

LAPORAN PRAKTIKUM PEMROGRAMAN WEB 2

PERTEMUAN 3

ROUTING, VIEW, DAN BLADE LARAVEL



Disusun Oleh:

Wildan Dzaky Ramadhani

22/505766/SV/21917

Dosen Pengampu:

Dinar Nugroho Pratomo, S.Kom., M.IM., M.Cs.

Faza Maula Azif, S.Kom., M.Eng.

**PROGRAM STUDI D4 TEKNOLOGI REKAYASA PERANGKAT  
LUNAKDEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA  
SEKOLAH VOKASI  
UNIVERSITAS GADJAH  
MADAYOGYAKARTA**

**2023**

## DAFTAR ISI

DAFTAR ISI.....	i
DAFTAR GAMBAR.....	ii
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
BAB II.....	3
2.1. Laravel.....	3
2.2. Routing.....	3
2.3. Views.....	4
2.4. Blade Laravel.....	4
BAB III.....	5
3.1. Routing dan View.....	5
3.2. Latihan Praktikum.....	7
3.3. Blade Laravel.....	11
3.4. Tugas Praktikum.....	12
BAB IV.....	15
4.1. Kesimpulan.....	15
DAFTAR PUSTAKA.....	16

## DAFTAR GAMBAR

Gambar 1 Tampilan folder routing.....	5
Gambar 2 Tampilan folder views.....	6
Gambar 3 Perintah menjalankan projek laravel.....	6
Gambar 4 Tampilan isi dari file web.php.....	7
Gambar 5 Tampilan web dari projek kita.....	7
Gambar 6 File about.blade.php.....	7
Gambar 7 Memasukkan perintah Route::get('/')......	8
Gambar 8 Kode HTML di file about.blade.php.....	8
Gambar 9 Folder css dan js.....	9
Gambar 10 File css.....	9
Gambar 11 File js.....	9
Gambar 12 URL file about.....	9
Gambar 13 Inputkan nama.....	10
Gambar 14 Hasil alert dari kode js.....	10
Gambar 15 Tampilan web pada halaman about.....	10
Gambar 16 Tampilan layout.blade.php.....	12
Gambar 17 Tampilan profile.blade.php.....	13
Gambar 18 Tampilan web dari file profile.....	13
Gambar 19 Tampilan tombol tampilkan umur.....	14

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Pemrograman web modern kerap memerlukan penanganan rute (routing), pengaturan tampilan (view), dan penerapan template (blade) guna menghasilkan halaman web yang dinamis. Namun, mahasiswa kerap menghadapi tantangan dalam memahami serta mengimplementasikan konsep-konsep ini. Salah satu kesulitan yang umum adalah kurangnya pemahaman tentang bagaimana mengatur rute secara efisien, menghubungkan data ke tampilan, dan menggunakan template untuk memisahkan logika dari tampilan.

Laravel, sebagai salah satu framework PHP terdepan, menawarkan solusi yang kuat untuk masalah yang berkaitan dengan routing, view, dan blade. Dengan konfigurasi routing yang mudah, Laravel memungkinkan para pengembang untuk dengan cepat menentukan bagaimana permintaan HTTP akan ditangani. Di sisi lain, kemampuan Blade dalam memisahkan antara tampilan dan logika memungkinkan pengembangan web menjadi lebih terstruktur dan lebih mudah dikelola. Dengan penerapan yang tepat, Laravel membantu mengatasi kesulitan-kesulitan kompleks ini secara efisien, mempercepat proses pengembangan, dan meningkatkan kualitas kode.

Praktikum pemrograman web yang membahas routing, tampilan, dan blade dalam konteks Laravel memiliki peranan penting dalam memahami dasar-dasar pengembangan web modern. Ini memberikan dasar kuat bagi mahasiswa untuk memahami bagaimana sebuah aplikasi web beroperasi, bagaimana data disajikan kepada pengguna, dan bagaimana mengelola logika aplikasi dengan efisien. Pemahaman yang kuat tentang konsep-konsep ini menjadi landasan yang kokoh untuk langkah-langkah pengembangan selanjutnya dan berperan dalam meningkatkan daya saing di pasar kerja yang semakin kompetitif.

Diharapkan bahwa melalui praktikum ini, mahasiswa akan mendapatkan pemahaman yang mendalam tentang routing, tampilan, dan Blade Laravel. Harapannya, mahasiswa dapat menerapkan pengetahuan ini dalam pengembangan web nyata, mengatasi berbagai masalah, dan menciptakan pengalaman pengguna yang lebih baik. Selain itu, praktikum ini diharapkan akan memotivasi mahasiswa untuk terus mengeksplorasi pengetahuan dalam pengembangan web dan terus mengembangkan keterampilan mereka. Dengan penguasaan yang kuat dalam hal routing, view, dan Blade Laravel, mahasiswa akan siap menghadapi tantangan dalam dunia pengembangan web yang terus berubah dan berkembang.

