Langkah 1 - Membuat Controller Post

Tentu saja, hal pertama yang harus kita lakukan adalah membuat *controller*-nya terlebih dahulu. *Controller* digunakan untuk menghubungkan antara *Model* dengan *View* dan biasanya *Controller* akan berisi logika-logika dari aplikasi yang dibangun.

Silahkan jalankan perintah berikut ini di dalam terminal/CMD dan pastikan sudah berada di dalam *project* **Laravel**-nya.

php artisan make:controller PostController

Jika perintah di atas berhasil dijalankan, maka kita akan mendpaatkan *file* *controller* baru yang berada di dalam *folder* app/Http/Controllers/PostController.php.

Silahkan buka *file* tersebut dan ubah kode-nya menjadi seperti berikut ini.

<?php

namespace App\Http\Controllers;

//import Model "Post

use App\Models\Post;

//return type View

use Illuminate\View\View;

use Illuminate\Http\Request;

class PostController extends Controller

{

/\*\*

\* index

\*

\* @return View

\*/

public function index(): View

{

//get posts

$posts = Post::latest()->paginate(5);

//render view with posts

return view('posts.index', compact('posts'));

}

}

Dari perubahan kode di atas, pertama kita *import* *Model* Post di dalam *controller*.

//import Model "Post

use App\Models\Post;

Kemudian kita import return Type View.

//return type View

use Illuminate\View\View;

Setelah itu, di dalam *class* PostController kita membuat *method* baru dengan nama index.

public function index(): View

{

//...

}

Di dalam *method* tersebut, pertama kita membuat *variable* baru dengan nama $posts dan berisi *Model* Post yang mengambil data dari *database*.

Kemudian kita memanggil *method* latest yang berfungsi untuk mengurutkan data yang akan ditampilkan berdasarkan yang paling terbaru dan membatasi data yang ditampilkan menggunakan *method* paginate sejumlah 5.

//get posts

$posts = Post::latest()->paginate(5);

Setelah data berhasil ditampung di dalam *variable* $posts, langkah selanjutnya adalah mengirimkan *variable* tersebut ke dalam *view* menggunakan *method* compact,

//render view with posts

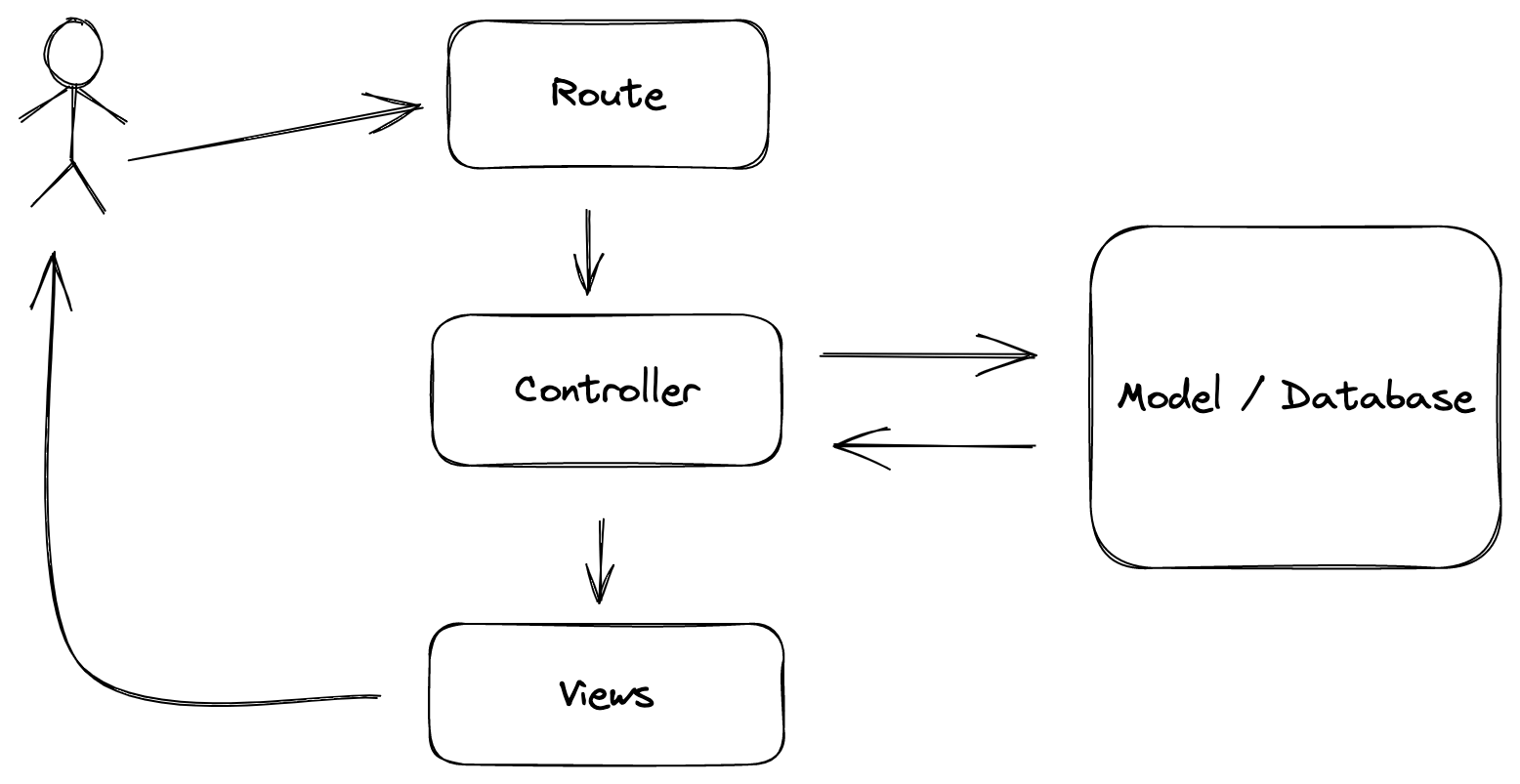
return view('posts.index', compact('posts'));

