

Laravel Blade Templating Engine Implementation (Case Study: Muhammadiyah Lamongan Hospital Job Management System)

Firmansyah^{1*}, Danur Wijayanto¹

¹Information Technology, Faculty of Science and Technology, Universitas 'Aisyiyah Yogyakarta, Indonesia

*Email korespondensi : danurwijayanto@gmail.com

Received Feb 03, 2022; Accepted Feb 20, 2022; Published Mar 08, 2022

Abstract. Laravel is one of several frameworks owned by the Hypertext Preprocessor (PHP) programming language. Laravel itself was created to facilitate programmers in creating a website with a framework that laravel is expressive and has syntax that is easy to understand. Laravel can also automate common things that are commonly used when building an application such as: Database library, Templating Engine, Session Management, etc. So much used to create a web application. Hypertext Markup Language (HTML), Cascading Style Sheets (CSS) and PHP programming languages: Hypertext Preprocessor is the most widely used programming language for creating a web sistem, using the Blade Templating Engine feature in the Laravel Framework will make it easier for programmers to create dynamic interfaces where programmers do not need to repeat the same code on each page, like; Navbar, Sidebar, footer and so on. By using directive yield, section and extends to call the template that has been created.

Keywords: *Laravel, Software Engineer, Waterfall, PHP*

1. Pendahuluan

Pada umumnya pembuatan sebuah sistem informasi baik berbasis aplikasi ataupun web menggunakan bahasa pemrograman native atau asli tanpa melibatkan pihak ketiga ataupun yang biasa dikenal dengan *framework*, *framework* sendiri merupakan koleksi atau kumpulan potongan-potongan program yang disusun atau diorganisasikan sedemikian rupa, sehingga dapat digunakan untuk membantu membuat aplikasi utuh tanpa harus membuat semua kodenya dari awal [1]. Setiap bahasa pemrograman memiliki frameworknya sendiri, seperti bahasa pemrograman PHP (*Hypertext Preprocessor*) dapat menggunakan *framework laravel* [1][2] dengan menggunakan framework seorang programmer bisa membuat sebuah aplikasi yang sempurna tanpa harus membuat kodenya dari awal karena beberapa hal yang umum dalam membuat sebuah aplikasi sudah tersedia dalam framework.

Framework laravel [3] merupakan kerangka kerja [1] web yang menyediakan struktur pengembangan sebuah web aplikasi [4], laravel juga menyediakan fitur-fitur yang memungkinkan programmer untuk dapat dengan mudah membuat serta mengembangkan aplikasi. Injeksi ketergantungan menyeluruh, lapisan abstraksi basisdata ekspresif, antrian dan pekerjaan terjadwal, pengujian unit dan integrasi [5] merupakan beberapa fitur yang dapat digunakan untuk pengembangan aplikasi yang ada pada framework laravel selain untuk *Backend developer*, laravel juga menyediakan fitur untuk mengelola tampilan [6] agar kode yang dibuat tidak terlalu banyak sehingga akan terlihat lebih profesional. Pada *framework laravel* sendiri

menggunakan fitur *Blade Templating Engine* [1] mengelola tampilan, adapun studi kasus yang digunakan adalah Sistem Manajemen Pekerjaan RS. Muhammadiyah Lamongan.

Sistem Manajemen Pekerjaan adalah cara untuk membuka, menempatkan, dan memantau semua sumber daya yang terlibat dalam suatu proyek untuk mencapai tujuan yang telah ditentukan. Proyek yang berhasil didefinisikan sebagai proyek yang memenuhi tanggal target, memenuhi biaya target, memenuhi tingkat kinerja yang diinginkan, menggunakan sumber daya secara efektif dan efisien [7] di unit SIRS (Sistem Informasi Rumah Sakit) yang ada pada RS. Muhammadiyah Lamongan. Proses bisnis di unit ini awalnya masih mengharuskan membuat suatu pertemuan untuk menentukan ketua tim beserta anggotanya, koordinasi tim dengan direksi dan melakukan monitoring evaluasi. Pada awalnya pembuatan sistem ini hanya menggunakan bahasanya pemrograman PHP tanpa [8] menggunakan *framework laravel* - sehingga programmer kesulitan terutama saat membuat tampilan harus ada pada setiap halamannya, dimana programmer perlu membuat kode yang sama agar tampilan layout tetap ada di masing-masing halaman. Hal ini tentunya akan mempengaruhi kinerja dan waktu yang dibutuhkan untuk menyelesaikan sistem tersebut. *Blade Templating Engine* [5] pada *Laravel* juga akan berguna ketika programmer menginputkan sebuah variabel yang berulang-ulang, fitur ini dapat memudahkannya tanpa harus mengganti pada setiap halaman yang ada.

