LARAVEL

**Fitur utama laravel** : MVC, Template Engine, Artisan Console, Eloquent ORM, Authentication & Authorization, Testing, Packaging System, Multiple File System, Task Scheduling, Websocket Programming.

Folder public digunakan untuk menyimpan folder css, js, img milik kita.

Blade Template, kita memanggil data yang dibuat sama seperti php echo

<?php echo $nama; ?> jika menggunakan blade cukup {{ $nama }}

Sebenarnya halaman kita itu dikerjakan / dicompile untuk menampilkan hasilnya ada di storage/framework/views.

@extends, yang akan digunakan oleh child view

@dd, digunakan untuk melakukan debugging dan tidak akan menjalankan code di bawahnya.

Slug itu versi lain dari title, biasanya tiap spasi diganti menjadi dash. Misalkan Post Pertama slug nya menjadi post-pertama.

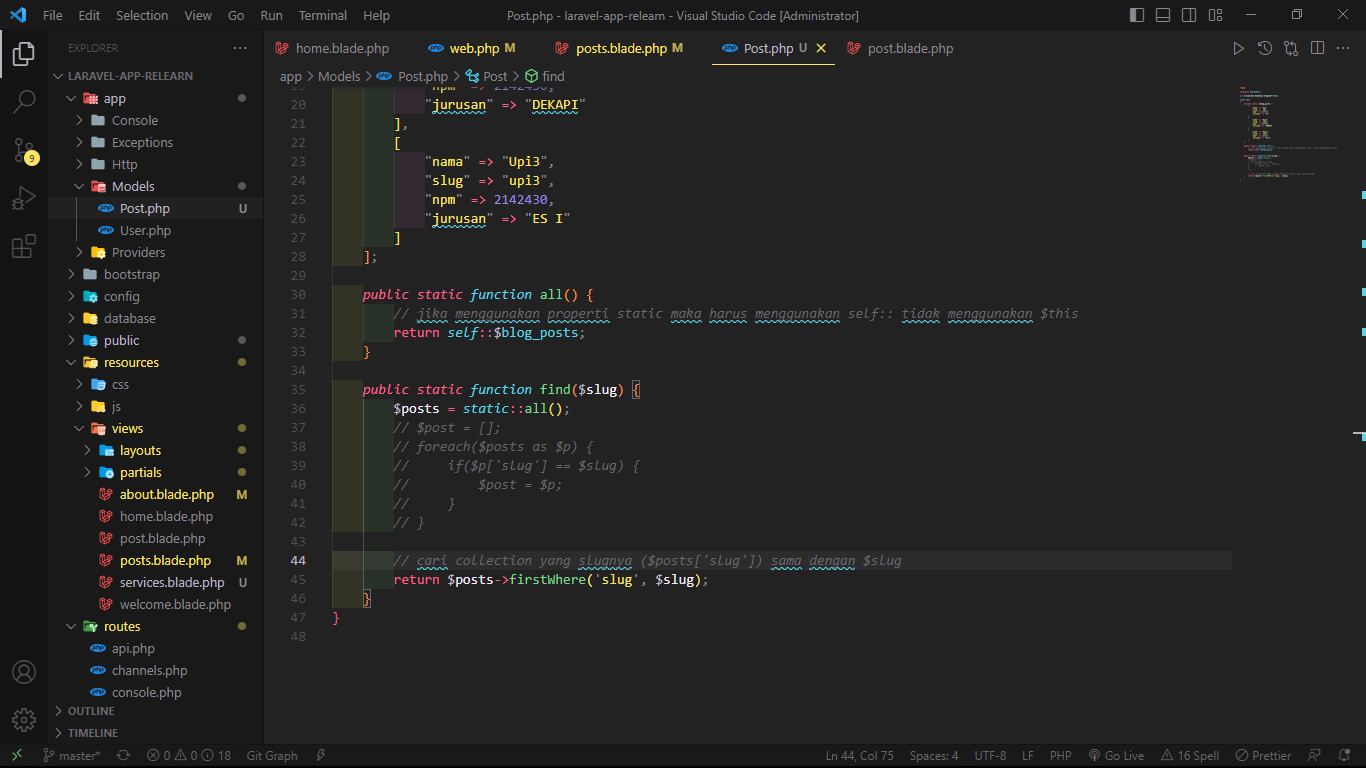
Eloquent model ORM (Object Relational Model), dengan melakukan ini setiap tabel di dalam database itu mempunyai sebuah “Model” agar dapat berinteraksi (dalam bentuk class) yang dapat dipetakkan ke dalam tabel.

