

Laporan Modul 4: Laravel Blade Template Engine

Mata Kuliah: Workshop Web Lanjut

Nama: Ammar

NIM: 2024573010129

Kelas: TI2B

Abstrak

Laporan ini membahas penerapan *Blade Template Engine* dalam framework Laravel, yang merupakan sistem templating bawaan Laravel untuk mengelola tampilan (*view*). Blade memungkinkan pengembang menulis kode PHP dalam template dengan sintaks yang lebih ringkas, bersih, dan mudah dibaca.

Melalui praktikum ini, dilakukan implementasi konsep dasar Blade, struktur kontrol, *layouts*, *components*, *partial views*, serta *theme switching* menggunakan *session*.

Hasil akhir menunjukkan bahwa penggunaan Blade mempercepat proses pengembangan antarmuka web, menjaga konsistensi tampilan, dan mendukung penerapan prinsip *Don't Repeat Yourself (DRY)* dalam pengelolaan *view* Laravel.

1. Dasar Teori

1.1 Pengertian Blade

Blade adalah *template engine* bawaan Laravel yang menyediakan cara sederhana dan efisien untuk mengelola tampilan. Blade tidak membatasi penggunaan sintaks PHP dan mendukung fitur seperti *template inheritance*, *components*, dan *sections* untuk membuat tampilan yang dinamis dan modular.

Ciri khas Blade:

- Ringan dan cepat.
- Mendukung *layout inheritance* dan *components*.
- Menggunakan sintaks kontrol seperti `@if`, `@foreach`, `@include`, `@yield`, `@extends`, dan lainnya.

Contoh view sederhana:

```
<h1>Hello, {{ $name }}</h1>
```

Data `$name` diteruskan dari controller menggunakan:

```
return view('greeting', ['name' => 'Agus']);
```

1.2 Struktur Kontrol Blade

Blade menyediakan struktur logika layaknya PHP, tetapi dengan sintaks yang lebih bersih:

```
@if ($user)
    Hello, {{ $user->name }}
@else
    Welcome, Guest!
@endif
```

Untuk perulangan:

```
@foreach ($posts as $post)
    <p>{{ $post->title }}</p>
@endforeach
```

1.3 Layout dan Sections

Blade mendukung konsep layout inheritance untuk membuat tampilan yang konsisten.

Contoh layout (layouts/app.blade.php):

```
<html>
<head>
    <title>My App - @yield('title')</title>
</head>
<body>
    @yield('content')
</body>
</html>
```

Contoh child view:

```
@extends('layouts.app')

@section('title', 'Home Page')

@section('content')
    <h1>Welcome to the Home Page</h1>
@endsection
```

1.4 Blade Components dan Partial Views

Partial View: menggunakan `@include` untuk menyisipkan bagian tampilan berulang seperti navbar atau footer.

Component: digunakan untuk elemen UI kompleks yang dapat menerima props dan slot.

Contoh komponen sederhana:

```
<div class="alert alert-danger">  
    {{ $slot }}  
</div>
```

Digunakan di view:

```
<x-alert>Terjadi kesalahan!</x-alert>
```

2. Langkah-Langkah Praktikum

2.1 Praktikum 1 – Meneruskan Data dari Controller ke Blade View

Langkah-langkah:

1. Buat Projek laravel pada terminal vscode

```
laravel new modul-4-blade-view
```

lalu masuk ke dalam folder projek tsb.

```
cd modul-4-blade-view
```

2. Tambahkan route pada routes/web.php.

```
use Illuminate\Support\Facades\Route;  
use App\Http\Controllers\DasarBladeController;  
  
Route::get('/dasar', [DasarBladeController::class, 'showData']);
```

3. Buat controller DasarBladeController.

Buat file DasarBladeController dengan perintah artisan:

```
php artisan make:controller DasarBladeController
```

Masuk ke file [app/Http/Controllers/DasarBladeController](#) dan isikan code berikut pada class DasarBladeController:

```
public function showData()
{
    $name = 'Devi';
    $fruits = ['Apple', 'Banana', 'Cherry'];
    $user = [
        'name' => 'Eva',
        'email' => 'eva@ilmudata.id',
        'is_active' => true,
    ];
    $product = (object) [
        'id' => 1,
        'name' => 'Laptop',
        'price' => 12000000
    ];

    return view('dasar', compact('name', 'fruits', 'user', 'product'));
}
```

4. Buat view dasar.blade.php.

Buat file **dasar.blade.php** di **resources\views** lalu masukkan kode berikut:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Data Passing Demo</title>
</head>
<body>
    <h1>Passsing Data to Blade View</h1>

    <h2>String</h2>
    <p>Name: {{ $name }}</p>

    <h2>Array</h2>
    <ul>
        @foreach ($fruits as $fruit)
            <li>{{ $fruit }}</li>
        @endforeach
    </ul>