Berdasarkan uraian dari latar belakang masalah tersebut penulis mengangkat sebuah penelitian dengan judul “Implementasi *Blade Templating Engine* Pada *Laravel* (Studi Kasus Sistem Manajemen Pekerjaan RS. Muhammadiyah Lamongan).

2. Metode Penelitian

Metode penelitian merupakan cara ilmiah untuk mendapatkan data dengan tujuan dapat dideskripsikan, dibuktikan, dikembangkan dan ditemukan pengetahuan, teori, untuk memahami, memecahkan, dan mengantisipasi masalah dalam kehidupan manusia. Pada bab ini menjelaskan mengenai metode yang digunakan dalam pengembangan aplikasi dan juga implementasi fitur *blade templating engine* yang ada pada *laravel*.

2.1. Metode Pengembangan aplikasi

Pada pengembangan sistem ini menggunakan model *waterfall* [5]. Model ini sudah banyak digunakan dalam pengembangan perangkat lunak, dinamakan *waterfall* karena tahapan-tahapan yang dilalui akan berurutan dimana jika tahap pertama belum selesai maka belum bisa untuk melanjutkan ke tahap berikutnya. Model ini juga disebut sebagai model air terjun karena memiliki alur hidup perangkat lunak secara skematis [9] atau terurut, mulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (Support). Tahapan-tahapan model *waterfall* ditunjukkan pada Gambar 1.



Gambar 1. Tahap-tahap Model Waterfall

2.1.1. Analisis Kebutuhan

Proses akuisisi data terpusat sehingga pengguna dapat menentukan kebutuhan perangkat lunak yang digunakan, seperti *Laravel*, *PHP*, *Bootstrap*, *XAMPP*, dan *Composer*.

