

Laporan Modul 4: Laravel Blade Template Engine

Mata Kuliah: Workshop Web Lanjut

Nama: Fathan Mubina

NIM: 2024573010058

Kelas: TI-2C

Abstrak

Praktikum ini membahas penggunaan Blade Template Engine pada framework Laravel sebagai alat untuk mengelola tampilan (view) secara efisien, modular, dan dinamis. Tujuan dari kegiatan ini adalah agar mahasiswa memahami cara meneruskan data dari controller ke view, menggunakan struktur kontrol pada Blade, serta membangun layout dengan konsep templating, partial view, dan komponen. Dalam praktikum dilakukan empat percobaan, yaitu: meneruskan data dari controller ke Blade, penggunaan struktur kontrol, pembuatan layout dengan Bootstrap, serta implementasi partial views dan Blade components. Hasil praktikum menunjukkan bahwa Blade Template Engine mempermudah pengembangan antarmuka pengguna dengan kode yang lebih terorganisir, dapat digunakan kembali (reusable), dan mudah dipelihara. Dengan demikian, penggunaan Blade dapat meningkatkan produktivitas serta konsistensi tampilan dalam pengembangan aplikasi Laravel.

1. Dasar Teori

Laravel adalah salah satu framework PHP modern yang menerapkan pola arsitektur Model-View-Controller (MVC). Dalam arsitektur ini, bagian View bertanggung jawab menampilkan data kepada pengguna. Untuk mengelola tampilan dengan lebih efisien, Laravel menyediakan sistem template bawaan bernama Blade Template Engine.

Blade Template Engine merupakan mesin templating yang ringan, cepat, dan mudah digunakan untuk memisahkan logika pemrograman dari tampilan antarmuka. Blade memungkinkan penggunaan sintaks yang lebih sederhana dibandingkan PHP murni, seperti penggunaan @if, @foreach, @extends, dan @include. Dengan Blade, pengembang dapat membuat tampilan yang lebih terstruktur, mudah dipelihara, serta mendukung konsep template inheritance dan component-based design.

Beberapa fitur utama dari Blade Template Engine antara lain:

- **Template Inheritance** — memungkinkan satu layout dasar digunakan oleh beberapa halaman dengan menggunakan direktif @extends dan @section.
- **Component dan Partial Views** — membantu memecah tampilan besar menjadi bagian kecil yang dapat digunakan kembali, seperti header, footer, dan sidebar.
- **Control Structures** — mendukung logika seperti if, foreach, dan for langsung di dalam file Blade dengan sintaks yang sederhana.
- **Data Binding** — memungkinkan data dari controller dikirim ke view dan ditampilkan menggunakan tanda kurung kurawal ganda {{ }} secara aman.
- **Integrasi Bootstrap dan UI** — Blade dapat dipadukan dengan framework CSS seperti Bootstrap untuk mempercepat pembuatan tampilan yang responsif dan menarik.

Dalam konteks pengembangan aplikasi web, Blade memegang peranan penting dalam meningkatkan modularitas, efisiensi, dan konsistensi tampilan antar halaman. Dengan memanfaatkan fitur seperti komponen, layout, dan partial view, proses pengembangan front-end menjadi lebih terstruktur dan produktif.

2. Langkah-Langkah Praktikum

Praktikum 1: Meneruskan Data dari Controller ke Blade View

1. Langkah 1: Buat dan Buka Proyek laravel

- laravel new modul-4-blade-view
- cd modul-4-blade-view
- code .

2. Langkah 2: Buat Sebuah Controller php artisan make:controller DasarBladeController, ini akan membuat file di app/Http/Controllers DasarBladeController.php.

3. Langkah 3: Definisikan Rute Buka file Routes/Web.php Dan isi dengan code berikut:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\DasarBladeController;
5 use App\Http\Controllers\LogicController;
6 use App\Http\Controllers\PageController;
7
8 Route::get('/admin', [PageController::class, 'admin']);
9 Route::get('/user', [PageController::class, 'user']);
10 Route::get('/logic', [LogicController::class, 'show']);
11 Route::get('/dasar', [DasarBladeController::class, 'showData']);
12 Route::get('/', function () {
13     return view('welcome');
14 });
15
```

4. Langkah 4: Buat Metode Untuk menghandle Data pada Controller Buka app/Http/controllers/dasarbladecontroller.php Isi Dengan Code Berikut:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class DasarBladeController extends Controller
8 {
9     public function showData()
10    {
11        $name = 'Devi';
12        $fruits = ['Apple', 'Banana', 'Cherry'];
13        $user = [
14            'name' => 'Eva',
15            'email' => 'eva@ilmudata.id',
16            'is_active' => true,
17        ];
18        $product = (object) [
19            'id' => 1,
20            'name' => 'Laptop',
21            'price' => 12000000
22        ];
23
24        return view('dasar', compact('name', 'fruits', 'user', 'product'));
25    }
26 }
27
```

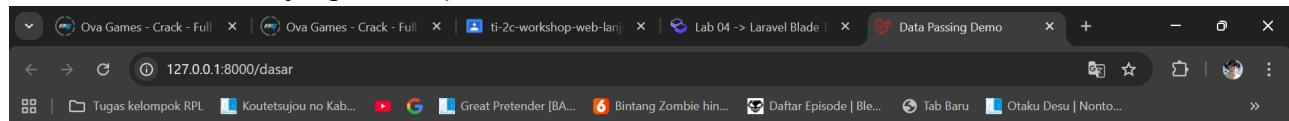
5. Langkah 5: Buat Blade View Buat File Baru di Direktori resources/views/dasar.blade.php.
Isi Dengan Code Berikut

