

LAPORAN TUGAS PEMROGRAMAN WEB LANJUT
(Pengantar dan Instalasi Laravel)



Disusun Oleh :

Nama: Ahmad Bahrul Ulum

Nim: 1841720150

**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020**

Tunggulah hingga proses penginstalan selesai. Setelah itu, Kita bisa mulai membuat project laravel dengan mengetikkan perintah:

laravel new laravelapp

Keterangan:

laravelapp: merupakan folder kerja laravel, Kita dapat menggantinya sesuai dengan keinginan

2. Install Via Composer Create-Project

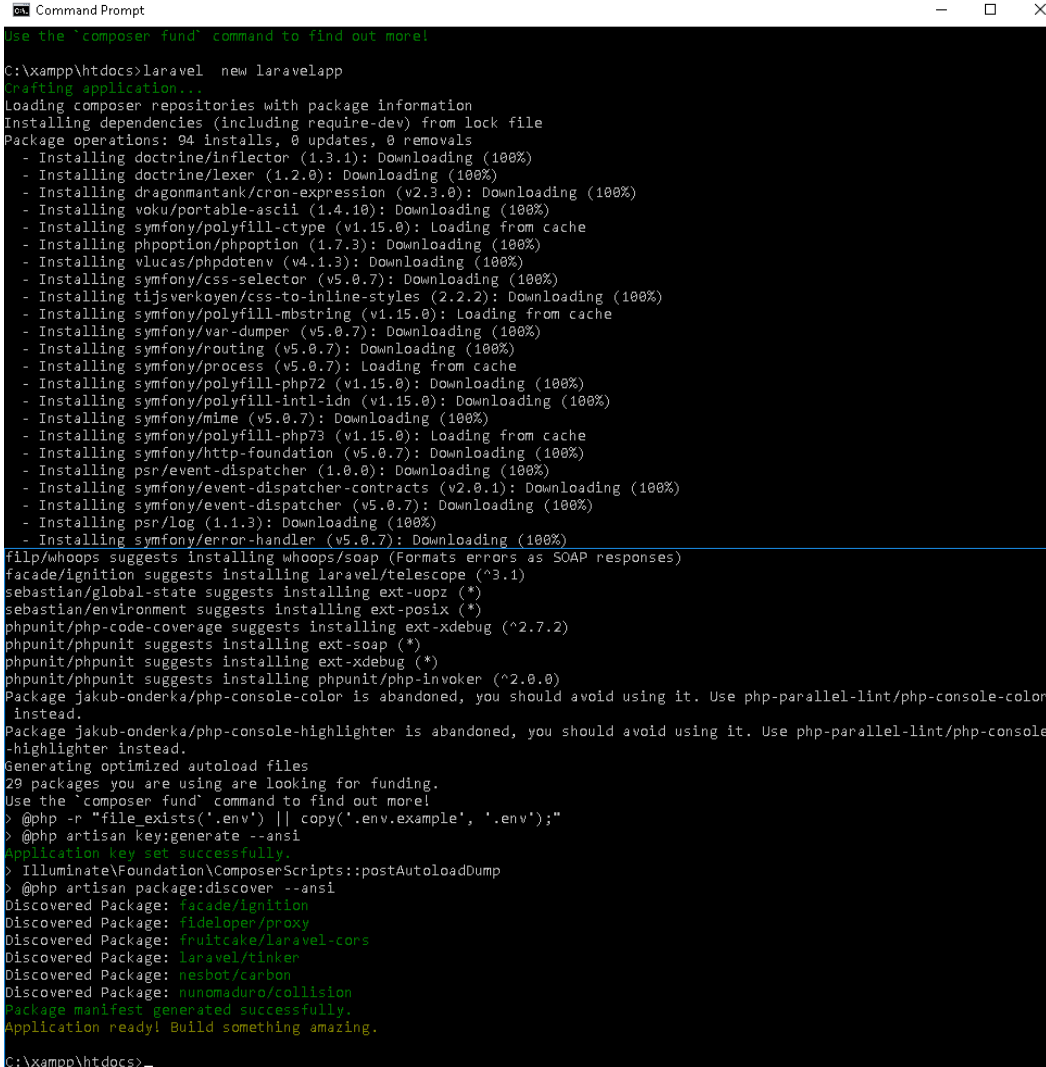
Selain menggunakan cara pertama, Kita dapat melakukan penginstalan laravel dengan mengetikkan perintah berikut pada command prompt:

