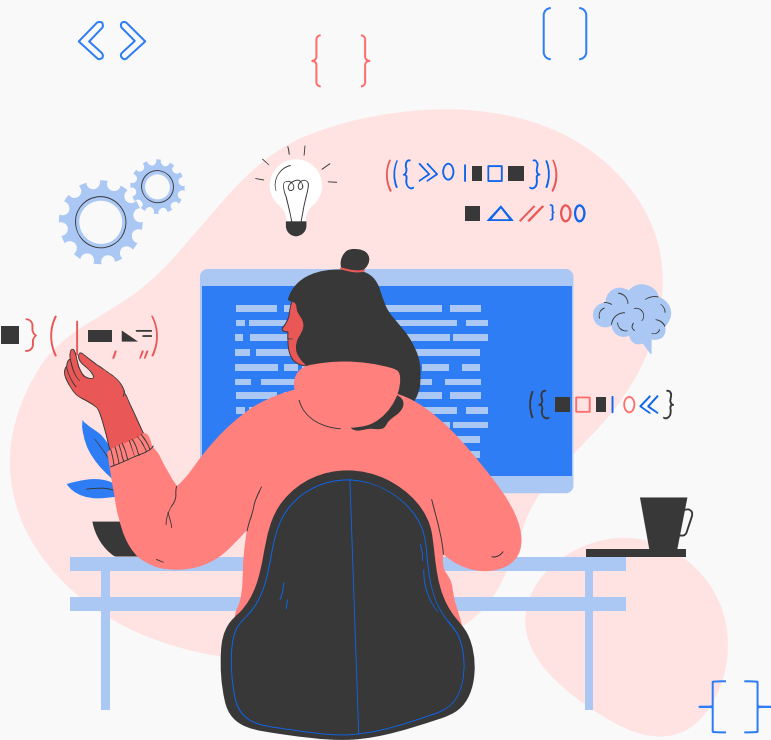


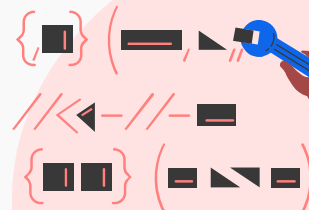
INTRO MVC & LARAVEL

Pemrograman Web #9



01

WHAT



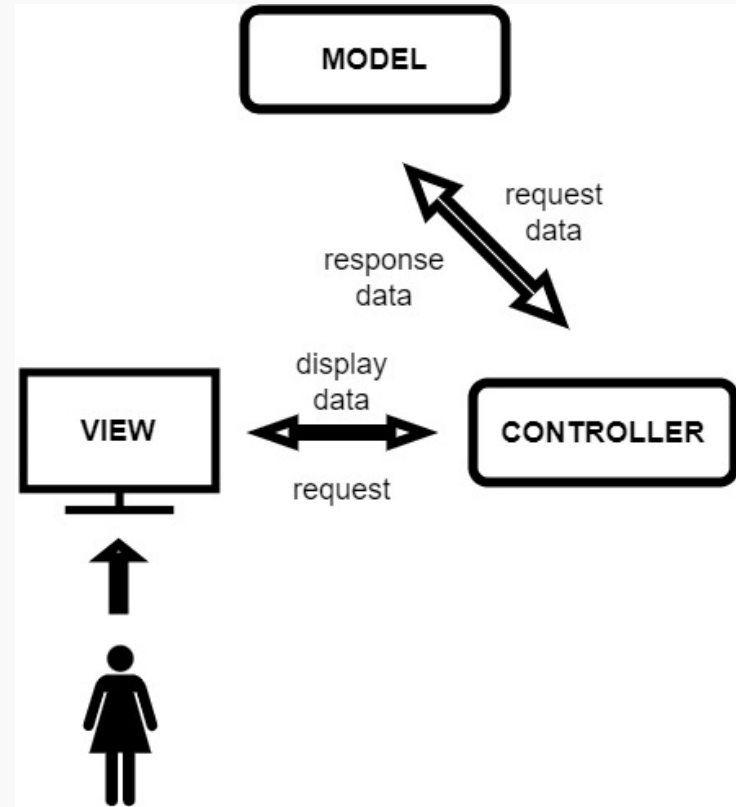


Bagian I

MVC

- suatu pola arsitektur aplikasi
- Setiap bagian aplikasi **punya tugas masing-masing**

