

LAPORAN PRAKTIKUM PEMROGRAMAN WEB 2

PERTEMUAN 2

INSTALASI DAN PENGENALAN LARAVEL



Disusun Oleh:

Wildan Dzaky Ramadhani

22/505766/SV/21917

Dosen Pengampu:

Dinar Nugroho Pratomo, S.Kom., M.IM., M.Cs.

Faza Maula Azif, S.Kom., M.Eng.

PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT LUNAK

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2023

DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
BAB II.....	3
2.1. Composer.....	3
2.2. Laravel.....	3
BAB III.....	5
3.1. Instalasi Composer.....	5
3.2. Penggunaan Laravel.....	8
BAB IV.....	13
4.1. Kesimpulan.....	13
DAFTAR PUSTAKA.....	14

DAFTAR GAMBAR

Gambar 1 Halaman utama web composer.....	5
Gambar 2 Composer installer.....	5
Gambar 3 Opsi Instalasi.....	6
Gambar 4 Mengatur tempat instalasi composer.....	6
Gambar 5 Opsi menggunakan proxy.....	7
Gambar 6 Instalasi composer.....	7
Gambar 7 Composer telah selesai terinstall.....	8
Gambar 8 Tampilan depan web dokumentasi Laravel.....	8
Gambar 9 Masuk ke folder tempat project akan digunakan.....	9
Gambar 10 Perintah membuat projek menggunakan laravel.....	9
Gambar 11 Menjalankan perintah untuk membuat projek.....	9
Gambar 12 laravel berhasil terinstall.....	9
Gambar 13 perintah untuk masuk ke folder projek.....	10
Gambar 14 Perintah menjalankan project laravel.....	10
Gambar 15 Mengakses project laravel.....	10
Gambar 16 Tampilan awal projek Laravel.....	10
Gambar 17 Struktur folder pada laravel.....	11

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam konteks praktikum pemrograman web, mahasiswa sering dihadapkan pada permasalahan kompleks terkait pengembangan aplikasi web. Salah satu masalah yang sering muncul adalah efisiensi dalam mengelola dan mengembangkan kode sumber, terutama ketika proyek menjadi lebih besar dan rumit. Tugas-tugas rutin seperti mengelola rute, berinteraksi dengan basis data, dan mengatur sesi dapat menjadi rumit dan memakan waktu.

Laravel, yang merupakan salah satu framework PHP yang sangat populer, memberikan solusi yang kuat untuk tantangan tersebut. Didesain dengan baik, Laravel memudahkan pengembang dalam membangun aplikasi web dengan cepat dan efisien. Fitur-fitur seperti sistem routing yang kuat, ORM (*Object-Relational Mapping*) yang canggih, dan berbagai alat bantu lainnya, membantu menghemat waktu dan usaha dalam pengembangan aplikasi web yang berkualitas. Oleh karena itu, praktikum ini akan membahas mengapa Laravel menjadi pilihan yang bijak untuk mengatasi permasalahan tersebut.

Laravel merupakan salah satu framework web yang sangat diminati dan digunakan secara luas di industri saat ini. Pemahaman terhadap Laravel akan meningkatkan keterampilan dan daya saing para pengembang web, termasuk penggunaan konsep MVC (*Model-View-Controller*) dan pengaturan proyek yang terstruktur. Ini memberikan dasar yang kuat bagi mahasiswa yang berencana melanjutkan karir dalam pengembangan web. Praktikum ini diharapkan memberikan pemahaman yang mendalam tentang Laravel serta penerapannya dalam pengembangan aplikasi web, sehingga mahasiswa dapat mengatasi tantangan pengembangan web secara efisien. Diharapkan pula bahwa pengetahuan yang diperoleh akan diaplikasikan dalam proyek-proyek masa depan, meningkatkan kualitas dan produktivitas dalam pengembangan aplikasi web secara keseluruhan.

1.2.Rumusan Masalah

1. Bagaimanakah cara melakukan instalasi Laravel yang tepat dan benar?.
2. Apa yang harus diperhatikan saat melakukan instalasi Laravel pada lingkungan produksi?.

