

Laporan Modul 1: Perkenalan Laravel

Mata Kuliah: Workshop Web Lanjut

Nama: AFDAL FAHRIZA

NIM: 2024573010038

Kelas: TI-2B

Abstrak

Tujuan dari laporan ini adalah untuk mengenal lebih dalam tentang laravel, cara instalasi, hingga cara menggunakannya.

1. Pendahuluan

- Tuliskan teori perkenalan tentang laravel Laravel adalah sebuah framework PHP open-source yang elegan, ekspresif, dan dirancang untuk mempermudah proses pengembangan web dengan mengikuti prinsip-prinsip yang rapi dan modular. Diciptakan oleh Taylor Otwell, Laravel hadir untuk menjawab kebutuhan developer akan alat yang powerful namun menyenangkan untuk digunakan, sehingga proses coding menjadi lebih efisien dan aman.

Framework ini dibangun dengan memanfaatkan komponen-komponen dari framework Symfony, yang memastikan fondasinya kuat dan andal. Filosofi inti Laravel adalah untuk meningkatkan pengalaman developer (developer experience) tanpa mengorbankan fungsionalitas aplikasi.

- Apa itu Laravel? Laravel adalah framework PHP untuk pengembangan aplikasi web yang mengadopsi pola arsitektur Model-View-Controller (MVC).

Secara sederhana, Laravel menyediakan sekumpulan alat (tools) dan struktur kode yang sudah terorganisir rapi. Alih-alih membangun segala sesuatu dari nol (misalnya, sistem autentikasi, routing, atau ORM), developer dapat menggunakan fitur-fitur yang sudah disediakan oleh Laravel. Hal ini memungkinkan developer untuk fokus pada logika bisnis unik dari aplikasi mereka, sehingga mempercepat waktu pengembangan dan menghasilkan kode yang lebih mudah dipelihara.

- Karakteristik utama (MVC, opinionated, dsb.) Arsitektur Model-View-Controller (MVC)

Laravel menganut pola MVC secara ketat, yang memisahkan logika aplikasi into tiga komponen inti:

Model: Mewakili struktur data dan berinteraksi dengan database.

View: Bertanggung jawab untuk menampilkan antarmuka pengguna (UI) kepada user (biasanya file blade template).

Controller: Menangani permintaan user, memproses data menggunakan Model, dan melemparkan data tersebut ke View.

Keuntungan: Pemisahan ini membuat kode lebih terorganisir, mudah di-debug, dan memungkinkan front-end dan back-end developer untuk bekerja secara paralel.

Eloquent ORM (Object-Relational Mapping)

Ini adalah fitur andalan Laravel. Eloquent menyediakan cara yang sangat sederhana dan ekspresif untuk berinteraksi dengan database. Alih-alih menulis query SQL manual, developer dapat berinteraksi dengan tabel database seperti layaknya sebuah objek dalam PHP.

Contoh: `$user = User::find(1);` akan mengambil data user dengan id=1 dari tabel users.

Artisan CLI (Command-Line Interface)

Laravel dilengkapi dengan alat command-line bawaan yang disebut Artisan. Artisan dapat digunakan untuk melakukan tugas-tugas otomatis dan berulang, seperti:

Membuat file Controller, Model, Migration, dll (`php artisan make:model Product`).

Menjalankan migrasi database (`php artisan migrate`).

Menghasilkan kunci aplikasi, clearing cache, dan banyak lagi.

Sistem Routing yang Powerful dan Fleksibel

Laravel memungkinkan developer untuk mendefinisikan rute aplikasi dengan sangat mudah dan intuitif di dalam file `routes/web.php` atau `routes/api.php`. Routing ini dapat diarahkan ke closure sederhana atau ke Controller.

Sistem Template Blade

Blade adalah mesin template yang ringan namun powerful. Blade memungkinkan developer untuk menggunakan kode PHP biasa dalam view, tetapi dengan sintaks yang lebih bersih dan ekspresif. Fitur seperti template inheritance dan sections memungkinkan pembuatan layout yang konsisten tanpa duplikasi kode.

