

MODUL 8

VIEW MANGGUNAKAN *TEMPLATE BLADE*



CAPAIAN PEMBELAJARAN

1. Mampu membuat view menggunakan Blade template
2. Mengimplementasikan komponen view



KEBUTUHAN ALAT/BAHAN/SOFTWARE

1. Sistem Operasi Linux/Window 10
2. PHP 5.x, diutamakan versi 7.x keatas
3. Composer
4. XAMPP (PHP, Apache, MySQL)
5. Text Editor Notepad++ atau Visual Studio Code



DASAR TEORI

1. Apa Blade itu?

Blade adalah mesin *template* sederhana, namun hebat yang disediakan bersama Laravel untuk membuat view. Tidak seperti mesin *template* PHP populer lainnya, Blade juga tidak membatasi untuk membangun view menggunakan kode PHP biasa. Sebenarnya, semua tampilan Blade dikompilasi menjadi kode PHP biasa dan di-*cache* sampai dimodifikasi, artinya Blade menambahkan dasarnya tidak ada *overhead* ke aplikasi yang ada. File tampilan blade menggunakan ekstensi file `.blade.php` dan biasanya disimpan di direktori sumber `resources/views`.

2. Menampilkan data unescaped

Secara default, pernyataan Blade `{{ }}` otomatis dikirim melalui fungsi html special chars PHP untuk mencegah serangan XSS. Supaya tidak ingin data Anda diloloskan, bisa menggunakan sintaks berikut:

```
Hello, {!! $nama !!}
```

Berhati-hatilah saat menggunakan konten yang diberikan oleh pengguna aplikasi. Selalu gunakan sintaks kurung ganda atau `{!! !!}`, untuk mencegah serangan XSS saat menampilkan data yang diberikan user.

3. Kerangka Blade & JavaScript

Karena banyak kerangka JavaScript juga menggunakan `"{{"` untuk menunjukkan ekspresi tertentu yang harus ditampilkan di browser, Anda dapat menggunakan simbol `@` untuk memberi tahu mesin *rendering* Blade sebuah ekspresi.

Sebagai contoh:

```
<h1>Laravel</h1>
Hello, @{{ nama }}.
```

4. Pernyataan If

Pernyataan if menggunakan `@if`, `@elseif`, `@else`, dan `@endif`. Cara ini berfungsi identik dengan penulisan PHP.

Contoh:

```
@if (status==0)
    {{ "Rusak" }}
}elseif (status==1)
    {{ "Baik" }}
@else
    {{ "Tidak diketahui" }}
@endif
```

5. Loop

Blade menyediakan arahan sederhana untuk bekerja dengan struktur loop PHP. Sekali lagi, masing-masing arahan ini berfungsi identik perintah pada PHP, contoh:

```
@for ($i=1; $i<=10; $i++)
    {{ $i }}
@endfor
```

Penulisan perulangan `while` menggunakan blade, seperti pada skrip berikut:

```
@while ($i <= 10)
    {{ $i ++ }}
@endwhile
```

Penulisan pernyataan perulangan foreach menggunakan blade

```
<?php
@foreach($arr as $value)
{
    $value=$value + 1;
    {{ $value }};
}
@endforeach
```



PRAKTIK

1. Buatlah controller menggunakan perintah artisan seperti berikut:
php artisan make:controller BldController
2. Buka file BldController.php tambahkan seperti pada kode program berikut:

```
<?php

namespace App\Http\Controllers;

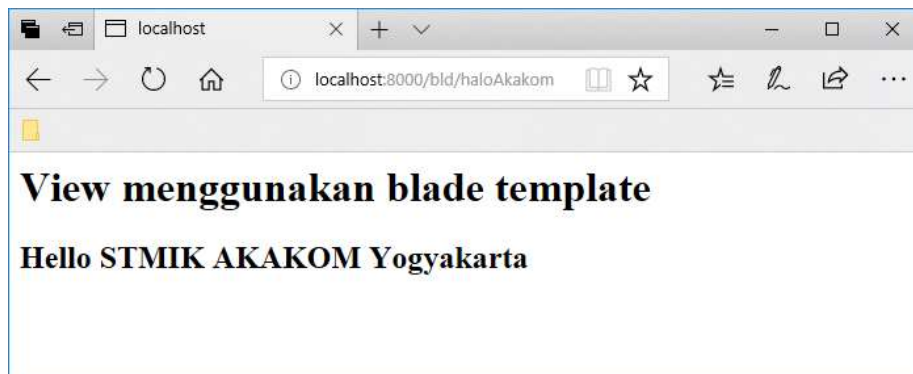
use Illuminate\Http\Request;
use Illuminate\Support\Facades\View;

class BldController extends Controller
{
    public function haloAkakom()
    {
        return View::make('bld.haloAkakom')
            ->with('nama', 'STMIK AKAKOM Yogyakarta');
    }
}
```

6. Buatlah view haloAkakom.blade.php kemudian simpan ke folder
views/bld/... seperti pada kode program berikut:

```
<h1>View menggunakan blade template </h1>
<h2>Hello {{ $nama }} </h2>
```

Perhatikan hasilnya:



7. Jika haloAkakom.blade.php ditambahkan

```
@extends('layouts.app')
@section('content')
    <h1>View menggunakan blade template </h1>
    <h2>Hello {{ $nama }} </h2>
@endsection
```

Amati hasilnya apa perbedaannya?

