

Konfigurasi Proyek Laravel

Panduan konfigurasi ini dibuat berdasarkan Laravel versi 8. Untuk versi berikutnya, silahkan mencocokkan dengan dokumentasi resmi Laravel untuk memastikan fungsi-fungsi apa saja yang kompatibel/tidak kompatibel dengan versi 8. Contoh dokumentasi yang memuat informasi tersebut dapat dilihat di link berikut: <https://laravel.com/docs/8.x/upgrade>. File-file yang diperlukan untuk melakukan konfigurasi proyek Laravel tersedia pada *repository* berikut: <https://github.com/durianpeople/laravel-onion> – repository ini akan disebut sebagai **Template** pada bagian tulisan berikutnya

Konfigurasi ini memanfaatkan Service Provider yang disediakan oleh Laravel untuk meng-*extend* fungsionalitas *framework* untuk mengakomodasi struktur yang diinginkan. Berikut langkah-langkah yang dapat diikuti untuk melakukan konfigurasi awal:

1. *Initialize* proyek Laravel. Silahkan melihat dokumentasi berikut: <https://laravel.com/docs/8.x/installation>. Hasil dari tahap ini akan disebut sebagai **Proyek** pada bagian tulisan berikutnya
2. *Clone* Template yang telah disediakan pada direktori yang berbeda dengan langkah 1.
3. Pada subdirektori `app/Providers` pada Template, terdapat beberapa Provider yang digunakan untuk meng-*extend* fungsionalitas Laravel:
 - a. Pada `AppServiceProvider.php` di Template, salin kode berikut ke dalam method `boot` di `AppServiceProvider` pada Proyek:

```
// Register view namespaces
foreach (scandir($path = app_path('Modules')) as $moduleDir) {
    View::addNamespace($moduleDir, "{$path}/{$moduleDir}
/Presentation/views");
    Blade::componentNamespace("App\\Modules\\{$moduleDir}
\\Presentation\\Components", $moduleDir);
}
```

Bagian kode tersebut berfungsi untuk menambahkan *namespace* untuk komponen *view* dan *blade component* untuk setiap modul. *View* dan *blade component* masing-masing disimpan ke dalam `app/Modules/[NamaModul]/Presentation/views` dan `app/Modules/[NamaModul]/Presentation/Components`. Komponen *view* dapat diakses dengan menggunakan prefix `view('NamaModul::path.menuju.view')`. Komponen *blade component* dapat diakses dengan prefix `<x-NamaModul::path.menuju.component/>`. Dokumentasi tentang *view* dan *blade component* dapat dilihat di <https://laravel.com/docs/8.x/views#nested-view-directories> dan <https://laravel.com/docs/8.x/blade#manually-registering-components>.

- b. Salin `AssetPublishingService.php` dari Template menuju Proyek di direktori yang bersesuaian. Service provider ini berfungsi untuk memindahkan aset-aset yang diperlukan untuk *frontend* yang berada di `app/Modules/[NamaModul]/Presentation/assets` di Proyek menuju `public/assets/[nama_modul]` di Proyek menggunakan perintah `php artisan vendor:publish`. Daftarkan service provider ini seperti dijelaskan pada dokumentasi berikut: <https://laravel.com/docs/8.x/providers#registering-providers>
 - c. Salin `DependencyInjectionServiceProvider.php` dari Template menuju Proyek di direktori yang bersesuaian. Service provider ini berfungsi untuk menambahkan tempat khusus untuk mendefinisikan *binding* untuk *service injection*. Daftarkan service provider di Proyek. Setelah didaftarkan, definisi *binding* untuk *service injection* dapat dilakukan dalam file `app/Modules/[NamaModul]/dependencies.php` dengan variabel `$app` yang sudah tersedia dengan tipe `\Illuminate\Contracts\Foundation\Application` dalam file tersebut.
 - d. Salin `EventServiceProvider.php` dari Template menuju Proyek di direktori yang bersesuaian, kemudian daftarkan service provider ini di Proyek. Service provider ini berfungsi untuk memberitahu Laravel bahwa *event listener* dapat ditemukan di `app/Modules/[NamaModul]/Application/EventListener`. Dokumentasi tentang *event listener* dapat ditemukan di <https://laravel.com/docs/8.x/events#defining-listeners>.
 - e. Salin `MessagingServiceProvider.php` dari Template menuju Proyek di direktori yang bersesuaian, kemudian daftarkan service provider ini di Proyek. Service provider ini berfungsi untuk mendefinisikan *listener* untuk mekanisme *messaging* yang akan digunakan untuk komunikasi *asynchronous* tiap modul. Definisi *listener* ini akan diletakkan pada `app/Modules/[NamaModul]/messaging.php`.
 - f. Salin `ModuleTranslationServiceProvider` dari Template menuju Proyek di direktori yang bersesuaian, kemudian daftarkan service provider ini di Proyek. Service provider ini berfungsi untuk menambahkan fungsionalitas *localization* untuk setiap modul. Setiap modul akan memiliki direktori `app/Modules/[NamaModul]/Presentation/lang/[kode_bahasa]`. Dokumentasi tentang *localization* dapat ditemukan di <https://laravel.com/docs/8.x/localization>.
4. Pada subdirektori `app/Modules` di Template, terdapat beberapa folder, yaitu `Shared`. Salin kedua folder tersebut ke dalam Proyek dalam direktori yang bersesuaian. File-file yang berada di dalam `app/Modules/Shared/Mechanism` berisi kelas-kelas yang memiliki fungsi penting yang dapat digunakan dalam Proyek, namun tidak bisa dideklarasikan dalam Service Provider. Folder `app/Modules/Welcome` berisi contoh folder salah satu modul dengan susunannya.

5. Salin method `schedule` dan `commands` dari `app/Console/Kernel.php` di Template ke dalam `app/Console/Kernel.php` di Proyek. Method-method ini berfungsi untuk menambahkan *scheduling* yang dapat dijalankan melalui mekanisme *cron* Laravel pada `app/Modules/[NamaModul]/scheduling.php` di Proyek, dan untuk menambahkan CLI sebagai salah satu bagian dari Presentation yang dapat didefinisikan dalam `app/Modules/[NamaModul]/Presentation/Commands` di Proyek.
6. Salin method `report` dari `app/Exceptions/Handler.php` di Template ke dalam `app/Exceptions/Handler.php` di Proyek. Method ini bertugas untuk menghapus semua *event* tertunda yang disimpan oleh kelas `EventManager` saat terjadi *exception* yang tidak ditangkap di bagian lain.
7. Salin file `api.php` dan `web.php` dari folder `routes` di Template ke dalam folder `routes` di Proyek. Dua file ini berfungsi untuk menambahkan definisi *route* ke dalam folder `app/Modules/[NamaModul]/Presentation/routes` dengan nama file `api.php` dan `web.php` di Proyek.