## **1.2.Rumusan Masalah**

1. Bagaimana cara memperoleh pemahaman dan mengatasi tantangan yang berkaitan dengan Routing, View, dan Blade dalam proses pengembangan web? Selain itu, bagaimana penggunaan framework Laravel dapat mendukung para mahasiswa dalam menghadapi kendala-kendala tersebut?

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Laravel**

Laravel merupakan salah satu framework PHP yang sangat populer dalam pengembangan aplikasi web modern. Framework ini mempermudah proses pengembangan dengan panduan dan alat yang jelas. Salah satu keunggulan utama dari Laravel adalah sistem routing yang mempermudah penanganan permintaan HTTP. Hal ini memungkinkan para pengembang untuk dengan tegas menentukan bagaimana permintaan URL akan diarahkan ke bagian yang sesuai dalam aplikasi.

Selain itu, Laravel juga dilengkapi dengan sistem ORM yang kuat, yang dikenal sebagai Eloquent. Sistem ini memudahkan interaksi dengan basis data menggunakan objek-objek dalam bahasa PHP. Dengan adanya Eloquent, kebutuhan untuk menulis query SQL secara manual dapat dihindari. Karena fitur-fitur yang dimilikinya, Laravel menjadi pilihan utama para pengembang web yang ingin membangun aplikasi web yang tangguh dan efisien.

#### **2.2. Routing**

Laravel muncul sebagai salah satu framework PHP yang meraih popularitas tinggi dalam dunia pengembangan aplikasi web modern. Framework ini merupakan alat yang sangat membantu dalam menyederhanakan proses pengembangan melalui pedoman dan instrumen yang jelas. Salah satu poin keunggulan terbesar yang dimiliki oleh Laravel adalah sistem routing yang berperan penting dalam penanganan permintaan HTTP. Fasilitas ini memungkinkan para pengembang untuk dengan tegas menentukan bagaimana setiap permintaan URL akan diarahkan ke bagian yang tepat dalam aplikasi.

Selain fitur routing yang kuat, Laravel juga dilengkapi dengan sistem ORM yang bertenaga, yang dikenal dengan nama Eloquent. Sistem ini mempermudah interaksi dengan basis data dengan menggunakan konsep objek-objek dalam bahasa PHP. Dengan adanya Eloquent, kebutuhan untuk secara manual menulis query SQL dapat dihindari. Karena keterampilan-keterampilan ini, Laravel menjadi pilihan utama para pengembang web yang ingin membangun aplikasi dengan kekokohan dan efisiensi yang tinggi.

### **2.3. Views**

Tampilan (views) merupakan komponen penting dalam kerangka kerja (framework) Laravel yang memiliki tanggung jawab utama dalam menampilkan data yang diterima dari kontroler, yang berasal dari model. Mayoritas isi dari file tampilan terdiri dari kode HTML yang akan diinterpretasikan oleh peramban web (web browser), kemudian ditampilkan kepada pengguna. Laravel memiliki manajemen struktur tampilan yang terorganisir, dan umumnya direktori tampilan ditempatkan di dalam folder `resources/views`. Dalam penggunaannya, tampilan memisahkan secara tegas antara logika aplikasi dan elemen tampilan, yang berkontribusi pada keteraturan pengembangan web dan mempermudah pembuatan antarmuka pengguna yang dinamis. Blade, sistem templating yang digunakan oleh Laravel, juga mengizinkan penggunaan kontrol struktur serta inheritance untuk menciptakan tampilan yang lebih dinamis dan mudah dikelola.

### **2.4. Blade Laravel**

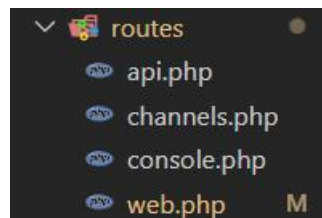
Blade, sebagai komponen yang tak terpisahkan dari Laravel, berfungsi sebagai mesin template yang mempermudah proses pembuatan template dalam pengembangan aplikasi. Ini memberikan kesempatan kepada pengembang untuk merancang tampilan aplikasi dengan sifat dinamis tanpa harus mengulangi kode statis. Blade menjaga tampilan sebagai berkas `.blade.php` dalam direktori `resources/views` dan mengadopsinya dengan menggunakan prinsip inheritance serta section.