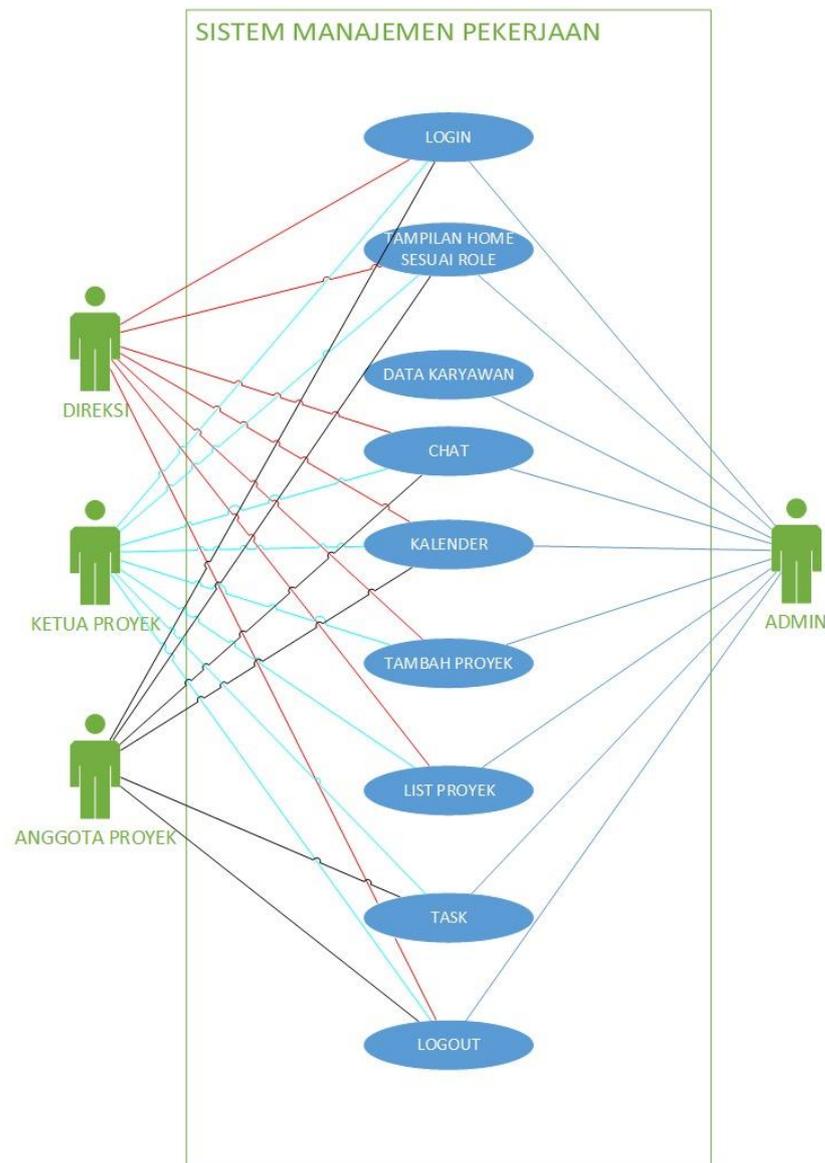
Use Case Diagram

Use Case Diagram merupakan gambaran fungsionalitas [10] dari aktor yang telah dibuat terhadap view apa saja yang bisa mereka akses, pada penelitian ini terdapat 4 aktor, antara lain:

a. Admin

Admin memiliki kendali penuh atas sebuah sistem, menampilkan data karyawan, tambah data, edit data, dan semua halaman yang ada pada sistem

- b. Direksi
Direksi memiliki kendali yang hampir sama dengan admin perbedaannya ialah direksi tidak bisa untuk menampilkan data karyawan.
 - c. Ketua tim
Ketua tim memiliki kendali atas sebuah proyek yang akan dibuat, termasuk mengedit dan menambahkan anggota baru dalam sebuah proyek.
 - d. Anggota tim
Anggota tim hanya memiliki kendali atas proyek apa yang sedang dikerjakan
- Gambar 2 menunjukkan gambaran diagram *use case*.



Gambar 2. Use Case Diagram

Analisis Views

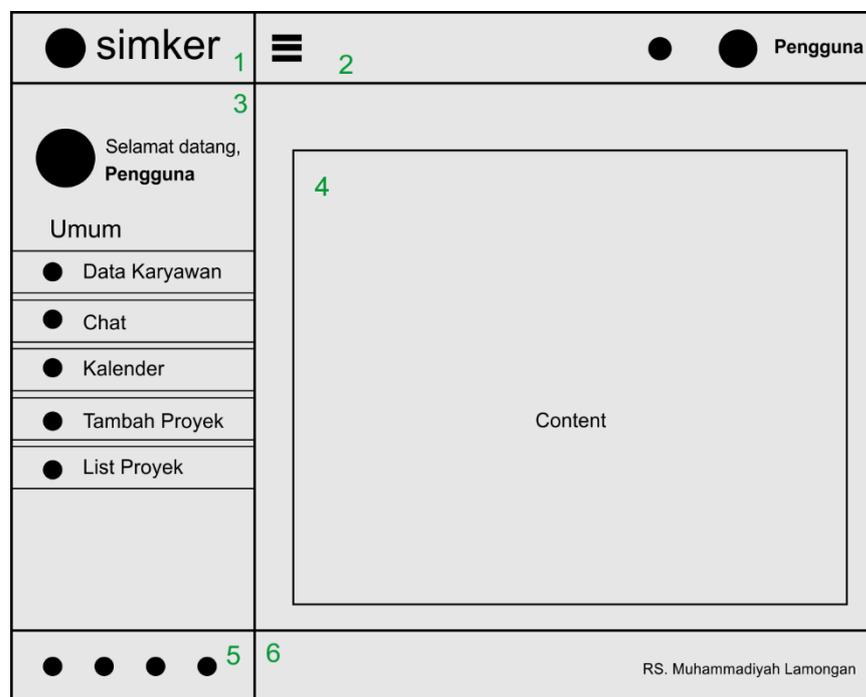
Pada tahap ini didapatkan *view* apa aja yang dihasilkan dengan merujuk pada *use case* pada Gambar 2 sehingga aplikasi dapat digunakan dengan baik menggunakan kelas-kelas tersebut. Hasil analisis kelas dapat dilihat pada Tabel 1.

