditampilkan notifikasi bahwa data berhasil ditambahkan.

2. Tambahkan pada view pilot agar menampilkan daftar flight dari pilot yang tampil (misalnya pada bagian bawah setelah link/tombol Add Flight)

Untuk menampilkan list penerbangan seorang pilot, diperlukan penambahan baris pada method view PilotController.

```

@RequestMapping("/pilot/view")
private String view(@RequestParam("licenseNumber") String licenseNumber, Model model) {
    PilotModel archive = pilotService.getPilotDetailByLicenseNumber(licenseNumber);
    if (archive == null) {
        return "no-data";
    } else {
        model.addAttribute("pilot", archive);
        model.addAttribute("flight", archive.getPilotFlight());
        return "view-pilot";
    }
}

```

Baris yang ditambahkan adalah `model.addAttribute("flight", archive.getPilotFlight())`. Code tersebut bertujuan untuk mendapatkan data flight sesuai dengan license number pilot.

Selain itu dilakukan penambahan baris pada `view-pilot.html`

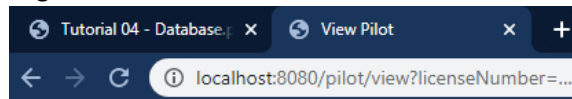
```

<h3>Flight List</h3>
<table border>
    <tr>
        <th>Flight Number</th>
        <th>Origin</th>
        <th>Destination</th>
        <th>Time</th>
    </tr>
    <tr th:each="flight : ${flight}">
        <td th:text="${flight.flightNumber}"></td>
        <td th:text="${flight.origin}"></td>
        <td th:text="${flight.destination}"></td>
        <td th:text="${flight.time}"></td>
    </tr>
</table>

```

Dilakukan perulangan untuk mengeluarkan isi dari masing-masing field pada flight.

#### - Flight List



**Pilot License Number : 12345**

**Name : Caramia**

**Fly Hour : 3**

[Add Flight](#)

#### **Flight List**

Flight Number	Origin	Destination	Time
101	Indonesia	Osaka	2020-02-01

Flight list akan ditampilkan berdasarkan data seorang pilot.

### 3. Buatlah fitur delete untuk menghapus seorang pilot dan sebuah flight.

Untuk menghapus flight diperlukan penambahan code pada `FlightDb`, `FlightService` hingga penambahan method pada `FlightController`. Hapus Flight ini memungkinkan untuk menghapus flight pada pilot.

```

@RequestMapping("/flight/delete/{flightNumber}")
private String delete(@PathVariable(value = "flightNumber") String flightNumber) {
    flightService.deleteFlight(flightNumber);
    return "delete";
}

```

Code di atas digunakan untuk memanggil method deleteFlight untuk memproses delete flight pada database berdasarkan flightNumber.

```

@Override
public void deleteFlight(String flightNumber) {
    flightDb.deleteByFlightNumber(flightNumber);
}

```

Perintah delete tersebut dipanggil di dalam FlightService.

```

public interface FlightDb extends JpaRepository<FlightModel, Long>{
    void deleteByFlightNumber(String flightNumber);
}

```

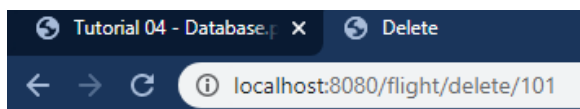
Pemanggilan method yang ada pada FlightService, sebelumnya telah dideklarasikan pada interface FlightDb.

- Hapus Flight

### Flight List

Flight Number	Origin	Destination	Time	Action
101	Indonesia	Osaka	2020-02-01	<a href="#">Delete</a>

Ketika ingin dilakukan perintah delete pada flight, klik action delete.



### Data Berhasil Dihapus!

Akan ditampilkan notifikasi bahwa data berhasil dihapus.