    <!-- <h2>Array Asosiatif</h2>
    <ul>
        @foreach ($user as $key => $value)
            <li>User {{ $key }} = {{ $value }}</li>
        @endforeach
    </ul> -->

    <h2>Associative Array</h2>
```

```
<p>Name: {{ $user['name'] }}</p>
<p>Email: {{ $user['email'] }}</p>
<p>Status: {{ $user['is_active'] ? 'Active' : 'Inactive' }}</p>

<h2>Object</h2>
<p>ID: {{ $product->id }}</p>
<p>Product: {{ $product->name }}</p>
<p>Price: RP{{ number_format($product->price, 0, ',', '.') }}</p>
</body>
</html>
```

5. Jalankan aplikasi dan tunjukkan hasil di browser.

Untuk menjalankan aplikasi kita bisa menggunakan perintah artisan berikut:

```
php artisan serve
```

lalu ctrl+klik <http://127.0.0.1:8000> sehingga akan diredirect ke web browser.

pada URL masukkan </dasar> agar masuk ke halaman web yang telah kita buat.

Screenshot Hasil:

Passsing 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

2.2 Praktikum 2 – Menggunakan Struktur Kontrol Blade

1. Pada projek `modul-4-blade-view` tambahkan route pada `routes/web.php`.

```
use App\Http\Controllers\LogicController;  
  
Route::get('/logic', [LogicController::class, 'show']);
```

2. Buat controller LogicController.

Buat file LogicController dengan perintah artisan:

```
php artisan make:controller LogicController
```

Masuk ke file `app/Http/Controllers/LogicController` dan isikan code berikut pada class LogicController:

```

public function show()
{
    $isLoggedIn = true;
    $users = [
        ['name' => 'Marcel', 'role' => 'admin'],
        ['name' => 'Thariq', 'role' => 'editor'],
        ['name' => 'Ellian', 'role' => 'subscriber'],
    ];
    $products = []; // Simulasi array kosong untuk @forelse
    $profile = [
        'name' => 'Thariq',
        'email' => 'thariq@ilmudata.id'
    ];
    $status = 'active';

    return view('logic', compact('isLoggedIn', 'users', 'products',
        'profile', 'status'));
}

```

3. Buat view logic.blade.php.

Buat file `logic.blade.php` di `resources\views` lalu masukkan kode berikut:

```

<!DOCTYPE html>
<html>
<head>
    <title>Blade Logic Demo</title>
</head>
<body>
    <h1>Blade Control Structures Demo</h1>

    <h2>1. @if / @else</h2>
    @if ($isLoggedIn)
        <p>Welcome back, user!</p>
    @else
        <p>Please log in.</p>
    @endif

    <h2>2. @foreach</h2>
    <ul>
        @foreach ($users as $user)
            <li>{{ $user['name'] }} - Role: {{ $user['role'] }}</li>
        @endforeach
    </ul>

    <h2>3. @forelse</h2>
    @forelse ($products as $product)
        <p>{{ $product }}</p>
    @forelse

```

```
@empty
    <p>No products found.</p>
@endforelse

<h2>4. @isset</h2>
@isset($profile['email'])
    <p>User Email: {{ $profile['email'] }}</p>
@endisset

<h2>5. @empty</h2>
@empty($profile['phone'])
    <p>No phone number available.</p>
@endempty

<h2>6. @switch</h2>
@switch($status)
    @case('active')
        <p>Status: Active</p>
        @break
    @case('inactive')
        <p>Status: Inactive</p>
        @break
    @default
        <p>Status: Unknown</p>
@endswitch
</body>
</html>
```

4. Jalankan aplikasi dan tunjukkan hasil di browser.

Untuk menjalankan aplikasi kita bisa menggunakan perintah artisan berikut:

```
php artisan serve
```

lalu ctrl+klik <http://127.0.0.1:8000> sehingga akan diredirect ke web browser.

pada URL masukkan `/logic` agar masuk ke halaman web yang telah kita buat.

Screenshot Hasil:

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

2.3 Praktikum 3 – Layout dan Personalisasi dengan Bootstrap

Langkah-langkah:

1. Pada projek `modul-4-blade-view` tambahkan route pada `routes/web.php`.