3. Menggunakan loop for dan logika if, buka file BldController.php tambahkan seperti pada kode program berikut:

```
public function loopFor()
{
    return View::make('bld.loop');
}
```

8. Buatlah view loop.blade.php kemudian simpan ke folder views/bld/... seperti pada kode program berikut:

```
@extends('layouts.app')
@section('content')
    <table border="1">
    <h2>Menampilkan menggunakan Loop</h2>
    <tr><th>NILAI I</th><th>BILANGAN</th></tr>
    @for ($i=0; $i<10; $i++)
    <tr>
        @if ($i%2==1)
```

```

        <td>{{ $i }}</td><td>Ganjil</td>
    @else
        <td>{{ $i }}</td><td>Genap</td>
    @endif
@endfor
</tr>
</table>
@endsection

```

4. Menggunakan rekamanan tabel, tambahkan file BldController.php seperti pada kode program berikut:

```

public function bacaTabel()
{
    $kota=\App\models\Kota::all();
    return View::make('bld.bacaTabel')->with('kota',$kota);
}

```

9. Buatlah file view bacaTabel.blade.php simpan ke dalam folder views/bld/... seperti kode berikut:

```

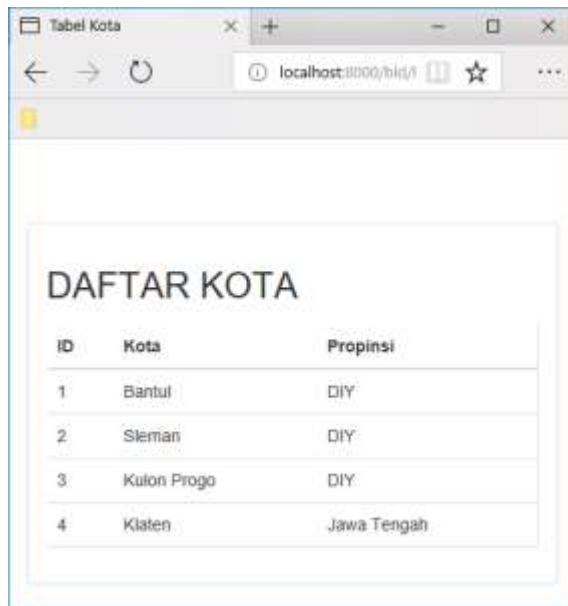
@extends('layouts.app')

@section('content')
<h2>DAFTAR KOTA</h2>
<table class="table table-responsive mtop-sm">
    <thead>
        <th>ID</th>
        <th>Kota</th>
        <th>Propinsi</th>
    </thead>
    <tbody>
        @foreach ($kota as $k)
            <tr>
                <td>{{ $k->id }}</td>
                <td>{{ $k->nama_kota }}</a>
                <td>{{ $k->getPropinsi->propinsi }}</a>

            </td>
            </tr>
        @endforeach
    </tbody>
</table>
@endsection

```

Hasilnya seperti pada gambar berikut:



ID	Kota	Propinsi
1	Bantul	DIY
2	Sleman	DIY
3	Kulon Progo	DIY
4	Klaten	Jawa Tengah

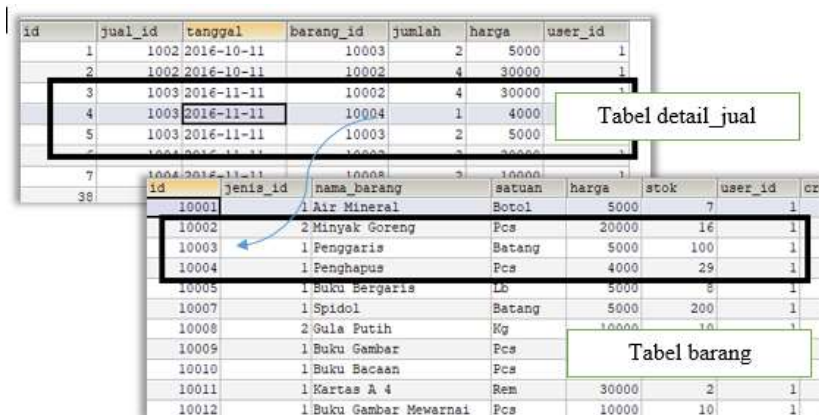


LATIHAN

1. Buatlah laporan daftar barang seperti tabel berikut menggunakan tabel barang di pertemuan sebelumnya:

Kode	Nama Barang	Stok	Satuan	Hara	Jumlah Harga
9999	xxxxxxxxxxxxxxxxxxx	999	xxxxxxxxx	999.999.999	999.999.999
9999	xxxxxxxxxxxxxxxxxxx	999	xxxxxxxxx	999.999.999	999.999.999
dst					