## BAB III

### DOKUMENTASI PRAKTIKUM

#### 3.1. Routing dan View

Pada Laravel, routing adalah sistem yang mengatur alamat atau URL yang digunakan untuk mengakses halaman web tertentu, memungkinkan perpindahan antara halaman-halaman dengan lebih mudah melalui browser web. Di dalam Laravel, kita akan menemukan routing terletak dalam direktori bernama 'routes'. Dalam konteks pengembangan proyek, routes dapat diartikan sebagai jalur atau rute yang harus diikuti untuk menjalankan berbagai bagian dari proyek, dan masing-masing bagian ini memiliki peranannya sendiri.



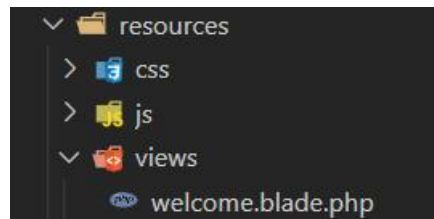
*Gambar 1 Tampilan folder routing*

Pada gambar diatas, dapat kita lihat bahwa pada folder routes terdapat 4 file yang sudah tergenerate oleh Laravel, yaitu :

- api.php : Berkas ini digunakan untuk membuat routing khusus API. Di dalamnya, kita dapat mendefinisikan rute-rute yang akan digunakan untuk layanan inti API dengan menggunakan fitur-fitur Laravel yang kuat.
- channels.php : Berkas ini digunakan untuk membuat routing yang berkaitan dengan broadcasting event, seperti notifikasi. Routing di sini memungkinkan aplikasi untuk mengirimkan pesan secara real-time kepada pengguna.
- console.php : Berkas ini digunakan untuk membuat routing yang berkaitan dengan perintah-perintah yang dapat dijalankan melalui terminal. Dengan adanya routing ini, kita dapat membuat perintah artisan kustom sesuai kebutuhan aplikasi.
- web.php : Berkas ini digunakan untuk membuat routing untuk aplikasi web biasa. Di dalamnya, kita mendefinisikan rute-rute yang digunakan untuk mengatur tampilan dan perilaku aplikasi web yang diakses oleh pengguna melalui peramban web.



Sedangkan views adalah tampilan halaman yang ingin ditampilkan. Folder views terletak pada folder resources, dan didalamnya sudah tergenerate file yang bernama welcome.blade.php.



*Gambar 2 Tampilan folder views*

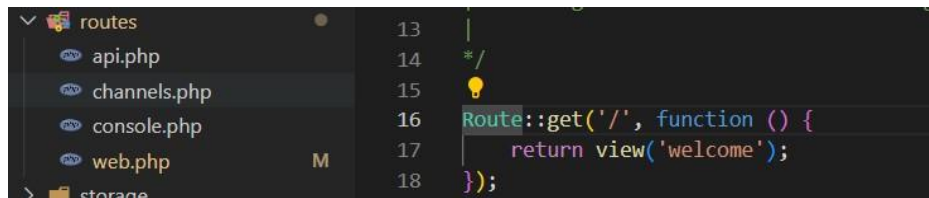
Gambar diatas merupakan struktur folder yang telah dihasilkan oleh Laravel, termasuk folder resources. Dalam direktori resources ini, kita dapat menemukan direktori views yang berfungsi sebagai tempat untuk menyimpan semua tampilan aplikasi. Di dalamnya juga terdapat file welcome.blade.php yang akan tampil ketika kita pertama kali menjalankan Laravel. Oleh karena itu, saat ingin mengakses halaman melalui web browser, fokus dapat diberikan pada file web.blade.php, kita bisa membuat file baru juga dengan format nama\_file.blade.php sesuai kebutuhan.

Selanjutnya kita dapat menjalankan perintah “php artisan serve” untuk menjalankan proyek Laravel kita, dimana nantinya kita akan diberikan “url host” untuk menjalankan proyek kita seperti pada gambar di bawah ini.

