

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 3

Routing

Routing berperan sebagai penghubung antara user dengan keseluruhan framework. Dalam laravel, setiap alamat web yang kita ketik di web browser akan melewati route terlebih dahulu. Route lah yang menentukan mana proses yang akan dibawa, apakah ke controller atau ke view

Cara membuat Route

Untuk pembuatan aplikasi route dibuat pada file routes/web.php. contoh penulisan route sederhana seperti gambar berikut

```
1 Route::get('/', function(){  
2     echo 'welcome';  
3 });
```

Lalu jika kita mengetikkan perintah di browser 127.0.0.1:8000/ maka hasilnya seperti dibawah



Jenis Jenis Route Method

1. Untuk menampilkan atau mengambil data dapat menggunakan method GET. Route dengan method ini dapat langsung kita akses melalui link pada browser. Contohnya dapat dilihat pada latihan dibawah

```
1 Route::get('/index', function () {  
2     echo "Uji Coba route dengan method GET";  
3 });
```

2. Untuk mengirim data dari form dengan dapat menggunakan method POST, biasanya digunakan untuk menambah data. Adapun penulisan dari route ini adalah sebagai berikut:

```
1 Route::post('/store', function () {  
2     // sintak untuk menyimpan data  
3 });
```

3. Untuk mengirim data dari form dengan tujuan untuk memperbaharui data dapat menggunakan method PUT, contoh penulisan route dengan method PUT adalah sebagai berikut:

```
1 Route::put('/update', function () {  
2     // sintak untuk update data  
3 });
```

4. Untuk mengirim data dari form dengan tujuan untuk menghapus data dapat menggunakan method DELETE. Contoh dari penulisan route dengan method delete adalah sebagai berikut:

```
1 Route::delete('/delete', function () {  
2     // sintak untuk menghapus data  
3 });
```

5. Untuk route yang dapat merespons beberapa HTTP request, kita dapat menggunakan method `match()`.

```
1 Route::match(['get', 'post'], '/welcome' function () {  
2     //  
3 });
```

6. Sedangkan untuk route yang dapat merespons semua HTTP request, dapat menggunakan method `any()`. Contohnya pada script berikut:

```
1 Route::any('/welcome' function () {  
2     //  
3 });
```

Cara Penulisan Route

1. Route Dengan Parameter

Parameter pada route ditandai dengan tanda `{ }`. Artinya kata pada URL yang berada pada posisi ini akan menjadi nilai dari parameter. Untuk penulisannya buatlah satu route baru pada file `web.php` yang ada pada folder `routes/web.php` dengan skrip sebagai berikut:

```
1 Route::get('/show/{id}', function ($id) {  
2     echo "Nilai Parameter Adalah ".$id;  
3 });
```

Kemudian bukalah browser dan ketikkan pada link `127.0.0.1:8000/show/5` angka 5 tersebut akan menjadi parameter dari route yang dibuat yang artinya bebas diinputkan berapa saja dan jika dilihat pada browser maka hasilnya akan menjadi seperti gambar dibawah:

```
1 Route::get('/show/{id?}', function ($id=1) {  
2     echo "Nilai Parameter Adalah ".$id;  
3 });
```

2. Route dengan Regular Expression

Route ini akan dijalankan hanya jika nilai dari parameter pada URL memenuhi syarat tertentu. Misalnya pada baris pertama contoh di bawah, parameter judul harus berupa karakter A sampai Z atau a sampai z. Dengan demikian, maka URL `localhost:8000/edit/samsul` memenuhi syarat dari route tersebut, sedangkan URL `http://127.0.0.1:8000/edit/123` tidak dapat menggunakan route tersebut. Adapun contoh penulisan route dengan prasyarat dapat dapat dipraktekan dengan menambahkan route dibawah pada file `web.php` pada folder `routes/web.php`.

```
1 Route::get('/edit/{nama}', function ($nama) {  
2     echo "Nilai Parameter Adalah ".$nama;  
3 })->where('nama', '[A-Za-z]+');
```

3. Route dengan nama

Route dengan pendefinisian nama atau identitas memiliki kelebihan yang dapat digunakan hanya dengan menyebutkan nama yang diberikan. Contohnya buatlah 2 route baru pada file web.php yang ada pada folder routes/web.php

```
1 Route::get('/index', function () {  
2     echo "<a href='".route('create')."'>Akses Route dengan nama </a>";  
3 });
```

Route diatas akan digunakan untuk menampilkan tag dalam atribut href nya akan diisi nama dari route yang akan diakses.

```
1 Route::get('/create', function () {  
2     echo "Route diakses menggunakan nama";  
3 })->name('create');
```

Route diatas adalah route dengan pendefinisian nama yang akan coba kita akses dari route diatasnya.

4. Route dengan Controller

Route seperti ini ketika dijalankan akan mengakses controller yang disebutkan pada parameter kedua. Jika ingin mengakses method atau function tertentu pada controller, maka antara nama controller dengan nama fungsi dipisahkan dengan tanda @. Untuk mempraktekan route ini buatlah satu controller dengan menggunakan perintah artisan sebagai berikut: ***php artisan make:controller ProdukController***

Perintah artisan diatas akan menghasilkan satu file baru bernama produkController.php yang terletak di folder app\Http\Controller. Bukalah file controller tersebut dan tambahkan satu fungsi bernama index seperti pada contoh skrip dibawah:

```
1 class ProdukController extends Controller  
2 {  
3     public function index()  
4     {  
5         return 'Mengakses Fungsi di Controller menggunakan route';  
6     }  
7 }
```