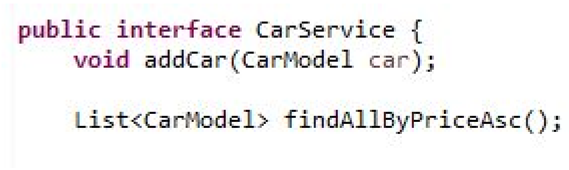
Nama : Hanifa Arrumaisha
NPM : 1606824332
Kelas : APAP-A

Notes to Asdos ku yang baik hati:
“Maaf ya kak, saya belum bisa mengisi write-up dengan baik dikarenakan harus ngerjain deadline PDB:(tapi saya mengerjakan semuanya sendiri dan insyaAllah memahami step by stepnya”

Lesson Learned

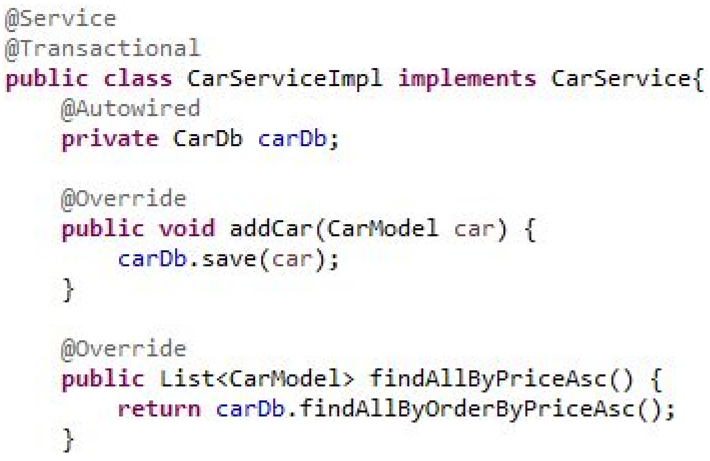
1. Apa itu JsonIgnore?
2. Apa itu fetch?
3. Apa maksud hibernate ddl-auto?
4. Buat apa dibuat interface dulu baru dibuat implementasinya?
5. Apa gunanya tipe data optional<T>

1. Ubah method view dealer agar menampilkan daftar mobil beserta detail terurut berdasarkan harga mobil.
    a. Membuat interface untuk findAllByPriceAsc() di CarService



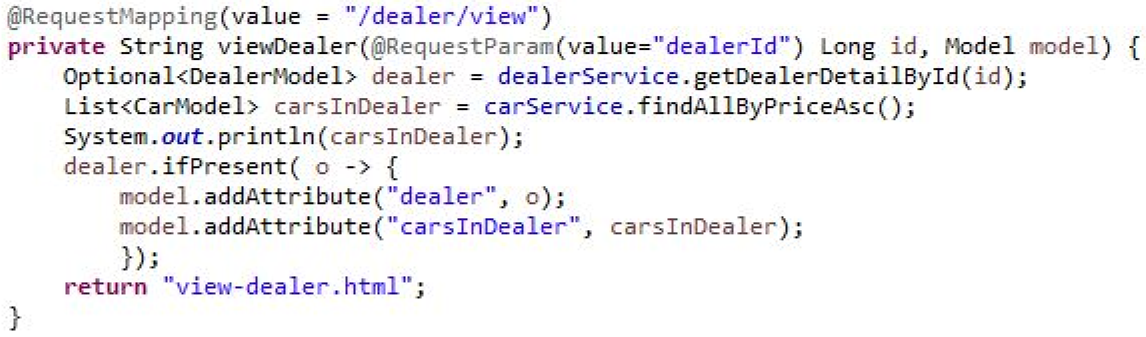
```
public interface CarService {
    void addCar(CarModel car);

    List<CarModel> findAllByPriceAsc();
```

