

6.1. Framework & MVC

Framework adalah kerangka kerja yang digunakan sebagai dasar dalam membangun aplikasi kompleks. Berisi aturan dan kode-kode yang telah ditentukan sebelumnya, framework membantu mempercepat proses pembuatan aplikasi dengan menyediakan modul-modul dan fungsi-fungsi siap pakai. Dengan menggunakan framework, pengembang dapat meningkatkan produktivitas dan mengurangi biaya serta waktu pembuatan aplikasi.

MVC merupakan sebuah arsitektur perangkat lunak yang digunakan dalam pengembangan aplikasi, yang membagi aplikasi menjadi tiga komponen utama yaitu:

- Model bertanggung jawab untuk memproses data.
- View bertanggung jawab untuk menampilkan data dan interaksi pengguna.
- Controller mengatur interaksi antara Model dan View.

Dengan memisahkan tampilan aplikasi dari pemrosesan data dan pengendali interaksi, MVC memungkinkan pengembangan aplikasi yang lebih terorganisir dan mudah dikelola. Arsitektur MVC juga sering digunakan dalam pengembangan web untuk memisahkan kode HTML, CSS, dan JavaScript dari kode server-side, sehingga memungkinkan perubahan tampilan tanpa harus memodifikasi kode server-side.

6.2. Pengenalan Laravel

Laravel adalah sebuah framework aplikasi web yang bersifat open source dan ditulis dalam bahasa pemrograman PHP. Framework ini menyediakan berbagai fitur dan alat yang membantu pengembangan aplikasi web dengan cepat dan mudah, termasuk manajemen basis data, autentikasi pengguna, dan pembuatan API. Laravel menggunakan konsep arsitektur Model-View-Controller (MVC) untuk memisahkan tampilan aplikasi dari pemrosesan data dan pengendali interaksi. Selain itu, Laravel juga dilengkapi dengan sistem migrasi basis data yang memudahkan pengembang dalam mengelola struktur basis data dan melakukan pembaruan secara otomatis. Framework ini populer dalam pengembangan aplikasi web modern dan telah digunakan oleh banyak perusahaan dan pengembang di seluruh dunia.

6.3. Cara Kerja Laravel

Laravel bekerja berdasarkan konsep arsitektur Model-View-Controller (MVC) yang memisahkan tampilan aplikasi (View) dari pemrosesan data (Model) dan pengendali interaksi (Controller). Proses kerja Laravel dapat dijelaskan sebagai berikut:

1. Routing: Setiap permintaan HTTP akan melewati sistem routing di Laravel. Sistem routing akan menentukan alamat URL yang diminta dan menentukan fungsi Controller mana yang harus dijalankan.

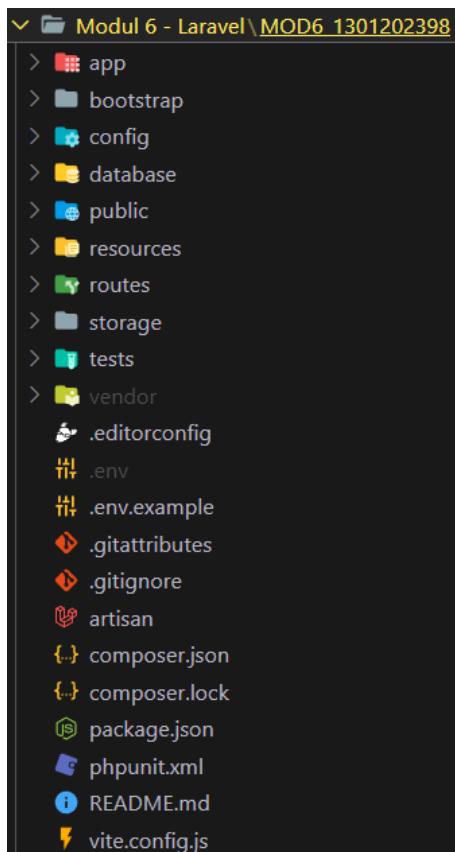
Nama: Muhamad Fachri Haikal
NIM : 1301202398
Kelas : IF4401
Praktikum ABP MOD 6 Laravel

2. Controller: Setelah sistem routing menentukan fungsi Controller yang harus dijalankan, Controller akan memproses permintaan dan memanipulasi data melalui Model.
3. Model: Model adalah representasi dari basis data yang digunakan oleh aplikasi. Model digunakan untuk memproses data seperti membaca, menulis, atau menghapus data dari basis data.
4. View: Setelah data diolah oleh Model, Controller akan memberikan data tersebut ke View untuk ditampilkan kepada pengguna. View bertanggung jawab untuk menampilkan data ke pengguna dalam bentuk tampilan halaman web.
5. Response: Setelah View selesai menampilkan data, Laravel akan mengirimkan respons HTTP ke pengguna, yang berisi tampilan halaman web yang diminta.

