

LAPORAN

PEMROGRAMAN WEB LANJUT

Dosen Pengampu :

Lintang Setyo Kurniawati S.Kom, M.Pd



DISUSUN OLEH :

BAYU KRISNA (2010651095)

TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH JEMBER
2022

BAB 4

Dasar Teori

View merupakan bagian yang menampilkan informasi untuk disampaikan kepada users. View terdiri dari script HTML dengan bantuan CSS dan javascript. Sesuai dengan aturan konsep MVC, di dalam view tidak boleh ada script logika maupun script untuk mengakses database. Di dalam laravel view diletakan di dalam folder resources/views/. Sedangkan blade template merupakan engine yang disediakan laravel untuk memudahkan developer dalam menampilkan data pada view. Dengan blade template ini kita tidak perlu lagi menggunakan untuk menampilkan variabel data pada view. Untuk menggunakan blade template, file view harus disimpan dengan akhiran .blade.php dan disimpan pada folder resources/views.

Membuat View

Untuk mempraktekan pembuatan view, terlebih dahulu kita membuat sebuah route untuk mengakses view tersebut. Route yang dibuat akan langsung mengarah ke controller produkController.php yang telah dibuat pada latihan minggu. Route yang dibuat adalah seperti pada sintak dibawah:

```
1 Route::get('/produk',[ProdukController::class, 'index']);
```

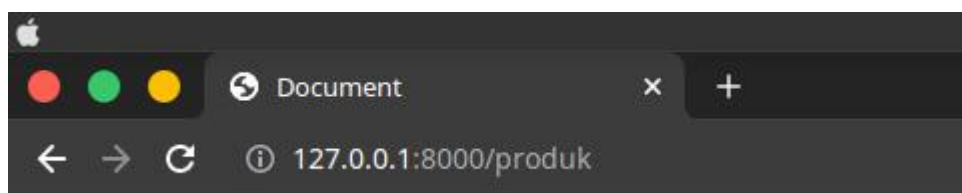
Pada route diatas, jika pada browser dituliskan alamat localhost:8000/produk maka kita akan diarahkan ke class produkController ke dalam fungsi index. Selanjutnya bukanlah file produkController.php yang ada di dalam folder app\Http\Controller dan kemudian modifikasi script nya menjadi seperti berikut:

```
1 class ProdukController extends Controller
2 {
3     public function index()
4     {
5         return view('produk.index',[
6             'produk'=>'Aqua 150 L'
7         ]);
8     }
9 }
10 }
```

Di dalam fungsi `index()` yang telah dibuat kita mendefinisikan satu variabel bernama `produk` yang langsung kita ikut sertakan lempar bersama dengan `return view` yang dimana `view` yang akan dicari berada pada folder proyek Laravel yaitu `penjualan/resources/views/produk/` dengan nama `index.blade.php`. karena `view` tersebut belum dibuat maka buatlah satu file baru pada folder `penjualan/resources/views/produk/` dan beri nama `index.blade.php` dan isikan skript sebagai berikut:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5     <title>Document</title>
6 </head>
7 <body>
8     <h1>Produk : {{$produk}} </h1>
9 </body>
10 </html>
```

Pada script diatas kita menampilkan value dari variabel `produk` dengan hanya menggunakan tanda `{{` dan ditutup dengan tanda `}}`. Fitur yang memungkinkan untuk menampilkan data dengan cara tersebut yaitu `blade template` yang akan dibahas pada materi selanjutnya. Dan jika kita membuka link `localhost:8000/produk` maka akan muncul halaman seperti berikut:



Produk : Aqua 150 L

Menampilkan data dengan blade

. Dengan blade template, kita dapat memodifikasi sintak yang ada pada file index.blade.php yang berada pada folder penjualan/resources/views/produk/ menjadi seperti berikut:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4
5     <title>Document</title>
6 </head>
7 <body>
8     <h1>Produk : {{isset($produk) ? $produk : 'Produk Kosong'}} </h1>
9 </body>
10 </html>
```

Control structure

Blade template juga menyediakan cara tersendiri untuk melakukan logika percabangan maupun perulangan. Untuk melakukan percabangan dengan blade template dapat menggunakan @if.. @else.. @endif. Untuk mencobanya buatlah satu route baru pada file web.php yang berada pada folder routes/web.php dengan sintak sebagai berikut:

```
1 Route::get('/produk/show',[ProdukController::class,'show']);
```

