

1. - Menambahkan kode di fragment.html

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

<head>

    <head th:fragment="title">
        <title th:text="${title}"></title>
    </head>
```

Dengan ini, fragment tidak hanya berisi navbar, tetapi juga title yang akan di implementasi oleh setiap halaman yang ada pada proyek.

- Menambahkan kode untuk setiap halaman

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.com">
<head>
    <object th:include="fragments/fragment :: title" th:remove="tag"></object>
    <object th:include="fragments/fragment :: css" th:remove="tag"></object>
    <object th:include="fragments/fragment :: js" th:remove="tag"></object>
</head>
<body>
    <nav th:replace="fragments/fragment :: navbar"></nav>
```

Kode ini akan membuat setiap halaman html yang ada untuk meng include isi dari fragment (title, navbar)

- Menambahkan kode di setiap method yang ada di controller

```
@RequestMapping(value = "/car/add/{dealerId}", method = RequestMethod.GET)
private String add(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    DealerModel dealer = dealerService.getDealerDetailById(dealerId).get();
    ArrayList<CarModel> list = new ArrayList<CarModel>();
    list.add(new CarModel());
    dealer.setListCar(list);
    model.addAttribute("title", "Tambah Mobil");
    model.addAttribute("dealer", dealer);
    return "addCar";
}
```

Hal ini akan membuat setiap mapping ke html akan berbeda beda sesuai tujuannya masingmasing.

2. Mengganti method add di carController

```

@RequestMapping(value = "/car/add/{dealerId}", method = RequestMethod.GET)
private String add(@PathVariable(value = "dealerId") Long dealerId, Model model) {
    DealerModel dealer = dealerService.getDealerDetailById(dealerId).get();
    ArrayList<CarModel> list = new ArrayList<CarModel>();
    list.add(new CarModel());
    dealer.setListCar(list);
    model.addAttribute("title", "Tambah Mobil");
    model.addAttribute("dealer", dealer);
    return "addCar";
}

@RequestMapping(value = "/car/add/{dealerId}", method = RequestMethod.POST, params= {"save"})
private String addCarSubmit (@ModelAttribute DealerModel dealer) {
    DealerModel dealer_now = dealerService.getDealerDetailById(dealer.getId()).get();
    for(CarModel car : dealer.getListCar()) {
        car.setDealer(dealer_now);
        carService.addCar(car);
    }

    return "add";
}

@RequestMapping(value = "/car/add/{dealerId}", params= {"addRow"}, method = RequestMethod.POST)
private String addRow (@ModelAttribute DealerModel dealer, Model model) {
    dealer.getListCar().add(new CarModel());
    model.addAttribute("dealer", dealer);
    return "addCar";
}

@RequestMapping(value="/car/add/{dealerId}", method = RequestMethod.POST, params={"removeRow"})
private String removeRow (@ModelAttribute DealerModel dealer, final BindingResult bindingResult, final HttpServletRequest req, Model model) {
    final Integer row = Integer.valueOf(req.getParameter("removeRow"));
    dealer.getListCar().remove(row.intValue());
    model.addAttribute("dealer", dealer);
    return "addCar";
}

```

Membuat 4 method untuk menambahkan. 2 method untuk menambahkan dan mengurangi row, 1 method untuk menambahkan mobil ke dalam list car yang ada pada setiap dealer (dengan menambahkannya ke carService dan meng set dealer nya). Dan 1 method lagi untuk meng instansiasi sebuah arraylist (karena -penambahan mobil bisa dilakukan untuk banyak mobil) dan meng set arraylist tersebut sehingga semua mobil yang ada di dalamnya mempunyai dealer yang sama.

- Mengganti addCar.html

```

<body>
  <nav th:replace="fragments/fragment :: navbar"></nav>

  <h2>Hello!</h2>

  <h3>Tambah Mobil</h3>
  <form th:action="@{/car/add/} + ${dealer.id}" th:object="${dealer}" method = "POST">
    <input type="hidden" th:field="**{id}"></input>
    <table>
      <thead>
        <tr>
          <th>Brand</th>
          <th>Type</th>
          <th>Price</th>
          <th>Amount</th>
          <th><button type="submit" name="addRow">Add Row</button></th>
        </tr>
      </thead>
      <tbody>
        <tr th:each="car,rowStat : ${listCar}">
          <td><input type="text" th:field="**{listCar[__${rowStat.index}__].brand}"/></td>
          <td><input type="text" th:field="**{listCar[__${rowStat.index}__].type}"/></td>
          <td><input type="number" th:field="**{listCar[__${rowStat.index}__].price}"/></td>
          <td><input type="number" th:field="**{listCar[__${rowStat.index}__].amount}"/></td>
          <td><button type="submit" name="removeRow" th:value="${rowStat.index}">Remove</button></td>
        </tr>
      </tbody>
    </table>
    <button type="submit" name="save">Submit</button>
  </form>
</body>

```

3. Perbedaan th:include dan th:replace

th :include membuat fragmen ditambahkan ke dalam tag

Th:replace membuat fragmen mereplace apa yang sudah ada di dalam tag tersebut.

Sumber:teman dan stackoverflow :))