```
use App\Http\Controllers\PageController;

Route::get('/admin', [PageController::class, 'admin']);
Route::get('/user', [PageController::class, 'user']);
```

2. Buat controller `PageController`.

Buat file PageController dengan perintah artisan:

```
php artisan make:controller PageController
```

Masuk ke file `app/Http/Controllers/PageController` dan isikan code berikut pada class PageController:

```
public function admin()
{
    $role = 'admin';
    $username = 'Yamato Admin';
    return view('admin.dashboard', compact('role', 'username'));
}

public function user()
{
    $role = 'user';
    $username = 'Liu User';
    return view('user.dashboard', compact('role', 'username'));
}
```

3. Buat layouts directory di `resources\views` `mkdir resources/views/layouts`

4. Buat file `app.blade.php` di `resources\views\layouts` lalu masukkan kode berikut:

```
<!DOCTYPE html>
<html>
<head>
    <title>@yield('title') | Layout and Personalization</title>
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">
</head>
<body>
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark mb-4">
        <div class="container">
            <a class="navbar-brand" href="#">Layout and Personalization</a>
            <div class="collapse navbar-collapse">
                <ul class="navbar-nav ms-auto">
                    <li class="nav-item">
                        <span class="nav-link active">Welcome, {{ $username }}</span>
                    </li>
                </ul>
            </div>
        </div>
    </nav>
```

```
</nav>

<div class="container">
    @if ($role === 'admin')
        <div class="alert alert-info">Admin Access Granted</div>
    @elseif ($role === 'user')
        <div class="alert alert-success">User Area</div>
    @endif

    @yield('content')
</div>

<footer class="bg-light text-center mt-5 p-3 border-top">
    <p class="mb-0">&copy; 2025 Layout and Personalization. All rights reserved.</p>
</footer>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

5. Buat admin directory di `resources\views` `mkdir resources/views/admin`

6. Buat file `dashboard.blade.php` di `resources\views\admin` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'Admin Dashboard')

@section('content')
    <h2 class="mb-4">Admin Dashboard</h2>
    <div class="list-group">
        <a href="#" class="list-group-item list-group-item-action">Manage Users</a>
        <a href="#" class="list-group-item list-group-item-action">Site Settings</a>
        <a href="#" class="list-group-item list-group-item-action">System Logs</a>
    </div>
@endsection
```

7. Buat user directory di `resources\views` `mkdir resources/views/user`

8. Buat file `dashboard.blade.php` di `resources\views\user` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'User Dashboard')

@section('content')
    

## User Dashboard



Welcome to your dashboard, {{ $username }}!



View Profile
        Edit Settings
        Logout


@endsection
```

9. Jalankan aplikasi dan tunjukkan hasil di browser.

Untuk menjalankan aplikasi kita bisa menggunakan perintah artisan berikut:

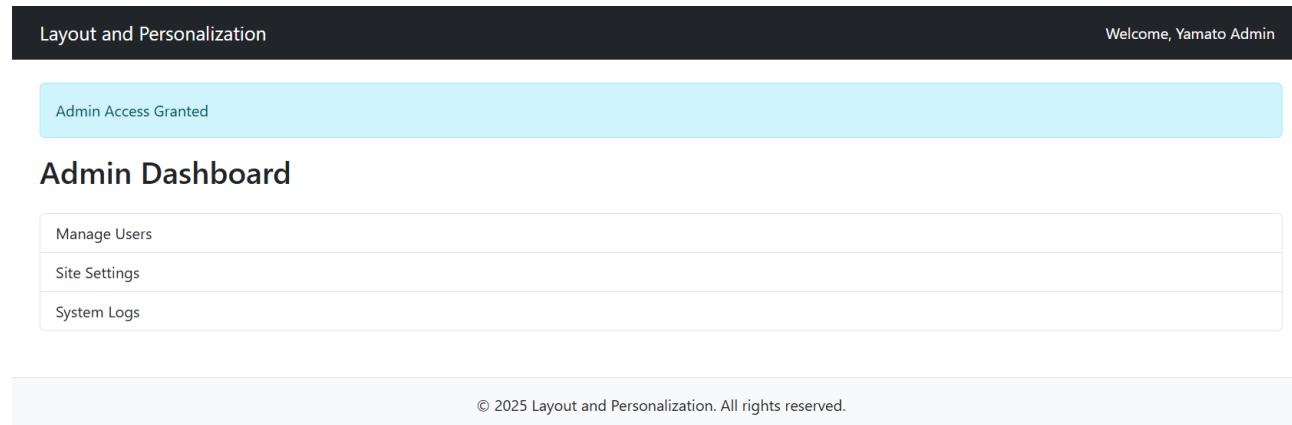
```
php artisan serve
```

lalu ctrl+klik <http://127.0.0.1:8000> sehingga akan diredirect ke web browser.

