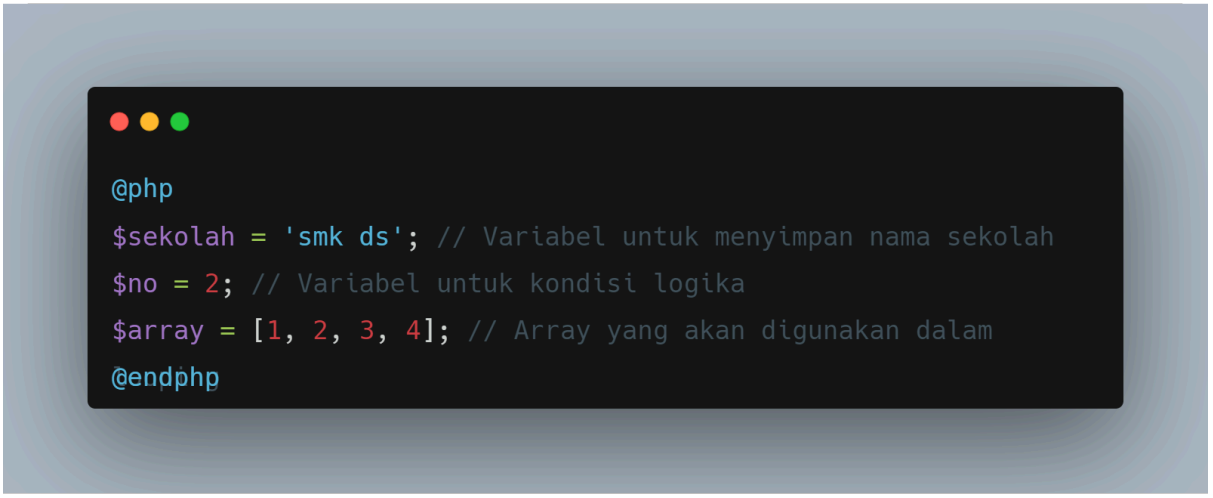
Pertemuan 2

Dokumentasi Template Blade Laravel

File `index.blade.php` merupakan template view yang menggunakan Blade, template engine dari Laravel. Berikut adalah penjelasan dari kode yang ada dalam file tersebut:

Deklarasi PHP

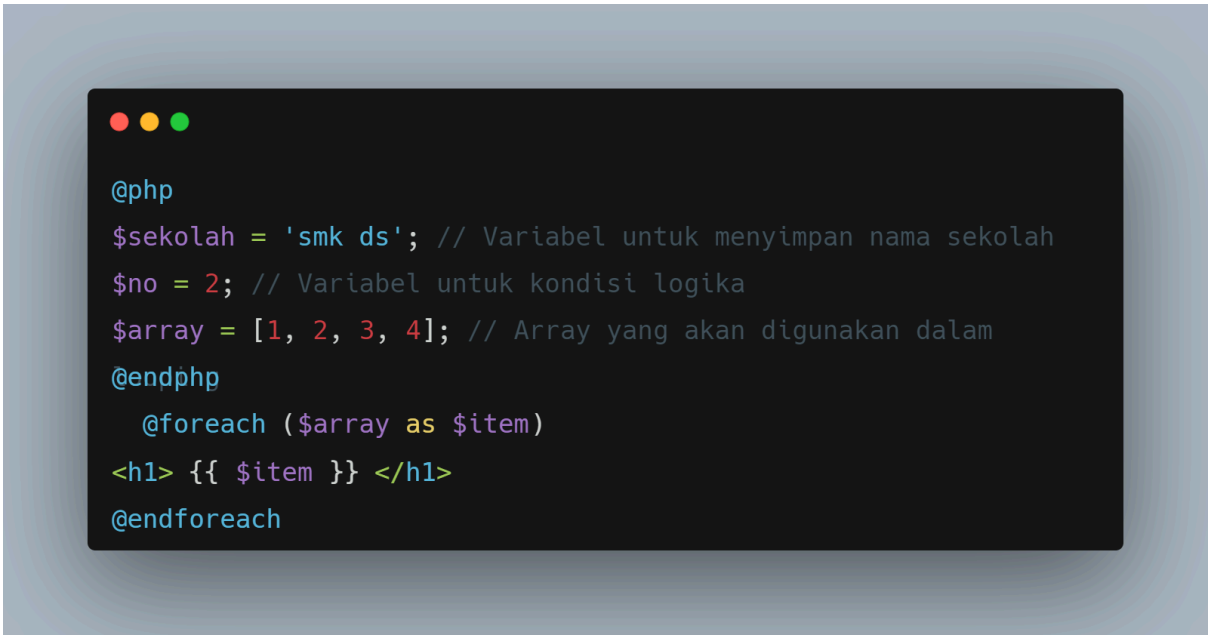
Di awal file, terdapat blok kode PHP yang digunakan untuk mendeklarasikan beberapa variabel yang akan digunakan dalam view:



```php
@php
$sekolah = 'smk ds'; // Variabel untuk menyimpan nama sekolah
$no = 2; // Variabel untuk kondisi logika
$array = [1, 2, 3, 4]; // Array yang akan digunakan dalam
@endphp
```

