


PEMROGRAMAN WEB LANJUT
JOBSHEET 9



Oleh:
SUGENG PRASTIYO
NIM. 1841720113

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2020

No	<p style="text-align: center;">Uraian Jawaban Instalasi Laravel di Windows</p>
1	<p>Laravel menggunakan Composer untuk mengelola dependensinya. Jadi, sebelum menggunakan Laravel, pastikan Composer telah terinstall terlebih dahulu. Composer adalah dependensi manajer untuk library PHP. Composer digunakan untuk menginstal, menghapus, dan memperbarui paket PHP. Unduh composer melalui https://getcomposer.org/, pilih Download untuk memulai proses unduh.</p> <div style="text-align: center;">  <p>COMPOSER</p> <p>A Dependency Manager for PHP</p> <p>Latest: 1.10.1</p> <div style="display: flex; flex-wrap: wrap; justify-content: center; gap: 10px;"> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">Getting Started</div> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">Download</div> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">Documentation</div> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">Browse Packages</div> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">Issues</div> <div style="border: 1px solid #ccc; padding: 5px 15px; background-color: #f9f9f9;">GitHub</div> </div> </div>
2	<p style="text-align: center;">Terdapat 2 cara dalam menginstall Laravel,</p> <p style="text-align: center;">1. Install Via Laravel Installer</p> <p>Setelah proses instalasi Composer selesai, buka Command Prompt lalu tuliskan lokasi folder yang akan menampung folder kerja kita. Karena pada praktikum sebelumnya telah menggunakan XAMPP, maka dapat menggunakan folder htdocs. Pindah ke direktori htdocs dengan ketik: cd C:\xampp\htdocs. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p style="text-align: center;">composer global require "laravel/installer"</p>

```

PS C:\Program Files\Ampps\www> composer global require "laravel/installer"
Changed current directory to C:/Users/Sugeng/AppData/Roaming/Composer
Using version ^2.3 for laravel/installer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 13 installs, 0 updates, 0 removals
- Installing symfony/process (v3.4.39): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.15.0): Downloading (100%)
- Installing symfony/filesystem (v3.4.39): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
- Installing psr/log (1.1.3): Downloading (100%)
- Installing symfony/debug (v3.4.39): Downloading (100%)
- Installing symfony/console (v3.4.39): 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 (v2.3.0): Downloading (100%)
symfony/console suggests installing symfony/event-dispatcher
symfony/console suggests installing symfony/lock
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (Emit PSR-7 response
S)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (I
DN) support)
Writing lock file
Generating autoload files
PS C:\Program Files\Ampps\www>

