PEMROGRAMAN WEB LANJUTAN DENGAN LARAVEL

EVI LESTARI PRATIWI
ABDUL ROZAQ
RAMADHANI NOOR PRATAMA

Pemrograman Web Lanjutan dengan Laravel

Undang-Undang No. 28 Tahun 2014 Tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Perlindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap :

i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp 100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp 500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp 1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara

Pemrograman Web Lanjutan dengan Laravel

Evi Lestari Pratiwi
Abdul Rozaq
Ramadhani Noor Pratama

POLIBAN PRESS

Pemrograman Web Lanjutan dengan Laravel

Penulis :
Evi Lestari Pratiwi; Abdul Rozaq;
Ramadhani Noor Pratama

ISBN :
978-623-5259-06-2 (PDF)

Editor dan Penyunting :
Adi Pratomo

Desain Sampul dan Tata letak :
Rahma Indera; Eko Sabar Prihatin

Penerbit :
POLIBAN PRESS
Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia) no.004.098.1.06.2019
Cetakan Pertama, 2024

Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry, Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp : (0511)3305052
Email : press@poliban.ac.id

Diterbitkan pertama kali oleh :
Poliban Press, Banjarmasin, Januari 2024

Kata Pengantar

Selamat datang dalam buku "Pemrograman Web Lanjut Menggunakan Laravel". Buku ini dirancang untuk memberikan pemahaman mendalam tentang pengembangan web dengan menggunakan Laravel, salah satu framework PHP yang paling populer dan powerful.

Laravel memiliki struktur yang elegan dan menyediakan banyak fitur yang mempermudah proses pengembangan web, mulai dari konfigurasi dasar, manajemen database, hingga pembuatan sistem otentikasi. Buku ini ditujukan untuk para pengembang web yang ingin mengambil langkah lebih jauh dalam menguasai Laravel dan membangun aplikasi web yang robust dan scalable.

Buku ini dibagi menjadi sepuluh bab, yang secara progresif membimbing pembaca dari pengenalan dasar Laravel hingga konsep yang lebih lanjut seperti Object-Oriented Programming (OOP) dengan PHP, konfigurasi database, penggunaan route dan view, pembuatan dan penggunaan controller, sistem template Blade Laravel, operasi CRUD, implementasi otentikasi, manajemen file storage, dan latihan-latihan praktis untuk memperdalam pemahaman.

Buku ini cocok untuk pengembang web yang memiliki pemahaman dasar tentang PHP dan web development, dan ingin menguasai Laravel untuk mengembangkan aplikasi web yang lebih kompleks dan efisien. Dengan panduan langkah demi langkah, latihan praktis, dan contoh-contoh nyata, diharapkan pembaca dapat memahami konsep-konsep yang diajarkan dan mengaplikasikannya dalam pengembangan proyek nyata.

Banjarmasin, November 2023

Poliban Press

Prakata

Puji syukur ke hadirat Allah SWT, Tuhan Yang Maha Esa, karena berkat rahmat, taufik serta hidayah-Nya, penulis dapat menyelesaikan bahan ajar Pemrograman Web Lanjutan dengan Laravel. Bahan ajar yang dibuat, disusun untuk mata kuliah yang berkaitan dengan pemrograman web, pemrograman web lanjutan, yang dilengkapi dengan latihan soal agar mahasiswa dapat lebih memahami materi yang terkait.

Penulis mengucapkan terimakasih yang teramat dalam kepada pihak yang telah memberikan dukungan penuh terutama Jurusan Administrasi Bisnis, Program Studi Manajemen Infomatika, dan unit P3M Politeknik Negeri Banjarmasin, dan pihak yang tidak bisa diucapkan satu per satu, sehingga bahan ajar dapat tersusun dengan baik.

Dalam penulisan bahan ajar, tentu masih banyak kekurangan. Oleh karena itu, saran dan kritik yang membangun agar dapat menyempurnakan bahan ajar menjadi lebih baik lagi. Semoga bahan ajar ini dapat bermanfaat bagi kita semua.

Banjarmasin, November 2023

Penulis

Daftar Isi

BAB 1
PENGENALAN LARAVEL

Capaian Pembelajaran :

1. Mampu memahami pengertian framework
2. Mampu memahami perbedaan framework dengan PHP Native
3. Mampu memahami pengertian laravel
4. Mampu memahami manfaat dan fitur laravel

1.1. Pengertian Framework

Framework merupakan suatu susunan yang terdiri dari kode-kode umum yang digunakan untuk membangun sistem dan aplikasi. Fungsi framework adalah sebagai pola dasar atau contoh yang menyediakan fitur cerdas dan elemen struktur standar, yang bertujuan untuk mempermudah tugas para pengembang. Framework juga menunjukkan konsistensi tinggi karena telah diuji secara ekstensif dan terbukti berhasil.

Cara kerja dari framework memungkinkan pengembang untuk lebih fokus pada tujuan utama proyek tanpa khawatir tentang unsur dasar strukturnya. Dengan tidak perlu memulai dari awal, para pengembang dapat menghemat waktu dan sumber daya finansial serta mengurangi risiko kesalahan.

Proses pelaksanaannya didasarkan pada penggunaan ulang serangkaian kode umum yang dapat diterapkan sebagai kerangka dasar proyek. Desain ini perlu sesuai dengan bahasa dan karakteristik dari framework yang dimaksud.

Sebagian besar framework menyediakan forum dan dokumentasi yang luas, yang dapat membantu para pengembang dalam memperoleh pengetahuan baru dan mengatasi masalah yang mungkin muncul.

1.1.1 PHP Framework

Framework PHP adalah alat yang mengkondisikan proses pengembangan aplikasi web menggunakan PHP dengan menyajikan kerangka dasar untuk membangun situs web. Oleh karena itu, framework PHP dapat mempercepat pengembangan aplikasi atau situs web secara keseluruhan. Kerangka kerja ini juga memiliki kapabilitas dalam menciptakan aplikasi yang lebih kokoh dan stabil.

PHP Framework merupakan kerangka kerja yang dibangun menggunakan bahasa pemrograman PHP.

Kerangka kerja PHP diciptakan sebagai implementasi Model View Controller (MVC), yakni sebuah struktur yang memecah arsitektur pemrograman menjadi komponen-komponen yang lebih khusus, memungkinkan modifikasi pada setiap komponen tanpa mempengaruhi yang lain.

Dengan pendekatan MVC, perubahan pada fungsionalitas aplikasi tidak akan merugikan komponen antarmuka. Ini mungkin berkat pemisahan pemrograman menjadi tiga wilayah: Model, View, dan Controller.

a. Model berkaitan dengan data aplikasi.
b. View berkaitan dengan elemen visual antarmuka.
c. Controller berkaitan dengan fungsi atau logika aplikasi.

Ada beragam pilihan kerangka kerja PHP, seperti CodeIgniter, Symfony, Yii, Zend, dan tentunya Laravel.

1.1.2 PHP Framework vs PHP Native

Pemrograman web PHP native merupakan gabungan bahasa pemrograman yang berbasis pada PHP, yang dapat mencakup penggunaan teks Javascript, CSS, Bootstrap, dan sejenisnya. "Native" di sini mengacu pada pemrograman PHP yang asli, di mana kode PHP dirancang dan dikodekan secara murni oleh para pengembang sendiri, tanpa memerlukan penambahan setting atau konfigurasi lainnya. Keuntungan dari pendekatan PHP Native sederhana, karena setelah

menguasainya, akan lebih mudah untuk beralih ke penggunaan PHP Framework.

Kerangka kerja PHP memiliki perbedaan dengan pendekatan PHP native, di mana dalam PHP native seorang pengembang harus membuat kerangka kerjanya sendiri, yang pada akhirnya dapat memperlambat proses pembuatan situs web. Namun, tidak dapat digeneralisasi sepenuhnya, karena pengembang yang terbiasa dengan alur kerja dan berpengalaman menggunakan kerangka kerja juga bisa menghasilkan karya dengan cepat.

1.2. Pengertian Laravel

Laravel adalah kerangka kerja PHP yang bersifat open-source dan mengandung banyak modul dasar yang bertujuan untuk meningkatkan efisiensi kinerja PHP dalam proses pengembangan aplikasi web. Terlebih lagi, mengingat bahasa pemrograman PHP yang dinamis, peran Laravel dalam hal ini adalah untuk mempercepat, meningkatkan keamanan, serta menyederhanakan pengembangan web.

Pada tanggal 9 Juni 2011, Taylor Otwell merilis kerangka kerja laravel. Awalnya, Laravel diciptakan sebagai alternatif untuk CodeIgniter. Taylor Otwell menyadari bahwa framework PHP lain, yaitu CodeIgniter, belum menyediakan beberapa fitur penting seperti dukungan bawaan untuk otentikasi dan otorisasi pengguna. Sejak saat itu, Laravel telah mengalami banyak pembaruan dan penambahan fitur. Saat ini, Laravel telah mencapai versi 9 dengan beragam fitur unggulannya. yang mengikuti pola arsitektur MVC (Model View Controller).

Laravel telah menjadi salah satu framework PHP paling populer di seluruh dunia. Karena disebabkan oleh kemudahan penggunaannya, dukungan yang melimpah seperti dokumentasi yang terperinci dan beragam perpustakaan yang lengkap, sehingga Laravel dianggap memberikan kenyamanan kepada penggunanya.

Laravel beroperasi di belakang layar atau yang biasa disebut sebagai server-side. Selain memiliki kekuatan yang luar biasa, Laravel juga memiliki tingkat pemahaman yang mudah. Dengan mengadopsi kerangka arsitektur model-view-controller (MVC), Laravel mampu mempercepat proses pembuatan aplikasi web. Dalam kerangka arsitektur MVC ini, pengembangan dapat dipercepat karena para pengembang dapat fokus pada komponen tertentu seperti model (komponen yang mengurus basis data), view (bagian yang mengontrol tampilan yang diberikan kepada pengguna), dan kontroler (bagian yang menghubungkan model dan view dalam menangani permintaan pengguna).

Laravel adalah sebuah kerangka kerja PHP yang bertujuan untuk mempercepat perkembangan pengembangan web. Namun, keunggulan Laravel tidak hanya sebatas itu. Berikut adalah penjelasan tentang keunggulan-keunggulan dari Laravel:

a. Mempercepat Waktu Pengembangan Aplikasi: Laravel memanfaatkan komponen dari kerangka kerja lain dan perpustakaan bawaan dalam proses pengembangan aplikasi web, yang mengakibatkan percepatan dalam pengembangan.
b. Meningkatkan Pengelolaan Sumber Daya: Menggunakan namespace dan antarmuka (interface) membantu dalam pengelolaan sumber daya, membuat pengembangan lebih terstruktur.
c. Kinerja Aplikasi Lebih Optimal: Laravel telah melalui pengujian kualitas dan kecepatan, menghasilkan aplikasi dengan performa lebih cepat.
d. Keamanan yang Ditingkatkan: Aplikasi yang dibangun dengan Laravel secara alami lebih aman terhadap serangan CSRF, injeksi skrip, dan SQL. Laravel juga dilengkapi dengan pengukuran keamanan yang mengadopsi prinsip-prinsip keamanan OWASP.

e. Kode yang Lebih Ringkas: Dengan menggunakan kerangka kerja Laravel, kamu dapat mengurangi penggunaan kode asli dengan memanfaatkan fungsi bawaan yang telah disediakan.
f. Dukungan Komunitas yang Besar: Komunitas besar yang ada di sekitar Laravel memastikan bahwa semua masalah yang mungkin kamu hadapi memiliki peluang resolusi yang tepat dan efektif.

1.2.1 Manfaat Laravel

Tingkat popularitas Laravel tercermin dalam kebutuhan industri. Laravel membawa beberapa keunggulan ketika dijadikan kerangka kerja dasar dalam pengembangan situs web, seperti :

a. Proses pembuatan aplikasi menjadi lebih efisien.
b. Para pengembang lebih mudah dalam merencanakan, membangun, dan merawat aplikasi.
c. Keandalan dan stabilitas aplikasi meningkat karena stabilitas dan kualitas kerangka kerja Laravel telah terbukti.
d. Kode program lebih mudah dibaca oleh pengembang, sehingga bugs dapat terdeteksi lebih cepat.
e. Tingkat keamanan ditingkatkan karena Laravel mengantisipasi potensi kerentanannya.
f. Pembuatan dokumentasi aplikasi menjadi lebih sederhana.
g. Ketersediaan panduan serta dokumentasi lengkap dan mudah diakses.

1.2.2 Fitur-Fitur Laravel

Laravel menawarkan beragam fitur yang tidak selalu tersedia dalam semua kerangka kerja. Sebagai sebuah kerangka kerja modern, Laravel memungkinkan pembuatan situs web dengan fitur-fitur terbaru, seperti proses otentikasi mutakhir.

Dibawah ini adalah beberapa fitur kunci dari Laravel yang penting untuk dipahami:

a. Composer

   Composer adalah alat yang berisi berbagai dependensi dan kumpulan perpustakaan. Semua dependensi disimpan dalam format file composer.json yang ditempatkan di direktori utama situs web. Oleh karena itu, Composer sering disebut sebagai manajemen dependensi.

b. Eloquent ORM

   Eloquent adalah penghubung antara objek-relasional (ORM) di Laravel. Eloquent adalah salah satu fitur utama dalam Laravel yang memungkinkan interaksi tak terbatas dengan model data dan basis data yang dipilih. Melalui Eloquent, Laravel menyederhanakan setiap hambatan yang terlibat dalam interaksi dan penyusunan kueri SQL kompleks untuk mengakses data dari basis data.

c. Artisan CLI

   Artisan CLI adalah antarmuka baris perintah dalam Laravel. Dengan Artisan, dapat mengubah atau memodifikasi aspek-aspek Laravel tanpa perlu menavigasi ke direktori yang sesuai. Bahkan, menggunakan Artisan, dapat langsung terhubung ke basis data melalui Laravel Tinker tanpa harus menginstal klien basis data.

d. Routing

   Dalam kerangka kerja Laravel, semua permintaan dipetakan melalui route. Dasar dari pengarahan ini adalah mengarahkan permintaan ke kontroler yang sesuai. Pengaturan rute ini dianggap dapat menyederhanakan pengembangan situs web dan meningkatkan performanya.

   Terdapat tiga jenis kategori route dalam Laravel: route dasar, parameter route , dan route bernama.

e. MVC Architecture

Laravel mengadopsi pola arsitektur MVC, yang memudahkan adaptasinya karena mengikuti pola yang umum dalam pengembangan aplikasi web. Pola MVC ini banyak digunakan dalam berbagai kerangka kerja, seperti AdonisJS dalam bahasa JavaScript dan ASP.NET MVC dalam bahasa C#.

f. Paginasi Otomatis

Paginasi adalah salah satu fitur Laravel yang berfungsi untuk membatasi tampilan data agar tetap rapi dan terkelola. Dalam situs web, data dapat dibagi menjadi beberapa halaman. Dengan fitur bawaan yang disediakan oleh Laravel, paginasi otomatis diciptakan saat menggunakan kerangka kerja ini.

1.3. Latihan

1. Sebutkan dan jelaskan perbedaan antara PHP framework dan PHP Native!
2. Kapan sebaiknya menggunakan PHP framework dan kapan sebaiknya menggunakan PHP Native?

BAB 2
INSTALASI LARAVEL MENGGUNAKAN COMPOSER

Capaian Pembelajaran :

1. Mampu memahami dan melakukan instalasi xampp
2. Mampu memahami dan melakukan instalasi composer
3. Mampu memahami dan melakukan instalasi laravel

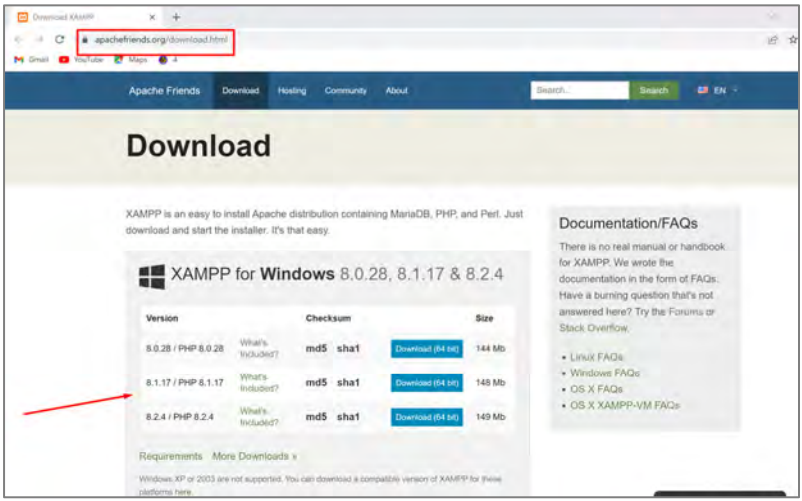
2.1. Instalasi XAMPP

XAMPP merupakan perangkat lunak atau server lokal untuk penggunaan offline. Dengan XAMPP, pengguna dapat mengelola basis data yang berada di komputer sendiri tanpa perlu terhubung ke internet. Ini memastikan bahwa ketika koneksi internet bermasalah dan akses ke server web terhambat, pengguna tidak perlu merasa cemas.

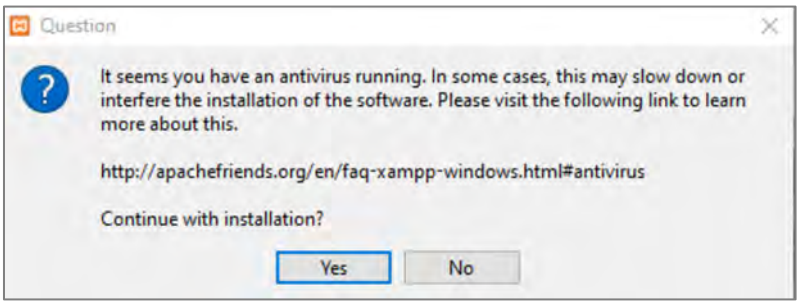
Sebagai perangkat lunak sumber terbuka yang berbasis pada server web, XAMPP dilengkapi dengan berbagai program dan mendukung sejumlah sistem operasi yang umum digunakan, termasuk Linux, Windows, MacOS, dan Solaris. XAMPP berperan sebagai server lokal yang menyediakan program-program seperti Apache, MySQL, dan PHP secara bersamaan.

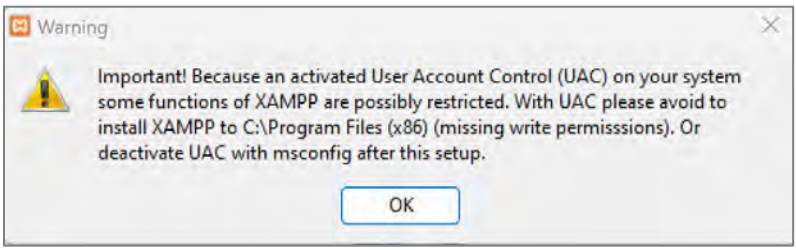
Berikut cara instalasi xampp di sistem operasi Windows :

1. Download aplikasi xampp terbaru :
   Download aplikasi xampp dapat dilakukan melalui :
   https://www.apachefriends.org/download.html
   Bisa juga mencari di mesin pencari dengan kata kunci : download xampp.



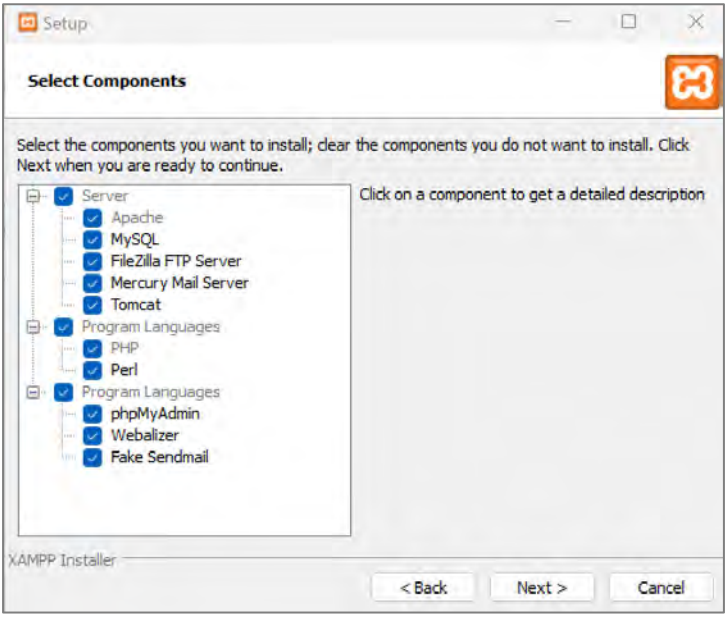
2. Klik dua kali file xampp yang sudah di download. Jika pada proses ini muncul pesan peringatan, maka abaikan saja, dan lanjutkan dengan klik Yes dan Ok.



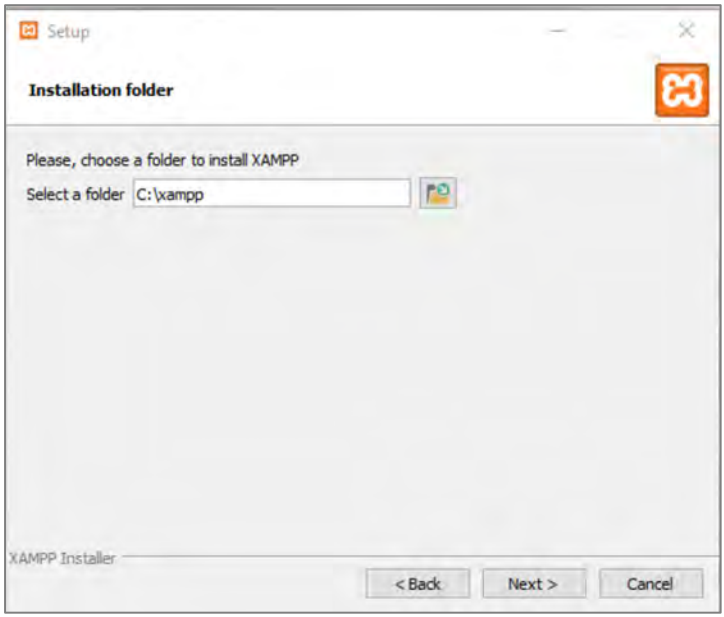


3. Klik next pada jendela installer

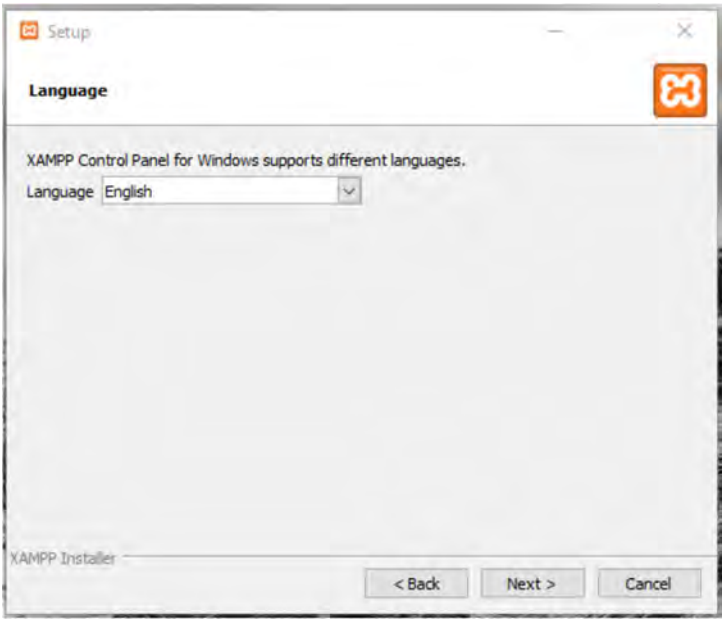
4. Pilih aplikasi yang mau diinstal



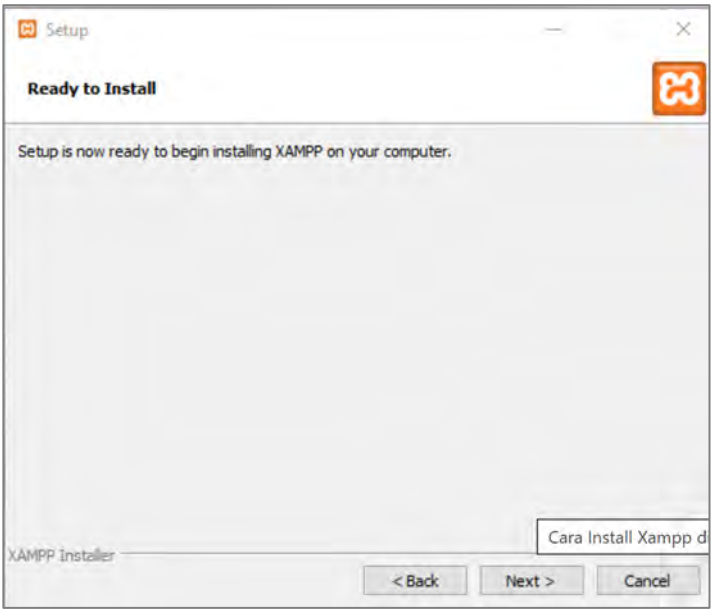
5. Pilih folder instalasi



6. Pilih Bahasa



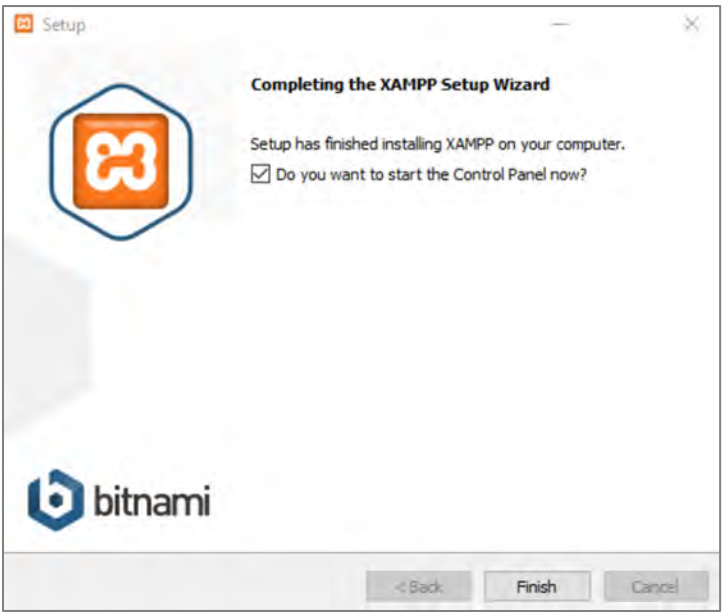
7. Jalankan instalasi



8. Tunggu proses instalasi selesai

9. Start xampp

   Klik tombol finish untuk menyelesaikannya.



2.2. Instalasi Composer

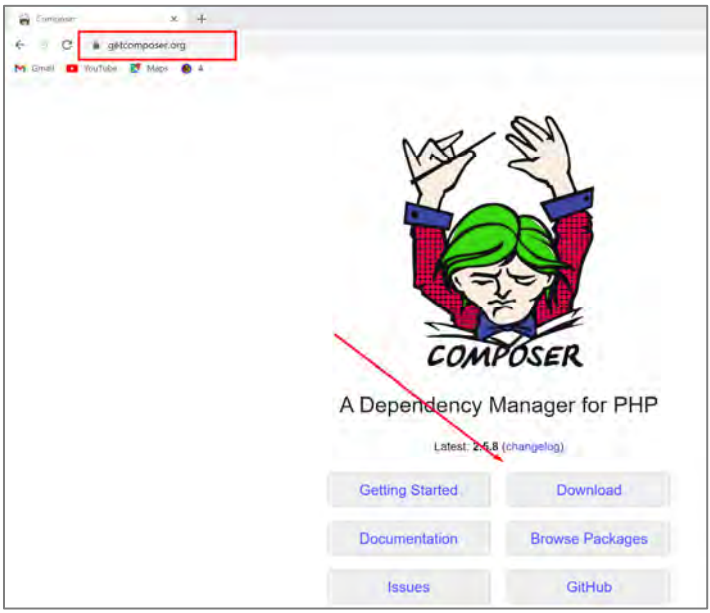
Composer merupakan manajer paket pada level aplikasi untuk bahasa pemrograman PHP. Aplikasi ini memberikan format standar untuk mengatur ketergantungan perangkat lunak dan perpustakaan yang diperlukan dalam PHP. Composer diunduh dalam konteks berbagai kerangka kerja, seperti Magento dan WordPress, yang memerlukan ketergantungan proyek.

Composer adalah manajer paket pada tingkat aplikasi untuk bahasa pemrograman PHP yang menawarkan kerangka kerja standar untuk mengelola ketergantungan perangkat lunak PHP dan perpustakaan yang dibutuhkan. Ini digunakan untuk memeriksa versi Composer, dan penginstalannya mudah dari internet. Composer memungkinkan untuk mengunduh, memperbarui, dan menginstal berbagai paket Laravel.

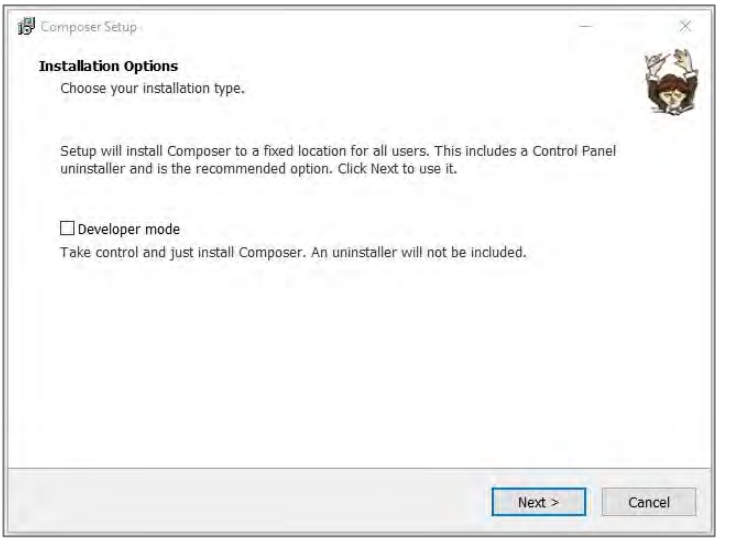
Composer merupakan pengelola paket yang berada pada level aplikasi khusus untuk bahasa pemrograman PHP. Aplikasi ini menawarkan kerangka kerja standar dalam mengatur ketergantungan perangkat lunak dan perpustakaan yang dibutuhkan dalam lingkungan PHP. Untuk memeriksa versi Composer, serta penginstalannya yang mudah melalui internet. Melalui Composer, programer dapat mengunduh, memperbarui, dan menginstal berbagai paket yang berkaitan dengan Laravel.

