

BAB IV

Eloquent: Filter and Search

4.1. TUJUAN

1. Mahasiswa dapat memahami struktur Model
2. Mahasiswa dapat memahami Konsep Model, View dan Controller (MVC)
3. Mahasiswa dapat memahami Object Relational Mapper (ORM).
4. Mahasiswa dapat menggunakan ORM untuk melakukan pencarian data

4.2. DASAR TEORI

Fitur pencarian adalah suatu fungsi untuk mencari dan mengidentifikasi informasi di dalam database yang sesuai dengan kata kunci yang dimasukkan oleh pengguna. Fitur ini menjadi sebuah standar wajib bagi setiap sistem informasi. Pada kasus ini, pencarian akan dilakukan menggunakan fitur Eloquent berdasarkan pada kata kunci berupa text dan filter kelompok tani.

4.3. LANGKAH PRAKTIKUM

Sebelum memulai praktikum pada BAB IV, pastikan terlebih dahulu praktikum sebelumnya telah dikerjakan dan lembar evaluasi telah tercentang semuanya. Karena, pada praktikum kali ini akan menggunakan controller dan view yang sama, termasuk fungsi didalamnya. Oleh karena itu, pastikan beberapa hal telah dipahami yakni eloquent model, CRUD, blade view, dan pagination. Silakan dibuka project sebelumnya kemudian lakukan langkah berikut.

1. Siapkan fungsi untuk melakukan pencarian, untuk itu silakan dibuka kembali `app/Http/Controllers/PetaniController.php` seperti yang dibuat pada langkah 10 BAB III. Kemudian sesuaikan fungsi `index`, karena pencarian dan pagination dibuat berkaitan. Mengapa? Sering terjadi pencarian yang dilakukan menemukan data melebihi jumlah data pada pagination. Oleh karenanya, pagination juga seharusnya dapat dilakukan terhadap hasil pencarian yang ditemukan.

```
public function index()
{
    $title="Daftar Petani";
    //mengambil instance Petani sebagai inisialisasi dari data petani
    $petanis=new Petani;
    //melakukan pengecekan kondisi jika ditemukan adanya kata kunci
    //kata kunci diterima menggunakan method GET[s]
    if(isset($_GET['s'])){
        $s=$_GET['s'];
        //melakukan query like berdasarkan param nama
        $petanis=$petanis->where('nama', 'like', "%$s%");
    }
    //menambahkan filter pencarian
    //jika ditemukan adanya filter berupa id_kelompok
    if(isset($_GET['id_kelompok'])&&$_GET['id_kelompok']!=''){
        $petanis=$petanis->where('id_kelompok_tani', $_GET['id_kelompok']);
    }
    //melakukan eksekusi query dengan paginate 5 data per page
    $petanis=$petanis->paginate(5);
    //mengambil list kelompok tani sebagai filter
    $kelompok=KelompokTani::all();
    return view('admin.daftarpetani',compact('title','petanis','kelompok'));
}
```

2. Selanjutnya mari buka kembali view daftar petani `resources/views/admin/daftarpetani.blade.php`. Beberapa hal perlu disesuaikan

