**LARAVEL 8**

[ Documentation 2](#_Toc103117496)

[ Tips atau Sering Digunakan 2](#_Toc103117497)

[ Php artisan 2](#_Toc103117498)

[ Artisan make 2](#_Toc103117499)

[ Artisan Migrate (migration) 2](#_Toc103117500)

[ Struktur Folder MVC, Routes dan View 2](#_Toc103117501)

[ Blade Templating Engine 3](#_Toc103117502)

[ Tinker 3](#_Toc103117503)

[ Seeder, Factory & Faker 4](#_Toc103117504)

[ N+1 Problem (Query Berlebihan) 5](#_Toc103117505)

# Instalasi & Documentation

* Documentataion : [https://laravel.com/docs/](https://laravel.com/docs/master/helpers)
* Normal Install : composer create-project laravel/laravel example-app
* Global Install : laravel new example-app

# Tips atau Sering Digunakan

* blade tidak menggunakan titik koma ;
* php echo : {{ $nama }}
* periksa : @dump(); atau @dd();
* @crsf : hash dibawah tag <form> pembuka
* mt\_rand(2, 8) : membuat bilangan random mt\_rand(minimal, maksimal)
* jika membuat controller menggunakan artisan make C+M Res, route bisa disingkat :

use App\Http\Controllers\StudentsController;

Route::resource('students', StudentsController::class);

# Php artisan

* php artisan db:seed : mengirim data seeder
* php artisan migrate:fresh –seed : mengirim data seeder & factory

# Artisan make

* controller : php artisan make:controller <namaController>
* model : php artisan make:model <nama>
* factory : php artisan make:factory <namaFactory>
* C Res : php artisan make:controller <namaController> --resource
* C+M Res : php artisan make:controller <namaController> -r -m <namaModel>
* M+ mfs : php artisan make:model <nama> -mfs

*ket : C = Controller m = migrate*

*M = Model f = factory*

*Res = Resource (CRUD) s = seeder*

# Artisan Migrate (migration)

* migrate : php artisan migrate
* rollback : php artisan migrate:rollback
* rollback + migrate : php artisan migrate:fresh
* migrate + seeder : php artisan migrate:fresh --seed

# Struktur Folder MVC, Routes dan View

* routes : routes/web.php : file utama yg diakses
* controller : app/Http/Controllers
* model : app/Models
* view : resources/Views
* assets : public/assets : folder assets buat manual
* .env : konfigurasi url, database, dll

# Blade Templating Engine

* membuat templates : views/layouts/main.blade.php

<head>

    <title>@yield('title')</title>

</head>

<body>

    @include('partials.navbar')

    <div class="container m-4">

        @yield('container')

    </div>

</body>

* partials (opsioanal) : views/partials/navbar.blade.php
* penggunaan di view : views/home.blade.php

@extends('layouts.main')

@section('title', 'Laravel 8 | Home')

@section('container')

    {{-- isi konten --}}

@endsection

# Tinker

* Menambah data ke table menggunakan tinker di cmd

1. jalankan tinker : php artisan tinker
2. buat variable. ada 2 cara contoh :

>>> $user = new App\Models\User;

>>> $user = new User;

1. isi attribut table yg fillable. contoh :

>>> $user->name = 'Mochamad Ihsan Saepulloh';

>>> $user->email ='mihsansaepulloh9@gmail.com';

>>> $user->password = bcrypt('09072002');

1. simpan data diatas ke table

>>> $user->save()

1. memeriksa data table

>>> $user->all()

# Seeder, Factory & Faker

* Merubah lokasi data faker :

1. ubah config : config/app.php

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

1. tambahkan baris dibawah di file .env

FAKER\_LOCALE=id\_ID

* Membuat seeder + factory

1. buat factory di cmd :

php artisan make:factory <namaFactory>

1. database/seeders/DatabaseSeeder.php

public *function* run()

{

    User::factory(3)->create();

*// manual tanpa factory*

    Category::create([

        'name'  => 'Web Programming',

        'slug'  => 'web-programming'

    ]);

    Post::factory(20)->create();

}

1. contoh factory User : database/factories/UserFactory.php

public *function* definition()

{

    return [

        'name' => *$this*->faker->name(),

        'username' => *$this*->faker->unique()->userName(),

        'email' => *$this*->faker->unique()->safeEmail(),

        'email\_verified\_at' => now(),

        'password' => '1234..’

        'remember\_token' => Str::random(10),

    ];

}

# N+1 Problem (Query Berlebihan)

* Untuk periksa jumlah query gunakan clockwork, cara instal :

1. onProject : pasang library clockwork lewat cmd

composer require itsgoingd/clockwork

1. onChrome : instal ekstensi clockwork by itsgoingd

* Untuk mencegah multi query ada 3 cara :

1. Jika di controllers/PostController.php : tambahkan with()

public *function* index()

{

    return view('posts',[

        "title" => "All Posts",

*// with(‘tabelRelasi1, dst..’)*

        "posts" => Post::with(['author', 'category'])->latest()->get()

    ]);

}

1. Jika di Web.php : untuk yg route model binding tambahkan load()

Route::get('/authors/{author:username}', *function*(User $author) {

    return view('posts', [

        'title' =>  "Post by Author : $author->name",

        'posts' => $author->posts->load('author', 'category')

    ]);

});

1. Jika di models/Post.php : tambahkan protected $with

protected $guarded = ['id'];

protected $with = ['author', 'category'];

# Searching

* controller/PostController.php : memanggil scope filter (function scopeFilter di model)

return view('posts',[

    "title" => "All Posts" . $title,

    'active' => 'posts',

*// Tambahkan filter()*

    "posts" => Post::latest()->filter(request(['search', 'category', 'author']))->get()

]);

* model/Post.php :

*// fitur searching*

public *function* scopeFilter($query, array $filters)

{

*// pengganti : isset( $filters['search'] ? $filters['search'] : false )*

*// menjadi   : $filters['search'] ?? false*

    $query->when($filters['search'] ?? false, *function*($query, $search) {

        return $query->where('title', 'like', '%' . $search . '%')

                        ->orWhere('body', 'like', '%' . $search . '%');

    });

*// menggunakan normal function*

    $query->when($filters['category'] ?? false, *function*($query, $category) {

        return $query->whereHas('category', *function*($query) use ($category) {

            $query->where('slug', $category);

        });

    });

*// menggunakan arrow function (fn(..) => ..)*

    $query->when($filters['author'] ?? false, *fn*($query, $author) =>

        $query->whereHas('author', *fn*($query) =>

            $query->where('username', $author)

        )

    );

}

* view/posts.blade.php : menambah pengkodisian dan ubah link href author/category

*{{-- tambahkan pengkondisian dibawah setelah form pembuka --}}*

<form *action*="/posts">

*{{-- Searching berdasarkan keyword dan category/author --}}*

    @if (request('category'))

    <input *type*="hidden" *name*="category" *value*="{{ request('category') }}">

    @endif

    @if (request('author'))

    <input *type*="hidden" *name*="author" *value*="{{ request('author') }}">

    @endif

...

</form>

*{{-- ubah link menjadi href="/posts?category={{ $posts[0]->category->slug }}" --}}*

<a *href*="/posts?category={{ $posts[0]->category->slug  }}" *class*="text-decoration-none">

    {{ $posts[0]->category->name }}

</a> {{ $posts[0]->created\_at->diffForHumans() }}