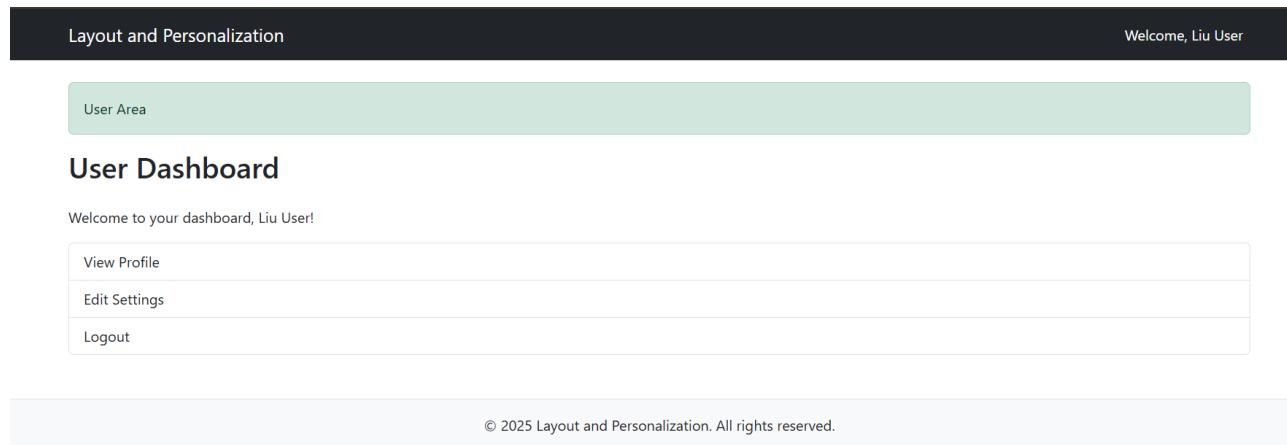
pada URL masukkan `/admin` dan `/user` agar masuk ke halaman web yang telah kita buat.

Screenshot Hasil:

- [/admin](#)



- [/user](#)



2.4 Praktikum 4 – Partial Views, Blade Components, dan Theme Switching

Langkah-langkah:

1. Buat Projek laravel pada terminal vscode

```
laravel new modul-4-laravel-ui
```

lalu masuk ke dalam folder projek tsb.

```
cd modul-4-laravel-ui
```

2. Tambahkan route pada routes/web.php.

```
use App\Http\Controllers\UIController;

Route::get('/', [UIController::class, 'home'])->name('home');
Route::get('/about', [UIController::class, 'about'])->name('about');
Route::get('/contact', [UIController::class, 'contact'])->name('contact');
Route::get('/profile', [UIController::class, 'profile'])->name('profile');
Route::get('/switch-theme/{theme}', [UIController::class, 'switchTheme'])
    ->name('switch-theme');
```

3. Buat controller UIController.

Buat file UIController dengan perintah artisan:

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

Masuk ke file `app/Http/Controllers/UIController` dan isikan code berikut pada class UIController:

```
public function home(Request $request)
{
    $theme = session('theme', 'light');
    $alertMessage = 'Selamat datang di Laravel UI Integrated Demo!';
    $features = [
        'Partial Views',
        'Blade Components',
        'Theme Switching',
        'Bootstrap 5',
        'Responsive Design'
    ];

    return view('home', compact('theme', 'alertMessage', 'features'));
}

public function about(Request $request)
{
    $theme = session('theme', 'light');
    $alertMessage = 'Halaman ini menggunakan Partial Views!';
    $team = [
        ['name' => 'Ahmad', 'role' => 'Developer'],
        ['name' => 'Sari', 'role' => 'Designer'],
        ['name' => 'Budi', 'role' => 'Project Manager']
    ];

    return view('about', compact('theme', 'alertMessage', 'team'));
}

public function contact(Request $request)
{
    $theme = session('theme', 'light');
    $departments = [
        'Technical Support',
        'Sales',
        'Billing',
        'General Inquiry'
    ];

    return view('contact', compact('theme', 'departments'));
}

public function profile(Request $request)
{
    $theme = session('theme', 'light');
    $user = [
        'name' => 'John Doe',
        'email' => 'john.doe@example.com',
        'join_date' => '2024-01-15',
        'preferences' => ['Email Notifications', 'Dark Mode', 'Newsletter']
    ];
}
```

```
        return view('profile', compact('theme', 'user'));
    }

    public function switchTheme($theme, Request $request)
    {
        if (in_array($theme, ['light', 'dark'])) {
            session(['theme' => $theme]);
        }
        return back();
    }
}
```