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Data Passing Demo</title>
5  </head>
6  <body>
7      <h1>Passing Data to Blade View</h1>
8
9      <h2>String</h2>
10     <p>Name: {{ $name }}</p>
11
12     <h2>Array</h2>
13     <ul>
14         @foreach ($fruits as $fruit)
15             <li>{{ $fruit }}</li>
16         @endforeach
17     </ul>
18
19     <h2>Associative Array</h2>
20     <p>Name: {{ $user['name'] }}</p>
21     <p>Email: {{ $user['email'] }}</p>
22     <p>Status: {{ $user['is_active'] ? 'Active' : 'Inactive' }}</p>
23
24     <h2>Object</h2>
25     <p>ID: {{ $product->id }}</p>
26     <p>Product: {{ $product->name }}</p>
27     <p>Price: Rp{{ number_format($product->price, 0, ',', '.') }}</p>
28 </body>
29 </html>
30
```

6. Langkah 6: Uji Rute Hidupkan server dengan ketik:

Php Artisan serve

Buka Browser Dan Kunjungi link <http://127.0.0.1:8000/dasar>



Passing Data to Blade View

String

Name: Devi

Array

- Apple
- Banana
- Cherry

Associative Array

Name: Eva

Email: eva@ilmudata.id

Status: Active

Object

ID: 1

Product: Laptop

Price: Rp12.000.000

Praktikum 2: Menggunakan Struktur Kontrol Blade

1. Langkah 1: Buat Controller Baru

Di Dalam Project Modul-4-blade-view Bikin controller

php artisan make:controller LogicController

Ini akan membuat file app/Http/Controllers/LogicController.php.

2. Langkah 2: Tambahkan route baru

Buka routes/web.php dan tambahkan:

```

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\DasarBladeController;
5 use App\Http\Controllers\LogicController;
6 use App\Http\Controllers\PageController;
7
8 Route::get('/admin', [PageController::class, 'admin']);
9 Route::get('/user', [PageController::class, 'user']);
10 Route::get('/logic', [LogicController::class, 'show']);
11 Route::get('/dasar', [DasarBladeController::class, 'showData']);
12 Route::get('/', function () {
13     return view('welcome');
14 });
15

```

3. Langkah 3: Tambahkan Logika Di Controller

isi dengan code berikut:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class LogicController extends Controller
8 {
9     public function show()
10    {
11        $isLoggedIn = true;
12        $users = [
13            ['name' => 'Marcel', 'role' => 'admin'],
14            ['name' => 'Thariq', 'role' => 'editor'],
15            ['name' => 'Ellian', 'role' => 'subscriber'],
16        ];
17        $products = []; // Simulasi array kosong untuk @forelse
18        $profile = [
19            'name' => 'Thariq',
20            'email' => 'thariq@ilmudata.id'
21        ];
22        $status = 'active';
23
24        return view('logic', compact('isLoggedIn', 'users', 'products', 'profile', 'status'));
25    }
26 }
27
```

4. Langkah 4: Buat Blade View

Buat file view Di direktori resources/views/logic.blade.php:

Kemudian Isi Dengan Code Berikut:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Blade Logic Demo</title>
5 </head>
6 <body>
7     <h1>Blade Control Structures Demo</h1>
8
9     <h2>1. @@if / @@else</h2>
10    @if ($isLoggedIn)
11        <p>Welcome back, user!</p>
12    @else
13        <p>Please log in.</p>
14    @endif
15
16    <h2>2. @@foreach</h2>
17    <ul>
18        @foreach ($users as $user)
```

```
19           <li>{{ $user['name'] }} - Role: {{ $user['role'] }}</li>
20       @endforeach
21   </ul>
22
23   <h2>3. @@forelse</h2>
24   @forelse ($products as $product)
25       <p>{{ $product }}</p>
26   @empty
27       <p>No products found.</p>
28   @endforelse
29
30   <h2>4. @@isset</h2>
31   @isset($profile['email'])
32       <p>User Email: {{ $profile['email'] }}</p>
33   @endisset
34
35   <h2>5. @@empty</h2>
36   @empty($profile['phone'])
37       <p>No phone number available.</p>
38   @endempty
39
40   <h2>6. @@switch</h2>
41   @switch($status)
42       @case('active')
43           <p>Status: Active</p>
44           @break
45       @case('inactive')
46           <p>Status: Inactive</p>
47           @break
48       @default
49           <p>Status: Unknown</p>
50   @endswitch
51 </body>
52 </html>
53
```

5. Langkah 5: Jalankan Aplikasi Jalankan Server Pengembangan

php artisan serve

Masuk ke aplikasi pada link berikut: <http://127.0.0.1:8000/logic>

The screenshot shows a browser window with the URL 127.0.0.1:8000/logic. The page content includes:

- Blade Control Structures Demo**
- 1. @if / @else**

Welcome back, user!
- 2. @foreach**
 - Marcel - Role: admin
 - Thariq - Role: editor
 - Ellian - Role: subscriber
- 3. @forelse**

No products found.
- 4. @isset**

User Email: thariq@ilmudata.id
- 5. @empty**

No phone number available.
- 6. @switch**

Status: Active

Praktikum 3: Layout dan Personalisasi di Laravel 12 dengan Bootstrap

1. Langkah 1: Buat Controller baru Masih Sama Di dalam Projek modul-4-view, buatlah controller untuk menangani rute dan logika: php artisan make:controller PageController
2. Langkah 2: Menambahkan Route Di Dalam Routes/web.php kita akan menambahkan rute baru: berikut adalah codenya:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\DasarBladeController;
5 use App\Http\Controllers\LogicController;
6 use App\Http\Controllers\PageController;
7
8 Route::get('/admin', [PageController::class, 'admin']);
9 Route::get('/user', [PageController::class, 'user']);
10 Route::get('/logic', [LogicController::class, 'show']);
11 Route::get('/dasar', [DasarBladeController::class, 'showData']);
12 Route::get('/', function () {
13     return view('welcome');
14 });
15
```