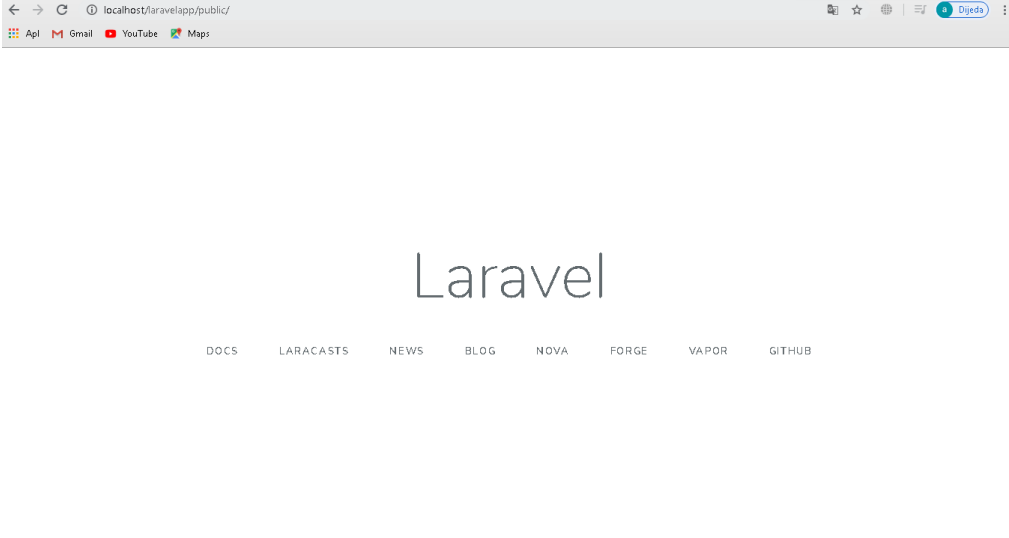
composer create-project --prefer-dist laravel/laravel laravelapp

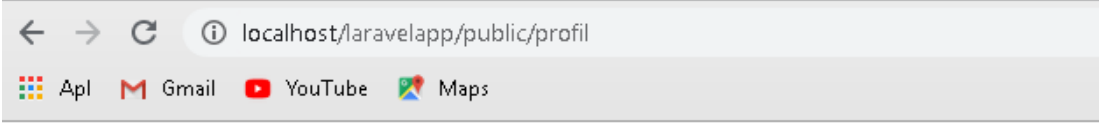
cd C:\xampp\htdocs

```
C:\xampp\htdocs>composer global require "laravel/installer"
Changed current directory to C:/Users/asus/AppData/Roaming/Composer
Using version ^3.0 for laravel/installer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 14 installs, 0 updates, 0 removals
- Installing symfony/process (v5.0.7): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.15.0): Downloading (100%)
- Installing symfony/filesystem (v5.0.7): Downloading (100%)
- Installing psr/container (1.0.0): Downloading (100%)
- Installing symfony/service-contracts (v2.0.1): Downloading (100%)
- Installing symfony/polyfill-php73 (v1.15.0): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
- Installing symfony/console (v5.0.7): Downloading (100%)
- Installing ralouphie/getallheaders (3.0.3): Downloading (100%)
- Installing psr/http-message (1.0.1): Downloading (100%)
- Installing guzzlehttp/psr7 (1.6.1): Downloading (100%)
- Installing guzzlehttp/promises (v1.3.1): Downloading (100%)
- Installing guzzlehttp/guzzle (6.5.2): Downloading (100%)
- Installing laravel/installer (v3.0.1): Downloading (100%)
symfony/service-contracts suggests installing symfony/service-implementation
symfony/console suggests installing symfony/event-dispatcher
symfony/console suggests installing symfony/lock
symfony/console suggests installing psr/log (For using the console logger)
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
Writing lock file
Generating autoload files
```

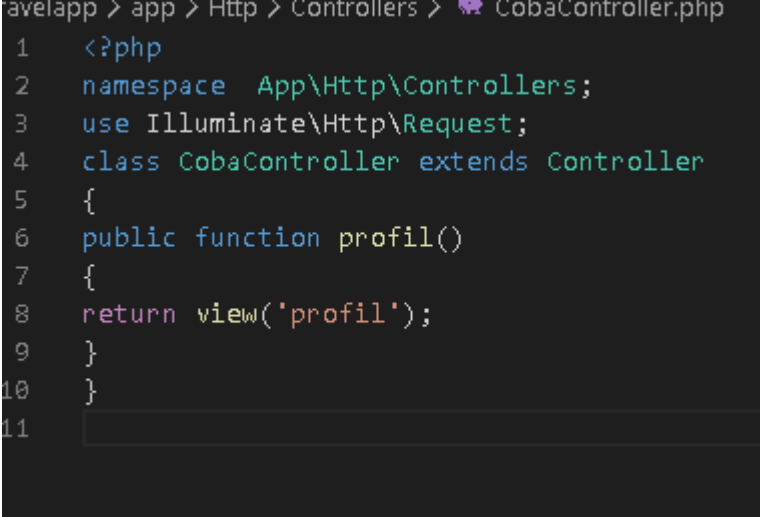
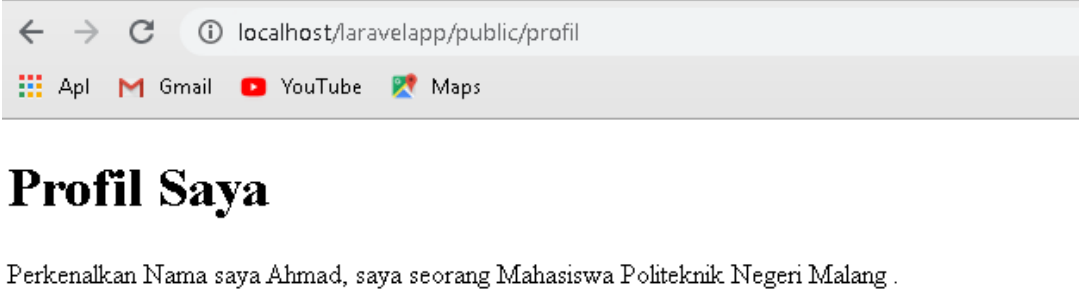
laravel new laravelapp