Static adalah fungsi yang terikat dengan class itu sendiri, bukan dengan instance object dari class tersebut. Fungsi static dapat dipanggil tanpa membuat instance object dari class, dan dapat diakses secara langsung dengan menggunakan sintaks “::”.

Kita bisa mengubat Data yang kita ambil menjadi sebuah collection di laravel.

Collection = pembungkus dari sebuah array, yang akan membuat array menjadi lebih sakti.

Cukup dengan : collect(self::$blog\_posts);



Controller, untuk menangani closure yang dibuat di dalam route sebagai pemisah.

.env digunakan untuk menyimpan setting pada lingkungan pengembangan pada aplikasi kita

Di dalam file config ada database.php yang didalamnya akan terdapat code berikut.

'default' => env('DB\_CONNECTION', 'mysql'),

Artinya secara default kita akan menggunakan database dari db\_connection yang ada di .env nya jika tidak ada maka akan menggunakan mysql.

Buat database di phpmyadmin, langkah selanjutnya kita akan melakukan migration (seperti version controll – seperti git) ke dalam database

Php artisan make:migration posts\_table

Php artisan migrate, php artisan migrate:rollback, php artisan migrate::fresh

Eloquent ORM, Object Relation Mapper sebuah fungsi untuk memetakkan tiap tiap data yang ada di dalam table / database ke dalam sebuah object. Eloquent model ini memungkinkan kita melakukan crud

Active Record Pattern.

Php artisan tinker

$user = new App\Models\User;

$user->name = ‘Luthfi’; $user->password = bcrypt(‘1234567’);

$user->email = ‘luthfiramadhan.lr55@gmail.com’; $user->save();

$user->all();

Post::create(['title' => 'Judul 1', 'slug' => 'judul-satu', 'excerpt' => 'judul kesatu', 'body' => '1290']);

Post::find(3)->update([‘title’=> ‘judul update’]);

Post::where(‘title’, ‘Judul 1’)->update([‘excerpt’ => ‘Judul 1 Update’]);

Php artisan help make:model

php artisan help make:model -m Post

tidak akan menggunakan notasi array $post[‘nama’] akan menggunakan notasi object $post->nama. Kenapa jalan karena collection memungkinkan kita untuk bisa menggunakan 2 notasi itu

Route Model Binding, pada saat kita menyuntikkan sebuah id ke dalam controller kita akan query record yang kita cari ke dalam id tadi (kita cari berdasarkan id dan di dalam controllernya kita query cari yang idnya ini).

Route model binding ini mengskip cara untuk melakuan query, jadi laravelnya akan langsung mengquerykan supaya langsung dapat data apa yang kita cari (secara otomatis)

Jadi jika kita mengarahkan routenya ke /users/id, kita cukup ubah menjadi

Route::get(‘/users/{user}’, function(User $user) {

Return $user->email;  
});

{user} harus sama dengan $user, User adalah sebuah model

\*jadi kan kalo kita cari berdasarkan slug (route), di (controller)-nya kita biasanya tangkap si slug nya $slug, lalu untuk mencari slugnya kita ke ‘posts’ => Post::find($slug) dengan Route Model binding, kita hanya cukup tuliskan {post:slug} pada (route -> ‘karena lebih make sense’) lalu di (controller)-nya kita samakan menjadi $post untuk menangkapnya sekalian kita bisa panggil Model nya maka tuliskan (Post $post) untuk menjalankannya cukup ‘posts’ => $post

Next 8