    b. Membuat implementasi interface pada CarServiceImpl



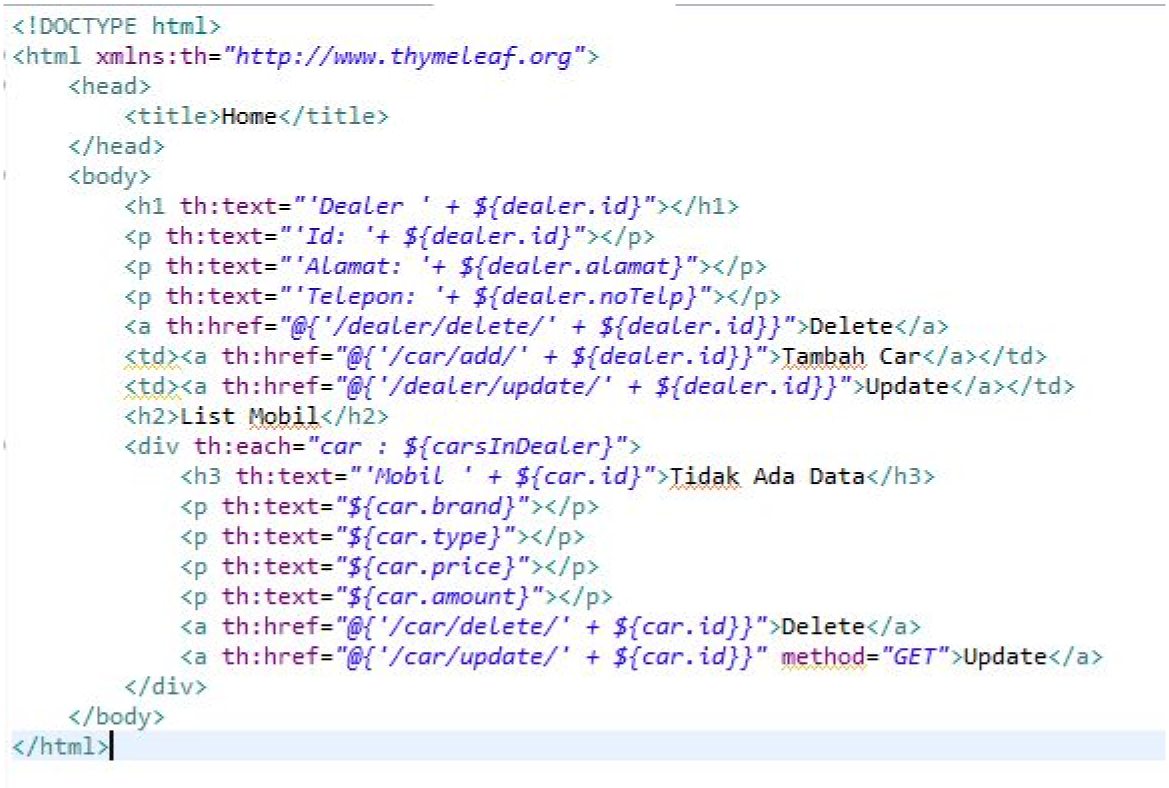
```
@Service
@Transactional
public class CarServiceImpl implements CarService{
    @Autowired
    private CarDb carDb;

    @Override
    public void addCar(CarModel car) {
        carDb.save(car);
    }

    @Override
    public List<CarModel> findAllByPriceAsc() {
        return carDb.findAllByOrderByPriceAsc();
    }
```

c. Mengubah method view dealer di DealerController



```
@RequestMapping(value = "/dealer/view")
private String viewDealer(@RequestParam(value="dealerId") Long id, Model model) {
    Optional<DealerModel> dealer = dealerService.getDealerDetailById(id);
    List<CarModel> carsInDealer = carService.findAllByPriceAsc();
    System.out.println(carsInDealer);
    dealer.ifPresent( o -> {
        model.addAttribute("dealer", o);
        model.addAttribute("carsInDealer", carsInDealer);
        });
    return "view-dealer.html";
}
```

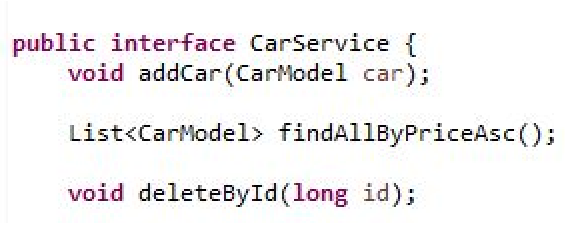
d. Mengupdate file HTML



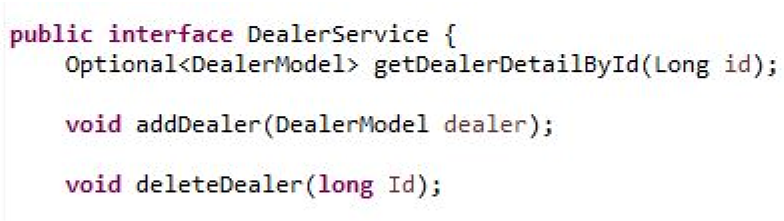
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Home</title>
    </head>
    <body>
        <h1 th:text="'Dealer ' + ${dealer.id}"></h1>
        <p th:text="'Id: '+ ${dealer.id}"></p>
        <p th:text="'Alamat: '+ ${dealer.alamat}"></p>
        <p th:text="'Telepon: '+ ${dealer.noTelp}"></p>
        <a th:href="@{'/dealer/delete/' + ${dealer.id}}">Delete</a>
        <td><a th:href="@{'/car/add/' + ${dealer.id}}">Tambah Car</a></td>
        <td><a th:href="@{'/dealer/update/' + ${dealer.id}}">Update</a></td>
        <h2>List Mobil</h2>
        <div th:each="car : ${carsInDealer}">
            <h3 th:text="'Mobil ' + ${car.id}">Tidak Ada Data</h3>
            <p th:text="${car.brand}"></p>
            <p th:text="${car.type}"></p>
            <p th:text="${car.price}"></p>
            <p th:text="${car.amount}"></p>
            <a th:href="@{'/car/delete/' + ${car.id}}">Delete</a>
            <a th:href="@{'/car/update/' + ${car.id}}" method="GET">Update</a>
        </div>
    </body>
</html>
```

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

a. Membuat interface Delete di CarService dan DealerService

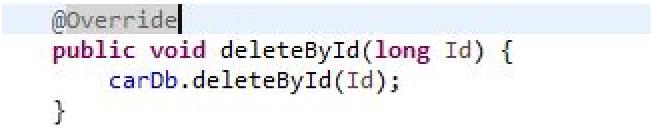