BAB II

LANDASAN TEORI

2.1. Composer

Composer merupakan perangkat lunak manajemen dependensi yang memiliki peran sangat krusial dalam ekosistem pengembangan aplikasi berbasis PHP. Fungsi utamanya adalah memberikan dukungan kepada para pengembang dalam mengelola dependensi perangkat lunak yang diperlukan dalam proyek PHP mereka. Dependensi ini mengacu pada pustaka atau paket-paket pihak ketiga yang menjadi kebutuhan dalam suatu proyek. Composer memungkinkan pengembang untuk mendefinisikan daftar paket ini dalam berkas 'composer.json'. Berkas tersebut berfungsi sebagai panduan bagi Composer untuk melakukan instalasi serta pengelolaan dependensi secara otomatis sesuai dengan spesifikasi yang telah ditetapkan.

Salah satu keunggulan utama dari Composer adalah kemampuannya dalam mengelola instalasi paket secara otomatis, menghilangkan kebutuhan untuk melakukan instalasi manual yang kompleks. Composer juga secara otomatis menghasilkan autoloader, yang mempermudah penggunaan kelas-kelas dari paket-paket tersebut tanpa perlu melakukan pengaturan manual.

Secara keseluruhan, Composer menjadi alat yang sangat penting dalam pengembangan aplikasi PHP modern. Alat ini memberikan pengelolaan dependensi yang efisien, menghemat waktu, dan memungkinkan para pengembang untuk fokus pada proses pengembangan aplikasi itu sendiri. Composer juga menjadi inti dari ekosistem PHP modern, memberikan akses kepada berbagai paket-paket pihak ketiga yang dapat digunakan dalam beragam proyek PHP.

2.2. Laravel

Laravel adalah framework PHP yang sangat populer untuk pengembangan aplikasi web modern. Framework ini menyederhanakan proses pengembangan dengan aturan dan alat yang jelas. Salah satu keunggulan utama Laravel adalah system routing yang memudahkan penanganan permintaan HTTP. Ini memungkinkan pengembang untuk menentukan dengan jelas bagaimana permintaan URL akan diarahkan ke bagian yang sesuai dalam aplikasi.

Laravel juga memiliki system ORM yang kuat, yaitu Eloquent, yang memfasilitasi interaksi dengan basis data menggunakan objek-objek PHP. Ini menghilangkan kebutuhan

untuk menulis query SQL secara manual. Laravel menjadi pilihan utama bagi pengembang web yang ingin membangun aplikasi web yang kuat dan efisien.

BAB III

DOKUMENTASI PRAKTIKUM

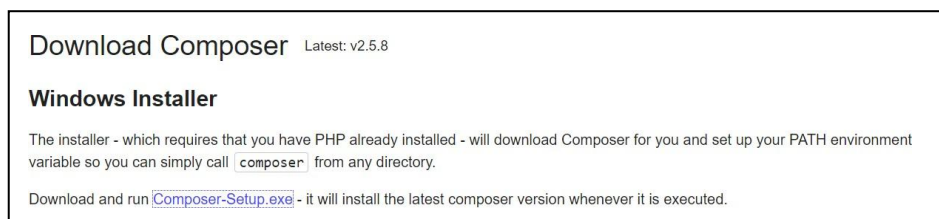
3.1. Instalasi Composer

Untuk menginstall composer kita bisa mengunjungi halaman website composer dengan mengunjungi link <https://getcomposer.org/>, nantinya setelah diakses, akan muncul tampilan seperti gambar di bawah.