4. Buat layouts directory di `resources\views` `mkdir resources/views/layouts`

5. Buat file `app.blade.php` di `resources\views\layouts` lalu masukkan kode berikut:

```
<!DOCTYPE html>
<html lang="id" data-bs-theme="{{ $theme }}">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@yield('title', 'Laravel UI Integrated Demo')</title>
    <link
        href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
        rel="stylesheet">
    <style>
        body {
            padding-top: 4rem;
            transition: all 0.3s ease;
            min-height: 100vh;
        }
        .theme-demo {
            border-radius: 10px;
            padding: 20px;
            margin: 10px 0;
            transition: all 0.3s ease;
        }
        .feature-card {
            transition: transform 0.2s ease;
        }
        .feature-card:hover {
            transform: translateY(-5px);
        }
    </style>
</head>
<body class="{{ $theme === 'dark' ? 'bg-dark text-light' : 'bg-light text-dark' }}>

    <!-- Navigation menggunakan Partial View -->
```

```

@include('partials.navigation')

<div class="container mt-4">

    <!-- Alert menggunakan Partial View -->
    @if(isset($alertMessage) && !empty($alertMessage))
        @include('partials.alert', ['message' => $alertMessage, 'type'
=> 'info'])
    @endif

    @yield('content')
</div>

<!-- Footer menggunakan Blade Component -->
<x-footer :theme="$theme" />

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
<script>
    // Smooth theme transition
    document.addEventListener('DOMContentLoaded', function() {
        const themeLinks = document.querySelectorAll('a[href*="switch-
theme"]');
        themeLinks.forEach(link => {
            link.addEventListener('click', function(e) {
                e.preventDefault();
                window.location.href = this.href;
            });
        });
    });
</script>
</body>
</html>

```

6. Buat partials directory di `resources\views` `mkdir resources/views/partials`

7. Buat file `alert.blade.php` di `resources\views\partials` lalu masukkan kode berikut:

```

@if(!empty($message))
<div class="alert alert-{{ $type ?? 'info' }} alert-dismissible fade show"
role="alert">
    {{ $message }}
    <button type="button" class="btn-close" data-bs-dismiss="alert">
</button>
</div>
@endif

```

8. Buat file `navigation.blade.php` di `resources\views\partials` lalu masukkan kode berikut:

```
<nav class="navbar navbar-expand-lg {{ $theme === 'dark' ? 'navbar-dark bg-dark' : 'navbar-light bg-light' }} fixed-top shadow">
    <div class="container">
        <a class="navbar-brand fw-bold" href="{{ route('home') }}">
            Laravel UI Demo
        </a>

        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav">
            <span class="navbar-toggler-icon"></span>
        </button>

        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav me-auto">
                <li class="nav-item">
                    <a class="nav-link {{ request()->routeIs('home') ? 'active' : '' }}" href="{{ route('home') }}">Home</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link {{ request()->routeIs('about') ? 'active' : '' }}" href="{{ route('about') }}">About</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link {{ request()->routeIs('contact') ? 'active' : '' }}" href="{{ route('contact') }}">Contact</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link {{ request()->routeIs('profile') ? 'active' : '' }}" href="{{ route('profile') }}">Profile</a>
                </li>
            </ul>

            <ul class="navbar-nav">
                <li class="nav-item dropdown">
                    <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown">
                        Theme: {{ ucfirst($theme) }}
                    </a>
                    <ul class="dropdown-menu">
                        <li><a class="dropdown-item" href="{{ route('switch-theme', 'light') }}">Light Mode</a></li>
                        <li><a class="dropdown-item" href="{{ route('switch-theme', 'dark') }}">Dark Mode</a></li>
                    </ul>
                </li>
            </ul>
        </div>
    </div>
```

```
</div>
</nav>
```

9. Buat file `team-stats.blade.php` di `resources\views\partials` lalu masukkan kode berikut:

```
<div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} mt-4">
    <div class="card-header">
        <h5>Statistik Tim</h5>
    </div>
    <div class="card-body">
        <div class="row text-center">
            <div class="col-md-3">
                <h3>3</h3>
                <p class="text-muted">Anggota</p>
            </div>
            <div class="col-md-3">
                <h3>12+</h3>
                <p class="text-muted">Proyek</p>
            </div>
            <div class="col-md-3">
                <h3>95%</h3>
                <p class="text-muted">Kepuasan</p>
            </div>
            <div class="col-md-3">
                <h3>2+</h3>
                <p class="text-muted">Tahun</p>
            </div>
        </div>
    </div>
</div>
```

10. Buat laravel component Buat file component dengan laravel artisan php artisan make:component Footer.php artisan make:component FeatureCard.php artisan make:component TeamMember.php artisan make:component ContactForm

11. Masuk ke file `resources\views\components\footer.blade.php` dan isikan code berikut:

```
<!-- Penambahan agar tidak error Undefined -->
@props([
    'theme' => 'light'
])
<footer class="mt-5 py-4 border-top {{ $theme === 'dark' ? 'border-secondary' : '' }}">
    <div class="container">
        <div class="row">
            <div class="col-md-6">
```

```

        <h5>Laravel UI Integrated Demo</h5>
        <p class="mb-0">Demonstrasi Partial Views, Blade Components,
dan Theme Switching</p>
        </div>
        <div class="col-md-6 text-md-end">
            <p class="mb-0">
                <strong>Current Theme:</strong>
                <span class="badge {{ $theme === 'dark' ? 'bg-primary' :
'bg-dark' }}>
                    {{ ucfirst($theme) }}
                </span>
            </p>
            <p class="mb-0">&copy; 2024 Laravel UI Demo. All rights
reserved.</p>
        </div>
    </div>
</div>
</div>
</div>
</footer>