"Batteries Included" & Opinionated

Laravel bersifat "batteries included", artinya ia menyediakan hampir semua yang Anda butuhkan untuk membangun aplikasi web modern secara out-of-the-box, seperti:

Autentikasi (login, register, reset password).

Autorisasi (policies, gates).

Emailing.

Queue (antrian untuk memproses tugas berat di background).

Task Scheduling.

Caching.

Storage Abstraction (untuk bekerja dengan sistem file lokal atau cloud seperti S3).

Meski kaya fitur, Laravel juga cukup fleksibel. Ia memiliki opini tentang cara terbaik untuk melakukan sesuatu (sehingga memudahkan pemula), tetapi tidak memaksa Anda untuk selalu mengikutinya jika Anda memiliki kebutuhan khusus.

Keamanan

Laravel menangani banyak aspek keamanan secara otomatis, seperti:

Proteksi terhadap SQL Injection (berkat Eloquent ORM yang menggunakan PDO binding).

Proteksi terhadap Cross-Site Request Forgery (CSRF) dengan token otomatis.

XSS (Cross-Site Scripting) protection saat menggunakan output di Blade (`{{ $variable }}` akan di-escape secara otomatis).

Hashing password yang aman menggunakan bcrypt.

Ekosistem yang Luas

Laravel bukan hanya framework inti. Ia memiliki ekosistem yang besar seperti:

Laravel Forge / Vapor: Untuk deployment dan server management.

Laravel Nova: Admin panel premium.

Laravel Echo & Broadcasting: Untuk fitur real-time.

Laravel Sail: Environment development menggunakan Docker.

Community Packages: Ribuan package pihak ketiga yang tersedia di Packagist.

- Untuk jenis aplikasi apa Laravel cocok? Laravel sangat serbaguna dan cocok digunakan untuk membangun berbagai macam jenis aplikasi web, mulai dari yang sederhana hingga sangat kompleks:

1. aplikasi Enterprise & Skala Besar (Large-Scale Applications) -Sistem Manajemen Perusahaan (ERP, CRM).

-Sistem e-Commerce yang kompleks (dengan bantuan package seperti Laravel Cashier).

-Platform SaaS (Software-as-a-Service).

2. Aplikasi Menengah dan Korporat

portal berita atau blog.

Sistem manajemen konten (CMS) custom.

Aplikasi manajemen proyek dan tim.

Sistem inventory dan manajemen toko.

3. Aplikasi Kecil dan Sederhana

Website profil perusahaan atau landing page.

Formulir kontak yang kompleks dengan fitur upload.

Situs web dengan sistem membership sederhana.

4. API Backend (RESTful API) Laravel sangat bagus untuk membangun backend API yang solid untuk melayani:

Aplikasi Single-Page Application (SPA) yang dibangun dengan Vue.js, React, atau Angular.

Aplikasi Mobile (iOS/Android).

Aplikasi Desktop.

5. Aplikasi Real-Time Dengan integrasi Laravel Echo dan Pusher/Laravel WebSockets, Laravel dapat digunakan untuk membuat aplikasi real-time seperti:

Chat application.

Live notification system.

Dashboard yang menampilkan update data secara live.

3. Berikan penjelasan untuk setiap folder dan files yang ada didalam struktur sebuah project laravel.

1. app (Application Core) Inti dari aplikasi. Berisi hampir semua kelas PHP yang kita buat sendiri.
2. bootstrap Berisi file yang menjalankan proses "bootstrapping" atau memulai framework.
3. config/ (Configuration) Berisi semua file konfigurasi aplikasi.
4. database/ (Database Related) Berisi segala sesuatu yang berhubungan dengan data
5. public/ (Web Root) Folder publik yang diakses langsung oleh web browser.
6. resources/ (Raw Assets & Templates) Berisi file sumber yang belum dikompilasi.
7. routes/ (Application Routes) Berisi semua definisi rute (endpoint) aplikasi.
8. storage/ (Storage) Tempat penyimpanan yang generated oleh framework.
9. tests/ (Automated Tests) Berisi semua test case Anda (Unit tests, Feature tests) yang dibuat dengan PHPUnit.
10. vendor/ (Third-Party Code) Berisi semua dependensi / library pihak ketiga yang diinstall oleh Composer (e.g., framework Laravel sendiri, Symfony components, package lain). Folder ini jangan diedit manual.