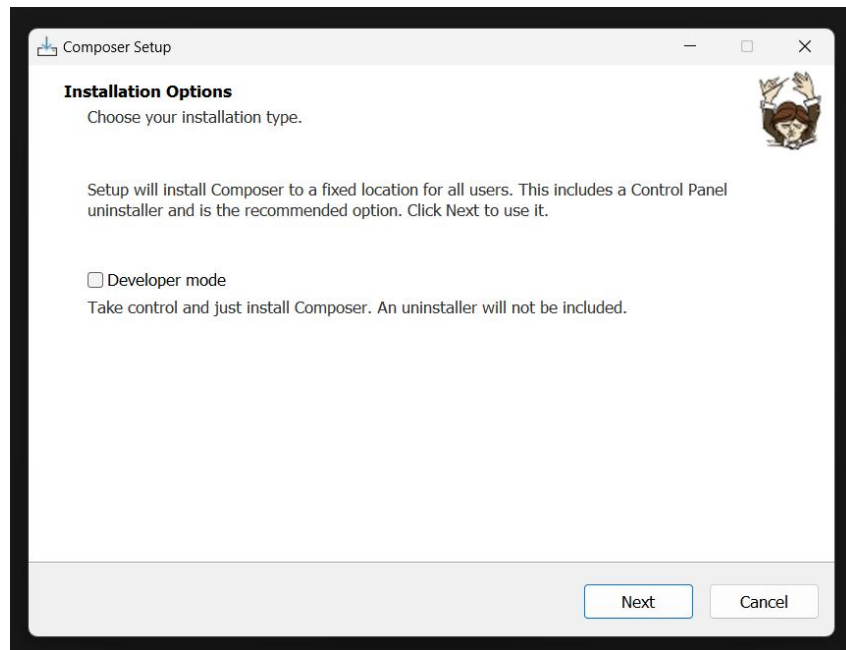
Gambar 1 Halaman utama web composer

Setelah klik “Download” pada halaman utama, selanjutnya kita mendownload file yang bernama “Composer-Setup.exe”.



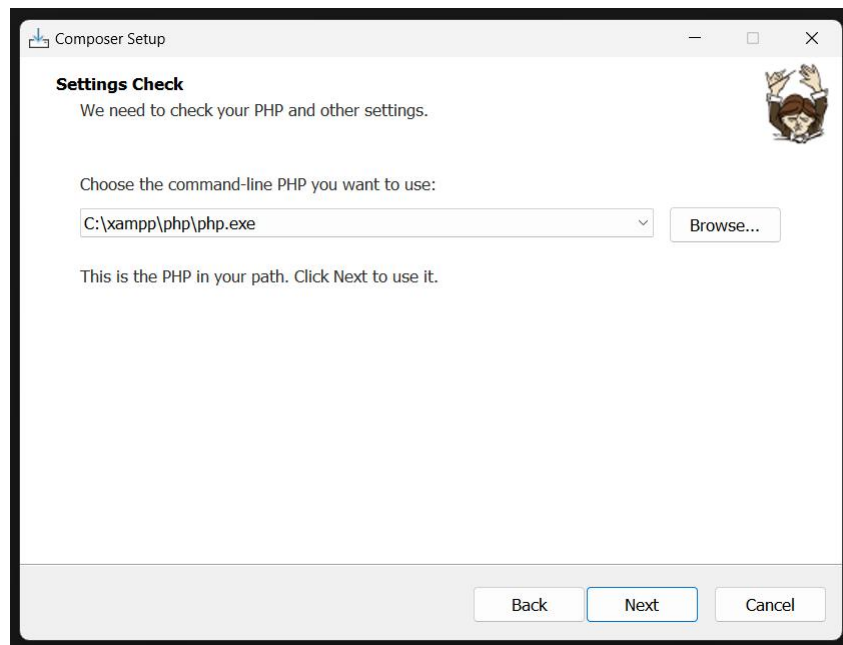
Gambar 2 Composer installer

Kemudian klik “Next”.



