

## Tutorial 4

Muhammad Iqbal Mahendra

1606874690

APAP – A

### View Dealer berdasarkan ID

1. Implementasikan method di DealerController untuk menampilkan sebuah dealer berdasarkan id dealer. Method ini menerima parameter id dealer dan mengembalikan view (view-dealer.html).

*Buat method viewDealer pada DealerController*

```
@RequestMapping(value= "/dealer/view", method = RequestMethod.GET)
private String viewDealer(@RequestParam ("dealerId") long dealerId, Model model) {

    DealerModel archive = dealerService.getDealerDetailById(dealerId).get();
    model.addAttribute("dealer", archive);

    return "view-dealer";
}
```

*Isi dari view-dealer.html*

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Dealer</title>
    </head>
    <body>
        <h3 th:text="'Alamat = ' + ${dealer.alamat}"></h3>
        <h3 th:text="'Telepon = ' + ${dealer.noTelp}"></h3>
        <td><a th:href="@{/dealer/update/} + ${dealer.id}">Update Dealer</a></td> <br>
        <td><a th:href="@{/car/add/} + ${dealer.id}">Tambah Mobil</a></td>
    </body>
</html>
```

*Car dealer berdasarkan ID*

**Hello!**

**Tambah Dealer**

[Tambah](#)

**Cari Dealer berdasarkan Id**

Id Dealer:

*Hasil pencarian*

**Alamat = bogor**

**Telepon = 123573**

[Tambah Mobil](#)

2. Buatlah tampilan view-dealer.html seperti gambar di bawah ini. (Tambahkan link untuk menambahkan mobil)

**Alamat = bogor**

**Telepon = 123573**

[Tambah Mobil](#)

3. Jalankan program untuk menambah mobil dalam sebuah dealer

*Buat method add untuk menambahkan mobil*

```
@RequestMapping(value= "/car/add/{dealerId}", method = RequestMethod.GET)
private String add(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    CarModel car = new CarModel();
    DealerModel dealer = dealerService.getDealerDetailById(dealerId).get();
    car.setDealer(dealer);

    model.addAttribute("car", car);
    return "addCar";
}

@RequestMapping(value= "/car/add", method = RequestMethod.POST)
private String addCarSubmit(@ModelAttribute CarModel car) {
    carService.addCar(car);
    return "add";
}
```

*Tampilan halaman untuk menambahkan mobil*

**Hello!**

**Tambah Mobil**

Brand:

honda

Type:

brio

Price:

175000000

Amount:

50

Submit

*Mobil berhasil ditambahkan*

**Data Berhasil Ditambahkan**

## Latihan

1. Ubah method view dealer agar menampilkan daftar mobil beserta detail terurut berdasarkan harga mobil.

Buat method viewDealer yang menggunakan sort untuk menampilkan list mobil secara terurut berdasarkan harga

```
@RequestMapping(value= "/dealer/view", method = RequestMethod.GET)
private String viewDealer(@RequestParam ("dealerId") long dealerId, Model model) {

    DealerModel archive = dealerService.getDealerDetailById(dealerId).get();

    List<CarModel> listCar = archive.getListCar();
    Collections.sort(listCar, new sortCar());

    model.addAttribute("dealer", archive);
    model.addAttribute("listCar", listCar);

    return "view-dealer";
}
```

Buat class yang mengimplementasikan Comparator untuk sort mobil berdasarkan harga

```
class sortCar implements Comparator<CarModel> {

    public int compare(CarModel a, CarModel b) {
        return (int) (a.getPrice() - b.getPrice());
    }
}
```

Isi dari view-dealer.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>View Dealer</title>
    </head>
    <body>
        <h3 th:text="'Alamat = ' + ${dealer.alamat}"></h3>
        <h3 th:text="'Telepon = ' + ${dealer.noTelp}"></h3>
        <td><a th:href="@{/car/add/} + ${dealer.id}">Tambah Mobil</a></td>

        <h3>Daftar Mobil</h3>
        <div th:each="car : ${listCar}">
            <h4 th:text="'Brand = ' + ${car.brand}"></h4>
            <h4 th:text="'Type = ' + ${car.type}"></h4>
            <h4 th:text="'Price= ' + ${car.price}"></h4>
            <h4 th:text="'Amount = ' + ${car.amount}"></h4>
        </div>
    </body>
</html>
```