Kemudian pada class controller produkController.php yang berada pada folder app\Http\Controller tambahkan satu fungsi baru bernama show() dan isikan script sebagai berikut:

```
1 public function show(){
2     return view('produk.show',[
3         "produk"=>['Aqua 115 ML','Minuman Bersoda','Buku Sejarah','Mouse','CPU']
4     ]);
5 }
```

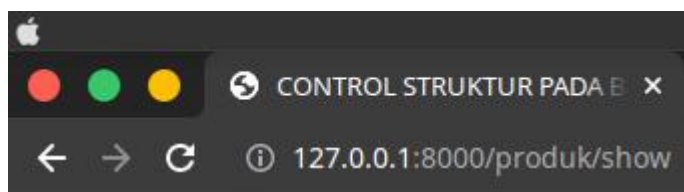
Selanjutnya buatlah view baru dengan nama show.blade.php pada folder resources/views/produk dan isikan skript seperti dibawah:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>CONTROL STRUKTUR PADA BLADE TEMPLATE</title>
5 </head>
6 <body>
7     @if (count($produk)==1)
8         <p>Jumlah Produk adalah satu</p>
9     @elseif(count($produk)>1)
10        <p>Jumlah Produk Lebih dari satu</p>
11    @else
12        <p>Tidak Ada Data Produk</p>
13    @endif
14 </body>
15 </html>

```

Jika dicoba dijalankan dengan mengakses link localhost:8000/produk/show maka tampilannya akan menjadi seperti pada gambar dibawah:

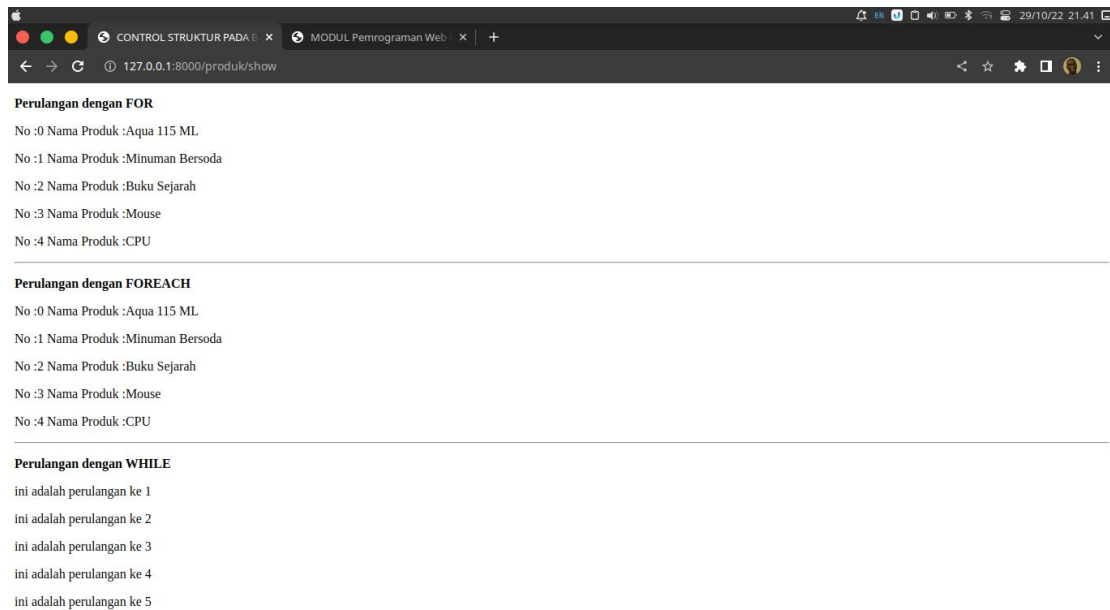


Sedangkan untuk perulangan pada blade template mendukung berbagai jenis perulangan, dengan contoh sebagai berikut:

Perulangan dengan @for, @foreach dan @while, untuk mempraktekan penggunaan perulangan menggunakan blade template kita masih akan menggunakan route localhost:8000/produk/show. Pada class Controller produkController.php di dalam method show() kita telah mendefinisikan satu array bernama produk, pada view show.blade.php yang berada pada folder resources/views/produk modifikasi koding di dalamnya menjadi seperti dibawah:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CONTROL STRUKTUR PADA BLADE TEMPLATE</title>
5 </head>
6 <body>
7     @if (count($produk)==1)
8         <p>Jumlah Produk adalah satu</p>
9     @elseif(count($produk)>1)
10        <p><b>Perulangan dengan FOR</b></p>
11        @for ($i = 0; $i < count($produk); $i++)
12            <p>{{ 'No :'.$i.' Nama Produk :'.$produk[$i] }}</p>
13        @endfor
14        <hr>
15        <p><b>Perulangan dengan FOREACH</b></p>
16        @foreach ($produk as $x => $v)
17            <p>{{ 'No :'.$x.' Nama Produk :'.$v }}</p>
18        @endforeach
19        <hr>
20        <p><b>Perulangan dengan WHILE</b></p>
21        @php
22            $no = 1;
23        @endphp
24        @while ($no<=5)
25            <p>ini adalah perulangan ke {{ $no }}</p>
26            @php
27                $no++;
28            @endphp
29        @endwhile
30    @else
31        <p>Tidak Ada Data Produk</p>
32    @endif
33 </body>
34 </html>
```