```
public interface CarService {
    void addCar(CarModel car);

    List<CarModel> findAllByPriceAsc();

    void deleteById(long id);
```

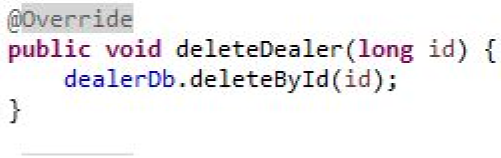


```
public interface DealerService {
    Optional<DealerModel> getDealerDetailById(Long id);

    void addDealer(DealerModel dealer);

    void deleteDealer(long Id);
```

b. Membuat implementasi dari interface ke CarServiceImpl dan DealerServiceImpl

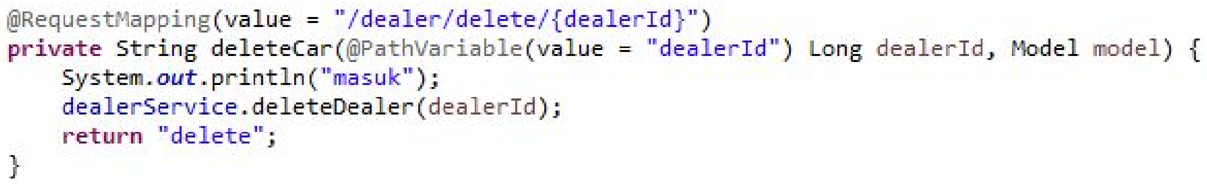


```
    @Override
    public void deleteById(long Id) {
        carDb.deleteById(Id);
    }
```

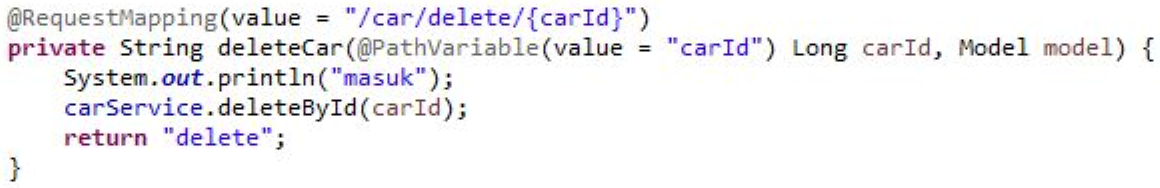


```
@Override
public void deleteDealer(long id) {
    dealerDb.deleteById(id);
}
```

c. Membuat function di DealerController dan CarController

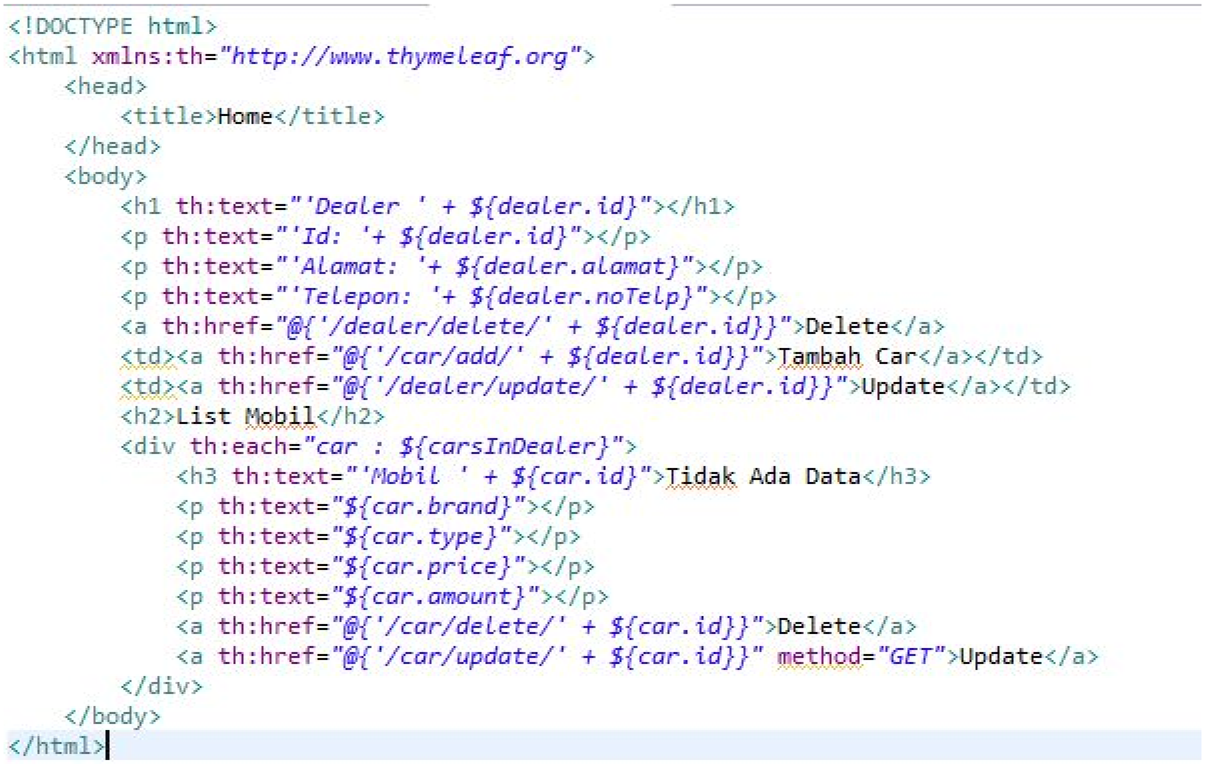


