

Nama : Fikri Maulana
Kelas : IF-44-10
NIM : 1301200239

MODUL 6

6.1 Framework dan MVC

Framework (Kerangka Kerja) adalah kumpulan dari fungsi – fungsi / prosedur – procedure / class – class yang sudah dirancang khusus untuk memudahkan programmer ketika akan membangun suatu sistem seperti web, aplikasi.

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi web.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah framework (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya framework menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll).
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.

Model-View-Controller (MVC) adalah konsep dalam pembangunan aplikasi web. Pada MVC dilakukan pemisahan dalam pengembangan aplikasi berdasarkan komponen utama yang membangun aplikasi seperti manipulasi data, user interface, dan bagian yang menjadi control aplikasi. Terdapat 3 jenis komponen dalam MVC pattern yaitu :

- View : bagian yang mengatur dan membuat user interface dalam web
- Model : bagian yang dapat mengakses dan mengatur manipulasi data pada database
- Controller: bagian yang mengatur alur kerja dari aplikasi web atau menghubungkan dan mengatur alur kerja antara model dan view

6.2 Pengenalan Laravel

Laravel adalah salah satu framework PHP yang open-source dan gratis untuk dikembangkan. Laravel dirancang untuk membantu pengembang dalam membangun aplikasi web dengan mudah, cepat, dan efisien. Laravel menawarkan fitur-fitur lengkap, termasuk autentikasi pengguna, routing, caching, dan fitur-fitur lainnya yang dapat membantu mempercepat proses pengembangan. Laravel juga menyediakan arsitektur MVC (Model-View-Controller) yang terorganisir dengan baik, sehingga memudahkan pengembang dalam mengorganisir kode dan membuat aplikasi yang mudah dipelihara.

Dengan kesederhanaan sintaksnya, Laravel menjadi salah satu framework PHP yang populer dan banyak digunakan oleh para pengembang di seluruh dunia.

Berikut adalah istilah yang sering ditemui di Laravel :

1. Model, class PHP yang dirancang untuk bekerja dengan informasi dari database.
2. Controller, inti aplikasi yang menentukan penanganan logic dari aplikasi web
3. Route, bagian yang menangani HTTP request.
4. View, halaman web seperti header, footer, sidebar, dan sebagainya yang ditanamkan di halaman web. View tidak pernah dipanggil secara langsung, tetapi harus dipanggil dari controller.
5. Library, class yang berisi fungsi-fungsi untuk penyelesaian kasus tertentu.

6.3 Cara Kerja Laravel

Ketika suatu URL dibuka, maka file index.php akan dibaca. Di dalam file ini, Laravel memeriksa apakah URL tersebut terdaftar pada Routing yang sudah dibuat. Jika adapemetaannya dan memenuhi Security (jika diharuskan, autentikasi misalnya) maka Laravel akan memanggil Controller sesuai yang dipetakan oleh Routing. Di dalam Controller semua logika back-end dieksekusi, lalu dapat mengembalikan View (tampilan) ke browser. Tampilan ini disimpan dalam Caching agar pemrosesan setelahnya lebih cepat.

app = Folder ini berisi controller, Model, Middleware, dan Provider

bootstrap = Folder ini berisi Cache untuk mempercepat pemrosesan

config = Folder ini berisi semua file konfigurasi untuk aplikasi

database = Folder ini berisi file-file migrasi dari/ke database

Public = Folder ini berisi file yang dapat diakses langsung, biasanya file asset (gambar, CSS dan JS) atau file konten (file untuk di-download)

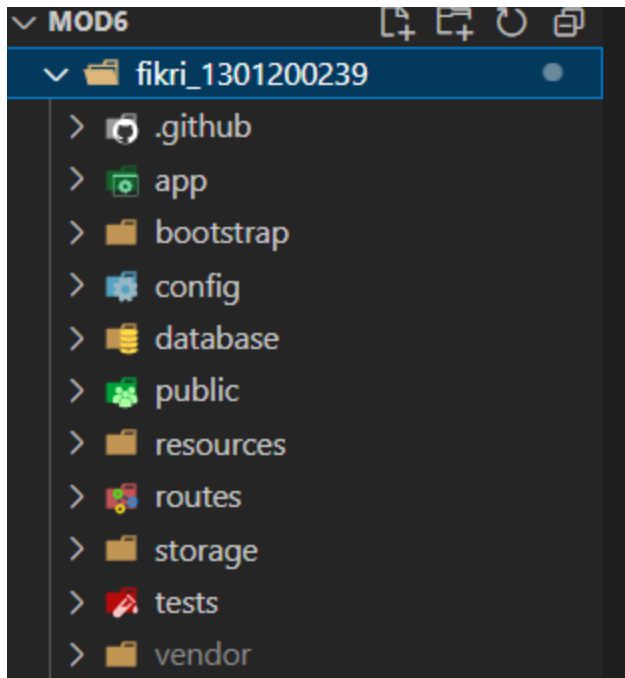
Resources = Folder ini berisi View (tampilan)