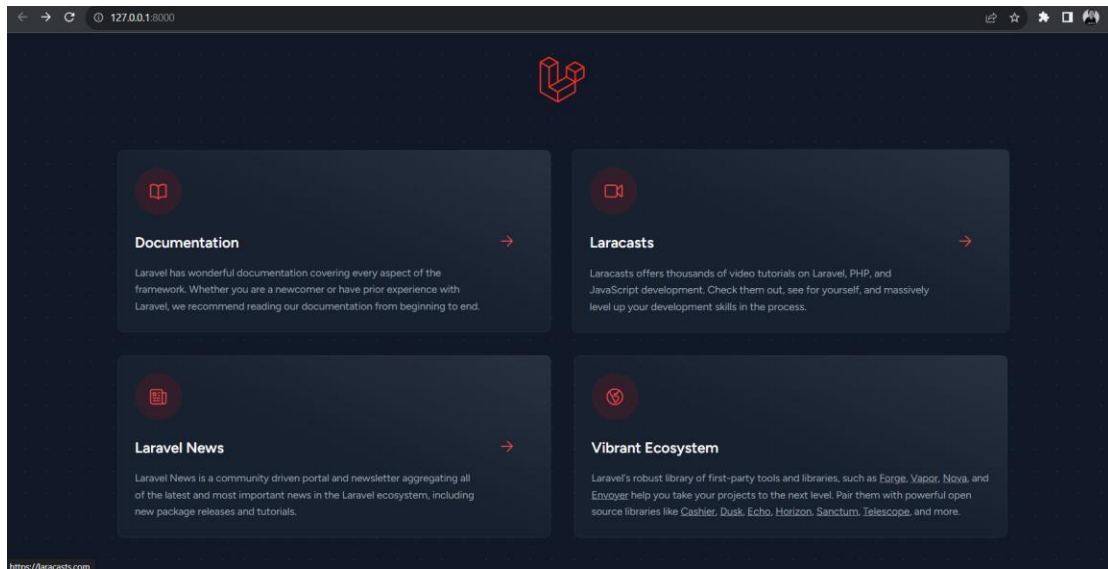
```
PS C:\xampp\htdocs\example-app> php artisan serve
INFO Server running on [http://127.0.0.1:8000].
Press Ctrl+C to stop the server
```

*Gambar 3 Perintah menjalankan proyek laravel*

Setelah kita mendapatkan alamat server diatas, kita dapat menjalankan proyek kita yang berada dalam direktori views ‘welcome.blade.php’ dengan melalui routes yang berada pada ‘web.php’ melalui alamat server diatas. Di dalam berkas ‘web.php’ terdapat perintah Route::get('/') yang berfungsi sebagai halaman awal saat kita baru saja menginstal Laravel. Berikut ini adalah tampilan kode dan tampilan webnya.



Gambar 4 Tampilan isi dari file web.php

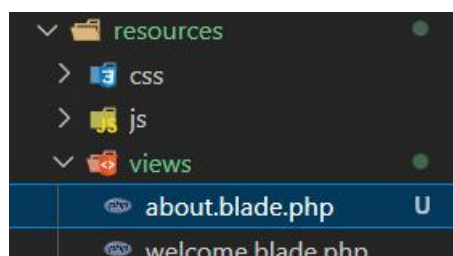


Gambar 5 Tampilan web dari projek kita

### 3.2. Latihan Praktikum

Berikut ini merupakan contoh sederhana menampilkan teks di web browser dengan menambahkan parameter menggunakan konsep routing.

- Pertama, buat file di dalam folder views dengan format nama\_file.blade.php. Seperti contoh di bawah kita akan membuat file dengan nama about.blade.php



Gambar 6 File about.blade.php

- Selanjutnya kita akan masukkan perintah 'route::get('/')' di dalam file 'web.php' yang terdapat pada folder routes, kemudian tambahkan nama halaman setelah tanda ('/')

sesuai dengan nama file yang telah kita buat di dalam folder views sebelumnya 'about.blade.php'. Kita juga dapat menambahkan data didalamnya dengan menggunakan [] dengan menuliskan key dan value yang dapat dihubungkan dengan tanda =>. Pastikan lagi bahwa nama route sama dengan nama file yang akan ditulis dalam folder views.

```
20 Route::get('/about', function () {
21     return view('about', [
22         "name" => "lala",
23         "email" => "lala@gmail.com",
24         "address" => "Sendowo, Yogyakarta",
25         "number" => "123-456-789"
26     ]);
27 });
28
```

Gambar 7 Memasukkan perintah Route::get('/')

- Selanjutnya, kita tuliskan kode HTML di file about.blade.php dan masukkan data tadi di dalam body. Untuk menuliskan data kita cukup memasukkan kode dan key '<?= kode ?>'.

```
resources > views > about.blade.php
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <link rel="stylesheet" href="/css/style.css">
7     <title>About</title>
8 </head>
9 <body>
10     <h1>Halaman about</h1>
11     <h1> <?= $name; ?></h1>
12     <h1> <?= $email; ?></h1>
13     <h1> <?= $address; ?></h1>
14     <h1> <?= $number; ?></h1>
15 <script src="/js/script.js"></script>
16 </body>
17 </html>
```

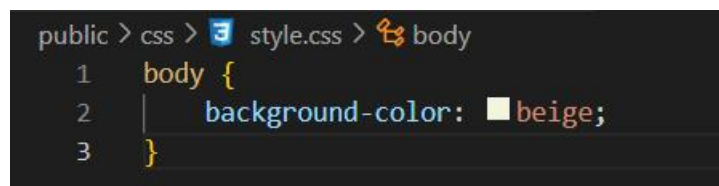
Gambar 8 Kode HTML di file about.blade.php

Kita juga dapat menambahkan file css dan javascript dengan menambahkan link untuk memanggil file css dan javascript, dimana file css dan javascript akan disimpan di folder public. Berikut ini merupakan contoh isi dalam file css dan javascript.