*Halaman yang ditampilkan*

**Alamat = bogor**

**Telepon = 123573**

[Tambah Mobil](#)

**Daftar Mobil**

**Brand = proton**

**Type = haha**

**Price= 34566798**

**Amount = 15**

**Brand = honda**

**Type = brio**

**Price= 175000000**

**Amount = 50**

## 2. Buatlah fitur delete untuk menghapus dealer dan sebuah mobil.

### A. Menghapus Dealer

*Buat method deleteDealer di dalam DealerService*

```
public interface DealerService {  
    Optional<DealerModel> getDealerDetailById(Long id);  
  
    void addDealer(DealerModel dealer);  
    void deleteDealer(DealerModel dealer);  
  
    DealerDb getDealerDb();  
}
```

*Buat implementasi dari method deleteDealer di dalam DealerServiceImpl*

```
@Override  
public void deleteDealer(DealerModel dealer) {  
    dealerDb.delete(dealer);  
}
```

*Buat method delete di dalam DealerController*

```
@RequestMapping(value= "/dealer/delete/{dealerId}", method = RequestMethod.GET)
private String delete(@PathVariable(value = "dealerId") Long dealerId, Model model) {

    DealerModel archive = dealerService.getDealerDetailById(dealerId).get();
    dealerService.deleteDealer(archive);

    return "delete-dealer";
}
```

*Sebelum dihapus*

**Daftar Dealer**

**Alamat = bogor**

**Telepon = 123573**

**ID = 1**

[Hapus](#)

*Dealer berhasil dihapus*

**Data Berhasil Dihapus**

*Setelah dihapus*

**Daftar Dealer**

## B. Menghapus Mobil

*Buat method deleteCar di dalam CarService*

```
public interface CarService {

    Optional<CarModel> getCarDetailById(Long id);

    void addCar(CarModel car);
    void deleteCar(CarModel car);

    CarDb getCarDb();
}
```

*Buat implementasi dari method deleteCar di dalam CarServiceImpl*

```
@Override
public void deleteCar(CarModel car) {
    carDb.delete(car);
}
```

*Buat method delete di dalam CarController*

```
@RequestMapping(value= "/car/delete/{carId}", method = RequestMethod.GET)
private String delete(@PathVariable(value = "carId") Long carId, Model model) {

    CarModel archive = carService.getCarDetailById(carId).get();
    carService.deleteCar(archive);

    return "delete-dealer";
}
```

*Sebelum dihapus*

**Alamat = bogor**

**Telepon = 123573**

[Tambah Mobil](#)

**Daftar Mobil**

**Brand = honda**

**Type = brio**

**Price= 175000000**

**Amount = 50**

[Hapus](#)

*Mobil berhasil dihapus*

**Data Berhasil Dihapus**

*Setelah dihapus*

**Alamat = bogor**

**Telepon = 123573**

[Tambah Mobil](#)

**Daftar Mobil**

### 3. Buatlah fitur update untuk seorang dealer dan sebuah mobil.

#### A. Update Dealer

*Buat method updateDealer di dalam DealerService*

```
public interface DealerService {
    Optional<DealerModel> getDealerDetailById(Long id);

    void addDealer(DealerModel dealer);
    void deleteDealer(DealerModel dealer);
    void updateDealer(DealerModel dealer);

    DealerDb getDealerDb();
}
```