Langkah instalasi composer yaitu :

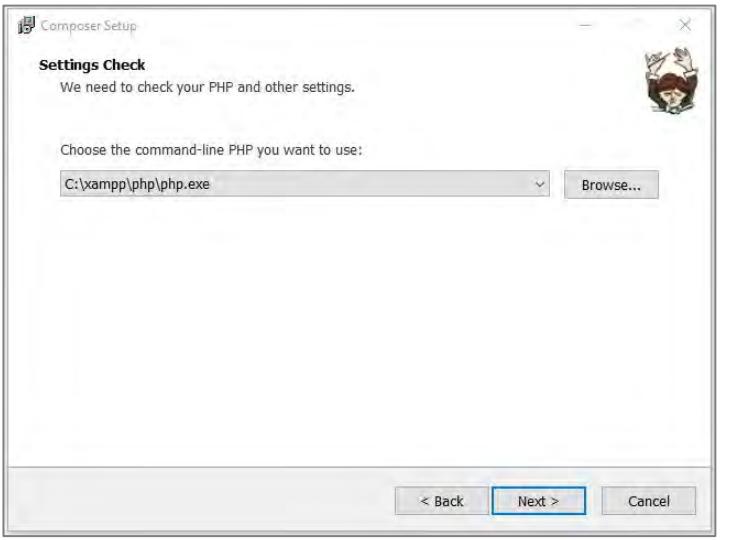
1. Download file installer dari web resminya di Getcomposer.org



2. Klik dua kali pada file `Composer-Setup.exe`. Di jendela pertama, biarkan checkbox **Developer mode** kosong dan langsung saja klik tombol Next.

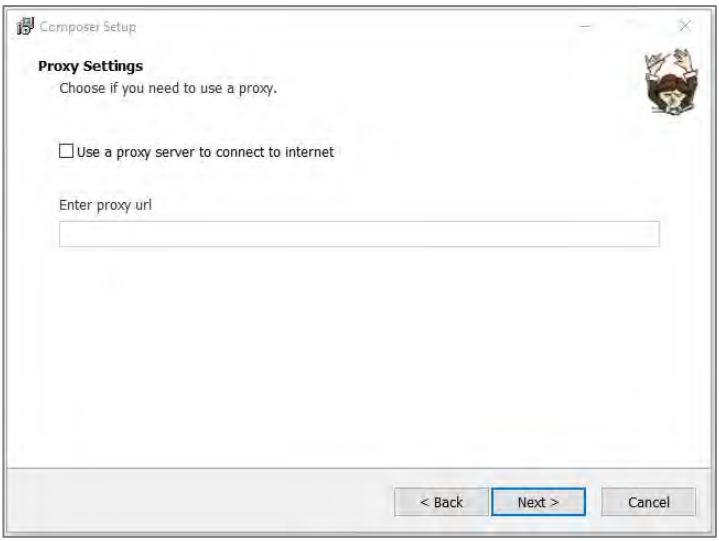


3. Tentukan lokasi file instalasi composer. File tersebut biasanya berada di dalam folder php. Jika menggunakan XAMPP, lokasi file akan berada di C:\xampp\php. Pastikan pilih lokasi instalasi Composer berjalan di C:\xampp\php\php.exe. Klik Next apabila lokasi file php sudah benar.

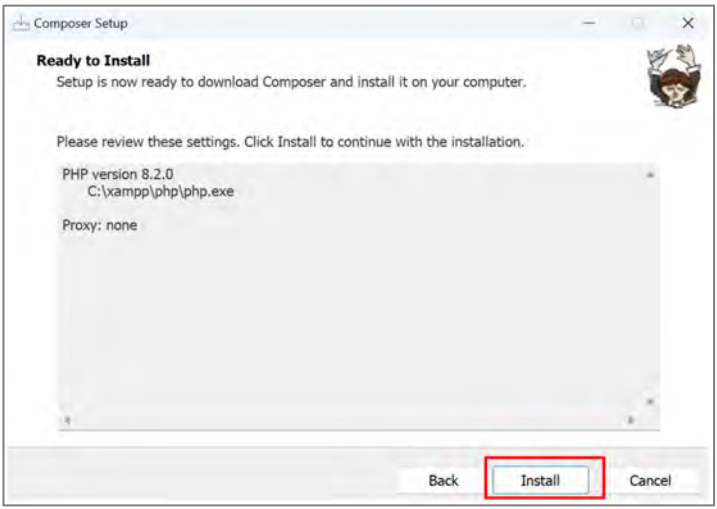


4. Pilih penggunaan proxy. Jika ingin menggunakan proxy, klik centang dan masukkan URL proxy. Apabila tidak ingin

menggunakan proxy, langsung klik Next untuk melanjutkan instalasi.



5. Review lokasi instalasi dengan memastikan jika proses instalasi berjalan di lokasi file yang seharusnya, yaitu C:\xampp\php\php.exe. Jika sudah benar, klik Install.

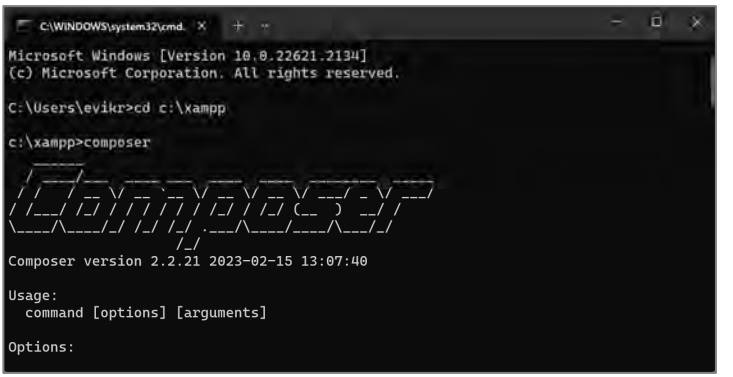


6. Review perubahan instalasi. Perubahan ini dimaksudkan agar Composer dapat dijalankan pada Command Prompt. Jika \sudah

pernah menggunakan Composer sebelumnya, opsi ini tidak akan muncul.

7. Proses instalasi selesai, klik Finish untuk menutup jendela instalasi Composer.

8. Cek instalasi composer dengan menggunakan command prompt. Caranya adalah Tekan Win+R lalu ketik cmd dan klik Ok. Masukan perintah seperti di bawah ini untuk mengecek instalasi sukses atau tidak.



2.3. Instalasi Laravel

Laravel merupakan sebuah kerangka kerja berbasis PHP yang membantu dalam pengembangan website dengan efektifitas maksimal.

Dengan penerapan Laravel, hasil website akan memiliki dinamika yang lebih tinggi.

Kehadiran Laravel sebagai sebuah kerangka kerja memberikan kekuatan lebih pada bahasa pemrograman PHP. Penting untuk diperhatikan bahwa Laravel selalu menawarkan fitur-fitur terkini yang berbeda dari kerangka kerja lainnya.

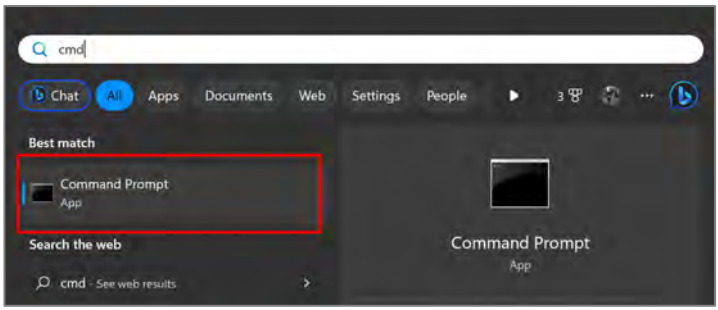
Framework Laravel mengikuti pola struktur MVC (Model View Controller). Pola MVC memisahkan antara data dan tampilan berdasarkan komponen aplikasi. Dengan penggunaan pola MVC, pengguna Laravel akan lebih mudah menguasai sistem ini dan mempercepat proses pembuatan aplikasi berbasis website.

Laravel juga menyajikan beragam fitur bawaan, di antaranya fitur otentikasi. Kerangka kerja ini lebih memusatkan perhatian pada pengalaman pengguna akhir. Keistimewaannya tampak dalam kesederhanaan baik dari segi penulisan kode maupun antarmuka. Meskipun begitu, Laravel tetap mampu menghasilkan aplikasi berbasis website yang kaya akan fitur.

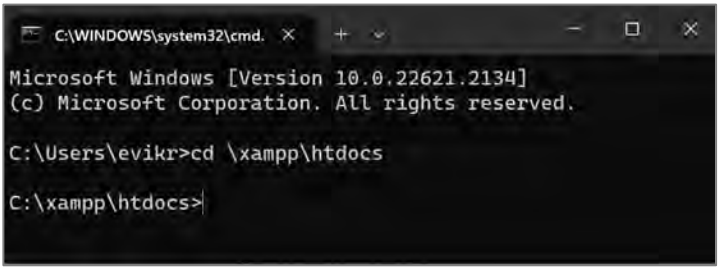
Berkat kemampuannya yang unggul dan fleksibel dalam membantu pengembangan aplikasi berbasis website, banyak perusahaan dan pengembang yang memilih Laravel. Mulai dari perusahaan skala kecil hingga besar.

Langkah-langkah instalasi Laravel, yaitu :

1. Masuk ke command prompt, dengan cara klik Win+R lalu ketik cmd dan klik OK.



2. Masuk ke folder xampp. Sebelum mulai menginstal Laravel, langkah pertama adalah membuka Command Prompt atau terminal dan mengarahkannya ke direktori file server. Lokasi bawaan file server pada XAMPP terletak di dalam folder xampp/htdocs. Dalam jendela Command Prompt, masukkan perintah berikut untuk memasuki direktori htdocs.

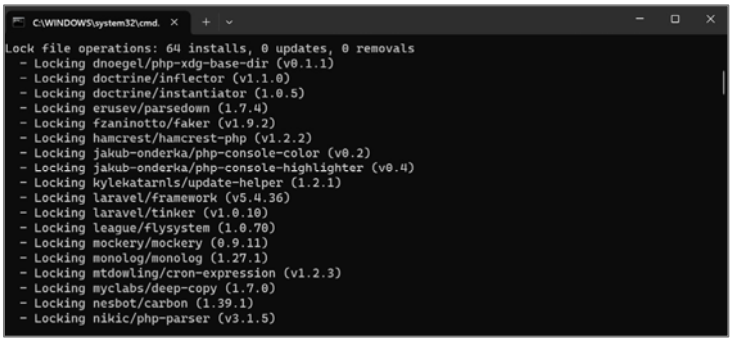


3. Mulai proses instalasi Laravel. Setelah berhasil masuk ke direktori htdocs, langkah berikutnya adalah membuat permintaan untuk mengunduh (dan menginstal) file Laravel yang telah tersedia di repositori Github. Gunakan perintah berikut untuk melakukan permintaan ini:

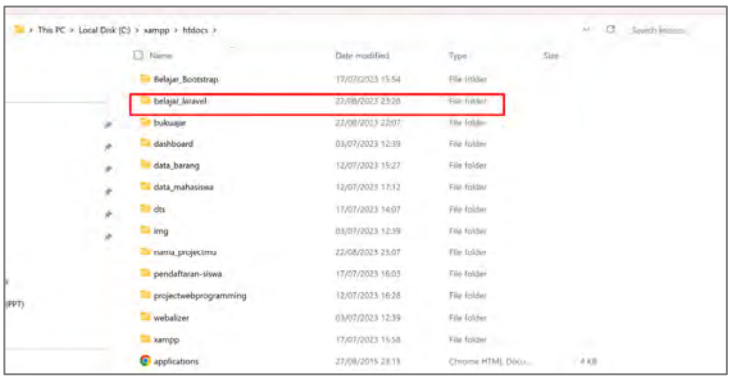
```
composer create-project --prefer-dist laravel/laravel belajar_laravel
```

   a. Create-project : perintah untuk membuat project baru
   b. -prefer-dist : untuk memerintahkan mengunduh Laravel versi yang direkomendasikan atau versi yang stabil (terbaru).
   c. Belajar_laravel : nama project Laravel yang dibuat.

   Setelah perintah dimasukkan, Composer akan memulai proses pengunduhan dan instalasi data Laravel ke dalam direktori yang telah ditentukan. Penting untuk memastikan bahwa koneksi internet stabil agar tidak terjadi gangguan selama proses pengunduhan data Laravel.

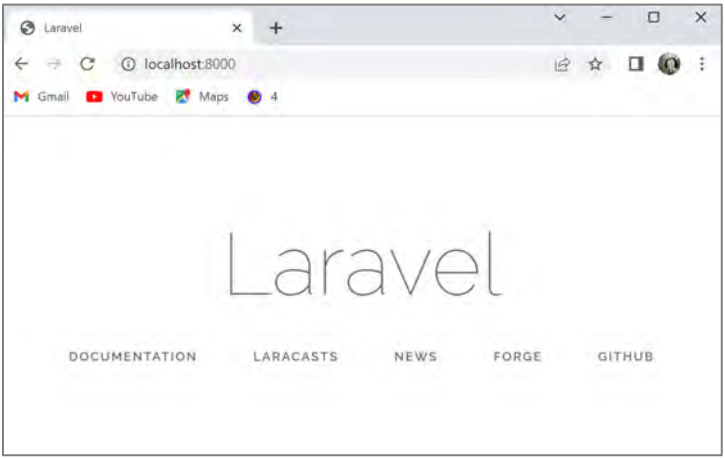


4. Cek instalasi Laravel di Web browser. Setelah proses pengunduhan file Laravel selesai, terdapat sebuah direktori baru di dalam lokasi file server, dan direktori ini akan dinamai sesuai dengan nama proyek yang telah ditentukan sebelumnya dalam folder /xampp/htdocs.



Untuk memverifikasi bahwa instalasi Laravel berhasil dan siap digunakan, arahkan Command Prompt atau Terminal ke direktori yang telah di buat sebelumnya. Setelah itu, masukkan perintah berikut ke dalam Command Prompt atau Terminal:

Jika melihat pesan "Laravel development server started" di Command Prompt atau Terminal, langkah selanjutnya adalah membuka tautan yang disediakan oleh Laravel. Secara default, akan diarahkan ke alamat server, yaitu 127.0.0.1:8000. Pada halaman beranda, akan melihat tulisan "Laravel" di tengahnya, seperti yang ditampilkan dalam gambar di bawah ini:



2.4. Latihan

1. Jelaskan kendala yang mungkin saja terjadi pada saat proses instalasi xampp, composer, maupun Laravel.
2. Sebutkan perintah untuk instalasi Laravel!

BAB 3
OOP BASIC PHP

Capaian Pembelajaran :

1. Mampu memahami pengertian OOP
2. Mampu memahami pengertian Objek
3. Mampu memahami pengertian class
4. Mampu memahami pengertian encapsulation
5. Mampu menerapkan OOP, objek, class, encapsulatin, contruktor, dan method di PHP

3.1. Pengertian OOP

Pemrograman berorientasi objek (Object-oriented programming atau OOP) adalah sebuah model pemrograman komputer yang mengorganisir desain perangkat lunak berdasarkan data, atau objek, daripada berdasarkan fungsi dan logika. Sebuah objek dapat didefinisikan sebagai bidang data yang memiliki atribut-atribut unik dan perilaku tertentu.

OOP berfokus pada objek-objek yang ingin dikendalikan oleh pengembang daripada logika yang diperlukan untuk mengendalikannya. Pendekatan ini dalam pemrograman sangat cocok untuk program-program yang besar, kompleks, dan secara aktif diperbarui atau dikelola. Hal ini termasuk program-program untuk manufaktur dan desain, serta aplikasi-aplikasi mobile; sebagai contoh, OOP bisa digunakan dalam perangkat lunak simulasi sistem manufaktur.

Struktur dari program berorientasi objek juga membuat metode ini bermanfaat untuk pengembangan kolaboratif, di mana proyek-proyek dibagi menjadi kelompok-kelompok. Manfaat tambahan dari OOP meliputi penggunaan ulang kode, skalabilitas, dan efisiensi.

Berbeda dengan logika pemrograman terstruktur, obyek dalam OOP mampu menerima pesan, memproses data, dan berkomunikasi dengan obyek lain. OOP diciptakan untuk mengatasi keterbatasan yang ada dalam bahasa pemrograman tradisional. Prinsip inti dari OOP adalah memecah masalah menjadi bagian-bagian yang lebih kecil dan menggabungkannya dalam bentuk obyek. Dalam konteks OOP, data dan fungsi yang bekerja pada data tersebut digabungkan menjadi suatu kesatuan yang disebut sebagai obyek.

3.2. Pengertian Class

Class merupakan suatu cetak biru yang merepresentasikan entitas dalam dunia nyata, dan menetapkan property dan method dari entitas tersebut.

Dalam pemrograman berorientasi objek (OOP), class digunakan untuk merancang kerangka kerja, hampir seperti sebuah library. Class memuat property dan method. Class berperan sebagai suatu kontainer yang mengandung property dan methode, dan object yang dibentuk sering kali didasarkan pada struktur class tersebut.

Sebagai analogi, Class dapat diumpamakan seperti handphone. Handphone memiliki karakteristik seperti merek, warna, kapasitas, dan beberapa fitur khas lainnya yang menandakan bahwa objek tersebut adalah handphone. Selain memiliki atribut-atribut tersebut, handphone juga dapat dijalankan aksinya, seperti menghidupkan atau mematikan.

Analogi ini menggambarkan bahwa Class adalah gambaran umum mengenai suatu objek. Dalam konteks pemrograman, contoh Class bisa berupa "koneksi_database" atau "mahasiswa".

Dalam PHP, penulisan Class dimulai dengan kata kunci "class", lalu diikuti dengan nama Class. Aturan penamaan Class serupa dengan aturan penamaan variabel dalam PHP, yaitu dimulai dengan huruf atau garis bawah untuk karakter pertama, dan dapat diikuti oleh huruf, garis

bawah, atau angka untuk karakter-karakter berikutnya. Isi dari kelas didefinisikan di dalam tanda kurung kurawal.

Contoh penulisan class dalam PHP :

```php
<?php
    class mahasiswa
    {
        //isi dari class mahasiswa
    }
?>

```

3.3. Pengertian Property

Atribut, yang sering disebut sebagai property, merupakan data yang ada dalam suatu class. Sebagai ilustrasi, property pada laptop bisa mencakup merek, jenis, warna, kapasitas mesin, dan lain sebagainya. Contoh lain class mahasiswa, property di dalamnya ada NIM, NamaMahasiswa, JenisKelamin, dan lain sebagainya. Dalam pemrograman PHP, property pada dasarnya mirip dengan variabel yang berada dalam class. Semua aturan dan jenis data yang umumnya digunakan dalam penulisan variabel, juga dapat diterapkan dalam mendefinisikan property. Pedoman untuk menamai properti juga sejalan dengan pedoman penamaan variabel.

Berikut contoh penulisan property di dalam PHP :

```php
<?php
    class mahasiswa
    {
        //property class mahasiswa
        public $NIM;
        public $nama;
        publik $jeniskelamin;

    }
?>
```

3.4. Pengertian Construktor

Constructor adalah metode atau fungsi yang secara otomatis dieksekusi saat suatu class diinisiasi (objek dibuat), dalam bagian konstruktor kita dapat melakukan segala hal yang umumnya dilakukan dalam metode atau fungsi, kecuali mengembalikan nilai atau nilai balik.

PHP memungkinkan pengembang untuk mendeklarasikan method constructor untuk Class. Class yang memiliki method constructor akan

memanggil method pada setiap objek yang baru dibuat, sehingga method constructor cocok untuk segala inisialisasi yang mungkin diperlukan oleh objek sebelum digunakan.

Constructor induk tidak dipanggil secara implisit jika class anak mendefinisikan sebuah constructor. Untuk menjalankan constructor induk, diperlukan pemanggilan parent ::__construct() di dalam constructor anak. Jika anak tidak mendefinisikan constructor, maka constructor dapat diwarisi dari kelas induk seperti methode class biasa (jika tidak dideklarasikan sebagai private).

3.4.1. Method Constructor dan Destructor

Method adalah aksi yang dapat dijalankan di dalam sebuah class. Contoh method adalah: menyalakan laptop, mematikan laptop, mengganti casing laptop, dan berbagai tindakan lainnya.

Constructor adalah method biasa yang dipanggil saat objek yang sesuai diinisiasi. Sebagai method biasa, mereka dapat mendefinisikan sejumlah argumen, yang mungkin diperlukan, mungkin memiliki tipe, dan mungkin memiliki nilai default. Argumen konstruktor dipanggil dengan menempatkan argumen-argumen tersebut dalam tanda kurung setelah nama class.

Ketika sebuah argumen constructor menyertakan modifikasi visibilitas, PHP akan mengartikannya sebagai properti objek dan juga argumen constructor, serta mengassign nilai argumen ke properti tersebut. Isi dari constructor dapat kosong atau berisi pernyataan lain. Pernyataan tambahan akan dieksekusi setelah nilai-nilai argumen telah diberikan kepada properti-properti yang sesuai.

PHP memiliki konsep destructor yang serupa dengan bahasa pemrograman berorientasi objek lainnya, seperti C++. Metode destructor akan dipanggil segera setelah tidak ada referensi lain terhadap suatu objek tertentu, atau dalam urutan apa pun selama proses shutdown.

Seperti constructor, destructor induk tidak akan dipanggil secara implisit oleh mesin. Untuk menjalankan destructor induk, seseorang harus secara eksplisit memanggil parent::__destruct() di dalam destructor. Sama seperti konstruktor, kelas anak dapat mewarisi destructor induk jika kelas anak itu sendiri tidak mengimplementasikan destructor.

Berikut contoh constructor dan destroctur.

```php
<?php
    class mahasiswa
    {
        //property class mahasiswa
        public $NIM;
        public $nama;
        publik $jeniskelamin;

        public function __construct($NIM, $nama, $jeniskelamin)
        {
            $this->NIM = $NIM;
            $this->nama = $nama;
            $this->jeniskelamin = $jeniskelamin;
        }
    }
?>
```

3.5. Pengertian Objek

Sebuah objek adalah sekumpulan variabel dan fungsi yang digabungkan menjadi satu kesatuan. Kesatuan tersebut juga bisa terdiri dari variabel biasa. Sebuah obyek terbentuk melalui suatu kelas atau yang biasa disebut sebagai instance dari kelas tersebut. Terdapat dua unsur utama dalam sebuah obyek:

1. Atribut atau Properti: Ini adalah nilai-nilai yang tersimpan dalam obyek dan secara langsung atau tidak langsung menggambarkan karakteristik dari obyek tersebut.
2. Method: merupakan aksi yang akan dilakukan atau dieksekusi oleh obyek..

   Contoh penggunaan object seperti :

```php
<?php
    class mahasiswa
    {
        //property class mahasiswa
        public $NIM;
        public $nama;
        public $jeniskelamin;

        public function __construct($NIM, $nama, $jeniskelamin)
        {
            $this->NIM = $NIM;
            $this->nama = $nama;
            $this->jeniskelamin = $jeniskelamin;
        }
        public function getNIM()
        {
            return $this->NIM;
        }
        public function getNama()
        {
            return $this->nama;
        }
        public function getJenisKelamin()
        {
            return $this->jeniskelamin;
        }
    }
    $objectMahasiswa = new mahasiswa('123456 ', 'Andi ', 'L ');
    echo $objectMahasiswa->getNIM(); // print 123456
    echo $objectMahasiswa->getNama(); // prints Andi
    echo $objectMahasiswa->getJenisKelamin(); // prints L

?>
```

3.6. Encapsulation

Encapsulation merupakan elemen yang signifikan dalam pemrograman berorientasi objek yang memungkinkan pembatasan terhadap akses terhadap properti atau metode dari objek khusus. Encapsulation dapat digunakan jika properti objek bersifat pribadi dan diperbarui melalui metode publik. Encapsulation dalam PHP dapat dicapai dengan menggunakan implementasi spesifier akses. Encapsulation sangat memperhatikan konsep pewarisan dalam pemrograman berorientasi objek karena sering kali pewarisan dapat mengganggu konsep Encapsulation.

Keuntungan dari Enkapsulasi:

1. Pengubahan Data dan Abstraksi: Detail yang tidak perlu, representasi internal, dan implementasi disembunyikan dari

pengguna akhir untuk perlindungan struktur data dan kelas tersebut.

2. Keamanan Data: Enkapsulasi membantu membuat data sangat kuat dan aman, karena data dan fungsi anggota dibungkus bersama untuk membentuk suatu objek. Semua tugas dilakukan di dalamnya tanpa kekhawatiran eksternal dan ini juga membuat kehidupan menjadi lebih mudah.
3. Mengurangi kompleksitas: Enkapsulasi membantu mengurangi kompleksitas pengembangan perangkat lunak dengan menyembunyikan detail implementasi dan mengungkapkan metode atau operasi.
4. Dapat digunakan kembali: tidak perlu menulis ulang fungsi yang sama yang diwarisi dari kelas induk.
5. Keandalan: dapat membuat kelas hanya bisa dibaca atau hanya bisa ditulis dengan menulis metode SET atau GET.
6. Pengujian kode lebih mudah: Kode PHP yang dienkapsulasi mudah diuji karena fungsi yang digunakan untuk menguji kelas anak memastikan pengujian fungsi kelas induk juga.
7. Peningkatan fleksibilitas: Variabel kelas dapat diakses melalui metode GET atau SET yang meningkatkan fleksibilitas. Ini mudah dipelihara karena implementasi internal dapat diubah tanpa mengubah kode.

Kesimpulan: Pemrograman berorientasi objek dalam PHP dicapai dengan menggunakan konsep Enkapsulasi yang digunakan untuk menyembunyikan informasi. Metode Getter dan Setter digunakan untuk menghindari akses eksternal yang tidak diinginkan, sehingga membantu dalam memvalidasi nilai-nilai baru yang diberikan ke properti.

Encapsulation dalam PHP adalah proses menyembunyikan semua detail rahasia dari sebuah objek yang sebenarnya tidak memberikan kontribusi banyak pada karakteristik penting dari class tersebut.

3.6.1. Jenis Encapsulation Publik

Apabila suatu method atau property dinyatakan sebagai public, maka method dan property tersebut bisa diakses dari luar class maupun dari dalam class itu sendiri, bahkan dalam class turunannya. Tetapi perlu diingat, untuk method dan property yang tidak diberikan hak akses khusus seperti public, private, dan protected, maka hak akses untuk method dan property tersebut secara otomatis akan menjadi public. Berikut contoh penggunaan encapsulation public:

```php
<?php
  class mahasiswa
  {
    public $name;
    public function getName()
    {
      return $this->name;
    }
  }
  $person = new mahasiswa();
  $person->name = 'Evi Lestari';
  echo $person->getName(); // prints 'Evi Lestari'
?>
```

Deklarasi properti name untuk public. Sehingga, dapat diatur dari mana saja di luar class.

3.6.2. Jenis Encapsulation Private

Private adalah hak akses yang membatasi method atau property yang menggunakannya dari dapat diakses dari luar class. Oleh karena itu, hak akses private hanya memperbolehkan akses dari dalam class itu sendiri.

Dengan demikian, jika sebuah property atau method diatur sebagai private, hanya class itu sendiri yang memiliki akses ke property atau method tersebut. Class lain, termasuk class turunan, tidak memiliki akses ke sana. Hak akses private umumnya digunakan untuk menyembunyikan properti dan method agar tidak dapat diakses dari luar class. Berikut contoh penggunaan encapsulation private :

```php
<?php
  class Mahasiswa
  {
    private $name;
    public function getName()
    {
      return $this->name;
    }
    public function setName($name)
    {
      $this->name = $name;
    }
  }
  $person = new Mahasiswa();
  $person->name = 'Evi Lestari'; // percobaan error, jika melakukan percobaan hapus ini
  $person->setName('Evi Lestari');
  echo $person->getName(); // prints 'Evi Lestari'
?>
```

3.6.3. Jenis Encapsulation Protected

Apabila sebuah properti atau metode dinyatakan sebagai protected, berarti properti atau metode tersebut tidak dapat dijangkau dari luar class. Namun, dapat diakses oleh class itu sendiri atau class yang mewarisi dari class tersebut. Jika mencoba mengakses properti atau method yang protected dari luar class, akan menyebabkan timbulnya suatu kesalahan. Berikut contoh penggunaan encapsulation protected :

```php
<?php
  class Mahasiswa
  {
    protected $name;
    public function getName()
    {
      return $this->name;
    }
    public function setName($name)
    {
      $this->name = $name;
    }
  }
  $person = new Mahasiswa();
  $person->name = 'Evi Lestari'; // percobaan error, jika melakukan percobaan hapus ini
  $person->setName('Evi Lestari');
  echo $person->getName(); // prints 'Evi Lestari'
?>
```

3.7. Inheritance

Inheritance adalah konsep dalam pemrograman berorientasi objek di mana sebuah class dapat mewariskan property dan method yang dimilikinya kepada class lain. Konsep inheritance digunakan untuk mengoptimalkan penggunaan kembali kode, dengan menghindari duplikasi kode program.

Konsep inheritance menciptakan struktur atau hierarki class dalam kode program. Class yang memberikan warisan disebut sebagai kelas induk (parent class), super class, atau class dasar.

Sementara kelas yang menerima warisan disebut sebagai kelas anak (child class), sub kelas, kelas turunan, atau kelas pewarisan.

Namun, tidak semua properti dan method dari kelas induk akan diwariskan. Properti dan metode dengan izin akses privat tidak akan diwariskan ke kelas anak. Hanya properti dan metode dengan izin akses dilindungi (protected) dan publik saja yang dapat diakses dari kelas anak.

Kata kunci yang digunakan untuk membuat class anak yaitu “extends” pada class.