Untuk membuat project Laravel, pastikan sudah menginstall composer. Kemudian pada terminal/cmd, gunakan command “composer create-project laravel/laravel <nama_project>”.

```
C:\Users\haikal\Documents\Kuliah\Semester 6\Aplikasi Berbasis Platform [GGM]\Modul 6 - Laravel>composer create-project laravel/laravel "MOD6_1301202398"
Creating a "laravel/laravel" project at "./MOD6_1301202398"
Info from https://repo.packagist.org: #StandWithUkraine
Installing laravel/laravel (v10.0.4)
- Downloading laravel/laravel (v10.0.4)
- Installing laravel/laravel (v10.0.4): Extracting archive
Created project in C:\Users\haikal\Documents\Kuliah\Semester 6\Aplikasi Berbasis Platform [GGM]\Modul 6 - Laravel\MOD6_1301202398
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies
Lock file operations: 106 installs, 0 updates, 0 removals
```

Berikut adalah hasil dari project Laravel yang telah dibuat.



Berikut penjelasan untuk setiap folder yang terbuat:

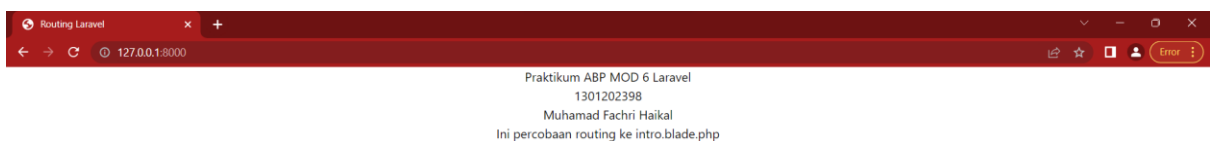
Nama: Muhamad Fachri Haikal
NIM : 1301202398
Kelas : IF4401
Praktikum ABP MOD 6 Laravel

1. app: folder ini berisi controller, model, middleware, dan provider.
2. bootstrap: folder ini berisi cache untuk mempercepat pemrosesan.
3. config: folder ini berisi semua file konfigurasi untuk aplikasi.
4. database: folder ini berisi file-file migrasi dari/ke database.
5. public: folder ini berisi file yang dapat diakses langsung, biasanya file asset (gambar, css, dan js) atau file konten (file untuk di-download).
6. resources: folder ini berisi view (tampilan).
7. routes: folder ini berisi routing untuk pemetaan url ke aplikasi.
8. storage: folder ini berisi hasil kompilasi view dan log dari aplikasi.
9. tests: folder ini berisi file-file untuk unit testing.
10. vendor: folder ini berisi file-file library yang dibutuhkan aplikasi.

6.3.1.Routing

```
Modul 6 - Laravel > MOD6_1301202398 > resources > views > intro.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3
4 <head>
5     <meta charset="utf-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7
8     <title>Routing Laravel</title>
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    <script src="https://cdn.tailwindcss.com"></script>
16 </head>
17 <body>
18     <div class="container mx-auto text-center">
19         <h1>Praktikum ABP MOD 6 Laravel</h1>
20         <p>1301202398</p>
21         <p>Muhamad Fachri Haikal</p>
22         <p>Ini percobaan routing ke intro.blade.php</p>
23     </div>
24 </body>
25 </html>
```

```
Route::get('/', function(){
    return view('intro');
});
```

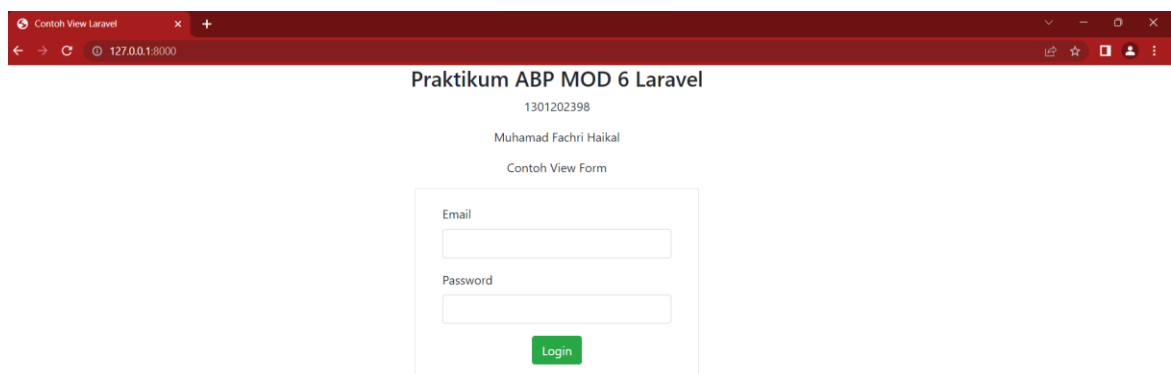


Dilakukan proses routing saat pertama kali proyek dijalankan ke halaman atau file intro.blade.php yang memiliki tampilan seperti diatas

