

PENDAHULUAN DAN INSTALASI LARAVEL

Laravel adalah sebuah kerangka kerja aplikasi web yang telah menjadi salah satu favorit di kalangan pengembang, terutama karena sintaksisnya yang ekspresif dan elegan. Dengan kerangka kerja ini, pengembang dapat membuat aplikasi dengan struktur yang jelas dan titik awal yang sudah ditetapkan, yang memungkinkan mereka untuk fokus pada aspek kreatif dari pengembangan perangkat lunak. Sementara itu, Laravel menangani berbagai detail teknis yang rumit, sehingga pengembang dapat berkonsentrasi pada apa yang benar-benar penting, menciptakan aplikasi yang menakjubkan dan berfungsi dengan baik. Laravel tidak hanya menawarkan kemudahan dalam menulis kode, tetapi juga menyediakan berbagai fitur canggih yang membuat pengembangan aplikasi menjadi lebih efisien dan dapat diandalkan. Dalam ekosistem Laravel, pengembang memiliki akses ke alat-alat seperti injeksi dependensi, lapisan abstraksi basis data, antrean, pekerjaan terjadwal, pengujian unit, dan pengujian integrasi, yang semuanya dirancang untuk mempercepat proses pengembangan sambil memastikan bahwa aplikasi yang dihasilkan berkualitas tinggi.

Sejarah Singkat Laravel

Sebelum kita menyelami lebih dalam mengenai fitur-fitur canggih dan bagaimana Laravel bisa tumbuh bersama pengembang, penting untuk memahami latar belakang dan sejarah dari kerangka kerja ini. Laravel diciptakan oleh Taylor Otwell pada tahun 2011. Saat itu, Otwell merasa bahwa kerangka kerja PHP yang ada tidak menyediakan solusi yang memadai untuk kebutuhan pengembangan web modern. Framework seperti CodeIgniter, meskipun populer, memiliki keterbatasan dalam hal fitur-fitur seperti otentikasi dan dukungan untuk Object-Relational Mapping (ORM). Maka dari itu, Otwell memutuskan untuk membuat kerangka kerja baru yang akan memenuhi kebutuhan pengembang modern. Dengan fokus pada kemudahan penggunaan, sintaksis yang bersih, dan fitur-fitur canggih, Laravel dirancang untuk menjadi alat yang tidak hanya memudahkan pekerjaan pengembang tetapi juga meningkatkan kualitas dan skalabilitas aplikasi yang dihasilkan. Sejak peluncuran pertamanya, Laravel telah berkembang pesat dan menjadi pilihan utama bagi banyak pengembang PHP di seluruh dunia.

Struktur dan Arsitektur Laravel

Laravel didasarkan pada arsitektur Model-View-Controller (MVC), yang merupakan pola desain yang memisahkan logika aplikasi menjadi tiga komponen utama: Model, View, dan Controller. Struktur ini tidak hanya memudahkan dalam pengembangan tetapi juga

dalam pemeliharaan aplikasi, karena setiap komponen memiliki tanggung jawab yang jelas dan terpisah.

- a. Model, Model bertanggung jawab untuk berinteraksi dengan database. Dalam Laravel, interaksi ini difasilitasi oleh Eloquent ORM, yang merupakan lapisan abstraksi database yang kuat dan intuitif. Dengan Eloquent, pengembang dapat melakukan operasi database dengan menggunakan objek PHP, sehingga mengurangi kerumitan menulis kueri SQL secara langsung.
- b. View, View adalah bagian dari aplikasi yang bertanggung jawab untuk menampilkan data kepada pengguna. Laravel menggunakan Blade sebagai mesin templatnya, yang memungkinkan pengembang untuk membuat tampilan yang dinamis dengan sintaksis yang sederhana namun fleksibel. Blade mendukung penggunaan layout, inheritance, dan komponen, yang semuanya dirancang untuk memaksimalkan efisiensi dalam penulisan kode tampilan.
- c. Controller adalah komponen yang menghubungkan Model dan View. Ia bertanggung jawab untuk menerima input dari pengguna, memprosesnya (biasanya dengan bantuan Model), dan kemudian mengembalikan hasilnya kepada View untuk ditampilkan. Dengan mengikuti pola MVC, Laravel memastikan bahwa aplikasi tetap terorganisir dengan baik, bahkan saat kompleksitasnya meningkat.

