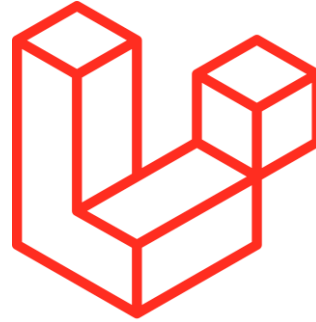


Pemrograman Backend



Pesantren PeTIK II YBM PLN

Jl. KH. Bisri Syansuri RT/01 RW/05, Plosogeneng,
Kec. Jombang, Kabupaten Jombang, Jawa Timur



4. Blade Templating



Pengertian Blade Template

Blade adalah fitur yang disediakan Laravel untuk proses templating sederhana namun sangat bermanfaat dalam proses pengembangan tampilan halaman web. Tidak seperti fitur templating PHP populer lainnya, Blade 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.

File tampilan blade menggunakan ekstensi file `.blade.php` dan biasanya disimpan di direktori `resources/views`. Dalam hal ini Blade pada Laravel menggunakan basis template inheritance dan sections.





Fungsi Blade Template

- Salah satu fungsi dari Laravel Blade templating adalah penggunaan layout, agar tampilan yang berulang misalnya header, footer, sidebar dan sebagainya tidak perlu dibuat berkali-kali sehingga rawan inkonsistensi.
- Pada umumnya layout ditempatkan dalam sebuah folder yang bernama layout dalam folder views, namun demikian programmer tidak dibatasi apabila ingin menggunakan nama lain.



Menampilkan Data dengan Template Blade(1)

Kita dapat menampilkan data yang ditampilkan dengan template blade. Kita bisa membungkus variabel dalam kurung kurawal. Misalnya dengan menambahkan route berikut ini:

```
//Tambahkan route baru di routes/web.php
Route::get('/hello', function () {
    return view('hello', ['name' => 'Inaya']);
});
```



Menampilkan Data dengan Template Blade(2)


Selanjutnya buat file baru Bernama hello.blade.php di bawah folder resources/views. Silahkan tambahkan kode seperti di bawah ini:

Hello, `{{ $name }}`.



Menampilkan Data dengan Template Blade(3)

Selanjutan di terminal kita jalankan perintah: `php artisan serve`. Maka dalam route di url bisa diakses `http://localhost:8000/hello`, maka tampilah seperti di bawah ini:

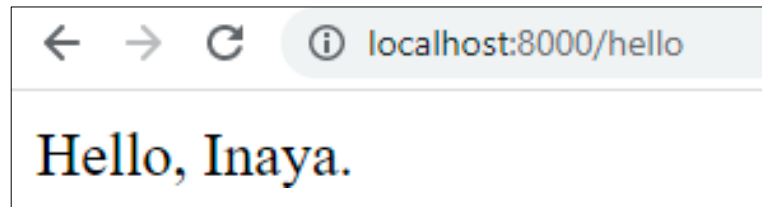
```
routes >  web.php > ...
60
61 Route::get('/hello', function () {
62     return view('hello', ['name' => 'Inaya']);
63 });
64
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS E:\xampp\htdocs\example-app> `php artisan serve`



Struktur Kendali dengan Template Blade(1)

Kita bisa menggunakan struktur kendali dengan template blade. Tambahkan route baru di routes/web.php seperti kode di bawah ini:

```
//Tambahkan route baru di routes/web.php
Route::get('/nilai', function () {
    return view('nilai');
});
```





Struktur Kendali dengan Template Blade(2)

Tambahkan file baru di views/nilai.blade.php seperti kode di bawah ini:

```
@php
$nama = "Fawwaz";
$nilai = 90;
@endphp
{{-- Struktur Kendali If --}}
@if($nilai >= 60) @php $ket = "Lulus"; @endphp
@else @php $ket = "Gagal"; @endphp
@endif
{{-- Mencetak --}}
Nama Siswa : {{ $nama }}
<br/>Nilai : {{ $nilai }}
<br/>Keterangan : {{ $ket }}
```



Struktur Kendali dengan Template Blade(3)

Selanjutnya di terminal kita jalankan perintah: `php artisan serve`. Maka dalam route di url bisa diakses `http://localhost:8000/nilai`, maka tampilah seperti di bawah ini:

routes > `php` web.php > ...

