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

Blade adalah fitur yang disediakan Laravel untuk proses templating sederhana namun sangat bermanfaat dalam proses pengembangan tampilan halaman web. Tidak seperti fitur templating PHP populer lainnya, Blade tidak membatasi pemrogram untuk menggunakan kode PHP biasa dalam membuat kode untuk tampilan.

Jika menggunakan php

*<?php echo $nama; ?>*

Jika menggunakan blade

*{{ $nama }}*

Fungsi dari Laravel Blade templating adalah penggunaan layout, agar tampilan yang berulang misalnya header, footer, sidebar dan sebagainya tidak perlu dibuat berkali-kali, seperti kita menggunakan @extends, @include, @yield, @foreach, dll

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

# Eloquent Model

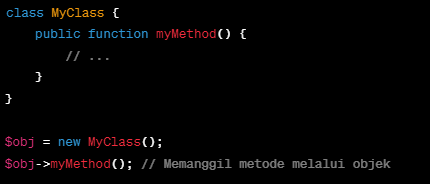
Model merupakan representasi dari object dalam database yang memungkinkan kita melakukan interaksi dengan efisien. Eloquent, sebuah object-relational (Object Relational Model) yang membuatnya mudah dalam bertintegrasi dengan database. Setiap database memiliki ‘model’ terkait yang digunakan untuk berinteraksi dengan tabel tersebut.

Static adalah function yang terkait dengan class itu sendiri, bukan instasiasi dari object class tersebut. Fungsi static dapat diakses langsung dengan menggunakan sintaks ::

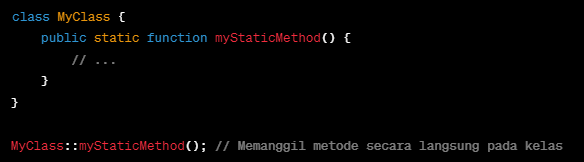
Jika static harus menggunakan keyword ***self::*** bukan this

Untuk static:: biasanya digunakan ketika mengakses variabel di dalam classnya

Tanpa Static



Dengan Static



# Collection

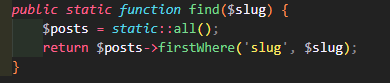
Collection adalah fitur laravel untuk memanipulasi data array. Collection bisa disebut pembungkus dari sebuah array.

Menggunakan Collection cukup dengan

***collect(self::$blog\_posts);***

*biasanya digunakan saat mengembalikan data, perlu diingat bahwa ada self:: yang menandakan bahwa $blog\_posts adalah array static yang ada di dalam class.*

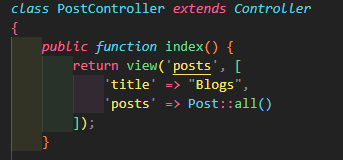
Dengan menggunakan collection membuat kita bisa menggunakan method2 yang disediakan laravel. Misalkan seperti berikut



*\*berikut adalah contoh bahwa kita akan mencari post berdasarkan slug, disini kita bisa menggunakan method firstWhere (cari yang pertama ketemu) lalu di return.*

# Controller

Controller dalam laravel adalah untuk menangani closure yang di buat di dalam route sebagai pemisah yang lebih baik.



# .env

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

Banyak file yang akan saling terhubung dengan file .env ini.

Contoh di dalam file config/database.php di dalamnya terdapat code berikut

***‘default’ => env(‘DB\_CONNECTION’, ‘mysql’)***

*\*artinya secara default database kita akan menyesuaikan dengan database yang telah kita set di env DB\_CONNECTION, jika tidak kita set maka secara default database yang digunakan adalah mysql.*

* Buat Database Baru, lalu di laravel lakukan migration.

*\*migration itu sama seperti version control (contoh: git) dimana kita bisa melihat perubahannya namun ini ke dalam database.*

php artisan make:migration posts\_table = membuat migrasi

php artisan migrate = melakukan migrasi table

php artisan migrate:rollback = menarik migrasi table

php artisan migrate:fresh = mere-migrate tabel dan isi datanya

# Eloquent ORM & Tinker

Eloquent ORM (Object Relation Mapper), adalah sebuah fungsi untuk memetakkan tiap tiap data yang ada di dalam table database ke dalam sebuah object yang kemudian memungkinkan kita untuk melakukan crud.

Active Record Pattern adalah seperti kita membuat object instansiasi baru untuk membuat data baru, selanjutnya kita melakukan crud di dalamnya.

***Laravel Tinker***

Laravel Tinker adalah sebuah REPL (Read-Eval-Print Loop) interaktif yang terintegrasi secara langsung dengan Laravel. Ini memungkinkan Anda berinteraksi dengan aplikasi Laravel Anda melalui baris perintah (command line) untuk menjalankan kode PHP dan menjelajahi model, database, dan komponen lainnya secara langsung.

php artisan tinker = untuk memulai tinker

$user = new App\Models\User; = untuk inisialisasi & menginstansiasi user baru.

$user->name = ‘Luthfi’; = mengisikan data

$user->al(); = menampilkan semua data

\*jangan lupa di modelnya harus diberikan akses mana field yang bisa diisi mana yang tidak.

