



Andre Pratama

Sample eBook

Laravel Uncover

Panduan Belajar Framework Laravel 6

Duniailkom

Daftar Isi

Terima kasih.....	3
Asumsi / Pengetahuan Dasar.....	2
1. Berkenalan Dengan Laravel.....	3
1.1. Pengertian Framework.....	3
1.2. Kenapa Harus Laravel?.....	4
2. Mengakses Laravel.....	7
2.1. Folder Instalasi Laravel.....	7
3. Blade Template Engine.....	10
3.1. Pengertian Blade.....	10
3.2. Menampilkan Data.....	11
4. Eloquent ORM.....	13
4.1. Pengertian Eloquent ORM.....	13
5. Case Study: ILKOOM Profile Manager.....	15
5.1. Instalasi Laravel Authentication.....	17
Cara pemesanan eBook Laravel Uncover Duniaikom.....	19

Terima kasih

Terima kasih sudah mendownload sample buku Laravel Uncover Duniaikom. Materi yang tersedia diambil dari pembahasan eBook Laravel Uncover. Ini hanya sekedar pengenalan agar rekan-rekan bisa melihat bagaimana penulisan serta layout buku.

Pembelian buku ini jadi dukungan tak ternilai bagi saya untuk terus bisa berkarya. Kedepannya, saya (dan duniaikom) akan berusaha menghadirkan lebih banyak eBook serta tutorial programming lain.

eBook Laravel Uncover versi full bisa didapat dengan harga Rp 70.000 melalui transfer bank (tersedia Bank Mandiri, BCA, BRI, dan BNI), transfer e-payment Gopay / Ovo / Dana atau bisa melalui juga transfer pulsa (kartu AS / XL / Axis).

Untuk pemesanan atau tanya-tanya bisa menghubungi saya di duniaikom@gmail.com atau WA ke 083180285808 atau bisa ikuti panduannya ke [Cara Pembelian eBook Duniaikom](#).

Salam,

Andre Pratama
www.duniaikom.com

Asumsi / Pengetahuan Dasar

Laravel adalah framework PHP, sehingga saya berasumsi rekan-rekan sudah paham tentang **PHP** sebelum membaca buku ini. PHP yang dimaksud bukan hanya PHP dasar (*procedural*), tapi juga mencakup materi pemrograman object PHP (**OOP PHP**).

Tidak hanya itu, sepanjang pembahasan nanti kita juga banyak mengakses database MySQL, termasuk menulis query dan istilah-istilah lain terkait database. Sehingga saya menganggap juga sudah paham perintah untuk membuat tabel MySQL serta query dasar seperti **SELECT**, **INSERT**, **UPDATE** dan **DELETE**.

Untuk sisi design, dalam buku ini saya akan memakai **Bootstrap** agar tampilan program menjadi lebih sedap untuk dipandang. Ini tidak wajib, tapi agar bisa memahami seluruh materi, sebaiknya juga punya dasar Bootstrap terlebih dahulu. Dan, tentu saja agar bisa ke Bootstrap harus dari **HTML** dan **CSS** terlebih dahulu.

Huff... syarat yang cukup banyak... Ini karena Laravel memang bukan ditujukan untuk pemula.

Materi-materi di atas juga tidak harus dipelajari dari buku Duniaikom, tapi bisa juga dari tutorial/buku lain. Namun yang paling ideal adalah dari 7 buku duniaikom berikut:

1. **HTML Uncover**
2. **CSS Uncover**
3. **PHP Uncover**
4. **MySQL Uncover**
5. **JavaScript Uncover**
6. **Bootstrap Uncover**
7. **OOP PHP Uncover**

1. Berkenalan Dengan Laravel

Dalam bab pertama buku **Laravel Uncover** ini kita akan berkenalan dengan Laravel, sejarah singkat Laravel, membahas Laravel sebagai PHP Framework, serta perbandingan Laravel dengan framework lain.

1.1. Pengertian Framework

Laravel adalah **sebuah framework PHP**. Untuk bisa memahami ini maka kita harus membahas terlebih dahulu apa yang dimaksud dengan *framework*.

Secara sederhana, **framework** adalah kumpulan kode program siap pakai dengan aturan penulisan tertentu yang bertujuan untuk memudahkan serta mempercepat pembuatan aplikasi. Lebih spesifik lagi, PHP framework adalah framework yang dibuat menggunakan bahasa pemrograman PHP.

Tujuan utama kenapa menggunakan framework adalah untuk **mempercepat pembuatan aplikasi**, karena di dalam framework sudah tersedia berbagai fitur siap pakai. Kita tinggal menggunakan fitur ini tanpa perlu membuat semuanya dari nol. Selain itu aturan penulisan di framework akan memaksa kita menggunakan cara penulisan yang baik (mengikuti standar *best practice*).

Framework pada awalnya berasal dari kebutuhan programmer untuk mengurangi pembuatan kode yang sama berulang kali.

Sebagai contoh, misalkan saya bekerja sebagai programmer di sebuah perusahaan software. Project pertama disuruh membuat sistem informasi sekolah, dimana perlu fitur pendaftaran mahasiswa, pendaftaran dosen, login mahasiswa dan login dosen.