Jika halaman pada link localhost:8000/produk/show di refresh atau dimuat ulang maka halamannya akan nampak seperti pada gambar dibawah;



Layout dengan blade

Untuk membuat layout pada template aplikasi, Laravel menggunakan cara yang mudah menggunakan blade template. Untuk mempraktekannya buatlah satu folder baru yang bernama layout pada folder resource/views, lalu buatlah file dengan nama layout.blade.php di dalamnya dan isikan script sebagai berikut:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>@yield('title')</title>
5 </head>
6 <body>
7 <header>
8     @include('layout.header')
9 </header>
10 <ul>
11     @section('sidebar')
12         <li>HTML</li>
13         <li>CSS</li>
14         <li>JS</li>
15 @show
16 </ul>
17 <div class="container">
18     @yield('content')
19 </div>
20 </body>
21 </html>
```


Selanjutnya buatlah satu file baru pada folder yang sama yaitu resources/views/layout dengan nama header.blade.php dan isikan script sebagai berikut:

```
1 <h1>Layout dengan Blade di Latavel</h1>
```

Selanjutnya untuk membuat contoh halaman yang menggunakan layout. Buatlah satu folder baru dan bernama konten pada folder resourcecess/views untuk menyimpan contoh halaman yang akan kita buat. Misalnya kita buat satu file dengan nama halaman.blade.php dengan script sebagai berikut:

```
1 @extends('layouts.layout')
2 @section('title',$title)
3 @section('sidebar')
4 @parent
5 <li>PHP</li>
6 @endsection
7 @section('content')
8 <p>{{ $konten }}</p>
9 @stop
```

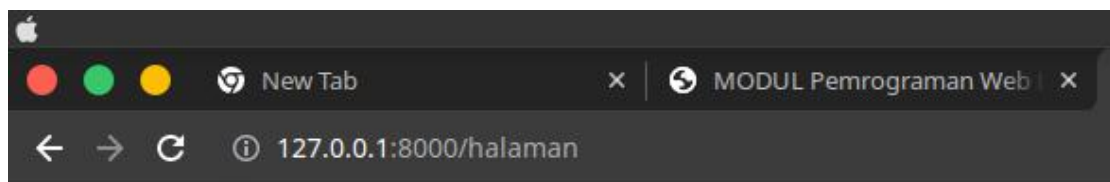
Untuk membuktikan hasil dari pembuatan layout diatas, silahkan buat route pada file web.php pada file routes/web.php dengan script sebagai berikut:


```

1 Route::get('/halaman',function(){
2     $title = 'Harry Pooter';
3     $konten = 'harry potter and the deathly hallows: part 2';
4     return view('konten.halaman',[
5         "title"=>$title,
6         'konten'=>$konten
7     ]);
8
9 });

```

Pada script diatas kita membuat variabel \$title dan \$konten yang akan dikirimkan ke view dengan fungsi compact(). Jika kita membuka localhost:8000/halaman maka hasilnya akan nampak seperti gambar dibawah:



Layout dengan Blade di Latavel

- HTML
- CSS
- JS
- PHP

harry potter and the deathly hallows: part 2