```
49 Route::get('/nilai', function () {  
50     return view('nilai');  
51 });  
52
```

OUTPUT

TERMINAL

DEBUG CONSOLE

PROBLEMS

`php`

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS E:\xampp\htdocs\example-app> `php` artisan serve

Starting Laravel development server: `http://127.0.0.1:8000`

`localhost:8000/nilai`

Nama Siswa : Fawwaz

Nilai : 90

Keterangan : Lulus

Looping dengan Template Blade(1)

Kita bisa menggunakan looping dengan template blade. Tambahkan route baru di routes/web.php seperti kode di bawah ini:

```
//Tambahkan route baru di routes/web.php
Route::get('/daftarnilai', function () {
    return view('daftar_nilai');
});
```



Looping dengan Template Blade(2a)

Tambahkan file baru di views/daftar_nilai.blade.php seperti kode di bawah ini:

```
@php
$no = 1;
//array scalar
$s1 = ['nama'=>'Fawwaz', 'nilai'=>90];
$s2 = ['nama'=>'Inaya', 'nilai'=>89];
$s3 = ['nama'=>'Bedu', 'nilai'=>59];
$s4 = ['nama'=>'Mimin', 'nilai'=>80];
$judul = ['No','Nama','Nilai','Keterangan'];
//array associative
$siswa = [$s1,$s2,$s3,$s4];
@endphp
```



Looping dengan Template Blade(2b)

Lanjutan file daftar_nilai.blade.php:

```
<h3 align="center">Daftar Nilai Siswa</h3>
<table border="1" align="center" cellpadding="10">
  <thead>
    <tr bgcolor="pink">
      @foreach($judul as $jd1)
        <th>{{ $jd1 }}</th>
      @endforeach
    </tr>
  </thead>
```





Looping dengan Template Blade(2c)


Lanjutan file daftar_nilai.blade.php:

```
<tbody>
    @foreach($siswa as $sis)
        {{-- logic kelulusan dan warna-warni dgn ternary --}}
        @php
            $ket = ($sis['nilai'] >= 60) ? 'Lulus' : 'Gagal';
            $warna = ($no % 2 == 0) ? 'greenyellow' : 'yellow';
        @endphp
        <tr bgcolor="{{ $warna }}">
            <td>{{ $no++ }}</td>
            <td>{{ $sis['nama'] }}</td>
            <td>{{ $sis['nilai'] }}</td>
            <td>{{ $ket }}</td>
        </tr>
    @endforeach
</tbody>
</table>
```



Looping dengan Template Blade(3)

Selanjutnya di terminal kita jalankan perintah: `php artisan serve`. Maka dalam route di url bisa diakses `http://localhost:8000/daftarnilai`, maka tampilah seperti di bawah ini:

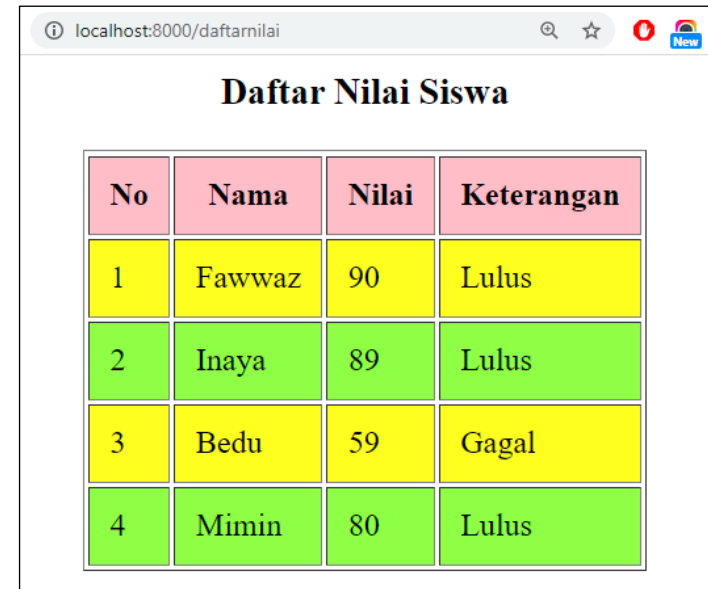
```
routes >  web.php > ...
52
53 Route::get('/daftarnilai', function () {
54     return view('daftar_nilai');
55 });|
56
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