App\Models\Post::create([‘title’ => ‘Lorem Ipsum 1’, ‘body’ => ‘Lorem Ipsum’]);

App\Models\Post::find(1)->update([‘title’ => ‘Title Update’, ‘body’ => ‘Lorem Ipsum’]);

App\Models\Post::where(‘title’, ‘Title Update’)->update([‘body’ => ‘Body Update’]);

Php artisan help make:model

php artisan help make:model -m Post

# Collection Notation

Pada dasarnya Collection mengembalikan data berupa object namun karena laravel mengimplementasi antarmuka ArrayAccess sehingga kita dapat menggunakan 2 notasi, seperti berikut

$post->nama; = notasi object

$post[‘nama’]; = notasi array

# Route Model Binding

Route Model Binding, saat kita menyuntikkan sebuah id ke dalam controller maka akan melakukan query berdsarkan id yang kita cari.

Route Model Binding ini akan meng-skip cara untuk melakukan querynya jadi secara otomatis akan dilakukan oleh laravel supaya dapat mencari data apa yang kita cari. Misalkan kita akan mencari post berdasarkan slug

*Route::get(‘/post/id’, function() { … } );*

*Route::get(‘/post/{post:slug}’, function (Post $post) { … } );*

*\*dengan menggunakan cara ke dua kita bisa langsung menggunakan slug*

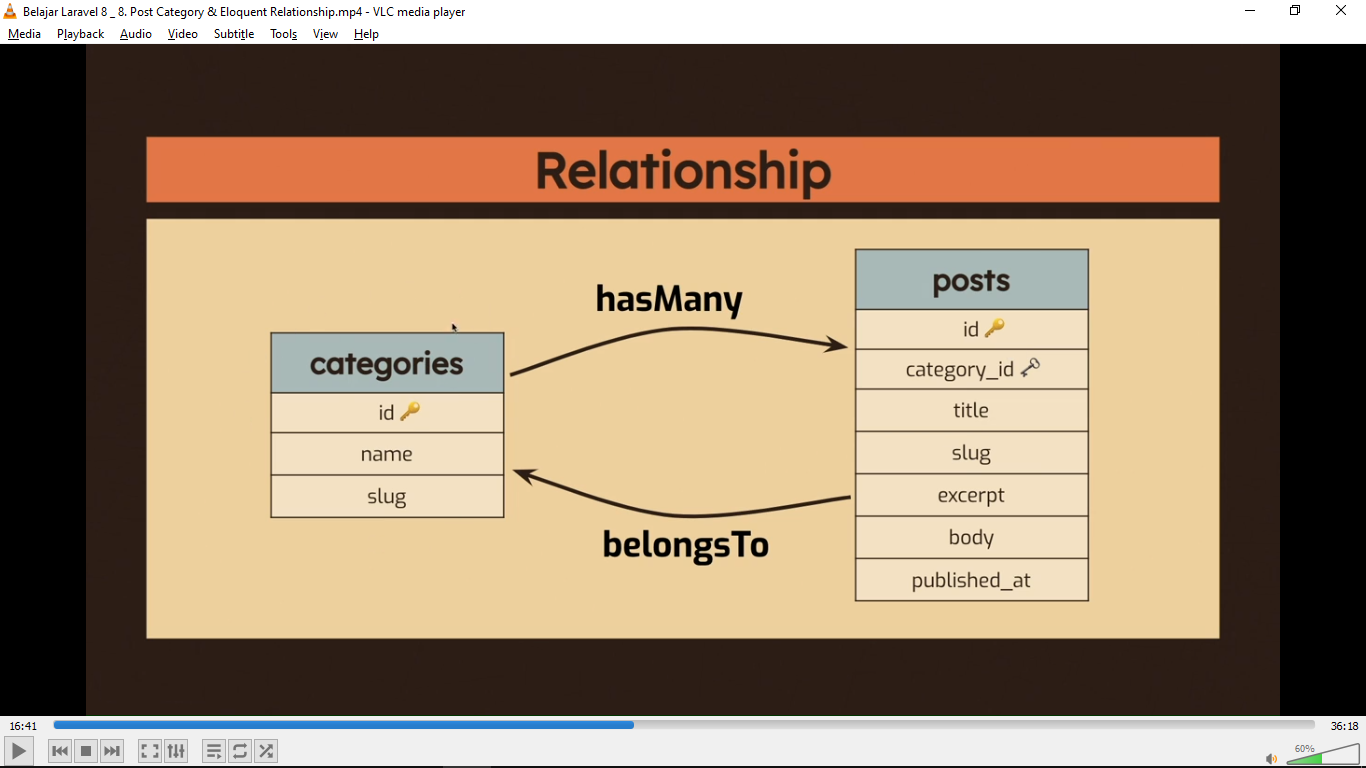
*{post} itu harus sama dengan $post.*

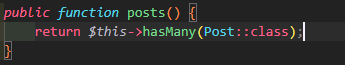
# Relationship Database One to Many

Primary Key = Setiap tabel hanya memiliki 1 primary key yang dimana nilainya harus unik dan tidak boleh sama.

