

REFERENSI CEPAT LARAVEL

Artisan, routing, Eloquent, Blade, middleware, auth

Artisan

Perintah Umum

php artisan serve Mulai server

php artisan make:model Name -m Buat model dengan migration

php artisan make:controller NameController Buat class controller

php artisan make:middleware Name Buat class middleware

php artisan migrate Jalankan migration yang tertunda

php artisan migrate:rollback Rollback batch migration terakhir

php artisan db:seed Jalankan database seeder

php artisan tinkex REPL interaktif untuk app

php artisan route:list Daftar semua route yang terdaftar

php artisan cache:clear Bersihkan cache aplikasi

php artisan config:clear Bersihkan config yang di-cache

php artisan queue:work Mulai memproses job antrian

Routing

Route Dasar

```
Route::get('/users', [UserController::class, 'index']);
```

```
Route::post('/users', [UserController::class, 'store']);
```

```
Route::put('/users/{id}', [UserController::class, 'update']);
```

```
Route::delete('/users/{id}', [UserController::class, 'destroy']);
```

Parameter Route & Grup

```
Route::get('/user/{id}', function (int $id) {
```

```
    return User::findOrFail($id);
```

```
});
```

```
Route::prefix('api')->middleware('auth')->group(function () {
```

```
    Route::get('/profile', [ProfileController::class, 'show']);
```

```
});
```

Fitur Route

->name('route.name') Route bernama untuk pembuatan URL

->where('id', '[0-9]+') Constraint regex pada parameter

Route::resource() Route resource RESTful (7 route)

Route::apiResource() Resource API (tanpa view create/edit)

Route::fallback() Catch-all untuk route yang tidak cocok

Controller

Resource Controller

```
class PostController extends Controller {
```

```
    public function index() {
```

```
        return view('posts.index', ['posts' => Post::all()]);
```

```
    }
```

```
    public function store(Request $request) {
```

```
        $validated = $request->validate(['title' => 'required|
```

```
max:255']);
```

```
        Post::create($validated);
```

```
        return redirect()->route('posts.index');
```

```
    }
```

```
}
```

Method Resource

index() GET /resource -- daftar semua

create() GET /resource/create -- tampilan form

store() POST /resource -- simpan baru

show(\$id) GET /resource/{id} -- tampilan satu

edit(\$id) GET /resource/{id}/edit -- form edit

update(\$id) PUT /resource/{id} -- perbarui

destroy(\$id) DELETE /resource/{id} -- hapus

Template Blade

Layout & Section

```
{{-- layouts/app.blade.php --}}
```

```
<html><body>
```

```
    @yield('content')
```

```
</body></html>
```

```
{{-- pages/home.blade.php --}}
```

```
@extends('layouts.app')
```

```
@section('content')
```

```
<h1>Home</h1>
```

```
@endsection
```

Directive

@@ \$var @@ Echo dengan HTML escaping

@@! \$html !@@ Echo raw (tanpa escaping)

@if / @elseif / @else Blok kondisional

@foreach (\$items as \$item) Loop melalui collection

@forelse / @empty Loop dengan fallback kosong

@include('partial') Sertakan Blade view lain

@component / @slot Komponen Blade yang dapat digunakan kembali

@csrf Hidden field token CSRF

@auth / @guest Periksa status autentikasi

@error('field') Tampilkan error validasi

Eloquent ORM

Dasar Model

```
class Post extends Model {
```

```
    protected $fillable = ['title', 'body', 'user_id'];
```

```
    public function user() {
```

```
        return $this->belongsTo(User::class);
```

```
    }
```

```
}
```

Querying

```
Post::all(); // all records
```

```
Post::find(1); // by primary key
```

```
Post::where('status', 'published')->get();
```

```
Post::where('views', '>', 100)->orderBy('created_at', 'desc')->
```

```
first();
```

Operasi CRUD

```
$post = Post::create(['title' => 'New', 'body' => '...']);
```

```
$post->update(['title' => 'Updated']);
```

```
$post->delete();
```

```
Post::destroy([1, 2, 3]); // delete by IDs
```

