id-laravel.com

Halo Bro: Membuat Halaman Web Pertama Dengan Laravel

Kali ini kita akan bersama-sama membuat halaman web dengan laravel. Karena hello world sudah terlalu mainstream, maka kita akan memodifkasinya sedikit. Ya, kita akan membuat sebuah halaman sederhana, tidak ada fungsi macam-macam selain menampilkan teks “**Halo, bro**”.

Router

Kebanyakan framework PHP melakukan mapping otomatis antara URL dengan `controller`. Oleh sebab itu, untuk membuat sebuah halaman, Anda minimal harus membuat satu buah controller. Tetapi, laravel melakukan pendekatan yang sedikit berbeda. Mapping URL harus didaftarkan secara manual di file `app/routes.php`. Anda bisa me-mapping suatu URL ke sebuah `controller`, tapi hal tersebut tidak wajib.

Untuk sebuah aksi sederhana, Anda bisa langsung melakukannya di `Router`.

Untuk lebih jelasnya, mari kita buat sebuah halaman selamat datang dengan URL [http://localhost/laravel /public/halo] yang akan menampilkan tulisan “Halo bro”.

Buka file `app/routes.php`, lalu tambahkan kode untuk routing:

```
Route::get('/halo', function()
{
    return "Halo, bro";
});
```

Selesai, buka browser Anda dan ketikkan URL [http://localhost/laravel/public/halo], Anda sudah berhasil membuat halaman web pertama dengan laravel.

Pada prinsipnya website hanyalah masalah **request-response**, Anda minta URL apa, maka server akan memberikan respon yang bersesuaian. Seperti itulah tugas `router`, **memetakan URL yang diminta ke**

bagian kode tertentu.

Controller

Dalam konsep MVC biasanya sebuah URL dipetakan ke sebuah `controller`, dan karena konsep MVC sudah mendarah daging di kalangan web programmer, maka kita akan membuat satu halaman lagi dengan url /**halo-juga**, outputnya mirip, tapi kali ini menggunakan `controller`.

Buka kembali file `app/routes.php`, lalu tambahkan `router` baru:

```
Route::get('/halo-juga',
'SiteController@haloJuga');
```

Penjelasan dari kode di atas, jika ada yang meminta url /**halo-juga**, maka laravel akan mengeksekusi fungsi `haloJuga()` di dalam `SiteController`. Nah, karena SiteController-nya belum ada, maka langkah selanjutnya adalah membuat `controller`.

Tambahkan file baru `app/controllers`

/SiteController.php:

```
class SiteController extends
BaseController {
    public function haloJuga()
    {
        return 'halo juga, bro';
    }
}
```

Buka [http://localhost/laravel/public/halo-juga] dan lihat hasilnya. Selamat, Anda berhasil membuat halaman kedua dengan laravel, kali ini memanfaatkan `controller`.

View

Pada contoh di atas, kita cuma menampilkan string sederhana ke browser. Ngomong-ngomong soal browser, agar kelihatan valid tentunya string tersebut harus dibungkus dengan tag html yang lengkap, seperti berikut ini:

```
<html>
```

```
    <head>
        <title>d</title>
    </head>
    <body>
        halo juga, bro
    </body>
</html>
```

Tugas Anda selanjutya adalah memodifkasi contoh yang sudah diberikan, sehingga bisa menghasilkan tag html yang lengkap.

Hehe, apakah Anda akan memodifikasi kodenya menjadi seperti ini:

```
public function haloJuga()
{
return '<html><head><title></title>
</head><body>halo juga, bro</body>
</html>';
}
```

Memformat tampilan langsung dari `controller` jelas tidak menyenangkan, dan memang tidak dianjurkan.

Tampilan adalah tugas front-end developer, dan front-end developer tidak suka `controller`, mereka cuma suka `view`. Untuk menghindari kebencian mereka, mari kita buatkan view-nya.

Modifikasi kembali fungsi `haloJuga()` seperti berikut ini:

```
public function haloJuga()
{
        return View::make('halo_juga');
}
```

Fungsi `View::make('nama_file')` akan memanggil file view terpisah sesuai nama yang diberikan. Selanjutnya kita buat file baru `app/views/halo_juga.php`:

```
<html>
    <head>
        <title></title>
    </head>
    <body>
        halo juga, bro
```

```
    </body>
</html>
```

Jika Anda ingin menerapkan prinsip satu folder view untuk satu controller, yang berarti file `app/view/halo_juga.php` dipindahkan ke `app/views/site/halo_juga.php`, maka kodenya tinggal diedit sedikit menjadi `View::make('site.halo_juga')`. Tanda titik (dot) digunakan sebagai separator folder.

Refresh kembali browser Anda, tetap tidak ada bedanya kan? Hehe, jangan dilihat outputnya, tapi dilihat sourcenya. Sekarang website kita sudah dibungkus tag html yang lengkap. Satu lagi, front-end developer semakin sayang dengan kita ^_^.

Kita sudah mengenal fungsi router, dan sudah mempelajari unsur **V(view)** dan **C(controller)** dari konsep MVC. Pada tulisan berikutnya akan dibahas tentang unsur ketiga, yaitu **M(model)**. Tetap di saluran kesayangan Anda ini, follow twitter id_laravel untuk update tulisan, tip terbaru, dan tanya-tanya seputar laravel ;)