Nama: Muhamad Fachri Haikal
NIM : 1301202398
Kelas : IF4401
Praktikum ABP MOD 6 Laravel
6.3.2. View

```
Modul 6 - Laravel > MOD6_1301202398 > resources > views > contohView.blade.php > html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Contoh View Laravel</title>
5 <link rel="stylesheet"
6 href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"
7 integrity="sha384-zCbKRCUGaJDkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdKlf6/eSrtxUkn"
8 crossorigin="anonymous">
9 <script src="{{ asset('assets/jquery.js') }}"></script>
10 <script
11 src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.bundle.min.js"
12 integrity="sha384-fQybjgWLrvvRgtW6bFLB7jaZrFsaBXjsOMm/tB9LTS58ONXgqbR9W8oWht/ampnF"
13 crossorigin="anonymous"></script>
14 </head>
15 <body style="width:95%">
16 <div class="container mx-auto text-center">
17 <h3>Praktikum ABP MOD 6 Laravel</h3>
18 <p>1301202398</p>
19 <p>Muhamad Fachri Haikal</p>
20 <p>Contoh View Form</p>
21 </div>
22 <div class="row justify-content-center">
23 <div class="col-3 border">
24 <form style="margin:20px" method="POST" action="/auth">
25 @csrf
26 <div class="form-group">
27 <label for="email">Email</label>
28 <input type="email" name="em" class="form-control" />
29 </div>
30 <div class="form-group">
31 <label>Password</label>
32 <input type="password" name="pwd" class="form-control" />
33 </div>
34 <div style="text-align:center">
35 <button class="btn btn-success">Login</button>
36 </div>
37 </form>
38 </div>
39 </div>
40 </body>
41 </html>
```

```
Route::get('/', function(){
    return view('contohView');
});
```



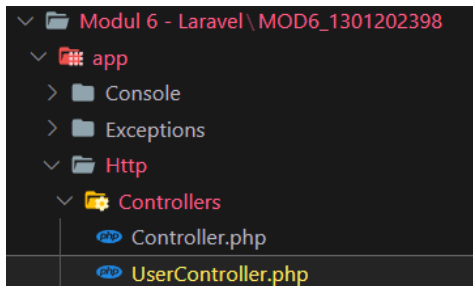
Di atas adalah contoh View dengan nama contohView.blade.php pada folder resources/views/ yang berisi laman form login yang dipanggil dengan fungsi view() pada web.php.

Nama: Muhamad Fachri Haikal
NIM : 1301202398
Kelas : IF4401
Praktikum ABP MOD 6 Laravel

6.3.3.Controller

```
C:\Users\haikal\Documents\Kuliah\Semester 6\Aplikasi Berbasis Platform [GGM]\Modul 6 - Laravel\MOD6_1301202398>php artisan make:controller UserController --resource
INFO Controller [C:\Users\haikal\Documents\Kuliah\Semester 6\Aplikasi Berbasis Platform [GGM]\Modul 6 - Laravel\MOD6_1301202398\app\Http\Controllers\UserController.php] created successfully.
```

Untuk membuat controller, gunakan command “php artisan make::controller <namaController>”. Lalu untuk opsi –resource berguna untuk membuat fungsi-fungsi default yang akan digunakan untuk manajemen data di-generate secara otomatis oleh Laravel.



Untuk melihat controller yang sudah dibuat, kita dapat pergi ke folder app/Http/Controllers.

```
Modul 6 - Laravel > MOD6_1301202398 > app > Http > Controllers > UserController.php >
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  /**
8   * @method GET
9   */
10  class UserController extends Controller
11  {
12      /**
13       * Display a listing of the resource.
14       *
15       * @return \Illuminate\Http\Response
16       */
17      public function index()
18      {
19          //
20      }
21
22      /**
23       * Show the form for creating a new resource.
24       *
25       * @return \Illuminate\Http\Response
26       */
27      public function create()
28      {
29          //
30      }
31
32      /**
33       * Store a newly created resource in storage.
34       *
35       * @param \Illuminate\Http\Request $request
36       * @return \Illuminate\Http\Response
37       */
38      public function store(Request $request)
39      {
40          //
41      }
42
43      /**
44       * Display the specified resource.
45       *
46       * @param int $id
47       * @return \Illuminate\Http\Response
48       */
49      public function show(string $id)
50      {
51          //
52      }
53
54      /**
55       * Show the form for editing the specified resource.
56       *
57       * @param int $id
58       * @return \Illuminate\Http\Response
59       */
60      public function edit(string $id)
61      {
62          //
63      }
64
65      /**
66       * Update the specified resource in storage.
67       *
68       * @param \Illuminate\Http\Request $request
69       * @param int $id
70       * @return \Illuminate\Http\Response
71       */
72      public function update(Request $request, string $id)
73      {
74          //
75      }
76
77      /**
78       * Remove the specified resource from storage.
79       *
80       * @param int $id
81       * @return \Illuminate\Http\Response
82       */
83      public function destroy(string $id)
84      {
85          //
86      }
87  }
```

Di atas adalah UserController yang dibuat menggunakan opsi –resource sehingga controller siap untuk mengimplementasi operasi CRUD (Create, Read, Update, Delete) dengan sangat mudah dan efisien.