Berikut contoh penggunaan pewarisan / inheritance :

```php
<?php
  class Orang
  {
    protected $nama;
    protected $usia;
    public function getName()
    {
      return $this->nama;
    }
    public function setNama($nama)
    {
      $this->nama = $nama;
    }
    private function callToPrivateNamadanUsia()
    {
      return "{$this->nama} is {$this->usia} years old.";
    }
    protected function callToProtectedNamadanUsia()
    {
      return "{$this->nama} is {$this->usia} years old.";
    }
  }
```

```php
class Mahasiswa extends Orang
{
  private $prodi;
  private $ipk;
  public function getAge()
  {
    return $this->usia;
  }
  public function setAge($usia)
  {
    $this->usia = $usia;
  }
  public function getProdi()
  {
    return $this->prodi;
  }
  public function setProdi($prodi)
  {
    $this->prodi = $prodi;
  }
  public function getIPK()
  {
    return $this->ipk;
  }
  public function setIPK($ipk)
  {
    $this->ipk = $ipk;
  }
  public function getNamadanUsia()
  {
    return $this->callToProtectedNamadanUsia();
  }
}

$Mahasiswa = new Mahasiswa();
$Mahasiswa->setNama('Evi Lestari');
$Mahasiswa->setAge(18);
$Mahasiswa->setProdi('Manajemen Informatika');
$Mahasiswa->setIPK('3.99');

echo 'Nama Mahasiswa : '.$Mahasiswa->getName(); // prints 'Evi Lestari'
echo '<br> Usia : ' .$Mahasiswa->getAge(); // prints '18'
echo '<br> Prodi : '.$Mahasiswa->getProdi(); // prints 'Manajemen Informatika'
echo '<br> IPK : '.$Mahasiswa->getIPK(); // prints '3.99'
echo '<br> Nama dan usia : '.$Mahasiswa->getNamadanUsia(); // prints 'Evi Lestari is
18 years old.'
// echo $Mahasiswa->callToPrivateNamadanUsia(); // produces 'Fatal Error'
?>
```

Maka hasil dari kode program di atas adalah :

3.8. Overriding di PHP

Overriding terjadi ketika sebuah class turunan mengganti implementasi suatu method yang telah didefinisikan di kelas induk.

Class turunan memberikan implementasi khusus untuk method tersebut yang berbeda dari implementasi di class induk.

Ketika suatu metode dioverride oleh class turunan, method tersebut dipanggil pada objek dari class turunan, maka akan dieksekusi implementasi method di class turunan, bukan implementasi method di class induk. Namun, jika metode tersebut dipanggil pada objek dari class induk, maka implementasi method di class induk yang akan dieksekusi.

Overriding memiliki peranan penting dalam konsep pewarisan dalam pemrograman berorientasi objek, karena memungkinkan class turunan untuk memperluas atau mengubah perilaku class induk tanpa harus memodifikasi class induk itu sendiri.

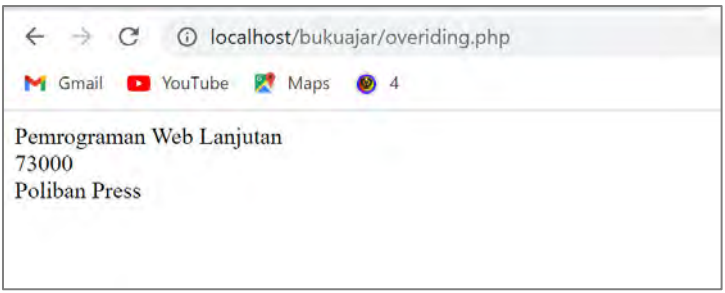
Dengan mengimplementasikan overriding, subclass dapat menerima implementasi method dari class induk dan hanya perlu mengubah bagian-bagian yang relevan, sehingga mengurangi duplikasi kode dan mempercepat proses pengembangan program.

Berikut contoh penggunaan overriding di PHP :

```php
<?php
 class Buku
 {
   var $judul;
   var $harga;

   function getHarga()
   {
     echo $this->harga ."<br/>";
   }

   function getJudul()
   {
     echo $this->judul ." <br/>";
   }

   function __construct( $parameter1, $parameter2 )
   {
        $this->judul = $parameter1;
        $this->harga = $parameter2;
   }
 }
```

```php
Class BukuAjar extends Buku
{
 var $penerbit;

 function getHarga()
 {
  echo $this->harga-5000 ."<br/>";
 }

 Function setPenerbit($parameter)
 {
  $this->penerbit = $parameter;
 }

 Function getPenerbit()
 {
  echo $this->penerbit. "<br />";
 }

 function __construct( $parameter1, $parameter2, $parameter3 )
 {
   $this->judul = $parameter1;
   $this->harga = $parameter2;
   $this->penerbit = $parameter3;
 }
}

/* Buat sebuah object buku ajar dari kelas BukuAjar */
$BukuAjar = new BukuAjar( "Pemrograman Web Lanjutan", 78000, "Poliban Press" );
/* Memanggil Method pada kelas komik */
$BukuAjar->getJudul();
$BukuAjar->getHarga();
$BukuAjar->getPenerbit();
?>
```

Maka hasilnya adalah :



3.9. Latihan

1. Jelaskan apa yang dimaksud dengan variable dalam class? Berikan contoh penggunaannya!
2. Apa yang dimaksud dengan class? Sebutkan bagian dari class!
3. Berikan contoh lain penggunaan inheritance!

BAB 4
ROUTE DAN VIEW

Capaian Pembelajaran :

1. Mampu memahami route
2. Mampu memahami view
3. Mampu membuat route dan view

Mengerti konsep dasar route dan view adalah hal yang mendasar saat mempelajari penggunaan Laravel.

4.1. Pengertian Route

Arti "Route" dalam konteks bahasa Indonesia merujuk pada "rute" atau "jalur". Dalam Laravel, istilah "route" mengacu pada elemen yang mengatur jalur aplikasi yang dibangun menggunakan Laravel. Elemen ini menciptakan suatu URL lengkap yang dapat diakses. Contohnya, ketika ingin membuat halaman seperti localhost/blog, langkahnya adalah membuat rute untuk blog. Dengan mengkonfigurasi rute ini, kita bisa membuka tampilan, menjalankan pengontrol, dan melaksanakan tindakan lain saat rute blog diakses.

Route atau Routing berperan sebagai penghubung antara pengguna dengan kerangka kerja secara keseluruhan. Di Laravel, setiap alamat web yang dimasukkan melalui peramban akan melalui proses routing terlebih dahulu. Melalui rute ini, ditentukan kemana proses selanjutnya akan diarahkan, apakah ke Pengontrol atau Tampilan (View).

Dalam direktori proyek Laravel, terdapat subdirektori "routes" yang memuat empat berkas, yakni api.php, channels.php, console.php, dan web.php. Masing-masing berkas memiliki tujuan spesifik seperti berikut:

1. api.php: Berkas ini berfungsi untuk menetapkan rute-rute API. Selain itu, dalam berkas ini juga mungkin digunakan untuk membangun layanan inti API dengan bantuan Laravel.
2. channels.php: Berkas ini ditujukan untuk menetapkan rute-rute yang berkaitan dengan penyiaran acara (broadcasting event), contohnya notifikasi.
3. console.php: Berkas ini digunakan untuk menentukan rute-rute perintah yang dapat dijalankan melalui terminal. Ini memberi kita kemampuan untuk menciptakan perintah Artisan kustom sesuai kebutuhan kita.
4. web.php: Berkas ini berperan dalam menetapkan rute-rute web standar. Di sini, kita dapat mengatur rute-rute yang terkait dengan navigasi biasa dalam aplikasi web.

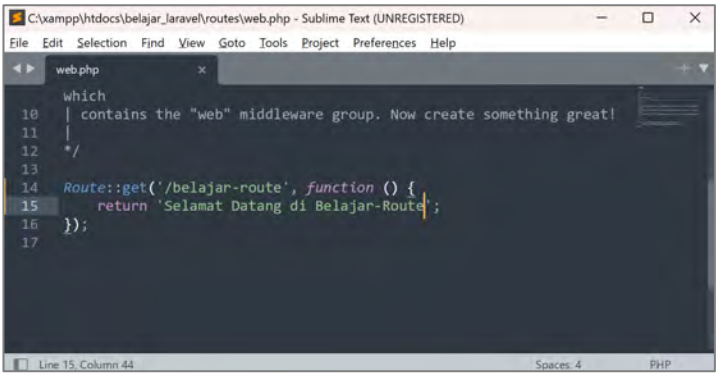
4.1.1. Dasar Route

Laravel menyuguhkan pendekatan yang sangat mudah dipahami dan jelas dalam mengartikulasikan rute-rute. Prinsip dasar dari Rute Laravel bahkan hanya memerlukan penerimaan URI (*Uniform Resource Identifier*), fungsinya agar pengguna dapat mengetahui setiap alamat yang bisa diakses berdasarkan respon dari HTTP verb.

Dasar route dapat dibuat dengan Langkah :

1. Buka project (belajar_laravel) yang telah dibuat pada saat instalasi (lihat BAB 2 instalasi Laravel). Sebelumnya jangan lupa menjalankan kode `Php artisan serve` pada command prompt.
2. Buka file web.php

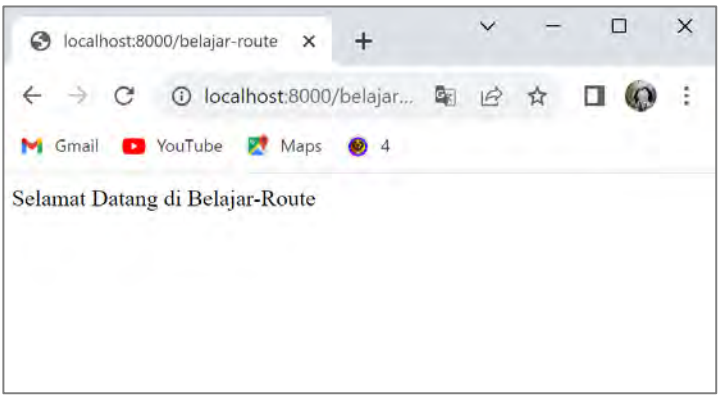
3. Tambahkan kode berikut :



Keterangan:

a. get merupakan *method* yang diizinkan untuk menjalankan fungsi pada *route*.
b. '/belajar-route' merupakan alamat URI yang ingin diakses untuk menjalankan sebuah fungsi pada *route*.
c. return 'Selamat Datang di Belajar-Route'; merupakan *callback function* yang akan dijalankan ketika suatu URI diakses dengan *method* yang sesuai.

4. Sehingga hasilnya :



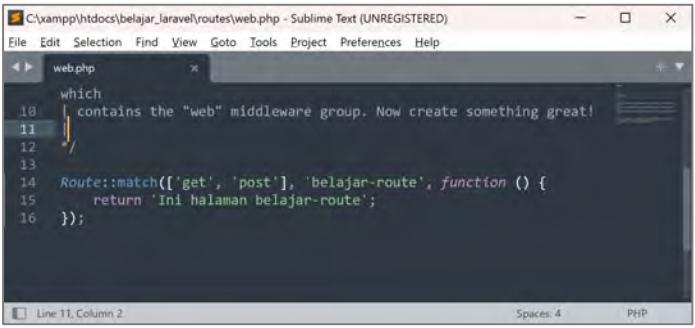
3.9.1. Router Methods

Pemisahan permintaan oleh router berdasarkan metode HTTP juga memungkinkan untuk menentukan rute yang akan merespons metode

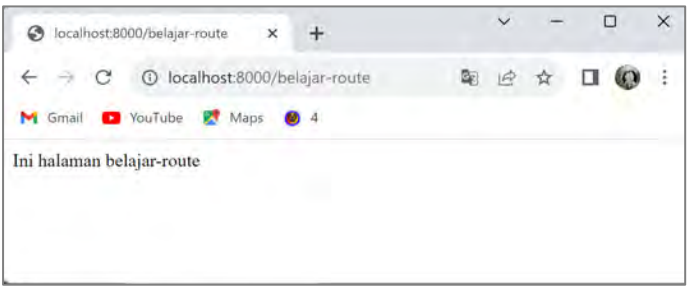
HTTP tertentu. Ada 6 jenis method yang bisa digunakan pada route Laravel untuk merespon HTTP verb, antara lain :

```
Route::get($uri, $callback);
Route::post($uri, $callback);
Route::put($uri, $callback);
Route::patch($uri, $callback);
Route::delete($uri, $callback);
Route::options($uri, $callback);
```

Sebagai contoh, ketikan kode berikut ke dalam file web.php.



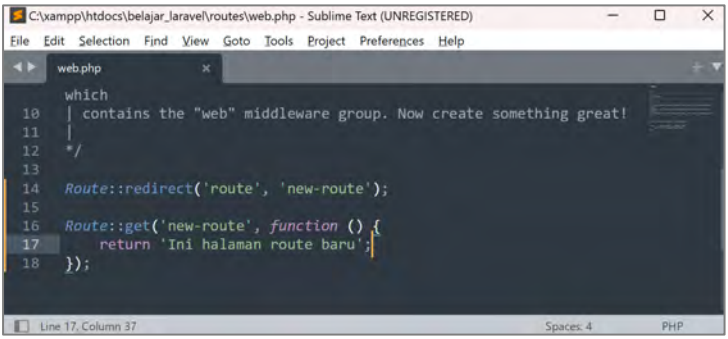
Sehingga hasilnya :



4.1.2. Redirect Routes

Router bukan hanya bertujuan untuk menampilkan hasil kepada pengguna, melainkan juga dapat digunakan untuk mengalihkan pengguna ke URL yang berbeda.

Ketikan kode berikut pada web.php.

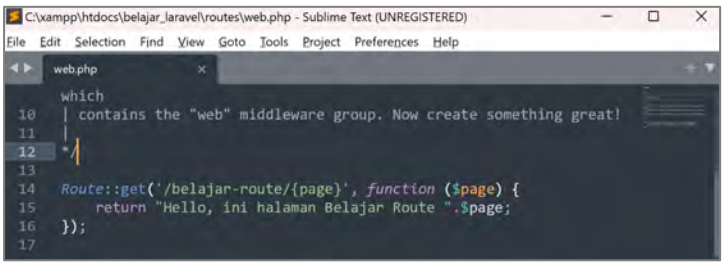


Misalnya, ketika akses dilakukan ke https://localhost:8000/route, pengguna akan diarahkan ke halaman https://localhost:8000/new-route.

4.1.3. Route Parameters

Kadang-kadang, saat menciptakan suatu URI, diperlukan pengambilan parameter yang merupakan bagian dari segmen URI di dalam rute.

Sebagai contoh, pengambilan ID dari URI. Hal ini bisa dicapai dengan menetapkan parameter rute. Silakan tambahkan kode berikut ke dalam berkas routes/web.php:

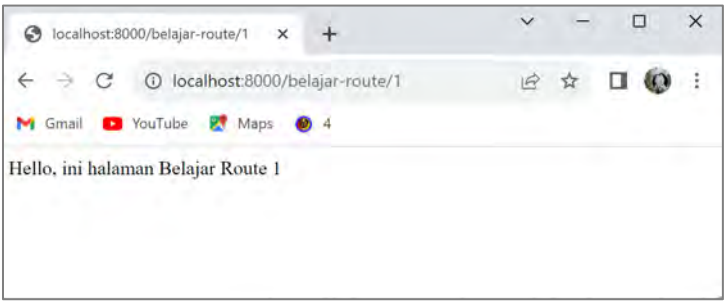


Keterangan:

a. '/belajar-route/{page}' merupakan alamat URI yang memiliki nilai parameter saat akan diakses untuk menjalankan sebuah fungsi pada *route*.
b. function ($page) merupakan fungsi yang menangkap nilai dari parameter.
c. return "Hello, ini halaman Belajar Route ".$page; merupakan *callback function* yang akan dijalankan ketika suatu URI diakses dengan *method* yang sesuai.

d. Namun, apabila nilai parameter tidak dimasukan .. maka page tidak akan ditemukan

Sehingga hasilnya adalah :

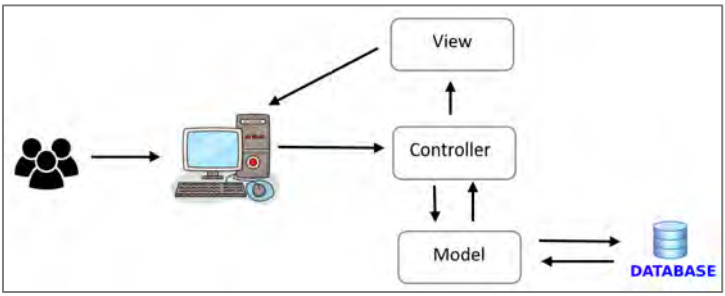


4.2. Pengertian View

View adalah segmen spesifik yang bertanggung jawab atas pengaturan tampilan (antarmuka pengguna) yang akan ditampilkan di peramban web. Istilah "view" merujuk pada hasil tampilan yang akan dihasilkan oleh aplikasi yang akan dibuat.

Views merupakan elemen terintegrasi dalam kerangka kerja Laravel yang secara umum menangani tugas untuk memperlihatkan data yang diterima oleh Pengontrol (Controller) dari Model. Mayoritas isi berkas tampilan (views) terdiri dari kode HTML yang akan diolah oleh peramban web dan kemudian diproyeksikan di layar pengguna.

Views mewakili visualisasi suatu halaman web yang biasanya bertugas untuk menampilkan informasi yang diterima oleh Pengontrol (Controller) dari Model. Dalam konteks ini, Views merupakan komponen inti dari sistem Laravel di mana kode HTML dihasilkan dan kemudian ditampilkan di perangkat layar pengguna. Penggunaan Views sangat menguntungkan dalam pengembangan web, terutama untuk menjaga dan menyediakan penambahan fungsi, karena memisahkan logika utama atau program dengan elemen tampilan.



Dalam ilustrasi gambar di atas, terlihat bahwa tampilan (view) pada dasarnya diakses melalui pengontrol (*controller*), walaupun Laravel juga mengizinkan akses langsung ke tampilan melalui rute. Jalur dari Rute ke Tampilan ini umumnya digunakan untuk menampilkan halaman-halaman statis yang tidak memerlukan pemrosesan yang kompleks.

Namun, apabila suatu saat diperlukan untuk memproses data yang lebih kompleks, maka akses ke tampilan harus melalui pengontrol terlebih dahulu.

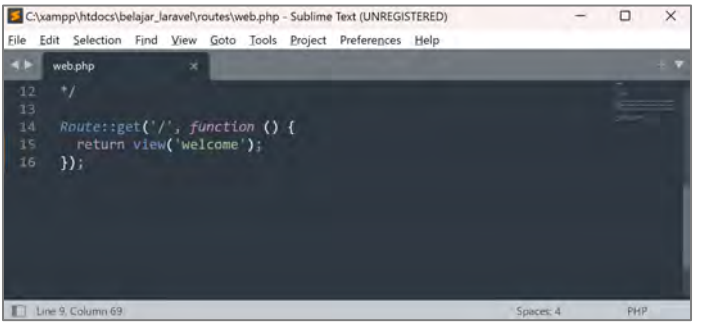
Blade merupakan fitur yang disediakan oleh Laravel untuk proses pembuatan templat yang sederhana namun sangat berguna dalam pengembangan tampilan halaman web. Tidak seperti fitur templating PHP yang umum digunakan, Blade tidak menghalangi pengembang dalam menggunakan kode PHP standar saat membuat kode tampilan. Semua tampilan yang dibuat dengan Blade akan diubah menjadi kode PHP standar dan disimpan dalam cache hingga mengalami perubahan. Ini berarti bahwa Blade pada dasarnya tidak menambahkan beban atau beban ekstra saat aplikasi dijalankan.

File tampilan Blade memiliki ekstensi .blade.php dan biasanya diletakkan di direktori resources/views. Dalam Laravel, Blade menggunakan konsep dasar pewarisan template (template inheritance) dan bagian-bagian (sections).

Salah satu fungsi penting dari fitur Blade templating di Laravel adalah penggunaan tata letak (layout) yang memungkinkan komponen-komponen tampilan yang berulang seperti header, footer, dan sidebar tidak perlu dibuat berulang kali, sehingga mengurangi risiko ketidakkonsistenan. Tata letak ini umumnya ditempatkan dalam folder yang disebut "layout" di dalam direktori views, tetapi pengembang tidak terbatas untuk menggunakan nama lain jika diinginkan.

4.3. Membuat Route dan View

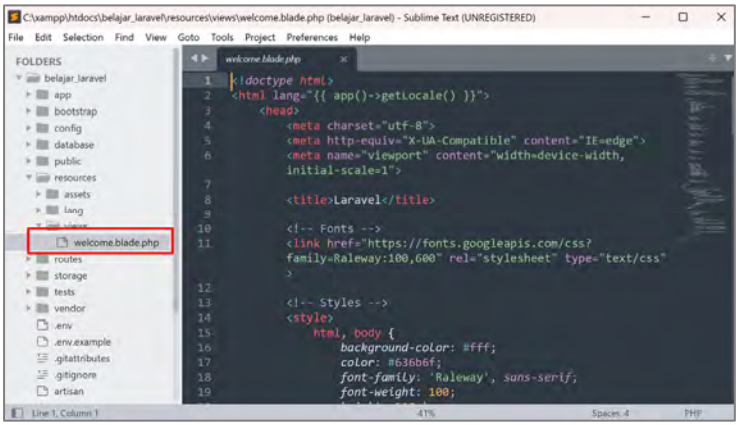
Laravel sudah menyertakan sebuah tampilan default yang akan terlihat saat halaman beranda atau halaman root Laravel diakses. Tampilan ini dipanggil menggunakan kode berikut yang ada di dalam berkas route/web.php:



Kode tersebut dapat diartikan sebagai berikut: "Jika halaman beranda '/' diakses, tampilkan tampilan dengan nama 'welcome'."

Dalam konteks Laravel, nama tampilan mencerminkan nama berkas, sehingga harus ada berkas dengan nama 'welcome' di dalam direktori instalasi Laravel. Laravel menaruh tampilan-tampilan ini di dalam folder resources\views.

Bila direktori ini dibuat, maka akan ditemukan sebuah berkas bernama welcome.blade.php. Berkas inilah yang diacu sebagai tampilan 'welcome'.

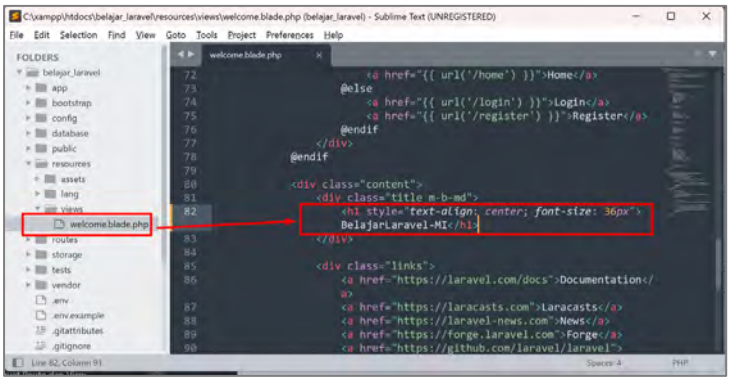


Dalam Laravel, setiap tampilan harus diberi nama dengan format berikut:

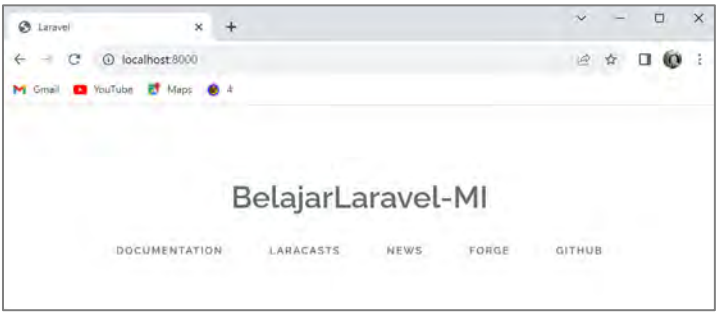
`<nama_file>.blade.php`

Sehingga apabila ingin membuat tampilan 'welcome', berkasnya akan bernama welcome.blade.php. Berkas welcome.blade.php yang telah disediakan oleh Laravel terdiri dari kode HTML, CSS, dan beberapa perintah blade.

Untuk mengkonfirmasi bahwa ini adalah berkas yang mengatur tampilan halaman beranda di Laravel, buka berkas welcome.blade.php ubah kode, seperti contoh berikut:



Sehingga tampilan menjadi :



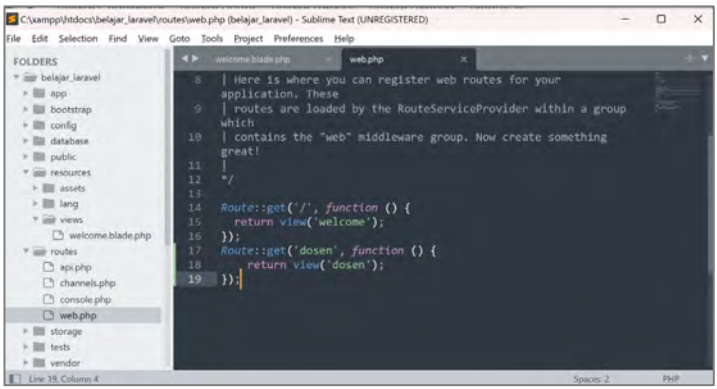
4.4. Membuat Route Baru Laravel

Kesimpulan yang diambil untuk membuat route yaitu :

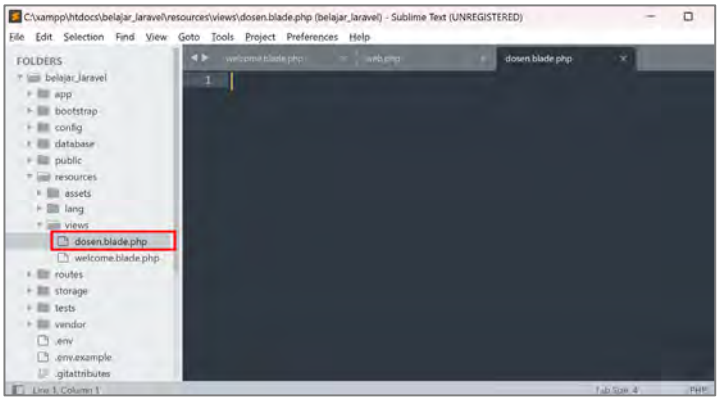
1. Buat route yang akan mengembalikan view.
2. Buat file view di folder resources \ views dengan format **<nama_file>.blade.php**

Untuk membuat route dan view yang baru di Laravel dengan langkah sebagai berikut :

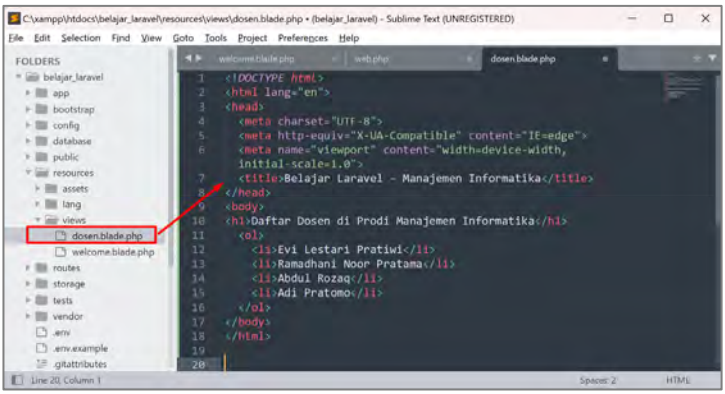
1. Membuat route baru untuk menampilkan view baru. Buat route baru dengan nama “dosen” untuk menampilkan view “dosen”. Buka file `web.php` dan isikan kode menjadi :