Setelah selesai, lanjut ke project kedua untuk membuat aplikasi perpustakaan yang diantaranya perlu halaman pendaftaran karyawan serta login karyawan. Project ketiga berupa website berita yang juga perlu fitur pendaftaran editor dan login editor.

Sampai di sini bisa dilihat bahwa untuk ketika project, saya perlu membuat fitur pendaftaran dan login. Dan besar kemungkinan untuk project keempat dan seterusnya juga butuh fitur serupa.

Daripada membuat dari nol terus menerus, akan lebih efisien jika saya menyiapkan sebuah kode dasar untuk halaman pendaftaran dan login. Jika project baru butuh fitur yang sama,

tinggal copy kode ini dan edit sedikit di bagian tertentu. Pekerjaan selesai dengan jauh lebih cepat.

Seiring bertambahnya pengalaman, kode dasar ini saya modifikasi dengan tambahan berbagai fitur baru agar semakin lengkap, misalnya menambah cara koneksi ke database, penanganan cookie, validasi form, dst. Akhirnya, jadilah sebuah **framework**.

Beberapa waktu kemudian, saya merasa framework ini sangat memudahkan dan ingin berbagi dengan programmer lain. Selain berbuat baik (karena banyak yang akan terbantu), saya berharap programmer lain juga bisa memberikan koreksi serta menambah fitur-fitur baru.

Jadi, sebenarnya siapa saja bisa membuat framework (tentunya selama memiliki skill), namun di antara sekian banyak framework, ada yang lebih populer dari yang lain. Alasannya bisa jadi karena fitur yang disediakan lebih banyak, penggunaannya lebih mudah, serta memiliki komunitas yang aktif.

Jika framework itu dibuat dengan struktur yang rapi dan mudah digunakan, maka secara perlahan makin banyak programmer lain ikut berkontribusi. Mayoritas framework juga bagian dari *open source project* sehingga bisa kita pakai dengan gratis.

Terdapat berbagai framework untuk keperluan yang berbeda-beda. Sebagai contoh, **Bootstrap** adalah sebuah framework CSS yang berisi kumpulan kode CSS untuk mempercepat pembuatan design web. Selain Bootstrap masih banyak framework CSS lain seperti **Materialize**, **Zurb Foundation**, **Bulma**, dan **Semantic UI**.

Begitu juga di PHP, terdapat berbagai pilihan framework seperti **Code Igniter**, **Symfony**, **Yii**, **Zend** dan tentu saja **Laravel**.

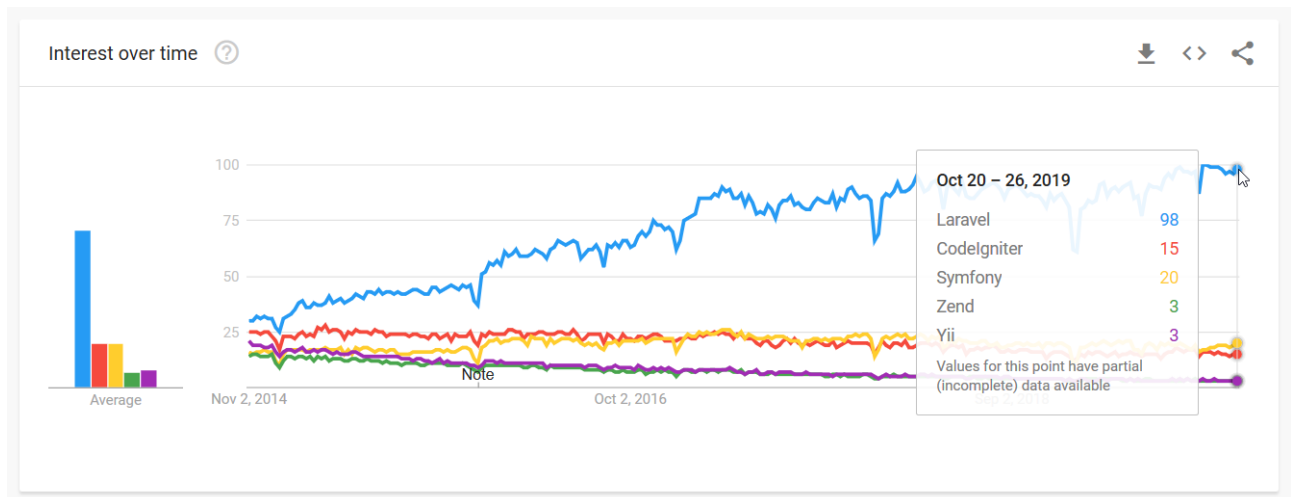
Dengan menggunakan framework, ibaratnya kita memanfaatkan keahlian ribuan programmer yang sudah lebih dahulu memikirkan apa yang harus dibuat dan bagaimana cara terbaik untuk membuatnya.

1.2. Kenapa Harus Laravel?

Telah kita singgung bahwa penggunaan framework bisa mempercepat pembuatan sebuah aplikasi. Namun di luar sana tersedia cukup banyak pilihan framework PHP. Jadi kenapa harus menggunakan **Laravel**?

