

LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB LANJUT

Jobsheet-9  
Pengantar dan Instalasi Laravel



NAMA : DINA RISKY ALIN SAPUTRI  
NIM : 1841720016  
KELAS : 2B

PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2020

## Praktikum – Bagian 1: Instalasi Laravel di Windows

Langka h	Keterangan
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 <a href="https://getcomposer.org/">https://getcomposer.org/</a>, pilih Download untuk memulai proses unduh.</p>  <p>The screenshot shows the Composer website. At the top is a cartoon character with the word 'COMPOSER' below it. Underneath is the text 'A Dependency Manager for PHP' and 'Latest: 1.10.1'. There are several buttons: 'Getting Started', 'Download', 'Documentation', 'Browse Packages', 'Issues', and 'GitHub'. At the bottom, it says 'Authors: Nils Adermann, Jordi Boggiano and many community contributions' and 'Sponsored by:'.</p>
2	<p><b>Install Via Laravel Installer</b></p> <p>Setelah proses instalasi Composer selesai, buka Command Prompt lalu tujulah 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: <b>cd C:\xampp\htdocs\LARAVEL</b>. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p style="text-align: center;"><b>composer global require "laravel/installer"</b></p>  <p>The screenshot shows a terminal window with the following output:</p> <pre>PS C:\xampp\htdocs\LARAVEL&gt; 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): Loading from cache - Installing psr/http-message (1.0.1): Loading from cache - Installing guzzlehttp/psr7 (1.6.1): Loading from cache - Installing guzzlehttp/promises (v1.3.1): Loading from cache - Installing guzzlehttp/guzzle (6.5.2): Loading from cache - 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 0 packages you are using are looking for funding. Use the composer fund command to find out more!</pre>

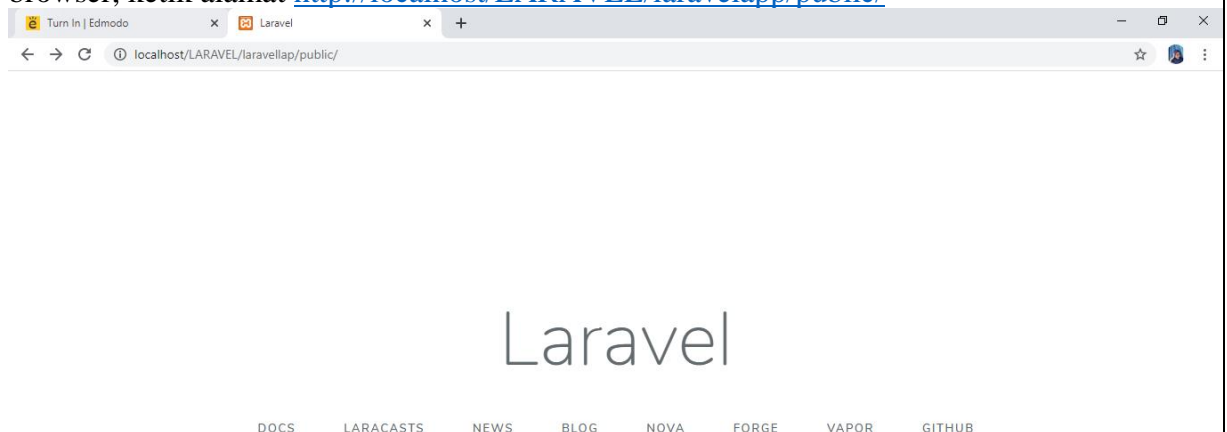
Tunggulah hingga proses penginstalan selesai. Setelah itu, Kita bisa mulai membuat project laravel dengan mengetikkan perintah:

### laravel new laravelapp

```
PS C:\xampp\htdocs\LARAVEL> laravel new laravelapp
Crafting application...
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 94 installs, 0 updates, 0 removals
- Installing doctrine/inflector (1.3.1): Downloading (100%)
- Installing doctrine/lexer (1.2.0): Downloading (100%)
- Installing dragonmantank/cron-expression (v2.3.0): Downloading (100%)
- Installing voku/portable-ascii (1.4.10): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.15.0): Loading from cache
- Installing phoption/phoption (1.7.3): Downloading (100%)
- Installing vlucas/phpdotenv (v4.1.3): Downloading (100%)
- Installing symfony/css-selector (v5.0.7): Downloading (100%)
- Installing tijsverkoyen/css-to-inline-styles (2.2.2): Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Loading from cache
- Installing symfony/var-dumper (v5.0.7): Downloading (100%)
- Installing symfony/routing (v5.0.7): Downloading (100%)
- Installing symfony/process (v5.0.7): Loading from cache
- Installing symfony/polyfill-php72 (v1.15.0): Downloading (100%)
- Installing symfony/polyfill-intl-idn (v1.15.0): Downloading (100%)
- Installing symfony/mime (v5.0.7): Downloading (100%)
- Installing symfony/polyfill-php73 (v1.15.0): Loading from cache
- Installing symfony/http-foundation (v5.0.7): Downloading (100%)
- Installing psr/event-dispatcher (1.0.0): Downloading (100%)
- Installing symfony/event-dispatcher-contracts (v2.0.1): Downloading (100%)
- Installing symfony/event-dispatcher (v5.0.7): Downloading (100%)
- Installing psr/log (1.1.3): Downloading (100%)
- Installing symfony/error-handler (v5.0.7): Downloading (100%)
- Installing symfony/http-kernel (v5.0.7): Downloading (100%)
- Installing symfony/finder (v5.0.7): Downloading (100%)
- Installing psr/container (1.0.0): Loading from cache
- Installing symfony/service-contracts (v2.0.1): Loading from cache
- Installing symfony/console (v5.0.7): Loading from cache
- Installing symfony/polyfill-iconv (v1.15.0): Downloading (100%)
- Installing egulias/email-validator (2.1.17): Downloading (100%)
- Installing swiftmailer/swiftmailer (v6.2.3): Downloading (100%)
- Installing ramsey/collection (1.0.1): Downloading (100%)
- Installing brick/math (0.8.14): Downloading (100%)
- Installing ramsey/uuid (4.0.1): Downloading (100%)
```

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

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/LARAVEL/laravelapp/public/>