2. Buat file view pada direktori `resources/views` bernama `dosen.blade.php`.

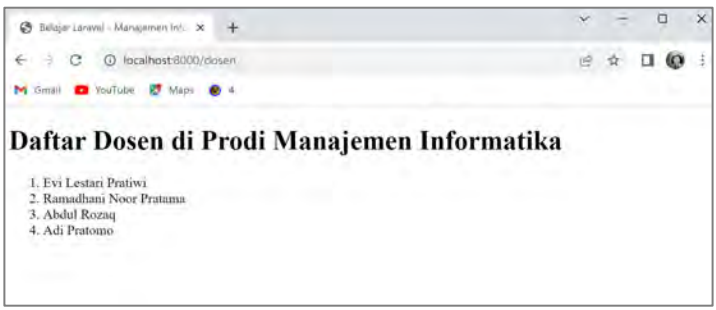


3. Ketikan kode berikut :



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width,
  initial-scale=1.0">
  <title>Belajar Laravel - Manajemen Informatika</title>
</head>
<body>
<h1>Daftar Dosen di Prodi Manajemen Informatika</h1>
  <ol>
    <li>Evi Lestari Pratiwi</li>
    <li>Ramadhani Noor Pratama</li>
    <li>Abdul Rozaq</li>
    <li>Adi Pratomo</li>
  </ol>
</body>
</html>
```

4. Lihat hasilnya dengan mengakses route dosen pada `http://localhost:8000/dosen`. Sehingga hasilnya :



Seperti yang dapat diamati, berkas tampilan dapat diisi dengan kode HTML umum, termasuk juga dapat memuat kode CSS, JavaScript, serta PHP.

Khususnya jika ingin menjalankan kode PHP, Laravel menyediakan sintaks blade yang mengakomodasi penulisan kode program secara lebih efisien. Contohnya, jika ingin menampilkan data dari suatu array, penggunaan blade menjadi lebih praktis dan efektif.

4.5. Latihan

Silahkan mengerjakan view untuk halaman website portfolio yang terdiri dari halaman:

1. Home
2. About
3. Education
4. Project

Silahkan mengubah berbagai view yang ada pada project dengan memanfaatkan fitur Blade Templating.

BAB 5
CONTROLLER

Capaian Pembelajaran :

1. Mampu memahami pengertian controller
2. Mampu membuat controller
3. Mampu menggunakan controller
4. Mampu membuat passing data controller ke view
5. Mampu membuat request data pada Laravel

5.1. Pengertian Controler

Controller merupakan komponen penting dalam pola arsitektur MVC yang bertugas sebagai penghubung antara permintaan pengguna (View) dengan model, dan kemudian mengembalikan respons kepada View dalam bentuk tanggapan. Controller sering berisi banyak logika untuk menyusun fungsi-fungsi tertentu, seperti dalam kegiatan CRUD (Create, Read, Update, Delete), yang seluruhnya diatur di dalam Controller.

Perbedaan utama antara Model dan Controller dalam Laravel terletak pada peran masing-masing. Model difokuskan pada pengelolaan data dalam basis data. Sementara itu, Controller mengkhususkan diri dalam pengaturan logika aplikasi dan penanganan permintaan dari pengguna, serta memberikan respons yang sesuai kepada pengguna terkait permintaan tersebut. Dalam konteks Laravel, Model dan Controller terhubung secara erat dan bekerja bersama-sama untuk membangun aplikasi Laravel. Model memfasilitasi Controller untuk mengambil data yang diperlukan dari basis data, dan kemudian Controller memanipulasi data yang diterima dari Model.

Peran controller dalam Laravel adalah mengatur logika bisnis, menerima input dari pengguna, memproses data, dan menghasilkan respons yang akan ditampilkan pada tampilan.

Fungsi-fungsi utama Controller pada Laravel :

1. Menghandle Permintaan HTTP

   Controller berperan dalam menangani permintaan HTTP yang masuk dari pengguna. Saat pengguna mengakses suatu rute dalam aplikasi web, controller yang sesuai akan dipanggil. Controller ini akan menjalankan tugas yang diperlukan, seperti memproses data, memvalidasi input, dan menentukan tanggapan yang akan dikirim kembali kepada pengguna.

2. Mengurus Logika Bisnis

   Salah satu peran inti dari controller adalah mengelola logika bisnis dalam aplikasi web. Ketika ada permintaan, controller akan memproses data, memvalidasi input, menghubungi model untuk mengakses atau mengubah data, serta menjalankan logika bisnis tertentu. Dengan menempatkan logika bisnis di dalam controller, alur program menjadi lebih terstruktur dan peran-peran dalam MVC dapat dipisahkan dengan jelas.

3. Manipulasi Data

   Controller juga bertugas memanipulasi data. Saat pengguna mengirimkan data melalui permintaan, controller akan menerima dan memeriksa data tersebut. Kemudian, controller dapat menggunakan model untuk mengakses basis data, melakukan operasi CRUD (Create, Read, Update, Delete), serta mengelola data sesuai kebutuhan. Dengan menggunakan controller, manipulasi data dapat diatur dengan lebih mudah dan keutuhan data dalam aplikasi tetap terjaga.

4. Memberikan Respon ke Tampilan (View)

   Setelah controller menyelesaikan pemrosesan permintaan, langkah selanjutnya adalah memberikan respons ke tampilan (view). Controller dapat mengirimkan data yang telah diproses ke

tampilan agar dapat ditampilkan kepada pengguna. Selain itu, controller dapat mengatur format respons, seperti menghasilkan tampilan HTML, JSON, atau jenis respons khusus lainnya. Dengan menggunakan controller, pengaturan respons menjadi lebih teratur dan logika presentasi terpisah dari bisnis.

5. Mengatur Penggunaan Rute (Routing)
   Controller memiliki peran kunci dalam mengelola rute (routing) dalam Laravel. Rute merupakan aturan yang menuntun permintaan pengguna ke controller yang sesuai. Di dalam controller, bisa mendefinisikan metode-metode yang akan dieksekusi ketika suatu rute tertentu diakses. Melalui controller, manajemen rute dalam aplikasi web bisa diatur dengan lebih mudah dan logika rute terpisah dengan jelas.

Controller dalam Laravel sangat memungkinkan pemisahan antara logika bisnis, tampilan, dan data, sehingga memudahkan dalam pengembangan, pemeliharaan, dan pengujian aplikasi. Dengan menggunakan controller, dapat mengatur alur program, mengelola permintaan, dan menghasilkan respons dengan lebih terstruktur dan efisien.

Agar dapat memanfaatkan controller di dalam Laravel, langkah-langkah yang harus diikuti adalah sebagai berikut:

1. Buat controller Baru: Gunakan perintah `php artisan make:controller NamaController` untuk menghasilkan pengontrol baru. Pengontrol ini secara otomatis akan diciptakan dalam direktori `app/Http/Controllers`.
2. Tentukan Metode-Metode: Di dalam berkas pengontrol yang baru dibuat, kita bisa mendefinisikan metode-metode yang akan mengurus permintaan dari pengguna. Setiap metode mewakili rute berbeda dalam aplikasi web kita.

3. Panggil Metode dari Rute: Untuk memanggil metode pada pengontrol dari rute, bisa menggunakan sintaks `Route::get(), Route::post()`, atau metode rute lainnya yang ada di berkas routes/web.php. Sebagai contoh, `Route::get('/halaman', 'NamaController@metode')` akan menggambarkan metode tertentu dalam pengontrol NamaController ketika pengguna mengakses rute /halaman.
4. Proses Permintaan: Pada setiap metode pengontrol, kita mampu memproses permintaan pengguna, memanipulasi data, dan menghasilkan tanggapan yang sesuai. Penggunaan model, layanan (service), atau komponen lainnya yang diperlukan bisa terjadi dalam tahap ini.
5. Sajikan Tanggapan: Setelah selesai memproses permintaan, kita harus menghasilkan tanggapan yang akan ditampilkan di tampilan (view). Kami mampu menggunakan metode `return view()` untuk mengirimkan data yang nantinya ditampilkan oleh tampilan kepada pengguna. Pilihan lain termasuk tampilan JSON yang dihasilkan dengan metode-metode khusus yang disediakan oleh Laravel.

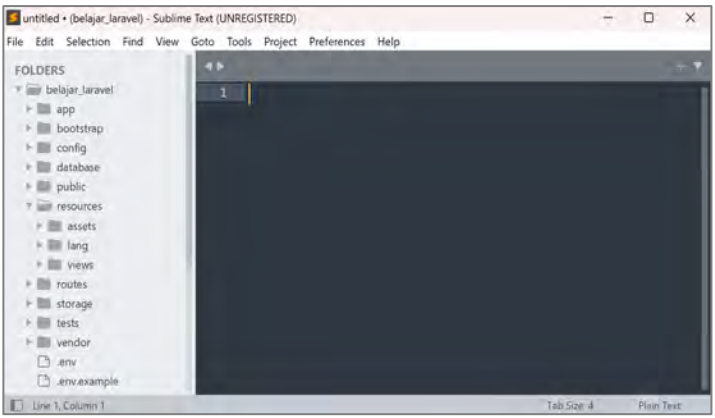
5.1.1. Membuat Controller Laravel

Terdapat dua metode untuk menghasilkan controller dalam Laravel. Pendekatan pertama melibatkan pembuatan file controller secara manual di dalam direktori app/Http/Controllers/. Sementara itu, opsi kedua melibatkan penggunaan perintah php artisan yang disediakan oleh Laravel.

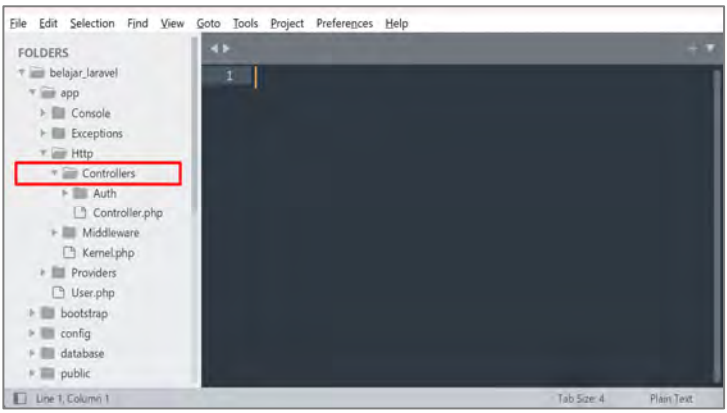
1. Membuat Controller Laravel dengan File Baru

   Membuat controller Laravel dengan membuat file baru adalah salah satu cara untuk menghasilkan controller secara manual di dalam proyek Laravel. Berikut adalah langkah-langkah untuk melakukannya:

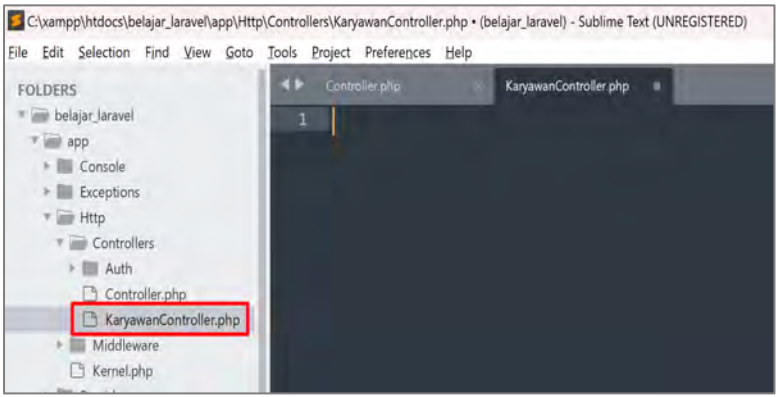
a. **Buka Direktori**: Buka proyek Laravel di editor teks atau lingkungan pengembangan yang di gunakan.



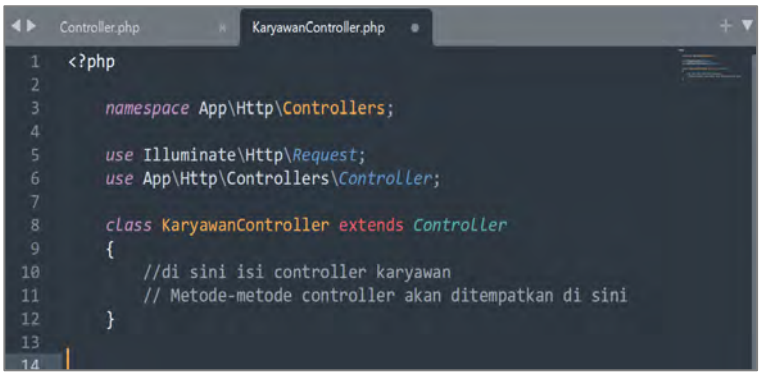
b. **Navigasi ke Direktori Controllers**: Buka direktori `app/Http/Controllers` di dalam proyek. Ini adalah tempat di mana semua controller Laravel akan ditempatkan.



c. **Buat File Baru**: Buat sebuah file baru dengan nama yang sesuai dengan nama controller yang ingin dibuat. Pastikan untuk memberikan ekstensi `.php` kepada file tersebut. Sebagai contoh akan dibuat file dengan nama `KaryawanController.php`.



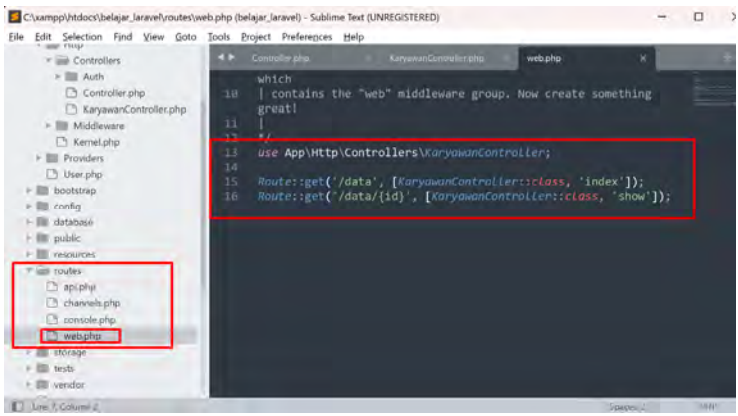
d. **Definisikan Namespace dan Kelas**: Di dalam file baru, perlu mendefinisikan namespace dan kelas controller. Namespace harus sesuai dengan struktur direktori proyek. Kelas controller harus meng-extend kelas Controller yang disediakan oleh Laravel. Contoh :



e. **Definisikan Metode**: Di dalam kelas controller yang baru dibuat, dapat mendefinisikan metode-metode yang akan mengatur logika bisnis sesuai kebutuhan aplikasi. Setiap metode akan mewakili suatu aksi atau tugas tertentu. Contoh :

```php
class KaryawanController extends Controller
{
    public function index()
    {
        // Logika untuk menampilkan daftar data
    }

    public function show($id)
    {
        // Logika untuk menampilkan detail data dengan ID tertentu
    }
}
```

f. **Gunakan Metode dari Rute**: Setelah mendefinisikan metode-metode di dalam controller, panggil dari berkas routes/web.php dengan menentukan rute yang sesuai. Contoh :



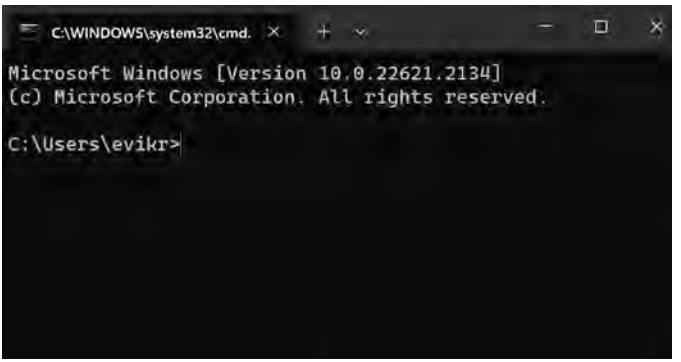
Melalui prosedur di atas, sehingga berhasil menciptakan pengontrol (controller) di Laravel dengan metode pembuatan file baru. Pengontrol ini siap untuk menangani permintaan dari pengguna sesuai dengan logika bisnis yang telah didefinisikan sebelumnya.

2. Membuat Controller Laravel dengan PHP Artisan

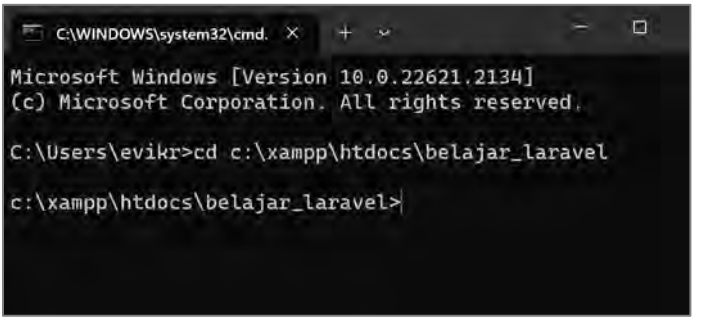
PHP artisan merupakan salah satu fitur utama yang dimiliki oleh Laravel, diciptakan dengan tujuan untuk menyederhanakan proses pengembangan aplikasi yang dikerjakan.

Membuat controller Laravel menggunakan perintah `php artisan` adalah cara lebih cepat dan terotomatisasi untuk menghasilkan controller dalam proyek yang dimiliki. Berikut adalah langkah-langkah untuk membuat controller dengan menggunakan perintah `php artisan`:

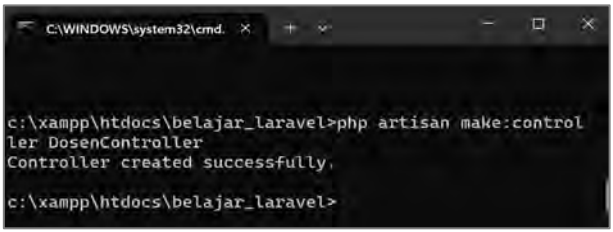
a. **Buka Terminal**: Buka terminal atau command prompt pada sistem.



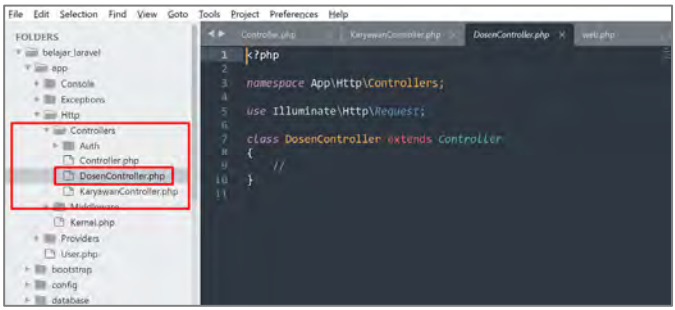
b. **Arahkan ke Direktori Proyek**: Gunakan perintah cd untuk berpindah ke direktori proyek Laravel. Pastikan sudah berada di direktori yang mengandung berkas artisan.



c. **Jalankan Perintah**: Ketik perintah berikut ini untuk membuat controller baru:



d. **Pengontrol Baru Dibuat**: Setelah perintah dijalankan, Laravel akan membuatkan controller baru dengan nama yang telah ditentukan. Controller yang dibuat akan diletakkan di dalam direktori app/Http/Controllers.



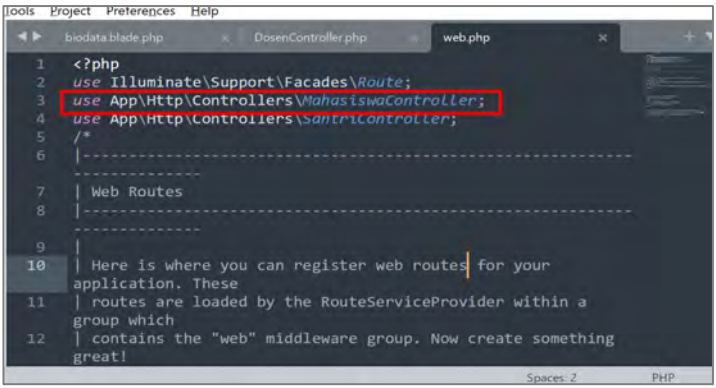
e. **Panggil Metode dari Rute**: Seperti biasa, dapat memanggil metode-metode dari controller yang baru dibuat dari berkas `routes/web.php` menggunakan metode rute seperti `Route::get(), Route::post(),` dan lainnya.

Dengan menggunakan perintah `php artisan` untuk membuat controller, sehingga dapat mempercepat proses pembuatan controller baru dan memastikan bahwa struktur dasar yang diperlukan oleh Laravel telah diatur dengan benar. Controller ini dapat segera digunakan untuk mengelola permintaan dari pengguna sesuai dengan kebutuhan aplikasi.

5.1.2. Menggunakan Controller Laravel

Cara penggunaan controller Laravel, yaitu dengan langkah sebagai berikut :

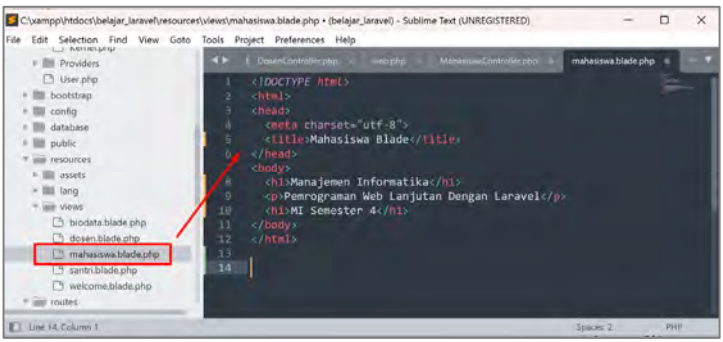
1. Buka file `web.php` pada folder routes, tambahkan baris kode berikut ini paling atas.



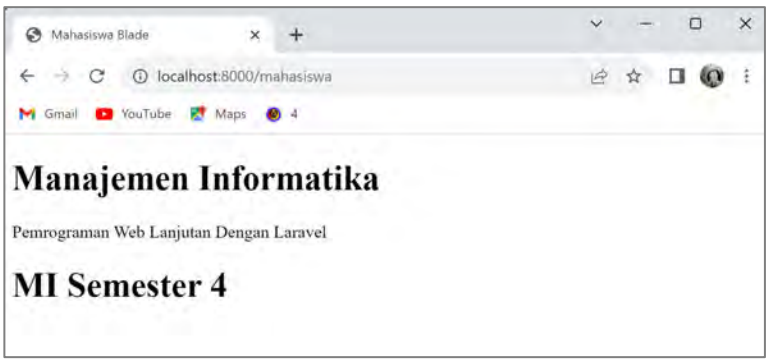
2. Kemudian tambahkan baris kode berikut ini paling bawah.

3. Langkah selanjutnya membuat function index dalam MahasiswaController yang sudah dibuat. Buka file MahasiswaController.php dan isikan seperti kode di bawah ini:

4. Perhatikan kode di atas, function index akan me-return view mahasiswa. Karena view mahasiswa belum ada, buat file view `mahasiswa.blade.php` pada folder `resource/views`, berikut isi dari file view mahasiswa.blade.php:



5. Simpan dan akses route santri menggunakan alamat `http://localhost:8000/mahasiswa` pada web browser, hasilnya seperti tampilan di bawah ini:



5.2. Passing Data Controller ke View

Dalam aplikasi web menggunakan framework Laravel, Passing Data dari Controller ke View adalah proses mengirimkan data dari controller ke view agar dapat ditampilkan kepada pengguna. Sehingga memungkinkan untuk memisahkan logika bisnis dari tampilan dan memberikan konten yang dinamis ke halaman web.

Passing data melibatkan proses mentransmisikan informasi dari controller ke view untuk ditampilkan. Data yang ditransmisikan mengacu pada data yang berada di dalam controller, yang kemudian diteruskan ke view. Dalam Laravel, pendekatan untuk mempassing atau mengirimkan data ke view dilakukan dengan menyertakan data dalam parameter kedua dari fungsi view().

Fungsi view() digunakan untuk memanggil tampilan yang diinginkan untuk disajikan. Data yang dimaksudkan untuk dipassing ditempatkan dalam parameter kedua dari fungsi view.

5.2.1. Memanggil View Dari Controller Laravel

Memanggil View dari Controller dalam Laravel adalah proses yang memungkinkan untuk menampilkan tampilan (HTML, CSS, JavaScript, maupun lainnya) kepada pengguna melalui respons dari controller. Dalam MVC (Model-View-Controller) pattern yang diadopsi oleh Laravel, controller bertanggung jawab untuk mengolah data dan logika bisnis, sedangkan view bertanggung jawab untuk menampilkan data kepada pengguna.

Berikut adalah langkah-langkah umum untuk memanggil view dari controller dalam Laravel:

1. Buka controller DosenController yang sudah dibuat sebelumnya, DosenController berada pada :
   `belajar_laravel/app/Http/Controller/DosenController.php.`

2. Buat view baru, dengan nama `biodata.blade.php`.

```blade.php
<!DOCTYPE html>
<html>
<head>
    <title>Passing Data Controller Ke View Laravel</title>
</head>
<body>

    <h1>Belajar Laravel - Manajemen Informatika</h1>
    <a href="https://admbisnis.poliban.ac.id/">
    www.manajemeninformatika.ac.id</a>

</body>
</html>
```

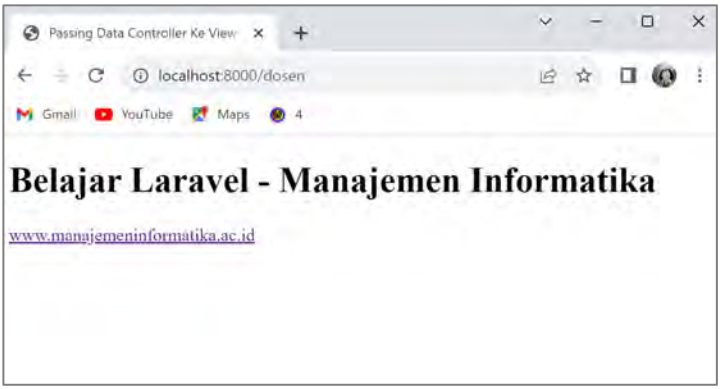
3. Pada method index dalam controller DosenController, panggil view `biodata.blade.php`.

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class DosenController extends Controller
{
    public function index(){
        return view('biodata');
    }
}
```

4. Buka file `web.php` pada folder routes, tambahkan baris kode berikut:

```php
    return view('welcome');
});
Route::get('dosen', function () {
    return view('dosen');
});

Route::get('mahasiswa', 'MahasiswaController@index');
Route::get('santri', [SantriController::class, 'index']);
Route::get('dosen', 'DosenController@index');
```

5. Simpan, dan buka hasilnya pada halaman : `localhost:8000/dosen`, maka hasilnya adalah :



Dengan memanggil view dari controller, dapat menyatukan logika bisnis dengan tampilan dan menyajikan konten yang dinamis kepada pengguna. Ini memungkinkan untuk memisahkan peran-peran dalam MVC dengan cara yang efisien dan terstruktur.

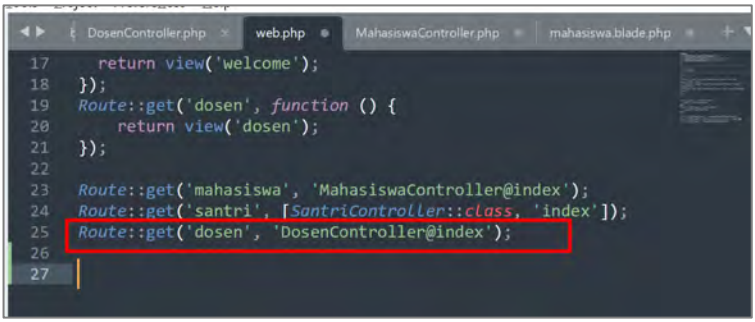
5.2.2. Passing Data Dari Controller Ke View Laravel

Passing data dari controller ke view dalam Laravel adalah proses mengirimkan dan menerima data dari bagian backend aplikasi (controller) ke bagian frontend yang menangani view. Sehingga memungkinkan untuk menampilkan informasi yang dinamis kepada pengguna di halaman web.

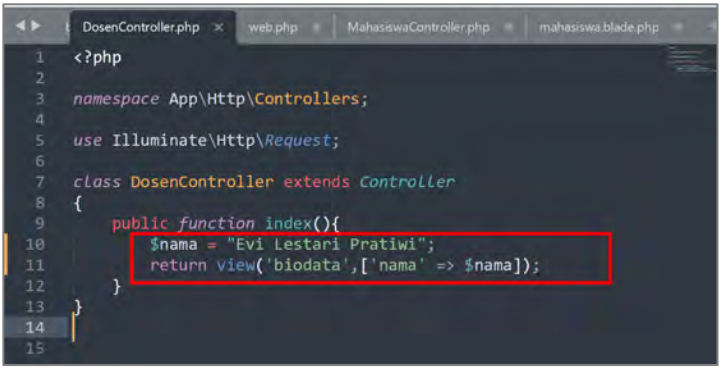
Dalam kerangka kerja Laravel, controller bertanggung jawab untuk mengatur logika bisnis, mengambil data dari database, dan mempersiapkannya untuk ditampilkan. Namun, view adalah tempat di mana data ini benar-benar ditampilkan kepada pengguna dalam bentuk halaman web.

Berikut adalah langkah-langkah umum untuk melewati data dari controller ke view dalam Laravel:

1. Definisikan route di file `routes/web.php`. Sebagai contoh, akan menggunakan controller DosenController yang sudah dibuat pada pembahasan sebelumnya.



2. Membuat controller. Jika controller yang akan digunanakan belum dibuat, maka buat controller dengan perintah artisan maupun menambahkan file pada folder controller (silakan lihat pada pembahasan sebelumnya). Dalam kasus ini, silakan buka file controller DosenController yang sudah dibuat sebelumnya. Ketikan kode berikut :

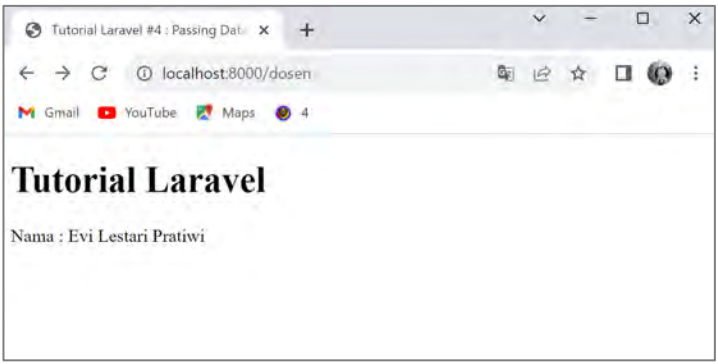


Pada kode program di atas, dibuat variable yang akan menyimpan nama “Evi Lestari Pratiwi”. Pada bagian `['nama' => $nama]` merupakan pengiriman variable nama.

3. Buat file view dengan nama yang sesuai dengan yang telah ditentukan di langkah sebelumnya. File view biasanya ditempatkan di direktori `resources/views`. File view yang digunakan pada kasus ini adalah `biodata.blade.php`, yang sudah dibuat pada pembahasan sebelumnya. Ketikan kode berikut :

```
biodata.blade.php | DosenController.php | dosen.blade.php | web.php | MahasiswaC
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Passing Data Controller Ke View Pada Laravel</title>
5  </head>
6  <body>
7      <h1>Tutorial Laravel</h1>
8      Nama: {{ $nama }}
9
10 </body>
11 </html>
```

4. Lihat hasilnya pada `localhost:8000/dosen`, seperti gambar berikut:



Itulah cara umum untuk melewati data dari controller ke view dalam Laravel menggunakan framework. Dengan ini, sehingga dapat dengan mudah menampilkan data dinamis kepada pengguna dalam tampilan aplikasi web.

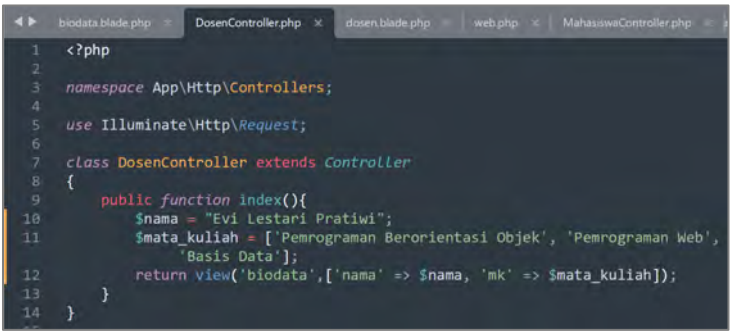
5.2.3. Passing Data Array Laravel

Passing data array dalam konteks Laravel mengacu pada proses mengirimkan dan menerima data dalam bentuk array dari bagian backend (controller) ke bagian frontend (view) aplikasi web. Memungkinkan untuk menampilkan data yang lebih kompleks dan struktur kepada pengguna di halaman web.

Berikut adalah langkah-langkah umum untuk melakukan passing data array dari controller ke view dalam Laravel:

1. Mengumpulkan data di Controller

   Mulailah dengan mengumpulkan data yang ingin dikirimkan dalam bentuk array di dalam method controller. Data ini bisa berasal dari berbagai sumber seperti database, API, atau berkas. Sebagai contoh buka file controller DosenController, kemudian ketikan kode berikut :

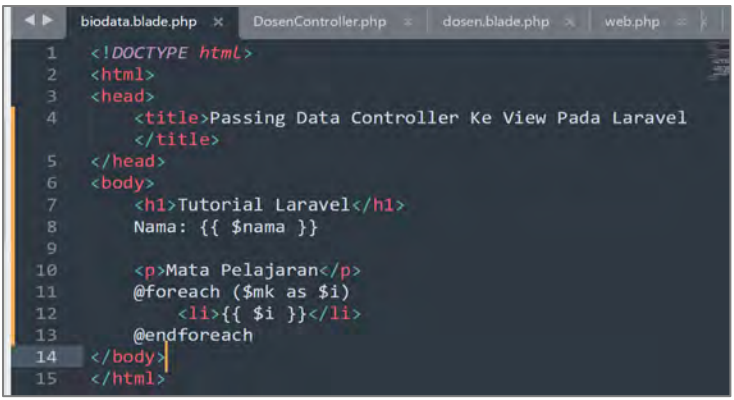


2. Mengirim data ke view

   Selanjutnya, kirimkan data array ke view menggunakan metode `view()` dan menyertakan array asosiatif dengan nama variabel sebagai kunci dan data array sebagai nilai.