Tabel 1. Data view

No	Nama View	Fungsi
1	home	Menampilkan halaman utama
2	karyawan	Menampilkan semua data karyawan
3	form	Menbuat proyek baru
4	projects	Menampilkan proyek-proyek yang telah dibuat
5	projects_detail	Menampilkan detaile dari proyek yang suda dibuat

2.1.2. Desain

Desain perangkat lunak. Fokusnya adalah pada pembuatan program perangkat lunak seperti struktur data, arsitektur perangkat lunak, representasi antarmuka, dan metode pengkodean. Pada fase ini, kebutuhan perangkat lunak dari fase analisis kebutuhan diubah menjadi representasi desain untuk implementasi selanjutnya dalam program. Gambar 3 merupakan desain antarmuka yang digunakan.



Gambar 3. Mock up desain antarmuka

Keterangan:

1. Bagian 1 merupakan bagian dari komponen sidebar yang digunakan untuk menempatkan logo dan nama sistem yang dibuat.
2. Bagian 2 merupakan komponen navbar dimana komponen ini menampilkan notifikasi dan gambar serta nama dari pengguna.
3. Bagian 3 merupakan komponen sidebar dimana komponen ini menampilkan gambar dan ucapan selamat datang bagi pengguna yang login, selain itu terdapat beberapa menu untuk ditampilkan, diantaranya:
 - a. Menu Data karyawan yang bertujuan untuk menampilkan karyawan-karyawan yang sudah diinputkan kedalam sistem.
 - b. Menu chat digunakan untuk berkomunikasi antar anggota dalam mengerjakan sebuah proyek.
 - c. Menu kalender digunakan untuk menampilkan tanggal serta melihat deadline dari proyek yang sedang di kerjakan.

- d. Menu tambah proyek digunakan untuk menambahkan proyek baru sekaligus menentukan ketua dan anggotanya
 - e. Dan menu list proyek digunakan untuk menampilkan proyek yang sudah dibuat dan melihat presentasi progress dari masing-masing proyek.
4. Bagian no 4 merupakan komponen main content dimana digunakan untuk menampilkan tabel, gambar, dan atau yang lainnya yang merupakan konten dari sistem tersebut.
 5. Bagian no 5 merupakan bagian dari komponen sidebar dimana terdapat beberapa menu yang bisa ditambahkan, seperti logout dan pengaturan.
 6. Bagian no 6 merupakan komponen footer dimana digunakan untuk menampilkan nama sistem, perusahaan dan atau versi dari sistem yang dibuat,

2.1.3. Pengkodean

Desain harus dituangkan kedalam sebuah program. Hasil dari tahap ini adalah program computer yang sesuai dengan desain yang sudah dibuat sebelumnya. Untuk melakukan implementasi dari desain yang sudah dibuat diatas penulis melakukan beberapa tahapan antara lain:

1. Menghubungkan Laravel dengan Template Admin

Pada tahap ini akan menghubungkan aplikasi laravel yang sudah dibuat dengan template admin yang sudah ada, template yang digunakan adalah gantella master yang sudah disesuaikan dengan studi kasus yang telah dibuat. Tahap pertama adalah memindahkan semua file asset ke dalam folder public dan juga file index kedalam folder views di aplikasi yang sudah dibuat sebelumnya. Melakukan sebuah penyesuaian terhadap file-file yang sudah dipindahkan sebelumnya agar dapat diakses oleh Laravel adalah dengan menggunakan fitur `{{ asset('folder') }}` yang ditunjukkan pada Gambar 4.

```
<!-- Bootstrap -->
<link href="{{ asset('/vendors') }}/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
<!-- Font Awesome -->
<link href="../../../vendors/font-awesome/css/font-awesome.min.css" rel="stylesheet">
<!-- NProgress -->
<link href="../../../vendors/nprogress/nprogress.css" rel="stylesheet">
<!-- iCheck -->
<link href="../../../vendors/iCheck/skins/flat/green.css" rel="stylesheet">
```

Gambar 4. Implementasi fitur asset

2. Tahap selanjutnya adalah membuat sebuah routes dengan method GET yang terletak di folder routes file web.php. Fungsi dari method GET adalah untuk mengembalikan file html ke web browser yang ditunjukkan pada Gambar 5.

