

DAFTAR ISI

DAFTAR ISI	1
Laravel	2
Tujuan	2
Daftar Materi	2
Apa Itu Laravel	2
Instalasi Laravel	3
1. Download dan install composer	3
2. Install Laravel	3
Buat Project Laravel Sederhana	4
1. Buka direktori	4
2. Buka terminal di VsCode	4
3. Mengetahui MVC (Model View Controller)	5
4. Membuat sistem pemesanan kafe	6

Laravel

Tujuan

Setelah mempelajari modul ini, peserta diharapkan dapat:

- Memahami konsep dasar Laravel
- Mampu menginstal dan menjalankan Laravel
- Mampu membuat projek Laravel sederhana

Prasyarat

- Peserta diharapkan memiliki pengetahuan dasar tentang PHP.

Daftar Materi

- Konsep Dasar Laravel
- Instalasi dan Pengaturan Laravel
- Membuat Proyek Laravel Sederhana

Apa Itu Laravel

Laravel adalah sebuah **framework** aplikasi web dengan sintaks yang elegan dan ekspresif. Berikut adalah fakta-fakta tentang Laravel:

- Sebuah kerangka yang mendukung pembangunan aplikasi web
- Menyediakan standar / cara untuk membangun aplikasi web
- Menyediakan *database library*

Instalasi Laravel

Syarat untuk menginstal dan menjalankan Laravel adalah mempunyai aplikasi **XAMPP** dan **composer** untuk menjalankan server local. Adapun langkah langkah untuk menginstall Laravel adalah sebagai berikut :

1. Download dan install composer

- Kunjungi halaman <https://getcomposer.org/>
- Click “Download”

Windows Installer

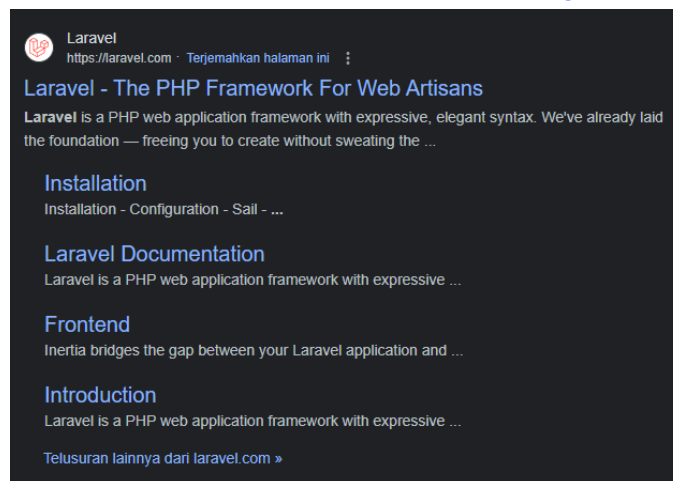
The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

- Click Composer-Setup.exe
- Download dan install seperti biasa
- Cek apakah composer sudah terinstall atau belum dengan cara mengetikan “composer --version”

2. Install Laravel

- Buka halaman dokumentasi Laravel
<https://laravel.com/docs/10.x/installation#creating-a-laravel-project>



- Cari bagian “Creating Project” dan jalankan:
“composer create-project laravel/laravel praktikumnew10”

Creating A Laravel Project

Before creating your first Laravel project, make sure that your local machine has PHP and [Composer](#) installed. If you are developing on macOS, PHP and Composer can be installed in minutes via [Laravel Herd](#). In addition, we recommend [installing Node and NPM](#).

After you have installed PHP and Composer, you may create a new Laravel project via Composer's `create-project` command:

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

```
labko@LAPTOP-981S5FTC MINGW64 /d
$ composer create-project laravel/laravel praktikumnew-app
Creating a "laravel/laravel" project at "./praktikumnew-app"
Installing laravel/laravel (v10.2.10)
- Downloading laravel/laravel (v10.2.10)
- Installing laravel/laravel (v10.2.10): Extracting archive
Created project in D:\\praktikumnew-app
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
```

- Laravel siap digunakan !