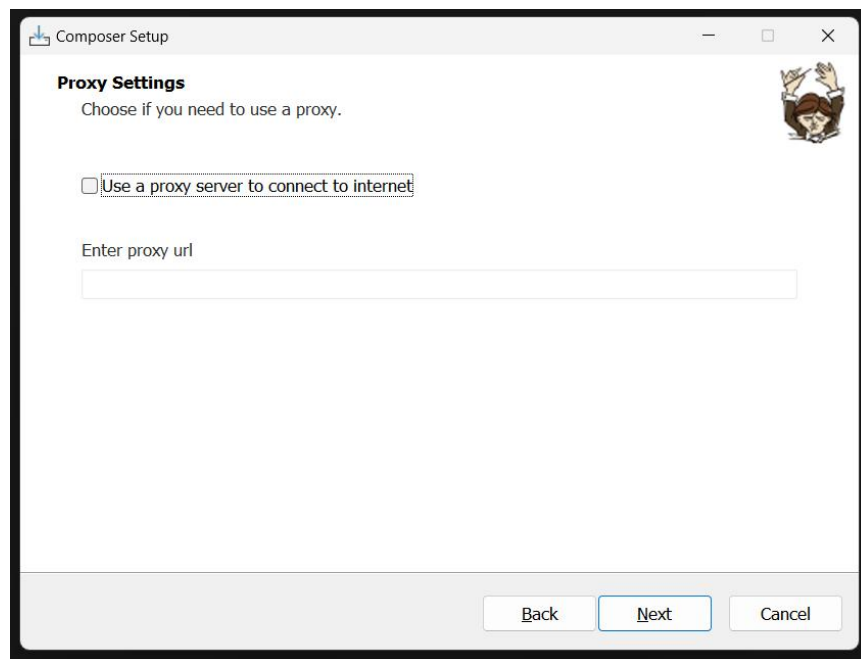
Gambar 3 Opsi Instalasi

Kemudian kita akan memasang instalasi composer di path php, pilih folder yang akan diinstal kemudian klik “Next”.



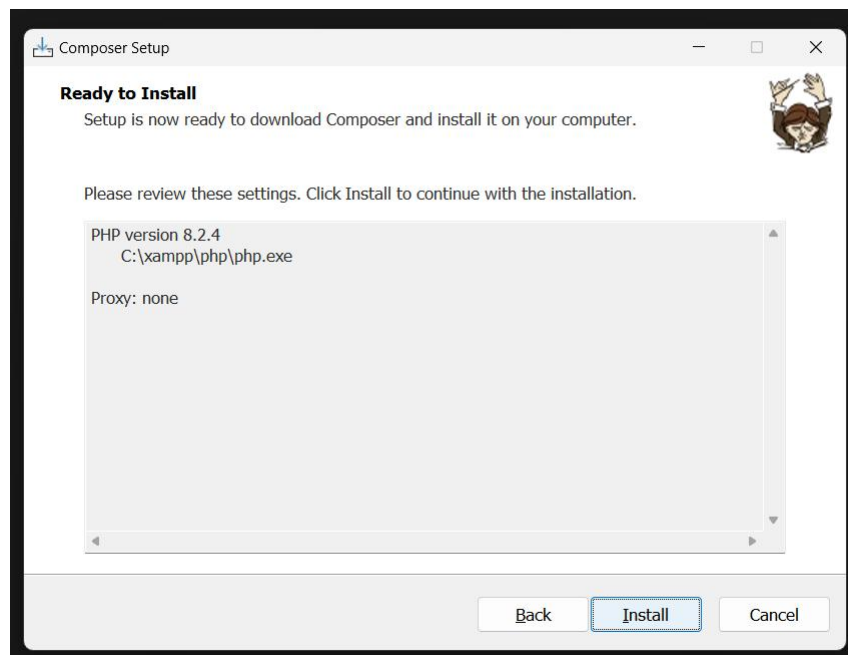
Gambar 4 Mengatur tempat instalasi composer

Kemudian klik “Next”.



Gambar 5 Opsi menggunakan proxy

Kemudian klik tombol “install”, lalu tunggu hingga proses instalasi selesai.