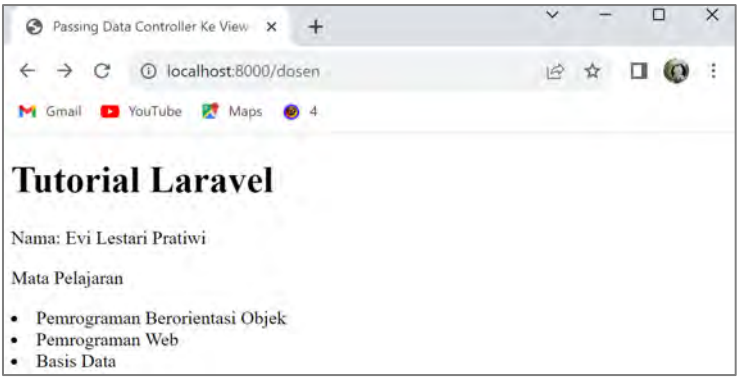
3. Menampilkan data di view

   Di dalam view, dapat mengakses data array tersebut menggunakan sintaks Blade. Buka `biodata.blade`, dan ketikan kode berikut :



Pada contoh di atas, menggunakan sintaks Blade `@foreach` untuk melakukan iterasi melalui setiap elemen dalam array dan menampilkannya di halaman.

4. Tampilkan Data Kepada Pengguna
   Data yang dikirimkan dari controller dan diolah di view akan ditampilkan kepada pengguna sebagai bagian dari halaman web. Silakan lihat hasilnya pada `localhost:8000/dosen`.



5.3. Request Data Pada Laravel

Dalam kerangka kerja Laravel, "Request" merujuk pada data yang diterima oleh aplikasi web melalui permintaan HTTP yang dibuat oleh klien (biasanya browser). Permintaan ini dapat berisi berbagai jenis data, seperti data formulir, data query, data header, data JSON, dan sebagainya. Laravel menyediakan mekanisme yang mudah untuk mengakses dan memanipulasi data permintaan ini melalui objek `Illuminate\Http\Request`.

Fitur pengambilan data dalam bahasa pemrograman PHP tentunya sudah tidak asing lagi, terutama dengan perintah-perintah GET dan POST. Perintah-perintah ini berfungsi untuk menerima dan mengambil data yang telah dikirimkan melalui formulir input atau melalui URL. Prinsip ini juga berlaku dalam pengembangan dengan menggunakan

kerangka kerja Laravel. Dalam konteks Laravel, istilah yang digunakan untuk pengambilan data adalah "Request".

Ada dua jenis perintah atau metode pengambilan data yang dapat digunakan dalam Laravel, yaitu:

1. Penerimaan data melalui URI
2. Penerimaan data melalui input

Ketika akan mengembangkan aplikasi menggunakan Laravel, akan sering mengandalkan perintah Request atau proses pengambilan data. Hal ini mencakup menerima dan mengambil data melalui URL, menangkap data dari inputan formulir seperti pada tahapan login, operasi CRUD, serta proses lain yang melibatkan manipulasi data.

5.3.1. Penerimaan data melalui URI

Pada Laravel, "Request Data dari URI" merujuk pada proses mengambil data yang dikirimkan melalui bagian URI (Uniform Resource Identifier) atau bagian path dari URL. URI adalah bagian dari URL yang mengidentifikasi sumber daya yang ingin diakses dalam aplikasi web. Contoh umum dari URI adalah bagian setelah nama domain, seperti `/produk/123` di mana "produk" adalah bagian dari controller atau route, dan "123" adalah parameter yang dikirim melalui URI.

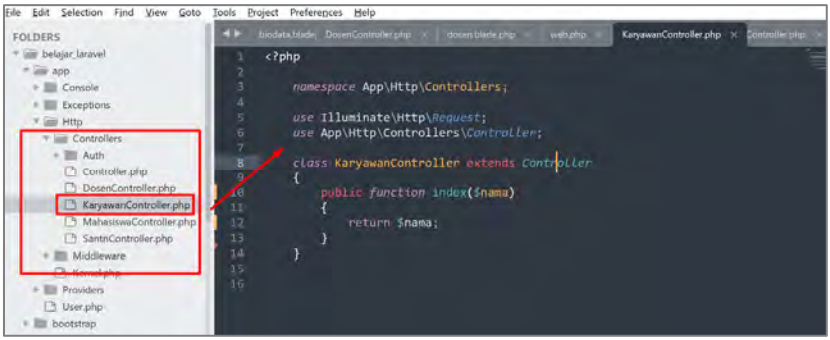
Untuk mengambil data dari URI dalam Laravel, dapat menggunakan objek `Illuminate\Http\Request`. Cara termudah adalah dengan mengakses parameter URI langsung dari objek `Request`. Berikut ini adalah langkah-langkahnya:

1. Mendefinisikan route

   Pertama, perlu mendefinisikan route dengan menggunakan placeholder (tanda kurung kurawal {}) untuk menunjukkan bahwa nilai akan diambil dari URI. Sebagai contoh, buka file `web.php` yang berada dalam folder `belajar_laravel/routes/web.php`. Buat route baru seperti gambar berikut :

2. Menerima Data di Controller

   Di dalam controller, ambil data dari URI dengan menggunakan metode `Route::parameter()` atau menginjeksi objek Request dan menggunakan metode `input()`. Sebagai contoh buka file controller KaryawanController yang telah dibuat pada pembahasan sebelumnya. Dan ketikan kode berikut :



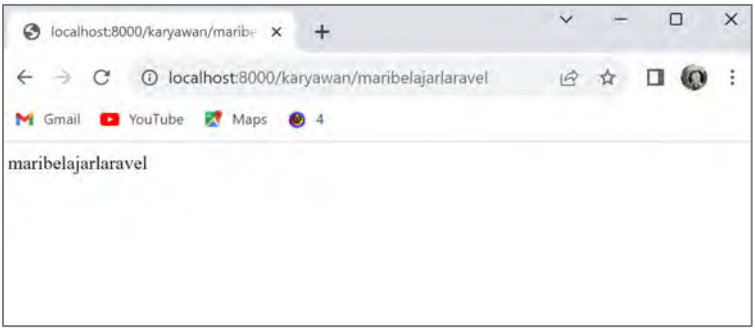
   Untuk mengambil data dari rute yang telah disebutkan sebelumnya (dengan placeholder {nama}), masukan `$nama` sebagai parameter dalam fungsi yang didefinisikan, seperti yang ditunjukkan dalam bagian `getNama($nama)`. Setelah itu, dapat mengembalikan nilai tersebut secara langsung menggunakan variabel yang sama, yaitu `$nama`.

3. Menampilkan Data

   Setelah mengambil data dari URI, dapat memprosesnya sesuai dengan kebutuhan, seperti mengambil data dari database atau

melakukan operasi bisnis lainnya. Sebagai contoh buka `localhost:8000/karyawan/maribelajarlaravel`.

Sehingga hasilnya :



Pengambilan data dari URI sangat berguna dalam berbagai skenario, seperti menampilkan detail produk berdasarkan ID, menampilkan profil pengguna berdasarkan nama pengguna, dan sebagainya. Dengan memahami cara mengambil data dari URI, sehingga dapat mengembangkan fitur yang lebih dinamis dan interaktif dalam aplikasi web yang dibuat.

5.3.2. Penerimaan Data Melalui Input

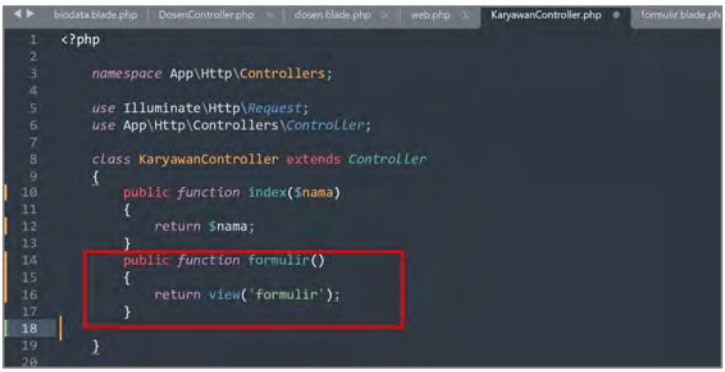
Mengambil data dari inputan (form input) dalam Laravel mengacu pada proses mengambil nilai-nilai yang dikirimkan oleh pengguna melalui formulir HTML dalam permintaan HTTP. Data ini dapat diambil menggunakan objek `Illuminate\Http\Request` atau dengan menggunakan fitur `"dependency injection"` yang disediakan oleh Laravel.

Berikut adalah langkah-langkah umum untuk mengambil data dari inputan dalam Laravel:

1. Buat route baru pada `routes/web.php`, seperti gambar dibawah :

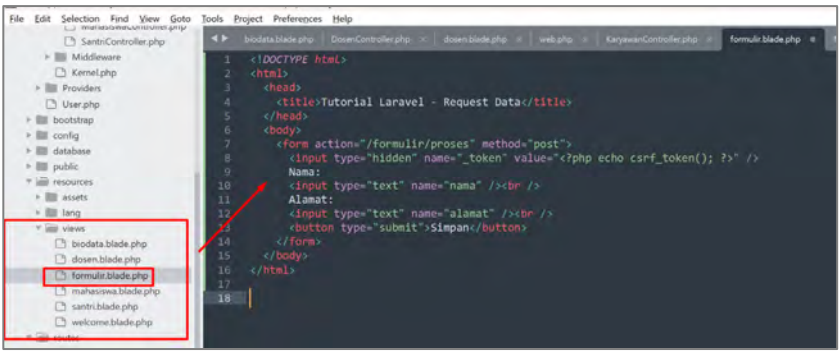
Di rute "/formulir", mengarahkan ke metode/fungsi "formulir" yang terdapat dalam KaraywanController dengan menggunakan metode `get()`. Sementara itu, di rute "/formulir/proses", mengarahkan ke metode "proses" dalam KaryawanController untuk mengambil data yang dikirimkan dari formulir dengan menggunakan metode `post()`. Metode ini menentukan bagaimana data akan diirimkan ke server.

2. Buatlah sebuah metode/fungsi bernama "formulir" dalam KaryawanController. Metode ini memanggil tampilan "formulir" sebagai antarmuka untuk formulir inputnya. Ketikan kode berikut pada KaryawanController :



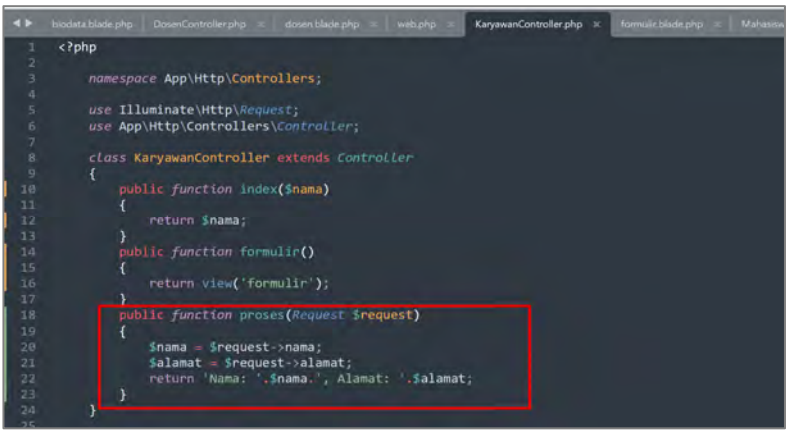
3. Kemudian, buatlah file formulir.blade.php yang mencakup elemen input dalam bentuk formulir dengan metode POST, dan tujuan formulirnya diarahkan ke "/formulir/proses" sesuai dengan routing yang telah kita buat sebelumnya. Berkas ini harus ditempatkan di

dalam direktori resources/views dengan nama `formulir.blade.php`. Kemudian ketikan kode berikut :



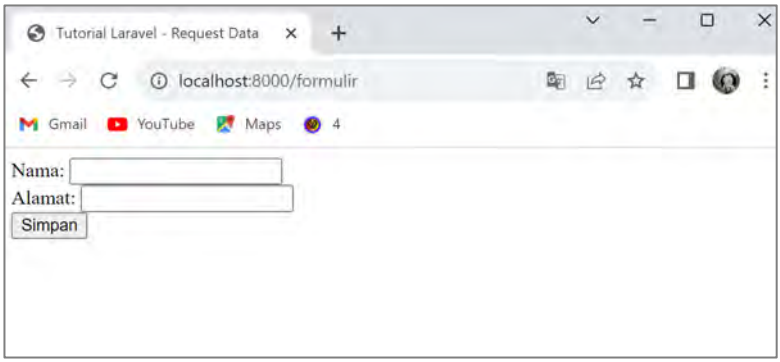
Pada kode program di atas, sebuah token CSRF telah ditambahkan. Cross-site request forgery, dikenal sebagai CSRF, merupakan jenis serangan yang dilakukan oleh pengguna yang tidak sah untuk menjalankan perintah tertentu. Dalam rangka mengatasi potensi ancaman ini, Laravel menyertakan CSRF Token. Token ini nantinya digunakan untuk melakukan verifikasi terhadap asal usul permintaan, memastikan bahwa permintaan tersebut berasal dari pengguna yang sah.

4. Buat method lagi yaitu `proses()`, pada KaryawanController, ketikan kode berikut :

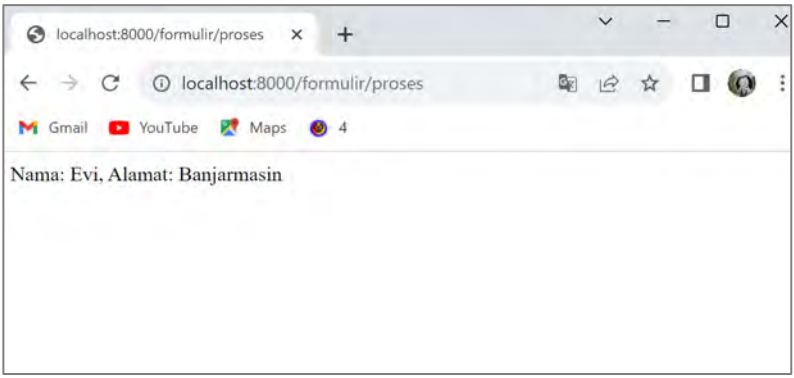


Untuk mengambil data dari URL, dapat langsung meneruskan variabel sebagai parameter dalam metodenya. Namun, ketika ingin mengambil data dari formulir, harus menggunakan kata kunci "Request" diikuti dengan variabel seperti $request sebagai parameter dalam metodenya. Variabel setelah kata kunci "Request" ini berfungsi untuk menyimpan data yang dikirim melalui formulir. Dengan cara ini, dapat menampilkan data dari formulir dengan merujuk ke `$request` diikuti dengan nama input di kolom formulir, seperti yang ditunjukkan dalam contoh kode di atas. Pada fungsi `input()`, sehingga harus menuliskan data yang ingin diambil (sesuai dengan nama formulir masing-masing).

5. Coba jalankan localhost:8000/formulir, sehingga hasilnya :



6. Kemudian isi, maka hasilnya :



5.4. Latihan

1. Apa peran utama dari controller dalam framework Laravel?
2. Bagaimana cara membuat controller baru di Laravel menggunakan perintah artisan?
3. Bagaimana cara mengirimkan data dari controller ke view dalam Laravel, dan bagaimana cara mengaksesnya di view?
4. Apa yang harus dilakukan jika Anda ingin menambahkan sebuah method baru ke dalam controller yang sudah ada dalam aplikasi Laravel?

BAB 6
KONFIGURASI DATABASE

Capaian Pembelajaran :

1. Mampu mengkonfigurasi dasar pada Laravel
2. Mampu memahami caching konfigurasi
3. Mampu memahami debug mode konfigurasi
4. Mampu memahami maintenance mode

Konfigurasi database merujuk pada pengaturan yang harus diatur dalam sebuah aplikasi atau proyek perangkat lunak untuk memungkinkan komunikasi dan interaksi dengan basis data. Dalam konteks pengembangan web dan kerangka kerja seperti Laravel, konfigurasi database mencakup informasi penting seperti tipe basis data yang digunakan, host server database, port, nama basis data, serta kredensial (*username* dan *password*) yang diperlukan untuk mengakses basis data.

6.1. Konfigurasi Dasar Pada Laravel

Dalam Laravel atau kerangka kerja lainnya, konfigurasi database adalah langkah awal yang harus dilakukan sebelum aplikasi dapat berinteraksi dengan basis data. Ini memastikan bahwa aplikasi memiliki informasi yang diperlukan untuk membuka koneksi ke basis data, melakukan query, mengambil dan menyimpan data, serta melaksanakan operasi lainnya yang melibatkan basis data.

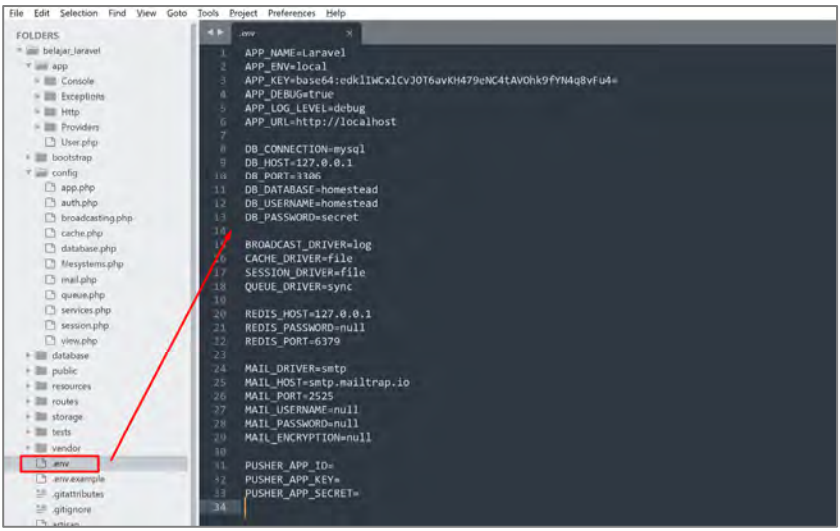
Konfigurasi database biasanya diatur dalam berkas konfigurasi atau berkas lingkungan yang dapat diubah sesuai kebutuhan. Pengaturan ini mencakup berbagai detail teknis yang diperlukan untuk menjembatani komunikasi antara aplikasi dan basis data.

Setelah menyelesaikan tahap instalasi Laravel, akan menemukan sebuah berkas yang dinamakan `.env`. Berkas ini terletak di direktori utama proyek Laravel.

Kandungan dalam berkas `.env` mengandung konfigurasi serta pengaturan-pengaturan yang diperlukan untuk proyek Laravel. Semua pengaturan yang relevan diintegrasikan ke dalam berkas `.env` ini.

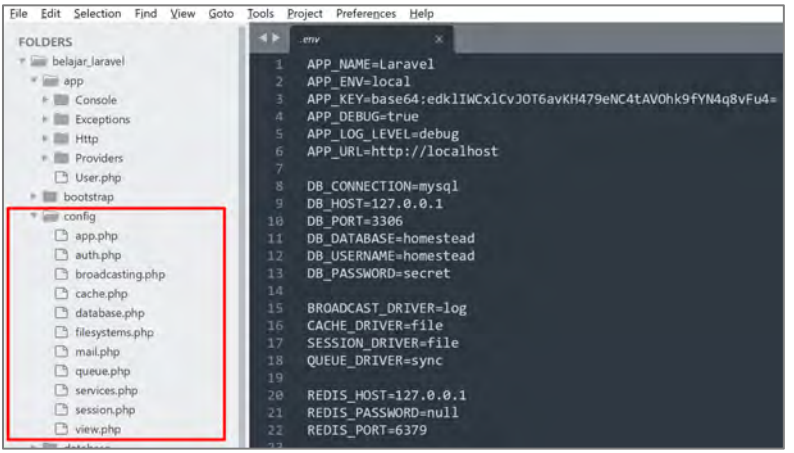
Berkas `.env` memiliki tujuan untuk mempermudah penyesuaian pengaturan sesuai kebutuhan proyek yang dikembangkan menggunakan Laravel.

Dibawah ini adalah isi baku dari berkas .env:



Seluruh pengaturan dalam Laravel terletak di dalam direktori "config" yang terbagi ke dalam beberapa berkas terpisah. Sebagai contoh, konfigurasi basis data di Laravel ditempatkan dalam berkas "database.php", pengaturan pengiriman email terletak di berkas "mail.php", sedangkan konfigurasi aplikasi ada dalam folder "app.php", bersama dengan berbagai pengaturan lainnya.

Berikut adalah isi dari direktori "config" dalam Laravel:



Dengan kehadiran berkas `.env`, semua pengaturan dapat dikonsolidasikan dalam satu berkas saja, menghindarkan kebutuhan untuk berpindah-pindah antar berkas saat ingin merubah pengaturan proyek Laravel. Sebagai contoh, jika pengaturan basis data "karyawan" didefinisikan dalam berkas config/database.php, dan juga "mahasiswa" diatur dalam berkas `.env`, maka yang akan digunakan adalah konfigurasi basis data "mahasiswa" dari berkas `.env`. Pengaturan dalam berkas `.env` diutamakan lebih tinggi daripada pengaturan dalam direktori config. Namun, jika berkas `.env` dihapus, pengaturan dari berkas dalam direktori config akan berlaku kembali.

Ada dua berkas `.env` dalam proyek, satu lagi bernama .env.example. Berkas .env.example sejatinya adalah salinan cadangan, digunakan jika perlu memulihkan pengaturan asli dari `.env`. Dengan adanya berkas example ini, apat mengembalikan pengaturan ke keadaan awal dengan merujuk isi dari `.env.example`.

Manfaat dari berkas .env adalah sebagai berikut:

1. Berkas .env berperan sebagai alat untuk mengonfigurasi Laravel.
2. Semua pengaturan yang relevan dengan proyek Laravel terkumpul dalam berkas .env.

3. Pengaturan yang terdapat dalam .env mendapatkan prioritas lebih tinggi dari sistem dibandingkan dengan pengaturan yang berada dalam folder konfigurasi.

6.2. Caching Konfigurasi

Untuk meningkatkan kecepatan memuat aplikasi yang dibuat, opsi untuk meng-cache semua pengaturan yang sudah dikonfigurasi menjadi satu berkas. Ini akan menggabungkan semua pengaturan konfigurasi ke dalam satu berkas, sehingga dapat diakses lebih cepat oleh kerangka kerja. Untuk melakukan hal ini, dapat menggunakan perintah `php artisan config:cache`.

Setelah perintah di atas dijalankan, berkas `.env` tidak akan lagi dipanggil, karena Laravel telah memuat hasil cache dari konfigurasi. Namun, disarankan untuk menggunakan konfigurasi ini hanya saat aplikasi yang dibangun telah mencapai status produksi, bukan pada tahap pengembangan di mana pengubahan berkas konfigurasi masih sering terjadi.

Jika perlu membersihkan cache, dapat menggunakan perintah `php artisan config:clear`.

6.3. Debug Mode

Dalam konfigurasi `.env` di Laravel, terdapat variabel `APP_DEBUG` yang secara default memiliki nilai true. Dalam fase pengembangan, disarankan untuk membiarkan nilai `APP_DEBUG` tetap true agar pesan-pesan kesalahan yang muncul selama tahap pengembangan dapat dilihat oleh pengembang. Namun, saat aplikasi siap untuk produksi, disarankan untuk mengubah nilai `APP_DEBUG` menjadi false agar informasi sensitif yang terdapat dalam konfigurasi tidak terlihat oleh pengguna yang mengakses aplikasi.

6.4. Maintenance Mode

Laravel memiliki keunggulan unik dibandingkan dengan kerangka kerja PHP lainnya, salah satunya adalah fitur mode pemeliharaan. Ketika mengembangkan aplikasi dengan Laravel, memiliki kemampuan untuk menjadikan status aplikasi sebagai mode pemeliharaan atau dalam tahap pengembangan. Dengan cara ini, pengunjung akan mendapat informasi bahwa aplikasi sedang berada dalam tahap pemeliharaan atau peningkatan.

Untuk mengaktifkan fitur mode pemeliharaan, dapat menggunakan salah satu dari perintah berikut melalui php artisan: `php artisan down`. Dengan menjalankan perintah tersebut, akan ditampilkan pemberitahuan bahwa aplikasi saat ini berada dalam mode pemeliharaan.

Untuk mematikan mode pemeliharaan, dapat mengetikkan perintah berikut: `php artisan up`. Dengan menjalankan perintah tersebut, akan ada pemberitahuan bahwa aplikasi telah kembali aktif. Sekarang, proyek telah berjalan normal kembali karena mode pemeliharaan di Laravel telah dinonaktifkan.

6.5. Latihan

1. Apa yang dimaksud dengan kelayakan ekonomi?
2. Apa yang harus Anda lakukan di dalam file .env untuk mengatur konfigurasi database dalam Laravel?
3. Di bagian mana dari file .env Anda akan menemukan pengaturan untuk konfigurasi database?
4. Apa yang dimaksud dengan DB_CONNECTION dalam konfigurasi database? Berikan contoh nilai yang mungkin digunakan.

BAB 7
SISTEM TEMPLATE BLADE LARAVEL

Capaian Pembelajaran :

1. Mampu memahami konsep pemodelan proses
2. Mampu mengetahui simbol-simbol dfd
3. Mampu membuat *context diagram*

Sistem template Blade dalam Laravel adalah cara untuk merancang antarmuka pengguna dalam aplikasi web secara terstruktur dan efektif. Blade adalah bahasa template khusus untuk Laravel yang membantu membuat tampilan yang dinamis dan konsisten.

7.1. Membuat *Template* Dinamis Dengan *Template* Blade Laravel

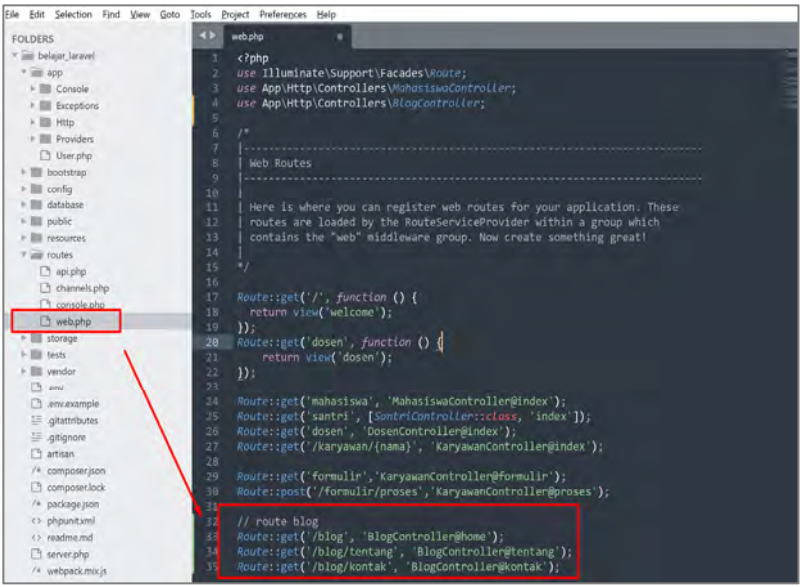
Belajar Sistem Blade Template dalam Laravel adalah proses memahami dan menggunakan mekanisme template yang disediakan oleh Laravel untuk membangun tampilan antarmuka pengguna dalam aplikasi web. Blade adalah bahasa template eksklusif yang diperkenalkan oleh Laravel, yang memungkinkan pengembang untuk membuat tampilan dengan cara yang efisien dan lebih terstruktur.

Dalam proses pembelajaran Sistem Blade Template dalam Laravel, akan merancang sebuah situs web sederhana yang terdiri dari tiga halaman dinamis: home, halaman tentang, dan halaman kontak.

Dikarenakan akan membuat tiga halaman tampilan yang berbeda, maka diperlukan tiga route yang berbeda pula untuk mengakses halaman-halaman tersebut.

Berikut Langkah – Langkah untuk membuat template dinamis dengan template blade Laravel :

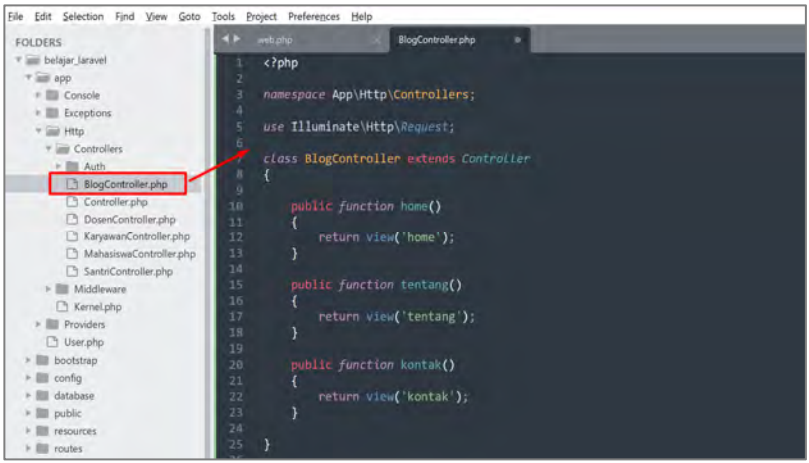
1. Buat tiga route baru dalam berkas `routes/web.php` :



Perhatikan rute-rute di atas, berikut ini penjelasannya:

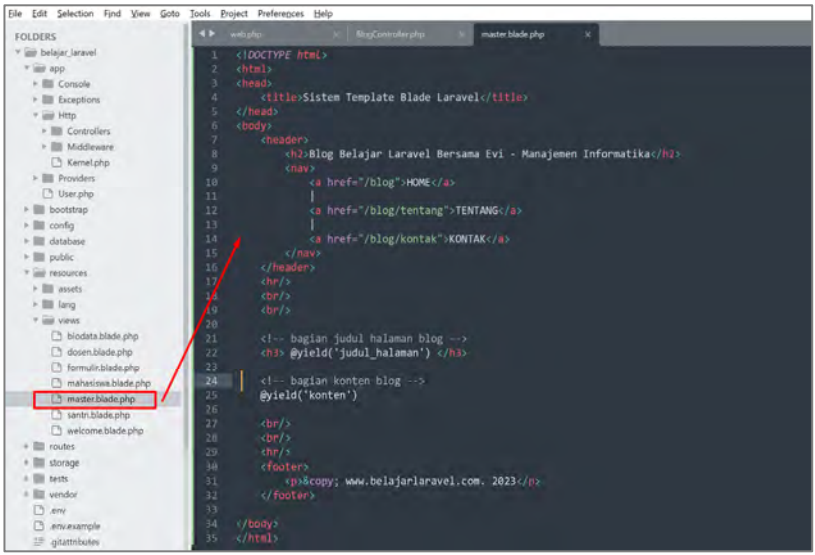
a. Ketika kita mengunjungi `route/`, maka yang akan dieksekusi adalah fungsi home yang terdapat dalam `controller BlogController`.
b. Ketika mengakses `route tentang`, maka yang akan dieksekusi adalah fungsi tentang yang terdapat dalam `controller BlogController`.
c. Apabila `route kontak` diakses, maka yang akan dieksekusi adalah fungsi kontak yang terdapat dalam `controller BlogController`.

