

Referensi Cepat Laravel

Artisan, routing, Eloquent, Blade, middleware, auth

Artisan

Perintah Umum	
<code>php artisan serve</code>	Mulai server pengembangan
<code>php artisan make:model Name -m</code>	Buat model dengan migration
<code>php artisan make:controller NameController</code>	Buat class controller
<code>php artisan make:middleware Name</code>	Buat class middleware
<code>php artisan migrate</code>	Jalankan migration yang tertunda
<code>php artisan migrate:rollback</code>	Rollback batch migration terakhir
<code>php artisan db:seed</code>	Jalankan database seeder
<code>php artisan tinker</code>	REPL interaktif untuk app
<code>php artisan route:list</code>	Daftar semua route yang terdaftar
<code>php artisan cache:clear</code>	Bersihkan cache aplikasi
<code>php artisan config:clear</code>	Bersihkan config yang di-cache
<code>php artisan queue:work</code>	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

<code>->name('route.name')</code>	Route bernama untuk pembuatan URL
<code>->where('id', '[0-9]+')</code>	Constraint regex pada parameter
<code>Route::resource()</code>	Route resource RESTful (7 route)
<code>Route::apiResource()</code>	Resource API (tanpa view create/edit)
<code>Route::fallback()</code>	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

<code>index()</code>	GET /resource -- daftar semua
<code>create()</code>	GET /resource/create -- tampilkan form
<code>store()</code>	POST /resource -- simpan baru
<code>show(\$id)</code>	GET /resource/{id} -- tampilkan satu
<code>edit(\$id)</code>	GET /resource/{id}/edit -- form edit
<code>update(\$id)</code>	PUT /resource/{id} -- perbarui
<code>destroy(\$id)</code>	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

<code>{{ \$var }}</code>	Echo dengan HTML escaping
<code>{!! \$html !!}</code>	Echo raw (tanpa escaping)
<code>@if / @elseif / @else</code>	Blok kondisional
<code>@foreach (\$items as \$item)</code>	Loop melalui collection
<code>@forelse / @empty</code>	Loop dengan fallback kosong
<code>@include('partial')</code>	Sertakan Blade view lain
<code>@component / @slot</code>	Komponen Blade yang dapat digunakan kembali
<code>@csrf</code>	Hidden field token CSRF
<code>@auth / @guest</code>	Periksa status autentikasi
<code>@error('field')</code>	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

<code>hasOne</code>	Satu-ke-satu (User->Phone)
<code>hasMany</code>	Satu-ke-banyak (Post->Comments)
<code>belongsToMany</code>	Kebalikan dari hasOne/hasMany
<code>belongsToMany</code>	Banyak-ke-banyak dengan tabel pivot
<code>hasManyThrough</code>	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

<code>\$table->id()</code>	Primary key BIGINT auto-increment
<code>\$table->string('col', 100)</code>	VARCHAR dengan panjang opsional
<code>\$table->text('col')</code>	Kolom TEXT
<code>\$table->integer('col')</code>	Kolom INTEGER
<code>\$table->boolean('col')</code>	Kolom BOOLEAN
<code>\$table->json('col')</code>	Kolom JSON
<code>\$table->timestamp('col')</code>	Kolom TIMESTAMP
<code>\$table->timestamps()</code>	created_at dan updated_at
<code>\$table->softDeletes()</code>	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');
```

Referensi Cepat Laravel

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