untuk memberikan form pencarian dan filter pada bagian <!-- PENCARIAN -->. Kedua memberikan request query pada pagination agar pagination dapat dilakukan pada pencarian dan filter pada bagian <!-- PAGINATION -->.

```
<!-- memanggil admin template -->
<!-- component layout yang dibuat pada langkah 3 -->
<x-admin-layout>
<h2 class="font-semibold text-xl text-gray-800 leading-tight">
<!-- menampilkan variabel title yang dikirim dari controller-->
{{{title}}}
</h2>
<div>
<div class="shadow px-6 py-4 bg-white rounded sm:px-1 sm:py-1 ">
<div class="grid grid-cols-1 md:grid-cols-2 p-4">
<div class="">
<!-- menampilkan url untuk menampilkan halaman input berita -->
<a href="{{route('petani.create')}}">
<button class="px-4 py-1 text-sm rounded text-purple-600 font-semibold border border-purple-200 hover:text-white hover:bg-purple-600 hover:border-transparent focus:outline-none">Tambah</button>
</a>
<button class="px-4 py-1 text-sm rounded text-green-600 font-semibold border border-green-200 hover:text-white hover:bg-green-600 hover:border-transparent focus:outline-none">Publish</button>
<button class="px-4 py-1 text-sm rounded text-red-600 font-semibold border border-red-200 hover:text-white hover:bg-red-600 hover:border-transparent focus:outline-none">Delete</button>
</div>
<div class="">
<!-- PENCARIAN -->
<!-- memberikan form pencarian menggunakan method get -->
<!-- method get lebih baik digunakan pada pencarian -->
<!-- salah satunya karena dapat merekam parameter query pada address bar-->
<form action="{{route('petani.index')}}" method="GET" >
<div class="grid grid-cols-1 md:grid-cols-2 " >
<div >
<!-- memberikan filter berupa kelompok tani-->
<select id="kelompok" name="id_kelompok"
class="block w-full py-2 px-4 border border-gray-300 bg-white rounded-l-2xl shadow-sm focus:outline-none focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm">
<option value="">Pilih Kelompok Tani</option>
<!-- menampilkan list kelompok petani-->
@foreach ($kelompok as $item)
<option value="{{ $item->id_kelompok_tani }}"
{{{isset($_GET['id_kelompok']) && $_GET['id_kelompok'] == $item->id_kelompok_tani ? 'selected' : ''}}}> {{{ $item->nama_kelompok }}}</option>
@endforeach
</select>
</div>
<div>
<!-- memberikan kata kunci dengan nama s -->
<input type="text" name="s" value="{{(isset($_GET['s'])) ? $_GET['s'] : ''}}" class="px-4 py-2 focus:ring-indigo-500 focus:border-indigo-500 sm:text-sm border-gray-300">
<button type="submit" class="rounded-r-md border border-l-0 px-4 py-1 border-gray-300 bg-gray-50 text-gray-500 text-blue-600 hover:text-white hover:bg-blue-600">
Cari</button>
</div>
</div>
</form>
<!-- END PENCARIAN -->
</div>
<div>
<div class="py-2 align-middle inline-block min-w-full sm:px-4 lg:px-4">
<div class="shadow overflow-hidden border-b border-gray-200 sm:rounded-lg">
<table class="min-w-full divide-y divide-gray-200">
<thead class="bg-gray-50">
<tr>
```

```

<th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">
    Nama
</th>
<th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">
    Kontak
</th>
<th scope="col" class="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">
    Kelompok
</th>
<th scope="col" class="px-6 py-3 text-right text-xs font-medium text-gray-500 uppercase tracking-wider">
    Aksi
</th>
</tr>
</thead>
<tbody class="bg-white divide-y divide-gray-200">
<!-- menggunakan foreach untuk menampilkan list array ataupun collection-->
@foreach ($petanis as $key=>$item)
<tr>
<td class="px-6 py-4 whitespace-nowrap">
    <div class="flex items-center"><input type="checkbox" name="" id="">
    <div class="flex-shrink-0 h-10 w-10">
        
    </div>
    <div class="ml-4">
        <div class="text-sm font-medium text-gray-900">
            <!-- Menampilkan elemen nama dari object item -->
            {{ $item->nama }}
        </div>
        <div class="text-sm text-gray-500">
            {{ $item->nik }}
        </div>
    </div>
</td>
<td class="px-6 py-4 whitespace-nowrap">
        <div class="text-sm text-gray-900">{{ $item->telpon }}</div>
        <div class="text-sm text-gray-500">{{ $item->alamat }}</div>
    </td>
<td class="px-6 py-4 whitespace-nowrap text-sm text-gray-500">
        <!-- memanggil fungsi kelompok pada model Petani, kemudian menampilkan elemen berupa nama_kelompok -->
        {{ $item->kelompok->nama_kelompok }}
    </td>
<td class="px-6 py-4 whitespace-nowrap text-right text-sm font-medium">
        <form action="{{route('petani.destroy',$item->id_penjual)}}" method="POST">
            <!-- Form lengkap dengan token csrf untuk method(DELETE)-->
            @csrf @method('DELETE')
            <!-- link untuk edit-->
            <a href="{{route('petani.edit',$item->id_penjual)}}" class="text-indigo-600 hover:text-indigo-900">Edit</a>
            <!-- button action untuk delete-->
            <button class="text-red-600 hover:text-red-900 onclick="return confirm('Anda Yakin ?')" type="submit">Del</button>
        </form>
    </td>
</tr>
</foreach>
</tbody>
</table>
</div>
</div>
<!-- PAGINATION -->
<!-- menampilkan pagination lengkap dengan parameter pencarian-->

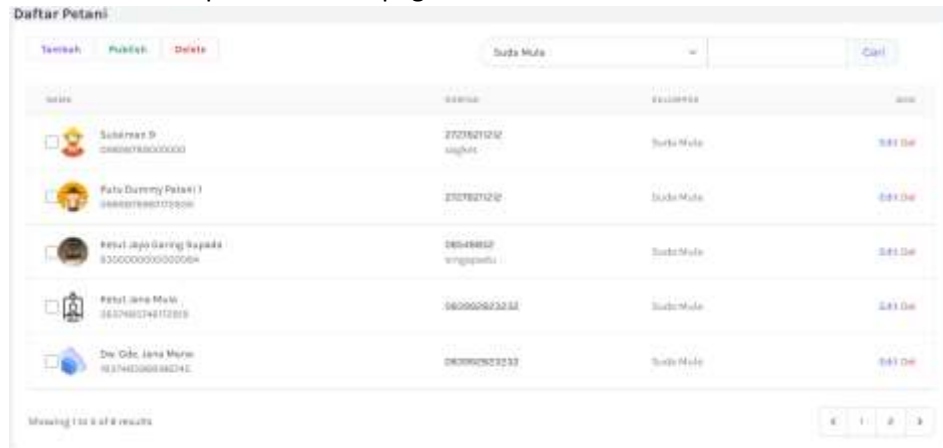
```

```

<!-- oleh karena itu pagination dilengkapi dengan request query-->
<div class="m-4"> {{ $petanis->appends(request()->query()->links()) }} </div>
<!-- END PAGINATION -->
</div>
</div>
</x-admin-layout>