```

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

```

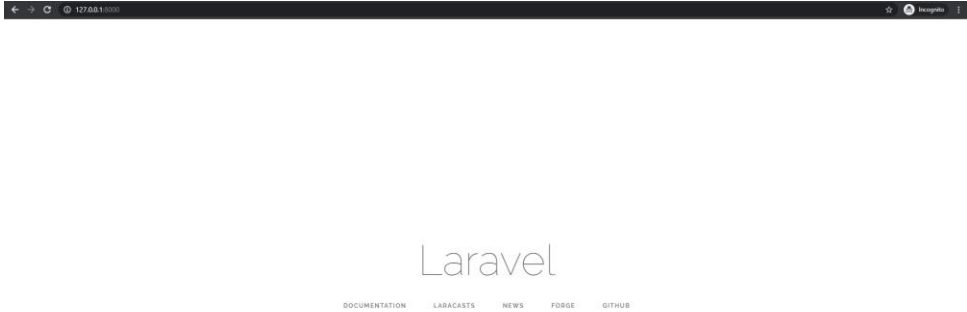
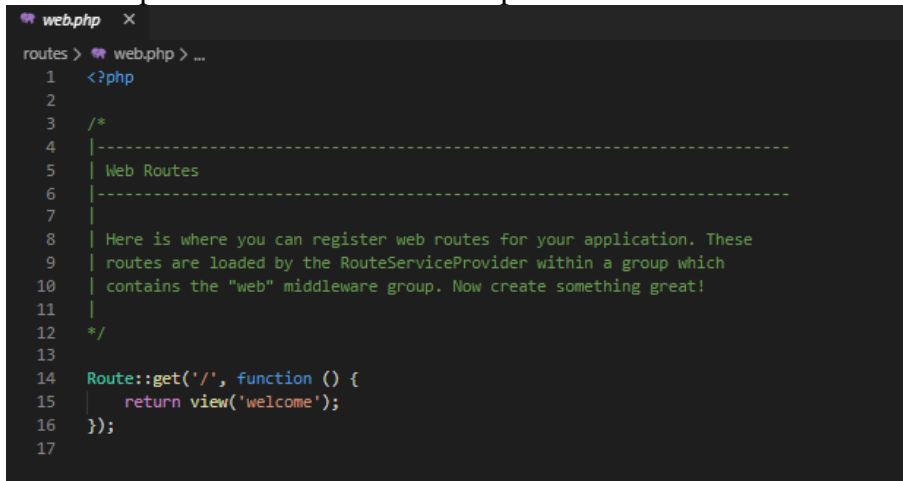
PS C:\Program Files\Ampps\www> composer create-project --prefer-dist laravel/laravel laravelapp
Installing laravel/laravel (v5.4.30)
- Installing laravel/laravel (v5.4.30): Downloading (100%)
Created project in laravelapp
> php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 64 installs, 0 updates, 0 removals
- Installing kylekatarnls/update-helper (1.2.0): Downloading (100%)
- Installing doctrine/inflector (v1.1.0): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Loading from cache
- Installing symfony/var-dumper (v3.4.39): Downloading (100%)
- Installing jakub-onderka/php-console-color (v0.2): Downloading (100%)
- Installing jakub-onderka/php-console-highlighter (v0.4): Downloading (100%)
- Installing nikic/php-parser (v3.1.5): Downloading (100%)
- Installing psr/log (1.1.3): Loading from cache
- Installing symfony/debug (v3.4.39): Loading from cache
- Installing symfony/console (v3.4.39): Loading from cache
- Installing dnoegel/php-xdg-base-dir (v0.1.1): Downloading (100%)
- Installing psy/psysh (v0.9.12): Downloading (100%)

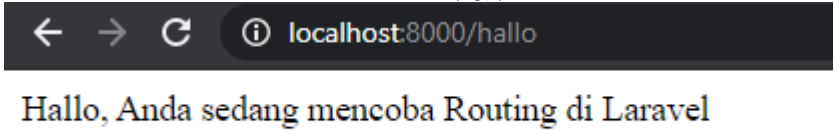
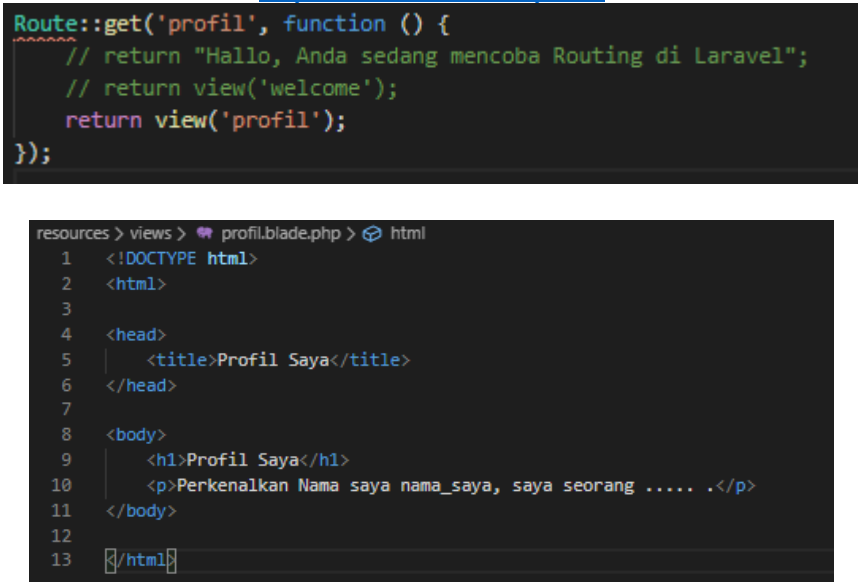
```