Di atas kita melakukan *render* ke dalam *view* resources/posts/index.blade.php dengan mengirimkan *variable* $posts.

Langkah 2 - Membuat Route Posts

Agar bisa mengakses *controller* melalui *URL* *browser*, maka kita perlu membuatkan *route*-nya terlebih dahulu. Jadi konsepnya kita akan memanggil *route*, kemudian *route* memanggil *controller* dan *controller* memanggil *view*.

Kurang lebih alur-nya seperti berikut ini.

[](https://raw.githubusercontent.com/maulayyacyber/assets-articles/main/laravel-10/Untitled-2023-02-09-0846.png)

Untuk membuat *Route*, silahkan teman-teman buka *file* routes/web.php, kemudian ubah semua kode-nya menjadi seperti berikut ini.

<?php

use Illuminate\Support\Facades\Route;

//route resource

Route::resource('/posts', \App\Http\Controllers\PostController::class);

Dari perubahan kode di atas, kita menambahkan *route* baru dengan path /posts dan jika teman-teman perhatikan, kita menggunakan *route* dengan jenis resource. Artinya apa? yaitu *route*-*route* untuk proses **CRUD** akan di*generate* secara otomatis oleh **Laravel**, seperti index, create, store, show, edit, update dan destroy, dibandingkan kita harus membuatnya secara manual satu-persatu.

Untuk memastikan apakah penambahan *route* kita berhasil, teman-teman bisa menjalankan perintah berikut ini di dalam terminal/CMD dan pastikan sudah berada di adlam *project* **Laravel**-nya.

php artisan route:list

Langkah 3 - Membuat View dan Menampilkan Data

Setelah *controller* dan route berhasil dibuat, maka sekarang kita akan lanjutkan belajar membuat sebuah *view* yang digunakan untuk menampilkan data.

Silahkan teman-teman buat *folder* baru dengan nama posts di dalam *folder* resources/views, kemudian di dalam *folder* posts tersebut buatlah *file* baru dengan nama index.blade.php dan masukkan kode berikut ini di dalamnya.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Data Posts</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<link rel="stylesheet" href="//cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.css">

</head>

<body style="background: lightgray">

<div class="container mt-5">

<div class="row">

<div class="col-md-12">

<div>

<h3 class="text-center my-4">Tutorial Laravel 10 untuk Pemula</h3>

<h5 class="text-center"><a href="#">Halaman</a></h5>

<hr>

</div>

<div class="card border-0 shadow-sm rounded">

<div class="card-body">

<a href="{{ route('posts.create') }}" class="btn btn-md btn-success mb-3">TAMBAH POST</a>

<table class="table table-bordered">

<thead>

<tr>

<th scope="col">GAMBAR</th>

<th scope="col">JUDUL</th>

<th scope="col">CONTENT</th>

<th scope="col">AKSI</th>

</tr>

</thead>

<tbody>

@forelse ($posts as $post)

<tr>

<td class="text-center">

<img src="{{ asset('/storage/posts/'.$post->image) }}" class="rounded" style="width: 150px">

</td>

<td>{{ $post->title }}</td>

<td>{!! $post->content !!}</td>

<td class="text-center">

<form onsubmit="return confirm('Apakah Anda Yakin ?');" action="{{ route('posts.destroy', $post->id) }}" method="POST">

<a href="{{ route('posts.show', $post->id) }}" class="btn btn-sm btn-dark">SHOW</a>

<a href="{{ route('posts.edit', $post->id) }}" class="btn btn-sm btn-primary">EDIT</a>

@csrf

@method('DELETE')

<button type="submit" class="btn btn-sm btn-danger">HAPUS</button>

</form>

</td>

</tr>

@empty

<div class="alert alert-danger">

Data Post belum Tersedia.

</div>

@endforelse

</tbody>

</table>

{{ $posts->links() }}

</div>

</div>

</div>

</div>

</div>

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

<script src="//cdnjs.cloudflare.com/ajax/libs/toastr.js/latest/toastr.min.js"></script>

<script>

//message with toastr

@if(session()->has('success'))

toastr.success('{{ session('success') }}', 'BERHASIL!');

@elseif(session()->has('error'))

toastr.error('{{ session('error') }}', 'GAGAL!');

@endif

</script>

</body>

</html>