```

12. Masuk ke file `resources\views\components\feature-card.blade.php` dan isikan code berikut:

```

<!-- Penambahan agar tidak error Undefined -->
@props([
    'title',
    'icon',
    'description',
    'theme' => 'light',
    'badge' => null
])
<div class="card feature-card h-100 {{ $theme === 'dark' ? 'bg-secondary
text-white' : '' }}>
    <div class="card-body">
        <div class="d-flex align-items-center mb-3">
            <span class="fs-2 me-3">{{ $icon ?? '★' }}</span>
            <h5 class="card-title mb-0">{{ $title }}</h5>
        </div>
        <p class="card-text">{{ $description }}</p>
        <!-- Pengurangan isset() -->
        @if($badge)
            <span class="badge {{ $theme === 'dark' ? 'bg-light text-dark' :
'bg-dark' }}>{{ $badge }}</span>
        @endif
    </div>
</div>

```

13. Masuk ke file `resources\views\components\team-member.blade.php` dan isikan code berikut:

```
<!-- Penambahan agar tidak error Undefined -->
@props([
    'name',
    'avatar',
    'description',
    'theme' => 'light',
    'role'
])
<div class="col-md-4 mb-4">
    <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">
        <div class="card-body text-center">
            <div class="mb-3">
                <span class="fs-1">{{ $avatar ?? '👤' }}{{ $name }}</h5>
            <p class="card-text text-muted">{{ $role }}</p>
            <p class="card-text">{{ $description }}</p>
        </div>
    </div>
</div>
```

14. Masuk ke file `resources\views\components\contact-form.blade.php` dan isikan code berikut:

```
<!-- Penambahan agar tidak error Undefined -->
@props([
    'theme' => 'light',
    'departments' => []
])
<div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">
    <div class="card-header">
        <h5>Form Kontak</h5>
    </div>
    <div class="card-body">
        <form>
            <div class="mb-3">
                <label class="form-label">Nama Lengkap</label>
                <input type="text" class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" placeholder="Masukkan nama Anda">
            </div>

            <div class="mb-3">
                <label class="form-label">Email</label>
                <input type="email" class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" placeholder="nama@example.com">
            </div>
        </form>
    </div>
</div>
```

```
<div class="mb-3">
    <label class="form-label">Department</label>
    <select class="form-select {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}">
        <option selected>Pilih department...</option>
        @foreach($departments as $dept)
            <option value="{{ $dept }}>{{ $dept }}</option>
        @endforeach
    </select>
</div>

<div class="mb-3">
    <label class="form-label">Pesan</label>
    <textarea class="form-control {{ $theme === 'dark' ? 'bg-dark text-light border-light' : '' }}" rows="4" placeholder="Tulis pesan Anda..."></textarea>
</div>

<button type="submit" class="btn btn-primary">Kirim Pesan</button>
</form>
</div>
</div>
```

15. Buat view home.blade.php.

Buat file `home.blade.php` di `resources\views` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'Home - Integrated Demo')

@section('content')
<div class="row">
    <div class="col-12">
        <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-5">
            <h1 class="display-4 mb-4">🔗 Laravel UI Integrated Demo</h1>
            <p class="lead">Demonstrasi lengkap Partial Views, Blade Components, dan Theme Switching dalam satu aplikasi terpadu.</p>

            <div class="row mt-5">
                <div class="col-md-4 mb-4">
                    <x-feature-card
                        title="Partial Views"
                        icon="📁"
                        description="Gunakan @include untuk reusable UI components dengan data sederhana.">
                </x-feature-card>
            </div>
        </div>
    </div>
</div>
```

```
        badge="Latihan 13"
    />
</div>
<div class="col-md-4 mb-4">
    <x-feature-card
        title="Blade Components"
        icon="⚙️"
        description="Komponen Blade dengan props dan slots
untuk UI yang lebih kompleks."
        badge="Latihan 14"
    />
</div>
<div class="col-md-4 mb-4">
    <x-feature-card
        title="Theme Switching"
        icon="🌙"
        description="Toggle antara light dan dark mode dengan
session persistence."
        badge="Latihan 15"
    />
</div>
</div>