2. Sebuah “Toko Serba Ada” mempunyai data transaksi dengan struktur tabel dan rekaman seperti pada, gambar berikut:

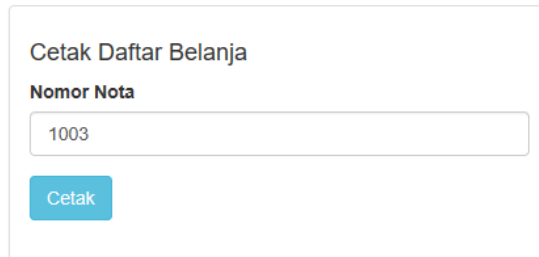


id	jual_id	tanggal	barang_id	jumlah	harga	user_id
1	1002	2016-10-11	10003	2	5000	1
2	1002	2016-10-11	10002	4	30000	1
3	1003	2016-11-11	10002	4	30000	1
4	1003	2016-11-11	10004	1	4000	1
5	1003	2016-11-11	10003	2	5000	1
6	1003	2016-11-11	10003	3	30000	1
7	1004	2016-11-11	10003	3	10000	1
38						

id	jenis_id	nama_barang	satuan	harga	stok	user_id	cre
10001	1	Air Mineral	Botol	5000	7	1	
10002	2	Minyak Goreng	Pcs	20000	16	1	
10003	1	Penggaris	Batang	5000	100	1	
10004	1	Penghapus	Pcs	4000	29	1	
10005	1	Buku Bergaris	Lb	5000	8	1	
10007	1	Spidol	Batang	5000	200	1	
10008	2	Gula Putih	Kg	10000	10	1	
10009	1	Buku Gambar	Pcs				
10010	1	Buku Bacaan	Pcs				
10011	1	Kertas A 4	Rem	30000	2	1	
10012	1	Buku Gambar Mewarnai	Pcs	10000	10	1	

3. Tambahkan controller seperti pada skrip berikut dan buatlah view masukan **No Nota** menggunakan data `jual_id`, seperti tampilan berikut:

```
public function cetakNota()  
{  
    return view('bld.CetakNota');  
}
```



4. Untuk membuat daftar belanja buatlah skrip dalam controller seperti pada program berikut:

```
public function bacaTabel1(Request $request)  
{  
    $djual =  
        \Illuminate\Support\Facades\DB::table('detail_jual as d')  
        ->join('barang as b', 'd.barang_id', '=', 'b.id')  
        ->select('d.barang_id', 'b.nama_barang', 'b.satuan',  
            'd.jumlah', 'd.harga')  
        ->selectRaw('d.jumlah * d.harga as total')  
        ->where('jual_id', $request->jual_id)  
        ->get();  
  
    return View::make('bld.bacaTabel1')  
        ->with(['djual'=>$djual,  
            'no_nota'=>$request->jual_id);  
}
```

5. Tambahkan beberapa barang yang lain pada `jual_id = 1003`,
4. Buatlah view `bacaTabel1.blade.php` hingga hasil keluaran seperti pada gambar berikut:

<p>Toko Serba Ada Jalan Janti 143 Yogyakarta</p> <p>DAFTAR BELANJA</p> <p>No Nota: 1003</p> <table> <tr> <th>ID</th><th>Nama Barang</th><th>Jumlah</th><th>Satuan</th><th>Harga</th><th>Total</th></tr> <tr> <td>10002</td><td>Minyak Goreng</td><td>4</td><td>Pcs</td><td>30.000</td><td>120.000</td></tr> <tr> <td>10004</td><td>Penghapus</td><td>1</td><td>Pcs</td><td>4.000</td><td>4.000</td></tr> <tr> <td>10003</td><td>Penggaris</td><td>2</td><td>Batang</td><td>5.000</td><td>10.000</td></tr> <tr> <td colspan="5">Jumlah</td><td>134.000</td></tr> </table> <p>Terima kasih atas kunjungan Anda</p>						ID	Nama Barang	Jumlah	Satuan	Harga	Total	10002	Minyak Goreng	4	Pcs	30.000	120.000	10004	Penghapus	1	Pcs	4.000	4.000	10003	Penggaris	2	Batang	5.000	10.000	Jumlah					134.000
ID	Nama Barang	Jumlah	Satuan	Harga	Total																														
10002	Minyak Goreng	4	Pcs	30.000	120.000																														
10004	Penghapus	1	Pcs	4.000	4.000																														
10003	Penggaris	2	Batang	5.000	10.000																														
Jumlah					134.000																														



TUGAS

1. Apa fungsi perintah `@include`, `@section`, `@extends`, `@field`, `@datetime($var)` pada template blade



REFERENSI

1. Blade Templates, <https://laravel.com/docs/5.8/blade>
2. Blade Templates, <https://laravel-guide.readthedocs.io/en/latest/blade/>
3. Blade Bagian 1: Berkenalan Dengan Template Engine,
4. <http://id-laravel.com/post/blade-bagian-1-berkenalan-dengan-template-engine>
5. Cheers, KodeBlog Team, Laravel 5 Views, <https://kode-blog.io/laravel-5-views>