Bagi saya pribadi, jawaban singkatnya adalah karena Laravel merupakan **framework PHP paling populer** saat ini.

Berikut hasil Google Trends untuk 5 framework PHP dalam 5 tahun terakhir (2014 - 2019):



Gambar: Perbandingan hasil Google Trends 5 framework PHP (sumber: trends.google.com)

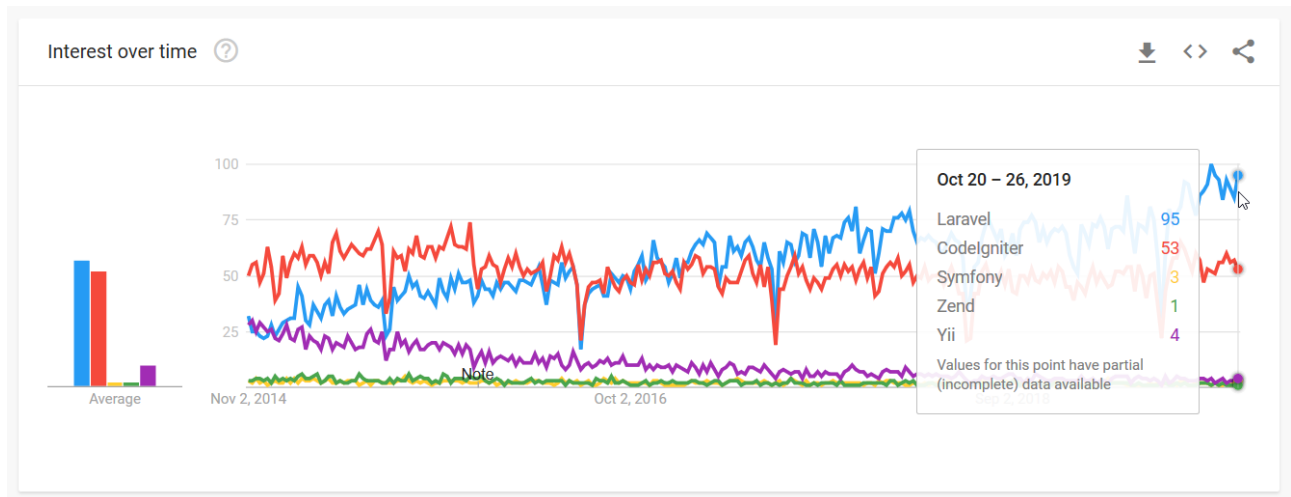
Google Trends menggambarkan apa yang paling banyak di search dari kotak pencarian Google. Terlihat bahwa Laravel sangat mendominasi jika dibandingkan dengan 4 framework PHP lain, yakni **CodeIgniter**, **Symfony**, **Zend** dan **Yii**.

Meskipun tidak bisa dibuktikan secara langsung, tapi semakin populer sebuah teknologi maka bisa dikatakan teknologi tersebut juga memiliki fitur yang relatif baik.

Hasil Google Trends ini bisa saja berasal dari blunder atau hal negatif yang membuat orang banyak mencari keyword tersebut, tapi kasus seperti itu biasanya hanya berefek singkat, tidak dalam waktu tahunan seperti grafik di atas.

Kepopuleran Laravel juga bisa dilihat dari kebutuhan industri. Jika anda mencari lowongan kerja programmer back-end, maka besar kemungkinan lowongan tersebut mensyaratkan harus menguasai PHP hingga framework Laravel. Ini juga salah satu alasan untuk belajar Laravel.

Sebagai info tambahan, grafik di atas adalah tingkat popularitas framework PHP di seluruh dunia. Bagaimana dengan di Indonesia? Hasilnya jadi lebih menarik:



Gambar: Perbandingan hasil Google Trends 5 framework PHP di Indonesia (sumber: trends.google.com)

Terlihat 2 garis yang saling mengejar, yakni garis biru dan garis merah. Artinya, di Indonesia terdapat 2 buah framework PHP yang sama-sama populer: **Laravel** dan **CodeIgniter**. Meskipun di luar negeri Laravel sangat mendominasi, tapi di Indonesia CodeIgniter masih jadi salah satu pilihan framework PHP.

Jika anda memutuskan fokus ke bidang PHP back-end, maka bisa coba pelajari 2 framework ini: **Laravel** dan **CodeIgniter**. Lowongan kerja untuk keduanya juga sama besar (setidaknya menurut pengamatan saya). Ada perusahaan yang menggunakan CI, banyak juga yang memakai Laravel.

Bagaimana dengan framework lain seperti **Symfony**, **Yii**, atau **Zend**? Kita tidak bisa mengatakan kalau framework ini tidak bagus karena pada dasarnya setiap framework punya kelebihan dan kekurangan masing-masing. Hanya kebetulan saja framework-framework tersebut kalah populer dari Laravel dan CodeIgniter. Ini sedikit miris karena di dalam Laravel banyak menggunakan komponen yang berasal dari framework Symfony.

Tujuan akhir dari penggunaan framework adalah agar pekerjaan cepat selesai, tidak masalah ingin dibuat dengan framework apa saja. Beberapa perusahaan software juga membuat framework khusus yang tidak di-publish.