3. Langkah 3: Update Controller Di dalam direktori app/Http/Controllers/PageController.php Isikan Kode Berikut:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class PageController extends Controller
8 {
9     public function admin()
10    {
11        $role = 'admin';
12        $username = 'Yamato Admin';
13        return view('admin.dashboard', compact('role', 'username'));
14    }
15
16    public function user()
17    {
18        $role = 'user';
19        $username = 'Liu User';
20        return view('user.dashboard', compact('role', 'username'));
21    }
22
23 }
24
```

4. Langkah 4: Buat Layout Dasar dengan Bootstrap.

Di dalam direktori resources/view bikin folder layouts kemudian di dalamnya buat file app.blade.php,

isi dengan code berikut:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>@yield('title') | Layout and Personalization</title>
5      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
6  </head>
7  <body>
8      <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
9          <div class="container">
10             <a class="navbar-brand" href="#">Layout and Personalization</a>
11             <div class="collapse navbar-collapse">
12                 <ul class="navbar-nav ms-auto">
13                     <li class="nav-item">
14                         <span class="nav-link active">Welcome, {{ $username }}</span>
15                     </li>
16                 </ul>
17             </div>
18         </div>
19     </nav>
20
21     <div class="container">
22         @if ($role === 'admin')
23             <div class="alert alert-info">Admin Access Granted</div>
24         @elseif ($role === 'user')
25             <div class="alert alert-success">User Area</div>
26         @endif
27
28         @yield('content')
29     </div>
30
31     <footer class="bg-light text-center mt-5 p-3 border-top">
32         <p class="mb-0">© 2025 Layout and Personalization. All rights reserved.</p>
33     </footer>
34
35     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
36 </body>
37 </html>
38
```

5. Langkah 5: Buat view untuk Admin

Buat direktori admin di resources/views. Kemudian buat resources/views/admin/dashboard.blade.php:
Dan masukkan Code Berikut ini:

```
1  @extends('layouts.app')
2
3  @section('title', 'Admin Dashboard')
4
5  @section('content')
6      <h2 class="mb-4">Admin Dashboard</h2>
7      <div class="list-group">
8          <a href="#" class="list-group-item list-group-item-action">Manage Users</a>
9          <a href="#" class="list-group-item list-group-item-action">Site Settings</a>
10         <a href="#" class="list-group-item list-group-item-action">System Logs</a>
11     </div>
12 @endsection
13
```

6. Langkah 6: Buat view untuk User

Buat direktori user di resources/views Kemudian, buat resources/views/user/dashboard.blade.php: Dan masukkan dengan Code Berikut ini:

```
1 @extends('layouts.app')
2
3 @section('title', 'User Dashboard')
4
5 @section('content')
6     <h2 class="mb-4">User Dashboard</h2>
7     <p>Welcome to your dashboard, {{ $username }}!</p>
8     <div class="list-group">
9         <a href="#" class="list-group-item list-group-item-action">View Profile</a>
10        <a href="#" class="list-group-item list-group-item-action">Edit Settings</a>
11        <a href="#" class="list-group-item list-group-item-action">Logout</a>
12    </div>
13 @endsection
14
```

7. Langkah 7: Jalankan Aplikasi

php artisan serve

Buka browser dan kunjungi

- Admin: <http://127.0.0.1:8000/admin>
- User: <http://127.0.0.1:8000/user>

Hasilnya:

Layout and Personalization

Welcome, Yamato Admin

Admin Access Granted

Admin Dashboard

Manage Users

Site Settings

System Logs

© 2025 Layout and Personalization. All rights reserved.

Welcome to your dashboard, Liu User!

User Area

View Profile

Edit Settings

Logout

© 2025 Layout and Personalization. All rights reserved.

Praktikum 4: Partial Views, Blade Components, dan Theme Switching di Laravel 12

1. Langkah 1: Buat Dan Buka Proyek Laravel Baru

- laravel new modul-4-laravel-ui
- cd modul-4-laravel-ui
- code .