**Pilot License Number : 12345**

**Name : Caramia**

**Fly Hour : 3**

[Add Flight](#)

### Flight List

Flight Number	Origin	Destination	Time	Action
---------------	--------	-------------	------	--------

Ketika dilakukan view data pilot, maka data flight sudah tidak ada.

Untuk menghapus data pilot, otomatis data flight di dalamnya juga harus terhapus. Oleh karena itu dilakukan pengubahan pada PilotModel dengan tipe Cascade remove

```
@OneToMany(mappedBy = "pilot", fetch = FetchType.LAZY, cascade = CascadeType.REMOVE)
private List<FlightModel> pilotFlight;
```

Selain itu dilakukan penambahan isi code pada PilotDb, PilotService hingga penambahan method pada PilotController.

```
void deleteByLicenseNumber(String licenseNumber);
```

Dilakukan deklrasi method pada PilotDb.

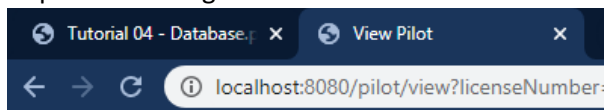
```
@Override
public void deletePilot(String licenseNumber) {
    pilotDb.deleteByLicenseNumber(licenseNumber);
}
```

Pada pilot service terdapat penggunaan method PilotDb dengan parameter licenseNumber.

```
@RequestMapping("/pilot/delete/{licenseNumber}")
private String delete(@PathVariable(value = "licenseNumber") String licenseNumber) {
    pilotService.deletePilot(licenseNumber);
    return "delete";
}
```

Pada pilotController dilakukan pemanggilan method pada PilotService dengan parameter licenseNumber. Method ini akan mengembalikan view delete.html

#### - Hapus Pilot & Flight



##### Data Pilot

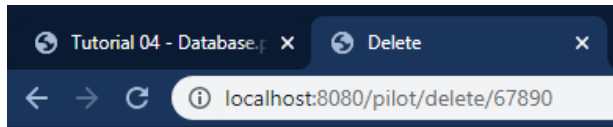
Pilot License Number	Name	Fly Hour	Action
67890	Axel	30	<a href="#">Delete</a>

[Add Flight](#)

##### Flight List

Flight Number	Origin	Destination	Time	Action
282	Tokyo	Jeju	2020-02-02	<a href="#">Delete</a>

Untuk melakukan hapus pada data pilot dilakukan dengan melakukan klik pada action delete di table data pilot.



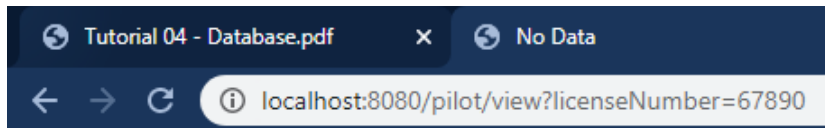
## Data Berhasil Dihapus!

Akan ditampilkan notifikasi data berhasil dihapus.

## Cari Pilot

License Number:

Dilakukan pencarian data pilot yang telah dihapus.



## Data tidak ada!

Ketika data dicari, data tersebut tidak ada karena telah terhapus.

4. Buatlah fitur update untuk seorang pilot dan sebuah flight. Catatan: License Number seorang pilot tidak bisa diubah.

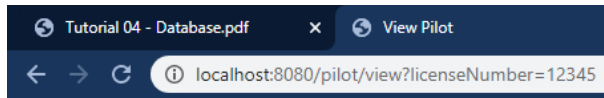
Untuk melakukan edit pada pilot dilakukan penambahan method pada pilotController serta file html untuk melakukan proses edit.