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

Gambar 5. Route index

Keterangan:

1. `get` merupakan method yang diizinkan untuk menjalankan fungsi pada route
2. `/index` merupakan alamat URI yang ingin diakses untuk menjalankan sebuah fungsi pada *route*.
3. `return view ('index');` merupakan *callback function* yang akan dijalankan ketika suatu URI diakses dengan *method* yang sesuai [11].

2.1.4. Implementasi Fitur Blade Templating Engine

Pada subbab ini akan menjelaskan mengenai implementasi yang akan dilakukan. Ini tentunya tidak akan optimal jika suatu website memiliki banyak halaman sehingga rawan terjadinya inkonsistensi [12]. Untuk meminimalisir itu semua Laravel sudah menyediakan fitur layouting [8] yang di namakan *Blade Templating Engine* yang berfungsi agar tampilan yang dipanggil berulang-ulang seperti, navbar, sidebar

dan footer tidak perlu dibuat berkali-kali. Pada umumnya layout akan di tempatkan di folder yang bernama layout, tentunya tidak ada penamaan khusus untuk ini, programmer bisa menggunakan nama lain. Akan tetapi untuk mempermudah pengerjaan biasanya view konten akan ditempatkan di folder yang berbeda dengan template.

3. Hasil dan Pembahasan

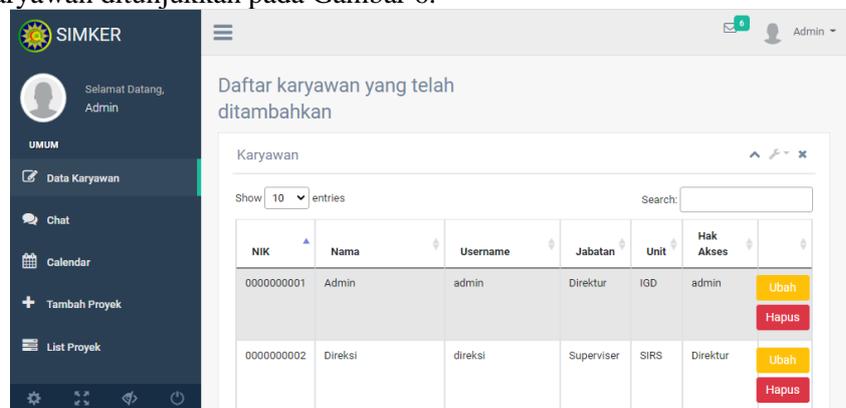
Pada bab ini akan menjelaskan uraian hasil dan pembahasan dari implementasi dan data analisis *view* pada Tabel 1.

3.1. Hasil

Pada umumnya *blade templating* hanya berfungsi untuk komponen yang dibuat secara berulang-ulang dan bisa di panggil kapan saja. Hal yang membedakan dari tampilan yang sudah menerapkan *blade templating* adalah di bagian konten/isi atau bagian tengah dalam sebuah layout. Untuk mempermudah mendefinisikan fila layout, untuk penamaannya diganti menjadi `main.blade.php`

a. Data Karyawan

Tampilan ini berfungsi untuk menampilkan seluruh data karyawan yang ada di Rumah Sakit Muhammadiyah Lamongan yang nantinya akan tergabung disuatu proyek tertentu. Tampilan data karyawan ditunjukkan pada Gambar 6.