2. Buat `controller BlogController.php` (membuat controller sudah ada pada pembahasan sebelumnya). Kemudian ketikan kode berikut pada `controller BlogController` :



a. Method home memanggil view home
b. Method tentang memanggil view tentang
c. Method kontak memanggil view kontak.

3. Buat template master yang merupakan template induk. Buat sebuah view baru di folder `resource/views` dengan nama `master.blade.php`, isinya sebagai berikut:

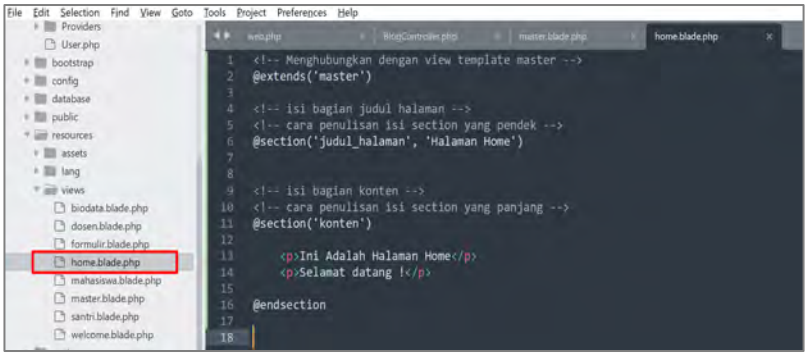


view `master.blade.php` sebagai template utama, di mana konten yang tetap seperti menu, footer, dan judul blog ditempatkan.

Dengan begitu, semua view yang dipanggil dalam fungsi-fungsi di `BlogController` akan dibuat dan dihubungkan dengan template view master.

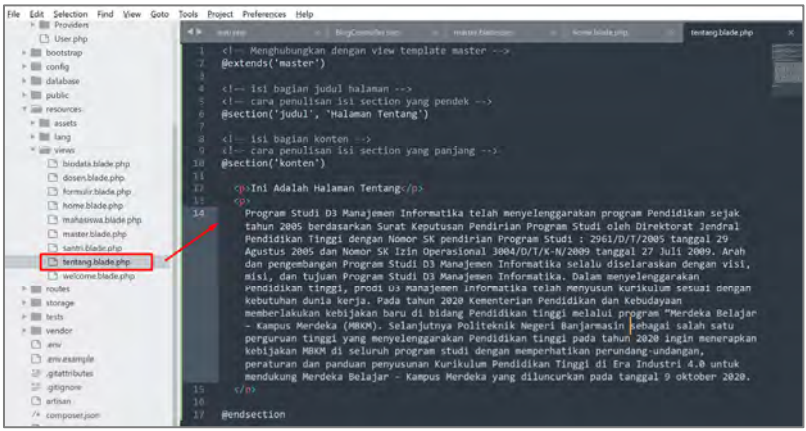
View `master.blade.php`: Di dalamnya terdapat pernyataan `yield()`. Fungsi `@yield()` digunakan untuk bagian-bagian tertentu pada tampilan situs web yang bersifat dinamis. Konten dari tampilan lain akan dimasukkan ke dalam parameter fungsi `@yield()`.

4. Buat 3 buah view baru dengan nama view home, tentang, dan kontak, pada folder `/resourse/views/`, `/resourse/views/`, `/resourse/views/`, ketikan kode berikut pada view home :



View `home.blade.php`: Terdapat fungsi dengan nama `@extends()` dan `@section()`. Fungsi `@extends()` berfungsi untuk menghubungkan view home dengan view master. Kemudian semua isi yang ada diantara fungsi `@section` pada view home akan di tampilkan pada bagian fungsi `@yield` pada view master.

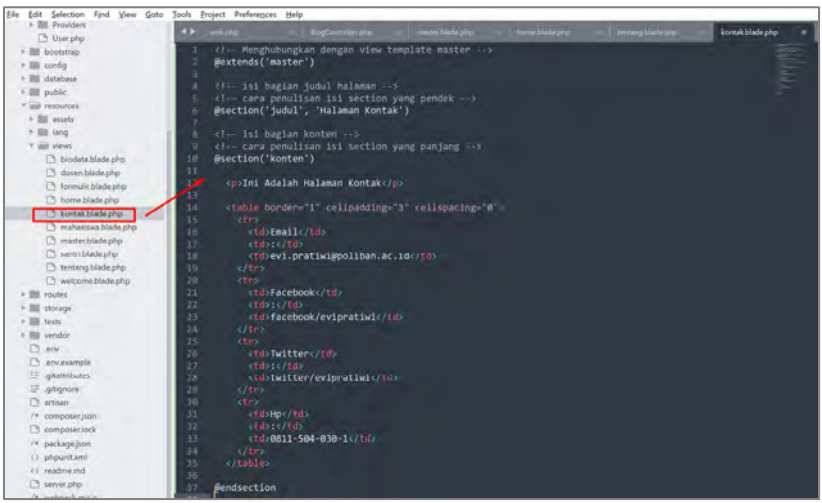
Ketikan kode berikut pada view tentang :



```
<!-- Menghubungkan dengan view template master -->
@extends('master')

<!-- isi bagian judul halaman -->
<!-- cara penulisan isi section yang pendek -->
@section('judul', 'Halaman Tentang')

<!-- isi bagian konten -->
<!-- cara penulisan isi section yang panjang -->
@section('konten')

  <p>Ini Adalah Halaman Tentang</p>
  <p>
    Program Studi D3 Manajemen Informatika telah menyelenggarakan program Pendidikan sejak
    tahun 2005 berdasarkan Surat Keputusan Pendirian Program Studi oleh Direktorat Jendral
    Pendidikan Tinggi dengan Nomor SK pendirian Program Studi : 2961/D/T/2005 tanggal 29
    Agustus 2005 dan Nomor SK Izin Operasional 3004/D/T/K-N/2009 tanggal 27 Juli 2009. Arah
    dan pengembangan Program Studi D3 Manajemen Informatika selalu diselaraskan dengan visi,
    misi, dan tujuan Program Studi D3 Manajemen Informatika. Dalam menyelenggarakan
    Pendidikan tinggi, prodi D3 manajemen informatika telah menyusun kurikulum sesuai dengan
    kebutuhan dunia kerja. Pada tahun 2020 Kementerian Pendidikan dan Kebudayaan
    memberlakukan kebijakan baru di bidang Pendidikan tinggi melalui program "Merdeka Belajar
    – Kampus Merdeka (MBKM). Selanjutnya Politeknik Negeri Banjarmasin sebagai salah satu
    perguruan tinggi yang menyelenggarakan Pendidikan tinggi pada tahun 2020 ingin menerapkan
    kebijakan MBKM di seluruh program studi dengan memperhatikan perundang-undangan,
    peraturan dan panduan penyusunan Kurikulum Pendidikan Tinggi di Era Industri 4.0 untuk
    mendukung Merdeka Belajar - Kampus Merdeka yang diluncurkan pada tanggal 9 oktober 2020.
  </p>

@endsection
```

View `tentang.blade.php`: Terdapat fungsi dengan nama `@extends()` dan `@section()`. Fungsi `@extends()` berfungsi untuk menghubungkan view tentang dengan view master. Kemudian semua isi yang ada diantara fungsi `@section` pada view tentang akan di tampilkan pada bagian fungsi `@yield` pada view master.

Ketikan kode berikut pada view kontak :

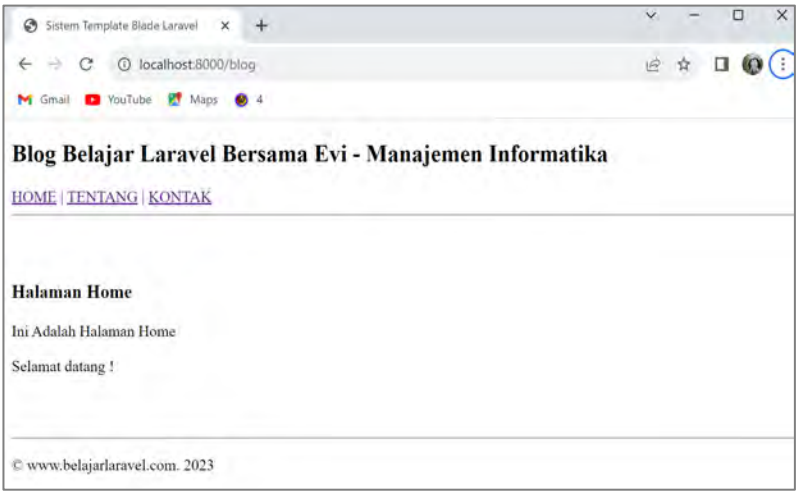


```
<!-- Menghubungkan dengan view template master -->
@extends('master')

<!-- isi bagian judul halaman -->
<!-- cara penulisan isi section yang pendek -->
@section('judul', 'Halaman Kontak')

<!-- isi bagian konten -->
<!-- cara penulisan isi section yang panjang -->
@section('konten')

  <p>Ini Adalah Halaman Kontak</p>

  <table border="1" cellpadding="3" cellspacing="0">
    <tr>
      <td>Email</td>
      <td>:</td>
      <td>evi.pratiwi@poliban.ac.id</td>
    </tr>
    <tr>
      <td>Facebook</td>
      <td>:</td>
      <td>facebook/evipratiwi</td>
    </tr>
    <tr>
      <td>Twitter</td>
      <td>:</td>
      <td>twitter/evipratiwi</td>
    </tr>
    <tr>
      <td>Hp</td>
      <td>:</td>
      <td>0811-504-030-1</td>
    </tr>
  </table>

@endsection
```

View `kontak.blade.php`: Terdapat fungsi dengan nama `@extends()` dan `@section()`. Fungsi `@extends()` berfungsi untuk

menghubungkan view kontak dengan view master. Kemudian semua isi yang ada diantara fungsi `@section` pada view kontak akan di tampilkan pada bagian fungsi `@yield` pada view master.

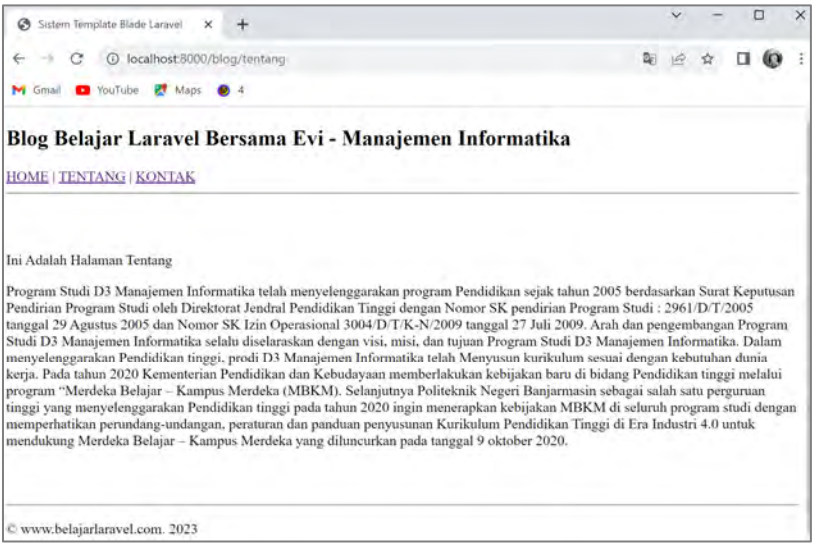
5. Silakan coba hasilnya dengan mengakses : localhost:8000/blog, maka hasilnya :



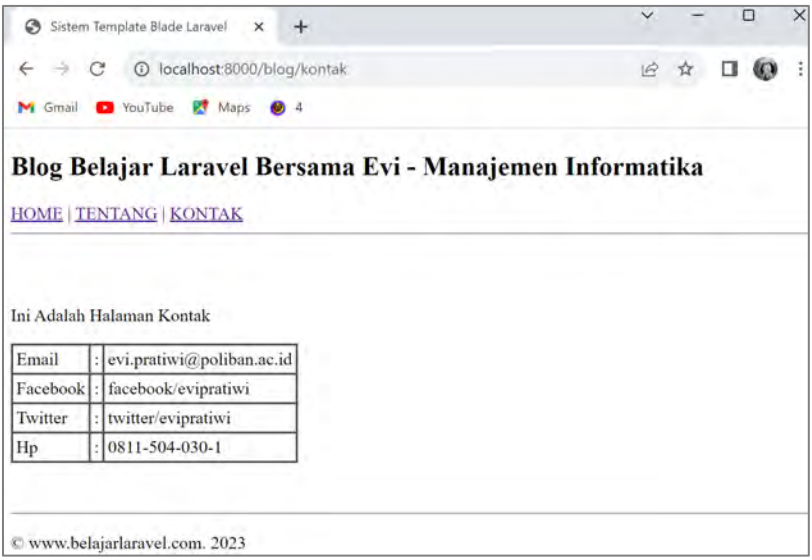
Lihatlah pada ilustrasi di atas, ketika `route'/'` diakses, function home pada `BlogController` akan dieksekusi, yang kemudian akan memanggil `view home.blade.php`.

Hal yang serupa terjadi pada halaman lain. Cobalah klik menu kontak dan tentang. Tampilan view master akan tetap dipertahankan, namun yang berubah adalah bagian judul dan isi konten sesuai halaman yang dipilih.

Contoh tampilan halaman Tentang seperti berikut:



Contoh tampilan halaman Kontak seperti berikut :



7.2. Latihan

1. Apa itu sistem template Blade dalam kerangka kerja Laravel, dan apa manfaatnya bagi pengembangan aplikasi web?
2. Bagaimana cara menggunakan file template Blade dalam Laravel untuk membuat tampilan yang konsisten di seluruh situs web?
3. Apa perbedaan utama antara file template Blade .blade.php dengan file .php biasa dalam konteks Laravel?
4. Bagaimana Anda mengirim data dari kontroler ke tampilan Blade, dan bagaimana cara menampilkannya di dalam tampilan?
5. Jelaskan penggunaan @extends dan @yield dalam Blade template. Bagaimana Anda menggunakan halaman induk (master) dan halaman anak dalam Blade?

BAB 8
CRUD

Capaian Pembelajaran :

1. Mampu menampilkan data
2. Mampu menambah data
3. Mampu memperbarui data
4. Mampu menghapus data

CRUD adalah singkatan dari Create, Read, Update, dan Delete. Ini adalah empat operasi dasar yang digunakan dalam manajemen data pada aplikasi web atau sistem basis data. Laravel, sebagai kerangka kerja PHP yang kuat, menyediakan dukungan lengkap untuk mengimplementasikan operasi CRUD dengan mudah.

Create (Membuat): Ini melibatkan penciptaan entitas baru dalam basis data. Dalam Laravel, dapat membuat formulir input menggunakan HTML dan kemudian menangani data yang di-submit melalui kontroler. Setelah itu, dapat menggunakan model Eloquent untuk menyimpan data baru ke dalam basis data.

Read (Membaca): Ini melibatkan penarikan data dari basis data dan menampilkannya kepada pengguna. Dalam Laravel, dapat menggunakan model Eloquent untuk mengambil data dari basis data dan kemudian menampilkannya melalui tampilan menggunakan Blade template.

Update (Memperbarui): Ini melibatkan memperbarui data yang sudah ada dalam basis data. Dalam Laravel, dapat membuat formulir pengeditan yang memuat data yang sudah ada dan kemudian menangani pembaruan data melalui kontroler. Model Eloquent juga dapat digunakan untuk menyimpan perubahan ini kembali ke basis data.

Delete (Menghapus): Ini melibatkan penghapusan data dari basis data. Dalam Laravel, dapat menggunakan tindakan penghapusan dalam kontroler untuk menghapus data dari basis data menggunakan model Eloquent.

Laravel menyediakan alat yang kuat seperti Eloquent ORM, routing, kontroler, dan Blade templating untuk memudahkan implementasi operasi CRUD. Hal ini memungkinkan pengembang untuk mengelola data dengan lebih terstruktur dan efisien dalam aplikasi web yang dibangun dengan Laravel.

8.1. Membuat CRUD

8.1.1. Membuat CRUD Pada Laravel Dengan Query Builder

Membuat operasi CRUD (Create, Read, Update, Delete) pada Laravel menggunakan Query Builder melibatkan penggunaan fitur Query Builder untuk memanipulasi data dalam basis data. Query Builder adalah sebuah antarmuka yang disediakan oleh Laravel yang memungkinkan untuk membuat dan menjalankan kueri basis data menggunakan sintaks berantai (method chaining) yang mudah dibaca.

Query Builder adalah sebuah fitur yang disediakan oleh Laravel yang memungkinkan untuk membangun dan menjalankan kueri basis data menggunakan metode-metode berantai (method chaining) yang mudah dibaca dan dikelola. Dengan Query Builder, dapat dibuat query SQL secara dinamis dalam bentuk pemanggilan berantai yang lebih mudah dipahami daripada menulis query SQL langsung.

Query Builder Laravel mengabstraksi query basis data dan menyediakan antarmuka yang lebih eksplisit dan terstruktur untuk berinteraksi dengan basis data. Sehingga membantu menghindari masalah kesalahan sintaksis, meningkatkan keamanan, dan memudahkan dalam pengembangan.

Contoh penggunaan Query Builder dalam Laravel:

```
$users = DB::table('users')
```

```
    ->where('name', 'John')
    ->orWhere('name', 'Jane')
    ->get();
```

Dalam contoh di atas, kita menggunakan Query Builder untuk mengambil semua data pengguna dari tabel 'users' di mana namanya adalah 'John' atau 'Jane'.

Fitur Query Builder termasuk berbagai metode seperti `select`, `where`, `orderBy`, `join`, dan banyak lagi, yang memungkinkan untuk membangun kueri kompleks tanpa harus menulis kueri SQL mentah. Fitur Query Builder membantu kode menjadi lebih terbaca dan mudah dikelola.

8.1.2. Pengaturan Database Dalam Laravel

Pengaturan database dalam Laravel merujuk pada konfigurasi yang harus ditentukan agar aplikasi Laravel dapat berinteraksi dengan basis data yang sesuai. Pengaturan database ini biasanya disimpan dalam berkas .env atau dalam konfigurasi lainnya dalam folder `config`.

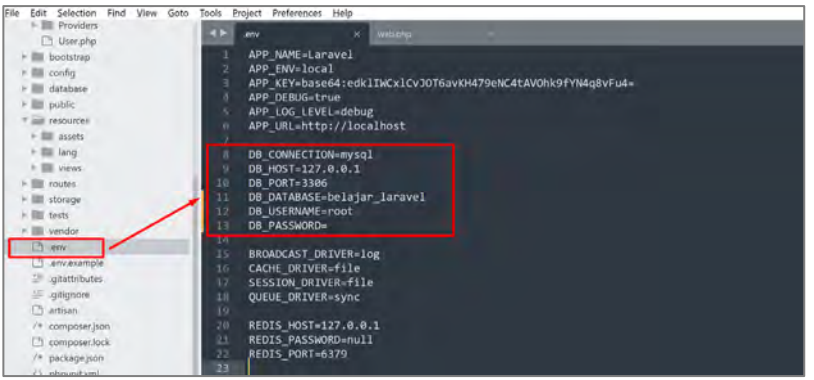
Sebagai contoh kasus dalam membut CRUD yaitu untuk karyawan, sehingga akan dibuat beberapa fungsi, yaitu :

1. Menampilkan data dari database dengan Laravel
2. Menambah data ke database dengan Laravel
3. Memperbarui data pada database dengan Laravel
4. Menghapus data dari database dengan Laravel.

Berikut langkah-langkah umum dalam mengatur database dalam Laravel:

1. Buka Berkas `.env`: Berkas `.env` adalah tempat utama di mana dapat mengatur parameter koneksi database. Berkah `.env` dapat ditemukan di direktori utama proyek Laravel. Jika tidak menemukan file .env, silakan atur koneksi database pada `belajar_laravel/config/database.php`.

2. Tentukan Tipe Database: Gunakan parameter `DB_CONNECTION` untuk menentukan tipe database yang akan di gunakan, seperti MySQL, PostgreSQL, SQLite, atau lainnya.
3. Konfigurasi Koneksi: Parameter seperti `DB_HOST`, `DB_PORT`, `DB_DATABASE`, `DB_USERNAME`, dan `DB_PASSWORD` digunakan untuk mengonfigurasi rincian koneksi ke basis data. Isi sesuai dengan informasi yang diberikan oleh penyedia hosting atau pengaturan lokal.
4. Optimasi dan Pengaturan Tambahan: Selain pengaturan dasar, Laravel juga memungkinkan untuk mengatur opsi-opsi tambahan seperti karakter set dan collation, penggunaan SSL, pool koneksi, dan lainnya melalui konfigurasi dalam berkas `config/database.php`.
5. Sebagai contoh, buat pengaturan database dalam file `.env`, seperti gambar dibawah, sesuaikan nama database yang akan digunakan, username, dan password mysql nya.



8.1.3. Mempersiapkan Database dan Tabel

Mempersiapkan database dan tabel adalah langkah awal yang penting dalam pengembangan aplikasi web menggunakan Laravel atau bahkan dalam pengembangan aplikasi apapun yang berinteraksi dengan basis data. Langkah-langkah ini melibatkan pembuatan basis data dan

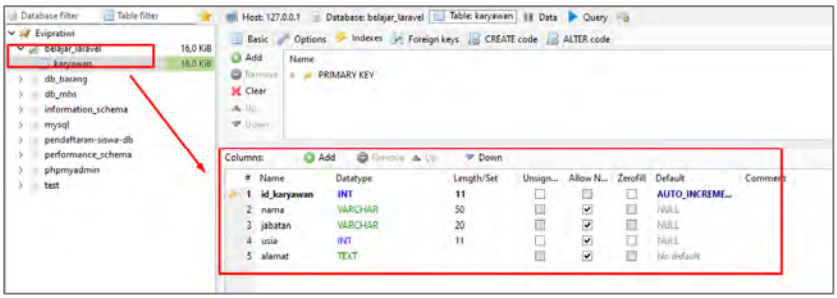
tabel yang diperlukan untuk menyimpan data aplikasi yang ingin dibuat.

Berikut adalah langkah-langkah umum dalam mempersiapkan database dan tabel dalam pengembangan menggunakan Laravel: (kasus : membuat table karyawan)

1. Membuat Database: Buat basis data kosong di server database yang akan digunakan, seperti `MySQL`, `PostgreSQL`, atau `SQLite`. Dalam kasus ini dibuat database dengan nama : belajar_laravel, kemudian buat table dengan nama karyawan, dengan ketentuan berikut :

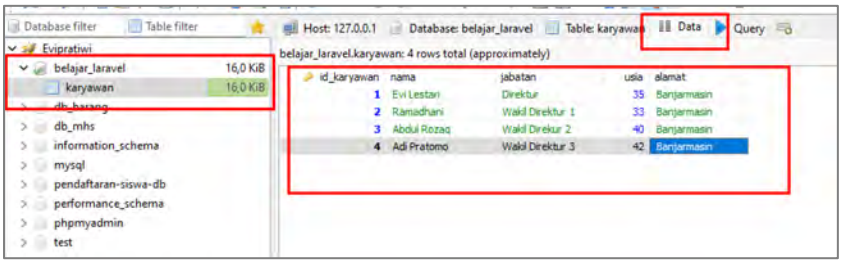
No	Field	Type	Length	Keterangan
1	Id_karyawan	Int	11	Auto Increment, primary key
2	Nama	Varchar	50	
3	Jabatan	Varchar	20	
4	Usia	Int		
5	Alamat	Text		

Sehingga hasilnya menjadi :



2. Konfigurasi `.env`: konfigurasikan pengaturan koneksi database dalam berkas `.env` di proyek Laravel. Ini melibatkan mengisi nilai-nilai seperti `DB_CONNECTION`, `DB_HOST`, `DB_PORT`, `DB_DATABASE`, `DB_USERNAME`, dan `DB_PASSWORD` sesuai dengan detail koneksi basis data. (sudah dibuat pada pembahasan sebelumnya).

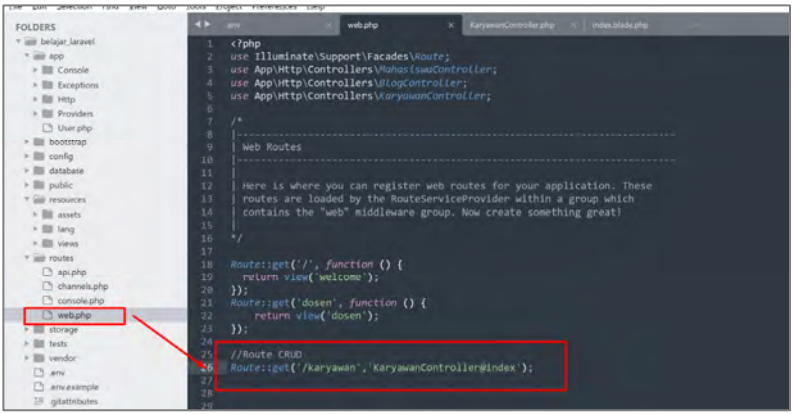
3. Migrasi Tabel: Dalam Laravel, dapat menggunakan migrasi untuk membuat dan mengelola struktur tabel. Migrasi adalah file PHP yang berisi kode untuk membuat tabel, menambahkan kolom baru, mengubah kolom, atau melakukan perubahan skema lainnya. Migrasi table dapat menggunakan perintah `php artisan make:migration` untuk membuat migrasi baru, dan `php artisan migrate` untuk menjalankan migrasi.
4. Mengisi Data Awal (Opsional): Jika diperlukan data awal dalam table, memasukkan data awal ke dalam tabel menggunakan model dan Query Builder. Sebagai contoh, silakan isi beberapa data ke dalam table karyawan, seperti gambar berikut :



8.2. Menampilkan Data

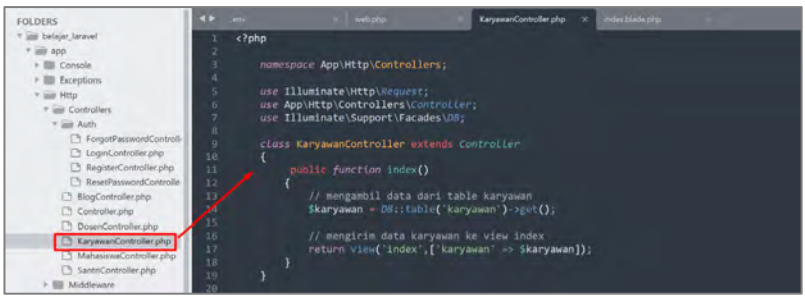
Dalam Laravel, dapat dengan mudah menampilkan data dari database ke tampilan menggunakan model Eloquent, kontroler, dan tampilan Blade. Berikut adalah langkah-langkah umum untuk menampilkan data dari database dengan Laravel:

1. Buat route dengan Alamat ‘/karyawan’, pada `belajar_laravel/routes/web.php` seperti berikut :



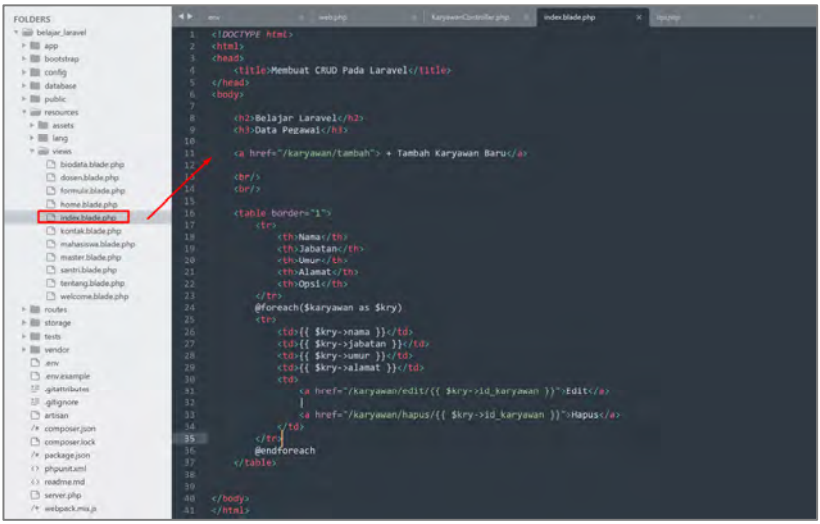
Method `index()` dijalankan pada Controller `KaryawanController` pada saat route `'/karyawan'` diakses.

2. Buat controller `KaryawanController.php` pada folder `belajar_laravel/app/Http/Controllers/KaryawanController.php`, dan ketikan kode berikut :

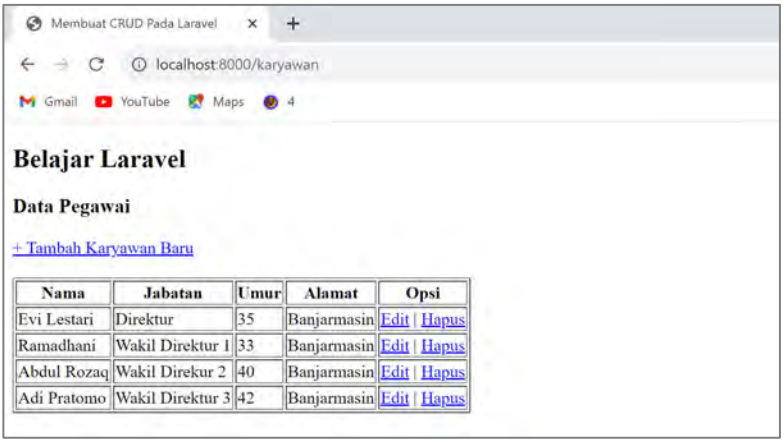


Fungsi `$karyawan = DB::table('karyawan')->get();` berfungsi untuk mengambil data dari table yang dipilih. Fungsi `return view('index',['karyawan' => $karyawan]);` berfungsi untuk mengirim data ke view agar dapat ditampilkan.