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

@RequestMapping(value = "/pilot/edit", method = RequestMethod.POST)
private String editPilotSubmit(@ModelAttribute PilotModel pilot, Long id) {
    pilotService.addPilot(pilot);
    return "update";
}
```

Method yang digunakan pada edit memiliki kemiripan dengan method yang digunakan pada proses add. Hal yang membedakan adalah file-file html yang melakukan load data yang telah tersimpan dan menyimpan data setelah dilakukan pengeditan.

- Edit pilot



### Data Pilot

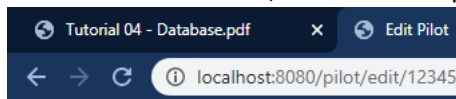
Pilot License Number	Name	Fly Hour	Action
12345	Caramia	3	<a href="#">Edit</a> <a href="#">Delete</a>

[Add Flight](#)

### Flight List

Flight Number	Origin	Destination	Time	Action
---------------	--------	-------------	------	--------

Untuk melakukan edit, dilakukan klik pada action edit.

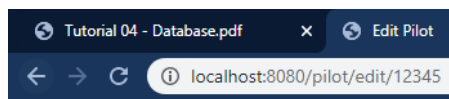


### Welcome!

#### Edit Pilot

Name:

Fly Hour:



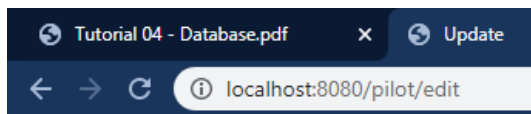
### Welcome!

#### Edit Pilot

Name:

Fly Hour:

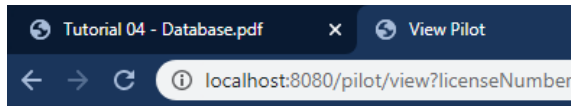
Lalu dilakukan edit pada field fly hour dari 3 menjadi 20.



### Data Berhasil Diupdate!

Akan muncul notifikasi bahwa data berhasil diupdate.





### Data Pilot

Pilot License Number	Name	Fly Hour	Action
12345	Caramia	20	<a href="#">Edit</a> <a href="#">Delete</a>

[Add Flight](#)

### Flight List

Flight Number	Origin	Destination	Time	Action
---------------	--------	-------------	------	--------

Ketika data pilot ditampilkan kembali, fly hour telah berubah.

Untuk melakukan edit flight dilakukan penambahan code pada FlightDb, FlightService serta penambahan method pada FlightController. Selain itu, dilakukan juga penambahan file editFlight.html

```
@Repository
public interface FlightDb extends JpaRepository<FlightModel, Long>{
    void deleteByFlightNumber(String flightNumber);
    FlightModel findByFlightNumber(String flightNumber);
}
```

Terdapat method pada FlightDB yang digunakan untuk mencari data flight berdasarkan flight number.

```
@Override
public FlightModel getFlightDetailByflightNumber(String flightNumber) {
    return flightDb.findByFlightNumber(flightNumber);
}
```

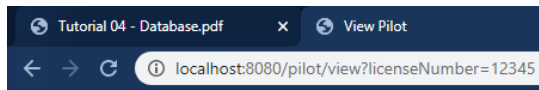
Pada FlightService terdapat penggunaan method FlightDb dengan parameter flightNumber.

```
@RequestMapping(value= "/flight/edit/{flightNumber}", method = RequestMethod.GET)
public String edit(@PathVariable(value = "flightNumber") String flightNumber, Model model) {
    FlightModel archive = flightService.getFlightDetailByflightNumber(flightNumber);
    model.addAttribute("flight", archive);
    return "editFlight";
}