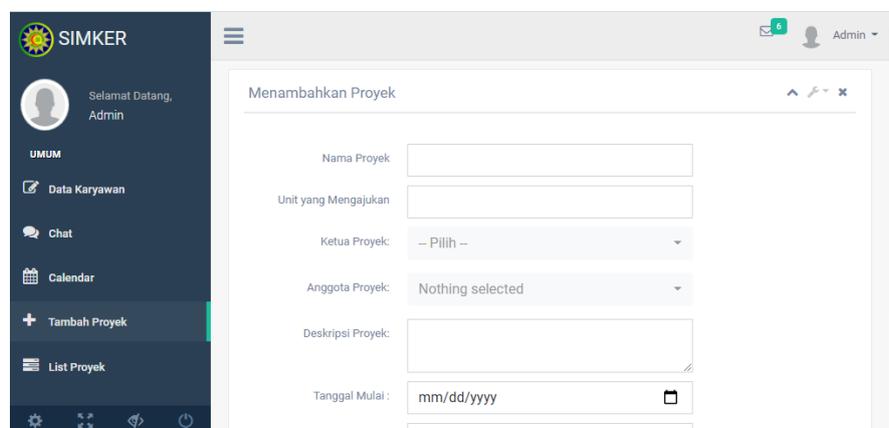


NIK	Nama	Username	Jabatan	Unit	Hak Akses	
000000001	Admin	admin	Direktur	IGD	admin	Ubah Hapus
000000002	Direksi	direksi	Supervisor	SIRS	Direktur	Ubah Hapus

Gambar 6. View Karyawan

b. Tambah Proyek

Tampilan ini berfungsi untuk membuat proyek baru, menentukan ketua dan anggota tim serta menentukan tanggal deadline suatu proyek. Tampilan Tambah Proyek ditunjukkan pada Gambar 7.

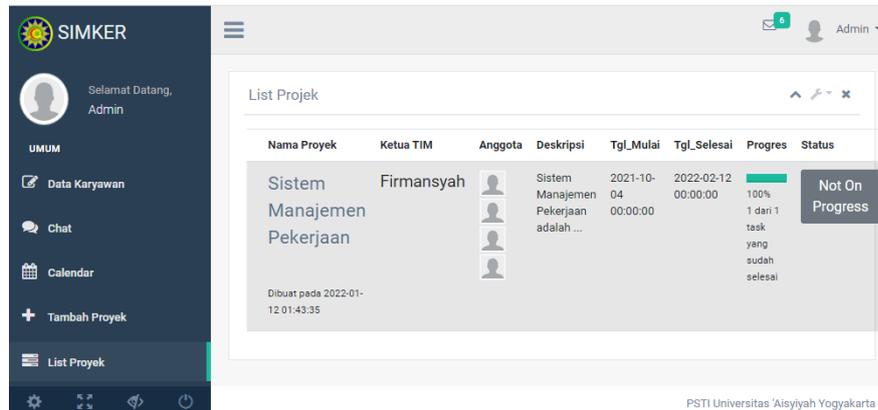


Gambar 7. View form tambah proyek

c. List Proyek

Tampilan ini berfungsi untuk menampilkan proyek apa saja yang sudah dibuat, sekaligus

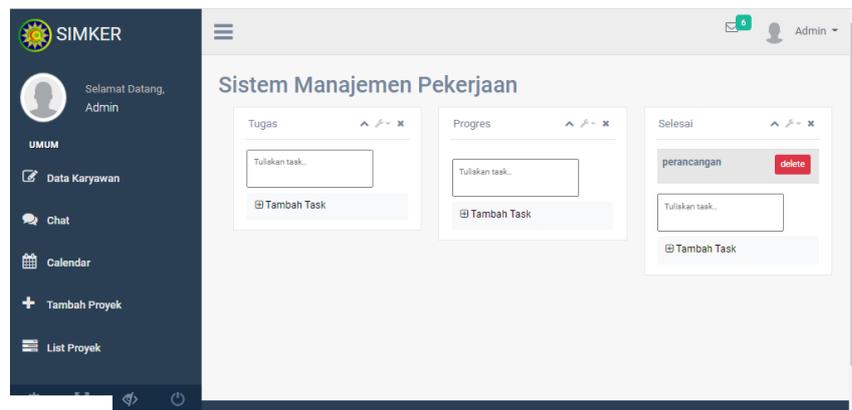
progress penyelesaian dari masing-masing proyek yang sudah dibuat yang ditunjukkan pada Gambar 8.