.env & .env.example (Environment Configuration) .env: File PALING PENTING yang menyimpan konfigurasi environment-specific (kunci rahasia). Berisi variabel seperti koneksi database, API keys, dll. File ini tidak boleh di-commit ke version control (sudah tercantum di .gitignore) untuk keamanan.

.env.example: Template dari file .env. Berisi struktur variabel tanpa nilai sebenarnya. File ini yang di-commit, digunakan sebagai panduan untuk membuat file .env di environment baru.

.gitignore Memberi tahu Git file/folder mana yang tidak boleh di-track. Sudah dikonfigurasi untuk Laravel (mengabaikan .env, vendor/, storage/app/public, dll).

artisan (Laravel CLI Tool) Command-line interface untuk Laravel. Digunakan untuk menjalankan perintah Artisan (php artisan ...) untuk membuat migrasi, model, menjalankan server, dll.

composer.json & composer.lock (PHP Dependencies) composer.json: Mendefinisikan semua dependensi PHP yang dibutuhkan project (termasuk laravel/framework) dan konfigurasi autoload.

composer.lock: Mencatat versi exact dari setiap dependensi yang diinstall. Memastikan semua developer menginstall versi library yang sama.

package.json (JavaScript/CSS Dependencies) Seperti composer.json tapi untuk assets front-end (Node.js/NPM). Mendefinisikan dependensi seperti vite, laravel-vite-plugin, axios, bootstrap, dll.

phpunit.xml (PHPUnit Configuration) File konfigurasi untuk PHPUnit, framework testing yang digunakan Laravel.

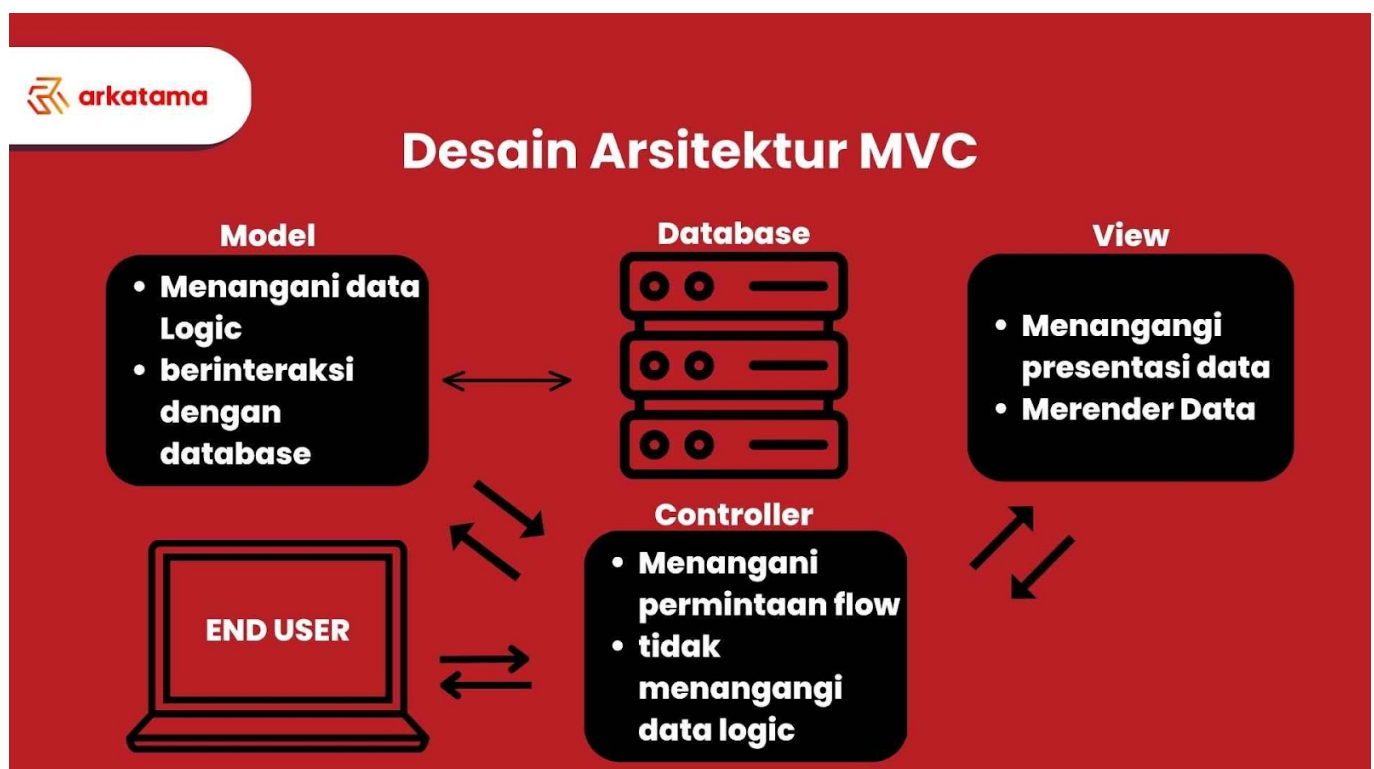
vite.config.js (Vite Configuration) File konfigurasi untuk Vite, alat build tool modern yang digunakan Laravel untuk mengompilasi dan mem-bundle assets CSS dan JavaScript.

