



Putu Alfred Crosby

Happy engineer. 😊

Feb 21 · 3 min read

Menambahkan CSS class active Pada Menu Sesuai Route yang di Akses di Laravel 5

Hal seperti menambahkan **class="active"** di menu navigasi di dalam sistem yang kita buat sudah menjadi hal yang wajib untuk di aplikasikan karena akan sangat menambah UX pada sistem kita. Ini pertama kalinya saya menggunakan Laravel, saya menggunakannya ketika Laravel masih versi 5.1.*. Dan ketika artikel ini ditulis Laravel sudah versi 5.2.*. Di dalam sistem yang saya buat saya menggunakan template admin gratis yaitu AdminLTE. Dan menu navigasi di template tersebut menyediakan class active untuk menu yang sedang aktif.

Disini permasalahan dimulai, saya ingin mengintegrasikan class active tersebut ketika Route yang bersangkutan sedang diakses (sedang aktif). Setelah berselancar di dunia maya, akhirnya saya menemukan cara yang **bagi saya** DRY(Don't Repeat Yourself) dan tidak WET(Write Everything Twice), yaitu menggunakan custom Helper pada Laravel.

Langsung saja kita mulai bagaimana langkah-langkahnya untuk menambahkan CSS class active pada menu di Laravel 5.

Langkah Pertama

Membuat file baru, yaitu file helper yang kita sudah bahas tadi.

- Buat folder baru di dalam folder **/app**, kita buat saja namanya **Helpers**.
- Masuk ke dalam folder Helpers, buat file baru bernama **Helper.php**.

```
2
3  /**
4   * Set any link as active by adding active class.
5   * @param [uri] $uri    Current URI.
6   * @param string $output CSS class name.
7   */
8  function set_active($uri, $output = 'active')
9  {
10     if( is_array($uri) ) {
11         foreach ($uri as $u) {
12             if (Route::is($u)) {
13                 return $output;
14             }
15         }
16     } else {
17         if (Route::is($uri)){
18             return $output;
19         }
20     }
21 }
```

Setelah langkah diatas selesai lanjutkan dengan paste code ini ke dalam file **Helper.php**

```
function set_active($uri, $output = 'active')
{
    if( is_array($uri) ) {
        foreach ($uri as $u) {
            if (Route::is($u)) {
                return $output;
            }
        }
    } else {
        if (Route::is($uri)){
            return $output;
        }
    }
}
```

Sedikit penjelasan mengenai fungsi diatas. Ada 2 parameter yang di set untuk fungsi **set_active(\$uri, \$output)**. Pada parameter pertama, kita membutuhkan URI yang sedang aktif(atau bisa nama route dan saya lebih menyarankan menggunakan nama route), yaitu kita harus memasukkannya secara manual. Jadi contohnya seperti ini **set_active('/home')** atau dalam bentuk Array **set_active(['/home', '/about'])**. Atau jika menggunakan nama route akan menjadi seperti ini **set_active('home.index')** dan jika menggunakan Array menjadi seperti ini **set_active(['home.index', 'about.index'])**.

Dan juga kita menggunakan Facade `Route::is()` untuk mengetahui uri yang sedang diakses(sedang aktif). Dan jika route yang diakses sesuai dengan route yang kita set sebagai parameter maka akan mengembalikan nilai **true**.

Pada parameter kedua yaitu nama CSS class yang ingin kita tambahkan, secara default saya tulis **“active”** tapi bisa diubah sesuai dengan kebutuhan anda.

Kita lanjutkan ke langkah kedua yaitu bagaimana cara menggunakan helper yang kita buat tadi.