Gambar 8. View list proyek

d. Proyek Detail *Task*

Pada tampilan ini berisi detail dari proyek yang sudah dibuat, terdapat tiga tahapan yaitu, tugas, progress dan selesai dimana masing-masing tahapan bisa di tambahkan tugas-tugas yang ada dalam proyek tersebut dan memindahkannya ke tahapan yang diinginkan. Ketika tugas di klik akan menampilkan modal dimana terdapat fitur untuk memberikan feedback terhadap tugas yang dikerjakan. Tampilan detail *task* dapat dilihat pada Gambar 9.



Gambar 9. View detail proyek

3.2. Pembahasan

Pada bab ini akan menjelaskan mengenai apa saja tag dan directive yang digunakan untuk dapat mengimplementasikan fitur blade templating dengan baik dan benar pada view-view yang telah dibuat. Adapun struktur file view bisa dilihat pada Gambar 10.

```

layout
    main.blade.php

partial
    footer.blade.php
    menu_footer.blade.php
    nav_bar.blade.php
    top_nav.blade.php

karyawan
    data_karyawan.blade.php

proyek
    form.blade.php
    projects.blade.php
    projects_detail.blade.php
  
```

Gambar 10. Struktur view

```

49 </head>
50
51 <body class="nav-md">
52 <div class="container body">
53 <div class="main_container">
54
55 @include('partial.nav_bar')
56
57 <!-- top navigation -->
58 @include('partial.top_nav')
59 <!-- /top navigation -->
60
61 <!-- page content -->
62 <div class="container">
63 @yield('container')
64 </div>
65 <!-- /page content -->
66
67 <!-- footer content -->
68 @include('partial.footer')
69 <!-- /footer content -->
70
71 <!-- script -->
72
73 <!-- /script -->
74 </div>
75 </div>
76

```

Gambar 11. Source code main.blade.php

Pada file main.blade.php menggunakan beberapa tag/directive seperti @include dan @yield. Pada Gambar 11 di baris 55, 58, dan baris 68 terdapat tag @include('nama_folder.nama_file) tag tersebut berfungsi untuk memanggil sekaligus menempatkan bagian-bagian yang ada di folder partial ke file main dan di line 63 terdapat directive @yield('container') yang berfungsi sebagai pengganti konten dari masing-masing halaman directive @yield akan berhubungan dengan directive @section('container') yang nantinya source code konten di setiap halaman diletakkan di dalam @section('container') dan ditutup dengan @endsection [13] yang ditunjukkan pada Gambar 12.

```

1 @extends('layout.main')
2
3 @section('container')
4 <!-- page content -->
5 <div class="right_col" role="main">
6 <div class="">
7 <div class="page-title">
8 </div>
9 </div>
10 <div class="clearfix"></div>
11
12 <div class="row">
13 <div class="col-md-12">
14 <div class="x_panel">
15 @if (Session::get('update'))
16 <div class="alert alert-success">
17 {{ Session::get('update') }}
18 </div>
19 @endif
20 @if (Session::get('gagal_update'))
21 <div class="alert alert-danger">
22 {{ Session::get('gagal_update') }}
23 </div>
24 @endif
25 @yield('container')
26 <h2>List Projek</h2>
27 <ul class="nav navbar-right panel_toolbox">
28 <li><a class="collapse-link"><i class="fa fa-chevron-up"></i></a>
29 </li>
30 <li class="dropdown">

```

Gambar 12. Source code view projects

Pada line 3 terdapat directive @section('container') dan di bawah nya merupakan source code konten halaman projects dan ditutup dengan @endsection, source code konten ini akan di panggil di file main pada directive @yield('container') hal ini berlaku untuk semua konten di setiap halaman. Adapun directive @extends('layout.main') berfungsi untuk memanggil template yang ada di file main.blade.php di folder layout.

4. Kesimpulan

Fitur *Blade Templating Engine* akan sangat membantu programmer dalam mengembangkan sebuah aplikasi khususnya berbasis website, dengan fitur ini programmer tidak perlu menuliskan kode pada bagian-bagian tertentu secara berulang-ulang cukup dengan memanggil bagian-bagian tersebut menggunakan directive `extends`, `yield`, dan `section`. Blade juga tidak membatasi pemrogram untuk menggunakan kode PHP biasa dalam membuat kode untuk tampilan. Semua tampilan Blade dikompilasi ke dalam kode PHP biasa dan kemudian disimpan dalam cache hingga diubah, yang berarti Blade pada dasarnya tidak menambahkan beban atau overhead pada saat aplikasi dijalankan.