Fitur-Fitur Utama Laravel

Laravel bukan hanya tentang struktur dan arsitektur; kerangka kerja ini juga dilengkapi dengan berbagai fitur yang dirancang untuk membuat pengembangan lebih mudah, cepat, dan efisien. Berikut ini adalah beberapa fitur utama yang membuat Laravel menonjol di antara kerangka kerja PHP lainnya:

- a. Eloquent ORM, Eloquent adalah salah satu fitur yang paling dihargai di Laravel. ORM ini memungkinkan pengembang untuk berinteraksi dengan database menggunakan model PHP daripada menulis kueri SQL mentah. Dengan Eloquent, setiap tabel dalam database memiliki model yang sesuai, dan setiap baris dalam tabel direpresentasikan sebagai objek di PHP. Ini membuat operasi database lebih mudah dimengerti dan dikelola.
- b. Blade Templating Engine, Blade adalah mesin templating Laravel yang kuat dan mudah digunakan. Blade memungkinkan pengembang untuk menulis tampilan dengan sintaksis yang rapi dan bersih. Salah satu keunggulan Blade adalah kemampuannya untuk membuat layout yang dapat diwariskan oleh halaman lain,

yang menghemat waktu dan usaha dalam mengembangkan antarmuka pengguna yang konsisten.

- c. Injeksi Dependensi dan Kontainer IoC, Laravel memiliki sistem injeksi dependensi yang canggih, yang memungkinkan pengembang untuk dengan mudah mengelola dependensi di aplikasi mereka. Dengan kontainer IoC (Inversion of Control) Laravel, pengembang dapat mendefinisikan dependensi mereka di satu tempat dan membiarkan Laravel mengurus sisanya. Ini tidak hanya membuat kode lebih modular dan mudah diuji, tetapi juga meningkatkan fleksibilitas dan skalabilitas aplikasi.
- d. Middleware adalah lapisan yang berada di antara request dan respons dalam aplikasi Laravel. Middleware memungkinkan pengembang untuk memeriksa dan memodifikasi request sebelum mereka mencapai controller. Fitur ini sangat berguna untuk menangani otentikasi, logging, dan penanganan akses lainnya.
- e. Artisan adalah alat baris perintah Laravel yang kuat yang menyediakan berbagai perintah untuk membantu dalam pengembangan. Dengan Artisan, pengembang dapat dengan mudah membuat model, controller, dan migrasi, serta menjalankan tugas-tugas rutin seperti clearing cache, menjalankan pengujian, dan banyak lagi. Artisan adalah salah satu fitur yang benar-benar mempercepat alur kerja pengembang.
- f. Laravel menyediakan sistem migrasi yang kuat untuk mengelola skema database Anda. Dengan migrasi, Anda dapat melacak perubahan skema dari waktu ke waktu dan dengan mudah membagikan perubahan tersebut kepada tim Anda. Seeding database adalah fitur yang memungkinkan Anda untuk mengisi database dengan data uji atau data awal secara otomatis.
- g. Laravel mendukung penggunaan antrean (queues) untuk menunda tugas-tugas yang memerlukan waktu lama, seperti pengiriman email atau pemrosesan gambar, ke dalam pekerjaan (jobs) yang akan dieksekusi nanti. Dengan antrean, aplikasi Anda tetap responsif terhadap pengguna, sementara tugas-tugas berat dijalankan di latar belakang.
- h. Laravel memudahkan penjadwalan tugas-tugas yang perlu dijalankan secara rutin, seperti backup database atau pengiriman email pengingat. Dengan sintaksis yang mirip dengan CRON, Anda dapat mendefinisikan tugas-tugas terjadwal di dalam aplikasi Anda tanpa perlu bergantung pada sistem operasi.