```

Terakhir pastikan pencarian dan paginationnya dapat berjalan berbarengan. Gambar 4.1 adalah tampilan untuk halaman pencarian dan pagination.



Gambar 4.1 Pencarian dan Pagination

- Selanjutnya mari membuat fungsi untuk menampilkan daftar petani berdasarkan id kelompok tani. Hampir sama dengan filter sebelumnya pada langkah 1, hanya saja pada langkah ini menampilkan daftar petani memanfaatkan class `hasMany` pada fungsi `petanis()` model `KelompokTani`. Untuk itu silakan dibuka kembali `PetaniController` kemudian tambahkan fungsi `kelompok()`.

```

public function kelompok($id){
    //memanggil kelompok berdasarkan id_kelompok_tani
    $kelompok=KelompokTani::where('id_kelompok_tani',$id)->first();
    //menampilkan nama kelompok pada title
    $title='Kelompok '.$kelompok->nama_kelompok;
    //memanggil daftar petani berdasarkan kelompok
    $petanis=$kelompok->petanis;
    $kelompok=KelompokTani::all();
    return view('admin.daftarpetani',compact('title','petanis','kelompok'));
}

```

- Membuat route baru pada route web `routes/web.php`.

```

Route::get('kelompok/{id}', [PetaniController::class,'kelompok'])-
>name('petani.kelompok');

```

- Menampilkan link kelompok tani pada halaman daftar petani `resources/views/admindaftarpetani.blade.php` tepatnya pada nama kelompok tani di setiap baris data petani. Silakan sesuaikan code berikut.

```

<!--
- memanggil fungsi kelompoktani pada model Petani, kemudian menampilkan elemen berupa nama_kelompok -->
<!-- KELOMPOK -->
<a href="{{route('petani.kelompok',$item->kelompoktani->id_kelompok_tani)}}"> {{ $item->kelompoktani->nama_kelompok }} </a>

```

- Jangan khawatir kalau menemukan error pada bagian pagination ketika link nama kelompok diklik. Karena dengan menggunakan `hasMany` tidak dapat memberikan pagination pada data, sehingga

link pagination error. Oleh karena itu, silakan tambahkan kondisi pada bagian `<!-- PAGINATION -->` berdasarkan url path yang diakses oleh pengguna.

```
<!-- PAGINATION -->
<!-- menampilkan pagination lengkap dengan parameter pencarian-->
<!-- oleh karena itu pagination dilengkapi dengan request query-->
<!-- Memberikan kondisi berdasarkan path url -->
<?php if (Request::path()=='petani') {?>
<div class="m-4"> {{$petanis->appends(request()->query())->links()}} </div>
<?php } ?>
<!-- END PAGINATION -->
```

Tampilan halaman daftar petani tidak akan berubah sama seperti pada gambar 4.1, hanya saja nama kelompok tani sekarang telah berisi link untuk menampilkan daftar petani berdasarkan kelompok.

4. 4. EVALUASI

Silakan lakukan evaluasi terkait latihan yang telah dikerjakan. Pastikan semua check list pada bagian evaluasi ini terpenuhi. Keberhasilan pada BAB ini akan menentukan pengerjaan latihan berikutnya. Silakan centang pada checklist yang telah terpenuhi.

- ☐ Fungsi filter pada index telah sesuai dan dapat digunakan
- ☐ Form pencarian dan filter dapat digunakan
- ☐ Pencarian berdasarkan kata kunci menampilkan data dengan benar
- ☐ Filter berdasarkan kelompok tani menampilkan data dengan benar
- ☐ Filter dan pencarian dapat menampilkan data dengan benar
- ☐ Link nama kelompok tani telah dibuat
- ☐ Sistem dapat menampilkan data berdasarkan kelompok tani
- ☐ Tidak terjadi error pada pagination