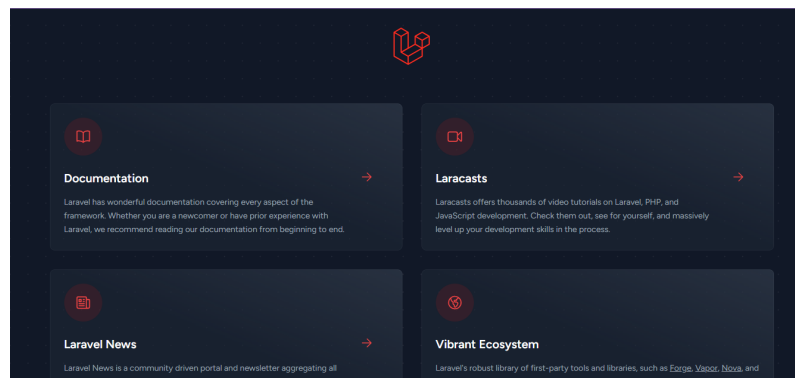
Buat Project Laravel Sederhana

1. Buka direktori

Buka direktori dengan cara buka folder yang sudah di instal laravel lalu klik kanan pilih "Git Bash here" lalu ketikan "**code .**".

2. Buka terminal di VsCode

Cari view > Terminal. Lalu ketikan "composer install" > "php artisan key:generate" lalu "php artisan serve"



Jika tampilan di atas tampil di halaman kalian, Selamat!! kalian sudah bisa memakai Laravel sesuai kreasi kalian!!.

3. Mengetahui MVC (Model View Controller)

Model dalam Laravel berfungsi sebagai representasi data dan aturan bisnis aplikasi, bertanggung jawab untuk berinteraksi dengan basis data atau sumber data lainnya, serta melakukan manipulasi data. Secara khusus, dalam konteks Laravel, model seringkali terkait dengan tabel dalam basis data dan menyediakan antarmuka untuk mengakses serta memanipulasi data.

View berfungsi sebagai bagian yang mengelola presentasi data dan interaksi pengguna. View bertanggung jawab untuk menampilkan informasi kepada pengguna dengan tata letak dan format yang sesuai. Dalam implementasi Laravel, view seringkali diwujudkan dalam bentuk file-template Blade. Sintaks Blade yang digunakan dalam file-template tersebut memungkinkan pembuatan tampilan dinamis dengan cara yang lebih ekspresif dan efisien.

Controller memiliki tanggung jawab untuk menerima permintaan dari pengguna, melakukan pemrosesan logika aplikasi, dan mengirimkan tanggapan (response) yang sesuai. Controller bertindak sebagai perantara antara Model (yang menangani representasi data dan aturan bisnis) dan View (yang menangani presentasi data kepada pengguna). Dalam implementasi Laravel, controller mendefinisikan metode-metode yang menangani logika aplikasi dan dipanggil oleh router untuk menanggapi permintaan HTTP, membantu memisahkan tanggung jawab dan memfasilitasi pengembangan terstruktur.

- `app/Http/Controllers`: Direktori ini berisi file-controller yang menangani logika aplikasi.
- `app/Models`: Direktori ini berisi file-model yang menangani akses dan manipulasi data.
- `resources/views`: Direktori ini berisi file-template Blade yang digunakan untuk menampilkan tampilan kepada pengguna.
- `routes/web.php` : Untuk mengarahkan kemana alur web akan dibawa.