Web perusahaan besar seperti **Tokopedia** atau **Bukalapak** sangat mungkin menggunakan framework sendiri, karena mereka punya kebutuhan khusus yang bisa jadi belum tersedia di Laravel atau Code Igniter.

Meskipun nantinya kita bisa saja tidak menggunakan framework, namun pemahaman alur kerja framework tetap bermanfaat. Framework modern dirancang untuk skala industri dan mengikuti standar terbaik dalam pembuatan aplikasi. Kita bisa belajar banyak dalam mengelola kode program dengan tingkat kerumitan menengah ke atas.

2. Mengakses Laravel

Dalam bab ini kita akan membahas cara mengakses Laravel. Laravel memiliki cara tersendiri agar bisa dijalankan.

2.1. Folder Instalasi Laravel

Dalam bab sebelumnya kita telah berhasil menginstall Laravel. Jika anda mengikuti semua langkah yang ada, maka di htdocs terdapat 4 folder: coba1, coba2, coba3 dan coba4.

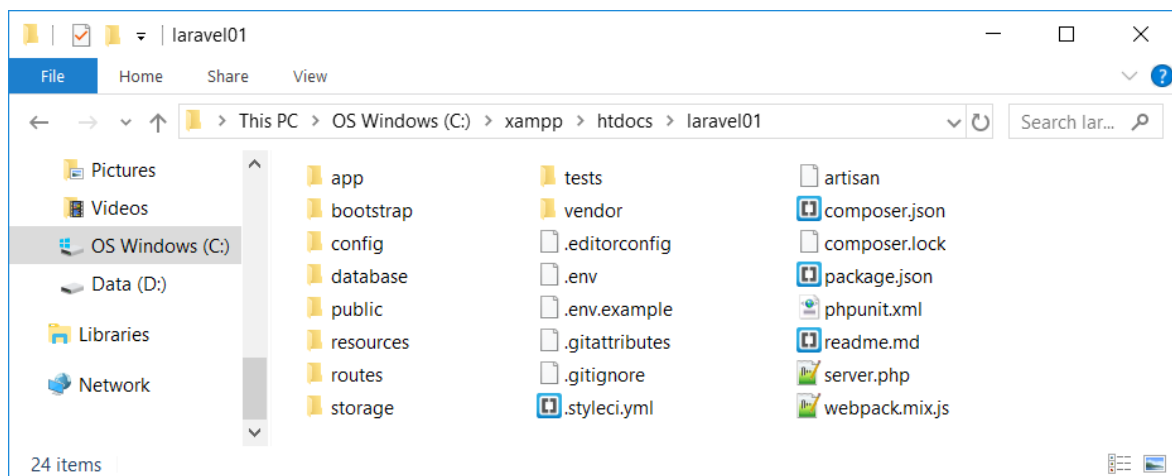
Pilih salah satu folder yang berisi Laravel 6 lalu ubah nama foldernya menjadi **laravel01**. Perubahan nama folder ini tidak wajib, hanya agar lebih rapi saja. Atau anda juga bisa jalankan perintah berikut untuk menginstall Laravel 6 ke dalam folder laravel01:

```
composer create-project --prefer-dist laravel/laravel="^6.0" laravel01
```

Sepanjang buku ini, saya akan terus memakai folder dengan nama "**laravel01**". Anda bisa saja menggunakan nama lain, karena itu tidak pengaruh ke kode program. Yang penting, di dalam folder tersebut terinstall **Laravel 6**.

Selain itu di beberapa awal bab saya akan menginstall ulang Laravel ke dalam folder ini. Tujuannya agar kita bisa fokus ke materi yang dibahas, tidak terganggu proses "utak-atik" dari bab sebelumnya.

Baik, mari lihat apa saja isi folder Laravel ini:



Gambar: Isi folder Laravel

Untuk Laravel 6.4 yang saya gunakan, total terdapat **9.368 file** di dalam **1.727 folder**. Yup, terbilang luar biasa banyak, dan itulah alasan kenapa ukuran file installer Laravel mencapai lebih dari 40MB.

Namun dari gambar terlihat hanya ada 10 folder serta beberapa file. Mayoritas file Laravel memang tersembunyi karena tidak perlu kita utak-atik. Sepanjang pembuatan project nanti, kita hanya butuh mengubah kurang dari 10 file bawaan Laravel (tidak termasuk file yang akan di tambah).

Berikut penjelasan singkat dari isi setiap folder:

- **App:** berisi berbagai file untuk aplikasi yang akan di bangun. Dalam folder inilah nantinya kita membuat controller dan model.
- **Bootstrap:** berisi file `app.php` yang berfungsi sebagai *bootstrap* atau file pengaturan awal dari Laravel. Selain itu terdapat juga folder `cache` untuk meningkatkan performa aplikasi.

Sebagai info tambahan, nama folder ini tidak ada kaitannya dengan framework CSS **Bootstrap**. Dalam istilah komputer, *bootstrap* memiliki makna yang sama dengan *booting* yakni proses awal dari sebuah aplikasi.