Gambar 6 Instalasi composer

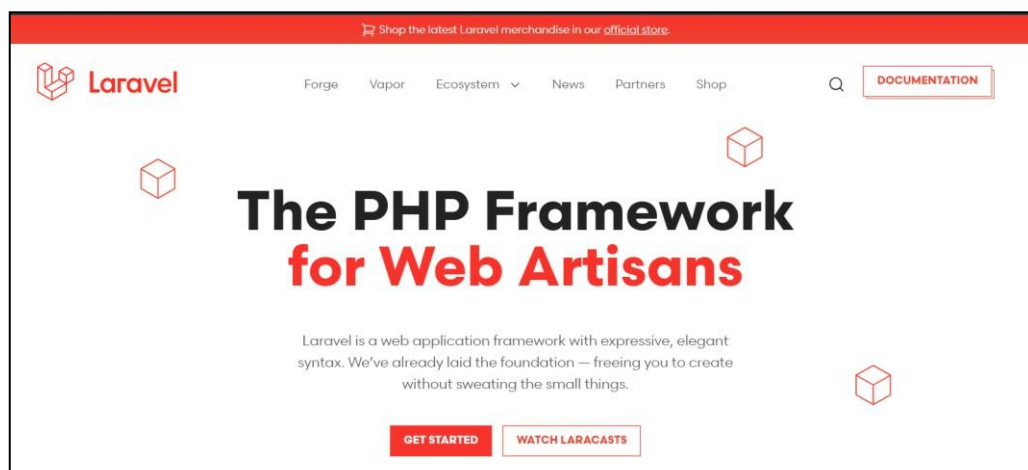
Composer telah selesai terinstall pada perangkat, selanjutnya kita bisa klik “finish”.



Gambar 7 Composer telah selesai terinstall

3.2. Penggunaan Laravel

Untuk menggunakan Laravel, kita dapat mengakses web dokumentasi Laravel dimana dapat kita akses melalui <https://laravel.com>. Di bawah ini merupakan tampilan awal dari web dokumentasi Laravel. Selanjutnya kita dapat mengklik tombol yang bertuliskan “documentation” pada pojok kanan atas.



Gambar 8 Tampilan depan web dokumentasi Laravel

Langkah pertama yang dapat kita lakukan adalah dengan masuk ke file directory dari proyek yang akan kita kerjakan. Kita dapat menggunakan command prompt atau dapat melalui terminal pada visual studio code dengan perintah “cd <tempat folder>”, seperti pada gambar di bawah, kita akan masuk ke dalam folder “C:\xampp\htdocs”.

```
PS C:\Users\LENOVO> cd C:\xampp\htdocs
PS C:\xampp\htdocs> █
```

Gambar 9 Masuk ke folder tempat project akan digunakan

Setelah kita masuk ke folder yang akan kita buat untuk membuat proyek, selanjutnya kita akan menjalankan perintah berikut ini.

```
composer create-project laravel/laravel example-app
```

Gambar 10 Perintah membuat proyek menggunakan laravel

```
PS C:\Users\LENOVO> cd C:\xampp\htdocs
PS C:\xampp\htdocs> composer create-project laravel/laravel example-app
```

Gambar 11 Menjalankan perintah untuk membuat proyek

Perintah “example-app” di atas dapat kita ganti sesuai dengan nama proyek yang kita kerjakan. Pada contoh ini kita menggunakan “example-app”, setelah itu kita tunggu hingga proses selesai.

```
62 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating optimized autoload files
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi

 INFO  Discovering packages.

laravel/sail ..... DONE
laravel/sanctum ..... DONE
laravel/tinker ..... DONE
nesbot/carbon ..... DONE
nunomaduro/collision ..... DONE
nunomaduro/termwind ..... DONE
spatie/laravel-ignition ..... DONE

82 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan vendor:publish --tag=laravel-assets --ansi --force

 INFO  No publishable resources for tag [laravel-assets].

No security vulnerability advisories found
> @php artisan key:generate --ansi

 INFO  Application key set successfully.
```

Gambar 12 laravel berhasil terinstall

Setelah Laravel telah selesai terpasang seperti gambar di atas, kemudian kita masuk ke folder proyek kita dengan perintah “cd <nama proyek>”. Kita dapat menggunakan perintah “cd example-app” seperti gambar di bawah.

```
PS C:\xampp\htdocs> cd example-app
PS C:\xampp\htdocs\example-app>
```

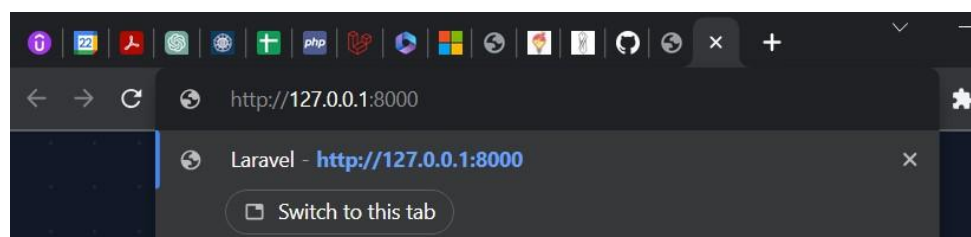
Gambar 13 perintah untuk masuk ke folder proyek