Langkah Kedua

Cara termudah dan akan saya gunakan terus adalah dengan mendaftarkan Custom Helper yang kita buat tadi ke file `composer.json`. Ok, berikut langkah-langkahnya:

1. Buka file `composer.json` dan tambahkan kode ini pada blok `autoload`.

```
"files": [
    "app/Helpers/Helper.php"
]
```

Jadi file `composer` akan terlihat seperti ini:

```
"autoload": {
    "classmap": [
        "database"
    ],
    "psr-4": {
        "App\\": "app/"
    },
    "files": [
        "app/Helpers/Helper.php"
    ]
},
```

2. Setelah itu, buka Terminal kita jalankan perintah compser

```
composer dump-autoload
```

Tadaaaaa... Selesai, file Custom Helper kita otomatis akan di-load. Kalau kurang yakin coba kita buktikan dengan membuka file autoload_files.php di folder **/vendor/composer/autoload_files.php**

```
1  <?php
2
3  // autoload_files.php @generated by Composer
4
5  $vendorDir = dirname(dirname(__FILE__));
6  $baseDir = dirname($vendorDir);
7
8  return array(
9      '0e6d7bf4a5811bfa5cf40c5ccd6fae6a' => $vendorDir . '/symfony/polyfill-mbstring/bootstrap.php',
10     '1d1b89d124cc9cb8219922c9d5569199' => $vendorDir . '/hamcrest/hamcrest-php/hamcrest/Hamcrest.php',
11     '667aeda72477189d0494fec327c3641' => $vendorDir . '/symfony/var-dumper/Resources/functions/dump.php',
12     'bd9634f2d41831496de0d3dfe4c94881' => $vendorDir . '/symfony/polyfill-php56/bootstrap.php',
13     '2c102faa651ef8ea5874edb585946bce' => $vendorDir . '/swiftmailer/swiftmailer/lib/swift_required.php',
14     '3919eeb97e98d4648304477f8ef734ba' => $vendorDir . '/phpseclib/phpseclib/phpseclib/Crypt/Random.php',
15     'e7223560d890eab89cda23685e711e2c' => $vendorDir . '/psy/psysh/src/Psy/functions.php',
16     '5255c38a0faeba867671b61dfda6d864' => $vendorDir . '/paragonie/random_compat/lib/random.php',
17     'a0edc8309cc5e1d60e3047b5df6b7052' => $vendorDir . '/guzzlehttp/psr7/src/functions_include.php',
18     'e23faeee409e941dc9b4c80386209c39' => $vendorDir . '/laracasts/flash/src/Laracasts/Flash/functions.php',
19     '4a1f389d6ce373bda9e57857d3b61c84' => $vendorDir . '/barryvdh/laravel-debugbar/src/helpers.php',
20     'f0906e6318348a765ffb6eb24e0d0938' => $vendorDir . '/laravel/framework/src/Illuminate/Foundation/helpers.php',
21     '58571171fd5812e6e447dce228f52f4d' => $vendorDir . '/laravel/framework/src/Illuminate/Support/helpers.php',
22     'f18cc91337d49233e5754e93f3ed9ec3' => $vendorDir . '/laravelcollective/html/src/helpers.php',
23     '8fde5feda9697fe0ee53a3938c839fb0' => $baseDir . '/app/Helpers/Helper.php',
24 );
25
```

Custom Helper yang kita buat berhasil di load oleh composer. Sip

Final. Langkah Terakhir

Saatnya kita gunakan custom helper yang kita buat tadi pada template Blade kita :D

Saya sudah sebutkan tadi saya menggunakan AdminLTE Template jadi pada HTML saya terlihat seperti ini.