2. Langkah 2: Buat Controller

Ketik ini di bash:

```
php artisan make:controller UIController
```

3. Langkah 3: Definisikan Routes

Buka Routes/web.php Dan Tambahkan Code Berikut:

```
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4 use App\Http\Controllers\UIController;
5
6 Route::get('/', [UIController::class, 'home'])->name('home');
7 Route::get('/about', [UIController::class, 'about'])->name('about');
8 Route::get('/contact', [UIController::class, 'contact'])->name('contact');
9 Route::get('/profile', [UIController::class, 'profile'])->name('profile');
10 Route::get('/switch-theme/{theme}', [UIController::class, 'switchTheme'])->name('switch-theme');
11
```

4. Langkah 4: Update Controller

Buka File UIController.php di Direktori app/Http/Controllers/UIController.php

Kemudian Isi dengan Code Berikut:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class UIController extends Controller
8 {
9     public function home(Request $request)
10    {
11         $theme = session('theme', 'light');
12         $alertMessage = 'Selamat datang di Laravel UI Integrated Demo!';
13         $features = [
14             'Partial Views',
15             'Blade Components',
16             'Theme Switching',
17             'Bootstrap 5',
18             'Responsive Design'
19         ];
20
21         return view('home', compact('theme', 'alertMessage', 'features'));
22     }
23
24     public function about(Request $request)
25     {
26         $theme = session('theme', 'light');
27         $alertMessage = 'Halaman ini menggunakan Partial Views!';
28         $team = [
29             ['name' => 'Ahmad', 'role' => 'Developer'],
30             ['name' => 'Sari', 'role' => 'Designer'],
31             ['name' => 'Budi', 'role' => 'Project Manager']
32         ];
33
34         return view('about', compact('theme', 'alertMessage', 'team'));
35     }
36
37     public function contact(Request $request)
38     {
39         $theme = session('theme', 'light');
40         $departments = [
41             'Technical Support',
42             'Sales',
43             'Billing',
44             'General Inquiry'
45         ];
46
47         return view('contact', compact('theme', 'departments'));
48     }
49
50     public function profile(Request $request)
51     {
52         $theme = session('theme', 'light');
53         $user = [
54             'name' => 'Fathan Mubina',
55             'email' => 'fatn.nubina20@gmail.com',
56             'join_date' => '2025-10-09',
57             'preferences' => ['Email Notifications', 'Dark Mode', 'Newsletter']
58         ];
59
60         return view('profile', compact('theme', 'user'));
61     }
62 }
```

```
61     }
62
63     public function switchTheme($theme, Request $request)
64     {
65         if (in_array($theme, ['light', 'dark'])) {
66             session(['theme' => $theme]);
67         }
68         return back();
69     }
70 }
71
```

5. Langkah 5: Buat Layout Utama Dengan Theme Support

Buat Direktori layout di resources/view kemudian Bikin File app.blade.php.

isi Dengan Code Berikut:

```
1  <!DOCTYPE html>
2  <html lang="id" data-bs-theme="{{ $theme }}>
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>@yield('title', 'Laravel UI Integrated Demo')</title>
7      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8      <style>
9          body {
10              padding-top: 4rem;
11              transition: all 0.3s ease;
12              min-height: 100vh;
13          }
14          .theme-demo {
15              border-radius: 10px;
16              padding: 20px;
17              margin: 10px 0;
18              transition: all 0.3s ease;
19          }
20          .feature-card {
21              transition: transform 0.2s ease;
22          }
23          .feature-card:hover {
24              transform: translateY(-5px);
25          }
26      </style>
27  </head>
28  <body class="{{ $theme === 'dark' ? 'bg-dark text-light' : 'bg-light text-dark' }}>
29
30      <!-- Navigation menggunakan Partial View -->
31      @include('partials.navigation')
32
33      <div class="container mt-4">
34
35          <!-- Alert menggunakan Partial View -->
36          @if(isset($alertMessage) && !empty($alertMessage))
37              @include('partials.alert', ['message' => $alertMessage, 'type' => 'info'])
38          @endif
39
40          @yield('content')
41      </div>
42
43      <!-- Footer menggunakan Blade Component -->
44      <x-footer :theme="$theme" />
45
46      <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
47      <script>
48          // Smooth theme transition
49          document.addEventListener('DOMContentLoaded', function() {
50              const themeLinks = document.querySelectorAll('a[href*="switch-theme"]');
51              themeLinks.forEach(link => {
52                  link.addEventListener('click', function(e) {
53                      e.preventDefault();
54                      window.location.href = this.href;
55                  });
56              });
57          });
58      </script>
59  </body>
60 </html>
```

6. Langkah 6: Buat Partial View

Buat Direktori Partial Di resources/view/partials dan bikin file navigation.blade.php

kemudian isi code Berikut:

```
1 <nav class="navbar navbar-expand-lg {{ $theme === 'dark' ? 'navbar-dark bg-dark' : 'navbar-light bg-light' }} fixed-top shadow">
2   <div class="container">
3     <a class="navbar-brand fw-bold" href="{{ route('home') }}>
4       Laravel UI Demo
5     </a>
6
7     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
8       <span class="navbar-toggler-icon"></span>
9     </button>
10
11    <div class="collapse navbar-collapse" id="navbarNav">
12      <ul class="navbar-nav me-auto">
13        <li class="nav-item">
14          <a class="nav-link {{ request()->routeIs('home') ? 'active' : '' }}" href="{{ route('home') }}>Home</a>
15        </li>
16        <li class="nav-item">
17          <a class="nav-link {{ request()->routeIs('about') ? 'active' : '' }}" href="{{ route('about') }}>About</a>
18        </li>
19        <li class="nav-item">
20          <a class="nav-link {{ request()->routeIs('contact') ? 'active' : '' }}" href="{{ route('contact') }}>Contact</a>
21        </li>
22        <li class="nav-item">
23          <a class="nav-link {{ request()->routeIs('profile') ? 'active' : '' }}" href="{{ route('profile') }}>Profile</a>
24        </li>
25      </ul>
26
27      <ul class="navbar-nav">
28        <li class="nav-item dropdown">
29          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
30            Theme: {{ ucfirst($theme) }}
31          </a>
32          <ul class="dropdown-menu">
33            <li><a class="dropdown-item" href="{{ route('switch-theme', 'light') }}>Light Mode</a></li>
34            <li><a class="dropdown-item" href="{{ route('switch-theme', 'dark') }}>Dark Mode</a></li>
35          </ul>
36        </li>
37      </ul>
38    </div>
39  </div>
40 </nav>
41
```

7. Langkah 7: Buat Blade Components

Ketik ini di dalam bash:

- php artisan make:component Footer
- php artisan make:component FeatureCard
- php artisan make:component TeamMember
- php artisan make:component ContactForm

Edit resources/views/components/footer.blade.php:

Isi Dengan Code Berikut:

```
● ● ●
```

```
1  @props(['theme' => 'light'])
2  <footer class="mt-5 py-4 border-top {{ $theme === 'dark' ? 'border-secondary' : '' }}">
3      <div class="container">
4          <div class="row">
5              <div class="col-md-6">
6                  <h5>Laravel UI Integrated Demo</h5>
7                  <p class="mb-0">Demonstrasi Partial Views, Blade Components, dan Theme Switching</p>
8              </div>
9              <div class="col-md-6 text-md-end">
10                 <p class="mb-0">
11                     <strong>Current Theme:</strong>
12                     <span class="badge {{ $theme === 'dark' ? 'bg-primary' : 'bg-dark' }}>
13                         {{ ucfirst($theme) }}
14                     </span>
15                 </p>
16                 <p class="mb-0">&copy; 2024 Laravel UI Demo. All rights reserved.</p>
17             </div>
18         </div>
19     </div>
20 </footer>
21
```

Edit resources/views/components/feature-card.blade.php

Isi Dengan Code Berikut:

```
● ● ●
```

```
1  @props(['theme' => 'light', 'icon' => '★', 'title' => '', 'description' => '', 'badge' => null])
2
3  <div class="card feature-card h-100 {{ $theme === 'dark' ? 'bg-secondary text-white' : '' }}>
4      <div class="card-body">
5          <div class="d-flex align-items-center mb-3">
6              <span class="fs-2 me-3">{{ $icon ?? '★' }}</span>
7              <h5 class="card-title mb-0">{{ $title }}</h5>
8          </div>
9          <p class="card-text">{{ $description }}</p>
10         @if(isset($badge))
11             <span class="badge {{ $theme === 'dark' ? 'bg-light text-dark' : 'bg-dark' }}>{{ $badge }}</span>
12         @endif
13     </div>
14 </div>
15
```

Dan Edit resources/view/components/team-member.blade.php

Isi Dengan Code Berikut:

```
1  @props(['name', 'role', 'theme' => 'light', 'avatar' => null, 'description' => '')  
2  <div class="col-md-4 mb-4">  
3      <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">  
4          <div class="card-body text-center">  
5              <div class="mb-3">  
6                  <span class="fs-1">{{ $avatar ?? '👤' }}</span>  
7              </div>  
8              <h5 class="card-title">{{ $name }}</h5>  
9              <p class="card-text text-muted">{{ $role }}</p>  
10             <p class="card-text">{{ $description }}</p>  
11         </div>  
12     </div>  
13 </div>  
14
```

8. Langkah 8: Buat Main Views Buat View-view Utama:

- resources/views/home.blade.php Isi Dengan Code Berikut:

```
1 @extends('layouts.app')
2
3 @section('title', 'Home - Integrated Demo')
4
5 @section('content')
6 <div class="row">
7     <div class="col-12">
8         <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-5">
9             <h1 class="display-4 mb-4"> Laravel UI Integrated Demo</h1>
10            <p class="lead">Demonstrasi lengkap Partial Views, Blade Components, dan Theme Switching dalam satu aplikasi terpadu.</p>
11
12            <div class="row mt-5">
13                <div class="col-md-4 mb-4">
14                    <x-feature-card
15                        title="Partial Views"
16                        icon="💡"
17                        description="Gunakan @include untuk reusable UI components dengan data sederhana."
18                        badge="Latihan 13"
19                    />
20                </div>
21                <div class="col-md-4 mb-4">
22                    <x-feature-card
23                        title="Blade Components"
24                        icon="✳️"
25                        description="Komponen Blade dengan props dan slots untuk UI yang lebih kompleks."
26                        badge="Latihan 14"
27                    />
28                </div>
29                <div class="col-md-4 mb-4">
30                    <x-feature-card
31                        title="Theme Switching"
32                        icon="⚙️"
33                        description="Toggle antara light dan dark mode dengan session persistence."
34                        badge="Latihan 15"
35                    />
36                </div>
37            </div>
38        </div>
39
40        <div class="row">
41            <div class="col-md-6">
42                <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} mb-4">
43                    <div class="card-header">
44                        <h5>Fitur Utama</h5>
45                    </div>
46                    <div class="card-body">
47                        <ul class="list-group list-group-flush">
48                            @foreach($features as $feature)
49                                <li class="list-group-item {{ $theme === 'dark' ? 'bg-dark text-light' : '' }}>
50                                    - {{ $feature }}
51                                </li>
52                            @endforeach
53                        </ul>
54                    </div>
55                </div>
56            </div>
57            <div class="col-md-6">
58                <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }}>
59                    <div class="card-header">
60                        <h5>Teknologi yang Digunakan</h5>
61                    </div>
62                    <div class="card-body">
63                        <div class="d-flex flex-wrap gap-2">
64                            <span class="badge bg-primary">Laravel 12</span>
65                            <span class="badge bg-success">Blade Templates</span>
66                            <span class="badge bg-info">Bootstrap 5</span>
67                            <span class="badge bg-warning">PHP 8.4</span>
68                            <span class="badge bg-danger">Session Management</span>
69                        </div>
70                    </div>
71                </div>
72            </div>
73        </div>
74    </div>
75 @endsection
76
77
```

- resources/views/about.blade.php Isi Dengan Code Berikut:

```
1  @extends('layouts.app')
2
3  @section('title', 'About - Partial Views Demo')
4
5  @section('content')
6  <div class="row">
7      <div class="col-12">
8          <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
9              <h1 class="mb-4">About - Partial Views</h1>
10             <p class="lead">Halaman ini mendemonstrasikan penggunaan <strong>Partial Views</strong> dengan <code>@include</code> directive.</p>
11         </div>
12
13         <h3 class="mb-4">Tim Kami</h3>
14         <div class="row">
15             @foreach($team as $member)
16                 <x-team-member
17                     :name="$member['name']"
18                     :role="$member['role']"
19                     :theme="$theme"
20                     :avatar="['👤', '👤👤', '👤👤👤'][${loop->index}]"
21                     :description="Bergabung sejak 2024 dan berkontribusi dalam pengembangan.">
22             />
23         @endforeach
24     </div>
25
26     <!-- Demonstrasi Partial View dengan Data -->
27     @include('partials.team-stats', ['theme' => $theme])
28 </div>
29 </div>
30 @endsection
31
```

- resources/views/partials/team-stats.blade.php Isi Dengan Code Berikut:

```
1  <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} mt-4>
2      <div class="card-header">
3          <h5>Statistik Tim</h5>
4      </div>
5      <div class="card-body">
6          <div class="row text-center">
7              <div class="col-md-3">
8                  <h3>3</h3>
9                  <p class="text-muted">Anggota</p>
10             </div>
11             <div class="col-md-3">
12                 <h3>12+</h3>
13                 <p class="text-muted">Proyek</p>
14             </div>
15             <div class="col-md-3">
16                 <h3>95%</h3>
17                 <p class="text-muted">Kepuasan</p>
18             </div>
19             <div class="col-md-3">
20                 <h3>2+</h3>
21                 <p class="text-muted">Tahun</p>
22             </div>
23         </div>
24     </div>
25 </div>
26
```

- resources/views/contact.blade.php Isi Dengan Code Berikut:

```

1  @extends('layouts.app')
2
3  @section('title', 'Contact - Components Demo')
4
5  @section('content')
6      <div class="row">
7          <div class="col-12">
8              <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
9                  <h1 class="mb-4">Contact - Blade Components</h1>
10                 <p class="lead">Halaman ini mendemonstrasikan penggunaan <strong>Blade Components</strong> dengan props dan slots.</p>
11             </div>
12
13             <div class="row">
14                 <div class="col-md-6">
15                     <x-contact-form :theme="$theme" :departments="$departments" />
16                 </div>
17                 <div class="col-md-6">
18                     <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">
19                         <div class="card-header">
20                             <h5>Informasi Kontak</h5>
21                         </div>
22                         <div class="card-body">
23                             <p><strong>Email:</strong> info@laraveldemo.com</p>
24                             <p><strong>Telepon:</strong> +62 21 1234 5678</p>
25                             <p><strong>Alamat:</strong> Jakarta, Indonesia</p>
26
27                             <h6 class="mt-4">Department Tersedia:</h6>
28                             <ul>
29                                 @foreach($departments as $dept)
30                                     <li>{{ $dept }}</li>
31                             @endforeach
32                             </ul>
33                         </div>
34                     </div>
35                 </div>
36             </div>
37         </div>
38     </div>
39     @endsection
40

```

- resources/views/components/contact-form.blade.php Isi dengan Code Berikut:

```

1  @props(['theme' => 'light', 'departments' => []])
2  <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">
3      <div class="card-header">
4          <h5>Form Kontak</h5>
5      </div>
6      <div class="card-body">
7          <form>
8              <div class="mb-3">
9                  <label class="form-label">Nama Lengkap</label>
10                 <input type="text" class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" placeholder="Masukkan nama Anda">
11             </div>
12
13             <div class="mb-3">
14                 <label class="form-label">Email</label>
15                 <input type="email" class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" placeholder="nama@example.com">
16             </div>
17
18             <div class="mb-3">
19                 <label class="form-label">Department</label>
20                 <select class="form-select {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}>
21                     <option selected>Pilih department...</option>
22                     @foreach($departments as $dept)
23                         <option value="{{ $dept }}>{{ $dept }}</option>
24                     @endforeach
25                 </select>
26             </div>
27
28             <div class="mb-3">
29                 <label class="form-label">Pesan</label>
30                 <textarea class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" rows="4" placeholder="Tulis pesan Anda...></textarea>
31             </div>
32
33             <button type="submit" class="btn btn-primary">Kirim Pesan</button>
34         </form>
35     </div>
36 </div>
37

```

- resources/views/profile.blade.php Isi Dengan Code Berikut:

```

1 @extends('layouts.app')
2
3 @section('title', 'Profile - Theme Demo')
4
5 @section('content')
6 <div class="row">
7     <div class="col-12">
8         <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
9             <h1 class="mb-4">Profile - Theme Demo</h1>
10            <p class="lead">Halaman ini menunjukkan implementasi <strong>Theme Switching</strong> dengan session persistence.</p>
11        </div>
12
13        <div class="row">
14            <div class="col-md-4">
15                <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} text-center">
16                    <div class="card-body">
17                        <div class="mb-3">
18                            <span class="fs-1">👤</span>
19                        </div>
20                        <h4>{{ $user['name'] }}</h4>
21                        <p class="text-muted">{{ $user['email'] }}</p>
22                        <p class="text-muted">Bergabung: {{ date('d M Y', strtotime($user['join_date'])) }}</p>
23                    </div>
24                </div>
25            </div>
26
27            <div class="col-md-8">
28                <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }}>
29                    <div class="card-header">
30                        <h5>Preferensi Pengguna</h5>
31                    </div>
32                    <div class="card-body">
33                        <h6>Theme Saat Ini:</h6>
34                        <div class="alert alert-{{ $theme === 'dark' ? 'dark' : 'info' }} d-flex align-items-center">
35                            <span class="me-2 fs-4">{{ $theme === 'dark' ? '🌙' : '☀️' }}</span>
36                        <div>
37                            <strong>Mode {{ ucfirst($theme) }}</strong> -
38                            @if($theme === 'dark')
39                                Nyaman di malam hari dan mengurangi ketegangan mata.
40                            @else
41                                Cerah dan jelas, cocok untuk siang hari.
42                            @endif
43                        </div>
44                    </div>
45
46                    <h6 class="mt-4">Preferensi Lainnya:</h6>
47                    <div class="d-flex flex-wrap gap-2">
48                        @foreach($user['preferences'] as $pref)
49                            <span class="badge bg-secondary">{{ $pref }}</span>
50                        @endforeach
51                    </div>
52
53                    <div class="mt-4">
54                        <h6>Ubah Tema:</h6>
55                        <div class="btn-group" role="group">
56                            <a href="{{ route('switch-theme', 'light') }}">
57                                <span class="btn btn-{{ $theme === 'light' ? 'primary' : 'outline-primary' }}>
58                                    Light Mode
59                                </span>
60                            </a>
61                            <a href="{{ route('switch-theme', 'dark') }}">
62                                <span class="btn btn-{{ $theme === 'dark' ? 'primary' : 'outline-primary' }}>
63                                    Dark Mode
64                                </span>
65                            </a>
66                        </div>
67                    </div>
68                </div>
69            </div>
70        </div>
71    </div>
72 @endsection
73

```

Langkah 9: Jalankan Dan Test Aplikasi

Jalankan php artisan serve di bash

Buka Browser Dan kunjungi:

- Home: <http://127.0.0.1:8000>
- about: <http://127.0.0.1:8000/about>

- Contact: <http://127.0.0.1:8000/contact>
- Profile: <http://127.0.0.1:8000/profile>

Berikut Gambar hasilnya:

- Hasil Home:

The screenshot shows a browser window with the URL 127.0.0.1:8000. The page title is "Laravel UI Demo". A dark-themed header bar contains the "Home", "About", "Contact", and "Profile" links. Below the header, a teal-colored banner displays the message "Selamat datang di Laravel UI Integrated Demo!". The main content area features a large title "Laravel UI Integrated Demo" with a rocket ship icon. A subtitle below it reads "Demonstrasi lengkap Partial Views, Blade Components, dan Theme Switching dalam satu aplikasi terpadu.". Three cards are displayed in a grid:

- Partial Views**: Described as using @include for reusable UI components with simple data. It includes a "Latihan 13" link.
- Blade Components**: Described as Blade components with props and slots for more complex UI. It includes a "Latihan 14" link.
- Theme Switching**: Described as toggling between light and dark mode with session persistence. It includes a "Latihan 15" link.

A "Theme: Dark" dropdown menu is visible in the top right corner of the page.

- Hasil About:

The screenshot shows a browser window with the URL 127.0.0.1:8000/about. The page title is "Laravel UI Demo". The "About" link in the header is highlighted. Below the header, a teal-colored banner displays the message "Halaman ini menggunakan Partial Views!". The main content area features a title "About - Partial Views". A subtitle below it states "Halaman ini mendemonstrasikan penggunaan Partial Views dengan @include directive.". A section titled "Tim Kami" lists three team members in cards:

- Ahmad** (Developer): Joined since 2024 and contributed. Includes a profile icon of a person with a laptop.