<div class="row">
    <div class="col-md-6">
        <div class="card {{ $theme === 'dark' ? 'bg-dark border-light'
: '' }} mb-4">
            <div class="card-header">
                <h5>Fitur Utama</h5>
            </div>
            <div class="card-body">
                <ul class="list-group list-group-flush">
                    @foreach($features as $feature)
                    <li class="list-group-item {{ $theme === 'dark' ?
'bg-dark text-light' : '' }}">
                        - {{ $feature }}
                    </li>
                    @endforeach
                </ul>
            </div>
        </div>
        <div class="col-md-6">
            <div class="card {{ $theme === 'dark' ? 'bg-dark border-light'
: '' }}>
                <div class="card-header">
                    <h5>Teknologi yang Digunakan</h5>
                </div>
                <div class="card-body">
                    <div class="d-flex flex-wrap gap-2">
```

16. Buat view about.blade.php.

Buat file `about.blade.php` di `resources\views` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'About - Partial Views Demo')

@section('content')
<div class="row">
    <div class="col-12">
        <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
            <h1 class="mb-4">About - Partial Views</h1>
            <p class="lead">Halaman ini mendemonstrasikan penggunaan
<strong>Partial Views</strong> dengan @include directive.</p>
        </div>

        <h3 class="mb-4">Tim Kami</h3>
        <div class="row">
            @foreach($team as $member)
            <x-team-member
                :name="$member['name']"
                :role="$member['role']"
                :theme="$theme"
                :avatar="['👨💻', '👩🎨', '👨💼'][$loop->index]"
                :description="'Bergabung sejak 2024 dan berkontribusi dalam
pengembangan.'"
            />
            @endforeach
        </div>
    </div>
</div>
```

```
    @include('partials.team-stats', ['theme' => $theme])
  </div>
</div>
@endsection
```

17. Buat view contact.blade.php.

Buat file `contact.blade.php` di `resources\views` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'Contact - Components Demo')

@section('content')


<div class="col-12">
    <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
      <h1 class="mb-4">Contact - Blade Components</h1>
      <p class="lead">Halaman ini mendemonstrasikan penggunaan
      <strong>Blade Components</strong> dengan props dan slots.</p>
    </div>

    <div class="row">
      <div class="col-md-6">
        <x-contact-form :theme="$theme" :departments="$departments"
        />
      </div>
      <div class="col-md-6">
        <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} h-100">
          <div class="card-header">
            <h5>Informasi Kontak</h5>
          </div>
          <div class="card-body">
            <p><strong>Email:</strong> info@laraveldemo.com</p>
            <p><strong>Telepon:</strong> +62 21 1234 5678</p>
            <p><strong>Alamat:</strong> Jakarta, Indonesia</p>

            <h6 class="mt-4">Department Tersedia:</h6>
            <ul>
              @foreach($departments as $dept)
                <li>{{ $dept }}</li>
              @endforeach
            </ul>
          </div>
        </div>
      </div>
    </div>
  </div>


```

```
</div>
</div>
@endsection
```

18. Buat view profile.blade.php.

Buat file `profile.blade.php` di `resources\views` lalu masukkan kode berikut:

```
@extends('layouts.app')

@section('title', 'Profile - Theme Demo')

@section('content')


<div class="col-12">
        <div class="theme-demo {{ $theme === 'dark' ? 'bg-dark border-light' : 'bg-white border' }} mb-4">
            <h1 class="mb-4">Profile - Theme Demo</h1>
            <p class="lead">Halaman ini menunjukkan implementasi
            <strong>Theme Switching</strong> dengan session persistence.</p>
        </div>

        <div class="row">
            <div class="col-md-4">
                <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }} text-center">
                    <div class="card-body">
                        <div class="mb-3">
                            <span class="fs-1">👤 </span>
                        </div>
                        <h4>{{ $user['name'] }}</h4>
                        <p class="text-muted">{{ $user['email'] }}</p>
                        <p class="text-muted">Bergabung: {{ date('d M Y', strtotime($user['join_date'])) }}</p>
                    </div>
                </div>
            </div>
        </div>

        <div class="col-md-8">
            <div class="card {{ $theme === 'dark' ? 'bg-dark border-light' : '' }}>
                <div class="card-header">
                    <h5>Preferensi Pengguna</h5>
                </div>
                <div class="card-body">
                    <h6>Theme Saat Ini:</h6>
                    <div class="alert alert-{{ $theme === 'dark' ? 'dark' : 'info' }} d-flex align-items-center">
                        <span class="me-2 fs-4">{{ $theme === 'dark' ?


```

```

'⌚' : '⌚ '} }</span>
        <div>
            <strong>Mode {{ ucfirst($theme) }}</strong>
        -
            @if($theme === 'dark')
                Nyaman di malam hari dan mengurangi
            ketegangan mata.
            @else
                Cerah dan jelas, cocok untuk siang hari.
            @endif
        </div>
    </div>

    <h6 class="mt-4">Preferensi Lainnya:</h6>
    <div class="d-flex flex-wrap gap-2">
        @foreach($user['preferences'] as $pref)
            <span class="badge bg-secondary">{{ $pref }}</span>
        @endforeach
    </div>