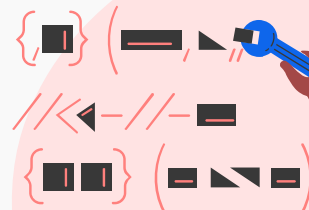
- Model - kode yang bertugas untuk membuat pemodelan data/bisnis
- View - kode yang bertugas untuk membuat tampilan aplikasi
- Controller - menyatukan Model dan View, memproses request dari view



[]

02

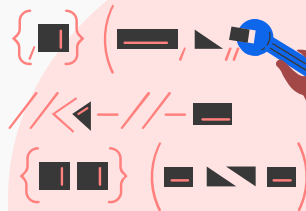
WHY



- mudah pembagian pekerjaan – large scale app, programming dapat berjalan parallel
- Konsisten – tidak ada spaghetti code
- maintenance mudah – mudah diextend (scalable)
- support TDD (Test-Driven Development) – setiap komponen MVC dapat diuji secara terpisah
- SEO friendly

03

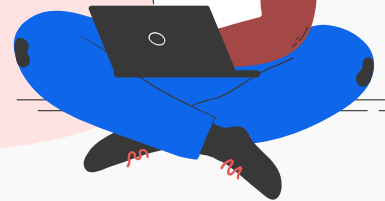
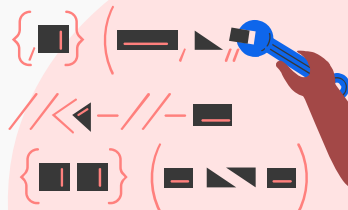
WHEN



Kriteria	MVC (Model-View-Controller)	Native / Non-Arsitektural
Skala Aplikasi	Menengah hingga besar	Kecil / sangat sederhana
Kompleksitas	Banyak fitur, interaksi kompleks	Satu atau dua fitur sederhana
Tim Pengembang	Dikerjakan oleh tim (perlu pembagian tugas)	Dikerjakan sendirian
Kecepatan Pembuatan Awal	Lebih lambat (karena butuh setup dan struktur)	Cepat (langsung coding, tanpa aturan khusus)
Tujuan Penggunaan	Aplikasi siap pakai & scalable (aplikasi akan terus berkembang)	Demo, latihan, quick prototype

04

WHO





{ }

front end developer

back end developer

[]