## php artisan serve

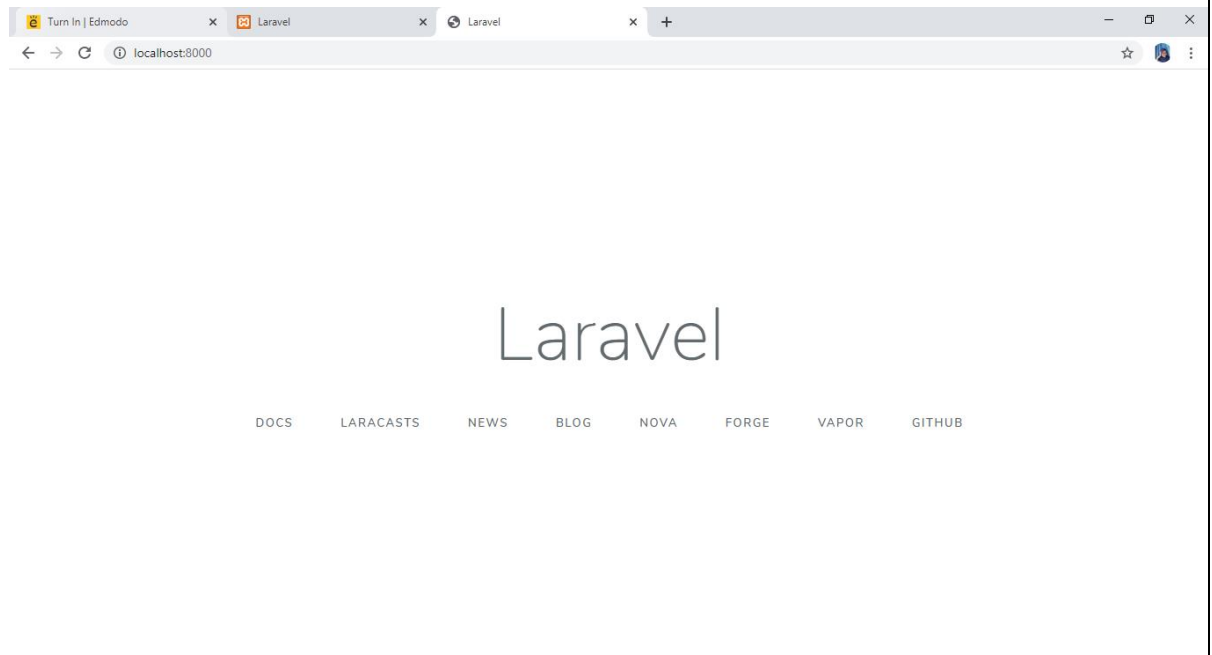
```

20: route::get('hello', function () {
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

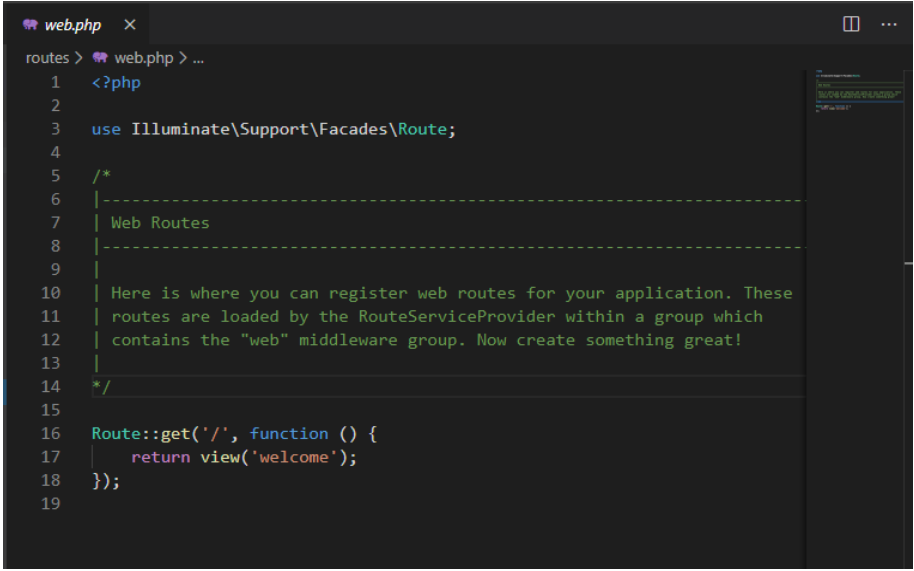
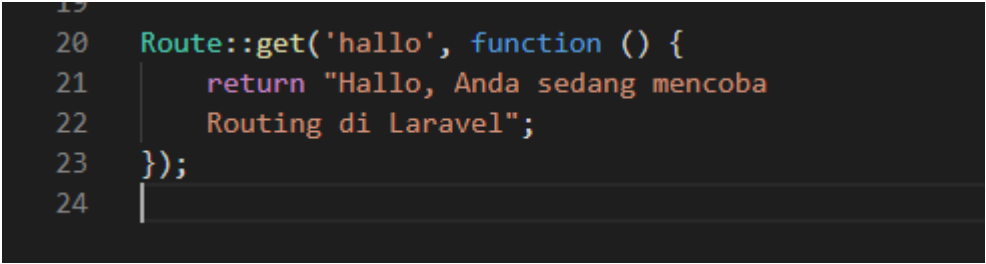
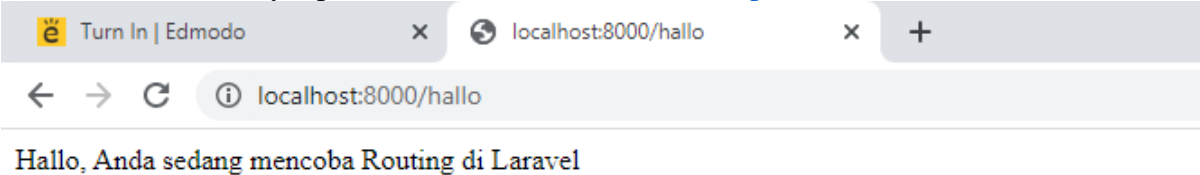
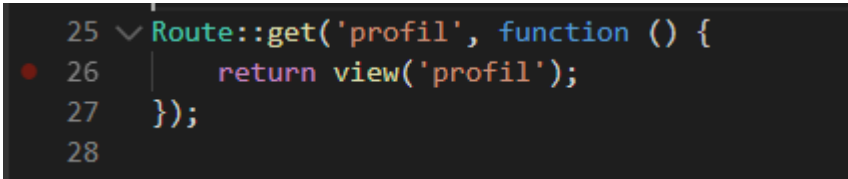
PS C:\xampp\htdocs\LARAVEL\laravelapp> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Apr  6 08:23:00 2020] 127.0.0.1:51348 Invalid request (An existing connection was forcibly closed by the remote host.
)
[Mon Apr  6 08:23:02 2020] 127.0.0.1:51345 Invalid request (An existing connection was forcibly closed by the remote host.
)
[Mon Apr  6 08:23:22 2020] 127.0.0.1:51363 [200]: /favicon.ico
  