Setelah masuk ke folder proyek, kita dapat menjalankan perintah “php artisan serve” untuk menjalankan proyek Laravel kita, dimana nantinya kita akan diberikan “url host” untuk menjalankan proyek kita seperti pada gambar di bawah ini.

```
PS C:\xampp\htdocs\example-app> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

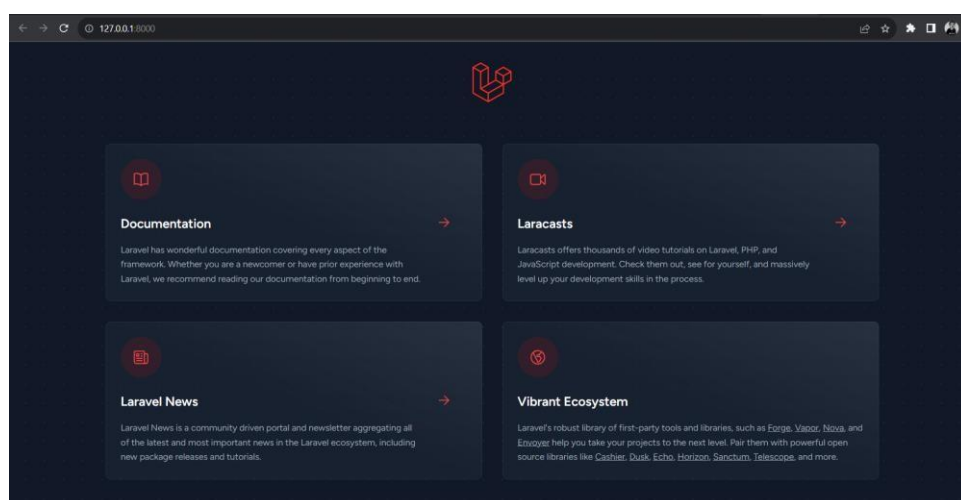
Gambar 14 Perintah menjalankan project laravel

Setelah kita menjalankan perintah di atas, selanjutnya kita akan menjalankan url pada web browser kita seperti pada gambar di bawah ini.



Gambar 15 Mengakses project laravel

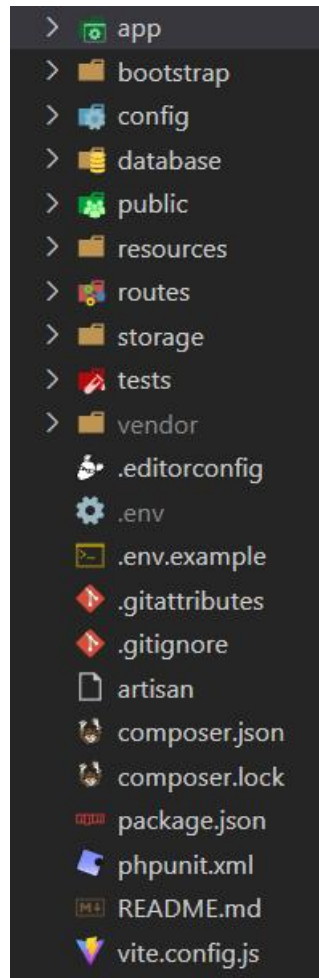
Berikut merupakan tampilan awal dari proyek Laravel yang telah kita inisiasi.



Gambar 16 Tampilan awal proyek Laravel

3.3. Penjelasan Struktur Direktori pada Laravel

Struktur folder Laravel adalah salah satu hal yang penting untuk kita pahami dikarenakan selama pengembangan proyek, struktur folder ini akan mempengaruhi kita dalam membuat sebuah proyek berbasis Laravel. Berikut ini merupakan gambar dari struktur folder Laravel itu sendiri.



Gambar 17 Struktur folder pada laravel

Folder-folder Laravel diatas memiliki kegunaan sebagai berikut:

- App :Direktori aplikasi ini berisi kode inti dari proyek kita.
- Bootstrap :Direktori bootstrap berisi berkas app.php yang mem-bootstrap kerangka kerja. Direktori ini juga berisi direktori cache yang berisi berkas-berkas yang dihasilkan oleh kerangka kerja untuk optimasi kinerja, seperti berkas cache rute dan layanan.
- Config : Direktori konfigurasi, sesuai namanya, berisi semua berkas konfigurasi aplikasi Anda.

- Database : Direktori database ini berisi migrasi basis data, pabrik model, dan data awal (seeds).
- Public : Direktori publik berisi berkas index.php, yang merupakan titik masuk untuk semua permintaan yang masuk ke aplikasi Anda dan mengonfigurasi autoload. Direktori ini juga berisi aset-aset seperti gambar, JavaScript, dan CSS Anda.
- Resource : Direktori resource berisi tampilan Anda serta aset mentah yang belum dikompilasi, seperti CSS atau JavaScript.
- Routes : Direktori routes berisi semua definisi routes untuk aplikasi Anda. Secara default, beberapa berkas routes disertakan dengan Laravel: web.php, api.php, console.php, dan channels.php.