05

HOW



{ }

CI4 Folder

petanikode.com

.github

Folder untuk konfigurasi repo Github

app

model

view

controller

Folder app berisi kode aplikasi.
Di sinilah nantinya kita akan menulis kode

public

Berisi file yang bisa diakses oleh publik
seperti index.php

tests

Berisi kode untuk melakukan testing

vendor

Berisi library yang dibutuhkan aplikasi.
Folder ini dibuat oleh composer

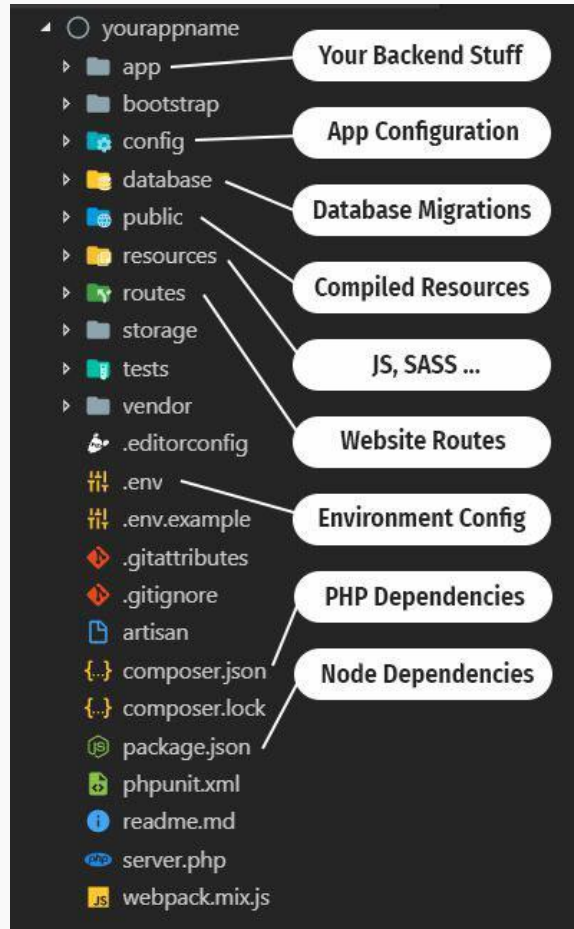
writable

Folder yang bisa ditulis oleh aplikasi.
Biasanya digunakan untuk menyimpan
file, log, session, dll.

