

Materi 12 : Search & Sorting

DAFTAR ISI

1.1 Search1

1.2 Sorting.....2

1.1 Search

Fitur search sudah disediakan oleh JPA, jadi kita tidak perlu repot-repot untuk membuat query mencari sebuah data. Untuk menggunakan fitur ini kita harus menggunakan sebuah method abstrak yang ada di interface Repository.

Fitur search ini terbagi menjadi dua, yaitu:

- **findAllBy**, mencari semua data berdasarkan perintah
- **findBy**, mencari sebuah data berdasarkan perintah

Misalnya untuk mencari id sebuah entitas, maka kita harus menggunakan method **findById**. Atau jika ingin mencari semua data berdasarkan nama, maka kita harus menggunakan method **findAllByName**.

Sebagai contoh, mari gunakan project **coffe_aht** sebelumnya. Kemudian tambahkan kode berikut ini di interface **CoffeeRepository.java**

```
public interface CoffeeRepository extends JpaRepository<Coffee, Integer> {  
    List<Coffee> findByMerkContainingIgnoreCase(String merk);  
}
```

Kode di atas berarti kita ingin mencari sebuah data berdasarkan **merk** yang kita inputkan. Kemudian arti dari **ContainingIgnoreCase** artinya data akan ditampilkan walaupun persamaan katanya minimum. Untuk lebih jelasnya nanti kalian coba sendiri hasilnya.

Kemudian di file html kita tambahkan elemen form untuk menampung inputan. Contohnya adalah sebagai berikut:

```
<div class="search">  
  <form action="#" th:action="@{/search}" method="get">  
    <input  
      type="text"  
      placeholder="Cari Berdasarkan Nama Merk..."  
      name="search"  
    />  
    <input type="submit" value="Cari" />  
  </form>  
</div>
```

Kode di atas akan menampung inputan dari kita, kemudian data yang ditampung diteruskan ke end-point **/sarch** yang isinya adalah sebagai berikut:

```
@GetMapping("/search")
public String searchByName(@RequestParam(name = "search")
                           String merk, Model model){
    List<Coffee> coffees = coffeeRepository.findByMerkContainingIgnoreCase(merk);
    model.addAttribute("coffees", coffees);
    return "show-coffee";
}
```

Kalian juga bisa membuat fitur search lain, misalnya search berdasarkan nama merk, harga, kuantitas, dll.

1.2 Sorting

Proses sorting di JPA terbilang cukup gampang. Karena kita hanya perlu menggunakan sebuah method yang sudah siapkan otomatis oleh JPA. Berikut adalah cara penggunaannya.

Pertama kita tambahkan kode berikut di interface CoffeeRepository.java

```
public interface CoffeeRepository extends JpaRepository<Coffee, Integer> {
    List<Coffee> findAllOrderByMerkAsc();
    List<Coffee> findAllOrderByMerkDesc();
}
```

Jika kita ingat pada pelatihan basis data lalu, kita menggunakan query **ORDER BY ... ASC** untuk mengurutkan entitas dari yang terkecil hingga ke yang terbesar. Di JPA juga sama, hanya menambahkan keyword **findAllBy** atau **findBy** saja.

Kemudian di file HTML kita perlu membuat sebuah tombol untuk melakukan penyortiran. Berikut adalah contoh kodenya:

```
<a th:href="@{/sort-by-merk-asc}">
    <button>Sort By Merk Asc</button>
</a>
```

Kode di atas merujuk pada end-point **sort-by-merk-asc** yang berisi:

```
@GetMapping("/sort-by-merk-asc")
public String sortByMerkAsc(Model model){
    List<Coffee> coffees = coffeeRepository.findAllOrderByMerkAsc();
    model.addAttribute("coffees", coffees);
    return "show-coffee";
}
```