Foreign Key = Sebuah kolom yang menyatakan adanya hubungan antar tabel dengan primary key atau unique key di dalam tabel lain.

*\*Setiap relasi antar tabel harus memiliki id / primary\_key untuk merujuk ke nilai tersebut dari tabel lain.*





*\*disini di dalam modelnya setiap Categori memiliki banyak postingan dan setiap postingan pasti memiliki 1 category*

# Database Seeder

Seeding, mengisi otomatis database kita dengan class seed.

Factories, pabrik untuk membuat data secara otomatis.

*php artisan db:seed = digunakan untuk menjalankan seeder*

Melakukan seeding sama seperti kita melakukan di tinker, kita bisa membuatnya di file DatabaseSeeder.php

# Factories & Faker

Buat factory baru untuk dapat membuat data secara otomatis.

*php artisan make:factory PostFactory*

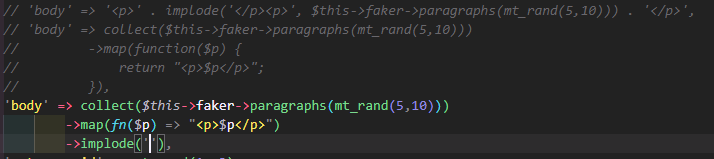
Install Faker, untuk apa? Agar kita bisa melakukan data diisi otomatis dengan data dummy / generate data random.

Pergi ke bagian config/app.php cari faker ubah faker\_locale

'faker\_locale' => env('FAKER\_LOCALE', 'en\_US'),

*php artisan migrate:fresh –seed = digunakan untuk merefresh database sekalian melakuan seeding.*

Merubah agar setiap data di factory diawali dengan huruf p per paragrafnya



# N + 1 Problem

N+1 disini artinya adalah 1 query untuk parent dan N adalah jumlah record pada table child. Untuk mengecek / melakukan debugging kita akan menggunakan Clockwork untuk laravelnya.

*Lakukan installasi clockwork “****composer require itsgoingd/clockwork****”*

*Install Extension Chrome “****clockwork****”*

Untuk memahami problemnya, misalkan kita punya posts dengan menghubungkan ke relasinya category & user. Saat melakukan pemanggilan data biasa maka akan melakukan pemanggilan ke 3 tabel di databasenya seperti *Select \* from ‘users’ limit 1*.

Lalu bagaimana jika kita melakuan foreach atau looping, maka yang terjadi pemanggilan data nya juga akan di looping beserta pemanggilan ke 3 tabel di databasenya.

* **Lazy Loading**

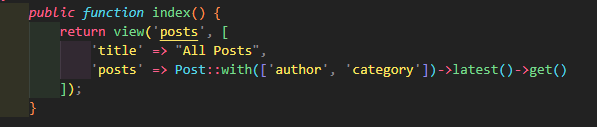
Kita melakuan pemanggilan data jika dibutuhkan saja.

(Penyebab N + 1 Problem jika datanya banyak).

* **Eager Loading**

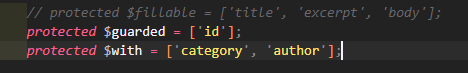
Kita Lakukan semuanya di awal ‘sehingga sudah punya datanya di awal’, sehingga saat looping tidak perlu melakukan pemanggilan ke database lagi. Dengan Eager Loading kita cukup gunakan keyword **with()**.

*PostController.php*



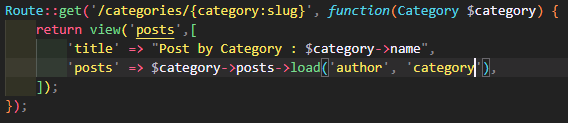
Kita juga bisa memindahkan with() ke dalam model post

*bahwa ketika kita memanggil tabel post panggil juga sekalian dengan category & author*



* **Lazy Eager Loading**

Terkadang kita butuh untuk melakukan eager loading ***\*pada relationship tapi setelah parent nya di dapatkan.*** Untuk menggunakan Lazy Eager Loading pada relationship disini kita gunakan method **load()**.

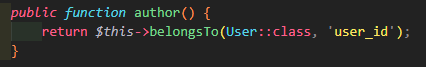


# Catatan :

Di dalam Models kita melakukan relasi dan memberikan nama pada method yang kita relasikan, misalkan: Kita mengidentifikasinya sebagai category bahwa ini sebagai penghubung relasi dari Model Category



Saat nanti di view nya kita memanggil relasinya $post->category->name itu akan bisa dan menampilkan nama category, jika nama method tidak sama dengan yang di panggil di viewnya maka akan error.



Sedangkan ini adalah contoh yang harusnya relasi dengan user diberi nama methodnya user. Karena jika nama method nya sama dengan nama modelnya maka secara otomatis akan didefinisikan oleh laravel, sehingga kita bisa akses semua.

Tapi jika kita melakukan costumize pada methodnya kita harus menambahkan parameter ‘user\_id yang menandakan bahwa ini merujuk ke foreignId yang ada di database / foreignId(‘user\_id’);