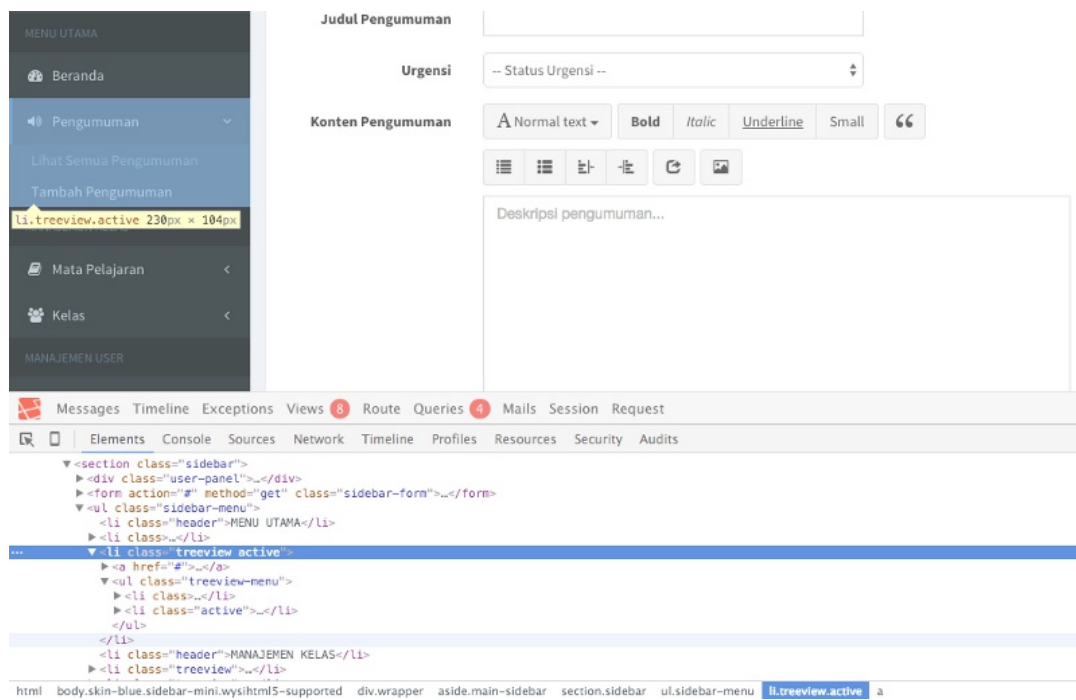
```
22  <ul class="sidebar-menu">
23  <li class="header">MENU UTAMA</li>
24  <li class="{ { set_active('auth.index') } }"><a href="{ { route('auth.index') } }"><i class="fa fa-dashboard"></i> <span>Beranda</span></a></li>
25  <li class="treeview { { set_active(['announcements.index', 'announcements.create', 'announcements.edit']) } }">
26  <a href="#"><i class="fa fa-volume-up"></i> <span>Pengumuman</span> <i class="fa fa-angle-left pull-right"></i></a>
27  <ul class="treeview-menu">
28  <li class="{ { set_active('announcements.index') } }"><a href="{ { route('announcements.index') } }">Lihat Semua Pengumuman</a></li>
29  <li class="{ { set_active('announcements.create') } }"><a href="{ { route('announcements.create') } }">Tambah Pengumuman</a></li>
30  </ul>
31  </li>
32
```

Mengaplikasikan custom helper pada Blade Template

Ok caranya cukup mudah bukan? Cukup memanggil **set_active('nama.route')** pada file template kita. Ok sip

Hasilnya menjadi seperti ini.

Tambahkan Pengumuman



Hasil akhir :')

Ok sekian yang bisa saya bagikan. Jangan lupa untuk klik like & share ya jika bermanfaat.

Juga tidak lupa saya berterimakasih kepada Mas [Rahmat Awaludin](#) yang sudah membantu saya dalam mendalami Laravel.



Buku [Menyelami Framework Laravel](#)

Mas Alfred merupakan pembaca dari buku [Menyelami Framework Laravel](#). Berisi 740+ halaman penjelasan komprehensif dan aplikatif menguasai framework **Laravel 5.2**. [Download sekarang!](#)

Laravel Bootstrap

4 2

Twitter Facebook



Putu Alfred Crosby

Happy engineer. 😊

Follow



Laravel Indonesia

Kumpulan Tutorial Laravel & Modern PHP Development

Follow