3. Buat view index, yang berada pada folder `belajar_laravel/resource/view/index.blade.php`, dan ketikan kode berikut :



4. Simpan, kemudian jalankan pada : localhost:8000/karyawan, maka hasilnya adalah :



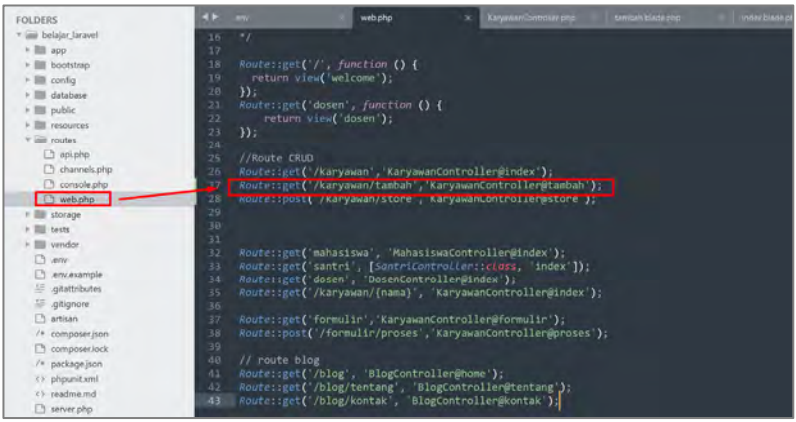
8.3. Tambah Data

Dalam konteks pengembangan web menggunakan framework Laravel, menambahkan data mengacu pada proses menyimpan informasi baru ke dalam basis data menggunakan model Eloquent. Eloquent adalah bagian dari Laravel yang memungkinkan dapat

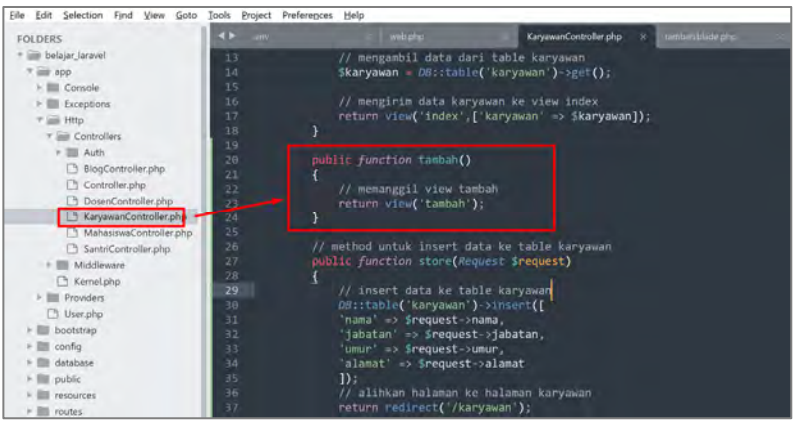
berinteraksi dengan basis data menggunakan objek-objek PHP, membuat proses manipulasi basis data menjadi lebih intuitif.

Berikut adalah langkah-langkah umum untuk menambahkan data ke dalam basis data menggunakan Laravel:

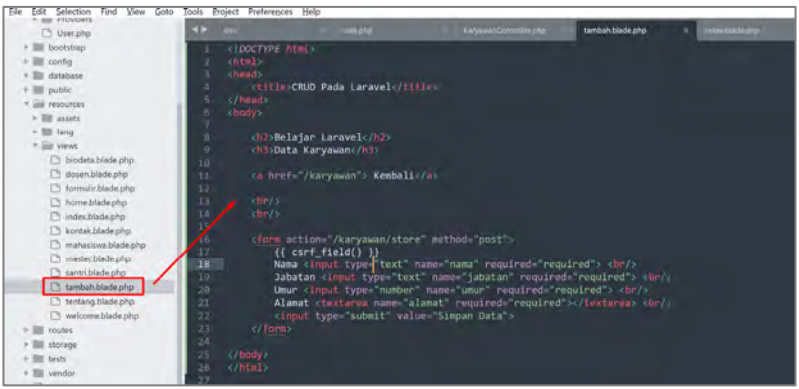
1. Buat route dengan Alamat `'/karyawan/tambah'`, pada `belajar_laravel/routes/web.php` seperti berikut :



2. Buat method tambah pada controller `KaryawanController.php`, seperti gambar berikut :



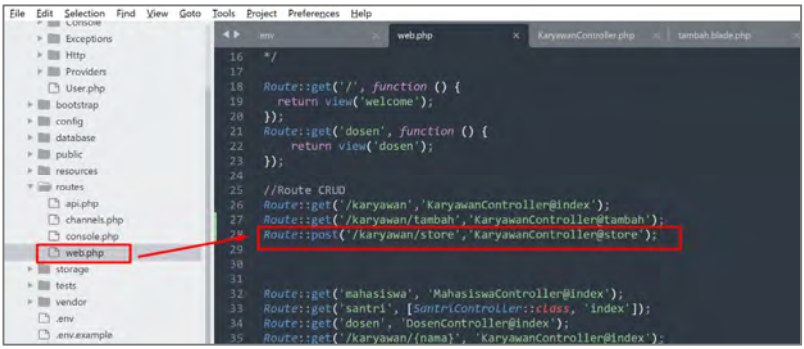
3. Buat view `tambah.blade.php`, yang berada pada folder `belajar_laravel/resource/view/tambah.blade.php`, dan ketikan kode berikut :



Amati bagian dalam file tambah.blade.php yang telah dibuat, dimana formulir diarahkan menuju route `'/karyawan/store'`. Nantinya, route `'karyawan/store'` akan mengurus proses pengolahan data yang diinput agar dapat dihandle oleh controller.

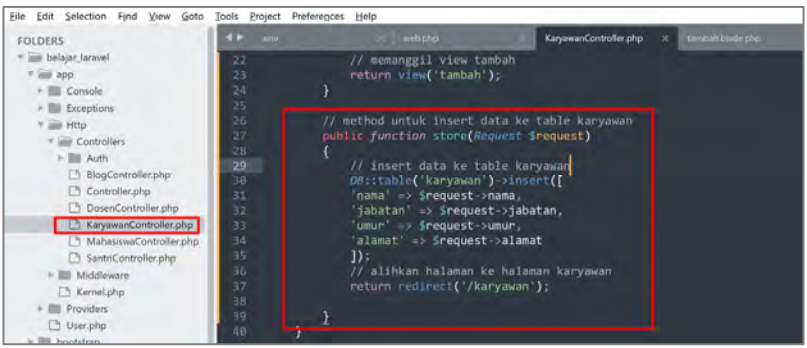
Terdapat fitur perlindungan CSRF (Cross-Site Request Forgery) yang berfungsi untuk mencegah masukan data yang berasal dari luar aplikasi atau sistem. Dengan menambahkan perintah `{{csrf_field()}}`, Laravel secara otomatis akan menghasilkan kode token otomatis yang ditempatkan dalam bentuk form tersembunyi.

4. Buat route baru, yaitu route `‘/karyawan/store’`, pada folder `belajar_laravel/routes/web.php`, dan ketikan kode berikut :



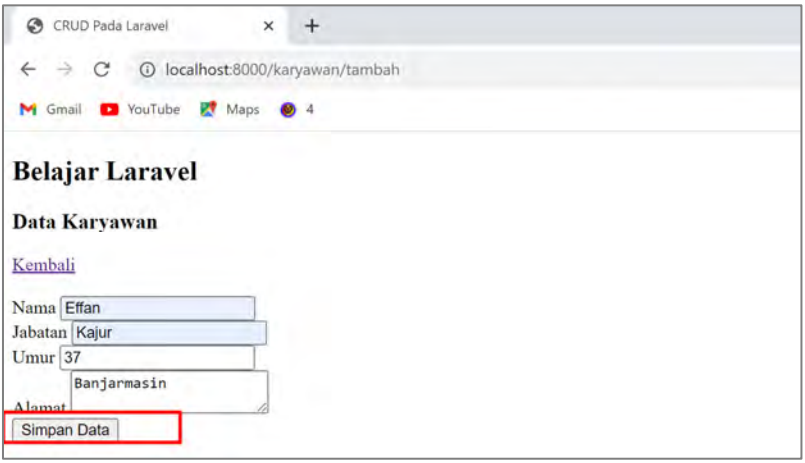
Pada route yang baru dibuat, menggunakan method "post", karena mengirimkan data melalui form ke route 'karyawan/store', sehingga wajib menggunakan method post pada route-nya.

5. Buat method 'store' pada controller KaryawanController.php, dan ketikan kode berikut :

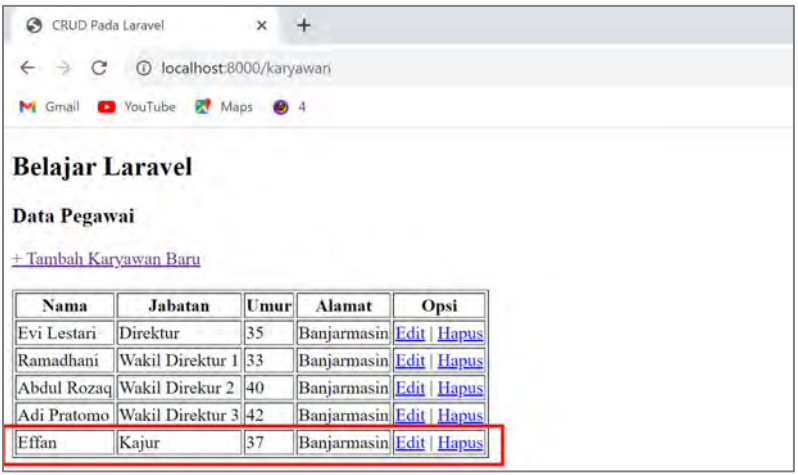


Fungsi table() untuk memberitahu nama table, fungsi insert() bertujuan untuk menambahkan data dan menetapkan data apa saya yang ingin ditambahkan.

6. Simpan dan jalankan localhost:8000/karyawan/tambah, dan hasilnya adalah :



Ketika tombol Simpan Data diklik, maka hasilnya bertambah pada database, seperti gambar berikut :



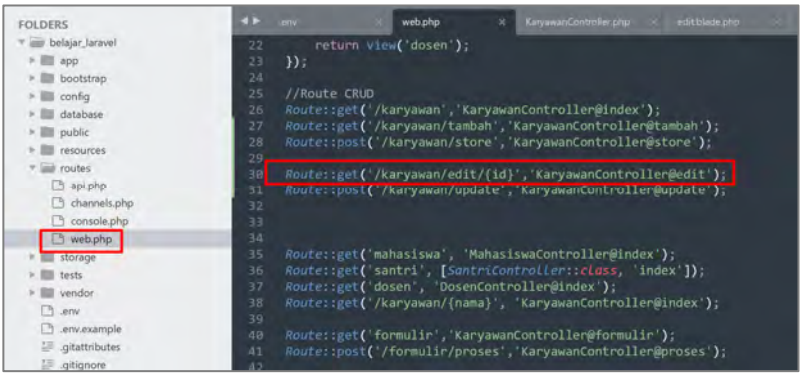
8.4. Perbarui Data

Dalam konteks pengembangan web menggunakan framework Laravel, memperbarui data mengacu pada proses mengubah atau mengedit informasi yang ada dalam basis data menggunakan model Eloquent. Eloquent adalah komponen Laravel yang memungkinkan

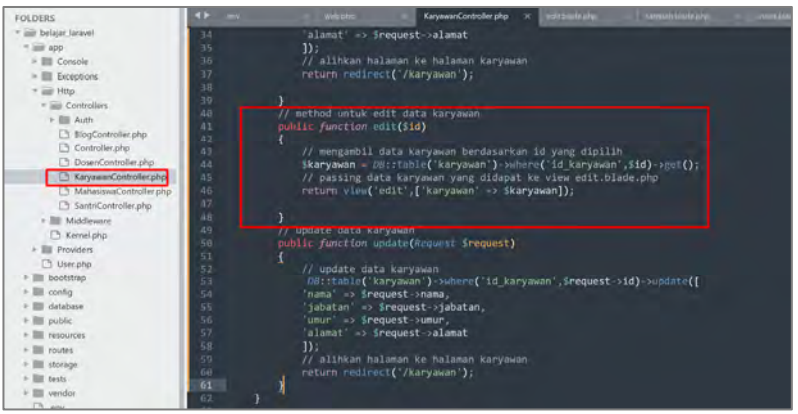
berinteraksi dengan basis data melalui objek-objek PHP, membuat proses manipulasi basis data menjadi lebih mudah dipahami.

Berikut adalah langkah-langkah umum untuk memperbarui data dalam basis data menggunakan Laravel:

1. Buat route dengan Alamat `'/karyawan/edit'`, pada `belajar_laravel/routes/web.php` seperti berikut :



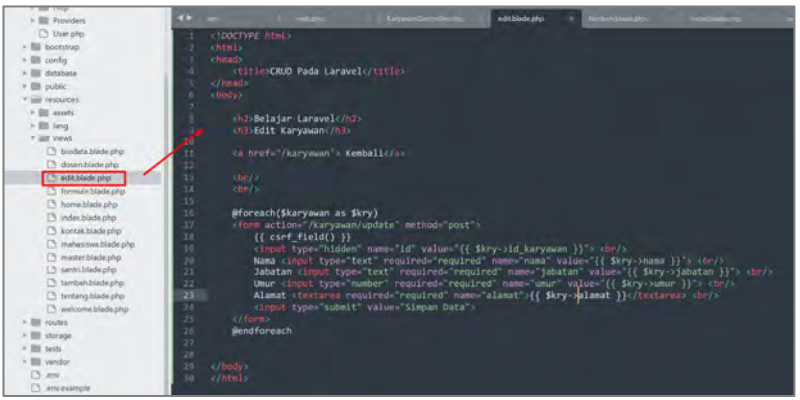
2. Buat method edit `KaryawanController.php`, seperti gambar berikut :



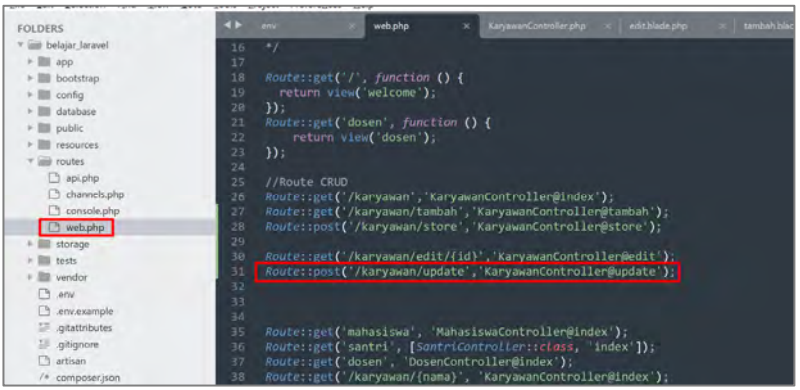
Pada baris 44, berfungsi untuk mengambil data karyawan dari database dengan menggunakan query `$karyawan = DB::table('karyawan')->where('id_karyawan',$id)->get();` seleksi data yang ingin diambil dengan menyeleksi karyawan_id

pada `where('id_karyawan',$id)`. Kemudian mempassing data dengan menggunakan return `view('edit',['karyawan' => $karyawan])`;

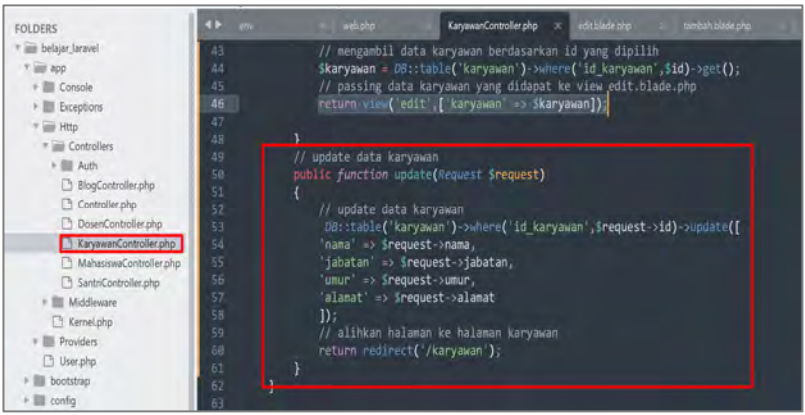
3. Buat view baru dengan nama `edit.blade.php`, yang berada pada folder `belajar_laravel/resource/view/edit.blade.php`, dan ketikan kode berikut :



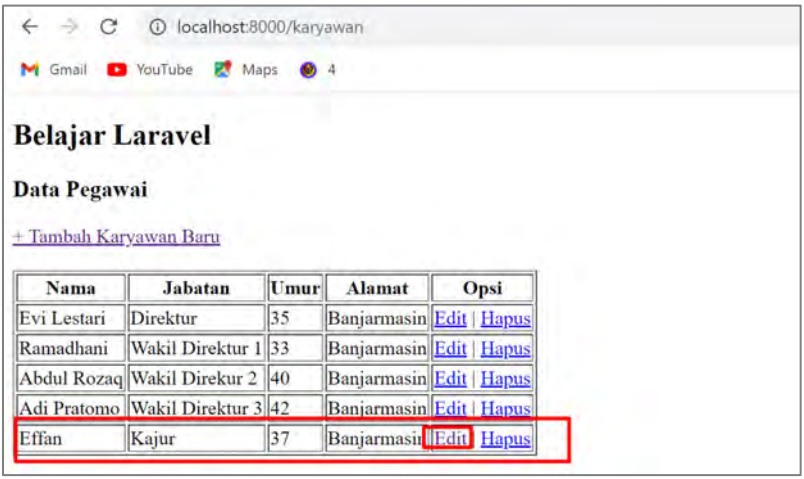
4. Untuk menghandle data dari form edit karyawan, masukan kode berikut pada `belajar_laravel/routes/web.php` :

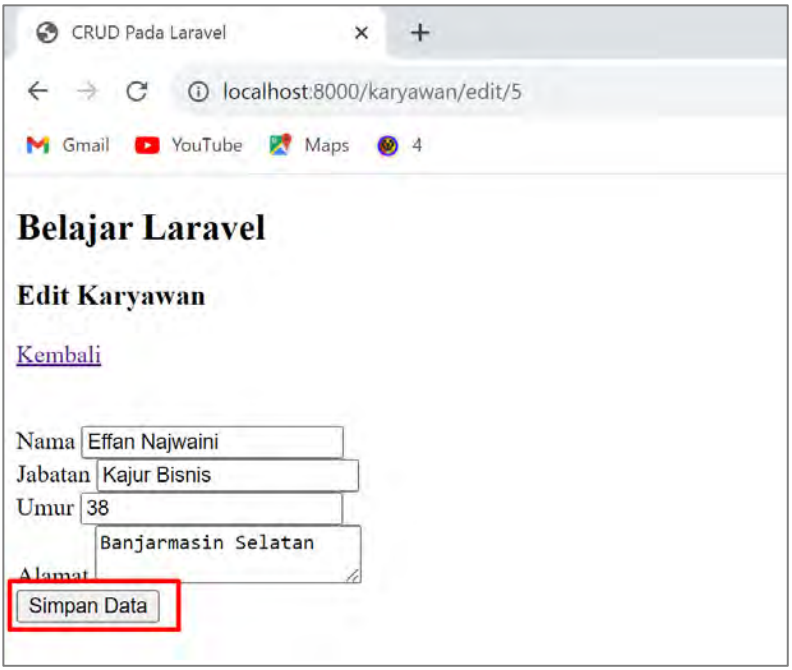


5. Buat method `'update'` pada controller `KaryawanController.php`, dan ketikan kode berikut :

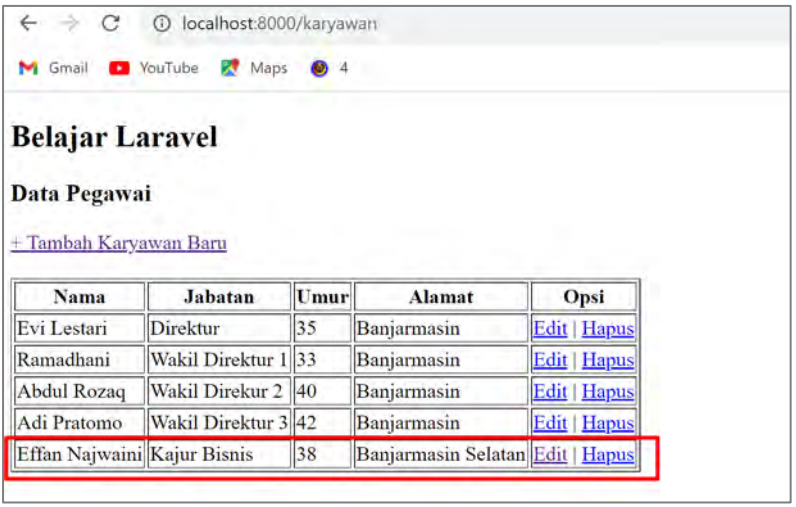


6. Simpan, dan jalankan pada `localhost:8000/karyawan/edit`, maka hasilnya adalah :





Ketika tombol simpan data diklik, maka hasilnya akan berubah menjadi :



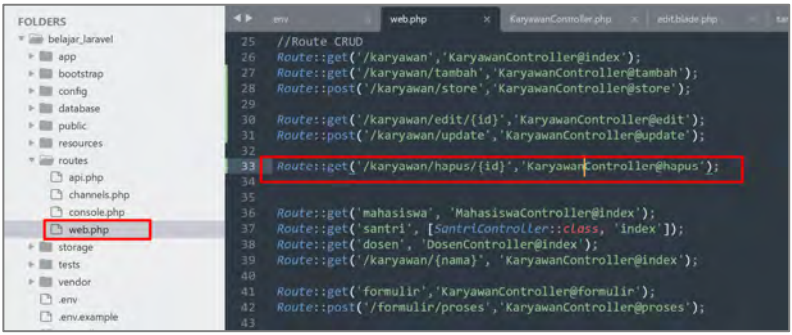
Nama	Jabatan	Umur	Alamat	Opsi
Evi Lestari	Direktur	35	Banjarmasin	Edit | Hapus
Ramadhani	Wakil Direktur 1	33	Banjarmasin	Edit | Hapus
Abdul Rozaq	Wakil Direkur 2	40	Banjarmasin	Edit | Hapus
Adi Pratomo	Wakil Direktur 3	42	Banjarmasin	Edit | Hapus
Effan Najwaini	Kajur Bisnis	38	Banjarmasin Selatan	Edit | Hapus

8.5. Menghapus Data

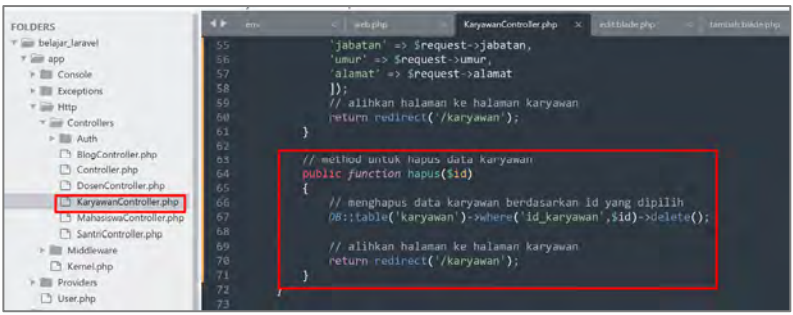
Dalam konteks pengembangan web menggunakan framework Laravel, menghapus data mengacu pada proses menghapus informasi dari basis data menggunakan model Eloquent. Eloquent adalah komponen Laravel yang memungkinkan untuk berinteraksi dengan basis data melalui objek-objek PHP, menjadikan proses manipulasi basis data lebih terstruktur.

Berikut adalah langkah-langkah umum untuk menghapus data dari basis data menggunakan Laravel:

1. Buat route dengan Alamat `'/karyawan/hapus'`, pada `belajar_laravel/routes/web.php` seperti berikut :

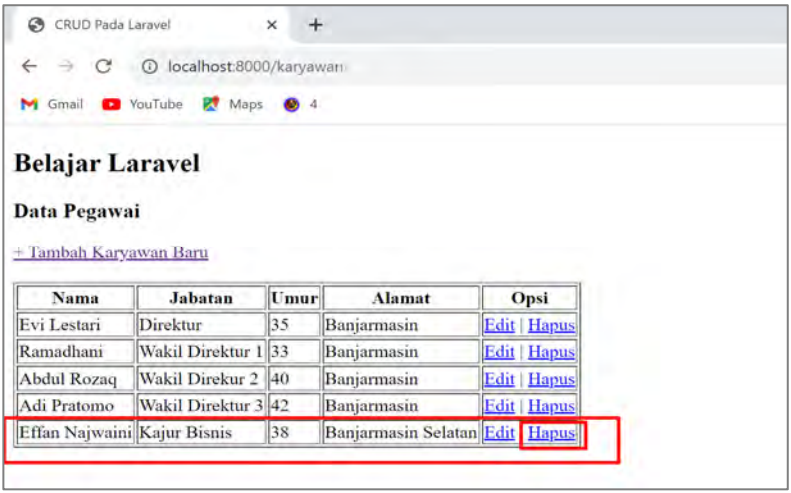


2. Buat method hapus `KaryawanController.php`, seperti gambar berikut :

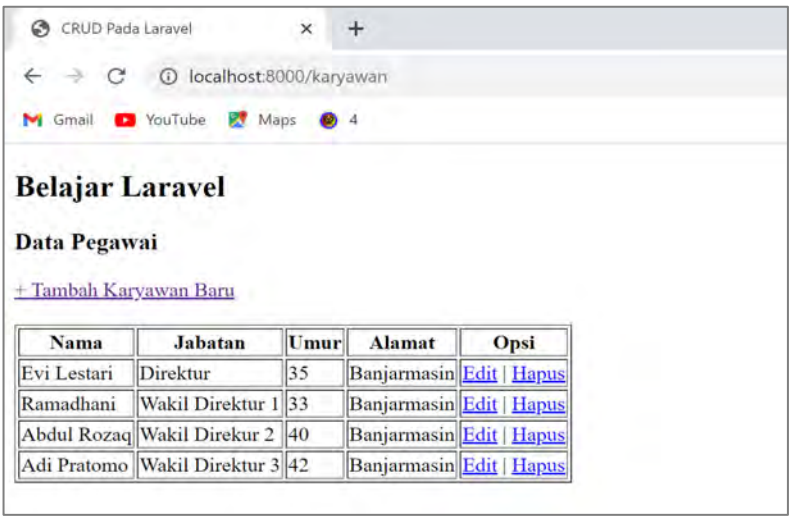


Query builder dibuat untuk menghapus data dari table karyawan fungsi : `DB::table('karyawan')->where('karyawan_id',$id)->delete();`

3. Simpan, dan jalankan pada `localhost:8000/karyawan/hapus`, maka hasilnya adalah :



Sehingga hasilnya adalah :



8.6. *Pagination*

Pagination adalah teknik yang digunakan dalam pengembangan web untuk membagi hasil data yang banyak menjadi beberapa halaman terpisah. Ini sangat bermanfaat ketika memiliki banyak data yang ingin ditampilkan kepada pengguna, seperti daftar posting blog, produk dalam toko online, atau entri dalam basis data.

Dalam konteks pengembangan web dengan framework Laravel, pagination digunakan untuk membagi dan menampilkan sejumlah besar data dalam bentuk halaman-halaman terpisah. Laravel menyediakan dukungan built-in untuk pagination melalui kelas-kelas dan metode-metodenya.

Berikut adalah langkah-langkah umum untuk menggunakan pagination dalam Laravel:

1. Modifikasi function index menjadi sebagai berikut

```
public function index(){

    // $students = Student::all();
    $students = Student::paginate(2);
    return view('index', ['students' => $students]);
}
```

   `Paginate` adalah fungsi yang membagi data menjadi beberapa halaman, adapun parameter yang dimasukkan adalah sejumlah data yang akan ditampikan pada satu halaman.

2. Tambahkan kode program berikut pada `index.blade.php` yang berfungsi untuk menambahkan navigasi pagination pada halaman tersebut

```
    @endforeach
</table>

Current page: {{ $students->currentPage() }}<br>
Total data: {{ $students->total() }}<br>
Data per page: {{ $students->perPage() }}

{{ $students->links('pagination::bootstrap-4') }}
```

3. Silakan dijalankan dan lihat hasilnya

8.7. *Foreign Key Option*

Foreign key option (opsi kunci asing) adalah konsep dalam basis data yang mengacu pada penggunaan relasi antara tabel-tabel yang berbeda dengan cara menggunakan kolom kunci asing (foreign key). Kolom kunci asing adalah kolom dalam tabel yang menyimpan nilai yang mengacu pada nilai unik di kolom kunci utama (primary key) tabel lain.

Foreign key digunakan untuk menjaga integritas referensial antara tabel-tabel dalam basis data, memungkinkan untuk menghubungkan dan menggabungkan data dari tabel yang berbeda dalam cara yang terstruktur. Dengan menggunakan kunci asing, sehingga dapat membangun hubungan antara data di berbagai tabel, yang bermanfaat untuk banyak keperluan, termasuk normalisasi basis data, mencegah anomali data, dan mengelola data yang berkaitan.

Jika bicara tentang database tentu biasanya di dalamnya terdapat 2 atau lebih tabel yang saling berhubungan satu sama lain (relasi). Namun terkadang sering terjadi bahwa jika mengubah salah satu tabel tersebut, maka akan kehilangan pasangannya pada tabel lainnya. Untuk mengatasi hal tersebut maka ada yang namanya *Foreign Key Options*. Dimana dalam penggunaannya *Foreign Key Options* ini digunakan untuk mengatur relasi antar 2 tabel. Jika dalam MySQL atau MariaDB, *Foreign Key Options* ini dapat digunakan jika kita menggunakan *engine* **InnoDB**.