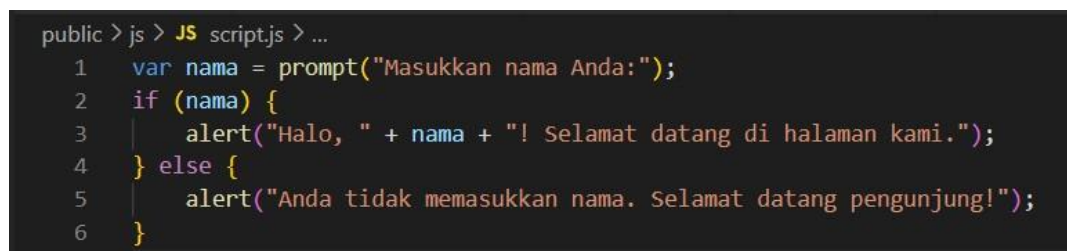
Gambar 9 Folder css dan js

Gambar diatas merupakan folder css dan js yang ditempatkan dalam folder public.



Gambar 10 File css

Gambar diatas merupakan isi dari file css yang akan digunakan dalam file html, dimana kode diatas akan menampilkan background color berwarna beige.



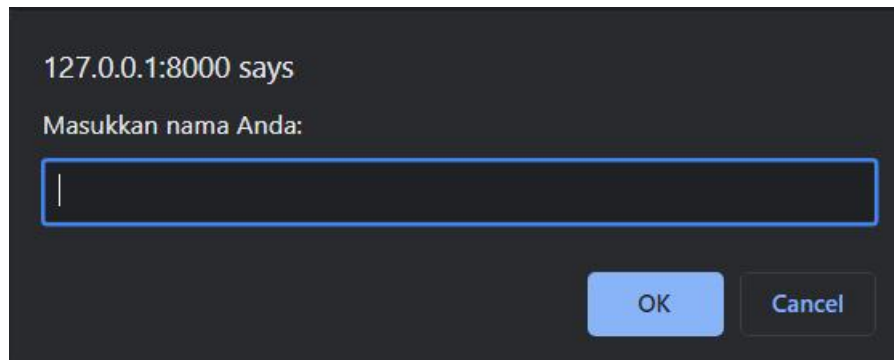
Gambar 11 File js

Gambar diatas merupakan isi dari file js yang akan digunakan dalam file html, dimana kode diatas akan menampilkan pop up dan kita akan diminta untuk mengisikan nama kita, setelah kita mengisikan nama kita, maka akan terdapat alert pop up yang berisikan kode diatas. Kemudian kita akan membuka URL dengan menambahkan nama file yang kita miliki '127.0.0.1:8000/about'.

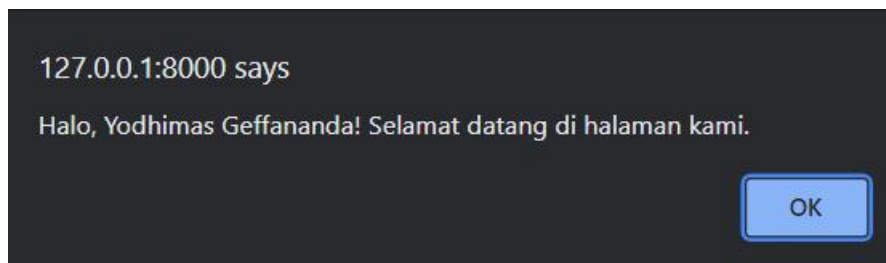


Gambar 12 URL file about

Kemudian browser akan mengarahkan ke halaman yang dituju yaitu di halaman about dan memuat program yang telah dibuat. Contohnya seperti dibawah ini



*Gambar 13 Inputkan nama*



*Gambar 14 Hasil alert dari kode js*

Pertama kita akan menjumpai alert dari kode js yang telah kita buat seperti pada gambar diatas. Berikut ini merupakan hasil dari web yang telah kita buat pada halaman about.



