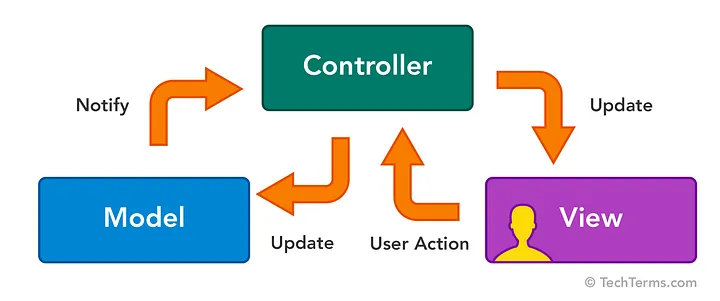
**Konsep MVC Pada Laravel**

****

MVC (Model, View, Controller) adalah sebuah pola desain perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama: Model, View, dan Controller. Pola ini membantu dalam pengorganisasian kode dan pemisahan tanggung jawab, yang membuat pengembangan dan pemeliharaan aplikasi menjadi lebih mudah.

1. Model

Model merupakan bagian dari aplikasi yang menangani logika bisnis dan interaksi dengan basis data. Model bertanggung jawab untuk mengambil, menyimpan, dan mengelola data. Dalam konteks aplikasi web, model biasanya berkomunikasi dengan database untuk mengambil atau menyimpan data yang diperlukan oleh aplikasi.

Contoh:

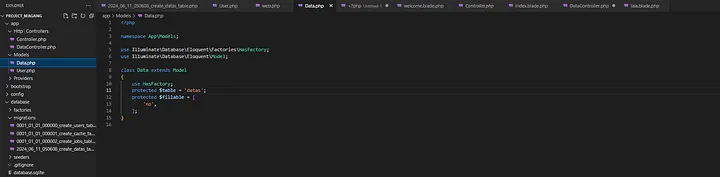
* Mengambil data pengguna dari basis data
* Menyimpan data pesanan ke basis data

Langkah-Langkah Membuat Model Pada laravel :

* Ketik kode berikut dalam terminal kalian

php artisan make:model Data

* Laravel akan membuat file model bernama “Data”



1. View

View adalah bagian dari aplikasi yang menangani presentasi data kepada pengguna. View bertanggung jawab untuk menampilkan data dalam format yang bisa dimengerti dan digunakan oleh pengguna. Ini biasanya berupa antarmuka pengguna (UI) yang menampilkan data yang diperoleh dari Model.

Contoh:

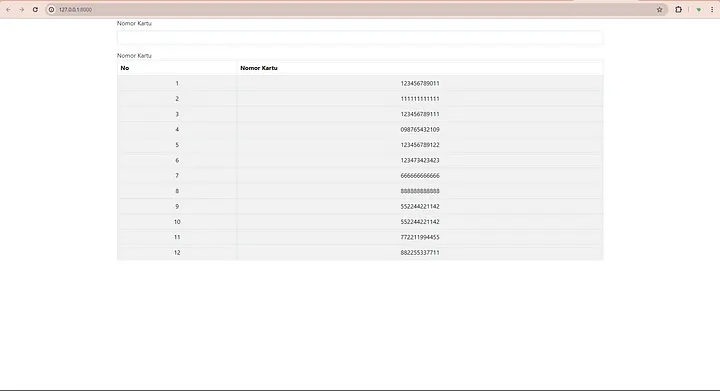
* Halaman HTML yang menampilkan daftar produk
* Tampilan grafis dari laporan penjualan

Langkah-Langkah Membuat View pada Laravel :

* Secara default, Laravel sudah memiliki folder resources/views. Jika Anda ingin mengorganisir view Anda dengan lebih baik, Anda dapat membuat sub-folder di dalam folder resources/views sesuai kebutuhan proyek Anda.
* Buat File Blade Baru
* Buat file Blade baru di dalam folder resources/views. Sebagai contoh, kita akan membuat file welcome.blade.php.
* Ketik kode dibawah ini dalam file welcome.blade.php kalian

<!doctype html>  
<html lang="en">  
  
<head>  
 <meta charset="utf-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1">  
 <title>Laravel</title>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">  
 <script>  
 document.addEventListener('DOMContentLoaded', function () {  
 const nomorInput = document.getElementById('no');  
  
 nomorInput.addEventListener('input', function () {  
 if (nomorInput.value.length === 12) {  
 document.getElementById('nomorForm').submit();  
 }  
 });  
 });  
 </script>  
</head>  
  