@RequestMapping(value = "/flight/edit", method = RequestMethod.POST)
private String editFlightSubmit(@ModelAttribute FlightModel flight, Long id) {
    flightService.addFlight(flight);
    return "update";
}
```

Pada flightController memiliki kemiripan dengan pilotController. Akan tetapi return file htmlnya pada metode edit memiliki perbedaan.

- Edit Flight



#### Data Pilot

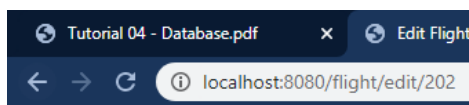
Pilot License Number	Name	Fly Hour	Action
12345	Caramia	20	<a href="#">Update</a> <a href="#">Delete</a>

[Add Flight](#)

#### Flight List

Flight Number	Origin	Destination	Time	Action
202	Bali	Granada	2019-11-27	<a href="#">Update</a> <a href="#">Delete</a>

Untuk melakukan edit flight dilakukan dengan melakukan klik pada update.



#### Welcome!

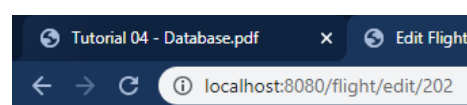
##### Edit Penerbangan

Flight Number:

Origin:

Destination:

Time:



#### Welcome!

##### Edit Penerbangan

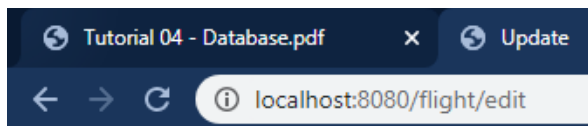
Flight Number:

Origin:

Destination:

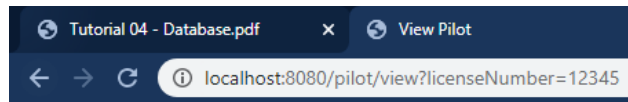
Time:

Data flight yang akan diubah merupakan field origin dari Bali menjadi Jakarta.



#### Data Berhasil Diupdate!

Akan ditampilkan notifikasi bahwa data berhasil diupdate.



### Data Pilot

Pilot License Number	Name	Fly Hour	Action
12345	Caramia	20	<a href="#">Update</a> <a href="#">Delete</a>

[Add Flight](#)

### Flight List

Flight Number	Origin	Destination	Time	Action
202	Jakarta	Granada	2019-11-26	<a href="#">Update</a> <a href="#">Delete</a>

Ketika data flight ditampilkan maka pada kolom origin telah berubah.

5. Buatlah fitur untuk melihat daftar flight termasuk menampilkan nama dan license number dari pilot yang bertugas (misalnya dalam bentuk tabel).

Untuk menampilkan semua penerbangan beserta nama pilot dan license number dibutuhkan penambahan method pada FlightController. Selain itu, ditambahkan file view-flight.html serta penambahan isi code pada FlightService.

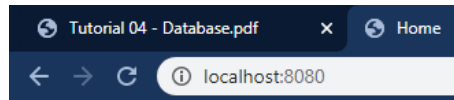
```
@Override
public List <FlightModel> listAll() {
    return flightDb.findAll();
}
```

Pada FlightService terdapat penggunaan findAll untuk menampilkan data flight keseluruhan.

```
@RequestMapping(value = "/flight/all", method = RequestMethod.GET)
private String all(Model model) {
    List <FlightModel> archive = flightService.listAll();
    model.addAttribute("flight", archive);
    return "view-flight";
}
```

Pada flightController terdapat pemanggilan method pada FlightService yang disimpan pada ArrayList dan digunakan pada model hingga nantinya akan direturn ke file view-flight.html

- View all flight



**Welcome!**

**List Penerbangan**

[List](#)

**Tambah Pilot**

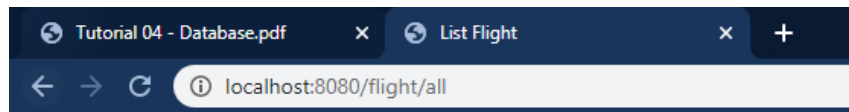
[Tambah](#)

**Cari Pilot**

License Number:

 Cari

Untuk menampilkan semua penerbangan dapat dilakukan dengan klik list.



**Daftar Penerbangan**

Flight Number	Origin	Destination	Time	Pilot	License Number
202	Jakarta	Granada	2019-11-26	Caramia	12345
888	Osaka	Seoul	2020-02-02	Caramia	12345
800	Surabaya	Jakarta	2020-02-11	Scarlett	54321

Setelah itu akan ditampilkan daftar penerbangan beserta nama pilot dan license number.