```
@RequestMapping(value = "/dealer/delete/{dealerId}")
private String deleteCar(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    System.out.println("masuk");
    dealerService.deleteDealer(dealerId);
    return "delete";
}
```



```
@RequestMapping(value = "/car/delete/{carId}")
private String deleteCar(@PathVariable(value = "carId") Long carId, Model model) {
    System.out.println("masuk");
    carService.deleteById(carId);
    return "delete";
}
```

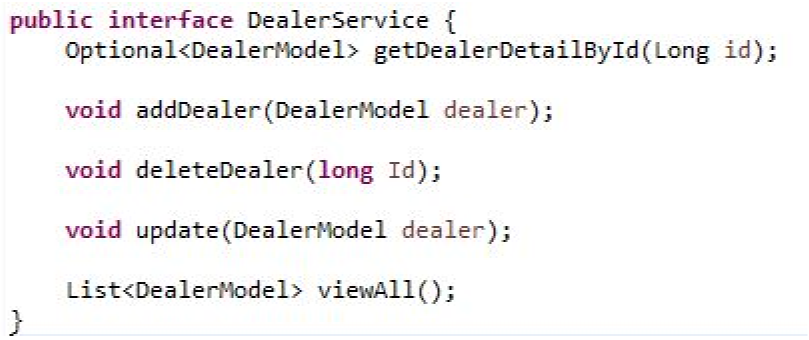
d. Membuat HTML untuk DealerController dan CarController



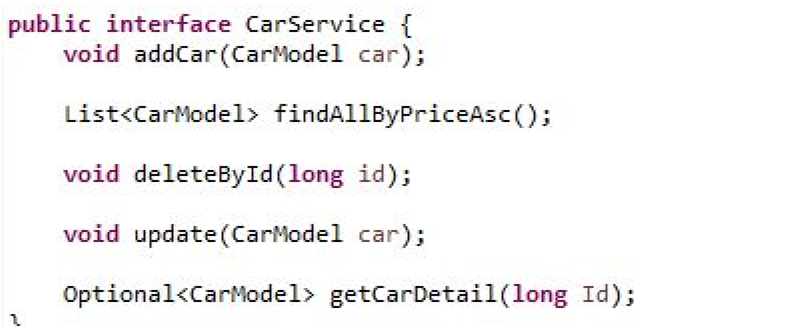
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Home</title>
    </head>
    <body>
        <h1 th:text="'Dealer ' + ${dealer.id}"></h1>
        <p th:text="'Id: '+ ${dealer.id}"></p>
        <p th:text="'Alamat: '+ ${dealer.alamat}"></p>
        <p th:text="'Telepon: '+ ${dealer.noTelp}"></p>
        <a th:href="@{'/dealer/delete/' + ${dealer.id}}">Delete</a>
        <td><a th:href="@{'/car/add/' + ${dealer.id}}">Tambah Car</a></td>
        <td><a th:href="@{'/dealer/update/' + ${dealer.id}}">Update</a></td>
        <h2>List Mobil</h2>
        <div th:each="car : ${carsInDealer}">
            <h3 th:text="'Mobil ' + ${car.id}">Tidak Ada Data</h3>
            <p th:text="${car.brand}"></p>
            <p th:text="${car.type}"></p>
            <p th:text="${car.price}"></p>
            <p th:text="${car.amount}"></p>
            <a th:href="@{'/car/delete/' + ${car.id}}">Delete</a>
            <a th:href="@{'/car/update/' + ${car.id}}" method="GET">Update</a>
        </div>
    </body>
</html>
```