- **Config:** berisi berbagai file konfigurasi Laravel. Folder ini cukup sering kita akses karena di sinilah berbagai pengaturan Laravel tersimpan.
- **Database:** berisi folder dan file yang 'mengurusi' database seperti `migrations`, `factories` dan `seeds`. Folder database ini juga bisa dipakai sebagai tempat menyimpan file tabel database SQLite (jika anda menggunakan SQLite).
- **Public:** berisi file `index.php` sebagai file awal dari semua request ke aplikasi Laravel. File `index.php` inilah yang nanti memanggil berbagai file-file lain. Namun kita tidak akan meng-edit file `index.php` ini secara langsung.
- **Resource:** berisi file *resource* 'mentah' seperti file CSS dan JavaScript. Di sini juga nantinya kita membuat view.
- **Routes:** berisi file yang akan menangani proses routing.
- **Storage:** berisi tempat penyimpanan file yang di generate oleh Laraval, seperti file log. Jika kita membuat form upload, file tersebut juga bisa disimpan di sini.
- **Test:** berisi file untuk proses testing seperti PHPUnit.
- **Vendor:** Berisi berbagai file internal framework Laravel. Di sinilah semua library *dependency* Laravel berada. Folder vendor ini mencakup 90% ukuran framework laravel (33MB yang terdiri lebih dari 8000an file). Meskipun berisi banyak file, folder ini tidak perlu kita utak-atik.

Sepanjang pembuatan aplikasi, folder yang akan sering kita akses adalah **app** (untuk membuat

model dan controller), **resource** (untuk membuat view) dan **routes** (untuk membuat routing). Penjelasan lebih lengkap akan kita bahas ketika masuk ke setiap materi tersebut.

<Ini merupakan potongan awal Bab tentang cara Mengakses Laravel>

3. Blade Template Engine

Bab kali ini masih berhubungan dengan view, dimana kita akan membahas **blade**, sebuah template engine bawaan Laravel. Blade sangat powerful dan akan memudahkan dalam memproses tampilan front-end Laravel.

Agar seragam dan menghindari error akibat praktek dari bab sebelumnya, saya akan mulai dari installer baru Laravel 6. Anda bisa hapus isi folder **laravel01**, atau menginstall Laravel ke folder baru, misalnya **laravel02**.

Berikut perintah untuk menginstall Laravel ke folder **laravel01**:

```
composer create-project --prefer-dist laravel/laravel="^6.0" laravel01
```

Alternatif lain, jika anda memiliki file asli Laravel 6 bisa juga di copy atau di rename menjadi folder **laravel01**.

Setelah itu input juga file Bootstrap (folder **css** dan folder **js**) ke dalam folder **public**, karena kita akan banyak menggunakan Bootstrap dalam bab ini.

3.1. Pengertian Blade

Blade adalah *template engine* bawaan Laravel. Secara sederhana, template engine berisi perintah tambahan untuk mempermudah pembuatan template. Template sendiri bisa disebut sebagai *kerangka dasar tampilan*.

Secara garis besar, terdapat 2 fungsi utama blade di dalam Laravel:

1. Mempersingkat penulisan perintah PHP.
2. Pemecahan file template (proses pembuatan layout).

Sebagai contoh, untuk menampilkan variabel di dalam view sebelumnya kita menggunakan perintah berikut:

```
<?php echo $nama; ?>
```

Dengan blade, bisa ditulis menjadi:

```
{{ $nama }}
```

Terlihat kodenya menjadi lebih rapi dan lebih singkat. Selain itu masih banyak fitur dari blade

yang akan kita pelajari secara bertahap.

3.2. Menampilkan Data

Kegunaan paling dasar dari **blade** adalah untuk mempermudah proses menampilkan data di dalam view.

Seperti yang sudah kita praktekan sebelumnya, data ini bisa berasal dari route atau sumber lain (seperti controller). Ketika sampai ke view, data tersebut bisa ditampilkan dengan perintah `echo PHP`.

Mari lihat kembali contoh prakteknya, silahkan buat route dengan kode berikut:

routes/web.php

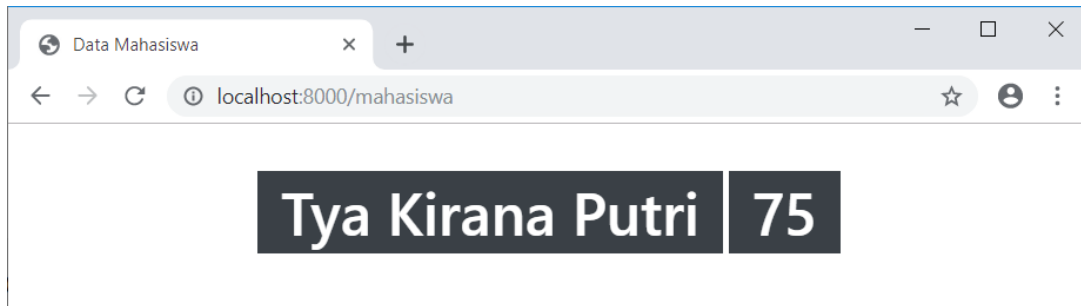
```
1 Route::get('/mahasiswa', function () {
2     $nama = 'Tya Kirana Putri';
3     $nilai = 75;
4     return view('mahasiswa', compact('nama', 'nilai'));
5 });
```

Saya yakin anda sudah bisa memahami route ini, dimana ketika alamat `http://localhost:8000/mahasiswa` diakses, view 'mahasiswa' akan dijalankan serta mengirim data `$nama` dan `$nilai` ke dalam view tersebut.