Keamanan di Laravel

Keamanan adalah salah satu aspek paling kritis dalam pengembangan aplikasi web, dan Laravel memberikan perhatian besar terhadap hal ini. Laravel dilengkapi dengan berbagai fitur keamanan bawaan yang dirancang untuk melindungi aplikasi dari berbagai jenis serangan, termasuk serangan injeksi SQL, Cross-Site Scripting (XSS), dan

Cross-Site Request Forgery (CSRF). Laravel menggunakan prepared statements untuk semua query database, yang berarti bahwa input dari pengguna selalu di-escape secara otomatis sebelum dimasukkan ke dalam query. Ini mengurangi risiko injeksi SQL secara signifikan, yang merupakan salah satu jenis serangan yang paling umum di aplikasi web. Selain itu, Laravel menyediakan middleware CSRF yang secara otomatis melindungi aplikasi dari serangan CSRF. Middleware ini menghasilkan token CSRF yang unik untuk setiap sesi pengguna dan memverifikasi bahwa setiap request yang masuk memiliki token yang valid sebelum diproses. Ini memastikan bahwa aplikasi Anda terlindungi dari serangan CSRF, yang mencoba untuk memaksa pengguna melakukan tindakan tanpa sepengetahuan mereka.

Laravel juga dilengkapi dengan hashing password bawaan menggunakan algoritma bcrypt. Bcrypt adalah salah satu algoritma hashing yang paling aman saat ini, dan Laravel menggunakannya secara default untuk menyimpan kata sandi pengguna. Ini memastikan bahwa meskipun terjadi kebocoran data, kata sandi

Pengujian dan Keandalan

Laravel memudahkan pengembang untuk menulis pengujian dan memastikan bahwa aplikasi mereka berfungsi dengan baik. Laravel dilengkapi dengan berbagai alat untuk membantu dalam pengujian unit dan pengujian integrasi, serta menyediakan lingkungan pengujian yang mudah dikonfigurasi. Pengujian unit memungkinkan pengembang untuk menguji bagian-bagian kecil dari aplikasi mereka secara terpisah, memastikan bahwa setiap komponen berfungsi dengan benar. Laravel menyediakan banyak utilitas untuk memudahkan pengujian unit, termasuk dukungan untuk pengujian database dan pembuatan mock untuk objek dan dependensi.

Selain itu, Laravel juga mendukung pengujian integrasi, yang memungkinkan pengembang untuk menguji bagaimana berbagai bagian dari aplikasi berinteraksi satu sama lain. Dengan pengujian integrasi, pengembang dapat memastikan bahwa semua komponen bekerja bersama-sama dengan benar dan bahwa aplikasi secara keseluruhan berfungsi sebagaimana mestinya.

Skalabilitas dan Kinerja

Laravel tidak hanya dirancang untuk kemudahan penggunaan, tetapi juga untuk skalabilitas dan kinerja. Salah satu keunggulan Laravel adalah kemampuannya untuk mendukung aplikasi dalam skala besar, yang mungkin memerlukan penanganan ribuan atau bahkan jutaan pengguna. Untuk mendukung aplikasi yang sangat skalabel, Laravel

menyediakan berbagai alat dan fitur yang memungkinkan pengembang untuk mengoptimalkan kinerja aplikasi mereka. Misalnya, Laravel mendukung caching yang memungkinkan aplikasi untuk menyimpan hasil query database atau output HTML dalam cache, sehingga dapat diakses lebih cepat di masa depan. Selain itu, Laravel juga mendukung penggunaan queues untuk memproses tugas-tugas yang memerlukan waktu lama di latar belakang, yang membantu menjaga responsivitas aplikasi meskipun beban kerja meningkat. Dengan dukungan untuk berbagai driver queue seperti Redis dan Beanstalkd, Laravel memungkinkan pengembang untuk memilih solusi yang paling sesuai dengan kebutuhan aplikasi mereka.

Ekosistem Laravel dan Komunitas

Salah satu alasan utama mengapa Laravel begitu populer adalah ekosistemnya yang luas dan komunitasnya yang aktif. Laravel tidak hanya menyediakan kerangka kerja inti yang kuat, tetapi juga ekosistem yang terdiri dari berbagai alat dan paket yang dapat digunakan untuk memperluas fungsionalitas aplikasi. Beberapa alat yang paling populer di ekosistem Laravel termasuk Laravel Forge, yang memudahkan pengelolaan dan deployment server, serta Laravel Vapor, yang memungkinkan pengembang untuk menjalankan aplikasi Laravel di infrastruktur serverless. Selain itu, Laravel juga memiliki banyak paket pihak ketiga yang dikembangkan oleh komunitas, yang mencakup segala hal mulai dari otentikasi hingga manajemen konten.

Komunitas Laravel adalah salah satu yang paling aktif dan suportif di dunia pengembangan web. Dengan ribuan pengembang yang berkontribusi pada proyek open-source, menulis tutorial, dan berbagi pengetahuan di forum dan konferensi, Laravel adalah tempat yang tepat bagi siapa saja yang ingin belajar dan berkembang sebagai pengembang web.