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

a. Membuat method pada interface DealerService dan CarService

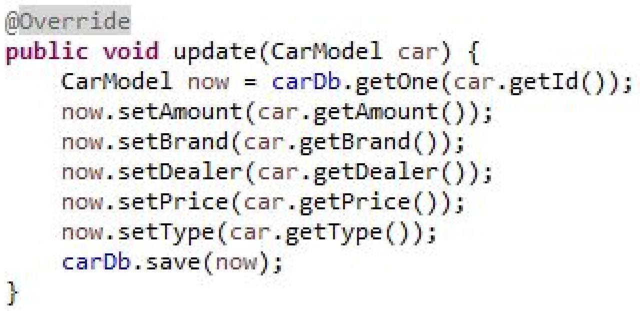


```
public interface DealerService {
    Optional<DealerModel> getDealerDetailById(Long id);

    void addDealer(DealerModel dealer);

    void deleteDealer(long Id);

    void update(DealerModel dealer);

    List<DealerModel> viewAll();
}
```

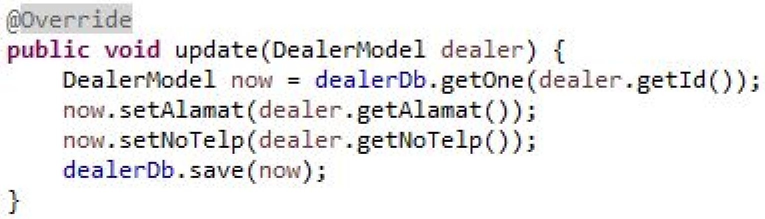


```
public interface CarService {
    void addCar(CarModel car);

    List<CarModel> findAllByPriceAsc();

    void deleteById(long id);

    void update(CarModel car);

    Optional<CarModel> getCarDetail(long Id);
}
```

b. Membuat implementasi dari method update pada DealerServiceImpl dan CarServiceImpl

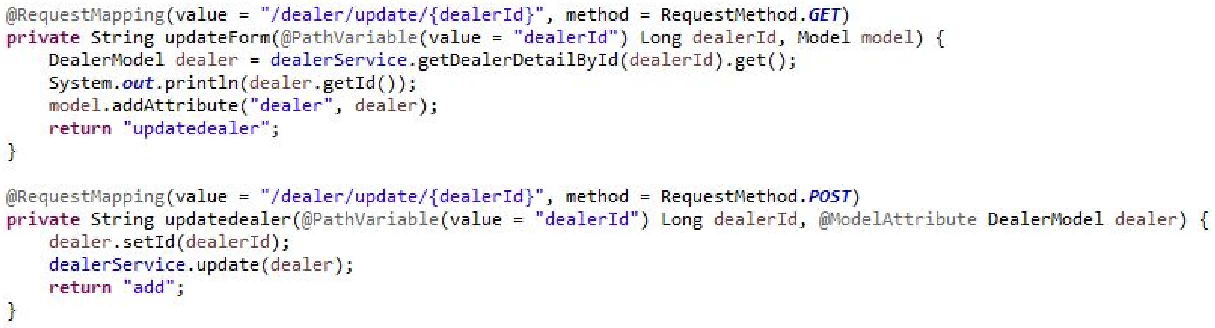


```
@Override
public void update(CarModel car) {
    CarModel now = carDb.getOne(car.getId());
    now.setAmount(car.getAmount());
    now.setBrand(car.getBrand());
    now.setDealer(car.getDealer());
    now.setPrice(car.getPrice());
    now.setType(car.getType());
    carDb.save(now);
}
```

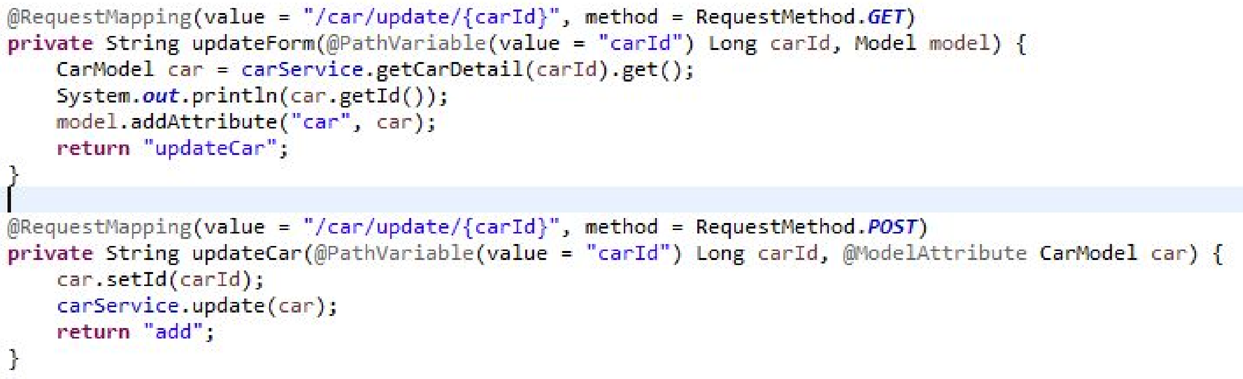


```
@Override
public void update(DealerModel dealer) {
    DealerModel now = dealerDb.getOne(dealer.getId());
    now.setAlamat(dealer.getAlamat());
    now.setNoTelp(dealer.getNoTelp());
    dealerDb.save(now);
}
```