*Buat implementasi dari method updateDealer di dalam DealerServiceImpl*

```
@Override
public void updateDealer(DealerModel dealer) {
    for (int i = 0; i < dealerDb.findAll().size(); i++) {
        if (dealerDb.findAll().get(i).getId() == (dealer.getId())) {

            DealerModel archive = dealerDb.findAll().get(i);
            int idx = dealerDb.findAll().indexOf(archive);

            dealerDb.findAll().get(idx).setAlamat(dealer.getAlamat());
            dealerDb.findAll().get(idx).setNoTelp(dealer.getNoTelp());

        }
    }
}
```

*Buat method update di dalam DealerController*

```
@RequestMapping(value= "/dealer/update/{dealerId}", method = RequestMethod.GET)
private String update(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    System.out.println("masukkk dealer");

    this.idSementara = dealerId;
    model.addAttribute("dealer", new DealerModel());

    return "updateDealer";
}

@RequestMapping(value= "/dealer/update", method = RequestMethod.POST)
private String updateDealerSubmit(@ModelAttribute DealerModel dealer) {

    dealer.setId(idSementara);
    dealerService.updateDealer(dealer);
    return "update";
}
```



### Isi dari updateDealer.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
    <title>Update Dealer</title>
</head>

<body>
    <h3>Update Dealer</h3>
    <form th:action="@{/dealer/update}" method="POST">
        Alamat: <br>
        <input type="text" name="alamat" /> <br>
        No. Telp: <br>
        <input type="text" name="noTelp" /> <br><br>

        <button type="submit">Submit</button>
    </form>
</body>

</html>
```

### Sebelum di-update

**Alamat = bogor**

**Telepon = 123573**

[Update Dealer](#)  
[Tambah Mobil](#)

### Daftar Mobil

**Brand = honda**

**Type = brio**

**Price= 175000000**

**Amount = 50**

### Update dealer

#### Update Dealer

Alamat:

No. Telp:

*Dealer berhasil di-update*

## **Data Berhasil Diubah**

*Setelah di-update*

**Alamat = Depok**

**Telepon = 128490**

[Update Dealer](#)

[Tambah Mobil](#)

### **Daftar Mobil**

**Brand = honda**

**Type = brio**

**Price= 175000000**

**Amount = 50**

#### **B. Update Mobil**

*Buat method updateCar di dalam CarService*

```
public interface CarService {  
  
    Optional<CarModel> getCarDetailById(Long id);  
  
    void addCar(CarModel car);  
    void deleteCar(CarModel car);  
    void updateCar(CarModel car);  
  
    CarDb getCarDb();  
}
```

Buat implementasi dari method `updateCar` di dalam `CarServiceImpl`

```
@Override
public void updateCar(CarModel car) {
    for (int i = 0; i < carDb.findAll().size(); i++) {
        if (carDb.findAll().get(i).getId() == (car.getId())) {

            CarModel archive = carDb.findAll().get(i);
            int idx = carDb.findAll().indexOf(archive);

            carDb.findAll().get(idx).setBrand(car.getBrand());
            carDb.findAll().get(idx).setType(car.getType());
            carDb.findAll().get(idx).setPrice(car.getPrice());
            carDb.findAll().get(idx).setAmount(car.getAmount());

        }
    }
}
```

Buat method `update` di dalam `CarController`

```
@RequestMapping(value= "/car/update/{carId}", method = RequestMethod.GET)
private String updateCar(@PathVariable(value = "carId") String carId, Model model) {