Routes = Folder ini berisi Routing untuk pemetaan URL ke aplikasi

Storage = Folder ini berisi hasil kompilasi View dan log dari aplikasi

Tests = Folder ini berisi file – file untuk unit testing

Vendor = Folder ini berisi file – file library yang dibutuhkan aplikasi



6.3.1 Routing

Pengaturan Routing berada pada file routes/web.php. Disini kita dapat mengatur pemetaan URL dengan aksi yang ingin kita lakukan. Method nya pun dapat diatur jika hanya untuk menerima method tertentu saja (post, get, atau yang lainnya).

```
Route::get('/', function () {  
    return view('welcome');  
});  
//Fikri Maulana - 1301200239  
Route::get('/halaman-user', [UserController::class, 'index']) -> middleware('auth');  
Route::get('/user', [UserController::class, 'index']) -> middleware('auth');  
Route::get('/tabel-user', [UserController::class, 'tabelUser']) -> middleware('auth');  
Route::get('/tambah-user', [UserController::class, 'create']) -> middleware('auth');  
Route::get('/upload-user', [UserController::class, 'store']) -> middleware('auth');  
Route::get('/user/edit/{id}', [UserController::class, 'edit']) -> middleware('auth');  
Route::get('/user/hapus/{id}', [UserController::class, 'destroy']) -> middleware('auth');
```

6.3.2 View

Menggunakan template engine, yaitu Blade. Sehingga, setiap file tampilan yang kita buat harus mengikuti penamaan namaview.blade.php dan diletakkan pada folder resources/views agar dapat di-load oleh fungsi view().

```
fikri_1301200239 > resources > views > welcome.blade.php
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6
7 <title>Laravel</title>
8 <!-- Fikri Maulana 1301200239 -->
9
10 <!-- Fonts -->
11 <link rel="preconnect" href="https://fonts.bunny.net">
12 <link href="https://fonts.bunny.net/css?family=figtree:400,600&display=swap" rel="stylesheet" />
13
14 <!-- Styles -->
15 <style>
16 /* ! tailwindcss v3.2.4 | MIT License | https://tailwindcss.com */
17 </style>
18 </head>
19 <body class="antialiased">
```

6.3.3 Controller

File Controller diletakkan pada folder `app/Http/Controllers`. Cara membuatnya bisa dengan cara pembuatan file manual dengan memperhatikan hal-hal berikut: meng-extend base Controller dari Laravel atau turunannya dan dituliskan namespace `"App\Http\Controllers"` agar dapat digunakan oleh file lain dan dipetakan dari Routing.

fikri_1301200239 > app > Http > Controllers > Controller.php

```
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
6  use Illuminate\Foundation\Validation\ValidatesRequests;
7  use Illuminate\Routing\Controller as BaseController;
8
9  class Controller extends BaseController
10 {
11     use AuthorizesRequests, ValidatesRequests;
12     // Fikri Maulana 1301200239
13     public function index(){
14         return view('content.login.loginUser');
15     }
16
17     public function auth(Request $request){
18         $credentials = $request->validate([
19             'email' => 'required',
20             'password' => 'required'
21         ]);
22
23         if(Auth::attempt($credentials) || Auth::attempt([
24             'email' => $request->email,
25             'password' => $request->password
26         ])){
27             return redirect()->intended('/halaman-user')->with(['succes' => 'Login Succesfull!']);
28         } else {
29             return back()->with('loginError', 'Wrong email or password!');
30         }
31     }
32 }
33
```

```
33     public function register(){
34         return view('content.register');
35     }
36 }
37
```