Looping dengan Blade

Kode berikut menggunakan direktif `@foreach` untuk melakukan iterasi atas array `$array`. Untuk setiap elemen dalam array, elemen tersebut ditampilkan dalam tag `<h1>`:



```php
@php
$sekolah = 'smk ds'; // Variabel untuk menyimpan nama sekolah
$no = 2; // Variabel untuk kondisi logika
$array = [1, 2, 3, 4]; // Array yang akan digunakan dalam
@endphp
  @foreach ($array as $item)
<h1> {{ $item }} </h1>
@endforeach
```

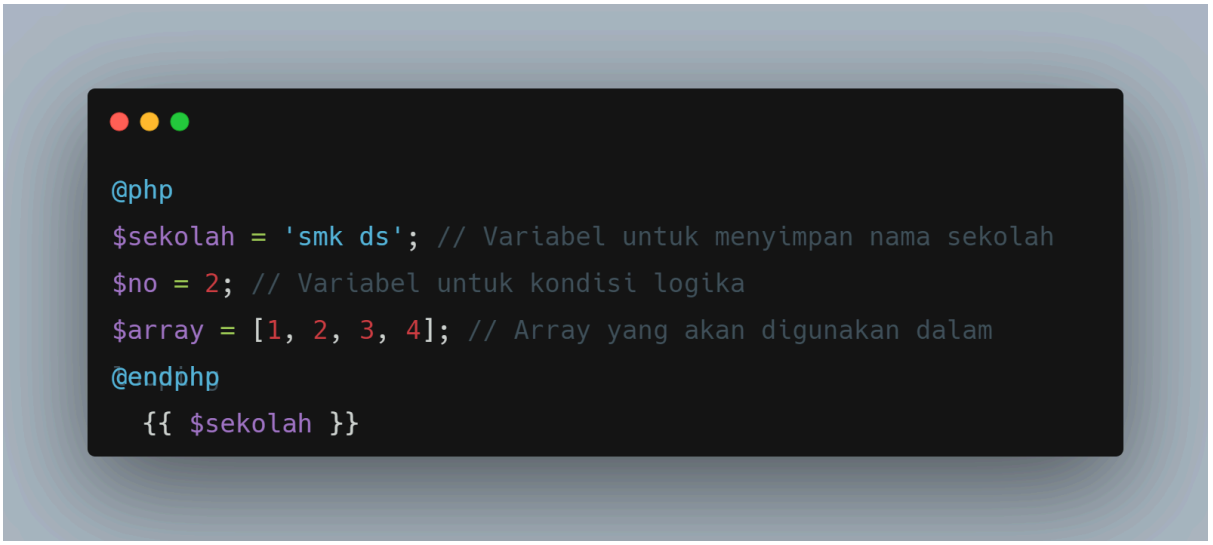
Kondisional dengan Blade

Kode ini menampilkan sebuah button dengan teks yang berbeda tergantung pada nilai dari variabel `$no`. Jika `$no` sama dengan 1, maka akan menampilkan "no 1", jika tidak, maka akan menampilkan "no bukan 1":

```
@php
$sekolah = 'smk ds'; // Variabel untuk menyimpan nama sekolah
$no = 2; // Variabel untuk kondisi logika
$array = [1, 2, 3, 4]; // Array yang akan digunakan dalam
@endphp
  @if($no == 1)
<button>no {{ $no }}</button>
@else
<button>no bukan 1</button>
@endif
```

Menampilkan Data

Kode ini menampilkan nilai dari variabel `$sekolah` yang telah didefinisikan sebelumnya:



```
@php
$sekolah = 'smk ds'; // Variabel untuk menyimpan nama sekolah
$no = 2; // Variabel untuk kondisi logika
$array = [1, 2, 3, 4]; // Array yang akan digunakan dalam
@endphp
  {{ $sekolah }}
```

Kesimpulan

Template ini menunjukkan penggunaan dasar dari Blade termasuk deklarasi variabel, looping, kondisional, dan menampilkan data. Pastikan semua variabel yang digunakan sudah didefinisikan atau diinisialisasi sebelumnya untuk menghindari error.