Dalam Foreign Key Options tersebut ada 4 pilihan pengaturan antara lain:

1. `RESCRICT` adalah jika menghapus atau mengubah baris data dalam tabel A maka tidak akan diperbolehkan jika pada tabel B masih ditemukan relasi datanya. InnoDB dapat menolak perintah perubahan atau penghapusan tersebut.
2. `CASCADE` adalah jika menghapus atau mengubah baris data dalam tabel A secara otomatis akan menghapus atau merubah baris yang sesuai dalam tabel B.

3. `SET NULL` adalah jika menghapus atau mengubah baris data dalam tabel A secara otomatis akan mengubah baris pada tabel B menjadi NULL pada kolom yang terelasi. Hal ini dapat dilakukan jika kolom foreign key tidak memiliki pengaturan NOT NULL.
4. `NO ACTION` dalam standar SQL, NO ACTION berarti tidak mengubah apapun pada tabel anak jika mengubah data pada salah satu tabelnya.

Berikut contoh pengaturan Foreign Key Option pada Laravel 9 :

```
$table->foreignId('teacher_id')->constrained('teachers')
      ->cascadeOnUpdate()->cascadeOnDelete();
```

8.8. Latihan

1. Jelaskan perbedaan antara metode create() dan save() dalam model Eloquent ketika membuat data baru.
2. Apa yang dimaksud dengan pagination dalam konteks Laravel? Bagaimana Anda mengimplementasikannya?
3. Bagaimana Anda mengupdate data yang sudah ada dalam Laravel?
4. Apa yang harus Anda pertimbangkan ketika menghapus data untuk memastikan integritas referensial?

BAB 9
ADDING AUTHENTICATION

Capaian Pembelajaran :

1. Mampu memahami authentication scaffolding
2. Mampu memahami authentication blade attribute
3. Mampu memahami Retrieve authenticated user data
4. Mampu memahami Add change password feature

"*Adding Authentication*" dalam konteks Laravel mengacu pada proses menambahkan mekanisme otentikasi (*authentication*) ke dalam aplikasi web. Otentikasi adalah cara untuk mengidentifikasi pengguna dan memberikan akses berdasarkan otorisasi yang sesuai. Laravel menyediakan sistem otentikasi yang kuat dan fleksibel yang dapat dengan mudah diintegrasikan ke dalam proyek yang dibuat.

9.1. *Authentication scaffolding*

Laravel auth adalah fitur esensial yang pasti ada di setiap aplikasi dan website yang dibuat memakai framework Laravel. Auth sangat penting karena memungkinkan *user* untuk melakukan verifikasi email, login akun, reset password dan lain sebagainya pada website maupun aplikasi. Laravel auth (*authentication*) sebagai elemen *security*/perlindungan untuk melindungi halaman-halaman dan bagian penting dari sebuah web atau aplikasi. Dengan adanya Laravel auth, maka hanya hak akses tertentu saja yang dapat mengakses halaman penting tersebut. Hak Akses tersebut biasanya dikenal sebagai admin dan sudah mempunyai hak authentication khusus.

Adapun langkah-langkah membuat fitur auth pada laravel adalah sebagai berikut :

1. Tuliskan perintah berikut pada **Terminal,** kemudian tekan Enter **:**

2. Maka composer akan mengunduh laravel/ui yang akan digunakan pada saat authentication
3. Tuliskan perintah berikut pada **Terminal**, kemudian tekan Enter :

4. Maka laravel akan mengunduh Bootstrap Scaffolding sebagai halaman authentication
5. Untuk menginstall ketik perintah pada **Terminal**, kemudian tekan Enter.

6. Untuk menjalankannya, ketik perintah pada **Terminal**, kemudian tekan Enter

7. Maka fitur authentication sudah dapat dijalankan.

Dengan menambahkan otentikasi, dapat memberikan lapisan keamanan yang penting untuk aplikasi web yang dibuat. Pengguna akan dapat mendaftar, masuk, dan keluar dengan aman, serta mengakses bagian-bagian tertentu dari aplikasi sesuai dengan otorisasi yang ditetapkan.

9.2. *Authentication blade attribute*

Authentication Blade directives adalah *directive* khusus yang disediakan oleh Laravel Blade untuk memudahkan dalam mengontrol tampilan berdasarkan status otentikasi pengguna. Directive ini memungkinkan untuk menampilkan atau menyembunyikan konten berdasarkan apakah pengguna telah terotentikasi atau belum.

Adapun langkah-langkah mengakses authentication melalui view adalah sebagai berikut :

1. Tambahkan kode berikut untuk menambahkan navigasi authentication pada **index.blade.php**

```
<body>

    @if(Auth::check())
        <form action="{{ route('logout') }}" method="post">
        @csrf
        <button type="submit">Logout</button>
        </form>
    @else
        <a href="{{ route('login') }}">Login</a>
        <a href="{{ route('register') }}">Register</a>
    @endif
```

2. Silakan dijalankan dan lihat hasilnya setelah melakukan authentication (login dan logout).

Directive Blade yang terkait dengan otentikasi memudahkan untuk mengontrol dan menampilkan konten dengan fleksibilitas berdasarkan status otentikasi dan izin pengguna.

9.3. *Retrieve authenticated user data*

"*Retrieve authenticated user data*" dalam Laravel merujuk pada proses mengambil informasi tentang pengguna yang saat ini telah diautentikasi (login) ke dalam aplikasi. Ini memungkinkan untuk mengakses data pengguna yang saat ini sedang menggunakan aplikasi.

Adapun langkah-langkah mengambil data user yang berhasil login adalah sebagai berikut :

1. Tambahkan kode berikut untuk mengambil daa user yang berhasil login pada **Controller**

```php
public function index(){
    $user = Auth::user();
    $id = Auth::id();

    // $students = Student::all();
    $students = Student::paginate(2);
    return view('index', ['students' => $students, 'user' => $user, 'id' => $id]);
}
```

   Pastikan sudah mengimport class Auth pada Controller

```php
use Illuminate\Support\Facades\Auth;
```

2. Tambahkan kode berikut untuk menampilkan data user yang berhasil login pada **index.blade.php**

```
@if(Auth::check())
    <p>ID : {{$id}}</p>
    <p>Name : {{$user->name}}</p>
    <p>Role : {{$user->role}}</p>
    <form action="{{ route('logout') }}" method="post">
    @csrf
    <button type="submit">Logout</button>
    </form>
@else
    <a href="{{ route('login') }}">Login</a>
    <a href="{{ route('register') }}">Register</a>
@endif
```

3. Silakan dijalankan dan lihat hasilnya setelah melakukan authentication (*login* dan *logout*)

Dengan menggunakan fitur ini, sehingga dapat mengakses dan menggunakan data pengguna yang terotentikasi dalam aplikasi Laravel yang dibuat.

9.4. *Add change password feature*

"*Add change password feature*" dalam Laravel merujuk pada penambahan kemampuan bagi pengguna untuk mengubah kata sandi (password) setelah mereka masuk ke dalam aplikasi. Ini adalah fitur penting dalam aplikasi web yang memungkinkan pengguna untuk menjaga keamanan akun mereka dengan mengganti kata sandi secara berkala.

Berikut adalah langkah-langkah umum untuk menambahkan fitur perubahan kata sandi pada proyek Laravel:

1. Buatlah function **update_password** pada *HomeController* (sudah ada jika berhasil menginstall *authentication scaffolding*) yang berfungsi untuk menampilkan formulir perubahan password.

```php
public function update_password(){
    $user = Auth::user();
    return view('update_password', compact('user'));
}
```

2. Buatlah satu file di dalam folder **resource/views** dan beri nama file dengan nama **update_password.blade.php**

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Update Password</title>
</head>
<body>
    @if ($errors->any())
        @foreach ($errors->all() as $error)
            <p>{{ $error }}</p>
        @endforeach
    @endif
```

```
    @if (Session::has('message'))
        <p>{{ Session::get('message') }}</p>
    @endif

    <form action="{{ route('store_password') }}" method="post">
        @method('patch')
        @csrf
        <input type="password" name="new_password">
        <input type="password" name="new_password_confirmation">
        <button type="submit">Update</button>
    </form>
</body>
</html>
```

3. Buatlah function **store_password** pada *HomeController* yang berfungsi untuk proses perubahan password.

```
public function store_password(Request $request){
    $request->validate([
        'new_password' => 'required|min:8|confirmed'
    ]);

    Auth::user()->update([
        'password' => Hash::make($request->new_password)
    ]);

    $request->session()->flash('message', 'Successfully change password');

    return Redirect::back();
}
```

Class **Hash** berfungsi untuk memberikan enkripsi pada inputan password sehingga password yang masuk ke dalam database tidak bisa terbaca dengan mudah. Pastikan class Hash sudah diimport pada Controller.

```
use Illuminate\Support\Facades\Hash;
```

4. Tambahkan kode program berikut pada **routes/web** yang berfungsi untuk mengarahkan endpoint pada address bar ke function tertentu

```
Route::get('/update_password', [HomeController::class, 'update_password'])
        ->name('update_password');
Route::patch('/store_password', [HomeController::class, 'store_password'])
        ->name('store_password');
```

5. Silakan dijalankan dan lihat hasilnya

Dengan mengikuti langkah-langkah di atas, dapat menambahkan fitur perubahan kata sandi yang aman dan fungsional kepada pengguna dalam aplikasi Laravel yang dibuat.

9.5. *Multirole*

Multirole pada Laravel merujuk pada kemampuan sistem untuk memberikan peran (role) yang berbeda kepada pengguna dalam suatu aplikasi. Multirole memungkinkan pengguna memiliki lebih dari satu peran atau peran yang berbeda di berbagai bagian aplikasi.

Dalam pengembangan aplikasi, terkadang ada kebutuhan untuk memberikan tingkat akses yang berbeda kepada pengguna berdasarkan tugas atau tanggung jawab mereka dalam sistem. Multirole memungkinkan pengguna untuk memiliki peran yang sesuai dengan fungsinya, baik itu sebagai admin, pengguna biasa, editor, atau peran khusus lainnya. Setiap peran dapat memiliki hak akses yang berbeda terhadap berbagai fitur atau bagian aplikasi.

Dalam konteks Laravel, multirole dapat diimplementasikan dengan menggunakan fitur otentikasi dan otorisasi yang disediakan oleh framework ini. Beberapa hal yang dapat dilakukan, yaitu :

1. Definisi Peran:
   Tentukan jenis peran yang akan ada dalam sistem, misalnya "Admin", "User", "Editor", dan lain-lain.
2. Menghubungkan Peran dengan Pengguna:
   Setiap pengguna dapat memiliki satu atau lebih peran. Ini bisa diwakili oleh relasi dalam model pengguna.
3. Menentukan Hak Akses:

Setiap peran dapat memiliki hak akses yang berbeda terhadap berbagai bagian aplikasi, seperti mengelola pengguna, mengedit konten, mengelola postingan, dan sebagainya.

4. Penerapan Middleware:
   a. Menggunakan middleware Laravel, dapat mengontrol akses ke rute atau tindakan tertentu berdasarkan peran pengguna.
   b. Middleware can dan canAny dapat digunakan untuk memeriksa izin berdasarkan peran.
5. Menampilkan Konten Terkait Peran:
   Dapat menggunakan directive Blade seperti @role untuk menampilkan konten berdasarkan peran tertentu.
6. Manajemen Peran:
   Untuk mengelola peran dan hak akses pengguna, dapat menggunakan paket-paket pihak ketiga atau membuat logika sendiri.

Dengan mengimplementasikan multirole, dapat mengatur tingkat akses yang lebih terperinci dan sesuai dengan kebutuhan aplikasi, sehingga memastikan bahwa pengguna memiliki akses yang sesuai dengan peran dan tanggung jawab mereka dalam sistem.

9.6. *Auth middleware*

Auth middleware dalam Laravel adalah suatu mekanisme yang berfungsi untuk memeriksa apakah pengguna telah terotentikasi sebelum mereka diizinkan mengakses rute atau tindakan tertentu dalam aplikasi. Dengan menggunakan middleware auth, aplikasi akan memastikan bahwa hanya pengguna yang telah melakukan login yang dapat mengakses bagian-bagian tertentu yang memerlukan otentikasi. Jika pengguna belum login, mereka akan diarahkan ke halaman login untuk melakukan otentikasi terlebih dahulu sebelum diizinkan masuk

ke halaman yang dituju. Ini adalah cara yang efektif untuk menjaga keamanan dan mengendalikan akses ke bagian sensitif dari aplikasi.

9.7. Latihan

1. Apa yang dimaksud dengan "ADDING AUTHENTICATION" dalam konteks pengembangan web dengan Laravel?
2. Mengapa fitur autentikasi penting dalam aplikasi web? Berikan alasan mengapa perlunya memiliki fitur autentikasi yang baik dalam aplikasi.
3. Apa itu middleware auth dalam Laravel? Bagaimana cara kerjanya dalam mengontrol akses ke rute atau tindakan tertentu?

BAB 10
File Storage

Capaian Pembelajaran :
Mampu menggunakan *File Storage* pada framework Laravel

File storage (penyimpanan berkas) mengacu pada lokasi atau infrastruktur yang digunakan untuk menyimpan berbagai jenis file, data, atau media dalam suatu sistem. Dalam konteks pengembangan web dan aplikasi, *file storage* menjadi penting untuk mengelola dan menyimpan berkas seperti gambar, video, dokumen, dan data lainnya yang diperlukan oleh aplikasi.

Dalam framework Laravel, termasuk Laravel 5 dan versi berikutnya, ada beberapa opsi untuk mengelola *file storage*:

1. *Public Disk*: *Public disk* menyimpan berkas di dalam direktori storage/app/public dan dapat diakses secara publik melalui URL. Ini cocok untuk menyimpan berkas-berkas yang perlu diakses oleh publik.
2. *Private Disk*: *Private disk* juga menyimpan berkas di dalam direktori storage/app, tetapi tidak dapat diakses secara langsung melalui URL. Biasanya digunakan untuk menyimpan berkas yang hanya dapat diakses oleh pengguna yang memiliki izin.
3. *Local Disk*: *Local disk* merupakan penyimpanan bawaan di Laravel dan menyimpan berkas di dalam direktori lokal proyek yang dibuat. Ini dapat digunakan untuk keperluan penyimpanan umum.
4. *Cloud Storage*: Laravel juga mendukung penyimpanan di berbagai penyedia layanan awan seperti Amazon S3, Google Cloud Storage,

dan lain-lain. Ini bermanfaat ketika perlu menyimpan berkas di luar server aplikasi.

Dalam konteks Laravel, dapat menggunakan fungsi-fungsi yang disediakan oleh framework ini, seperti Storage::put(), Storage::get(), dan lain-lain, untuk berinteraksi dengan berkas-berkas yang disimpan dalam sistem penyimpanan yang dipilih. Hal ini memungkinkan untuk dengan mudah mengunggah, mengunduh, atau memanipulasi berkas-berkas dalam aplikasi yang dibuat.

10.1. *Prerequisite File Storage*

Prerequisite (prasyarat) *file storage* merujuk pada persyaratan yang harus dipenuhi sebelum dapat menggunakan sistem penyimpanan berkas dalam aplikasi yang dibuat, terutama dalam konteks pengembangan web dengan framework Laravel.

Adapun sebelum memulai untuk manipulasi penyimpanan file, langkah-langkah yang perlu dilakukan adalah sebagai berikut :

1. Buat Model baru dengan nama **Picture**

2. Buat Migration dengan nama table **pictures**

3. Buat Controller dengan nama **PictureController**

4. Isi migration dengan stuktur sebagai berikut :

```php
public function up()
{
    Schema::create('pictures', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('path');
        $table->timestamps();
    });
}
```

5. Tambahkan kode program berikut pada Model **Picture**, fungsi ini berfungsi untuk mengizinkan kolom di database untuk dapat dimanipulasi.

```php
class Picture extends Model
{
    use HasFactory;

    protected $fillable = [
        'name',
        'path'
    ];
}
```

6. Ketikkan perintah pada **Terminal**, kemudian tekan Enter

```
TERMINAL

php artisan storage:link
```

10.2. *Store Files*

Storing files (penyimpanan berkas) merujuk pada tindakan mengunggah dan menyimpan berkas-berkas yang diunggah oleh pengguna atau aplikasi ke sistem penyimpanan yang telah ditentukan dalam aplikasi. Dalam konteks pengembangan web dengan framework

Laravel, dapat menggunakan fitur penyimpanan berkas yang disediakan oleh Laravel untuk melakukan tindakan ini.

Adapun langkah-langkah untuk menyimpan file adalah sebagai berikut :

1. Buatlah function **create** pada *PictureController* yang berfungsi untuk menampilkan formulir menambahkan file.

```
public function create(){
    return view('create_picture');
}
```

2. Buatlah satu file di dalam folder **resource/views** dan beri nama file dengan nama **create_picture.blade.php**
3. Kode program untuk membuat formulir menambahkan file adalah sebagai berikut :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Create Picture</title>
</head>
<body>
    <form action=" {{ route('picture.store') }} " method="post" enctype="multipart/form-data">
        @csrf
        <input type="text" name="name">
        <input type="file" name="file">
        <button type="submit">Submit</button>
    </form>
</body>
</html>
```

4. Tambahkan kode program berikut pada **routes/web** yang berfungsi untuk mengarahkan endpoint pada address bar ke function tertentu

```
Route::get('/picture/create', [PictureController::class, 'create'])->name('picture.create');
Route::post('/picture/create', [PictureController::class, 'store'])->name('picture.store');
```

5. Buatlah function **store** pada *PictureController* yang berfungsi untuk proses menambahkan data file.

```php
public function store(Request $request){
    $file = $request->file('file');
    $name = $request->name;

    $path = time() . '_' . $request->name . '.' . $file->getClientOriginalExtension();

    Storage::disk('local')->put('public/'. $path, file_get_contents($file));

    Picture::create([
        'name' => $name,
        'path' => $path,
    ]);

    return Redirect::route('picture.create');
}
```

a. function time berfungsi untuk mengambil waktu saat ini dalam bentuk detik (integer)
b. Function getClientOriginalExtension berfungsi untuk mengambil ekstensi dari file yang akan diunggah
c. Class Storage berfungsi untuk unggah file ke dalam folder yang diinginkan

Pastikan import class Storage

```php
use Illuminate\Support\Facades\Storage;
```

6. Silakan dijalankan dan lihat hasilnya

Laravel menyediakan API yang kuat untuk mengelola penyimpanan berkas dengan mudah dan aman. Dengan menggunakan metode-metode yang disediakan oleh Laravel, pengunggahan berkas dapat dikelola dan mengintegrasikan fitur penyimpanan berkas dalam aplikasi yang dibuat dengan cara yang efisien dan aman.

10.3. *Show Files*

Pada Laravel, "show files" dapat merujuk pada dua hal yang berbeda tergantung pada konteksnya:

1. Menampilkan Isi Berkas: Dalam konteks menampilkan isi berkas, "show files" mengacu pada tindakan menampilkan konten atau isi dari suatu berkas, seperti teks dalam dokumen atau gambar dalam

format visual. Dapat menggunakan metode bawaan PHP atau pustaka khusus dalam Laravel untuk membaca dan menampilkan isi berkas.

2. Menampilkan Daftar Berkas: Dalam konteks menampilkan daftar berkas, "show files" merujuk pada tindakan menampilkan daftar berkas atau direktori dari sistem penyimpanan. Ini dapat berguna jika ingin memberikan tampilan yang memungkinkan pengguna melihat daftar berkas yang ada dalam direktori tertentu.

Jika ingin mengambil arti pertama, yaitu menampilkan isi berkas, dapat menggunakan fungsi-fungsi bawaan PHP seperti `file_get_contents()` atau menggunakan fungsi-fungsi yang disediakan oleh Laravel seperti `Storage::get()` untuk membaca dan menampilkan isi berkas dalam aplikasi yang dibuat.

Namun, jika berbicara tentang menampilkan daftar berkas atau direktori, dapat menggunakan metode-metode dalam Laravel untuk mengakses daftar berkas dalam sistem penyimpanan. Misalnya, dapat menggunakan `Storage::files()` atau `Storage::allFiles()` untuk mendapatkan daftar berkas dalam direktori tertentu.

Pastikan untuk memahami konteks yang tepat untuk "show files" dalam skenario yang di hadapi, sehingga dapat mengambil langkah yang sesuai dalam mengimplementasikannya dalam aplikasi Laravel yang dibuat.

Adapun langkah-langkah untuk menampilkan file adalah sebagai berikut :

1. Buatlah function `show` pada *PictureController* yang berfungsi untuk menampilkan file.

```
public function show(Picture $picture){
    $url = Storage::url($picture->path);
    return view('show_picture', compact('url', 'picture'));
}
```

2. Buatlah satu file di dalam folder `resource/views` dan beri nama file dengan nama `show_picture.blade.php`
3. Kode program untuk membuat formulir menambahkan file adalah sebagai berikut :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Show Picture</title>
</head>
<body>
    <p>{{ $picture->name }}</p>
    <p>{{ $picture->path }}</p>
    <img src="{{ $url }}" alt="" height="200px">
</body>
</html>
```

4. Tambahkan kode program berikut pada `routes/web` yang berfungsi untuk mengarahkan endpoint pada address bar ke function tertentu

```
Route::get('/picture/{picture}', [PictureController::class, 'show'])
        ->name('picture.show');
```

5. Silakan dijalankan dan lihat hasilnya

10.4. *Delete Files*

Menghapus berkas (delete files) dalam Laravel adalah tindakan untuk menghapus berkas dari sistem penyimpanan yang telah ditentukan dalam aplikasi. Tindakan ini bisa sangat penting untuk membersihkan berkas-berkas yang tidak diperlukan lagi atau mengelola konten dalam aplikasi yang dibuat.

Adapun langkah-langkah untuk menghapus file adalah sebagai berikut :

1. Buatlah function `delete` pada `PictureController` yang berfungsi untuk menghapus file.

```php
public function delete(Picture $picture){
    Storage::delete('public/' . $picture->path);
    $picture->delete();

    return Redirect::route('picture.create');
}
```

2. Tambahkan kode program berikut pada `routes/web` yang berfungsi untuk mengarahkan endpoint pada address bar ke function tertentu

```php
Route::delete('/picture/{picture}', [PictureController::class, 'delete'])
        ->name('picture.delete');
```

3. Tambahkan kode berikut untuk menambahkan tombol Delete pada **show_picture.blade.php**

```php
<form action="{{ route('picture.delete', $picture) }}" method="post">
    @method('delete')
    @csrf
    <button type="submit">Delete</button>
</form>
```

4. Silakan dijalankan dan lihat hasilnya

Penting untuk berhati-hati saat menghapus berkas, terutama jika berkas tersebut memiliki implikasi penting dalam aplikasi. Pastikan bahwa tindakan penghapusan berkas dilakukan dengan bijak dan hanya untuk berkas yang memang tidak diperlukan lagi dalam konteks aplikasi.

10.5. *Copy and Move Files*

Dalam konteks pengembangan aplikasi web menggunakan Laravel, "copy and move files" mengacu pada tindakan menyalin (copy) dan memindahkan (move) berkas dari satu lokasi ke lokasi lain dalam sistem penyimpanan yang telah dikonfigurasi dalam aplikasi.

Laravel menyediakan API yang kuat untuk mengelola operasi ini dengan mudah.

Adapun langkah-langkah untuk menyalin dan memindahkan file adalah sebagai berikut :

1. Buatlah function `copy` pada `PictureController` yang berfungsi untuk menghapus file.

```
public function copy(Picture $picture){
    Storage::copy('public/' . $picture->path, 'copy/' . $picture->path);
    return Redirect::route('picture.create');
}
```

2. Buatlah function `move` pada `PictureController` yang berfungsi untuk menghapus file.

```
public function move(Picture $picture){
    Storage::move('public/' . $picture->path, 'move/' . $picture->path);
    return Redirect::route('picture.create');
}
```

3. Tambahkan kode program berikut pada `routes/web` yang berfungsi untuk mengarahkan endpoint pada address bar ke function tertentu

```
Route::get('/copy/{picture}', [PictureController::class, 'copy'])
        ->name('picture.copy');
Route::get('/move/{picture}', [PictureController::class, 'move'])
        ->name('picture.move');
```

4. Tambahkan kode berikut untuk menambahkan tombol Copy dan Move pada `show_picture.blade.php`

```
<form action="{{ route('picture.copy', $picture) }}" method="get">
    @csrf
    <button type="submit">Copy</button>
</form>
<form action="{{ route('picture.move', $picture) }}" method="get">
    @csrf
    <button type="submit">Move</button>
</form>
```

5. Silakan dijalankan dan lihat hasilnya.

Pastikan untuk memverifikasi sudah diketahui lokasi sumber dan tujuan berkas yang ditentukan valid dan bahwa sudah memiliki izin untuk melakukan operasi ini pada berkas-berkas tersebut. Selain itu, pastikan bahwa tindakan "copy and move files" sesuai dengan kebutuhan dan logika bisnis aplikasi.

10.6. Latihan

1. Apa yang dimaksud dengan "file storage" dalam pengembangan web dan bagaimana itu berhubungan dengan Laravel?
2. Apa tujuan utama dari menggunakan sistem penyimpanan berkas dalam aplikasi Laravel?
3. Bagaimana cara mengkonfigurasi sistem penyimpanan berkas dalam Laravel?
4. Apa perbedaan antara penyimpanan berkas publik dan pribadi dalam Laravel?
5. Jelaskan langkah-langkah yang diperlukan untuk mengunggah berkas menggunakan sistem penyimpanan berkas dalam Laravel.

.

GLOSARIUM

API (Application Programming Interface): Antarmuka yang memungkinkan berbagai aplikasi untuk berkomunikasi satu sama lain melalui permintaan HTTP. Laravel memungkinkan untuk membuat dan mengelola API dengan mudah.

Artisan: Perintah baris perintah (CLI) bawaan dalam Laravel yang membantu dalam tugas-tugas seperti membuat controller, model, migrasi, dan lain-lain.

Blade Templating Engine: Mesin templating bawaan dalam Laravel yang memungkinkan penggunaan sintaks Blade untuk membuat tampilan dinamis. Blade memungkinkan looping, kondisi, inheritance, dan lainnya.

Composer Packages: Pustaka atau komponen pihak ketiga yang dapat ditambahkan ke proyek Laravel menggunakan Composer. Ini dapat menyediakan fitur tambahan seperti integrasi API, pustaka utilitas, dan lainnya.

Composer: Manajer ketergantungan PHP yang digunakan dalam Laravel untuk mengelola pustaka pihak ketiga dan komponen yang diperlukan.

Controller: Bagian dalam aplikasi yang mengelola logika bisnis dan menghubungkan model dengan view. Controller merespons permintaan HTTP dan memproses data sebelum mengirimkannya ke tampilan.

JWT (JSON Web Token): Standar terbuka (RFC 7519) untuk membuat token akses yang aman dan terverifikasi, digunakan dalam autentikasi API dan layanan web.

Laravel: Kerangka kerja PHP yang kuat dan populer untuk pengembangan aplikasi web. Menggunakan konsep MVC untuk mempermudah pembuatan aplikasi yang efisien dan terstruktur.

Model: Representasi objek dari entitas dalam basis data. Model mengatur interaksi dengan basis data dan berisi logika untuk mengambil dan memanipulasi data.

MVC (Model-View-Controller): Pola desain yang memisahkan aplikasi web menjadi tiga komponen utama: Model (mengelola data), View (menampilkan data), dan Controller (mengendalikan logika bisnis). Laravel menggunakan pola ini secara intensif.

Route: Mekanisme yang digunakan untuk menghubungkan URL dengan fungsi atau controller dalam Laravel. Route mendefinisikan bagaimana aplikasi merespons permintaan HTTP.

Daftar Pustaka

Ayon Goding. "Membuat Route dan View Laravel." Link : https://www.ayongoding.com/

Codepolitan. "Belajar Routing Laravel." Link : https://codepolitan.com/blog/belajar-routing-laravel

Dewaweb. "Apa Itu Laravel?" Link : https://www.dewaweb.com/blog/apa-itu-laravel/

Duniailkom. "Tutorial Belajar Laravel: Cara Membuat View di Laravel." Link : https://www.duniailkom.com/tutorial-belajar-laravel-cara-membuat-view-di-laravel/

Lab Informatika. "Memahami Routing pada Laravel." Link : https://www.lab-informatika.com/memahami-routing-pada-laravel

Laravel Documentation. "Routing." Link : https://laravel.com/docs/

Malas Ngoding. "Belajar Route dan View pada Laravel." Link : https://www.malasngoding.com/

Muhammad Ibrahim. "Belajar Laravel: Kupas Tuntas Routing di Laravel." Link : https://medium.com/@muhammad.ibrahim/belajar-laravel-kupas-tuntas-routing-di-laravel-f8dfac2e64fd

Niagahoster. "Cara Install Composer." Link : https://www.niagahoster.co.id/blog/cara-install-composer/#2_Install_Composer

Pemburu Kode. "Fungsi Controller pada Laravel." Link : https://pemburukode.com/fungsi-controller-pada-laravel/#apa-itu-controller

Universitas Ciputra. "Laravel View (Blade)." Link : https://informatika.uc.ac.id/2019/09/laravel-view-blade/

PEMROGRAMAN WEB
LANJUTAN DENGAN LARAVEL
BAGIAN 1

EVI LESTARI PRATIWI
ABDUL ROZAQ
RAMADHANI NOOR PRATAMA

Framework merupakan suatu susunan yang terdiri dari kode-kode umum yang digunakan untuk membangun sistem dan aplikasi. Fungsi framework adalah sebagai pola dasar atau contoh yang menyediakan fitur cerdas dan elemen struktur standar, yang bertujuan untuk mempermudah tugas para pengembang. Framework juga menunjukkan konsistensi tinggi karena telah diuji secara ekstensif dan terbukti berhasil.

Cara kerja dari framework memungkinkan pengembang untuk lebih fokus pada tujuan utama proyek tanpa khawatir tentang unsur dasar strukturnya. Dengan tidak perlu memulai dari awal, para pengembang dapat menghemat waktu dan sumber daya finansial serta mengurangi risiko kesalahan.

Proses pelaksanaannya didasarkan pada penggunaan ulang serangkaian kode umum yang dapat diterapkan sebagai kerangka dasar proyek. Desain ini perlu sesuai dengan bahasa dan karakteristik dari framework yang dimaksud.

Penerbit Poliban Press
Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry, Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp : (0511)3305052
Email : press@poliban.ac.id

ISBN 978-623-5259-06-2 (PDF)