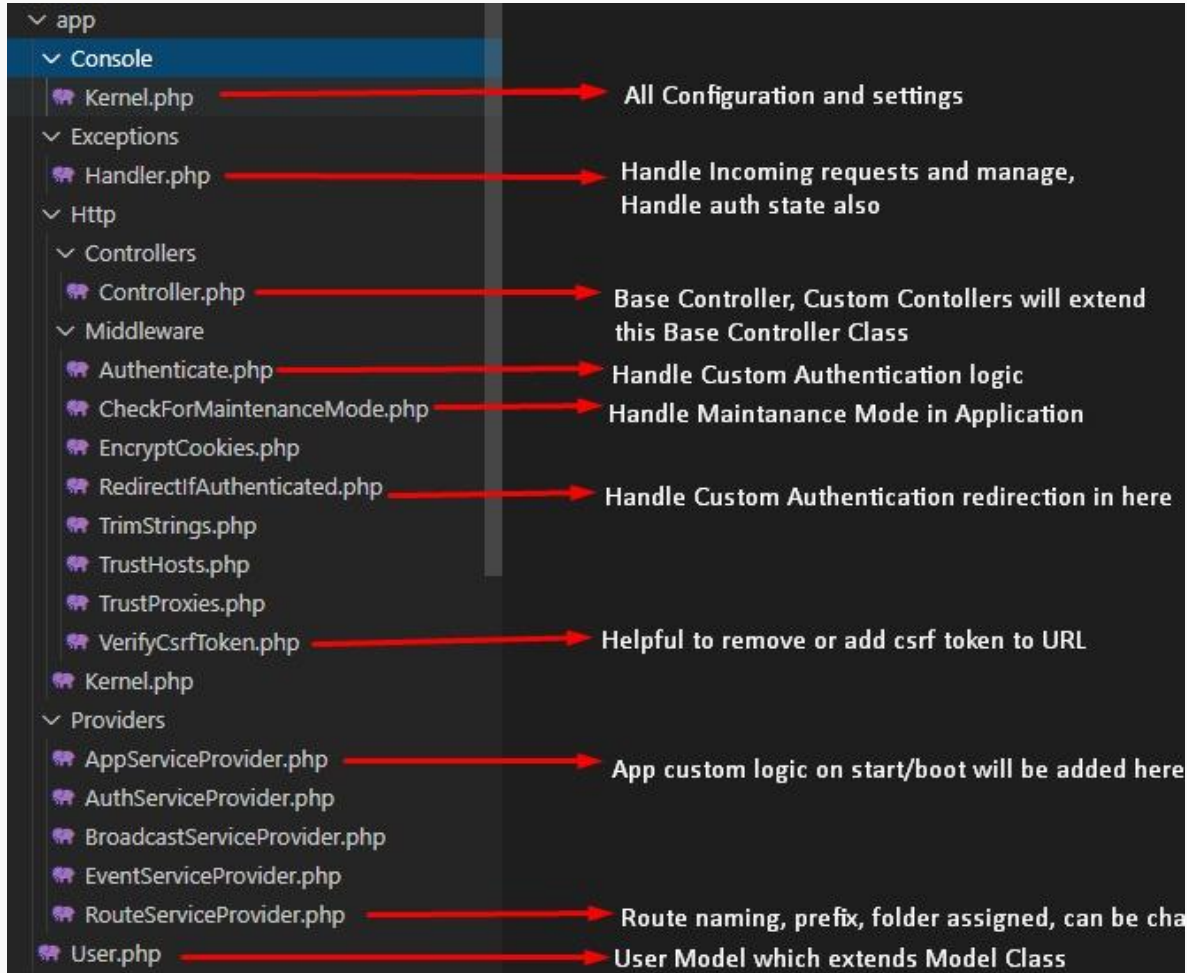
⋮

{ }

LARAVEL



LARAVEL



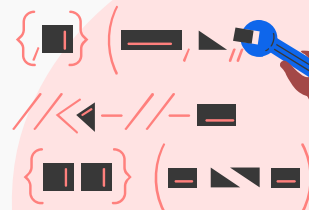


Bagian II

LARAVEL

01

WHAT



{ }

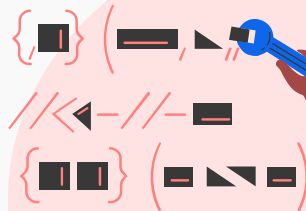
- Salah satu framework PHP yang dikembangkan oleh Taylor Otwell
- **didukung oleh server APACHE dan Nginx.**

[]

Fitur Laravel	Fungsi
Bundles (<i>sekarang disebut "Packages"</i>)	Untuk menambahkan fitur eksternal ke aplikasi Laravel. Contoh: payment gateway, excel export, dsb. Bisa diinstal via Composer.
Eloquent ORM	Abstraksi database berbasis objek. Untuk akses data seperti: <code>\$user->posts</code> tanpa query SQL manual.
Application Logic	Tempat menaruh logika bisnis aplikasi, biasanya diletakkan di Controller, Service, atau di dalam model.
Reverse Routing	Membuat URL berdasarkan nama rute, bukan hardcoded URL.
RESTful Controllers	Membuat controller yang mengikuti konvensi REST: index, show, store, update, destroy. Laravel bisa menggenerate otomatis.
Class Auto Loading	Laravel (via Composer) otomatis memuat file class yang dibutuhkan tanpa <code>require_once</code> .
IoC (Inversion of Control) Container	Menyimpan semua dependency. Membuat aplikasi lebih modular dan testable.
Composers (View Composers)	Menyuntikkan data ke view tertentu secara otomatis. Misal: sidebar di semua halaman mendapatkan data user tanpa diset manual di controller.
Migrations	Versi "source control"-nya database. Menulis struktur tabel dalam kode PHP, bisa diubah & rollback. Aman dan konsisten untuk tim.
Unit Testing	Laravel mendukung testing otomatis (PHPUnit). Dapat menguji controller, model, service, dll untuk memastikan kode jalan dengan benar.
Automatic Pagination	Fitur pagination langsung dari query Eloquent. Untuk membuat pagination otomatis.

02

WHY



- Penyempurnaan Resources
- Mudah dan Dokumentasi lengkap
- Open Source
- Arsitektur MVC
- Blade Template
- Fitur Migration
- Security
- Komunitas yang besar

03 HOW



1. Instalasi Laravel
2. Buat project
3. Konfigurasi env
4. Buat table/database
5. Buat model, controller, view
6. Buat route
7. Kembangkan fitur

Thanks!



CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

Please keep this slide for attribution