```

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 :



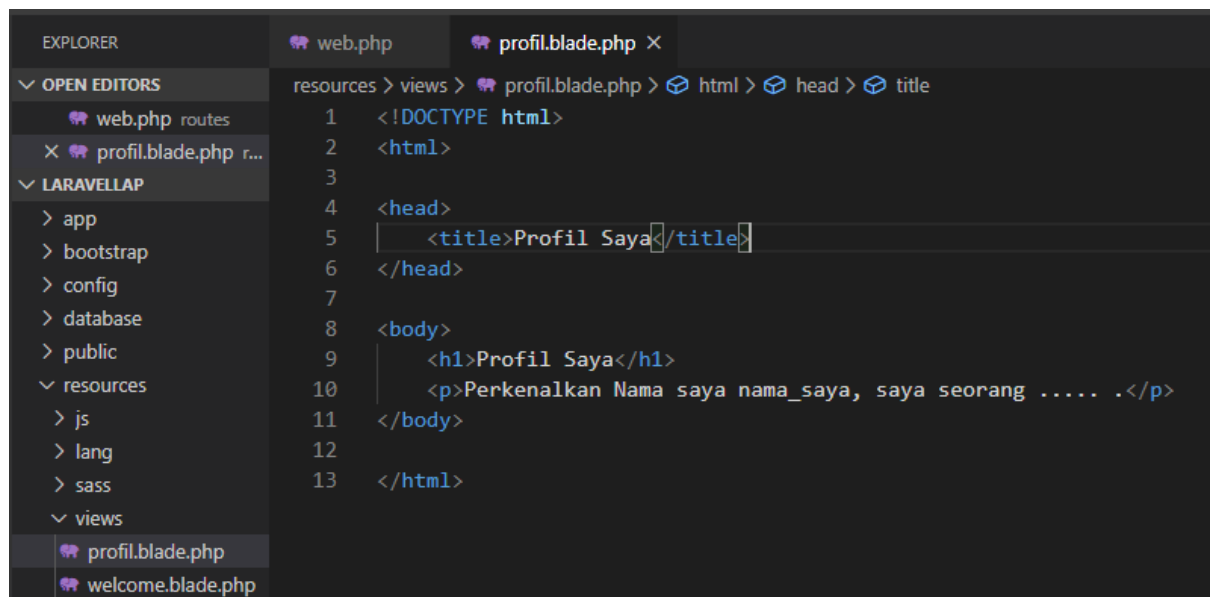
## Praktikum – Bagian 2: Mengenal dan membuat route