- Sari** (Designer): Joined since 2024 and contributed. Includes a profile icon of a person with a paint palette.
- Budi** (Project Manager): Joined since 2024 and contributed. Includes a profile icon of a person with a briefcase.

A "Theme: Dark" dropdown menu is visible in the top right corner of the page.

- Hasil Contact:

Contact - Blade Components

Halaman ini mendemonstrasikan penggunaan **Blade Components** dengan props dan slots.

Form Kontak

Nama Lengkap

Email

Department

Informasi Kontak

Email: info@laraveldemo.com

Telepon: +62 21 1234 5678

Alamat: Jakarta, Indonesia

Department Tersedia:

- Technical Support
- Sales
- Billing
- General Inquiry

- Hasil Profile:

Laravel UI Demo Home About Contact Profile

Theme: Dark

Halaman ini menunjukkan implementasi **Theme Switching** dengan session persistence.

Fathan Mubina

fathn.nubina20@gmail.com

Bergabung: 09 Oct 2025

Preferensi Pengguna

Theme Saat Ini:

Mode Dark - Nyaman di malam hari dan mengurangi ketegangan mata.

Preferensi Lainnya:

Email Notifications **Dark Mode** **Newsletter**

Ubah Tema:

Light Mode **Dark Mode**

3. Hasil dan Pembahasan

Praktikum 1: Meneruskan Data dari Controller ke Blade View

Pada praktikum ini, dilakukan percobaan untuk meneruskan data dari controller ke view menggunakan Blade. Dengan membuat controller bernama `DasarBladeController`, data dikirim melalui fungsi `return view()` beserta variabel yang di-passing. Di sisi view (`dasar.blade.php`), data tersebut ditampilkan menggunakan sintaks `{{ $variabel }}`.

Hasilnya, data dari controller berhasil ditampilkan di browser melalui URL `http://127.0.0.1:8000/dasar`. Hal ini

menunjukkan bahwa proses komunikasi antara controller dan view pada Laravel berjalan dengan baik menggunakan Blade.

Praktikum 2: Menggunakan Struktur Kontrol Blade

Praktikum kedua berfokus pada penggunaan struktur kontrol seperti @if, @else, @foreach, dan @for di dalam template Blade. Controller yang digunakan adalah LogicController, yang mengirimkan data array ke view.

Pada file view (logic.blade.php), data tersebut ditampilkan dalam bentuk daftar menggunakan struktur @foreach. Hasil pengujian menunjukkan bahwa logika percabangan dan perulangan Blade dapat dijalankan dengan benar tanpa perlu menulis kode PHP secara langsung, sehingga tampilan menjadi lebih rapi dan mudah dibaca.

Praktikum 3: Layout dan Personalisasi dengan Bootstrap

Pada praktikum ini, mahasiswa membuat layout utama menggunakan Bootstrap agar tampilan menjadi lebih menarik dan responsif. Struktur layout disimpan dalam file layouts/app.blade.php, kemudian digunakan oleh halaman admin/dashboard.blade.php dan user/dashboard.blade.php melalui direktif @extends. Hasil yang diperoleh menunjukkan bahwa konsep template inheritance pada Blade sangat memudahkan pengelolaan tampilan yang memiliki struktur serupa, seperti halaman admin dan user. Dengan demikian, perubahan pada satu file layout otomatis diterapkan ke seluruh halaman yang mewarisinya.

Praktikum 4: Partial Views, Blade Components, dan Theme Switching

Praktikum terakhir membahas implementasi partial view dan Blade components untuk membangun tampilan yang modular serta mendukung theme switching. Beberapa komponen seperti Footer, FeatureCard, TeamMember, dan ContactForm dibuat menggunakan perintah artisan php artisan make:component. Setiap komponen memiliki file Blade sendiri dan dapat digunakan kembali di berbagai halaman seperti home.blade.php, about.blade.php, contact.blade.php, dan profile.blade.php.

Hasilnya menunjukkan bahwa penggunaan komponen dan partial view membuat kode tampilan lebih terstruktur, mudah dikelola, dan dapat digunakan kembali (reusable). Selain itu, konsep theme switching memungkinkan tampilan aplikasi diubah tanpa perlu memodifikasi setiap halaman secara manual.

Secara keseluruhan, keempat praktikum ini membuktikan bahwa Blade Template Engine pada Laravel sangat efektif dalam membangun tampilan web yang dinamis, modular, dan efisien. Penggunaan fitur Blade juga mempercepat proses pengembangan dan meminimalkan duplikasi kode.

4. Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa Blade Template Engine merupakan salah satu fitur penting dalam framework Laravel yang berfungsi untuk memisahkan logika pemrograman dari tampilan (view) dengan cara yang efisien, terstruktur, dan mudah dipelihara. Melalui empat percobaan yang mencakup penerusan data dari controller ke view, penggunaan struktur kontrol seperti @if dan @foreach, penerapan layout dengan Bootstrap, serta pembuatan partial views dan Blade components, mahasiswa dapat memahami bagaimana Blade mempermudah proses pengembangan tampilan web. Penggunaan Blade Template Engine tidak hanya mempercepat proses pembuatan antarmuka pengguna, tetapi juga meningkatkan konsistensi, keteraturan struktur kode, serta memungkinkan tampilan yang lebih

dinamis dan mudah dikembangkan. Dengan demikian, Blade memberikan kontribusi besar dalam menciptakan aplikasi Laravel yang efisien, terorganisir, dan profesional.

5. Referensi

- Modul 2 - Laravel Fundamentals - <https://hackmd.io/@mohdrzu/r1AIUzWpII>
 - Laravel Documentation — Blade Templates. <https://laravel.com/docs/12.x/blade>
 - TutorialsPoint — Laravel Blade Templates.
https://www.tutorialspoint.com/laravel/laravel_blade_templates.htm
-