4. Membuat sistem pemesanan kafe

- Hapus "Welcome.blade.php" di view (Karena tidak terpakai)
- buat file "pesan.blade.php" dan "dashboardKasir.blade.php" di view untuk ditampilkan
- buat controller pemesanan dengan cara ketikan "php artisan make:controller OrderController" untuk mengatur alur data
- Isi controller dengan code dibawah berikut:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class OrderController extends Controller
{
    public function showFormPesan()
    {
        $menuItems = [
            ['id' => 'menu1', 'name' => 'Menu 1', 'harga'
=> 12000],
            ['id' => 'menu2', 'name' => 'Menu 2', 'harga'
=> 15000],
            ['id' => 'menu3', 'name' => 'Menu 3', 'harga'
=> 13000],
            ['id' => 'menu4', 'name' => 'Menu 4', 'harga'
=> 10000],
        ];

        return view('pesan', compact('menuItems'));
    }

    public function submitOrder(Request $request)
    {
        $menuItems = [
            'menu1' => ['name' => 'Menu 1', 'harga' =>
12000],
            'menu2' => ['name' => 'Menu 2', 'harga' =>
15000],
            'menu3' => ['name' => 'Menu 3', 'harga' =>
13000],
            'menu4' => ['name' => 'Menu 4', 'harga' =>
10000],
        ];
    }
}
```

```

];

$data = [
    'name' => $request->input('name'),
    'email' => $request->input('email'),
    'pesan' => $request->input('pesan'),
    'menuItems' => $menuItems,
];

$total = 0;
foreach ($data['pesan'] as $itemId => $quantity) {
    $total += $quantity *
$menuItems[$itemId]['harga'];
}

$data['total'] = $total;
return view('dashboardKasir', compact('data',
'menuItems'));
}
}

```

- Isi routes/web.php untuk mengatur alur halaman, isi dengan code dibawah berikut:

```

<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\OrderController;

Route::get('/pesan', [OrderController::class,
'showFormPesan'])->name('show-form-pesan');
Route::post('/submit-order', [OrderController::class,
'submitOrder'])->name('submit-order');

Route::get('/', function () {
    return redirect()->route('show-form-pesan'); // Ubah
rute ke show-form-pesan
});

```

- Isi views/dashboardKasir.blade.php untuk menampilkan hasil pesanan dengan code dibawah berikut:

```
<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale())
}}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Selamat Datang</title>
    <!-- Fonts -->
    <link rel="preconnect" href="https://fonts.bunny.net">
</head>
<body>
    <h1>Selamat Datang!</h1>

    <p>Nama: {{ $data['name'] }}</p>
    <p>Email: {{ $data['email'] }}</p>

    <p>Pesanan:</p>
    <ul>
        @foreach($data['pesan'] as $itemId => $quantity)
            @if($quantity > 0)
                <li>{{ $quantity }} x {{
$menuItems[$itemId]['name'] }} (Rp {{
$menuItems[$itemId]['harga'] }})</li>
            @endif
        @endforeach
    </ul>

    <p>Total Belanjaan: Rp {{ $data['total'] }}</p>
</body>
</html>
```


- Isi views/pesan.blade.php untuk menampilkan form dengan code dibawah berikut:

```
<!-- resources/views/pesan.blade.php -->

<!DOCTYPE html>
<html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Pemesanan</title>
    <!-- Fonts -->
    <link rel="preconnect" href="https://fonts.bunny.net">
</head>
<body>
    <h1>Form Pemesanan</h1>
    <form method="post" action="{{ route('submit-order') }}">
        @csrf
        <label for="name">Nama:</label>
        <input type="text" id="name" name="name" required><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br>

        <label>Pilihan Pesanan:</label><br>
        @foreach($menuItems as $menuItem)
            <div>
                <label for="{{ $menuItem['id'] }}">{{ $menuItem['name'] }} (Rp {{ $menuItem['harga'] }})</label>
                <input type="number" id="{{ $menuItem['id'] }}" name="pesan[{{ $menuItem['id'] }}" min="0">
            </div>
        @endforeach

        <button type="submit">Submit Pesanan</button>
```

```
</form>  
</body>  
</html>
```

- Jalankan php artisan serve di terminal!

Post Test

Buatlah sebuah aplikasi web pendataan penduduk yang meminta data berupa sebagai berikut:

- NIK
- Nama
- Provinsi
- Kota
- Nomor telpon

lalu tampilkan pada halaman **dashboardAdmin**