- Storage : Direktori storage berisi log Anda, templat Blade yang dikompilasi, sesi berbasis file, cache berbasis file, dan berkas-berkas lain yang dihasilkan oleh kerangka kerja. Direktori ini terbagi menjadi direktori app, framework, dan logs. Direktori app dapat digunakan untuk menyimpan berkas-berkas yang dihasilkan oleh aplikasi Anda. Direktori framework digunakan untuk menyimpan berkas-berkas yang dihasilkan oleh kerangka kerja dan cache. Terakhir, direktori logs berisi berkas log aplikasi Anda.
- Tests : Direktori tests berisi pengujian otomatis Anda. Contoh pengujian unit PHPUnit dan pengujian fitur disediakan secara bawaan. Setiap kelas pengujian harus diakhiri dengan kata "Test". Anda dapat menjalankan pengujian Anda menggunakan perintah phpunit atau php vendor/bin/phpunit. Atau, jika Anda ingin hasil pengujian yang lebih rinci dan menarik, Anda dapat menjalankan pengujian Anda dengan perintah Artisan php artisan test.
- Vendor : Direktori vendor berisi dependensi Composer Anda.

3.4. Contoh Proyek Laravel

Contoh dari proyek Laravel dapat dilihat pada repositori github yang telah saya buat, berikut ini merupakan link github tersebut:

<https://github.com/wildandr/ppw2>

Dimana pada repositori tersebut, saya telah membuat controller baru menggunakan perintah “php artisan make:controller test”. Controller tersebut digunakan untuk menghubungkan route ke view, sehingga ketika route dikunjungi, route akan memanggil fungsi pada controller, dan controller akan memanggil view yang dipanggil pada fungsi yang digunakan.

BAB IV

KESIMPULAN

4.1. Kesimpulan

Dalam praktikum pemrograman web, berbagai tantangan kompleks kerap muncul dalam pengembangan aplikasi web, terutama saat proyek semakin berkembang dan kompleks. Pengelolaan dan pengembangan kode sumber menjadi tugas yang menantang dan memakan waktu, terutama dalam menangani tugas-tugas rutin seperti mengatur rute, berinteraksi dengan basis data, dan mengelola sesi.

Laravel, sebagai salah satu framework PHP yang sangat populer, hadir sebagai solusi kuat untuk mengatasi permasalahan ini. Dengan dirancang dengan baik, framework ini membantu pengembangan aplikasi web dengan cepat dan efisien, berkat fitur-fitur seperti sistem routing yang kuat, ORM (*Object-Relational Mapping*) yang canggih, serta berbagai alat bantu lainnya. Harapannya, praktikum ini akan memberikan pemahaman yang kuat tentang Laravel serta pentingnya efisiensi dalam mengatasi permasalahan pengembangan web.

DAFTAR PUSTAKA

<https://laravel.com/docs/10.x/structure#the-root-app-directory>

<https://getcomposer.org/>

<https://github.com/wildandr/ppw2>