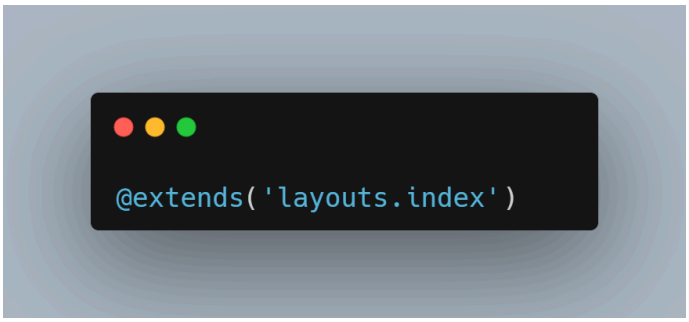
Laravel Blade: `@section` dan `@extends`

Laravel Blade adalah template engine yang disediakan oleh Laravel, yang memungkinkan Anda untuk mengelola tampilan aplikasi dengan lebih efisien dan bersih. Dua direktif yang sangat penting dalam Blade adalah `@section` dan `@extends`.

`@extends`

Direktif `@extends` digunakan untuk menentukan layout atau template induk yang akan digunakan oleh view tertentu. Ini memungkinkan Anda untuk mendefinisikan struktur dasar halaman pada satu tempat dan menggunakannya di berbagai halaman.

Contoh Penggunaan:



Dalam contoh di atas, view akan menggunakan layout yang didefinisikan dalam file resources/views/layouts/index.blade.php.

`@section`

Direktif `@section` digunakan untuk mendefinisikan bagian konten yang akan di-"inject" ke dalam layout yang ditentukan oleh `@extends`. Anda bisa mendefinisikan beberapa `@section` dalam satu view, dan setiap bagian tersebut akan mengisi bagian yang sesuai di layout.

Contoh Penggunaan:

Dalam contoh di atas, semua yang berada di antara `@section('content')` dan `@endsection` akan dimasukkan ke dalam bagian `content` dari layout yang telah didefinisikan oleh `@extends`.

Integrasi `@section` dan `@extends`

Ketika menggunakan `@extends` dan `@section` bersamaan, Anda menciptakan hubungan antara view dan layoutnya. Layout bertindak sebagai kerangka, sementara view menyediakan isi spesifik yang berbeda-beda tergantung kebutuhan halaman.

Contoh Lengkap:

```
<!-- File: resources/views/home/index.blade.php -->
@extends('layouts.index')
@section('content')
    <!-- Isi spesifik untuk halaman ini -->  Ini adalah konten halaman utama.
@endsection
```

Dalam contoh ini, `index.blade.php` meng-extend `layouts.index` dan mendefinisikan isi untuk `@section('content')` yang akan ditampilkan di tempat yang sesuai di layout `layouts.index`.

Penggunaan `@extends` dan `@section` memudahkan pengelolaan tampilan yang konsisten di seluruh aplikasi sambil memungkinkan variasi konten yang fleksibel di setiap halaman.

Dokumentasi Controller dan Cara Membuatnya

1. Membuat Controller

Untuk membuat HomeController di Laravel, Anda dapat menggunakan Artisan CLI yang disediakan oleh Laravel. Buka terminal Anda dan jalankan perintah berikut:

`php artisan make:controller HomeController`

Perintah ini akan membuat file controller baru di dalam direktori **`app/Http/Controllers`** dengan nama **`HomeController.php`**.

2. Menulis Fungsi di Controller

Setelah controller dibuat, Anda bisa membuka file **`HomeController.php`** dan menambahkan method yang diperlukan. Misalnya, kita akan membuat method **`index`** yang akan mengembalikan sebuah view.

HomeController.php

```php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    public function index()
    {
        return view('home.index'); // asumsikan Anda memiliki file view di
resources/views/home/index.blade.php
    }
}
```

3. Menghubungkan Controller ke Web Routes

Setelah controller dan method-nya siap, Anda perlu menghubungkannya dengan web route di Laravel. Ini telah Anda lakukan di file **`routes/web.php`** seperti yang terlihat pada potongan kode yang Anda berikan:

web.php

```php
use App\Http\Controllers\HomeController;

Route::get('/', [HomeController::class, 'index'])->name('home');
```

Dengan kode di atas, Laravel akan mengarahkan permintaan HTTP GET ke root URL (`/`) ke method **`index`** pada **`HomeController`.**

4. Membuat View

Untuk menyelesaikan alur, Anda perlu membuat view yang akan dikembalikan oleh method `index`. Buat file `index.blade.php` di dalam direktori `resources/views/home`.

index.blade.php

```html
<!DOCTYPE html>
<html>
<head>
    <title>Home Page</title>
</head>
<body>
    <h1>Welcome to the Home Page</h1>
</body>
</html>
```

Dengan langkah-langkah di atas, Anda telah berhasil membuat **`HomeController`**, menghubungkannya dengan route, dan menyiapkan view yang sesuai. Setiap kali pengguna mengakses URL root, mereka akan melihat halaman Home yang telah Anda definisikan.