c. Membuat fungsi di controller untuk menghandle request update pada CarController dan DealerController

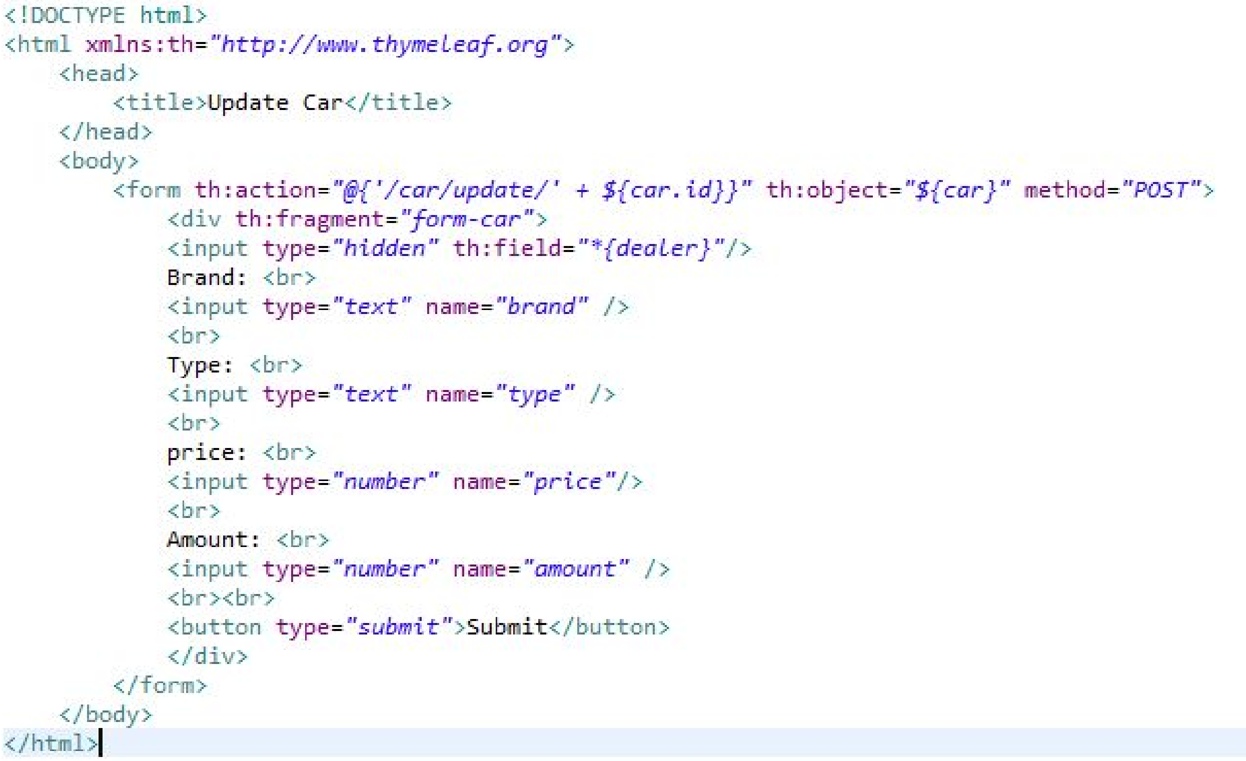


```
@RequestMapping(value = "/dealer/update/{dealerId}", method = RequestMethod.GET)
private String updateForm(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    DealerModel dealer = dealerService.getDealerDetailById(dealerId).get();
    System.out.println(dealer.getId());
    model.addAttribute("dealer", dealer);
    return "updatedealer";
}

@RequestMapping(value = "/dealer/update/{dealerId}", method = RequestMethod.POST)
private String updatedealer(@PathVariable(value = "dealerId") Long dealerId, @ModelAttribute DealerModel dealer) {
    dealer.setId(dealerId);
    dealerService.update(dealer);
    return "add";
}
```