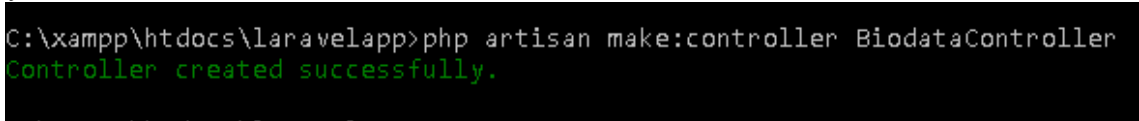
	
3	<p>Setelah proses instalasi Laravel selesai, Kita perlu menguji apakah hasil instalasi tersebut berjalan dengan baik atau tidak. Aktifkan Apache server lewat XAMPP Control Panel. Pada browser, ketik alamat http://localhost/laravelapp/public/</p> <p>Atau Kita dapat menggunakan perintah berikut di Command Prompt:</p> <p style="text-align: center;">php artisan serve</p> <p>Kita dapat menjalankan aplikasi LARAVEL di folder manapun dengan menggunakan perintah di atas sehingga tidak harus selalu berada di dalam folder htdocs (XAMPP).</p> <p>Jalankan dengan localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :</p>

	
	<p><u>Praktikum – Bagian 2: Mengenal dan membuat route</u></p>
<p>1</p>	<p>Setiap request yang datang pada laravel akan diarahkan melalui sebuah route. Route ini yang akan menentukan respon apa yang akan dikerjakan untuk membalas request tersebut. Halaman homepage default Laravel seperti yang ditampilkan pada gambar di Bagian 1 adalah hasil dari script yang terletak pada file laravelapp\routes\web.php Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut</p> <pre data-bbox="272 1045 1042 1251"> 15 16 Route::get('/', function () { 17 return view('welcome'); 18 }); 19 </pre>
<p>2</p>	<p>Kita dapat menuliskan perintah baru untuk membuat route, selain mengembalikan file view sebagai response, Kita dapat mengembalikan response berupa string atau teks biasa.</p> <p>Buka file laravelapp\routes\web.php, tambahkan isinya dengan script di bawah ini :</p> <p>Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; });</p> <pre data-bbox="285 1600 1390 1780"> Route::get('/', function () { return view('welcome'); }); Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; }); </pre> <p>Untuk melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</p>

	 <p>Halo, Anda sedang mencoba Routing di Laravel</p>
3	<p>Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:</p> <pre>Route::get('profil', function () { return view('profil'); });</pre> <p>Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:</p> <pre><!DOCTYPE html> <html> <head> <title>Profil Saya</title> </head> <body> <h1>Profil Saya</h1> <p>Perkenalkan Nama saya nama_saya, saya seorang </p> </body> </html></pre> <p>Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat http://localhost:8000/profil</p>  <p>Profil Saya</p> <p>Perkenalkan Nama saya Ahmad, saya seorang Mahasiswa Politeknik Negeri Malang .</p>