Software pendukung Laravel

Sebelum melakukan instalasi Laravel, ada beberapa hal penting yang perlu Anda persiapkan dan pahami agar proses instalasi berjalan dengan lancar dan tanpa hambatan. Laravel memiliki beberapa persyaratan sistem dan perangkat lunak yang harus dipenuhi. Berikut ini adalah beberapa *software* pendukung yang perlu diinstal

a. Menginstal Web Server

Laravel membutuhkan server web untuk menjalankan aplikasi. Dua opsi adalah:

- Apache adalah server web yang banyak digunakan dan bekerja sangat baik dengan PHP. Jika Anda menggunakan Apache, pastikan modul

mod_rewrite diaktifkan untuk mendukung fitur routing Laravel. Apache dapat didownload di link berikut <https://httpd.apache.org/download.cgi>

- Nginx juga merupakan server web yang populer, dikenal karena performa dan efisiensinya. Jika Anda menggunakan Nginx, Anda perlu mengkonfigurasi virtual host untuk proyek Laravel Anda. Nginx dapat didownload di link <http://nginx.org/en/download.html>

b. Instalasi PHP

Laravel dibangun di atas PHP, sehingga menginstal versi PHP yang sesuai adalah langkah krusial. Laravel merekomendasikan PHP versi 8.0 atau yang lebih baru. Pastikan PHP sudah terinstal di sistem Anda, dan Anda dapat memeriksa versi PHP yang terinstal dengan perintah berikut di terminal:

```
php -v
```

Selain itu, pastikan ekstensi PHP berikut ini juga sudah diaktifkan seperti OpenSSL, PDO, Mbstring, Tokenizer, XML, ctype, JSON

c. Instalasi composer

Composer adalah manajer dependensi untuk PHP yang diperlukan untuk menginstal Laravel. Composer memungkinkan Anda untuk mengelola paket-paket yang digunakan dalam proyek PHP Anda dengan mudah. Untuk menginstal Composer, ikuti langkah-langkah di situs resminya (<https://getcomposer.org/download/>). Setelah Composer diinstal, Anda dapat memeriksa instalasinya dengan perintah:

```
composer --version
```

d. Instalasi Basisdata

Laravel mendukung berbagai jenis database seperti MySQL, PostgreSQL, SQLite, dan SQL Server. Sebelum menginstal Laravel, Anda perlu menyiapkan database yang akan digunakan oleh aplikasi Anda. Biasanya, pengembang menggunakan MySQL atau PostgreSQL karena keduanya sangat populer dan didukung dengan baik oleh Laravel. Pastikan Anda sudah menginstal dan mengkonfigurasi server database di sistem Anda.

Jika Anda menggunakan MySQL, Anda bisa mengelola database dengan alat seperti phpMyAdmin atau MySQL Workbench. Pastikan Anda mencatat detail

koneksi seperti nama database, nama pengguna, dan kata sandi, karena Anda akan membutuhkannya nanti saat mengonfigurasi Laravel.

e. Instalasi integrated development environment (IDE)

Terdapat berbagai pilihan untuk IDE yang dapat digunakan, salah satunya seperti Visual Studio Code atau sering disebut dengan VsCode, vscode dapat diunduh dengan gratis di <https://code.visualstudio.com/download> vscode memiliki ekstensi yang cukup banyak untuk mendukung pemrograman web dengan laravel menjadi lebih mudah.

Instalasi Laravel

Instalasi Laravel dari online repository, yaitu menggunakan Composer untuk mengunduh Laravel langsung dari repositori resminya, adalah proses yang cukup mudah dan cepat jika Anda mengikuti langkah-langkah berikut ini. Pastikan Anda sudah melakukan persiapan seperti yang disebutkan sebelumnya, seperti menginstal Composer, menyiapkan lingkungan pengembangan, dan memenuhi persyaratan sistem. Anda juga dapat menginstal Laravel langsung dengan perintah create-project dari Composer seperti berikut

```
composer create-project --prefer-dist laravel/laravel nama-proyek "10.*"
```