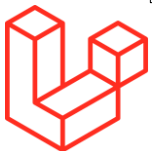
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS E:\xampp\htdocs\example-app> **php** artisan serve



No	Nama	Nilai	Keterangan
1	Fawwaz	90	Lulus
2	Inaya	89	Lulus
3	Bedu	59	Gagal
4	Mimin	80	Lulus



Template Inheritance

- Template inheritance di dalam laravel dikenal juga dengan layout extension.
- Layout inheritance adalah fitur yang memungkinkan kita untuk menggunakan menggunakan kode program lain pada kode program lainnya.
- Contohnya adalah kita dapat menuliskan kode program untuk header, footer sekali saja dan menggunakannya disemua tampilan.
- Kerangka tampilan yang akan kita buat akan mengikuti layout yang sudah kita tentukan, yaitu terdapat header, content, dan footer.



Membuat Kerangka Template Inheritance(1)

Buatlah file app.blade.php pada folder resources/views/layouts, seperti kode di bawah ini:

```
<html>
  <head>
    <title>@yield('title')</title>
  </head>
  <body>
    @include('layouts/header')
    <div style="margin-top: 20px; margin-bottom: 20px">
      @yield('content')
    </div>
    @include("layouts/footer")
  </body>
</html>
```



Membuat Kerangka Template Inheritance(2)

- Pada file di slide sebelumnya ada 2 keyword `@include` dan `@yield`.
- **@include** adalah blade helper yang digunakan untuk mengikutikan kode program dari file lain. Helper ini akan menerima satu parameter, yaitu lokasi dari file yang akanikutsertakan. Pada contoh di slide sebelumnya, parameter yang diberikan adalah `layouts/header` , artinya kita akan mengikutsertakan kode program dari file `resources/views/layouts/header.blade.php`
- **@yield** adalah blade helper yang digunakan untuk menyediakan suatu tempat yang nantinya akan diisi oleh kode program dari file lain.



Membuat Potongan Template Inheritance(1)

Buatlah file header.blade.php di folder resources/views/layouts seperti kode di bawah ini:

```
<div style="background: #388BF2; font-size: 24px; color: white">  
    PHP Framework  
</div>
```



Membuat Potongan Template Inheritance(2)

Buatlah file footer.blade.php di folder resources/views/layouts seperti kode di bawah ini:

```
<div style="background: #31B057; font-size: 20px; color: white">  
    Created by: Nasrul @ 2021  
</div>
```



Membuat Konten Template Inheritance

Buatlah file index.blade.php di folder resources/views/layouts seperti kode di bawah ini:

```
@extends('layouts.app')
@section('title')
    PHP Framework
@endsection
@section('content')
<ul>
    <li>Laravel</li>
    <li>Yii2</li>
    <li>CI</li>
</ul>
@endsection
```



Membuat Route Template Inheritance

Kita bisa menggunakan template inheritance di laravel. Tambahkan route baru di routes/web.php seperti kode di bawah ini:

```
//Tambahkan route baru di routes/web.php
Route::get('/phpframework', function ()
    { return view('layouts.index');
});
```