<body>  
 <div class="container">  
 <form method="POST" action="{{ route('store\_no') }}" id="nomorForm">  
 @csrf  
 <div class="mb-3">  
 <label for="no" class="form-label">Nomor Kartu</label>  
 <input type="number" class="form-control" id="no" name="no">  
 </div>  
 </form>  
 @if(session('success'))  
 <div class="alert alert-success mt-3">  
 {{ session('success') }}  
 </div>  
 @endif  
 </div>  
 <div class="container">  
 <table id="example1" class="table table-bordered table-striped">  
 <thead>  
 <label for="">Nomor Kartu</label>  
 <tr>  
 <th>No</th>  
 <th>Nomor Kartu</th>  
 {{-- <th>Action</th> --}}  
 </tr>  
 </thead>  
 <tbody>  
 @foreach ($datashow as $dt )  
 <td><center>{{$loop->iteration}}</center></td>  
 <td><center>{{$dt->no}}</center></td>  
 </tbody>  
 @endforeach  
 </table>  
 </div>  
  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous">  
 </script>  
</body>  
  
</html>

* Hasilnya akan seperti ini



1. Controller

Controller adalah bagian dari aplikasi yang menghubungkan Model dan View. Controller menangani input dari pengguna, memprosesnya, dan memperbarui Model serta View sesuai kebutuhan. Controller bertindak sebagai perantara yang mengarahkan data antara Model dan View.

Contoh:

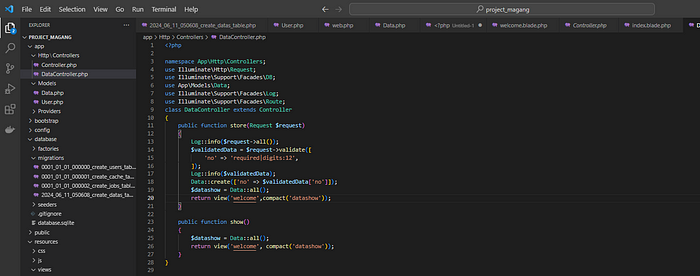
* Mengelola permintaan HTTP dari pengguna
* Memproses data yang dimasukkan oleh pengguna dalam formulir dan memperbarui Model

Langkah-Langkah Membuat Controller :

* Ketikan kode dibawah ini dalam terminal kalian

php artisan make:controller DataController

* Laravel akan membuat file controller secara otomatis



* Ubah kode dalam file controller kalian dengan kode dibawah ini

<?php  
  
namespace App\Http\Controllers;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
use App\Models\Data;  
use Illuminate\Support\Facades\Log;  
use Illuminate\Support\Facades\Route;  
class DataController extends Controller  
{  
 public function store(Request $request)  
 {  
 Log::info($request->all());  
 $validatedData = $request->validate([  
 'no' => 'required|digits:12',  
 ]);  
 Log::info($validatedData);  
 Data::create(['no' => $validatedData['no']]);  
 $datashow = Data::all();  
 return view('welcome',compact('datashow'));  
 }  
  
 public function show()  
 {  
 $datashow = Data::all();  
 return view('welcome', compact('datashow'));  
 }  
}

**Cara Kerja MVC**

1. Pengguna mengirimkan permintaan ke aplikasi (misalnya, mengklik tombol atau memasukkan URL).
2. Controller menerima permintaan ini dan memprosesnya, mungkin dengan berinteraksi dengan Model untuk mengambil atau memperbarui data.
3. Model mengelola data dan memberikan hasil kembali kepada Controller.
4. Controller memperbarui View berdasarkan data dari Model.
5. View menampilkan data yang diperbarui kepada pengguna.

**Contoh Implementasi**

Dalam sebuah aplikasi web, jika pengguna ingin melihat daftar produk:

1. Pengguna mengirimkan permintaan (HTTP GET) untuk halaman daftar produk.
2. Controller menerima permintaan ini dan meminta data produk dari Model.
3. Model mengambil data produk dari basis data dan mengembalikannya ke Controller.
4. Controller kemudian mengirimkan data tersebut ke View.
5. View menghasilkan halaman HTML yang menampilkan daftar produk dan mengirimkannya kembali ke pengguna.

**Manfaat MVC**

1. Pemeliharaan Kode yang Lebih Mudah: Karena kode dipisahkan berdasarkan tanggung jawabnya, lebih mudah untuk memperbarui atau memperbaiki bagian tertentu dari aplikasi tanpa mempengaruhi bagian lain.
2. Pengembangan Paralel: Tim pengembang dapat bekerja secara paralel pada Model, View, dan Controller, karena masing-masing komponen berdiri sendiri.
3. Skalabilitas: Memisahkan logika aplikasi ke dalam komponen yang berbeda membuat aplikasi lebih mudah untuk diskalakan dan diperluas di masa mendatang.

Original Source : https://medium.com/@ifajrig/penjelasan-tentang-mvc-pada-laravel-bersama-contohnya-147808432d20