*Gambar 15 Tampilan web pada halaman about*

### 3.3. Blade Laravel

Blade adalah template engine yang disertakan dalam Laravel untuk memfasilitasi proses templating dalam pengembangan aplikasi Laravel. Blade memiliki beberapa fitur yang sangat bermanfaat dalam memudahkan pengembangan tampilan aplikasi. Template engine ini memungkinkan pengembang untuk membuat kode program secara dinamis, tanpa perlu menyalin kode statis berulang-ulang.

Blade bekerja dengan cara mengkompilasi semua tampilan ke dalam kode PHP biasa. Hasil kompilasi ini kemudian disimpan dalam cache sehingga tidak menambah beban atau overhead saat aplikasi dijalankan. Anda dapat mengidentifikasi file Blade dengan ekstensi `.blade.php`, dan biasanya disimpan dalam direktori `resources/views`.

Salah satu fitur utama Blade adalah penggunaan dasar template inheritance dan section. Dengan menggunakan Blade, Anda dapat dengan mudah mengelola tampilan berulang seperti header, footer, sidebar, dan lainnya tanpa harus membuatnya berulang kali secara statis, yang dapat menyebabkan inkonsistensi dalam kode.

Cara kerja Blade dimulai dengan membuat file master template yang berisi layout dasar. Layout ini biasanya mencakup elemen-elemen tampilan yang konsisten di seluruh aplikasi. Kemudian, pada halaman-halaman lain yang menggunakan layout yang sama, Anda dapat menggunakan fungsi Blade seperti `@extends()`, `@section()`, dan `@yield()`.

- `@yield()` digunakan untuk menentukan bagian-bagian tertentu pada template website yang dapat diisi dengan konten dinamis. Anda dapat menentukan nama bagian yang akan diisi dalam parameter `@yield()`.
- `@include` mirip dengan konsep include dalam PHP dasar. Fungsi ini memungkinkan Anda untuk menyertakan tampilan "partial" ke dalam tampilan view yang dituju.
- `@extends()` memungkinkan Anda untuk memperluas sebuah template dengan mendefinisikan bagian-bagian tambahan sesuai kebutuhan. Template yang diperluas (extended) akan menentukan "section" sendiri dengan menggunakan fungsi `@yield()`, yang kemudian dapat diisi dengan konten dari file view.

Dengan Blade, pengembang dapat menciptakan tampilan yang dinamis dan modular dengan mudah, meningkatkan produktivitas dalam pengembangan aplikasi Laravel.

### 3.4. Tugas Praktikum

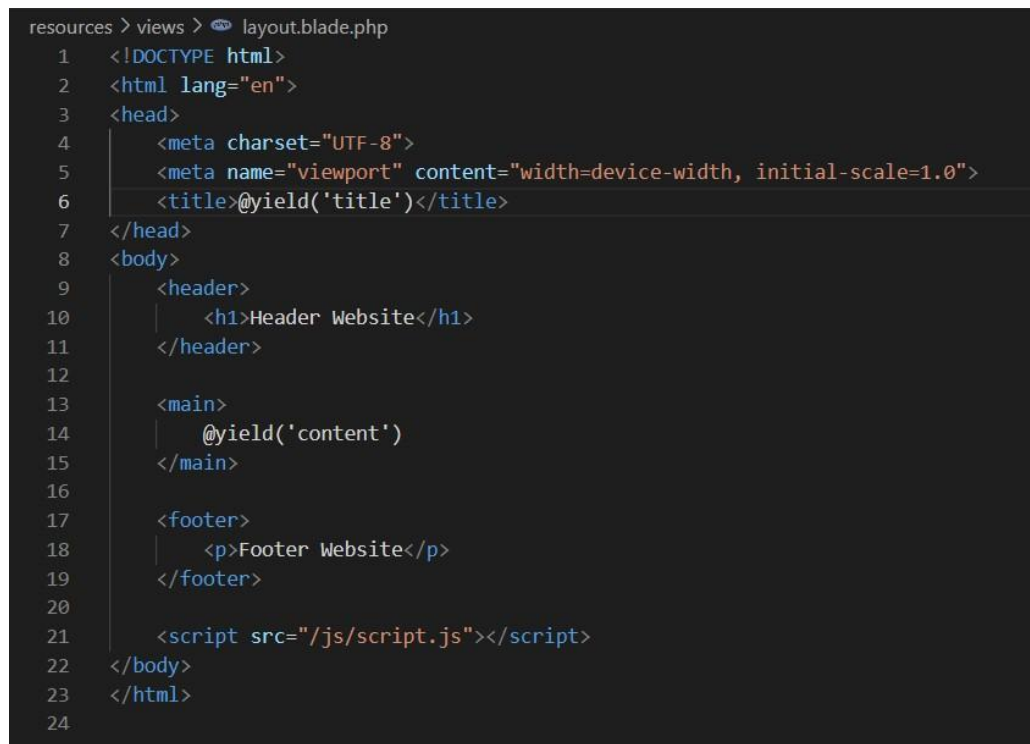
Melakukan pembuatan routing, view, dan blade sederhana seperti yang ditunjukkan dari Latihan praktikum pada proyek Laravel.