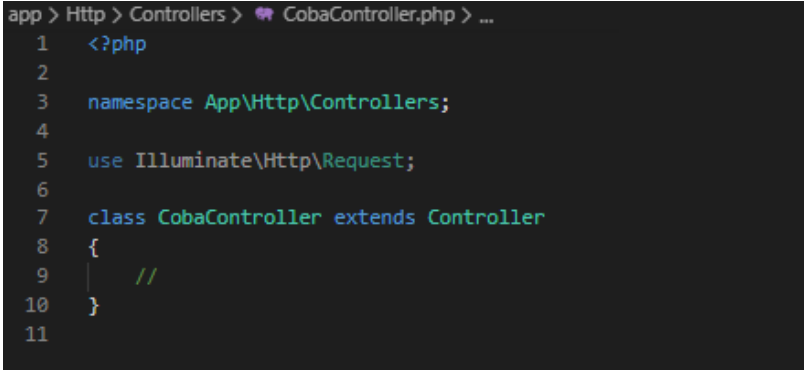
3

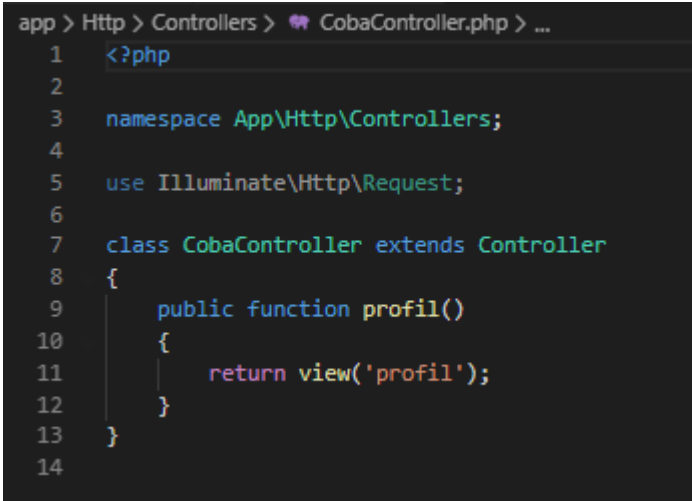
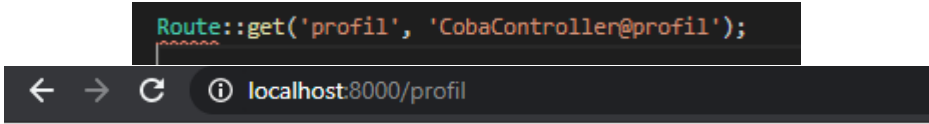
Setelah proses instalasi Laravel selesai, Kita perlu menguji apakah hasil instalasi tersebut berjalan dengan baik atau tidak. Aktifkan Apache server lewat XAMPP Control Panel. Pada browser, ketik alamat <http://localhost/laravelapp/public/> Atau Kita dapat menggunakan perintah berikut di Command Prompt:

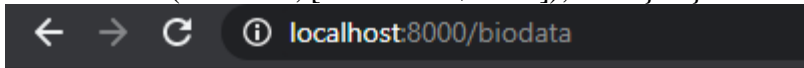
php artisan serve

	<pre>C:\Program Files\Ampps\www\laravelapp>php artisan serve Laravel development server started: <http://127.0.0.1:8000> [Mon Apr 06 10:09:47 2020] 127.0.0.1:50404 [200]: /favicon.ico [Mon Apr 06 10:09:52 2020] 127.0.0.1:50408 [200]: /favicon.ico</pre> <p>Kita dapat menjalankan aplikasi LARAVEL di folder manapun dengan menggunakan perintah di atas sehingga tidak harus selalu berada di dalam folder htdocs (XAMPP). Jalankan dengan localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :</p> 
NO	Praktikum- Bagian 2 Menenal dan membuat route
1	<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> 
2	<p>Kita dapat menuliskan perintah baru untuk membuat route, selain mengembalikan file view sebagai response, Kita dapat mengembalikan response berupa string atau teks biasa. 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>  <p>Untuk melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</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> <p>Route::get('profil', function () { return view('profil'); });</p> <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> 

	 
NO	PRAKTIKUM-Bagian 3:Cara Menggunakan Controller pada Framework Laravel
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>PS C:\Program Files\Ampps\www\laravellapp> php artisan make:controller CobaController Controller created successfully. PS C:\Program Files\Ampps\www\laravellapp></pre> 

2	<p>Ubah route halaman 'profil' pada praktikum sebelumnya menjadi seperti berikut</p> <p style="text-align: center;">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 <code>http://localhost:8000/profil</code>, dan Kita akan mendapatkan hasil yang sama seperti pada latihan sebelumnya.</p>   <p style="text-align: center;">Sugeng Prastiyo</p> <p style="text-align: center;">Perkenalkan Nama saya Sugeng Prasityo, saya seorang Mahasiswa .</p>
NO	Praktikum – Bagian 4: Memberikan Data Controller kepada View
1	<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> <p>php artisan make:controller BiodataController</p> <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>  <h1>Biodata</h1> <p>Nama : Sugeng Prastiyo</p>
NO	Praktikum – Bagian 5: Memberikan Data Array Kepada View
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>

	<pre> app > Http > Controllers > BiodataController.php > ... 1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class BiodataController extends Controller 8 { 9 // 10 public function index() 11 { 12 \$nama = 'Sugeng Prastiyo'; //ubah dengan nama kalian 13 \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; 14 return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); 15 } 16 } 17 </pre>
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> <pre> resources > views > biodata.blade.php > html 1 <!DOCTYPE html> 2 <html> 3 4 <head> 5 <title>Biodata</title> 6 </head> 7 8 <body> 9 <h1>Biodata</h1> 10 <p>Nama : {{ \$nama }}</p> 11 <p>Materi Mengajar</p> 12 @foreach(\$materi as \$datamateri) 13 {{ \$datamateri }} 14 @endforeach 15 </body> 16 17 </html> </pre>

Biodata

Nama : Sugeng Prastiyo

Materi Mengajar

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