Menampilkan Template Inheritance

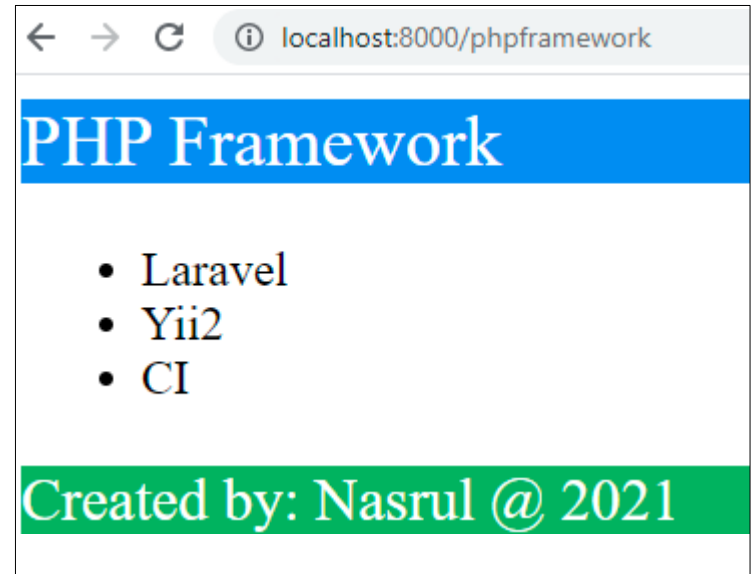
Selanjutnya di terminal kita jalankan perintah: `php artisan serve`. Maka dalam route di url bisa diakses `http://localhost:8000/phpframework`, maka tampilah seperti di bawah ini:

```
routes > php web.php > ...
69 Route::get('/daftarnilai', function () {
70     return view('daftar_nilai');
71 });
```

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS `php` + v

Try the new cross-platform PowerShell <https://aka.ms/powershell>

```
PS E:\xampp\htdocs\example-app> php artisan serve
```





Binding Data dan Injeksi

Saat menempatkan ID model ke routing, Anda akan sering meminta database untuk mengambil model yang sesuai dengan ID itu. Laravel route model binding menyediakan cara mudah untuk secara otomatis memasukkan contoh model langsung ke route Anda. Misalnya, alih-alih menginjeksi ID pengguna, Anda dapat menginjeksi seluruh instance model User yang cocok dengan ID yang diberikan. Contoh route dengan binding data dan injeksi sebagai berikut:

```
use App\Models\User;
```

```
Route::get('/users/{user}', function (User $user) {  
    return $user->email;  
});
```



Kustom Blade Directives

- Blade Directives adalah mesin templating yang mengkompilasi sintak yang dikembalikan ke PHP dan HTML.
- Blade Directives berfungsi menambahkan kode yang menyembunyikan kode kompleks.
- Pada slide sebelumnya kita telah membahas blade directive: kondisi if, variable, array, looping dan komentar di dalam blade template.
- Blade directives memungkinkan juga untuk Anda kustom.
- Untuk mengkustom blade directive dengan sintak di bawah ini:

```
\Blade::directive('directive_name', function ($expression) {  
    return $expression;  
});
```





Membuat Kustom Blade Directives

Buatlah fungsi baru di file AppServiceProvider.php seperti kode di bawah ini:

```
<?php
namespace App\Providers;
use Illuminate\Support\Facades\Blade;
use Illuminate\Support\ServiceProvider;
class AppServiceProvider extends ServiceProvider
{
    public function boot()
    {
        Blade::directive('hello', function ($expression) {
            return "<?php echo 'Hello ' . {$expression}; ?>";
        });
    }
}
```



TERIMA KASIH
ATAS SEGALA PERHATIAN
SEMOGA BERMANFAAT...