5. Ucapan Terima kasih

Terima kasih kepada Allah Subhanahu Wa Ta'ala yang telah memberikan peneliti kesanggupan dalam membuat paper ini. Tak tertinggal pula dosen serta pembimbing Merdeka Belajar Kampus Merdeka (MBKM) magang industri baik dari Program Studi maupun pembimbing lapangan, terima kasih juga peneliti ucapkan kepada unit SIRS (Sistem Informasi Rumah Sakit) Muhammadiyah Lamongan yang sudah membantu peneliti dalam berupaya untuk menyelesaikan proyek Sistem Manajemen Pekerjaan (SIMKER) dan teman-teman satu tim yang peneliti banggakan.

6. Daftar Pustaka

- [1] D. Mediana and A. I. Nurhidayat, "RANCANG BANGUN APLIKASI HELPDESK (A-DESK) BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL (STUDI KASUS DI PDAM SURYA SEMBADA KOTA SURABAYA)."
- [2] A. Maolana and C. K. Sastradipraja, "Strategi Pengembangan Sistem Informasi Pelaporan Kemajuan Kerja Di Masa Pandemi Covid 19," *J. Sist. Inf. dan Teknol. Informasi*, vol. 3, no. 3, pp. 9–18, 2021.
- [3] A. Nugrahaning Widhi, E. Sutanta, and E. Kumalasari Nurnawati, "PEMANFAATAN FRAMEWORKLARAVEL UNTUK PENGEMBANGAN SISTEM INFORMASI TOKO ONLINE DI TOKO NEW TREND BATURETNO," vol. 7, no. 2, 2019.
- [4] I. N. Hapsari, "PENYULUHAN MENGENAI PENGATURAN LARAVEL PADA WINDOWS BAGI SISWA SMK YAYASAN CINTA KASIH TZU CHI," 2020.
- [5] D. Purnama Sari, R. Wijanarko, and J. X. Menoreh Tengah, "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang)," vol. 2, no. 1, pp. 32–36, 2019.
- [6] A. Caesar, M. Hanif, and M. A. I. Pakereng, "Pengembangan Aplikasi Sistem Informasi Operasional Bus Berbasis Web Menggunakan Framework Laravel," *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 3, 2021, [Online]. Available: <http://jurnal.mdp.ac.id>.
- [7] I. Ketut Aditya Herdinata Putra, D. Pramana, N. Luh Putri Srinadi, and S. STIKOM Bali Jl Raya Puputan, "Sistem Manajemen Arsip Menggunakan Framework Laravel dan Vue.js (Studi Kasus: BPKAD Provinsi Bali)."
- [8] A. Ilmiah Peneliti, B. Bond Banjarnahor, K. Dwi Hartomo, and U. Kristen Satya Wacana Salatiga, "Penerapan Laravel Framework Dalam Perancangan Sistem Informasi Promosi Produk Unggulan UKM Berbasis Web (Studi Kasus Dinas Perindustrian Perdagangan dan UMKM Kota Salatiga)," 2016.
- [9] N. Hidayati, "Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan," 2019.
- [10] G. Wiro Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *J. Inform. Pengemb. IT*, vol. 2, no. 1, pp. 6–12, 2017.
- [11] S. E. Wahyudi, "Laravel View dan Blade (Belajar Laravel #5)," *Informatika.uc.ac.id*, 2019.
- [12] F. G. Robert Khoiril Husada, "IMPLEMENTATION OF LARAVEL FRAMEWORK TO BUILD WATER PAYMENT ADMINISTRATION SYSTEM IN PPAB PERUM DOLOG BY WATERFALL METHOD THESIS MADE BY," 2019.
- [13] Laravel, "Blade Templates," *laravel-guide.readthedocs.io*. <https://laravel-guide.readthedocs.io/en/latest/blade/#blade-templates>.