    <div class="mt-4">
        <h6>Ubah Tema:</h6>
        <div class="btn-group" role="group">
            <a href="{{ route('switch-theme', 'light') }}"
}}" class="btn btn-{{ $theme === 'light' ? 'primary' : 'outline-primary' }}">
                Light Mode
            </a>
            <a href="{{ route('switch-theme', 'dark') }}"
}}" class="btn btn-{{ $theme === 'dark' ? 'primary' : 'outline-primary' }}">
                Dark Mode
            </a>
        </div>
    </div>
</div>
</div>
</div>
</div>
</div>
@endsection

```

19. Jalankan aplikasi dan tunjukkan hasil di browser.

Untuk menjalankan aplikasi kita bisa menggunakan perintah artisan berikut:

```
php artisan serve
```

lalu ctrl+klik <http://127.0.0.1:8000> sehingga akan diredirect ke web browser.

pada URL masukkan /, /about, /contact, dan /profile agar masuk ke halaman web yang telah kita buat.

Screenshot Hasil:

- / Home page dengan dark theme

The screenshot shows the Laravel UI Integrated Demo home page. At the top, there's a navigation bar with 'Laravel UI Demo' and links for Home, About, Contact, and Profile. A dropdown menu for 'Theme: Dark' is visible. The main title 'Laravel UI Integrated Demo' is displayed with a rocket icon. Below it, a sub-headline says 'Demonstrasi lengkap Partial Views, Blade Components, dan Theme Switching dalam satu aplikasi terpadu.' There are three cards: 'Partial Views' (with a folder icon), 'Blade Components' (with a puzzle piece icon), and 'Theme Switching' (with a color palette icon). Each card has a brief description and a link: 'Latihan 13', 'Latihan 14', and 'Latihan 15'. At the bottom left is a 'Fitur Utama' section with a list: - Partial Views, - Blade Components, - Theme Switching, - Bootstrap 5. At the bottom right is a 'Teknologi yang Digunakan' section with tags: Laravel 12, Blade Templates, Bootstrap 5, PHP 8.4, Session Management.

- /about page dengan white theme

The screenshot shows the 'About - Partial Views' page. The top navigation bar includes 'Laravel UI Demo', 'Home', 'About', 'Contact', 'Profile', and a 'Theme: Light' dropdown. The main title is 'About - Partial Views'. Below it, a sub-headline says 'Halaman ini mendemonstrasikan penggunaan Partial Views dengan @include directive.' A 'Tim Kami' section lists three team members: Ahmad (Developer), Sari (Designer), and Budi (Project Manager), each with a profile picture and a brief bio. At the bottom is a 'Statistik Tim' section showing statistics: 3 Anggota, 12+ Proyek, 95% Kepuasan, and 2+ Tahun.

- /contact page dengan white theme

Contact - Blade Components

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

Form Kontak

Nama Lengkap

Email

Department

Pesan

Informasi Kontak

Email: info@laraveldemo.com

Telepon: +62 21 1234 5678

Alamat: Jakarta, Indonesia

Department Tersedia:

- Technical Support
- Sales
- Billing
- General Inquiry

- [/profile](#) page dengan dark theme

Profile - Theme Demo

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



John Doe

john.doe@example.com

Bergabung: 15 Jan 2024

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

Dari seluruh praktikum, diperoleh hasil bahwa:

- Blade mempercepat pembuatan view yang dinamis dan efisien.
- Layouts dan components membantu menjaga konsistensi desain.
- Partial views efektif untuk menyertakan elemen UI berulang.
- Dengan theme switching, tampilan aplikasi menjadi lebih interaktif dan adaptif terhadap preferensi pengguna.

- Struktur modular Blade mendukung prinsip DRY (Don't Repeat Yourself) dan meningkatkan keterbacaan serta pemeliharaan kode.

4. Kesimpulan

1. Blade Template Engine merupakan fitur penting Laravel yang menyederhanakan pengelolaan tampilan dan meningkatkan efisiensi pengembangan web.
2. Melalui penggunaan layouts, components, dan partials, pengembang dapat membangun antarmuka yang konsisten, terstruktur, dan mudah diperluas.
3. Selain itu, penerapan theme switching menunjukkan kemampuan Blade dalam mendukung interaktivitas dan personalisasi UI. Dengan memahami Blade secara mendalam, pengembang dapat membuat aplikasi Laravel yang lebih modular, bersih, dan profesional.

5. Referensi

- Laravel Documentation – Blade Templates: <https://laravel.com/docs/12.x/blade>
- Modul 4 – Laravel Blade Template Engine – <https://hackmd.io/@mohdrzu/r1AIUzWplI>