Contohnya :

```
composer create-project --prefer-dist laravel/laravel sistem-toko "10.*"
```

```
Installing dependencies from lock file (including require-dev)
Package operations: 112 installs, 0 updates, 0 removals
- Downloading doctrine/inflector (2.0.10)
- Downloading doctrine/lexer (3.0.1)
- Downloading symfony/polyfill-ctype (v1.30.0)
- Downloading webmozart/assert (1.11.0)
- Downloading dragonmantank/cron-expression (v3.3.3)
- Downloading symfony/deprecation-contracts (v3.5.0)
- Downloading fakerphp/faker (v1.23.1)
```

Gambar 1. Proses instalasi dan download package yang dibutuhkan

Menjalankan proyek laravel

Laravel sebenarnya tidak harus ditempatkan didalam direktori web server yang kita install sebelumnya, karena laravel memiliki build-in web server sendiri, Pertama-tama, buka terminal atau command prompt, dan navigasi ke direktori proyek Laravel yang sudah dibuat.

```
cd /path/ke/proyek/sistem-toko
```

Misal

```
cd C:\belajar program\sistem-toko
```

Setelah berada di dalam direktori proyek dan memastikan semua dependensi terinstal, jalankan perintah berikut untuk memulai server development Laravel

```
php artisan serve
```

Jika perintah php artisan serve berhasil dijalankan, Anda akan melihat output seperti berikut di terminal

```
Laravel development server started: <http://127.0.0.1:8000>
```

Ini berarti server development Laravel telah berjalan dan Anda bisa mengakses aplikasi Laravel Anda di browser melalui alamat <http://127.0.0.1:8000> atau <http://localhost:8000>.

Buka browser dan kunjungi salah satu alamat tersebut untuk melihat aplikasi Laravel Anda berfungsi.

Jika anda akan mengakses laravel proyek tidak melalui build-in web server laravel, misal menggunakan web apache atau nginx maka cara mengaksesnya adalah harus dipastikan proyek laravel kita berada didalam direktori web server, misalkan

```
C:\xampp\htdocs\sistem-toko
```

Maka cara mengakses dibrowsernya adalah <http://localhost/public>

Struktur Folder Laravel

Dalam setiap proyek pengembangan perangkat lunak, struktur yang jelas dan konsisten adalah fondasi yang kuat untuk memastikan proyek tersebut dapat dikelola dengan baik, terutama ketika proyek tumbuh menjadi lebih besar dan kompleks. Laravel, sebagai framework PHP yang modern dan populer, menawarkan struktur direktori yang dirancang untuk memisahkan logika aplikasi dari komponen-komponen lain seperti tampilan dan konfigurasi. Pemisahan ini memungkinkan pengembang untuk bekerja pada bagian-bagian spesifik dari aplikasi tanpa harus khawatir mengganggu bagian lain, yang pada gilirannya meningkatkan produktivitas dan mengurangi risiko kesalahan.

Salah satu tujuan utama dari struktur direktori Laravel adalah untuk mengikuti prinsip pemrograman yang baik seperti Single Responsibility Principle (SRP), di mana setiap komponen atau bagian dari aplikasi hanya bertanggung jawab atas satu hal. Dengan kata lain, direktori `app/` seharusnya hanya berisi logika bisnis, sementara tampilan diletakkan di dalam `resources/views/`, dan konfigurasi ditempatkan di dalam `config/`. Hal ini memudahkan pengelolaan kode, terutama ketika aplikasi berkembang menjadi lebih besar dengan lebih banyak fitur.

Struktur direktori pada Laravel dirancang dengan baik untuk memisahkan logika, tampilan, konfigurasi, dan berbagai aspek lain dari aplikasi. Memahami struktur ini sangat penting bagi pengembang untuk menavigasi dan mengelola proyek Laravel dengan lebih efisien. Berikut adalah penjelasan mengenai struktur direktori utama yang ada di dalam proyek Laravel:

1. Direktori `app/`

Direktori `app/` adalah jantung dari aplikasi Laravel Anda. Di sinilah logika aplikasi Anda ditempatkan, mencakup berbagai file yang berhubungan dengan model, controller, middleware, dan layanan. Memahami struktur di dalam direktori ini adalah kunci untuk memahami bagaimana aplikasi Laravel Anda bekerja.