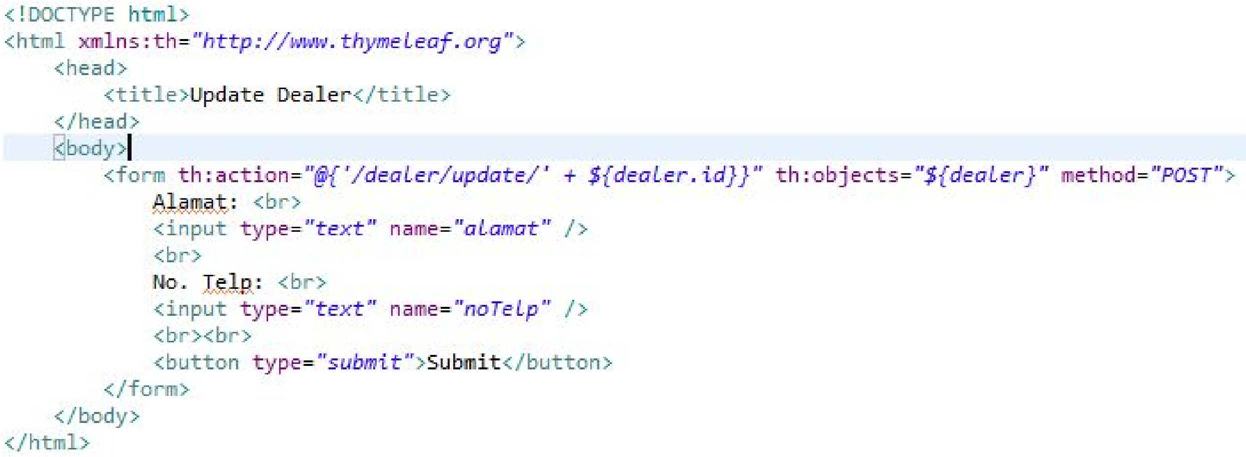


```
@RequestMapping(value = "/car/update/{carId}", method = RequestMethod.GET)
private String updateForm(@PathVariable(value = "carId") Long carId, Model model) {
    CarModel car = carService.getCarDetail(carId).get();
    System.out.println(car.getId());
    model.addAttribute("car", car);
    return "updateCar";
}

@RequestMapping(value = "/car/update/{carId}", method = RequestMethod.POST)
private String updateCar(@PathVariable(value = "carId") Long carId, @ModelAttribute CarModel car) {
    car.setId(carId);
    carService.update(car);
    return "add";
}
```

d. Membuat HTML untuk mengisi form update

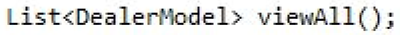


```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Update Car</title>
    </head>
    <body>
        <form th:action="@{'/car/update/' + ${car.id}}" th:object="${car}" method="POST">
            <div th:fragment="form-car">
            <input type="hidden" th:field="*{dealer}"/>
            Brand: <br>
            <input type="text" name="brand" />
            <br>
            Type: <br>
            <input type="text" name="type" />
            <br>
            price: <br>
            <input type="number" name="price"/>
            <br>
            Amount: <br>
            <input type="number" name="amount" />
            <br><br>
            <button type="submit">Submit</button>
            </div>
        </form>
    </body>
</html>
```