    CarModel archive = carService.getCarDetailById(Long.parseLong(carId)).get();
    this.idSementara = Long.parseLong(carId);
    model.addAttribute("car", archive);

    return "updateCar";
}

@RequestMapping(value= "/car/update/", method = RequestMethod.POST)
private String updateCarSubmit(@ModelAttribute CarModel car) {

    car.setId(idSementara);
    carService.updateCar(car);

    return "update";
}
```

Isi dari `updateCar.html`

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
    <title>Update Car</title>
</head>

<body>
    <h3>Update Car</h3>
    <form th:action="@{/car/update/{carId}(carId=${car.id})}" method="POST">
        Brand: <br>
        <input type="text" name="brand" /> <br>
        Type: <br>
        <input type="text" name="type" /> <br>
        Price: <br>
        <input type="text" name="price" /> <br>
        Amount: <br>
        <input type="text" name="amount" /> <br><br>

        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

*Sebelum di-update*

**Alamat = Depok**

**Telepon = 128490**

[Update Dealer](#)

[Tambah Mobil](#)

**Daftar Mobil**

**Brand = honda**

**Type = brio**

**Price= 175000000**

**Amount = 50**

[Hapus Update](#)

*Update mobil*

**Update Car**

Brand:

Type:

Price:

Amount:

*Mobil berhasil di-update*

**Data Berhasil Diubah**

Setelah di-update

**Alamat = Depok**

**Telepon = 128490**

[Update Dealer](#)  
[Tambah Mobil](#)

**Daftar Mobil**

**Brand = suzuki**

**Type = ertiga**

**Price= 150000000**

**Amount = 25**

[Hapus Update](#)

4. Buatlah fitur untuk melihat view all dealer sehingga dapat menampilkan brand mobil beserta amount dan price yang dijual.

Isi dari home.html yang ditambahkan fitur "Lihat Dealer"

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
  <title>Home</title>
</head>

<body>
  <h2>Hello!</h2>

  <h3>Tambah Dealer</h3>
  <td><a th:href="@{/dealer/add}">Tambah</a></td>

  <h3>Lihat Dealer</h3>
  <td><a th:href="@{/dealer/viewall}">Lihat</a></td>

  <h3>Cari Dealer berdasarkan Id</h3>
  <form th:action="@{/dealer/view}" method="GET">
    Id Dealer: <br>
    <input type="text" name="dealerId" />
    <button type="submit">Cari</button>
  </form>
</body>

</html>
```

Method viewAll di dalam DealerController

```
@RequestMapping("/dealer/viewall")
public String viewall (Model model) {

    DealerDb dealerDb = dealerService.getDealerDb();
    List<DealerModel> listDealer = dealerDb.findAll();

    model.addAttribute("listDealer", listDealer);
    return "viewall-dealer";
}
```

Isi dari viewall-dealer.html yang akan menampilkan semua dealer beserta mobil yang ada di dalamnya

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <title>View All Dealer</title>
</head>
<body>
    <h3>Daftar Dealer</h3>
    <div th:each="dealer : ${listDealer}">
        <h4 th:text="'Alamat = ' + ${dealer.alamat}"></h4>
        <h4 th:text="'Telepon = ' + ${dealer.noTelp}"></h4>
        <h4 th:text="'ID = ' + ${dealer.id}"></h4>

        <td><a th:href="@{/dealer/delete/} + ${dealer.id}">Hapus</a></td>

        <div th:each="car : ${dealer.listCar}">
            <h5 th:text="'Brand = ' + ${car.brand}"></h5>
            <h5 th:text="'Price = ' + ${car.price}"></h5>
            <h5 th:text="'Amount = ' + ${car.amount}"></h5>
        </div>
        <br>
    </div>
</body>
</html>
```

Tampilan Home

# Hello!

## Tambah Dealer

[Tambah](#)

## Lihat Dealer

[Lihat](#)

## Cari Dealer berdasarkan Id

Id Dealer:

*Tampilan View All Dealer (Lihat Dealer)*

### **Daftar Dealer**

**Alamat = Depok**

**Telepon = 128490**

**ID = 2**

[Hapus](#)

**Brand = suzuki**

**Price = 150000000**

**Amount = 25**

**Alamat = bogor**

**Telepon = 234990**

**ID = 3**

[Hapus](#)

**Brand = daihatsu**

**Price = 70000000**

**Amount = 30**