- **`Console/`:** Direktori ini berisi semua command Artisan yang didefinisikan oleh pengembang. Command ini adalah skrip yang dapat dijalankan melalui terminal untuk melakukan tugas-tugas khusus seperti memigrasi database, membersihkan cache, dan banyak lagi.
- **`Exceptions/`:** Tempat di mana semua handler exception didefinisikan. Di sini, Anda akan menemukan file `Handler.php`, yang bertanggung jawab untuk menangani semua exception yang terjadi di aplikasi, memungkinkan pengembang untuk mengelola error dengan cara yang lebih terkendali.

- **Http/**: Ini adalah direktori utama untuk logika HTTP dan routing. Direktori ini berisi controller, middleware, dan form request.
 - **Controllers/**: Tempat untuk semua controller aplikasi. Controller bertindak sebagai penghubung antara model dan view, menangani logika aplikasi yang berhubungan dengan permintaan HTTP.
 - **Middleware/**: Berisi middleware, yang merupakan lapisan perantara yang memproses request sebelum mencapai controller. Middleware dapat digunakan untuk berbagai tujuan seperti otentikasi, logging, dan banyak lagi.
 - **Requests/**: Berisi form request, yang digunakan untuk memvalidasi dan mengotorisasi input yang masuk. Ini membantu memastikan bahwa data yang diterima oleh aplikasi adalah valid sebelum diproses lebih lanjut.
- **Providers/**: Berisi service provider yang digunakan untuk mengikat layanan ke container, melakukan bootstrapping, dan mengatur berbagai komponen aplikasi. Service provider adalah komponen penting dalam bootstrap aplikasi Laravel.
- **Models/**: Secara default, model Eloquent biasanya diletakkan di sini, meskipun pengembang dapat memilih untuk menempatkannya langsung di direktori **app/**. Model digunakan untuk berinteraksi dengan tabel database.

2. Direktori **bootstrap/**

Direktori **bootstrap/** mengandung file yang digunakan untuk bootstrap framework Laravel itu sendiri. Bootstrap adalah proses awal di mana framework Laravel memulai, memuat service provider, dan menginisialisasi aplikasi.

- **app.php**: File ini adalah file utama yang menginisialisasi aplikasi Laravel Anda. Di sinilah service provider di-boot dan berbagai komponen inti dari Laravel dimuat.
- **cache/**: Direktori ini berisi file cache yang digunakan selama bootstrap untuk meningkatkan kinerja aplikasi. File-file di sini membantu Laravel memulai lebih cepat dengan menggunakan data yang telah di-cache sebelumnya.

3. Direktori **config/**

Direktori **config/** berisi semua file konfigurasi untuk aplikasi Laravel. Setiap file konfigurasi di sini mengatur aspek yang berbeda dari aplikasi Anda, memungkinkan pengembang untuk menyesuaikan perilaku aplikasi dengan mudah.

- Contoh File Konfigurasi:

- **database.php**: Mengatur konfigurasi database, termasuk koneksi ke database, jenis database yang digunakan, dan pengaturan terkait lainnya.
- **mail.php**: Digunakan untuk mengkonfigurasi layanan email yang digunakan oleh aplikasi, termasuk driver, host, port, dan sebagainya.
- **app.php**: File ini mengatur konfigurasi aplikasi umum, seperti timezone, locale, dan nama aplikasi.

4. Direktori **database/**

Direktori **database/** berisi semua hal yang berhubungan dengan database dalam proyek Laravel Anda.

- Sub-direktori:
 - **factories/**: Tempat di mana Anda dapat mendefinisikan factory untuk model Anda. Factory digunakan untuk menghasilkan data tiruan selama pengujian atau seeding database.
 - **migrations/**: Berisi file migrasi, yang digunakan untuk mengelola skema database. Migrasi memungkinkan pengembang untuk melacak dan mengelola perubahan pada struktur database dari waktu ke waktu.
 - **seeders/**: Berisi file seeder yang digunakan untuk mengisi database dengan data awal atau data uji. Seeder berguna untuk mengisi database dengan data contoh yang dapat digunakan selama pengembangan dan pengujian.

5. Direktori **public/**

Direktori **public/** adalah root web dari aplikasi Laravel Anda. Ini adalah direktori yang dapat diakses secara publik, di mana semua file yang dapat diakses oleh pengguna akhir ditempatkan.