Berikut kode program untuk file `mahasiswa.blade.php`:

resources/views/mahasiswa.blade.php

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <meta http-equiv="X-UA-Compatible" content="ie=edge">
7     <link href="/css/bootstrap.min.css" rel="stylesheet">
8     <title>Data Mahasiswa</title>
9 </head>
10 <body>
11     <div class="container text-center mt-3 pt-3 bg-white">
12         <h1 class="bg-dark px-3 py-1 text-white d-inline-block">
13             <?php echo $nama ?>
14         </h1>
15         <h1 class="bg-dark px-3 py-1 text-white d-inline-block">
16             <?php echo $nilai ?>
17         </h1>
18     </div>
19 </body>
20 </html>
```



Gambar: Tampilan view mahasiswa

Dalam view ini terdapat pemanggilan file CSS external `/css/bootstrap.min.css` (baris 7). Terkait file Bootstrap ini sudah kita bahas di akhir bab sebelumnya.

Sepanjang sisa buku saya juga akan terus menggunakan Bootstrap. Selain agar tampilan web tidak monoton, ini juga sebagai latihan penerapan Bootstrap terutama bagi yang memutuskan belajar **full-stack web development** (paham *front-end* dan *back-end*).

Ini semata-mata hanya untuk mempercantik tampilan web. Jika anda belum mempelajari Bootstrap, boleh diabaikan saja tambahan class CSS yang ada.

Fokus utama kita ada di baris 13 dan 16, yakni perintah `echo` PHP untuk menampilkan variabel `$nama` dan `$nilai` yang berasal dari route. Tidak ada hal yang baru di sini.

Menggunakan blade, bagian `<body>` dari view bisa ditulis sebagai berikut:

`resources/views/mahasiswa.blade.php`

```
1 <div class="container text-center mt-3 py-3 bg-white">
2   <h1 class="bg-dark px-3 py-1 text-white d-inline-block">
3     {{$nama}}
4   </h1>
5   <h1 class="bg-dark px-3 py-1 text-white d-inline-block">
6     {{$nilai}}
7   </h1>
8 </div>
```

Hasilnya akan sama seperti sebelumnya. Penulisan seperti ini membuat kode program di dalam view menjadi lebih singkat dan lebih rapi. Secara internal, perintah `{{ $nama }}` akan diproses oleh Laravel sebagai `<?php echo $nama ?>`.

<Ini merupakan potongan awal Bab tentang Blade Template Engine>

4. Eloquent ORM

Pada bab kali ini kita akan membahas **Eloquent ORM**, yakni sebuah cara modern untuk mengakses database dari Laravel. Selain menyediakan fitur untuk proses CRUD biasa, eloquent juga memiliki fitur **Soft Delete**, dimana kita bisa mengembalikan data yang sudah terhapus.

4.1. Pengertian Eloquent ORM

Eloquent ORM adalah cara pengaksesan database dimana setiap baris tabel dianggap sebagai sebuah object. Kata **ORM** sendiri merupakan singkatan dari **Object-relational mapping**, yakni sebuah teknik programming untuk mengkonversi data ke dalam bentuk object.

Sebagaimana yang sudah kita pahami, database terdiri dari kumpulan tabel yang saling terhubung. Di dalam setiap tabel, data disimpan dalam bentuk baris dan kolom. ORM dipakai untuk mengubah baris dan kolom ini menjadi sebuah object. Nantinya, setiap kolom akan menjadi property dari object tersebut.

Sebagai perbandingan, berikut cara menginput data ke tabel mahasiswa menggunakan **query builder**:

```
1 public function insert(){
2     $result = DB::table('mahasiswas')->insert(
3         [
4             'nim' => '19003036',
5             'nama' => 'Sari Citra Lestari',
6             'tanggal_lahir' => '2001-12-31',
7             'ipk' => 3.5,
8             'created_at' => now(),
9             'updated_at' => now(),
10        ]
11    );
12 }
```

Dengan query builder, data yang akan diinput kita tulis dalam bentuk associative array (baris 4 – 9). Array ini merupakan argument dari method `insert()` yang diakses dari `DB::table('mahasiswas')`.

Jika menggunakan **Eloquent ORM**, data tabel ditulis sebagai property dari object **Mahasiswa**:

```
1 public function insert(){
2     $mahasiswa = new Mahasiswa;
```

```
3     $mahasiswa->nim = '19003036';
4     $mahasiswa->nama = 'Sari Citra Lestari';
5     $mahasiswa->tanggal_lahir = '2001-12-31';
6     $mahasiswa->ipk = 3.5;
7     $mahasiswa->save();
8 }
```

Di baris 2 terdapat perintah untuk proses instansiasi (pembuatan object) class **Mahasiswa** yang disimpan ke dalam variabel `$mahasiswa`. Kemudian di baris 3 – 6 saya mengisi beberapa property ke dalam object `$mahasiswa`. Nama property ini harus sesuai dengan nama kolom yang ada di tabel `mahasiswas`.

Terakhir, di baris 7 terdapat pemanggilan method `$mahasiswa->save()` yang akan menyimpan object `$mahasiswa` sebagai baris baru ke dalam tabel `mahasiswas`.

Inilah cara kerja dari ORM, yakni mengkonversi baris tabel menjadi sebuah **object**. Teknik penulisan seperti ini sangat cocok di pakai dalam framework seperti Laravel, yang semuanya sudah menerapkan konsep pemrograman berbasis object.

Bagi yang sudah terbiasa menggunakan perintah query SQL, ORM mungkin terasa sedikit membingungkan, karena cukup berbeda dengan cara yang kita pakai selama ini (menulis perintah query). Namun lambat laun anda akan terbiasa, selain itu ORM membuat kode program kita menjadi lebih rapi dan 'seragam' dengan fitur OOP lain di Laravel.

<Ini merupakan potongan awal Bab tentang Eloquent ORM>

5. Case Study: ILKOOM Profile Manager

Dalam bab ini saya ingin membuat case study yang sedikit agak kompleks. Kita akan merancang semacam *profile manager*, dimana setiap user bisa mendaftarkan diri dengan mengisi beberapa inputan form serta mengupload gambar profil. Semua user bisa melihat profil user lain, tapi hanya bisa mengedit data user itu sendiri (pembatasan hak akses).

Studi kasus ini merangkum hampir semua materi yang kita bahas sepanjang buku, mulai dari konsep MVC, *blade*, *Laravel mix*, *migration*, *eloquent*, *form processing*, *form validation*, *localization*, *file upload*, *middleware*, serta *policy*.

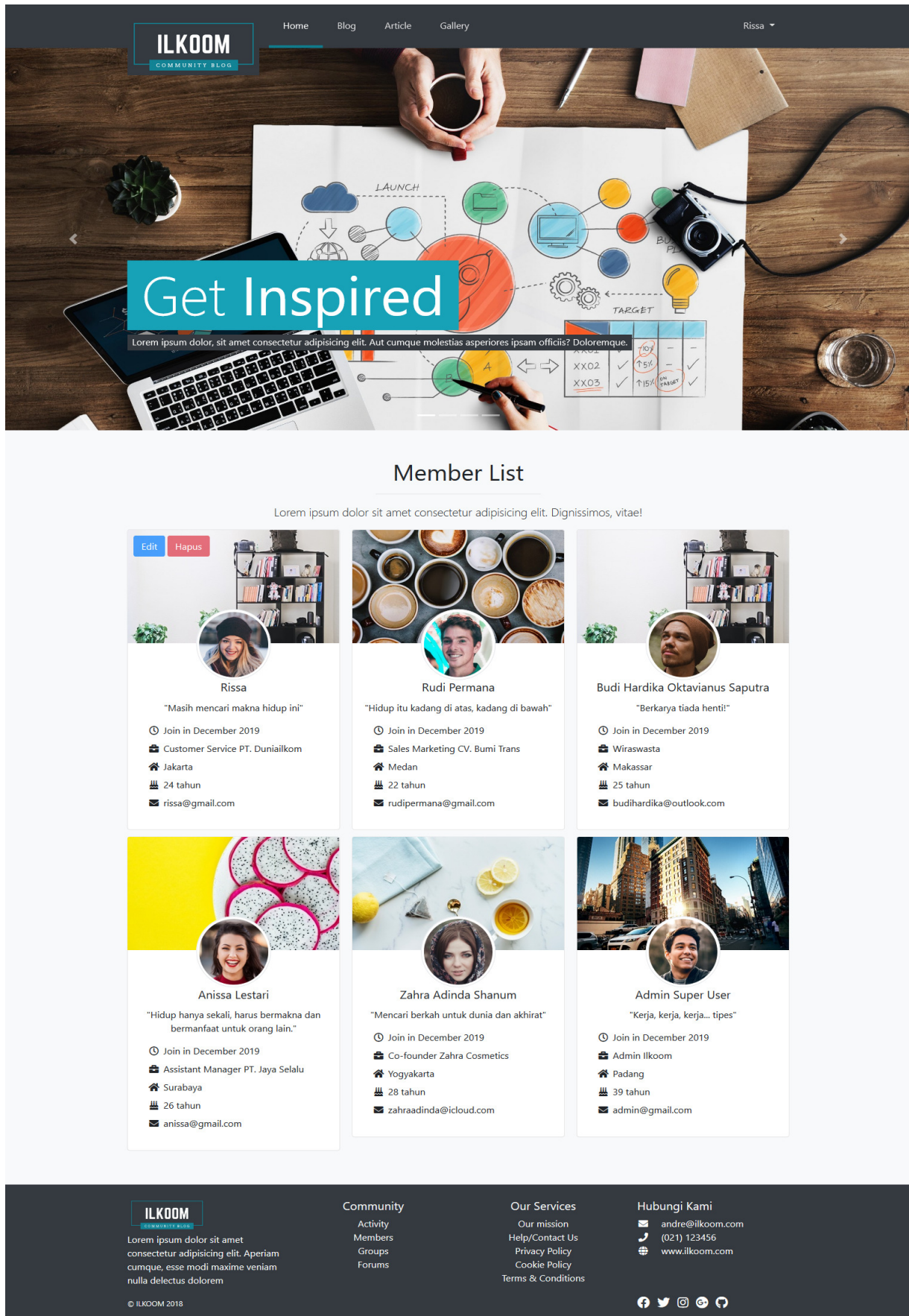
Agar seragam dan menghindari error akibat praktek dari bab sebelumnya, saya akan mulai dari installer baru Laravel 6. Anda bisa hapus isi folder **laravel01**, atau menginstall Laravel ke folder lain, misalnya **laravel02**.

Berikut perintah untuk menginstall Laravel ke folder `laravel01`:

```
composer create-project --prefer-dist laravel/laravel="^6.0" laravel01
```

Dalam bab ini kita juga akan menjalankan ulang proses migration, maka jika ada tabel di database `laravel`, silahkan hapus terlebih dahulu.

Berikut sedikit tampilan akhir dari studi kasus kita:



Gambar: Tampilan halaman utama ILKOOM Profile Manager

The screenshot displays the 'Edit Data' form for the ILKOOM Profile Manager. The form is structured as follows:

- Email ***: Input field containing 'rissa@gmail.com'.
- Nama ***: Input field containing 'Rissa Permata'.
- Tanggal Lahir ***: Date picker showing '19' for the day, 'Agustus' for the month, and '1995' for the year.
- Pekerjaan**: Input field containing 'Customer Service PT. Duniaillkom'.
- Kota**: Input field containing 'Jakarta'.
- Bio Profil**: Text area containing 'Masih mencari makna hidup ini'.
- Gambar Profil**: Input field showing 'rissa-1577106850.jpg' with a 'Browse' button.
- Background Profil**: A dropdown menu set to 'Gambar 7' and a grid of 12 image thumbnails numbered 1 to 12.
- Update**: A blue button at the bottom of the form.

The footer of the page includes the ILKOOM logo, a paragraph of placeholder text, and four columns of links: Community (Activity, Members, Groups, Forums), Our Services (Our mission, Help/Contact Us, Privacy Policy, Cookie Policy, Terms & Conditions), and Hubungi Kami (Email: andre@ilkoom.com, Phone: (021) 123456, Website: www.ilkoom.com). Social media icons for Facebook, Twitter, Instagram, and YouTube are also present.

Gambar: Tampilan halaman form register ILKOOM Profile Manager

Jika anda pernah membaca buku **Bootstrap Uncover** duniaillkom, sudah bisa langsung menebak kalau design tampilan yang saya pakai adalah studi kasus bab terakhir dari buku tersebut, yakni **ILKOOM Community Blog Template**.

Untuk menghemat tempat, saya tidak akan membahas kode-kode CSS dan Bootstrap yang dipakai, kita fokus ke kode program Laravel saja.

5.1. Instalasi Laravel Authentication

Setelah menginstall Laravel, proses pertama yang kita butuhkan adalah instalasi Laravel Authentication, yang tidak lain juga menginstall **Laravel UI** dan **npm** (node package manager).

Pertama, install **Laravel UI** dengan perintah:

```
composer require laravel/ui
```

Setelah itu install **Laravel Authentication**:

```
php artisan ui bootstrap --auth
```

Lanjutkan dengan proses install **npm** dan **compile assets**:

```
npm install && npm run dev
```

Setelah instalasi selesai, jalankan langsung **migration**:

```
php artisan migrate
```

Dan test buka halaman <http://localhost:8000/> di web browser:



Gambar: Halaman default Laravel + Authentication

Sip, fitur authentication sudah terinstall dan siap digunakan.

<Ini merupakan potongan awal bab Case Study: ILKOOM Profile Manager, yang merupakan materi terakhir di buku Laravel Uncover>

Cara pemesanan eBook Laravel Uncover Duniaikom

Demikian sample buku **Laravel Uncover Duniaikom**. Materi yang tersedia diambil dari pembahasan eBook Laravel Uncover. Ini hanya sekedar pengenalan agar rekan-rekan bisa melihat bagaimana penulisan serta layout buku.

eBook Laravel Uncover versi full bisa didapat dengan harga Rp 70.000 melalui transfer bank (tersedia Bank Mandiri, BCA, BRI, dan BNI), transfer e-payment Gopay / Ovo / Dana atau bisa melalui juga transfer pulsa (kartu AS / XL / Axis).

Untuk pemesanan atau tanya-tanya bisa menghubungi saya di duniaikom@gmail.com atau WA ke 083180285808 atau bisa ikuti panduannya ke [Cara Pembelian eBook Duniaikom](#).

Pembelian buku ini jadi dukungan tak ternilai bagi saya untuk terus bisa berkarya. Kedepannya, saya (dan duniaikom) akan berusaha menghadirkan lebih banyak eBook serta tutorial programming lainnya.

Sampai ketemu di buku...