Relasi

hasOne Satu-ke-satu (User -> Phone)

hasMany Satu-ke-banyak (Post -> Comments)

belongsToMany Kebalikan dari hasOne/hasMany

belongsToMany Banyak-ke-banyak dengan tabel pivot

hasManyThrough Has-many via model perantara

Migration

Membuat Tabel

```
Schema::create('posts', function (Blueprint $table) {
```

```
    $table->id();
```

```
    $table->foreignId('user_id')->constrained()->
```

```
cascadeOnDelete();
```

```
    $table->string('title');
```

```
    $table->text('body')->nullable();
```

```
    $table->timestamps();
```

```
});
```

Tipe Kolom

(\$table->id()) Primary key BIGINT auto-increment

(\$table->string('col', 100)) VARCHAR dengan panjang opsional

(\$table->text('col')) Kolom TEXT

(\$table->integer('col')) Kolom INTEGER

(\$table->boolean('col')) Kolom BOOLEAN

(\$table->json('col')) Kolom JSON

(\$table->timestamp('col')) Kolom TIMESTAMP

(\$table->timestamps()) created_at dan updated_at

(\$table->softDeletes()) deleted_at untuk soft delete

Middleware

Custom Middleware

```
class EnsureAdmin {
```

```
    public function handle(Request $request, Closure $next) {
```

```
        if (!$request->user()?->is_admin) {
```

```
            abort(403);
```

```
        }
```

```
        return $next($request);
```

```
    }
```

Mendaftarkan & Menggunakan

```
// bootstrap/app.php
```

```
->withMiddleware(function (Middleware $middleware) {
```

```
    $middleware->alias(['admin' => EnsureAdmin::class]);
```

```
});
```

```
// In routes
```

```
Route::get('/admin', fn() => '...')->middleware('admin');
```

Middleware Bawaan

auth Wajibkan autentikasi

guest Redirect jika sudah terautentikasi

throttle:60,1 Batasi rate (60 req/menit)

verified Wajibkan verifikasi email

signed Validasi URL bertanda tangan

Autentikasi

Helper Auth

```
Auth::check(); // is user logged in?
```

```
Auth::user(); // current User model
```

```
Auth::id(); // current user ID
```

```
Auth::attempt(['email' => $e, 'password' => $p]);
```

```
Auth::logout();
```

Starter Kit

Laravel Breeze Scaffolding auth minimal (Blade atau Inertia)

Laravel Jetstream Fitur lengkap (tim, 2FA, API token)

Sanctum Autentikasi token API SPA / mobile

Passport Implementasi server OAuth2 penuh

Melindungi Route

```
Route::middleware('auth')->group(function () {
```

```
    Route::get('/dashboard', [DashController::class, 'index']);
```

```
});
```

Validasi

Validasi Controller

```
$validated = $request->validate([
```

```
    'title' => 'required|string|max:255',
```

```
    'email' => 'required|email|unique:users',
```

```
    'age' => 'nullable|integer|min:0',
```

```
]);
```

Form Request

```
class StorePostRequest extends FormRequest {
```

```
    public function rules(): array {
```

```
        return [
```

```
            'title' => 'required|max:255',
```

```
            'body' => 'required|min:10',
```

```
        ];
```

```
    }
```

```
}
```

Aturan Umum

required Field harus ada dan tidak kosong

string | integer | boolean Validasi tipe

min:N | max:M Panjang atau nilai min/maks

email Format email yang valid

unique:table,column Harus unik di tabel DB

exists:table,column Harus ada di tabel DB

in:a,b,c Harus salah satu dari nilai yang terdaftar

confirmed Membutuhkan field confirmation yang cocok

date | after:date Validasi tanggal

Pola Umum

Respons API

```
return response()->json(['data' => $users], 200);
```

```
return response()->json(['error' => 'Not found'], 404);
```

Environment & Config

```
env('APP_KEY');
```

```
config('app.name');
```

```
config(['app.debug' => true]); // read .env value
```

```
// read config value
```

```
// set at runtime
```

Helper Berguna

route('name', \$params) Hasilkan URL untuk route bernama

redirect()->route('name') Redirect ke route bernama

back()->withErrors() Redirect kembali dengan error validasi

abort(404) Lempar HTTP exception

collect(\$array) Buat Collection dari array

now() DateTime Carbon saat ini

cache()->remember() Cache nilai dengan TTL