Tugas Praktikum juga dapat dilihat pada github berikut ini :

<https://github.com/wildanzake/ppwpertemuan2/>

Berikut ini merupakan penjelasan dari tugas praktikum yang saya buat.

- Pertama, kita akan membuat layout Blade yang akan digunakan sebagai kerangka umum untuk halaman-halaman dalam proyek kita. Buat file baru dengan nama layout.blade.php di dalam direktori resources/views.



```
resources > views > layout.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>@yield('title')</title>
7  </head>
8  <body>
9      <header>
10         <h1>Header Website</h1>
11     </header>
12
13     <main>
14         @yield('content')
15     </main>
16
17     <footer>
18         <p>Footer Website</p>
19     </footer>
20
21     <script src="/js/script.js"></script>
22 </body>
23 </html>
24
```

*Gambar 16 Tampilan layout.blade.php*

Pada layout di atas, kita menggunakan @yield untuk menentukan tempat di mana konten halaman spesifik akan disisipkan. @yield('title') digunakan untuk judul halaman, dan @yield('content') digunakan untuk isi halaman.

- Sekarang kita akan menggunakan layout Blade yang sudah dibuat untuk halaman "Profile". Buat file profile.blade.php di dalam direktori resources/views.

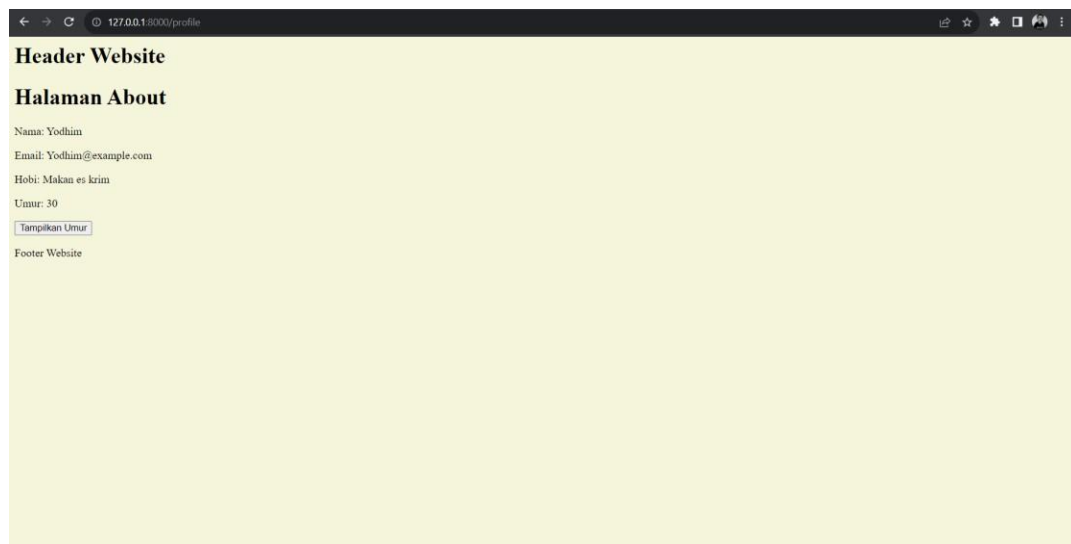
```

resources > views > profile.blade.php
1  <link rel="stylesheet" href="/css/style.css">
2
3  @extends('layout')
4
5  @section('title', 'Halaman About')
6
7  @section('content')
8      <h1>Halaman About</h1>
9      <p>Nama: {{ $nama }}</p>
10     <p>Email: {{ $email }}</p>
11     <p>Hobi: {{ $hobi }}</p>
12     <p>Umur: {{ $umur }}</p>
13
14     <button onclick="tampilkanUmur()">Tampilkan Umur</button>
15
16     <script>
17         function tampilkanUmur() {
18             const umur = {{ $umur }};
19             alert(`Umur: ${umur} tahun`);
20         }
21     </script>
22 @endsection
23