	<p><u>Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel</u></p>
1	<p>Pada pembahasan sebelumnya, Kita sudah bisa menampilkan file view langsung dari Routenya. Tetapi sebuah aplikasi tidaklah sesimpel itu, akan ada logika dan data-data yang harus diolah terlebih dahulu. Caranya adalah dengan menggunakan Controller, seperti yang telah dipelajari ketika menggunakan CodeIgniter.</p> <p>Ada dua cara yang dapat digunakan untuk membuat Controller. Cara pertama, dengan membuat file controller secara manual dan tuliskan code untuk extends controller secara manual. Cara kedua, adalah dengan membuat file Controller lewat Artisan di Laravel.</p> <p>Kita gunakan cara yang kedua, pada latihan kali ini kita akan membuat Controller dengan nama CobaController dengan menuliskan di command prompt / terminal :</p> <p style="text-align: center;">php artisan make:controller CobaController</p>  <p>Maka akan terbentuk sebuah file dengan nama CobaController.php pada folder laravelapp\app\Http\Controllers\</p>  <pre> 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class CobaController extends Controller 8 { 9 // 10 } 11 </pre>
2	<p>Ubah route halaman 'profil' pada praktikum sebelumnya menjadi seperti berikut</p> <p>Route::get('profil', 'CobaController@profil');</p> <p>Langkah selanjutnya, tambahkan fungsi / method profil() pada class CobaController seperti berikut ini:</p>

	<pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request;</pre> <pre>class CobaController extends Controller { public function profil() { return view('profil'); } }</pre> <p>Ketikkan alamat http://localhost:8000/profil, dan Kita akan mendapatkan hasil yang sama seperti pada latihan sebelumnya.</p>  
	<p><u>Praktikum – Bagian 4: Memberikan Data Controller kepada View</u></p>
<p>1</p>	<p>Dalam sebuah aplikasi, sangat jarang Kita temukan isi dari file view yang bersifat statis. Pada umumnya, view dipakai untuk menampilkan data, contohnya adalah data yang berasal dari database. Namun, karena kita belum sampai pada pembahasan database, maka pada latihan kali ini kita akan mencoba menampilkan data dari variabel.</p> <p>Buat route baru untuk halaman biodata</p> <pre>Route::get('biodata', 'BiodataController@index');</pre> <p>Buat controller baru dengan nama BiodataController.php</p> <pre>php artisan make:controller BiodataController</pre>

	<p>Ketikkan script berikut dengan memberikan method / fungsi index pada laravelapp\app\Http\Controllers\BiodataController.php</p> <pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index() { \$nama = 'Sugono Galih Aprianto'; //ubah dengan nama kalian return view('biodata' , ['nama' => \$nama]); } }</pre> 
2	<p>Buat file view dengan nama biodata.blade.php pada laravelapp\resources\views\</p> <pre><!DOCTYPE html> <html> <head> <title>Biodata</title> </head> <body> <h1>Biodata</h1> <p>Nama : {{ \$nama }}</p> </body> </html></pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat http://localhost:8000/biodata</p> 
	<p><u>Praktikum – Bagian 5: Memberikan Data Array Kepada View</u></p>
1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index() { \$nama = 'Sugono Galih Aprianto'; //ubah dengan nama kalian \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; return view('biodata' , ['nama' => \$nama, 'materi' => \$materi]); } }</pre>

	<div data-bbox="272 205 1226 235"> web.php BiodataController.php biodata.blade.php profil.b </div> <div data-bbox="272 254 1226 779"> <pre> laravelapp > app > Http > Controllers > BiodataController.php 1 <?php 2 namespace App\Http\Controllers; 3 use Illuminate\Http\Request; 4 class BiodataController extends Controller 5 { 6 public function index() { 7 \$nama = 'Ahmad Bahrul Ulum'; //ubah dengan nama kalian 8 \$materi = ["Web Design", "Web Programming", "Digital 9 Marketing","Graphic Design"]; 10 return view('biodata' , ['nama' => \$nama, 'materi' => 11 \$materi]); 12 } 13 } 14 </pre> </div>
2	<p>Tambahkan isi file view biodata.blade.php menjadi seperti berikut:</p> <pre> <!DOCTYPE html> <html> <head> <title>Biodata</title> </head> <body> <h1>Biodata</h1> <p>Nama : {{ \$nama }}</p> <p>Materi Mengajar</p> @foreach(\$materi as \$datamateri) {{ \$datamateri }} @endforeach </body></html> </pre> <div data-bbox="272 1010 932 1640"> <div data-bbox="272 1010 932 1039"> web.php BiodataController.php biodata.blade.php </div> <div data-bbox="272 1052 932 1640"> <pre> laravelapp > resources > views > biodata.blade.php 1 <!DOCTYPE html> 2 <html> 3 <head> 4 <title>Biodata</title> 5 </head> 6 <body> 7 <h1>Biodata</h1> 8 <p>Nama : {{ \$nama }}</p> 9 <p>Materi Mengajar</p> 10 11 @foreach(\$materi as \$datamateri) 12 {{ \$datamateri }} @endforeach 13 14 </body> 15 </html> 16 </pre> </div> </div>

← → ↻ ⓘ localhost/laravelapp/public/biodata

 Apl  Gmail  YouTube  Maps

Biodata

Nama : Ahmad Bahrul Ulum

Materi Mengajar

- Web Design
- Web Programming
- Digital Marketing
- Graphic Design