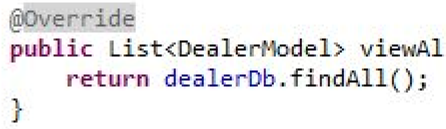
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Update Dealer</title>
    </head>
    <body>
        <form th:action="@{'/dealer/update/' + ${dealer.id}}" th:objects="${dealer}" method="POST">
            Alamat: <br>
            <input type="text" name="alamat" />
            <br>
            No. Telp: <br>
            <input type="text" name="noTelp" />
            <br><br>
            <button type="submit">Submit</button>
        </form>
    </body>
</html>
```

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



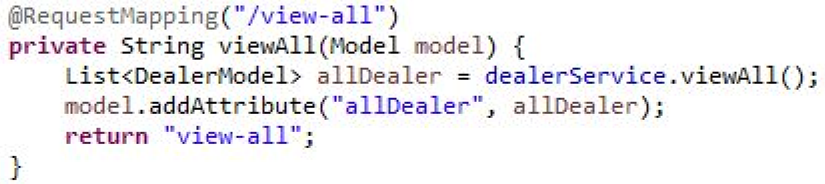
```
List<DealerModel> viewAll();
```

b. Membuat implementasi fungsi di DealerServiceImpl



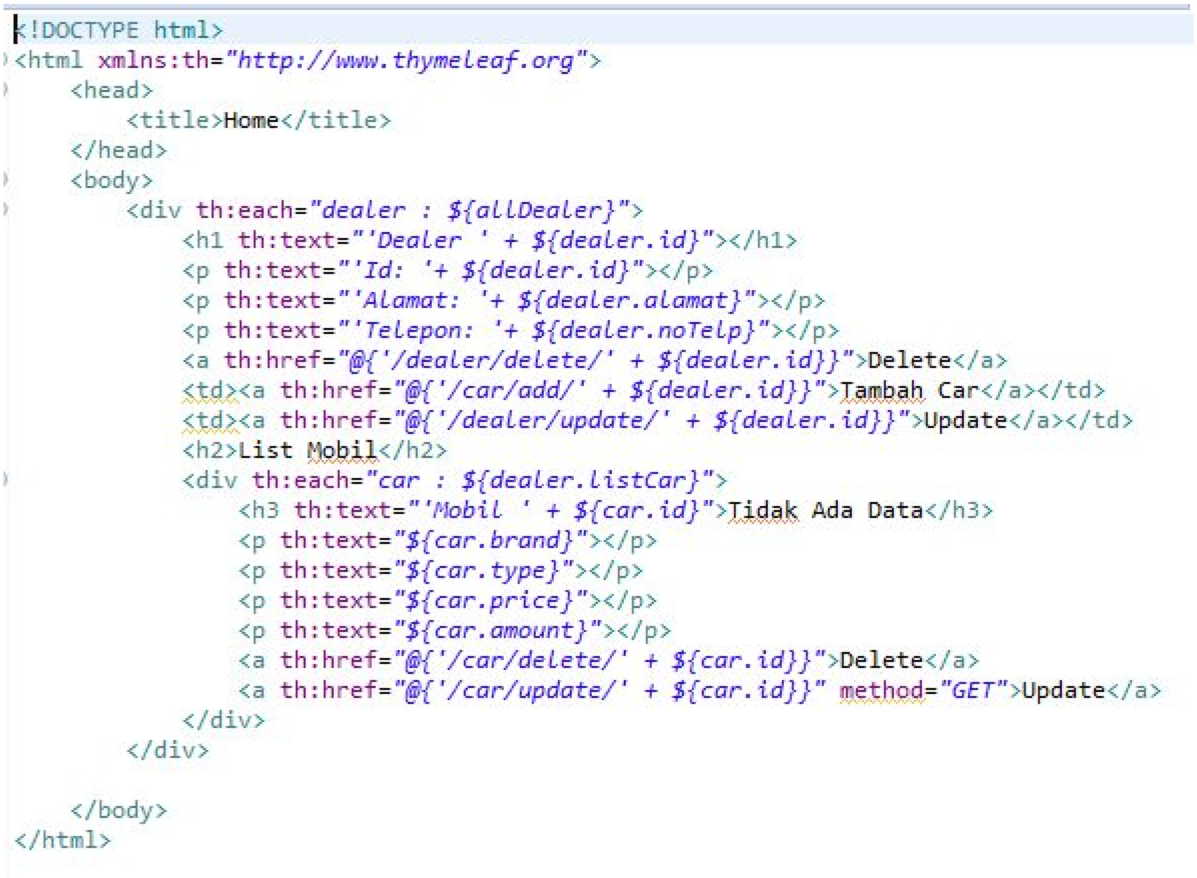
```
@Override
public List<DealerModel> viewAll() {
    return dealerDb.findAll();
}
```

c. Membuat fungsi di controller



```
@RequestMapping("/view-all")
private String viewAll(Model model) {
    List<DealerModel> allDealer = dealerService.viewAll();
    model.addAttribute("allDealer", allDealer);
    return "view-all";
}
```

d. Membuat HTML untuk response dari request



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
    <head>
        <title>Home</title>
    </head>
    <body>
        <div th:each="dealer : ${allDealer}">
            <h1 th:text="'Dealer ' + ${dealer.id}"></h1>
            <p th:text="'Id: '+ ${dealer.id}"></p>
            <p th:text="'Alamat: '+ ${dealer.alamat}"></p>
            <p th:text="'Telepon: '+ ${dealer.noTelp}"></p>
            <a th:href="@{'/dealer/delete/' + ${dealer.id}}">Delete</a>
            <td><a th:href="@{'/car/add/' + ${dealer.id}}">Tambah Car</a></td>
            <td><a th:href="@{'/dealer/update/' + ${dealer.id}}">Update</a></td>
            <h2>List Mobil</h2>
            <div th:each="car : ${dealer.listCar}">
                <h3 th:text="'Mobil ' + ${car.id}">Tidak Ada Data</h3>
                <p th:text="${car.brand}"></p>
                <p th:text="${car.type}"></p>
                <p th:text="${car.price}"></p>
                <p th:text="${car.amount}"></p>
                <a th:href="@{'/car/delete/' + ${car.id}}">Delete</a>
                <a th:href="@{'/car/update/' + ${car.id}}" method="GET">Update</a>
            </div>
        </div>

    </body>
</html>
```