```

*Gambar 17 Tampilan profile.blade.php*

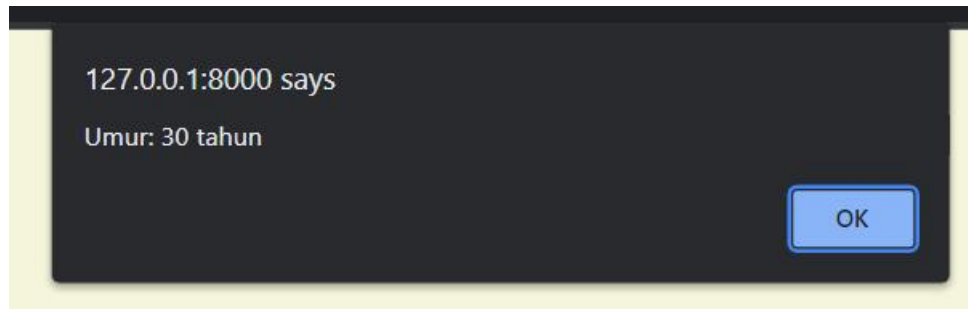
Pada halaman "Profile" di atas, kita menggunakan `@extends('layout')` untuk menghubungkan halaman ini dengan layout Blade yang telah dibuat. `@section('title')` dan `@section('content')` digunakan untuk mengisi judul halaman dan kontennya. Berikut ini merupakan tampilan web yang telah kita jalankan.



*Gambar 18 Tampilan web dari file profile*

Pada kode diatas juga terdapat kode javascript, kode tersebut merupakan bagian dari halaman web yang menampilkan tombol "Tampilkan Umur". Ketika tombol tersebut diklik, sebuah pesan peringatan (alert) akan muncul yang berisi umur. Kode JavaScript digunakan untuk mengeksekusi tindakan ini, dengan mengambil nilai umur dari data yang telah disiapkan di halaman tersebut. Berikut ini tampilan jika kita mengklik tombol tampilkan umur.





*Gambar 19 Tampilan tombol tampilkan umur*

Dengan menggunakan Blade, kita dapat membuat tata letak yang lebih terstruktur dan mudah diubah, dan menghindari duplikasi kode dalam proyek Laravel Anda.

## **BAB IV**

### **KESIMPULAN**

#### **4.1. Kesimpulan**

Pemrograman web modern memperkenalkan konsep-konsep penting seperti penanganan rute (routing), tampilan (view), dan penerapan template (Blade) sebagai pondasi inti dalam mengembangkan aplikasi web. Laporan praktikum ini dimulai dengan pemahaman tentang kendala yang kerap dihadapi oleh mahasiswa dalam menguasai konsep-konsep tersebut. Tantangan ini mencakup kesulitan dalam mengembangkan rute yang efisien, mengaitkan data dengan tampilan, dan memisahkan logika dari tampilan.

Dalam konteks ini, Laravel, yang merupakan framework PHP terkemuka, telah terbukti sebagai solusi yang sangat efisien. Dengan sistem routing yang mudah dikonfigurasi, Laravel memungkinkan para pengembang untuk dengan cepat menentukan bagaimana permintaan HTTP harus ditangani. Keunggulan Blade dalam memisahkan tampilan dari logika memberikan struktur yang kuat dalam pengembangan web dan membantu menjaga kode tetap teratur.

Praktikum pemrograman web ini bertujuan untuk membekali mahasiswa dengan pemahaman tentang konsep routing, view, dan Blade di dalam Laravel. Pemahaman ini menjadi landasan penting dalam mengembangkan aplikasi web modern. Mahasiswa akan memiliki keterampilan untuk memahami cara kerja sebuah aplikasi web, bagaimana data disajikan kepada pengguna, dan bagaimana mengelola logika aplikasi secara efisien. Praktikum ini membantu mahasiswa mengatasi tantangan pengembangan web yang semakin kompleks. Dengan penerapan yang tepat, Laravel membantu mengatasi kendala-kendala tersebut dengan efisiensi. Selain itu, praktikum ini mendorong mahasiswa untuk terus meningkatkan keterampilan dalam pengembangan web, menghadapi masalah, dan menciptakan pengalaman pengguna yang lebih baik. Dengan pemahaman yang kuat tentang routing, view, dan Blade di dalam Laravel, mahasiswa akan siap menghadapi tantangan dalam dunia pengembangan web yang selalu berubah dan kompetitif. Praktikum ini memberikan dasar yang kokoh bagi pengembangan kemampuan mereka dan membekali mereka sebagai calon pengembang web yang handal di masa depan.

## DAFTAR PUSTAKA

<https://github.com/wildanzake/ppwpertemuan2/>

<https://informatika.uc.ac.id/2019/09/laravel-view-blade/>