README.md Documentation project Anda. Biasanya berisi instruksi instalasi dan penjelasan singkat tentang project.

4. Diagram MVC dan Cara kerjanya

Letakkan gambar di dalam folder `laporan1/gambar/`. Kemudian masukkan gambar tersebut ke laporan.

lihat cara nya disini <https://www.ulas.in/komputer/markdown-memasukkan-gambar/>



cara kerjanya Diagram arsitektur MVC menggambarkan pemisahan aplikasi menjadi tiga komponen utama: Model (logika bisnis dan data), View (antarmuka pengguna), dan Controller (menghubungkan pengguna dan aplikasi). Cara kerjanya adalah: pengguna berinteraksi dengan View, yang mengirimkan permintaan ke Controller. Controller kemudian berinteraksi dengan Model untuk mengambil atau memodifikasi data, lalu Model mengirimkan data kembali ke Controller. Controller lalu

memperbarui View, yang akhirnya menampilkan data yang relevan kepada pengguna.

6. Kelebihan & Kekurangan (refleksi singkat)

- Kelebihan Laravel menurut Anda
 1. Fitur yang Komprehensif (Batteries Included)
 2. Pengalaman Pengembang yang Diprioritaskan (Developer Experience)
 3. Dukungan Komunitas dan Ekosistem yang Luas
 4. Dokumentasi yang Terstruktur dan Jelas
 5. Penerapan Keamanan yang Proaktif
 6. Struktur yang Terpola namun Tetap Fleksibel
- Hal yang mungkin menjadi tantangan bagi pemula
 1. Pemahaman konsep Model-View-Controller (MVC) menjadi kunci utama yang sering kali masih terbatas bagi pengembang pemula.
 2. Struktur direktori dan banyaknya komponen baru seperti Model, Controller, Migration, dan Seeder pada awalnya dapat menimbulkan rasa kewalahan.
 3. Proses penyiapan lingkungan pengembangan (environment setup) sering menjadi penghalang pertama yang menantang.
 4. Fitur-fitur canggih seperti Eloquent ORM yang mengandalkan metode dinamis PHP sering kali terasa membingungkan.
 5. Kurangnya pelatihan dalam proses pelacakan kesalahan (debugging) menjadi kendala signifikan lainnya.
 6. Kecenderungan untuk tidak memanfaatkan fitur lengkap framework juga sering terjadi.

7. Referensi

Cantumkan sumber yang Anda baca (buku, artikel, dokumentasi) — minimal 2 sumber. Gunakan format sederhana (judul — URL).

1. Laravel Documentation (The Laravel Framework) — <https://laravel.com/docs/11.x>
 2. <https://hackmd.io/@mohdrzu/By0Wc1Dule>
-