- **index.php**: File ini adalah entry point untuk semua request HTTP yang masuk. Semua permintaan web diarahkan melalui file ini, yang kemudian meneruskan permintaan tersebut ke framework Laravel untuk diproses.
- **Aset Publik**: File seperti gambar, JavaScript, CSS, dan file lainnya yang perlu diakses oleh pengguna web diletakkan di sini.

6. Direktori **resources/**

Direktori **resources/** berisi semua sumber daya yang tidak dapat dijalankan secara langsung oleh pengguna, tetapi diperlukan untuk membangun tampilan aplikasi.

- Sub-direktori:
 - **views/**: Berisi file Blade yang digunakan untuk tampilan (view) aplikasi. Blade adalah mesin template Laravel yang memungkinkan Anda menulis HTML yang dinamis dan terstruktur.
 - **lang/**: Berisi file terjemahan untuk berbagai bahasa yang didukung oleh aplikasi. Ini memungkinkan aplikasi Anda untuk mendukung banyak bahasa dengan mudah.
 - **css/**, **js/**, **sass/**: Berisi file frontend seperti CSS, JavaScript, dan Sass yang digunakan untuk mengembangkan tampilan aplikasi.

7. Direktori **routes/**

Direktori **routes/** berisi semua file rute (routes) untuk aplikasi Anda. Laravel menggunakan rute untuk menentukan bagaimana request HTTP tertentu harus diproses.

- **web.php**: Berisi rute untuk aplikasi web yang biasanya membutuhkan sesi dan cookie.
- **api.php**: Berisi rute untuk API yang biasanya tidak memerlukan sesi dan cookie, memungkinkan untuk komunikasi yang stateless.
- **console.php**: Berisi definisi rute untuk Artisan command, memungkinkan pembuatan command line khusus.
- **channels.php**: Berisi rute untuk saluran broadcasting, yang digunakan untuk fitur real-time seperti notifikasi dan chat.

8. Direktori **storage/**

Direktori **storage/** digunakan untuk menyimpan file yang dihasilkan oleh aplikasi, seperti log, cache, dan file yang diunggah.

- Sub-direktori:
 - **app/**: Berisi file yang dihasilkan oleh aplikasi atau diunggah oleh pengguna.
 - **framework/**: Berisi file yang dibutuhkan oleh framework Laravel, seperti cache session, cache view, dan sebagainya.
 - **logs/**: Berisi file log yang dihasilkan oleh aplikasi. Log ini digunakan untuk melacak aktivitas dan error yang terjadi dalam aplikasi.

9. Direktori `tests/`

Direktori `tests/` berisi semua file pengujian (tests) untuk aplikasi Anda. Laravel menggunakan PHPUnit sebagai framework pengujian default.

- Sub-direktori:
 - `Feature/`: Berisi pengujian fitur yang mencakup lebih dari satu komponen aplikasi. Pengujian ini menguji fungsionalitas yang lebih besar yang melibatkan beberapa bagian dari aplikasi.
 - `Unit/`: Berisi pengujian unit yang biasanya fokus pada pengujian bagian kecil atau individu dari aplikasi, seperti fungsi atau metode tertentu.

10. Direktori `vendor/`

Direktori `vendor/` berisi semua dependensi eksternal yang dikelola oleh Composer. Semua paket pihak ketiga yang diinstal melalui Composer ditempatkan di sini, menjadikan direktori ini penting untuk fungsi aplikasi Laravel.

11. File Lain yang Penting

Selain direktori utama, ada beberapa file penting lainnya dalam proyek Laravel yang perlu diketahui:

- `.env`: File ini berisi variabel lingkungan (environment variables) yang digunakan untuk mengkonfigurasi aplikasi secara lokal. Ini termasuk pengaturan seperti koneksi database, konfigurasi mail, dan pengaturan aplikasi lainnya.
- `artisan`: File ini digunakan untuk menjalankan command Artisan. Artisan adalah interface command-line untuk Laravel yang menyediakan banyak perintah berguna untuk pengembangan.
- `composer.json`: File ini berisi daftar dependensi PHP yang digunakan oleh proyek Laravel Anda, serta informasi konfigurasi Composer lainnya.
- `package.json`: Jika Anda menggunakan npm (Node Package Manager), file ini berisi daftar dependensi JavaScript untuk proyek Anda.
- `webpack.mix.js`: File konfigurasi untuk Laravel Mix, alat yang digunakan untuk mengkompilasi aset frontend seperti CSS dan JavaScript.