Langka h	Keterangan
1	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 <b>laravelapp\routes\web.php</b> Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut

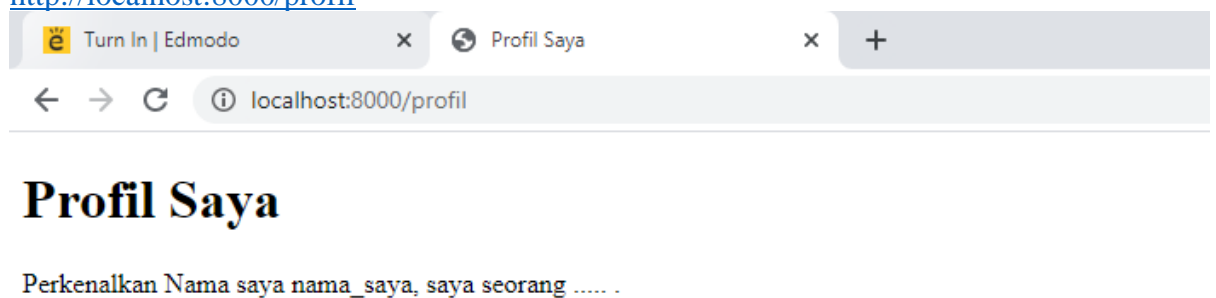
	 <pre> 1 &lt;?php 2 3 use Illuminate\Support\Facades\Route; 4 5 /* 6  ----- 7   Web Routes 8  ----- 9   10   Here is where you can register web routes for your application. These 11   routes are loaded by the RouteServiceProvider within a group which 12   contains the "web" middleware group. Now create something great! 13   14 */ 15 16 Route::get('/', function () { 17     return view('welcome'); 18 }); 19 </pre>
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 <code>laravelapp\routes\web.php</code>, tambahkan isinya dengan script di bawah ini :</p> <p><b>Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; });</b></p>  <pre> 19 20 Route::get('hallo', function () { 21     return "Hallo, Anda sedang mencoba 22     Routing di Laravel"; 23 }); 24 </pre> <p>Untuk melihat hasilnya, pada browser, ketikkan alamat <a href="http://localhost:8000/hallo">http://localhost:8000/hallo</a></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><b>Route::get('profil', function () { return view('profil'); });</b></p>  <pre> 25 Route::get('profil', function () { 26     return view('profil'); 27 }); 28 </pre>

Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:

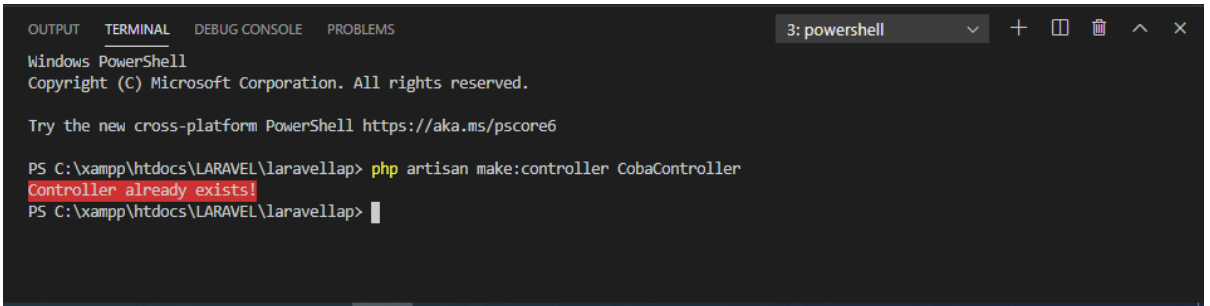
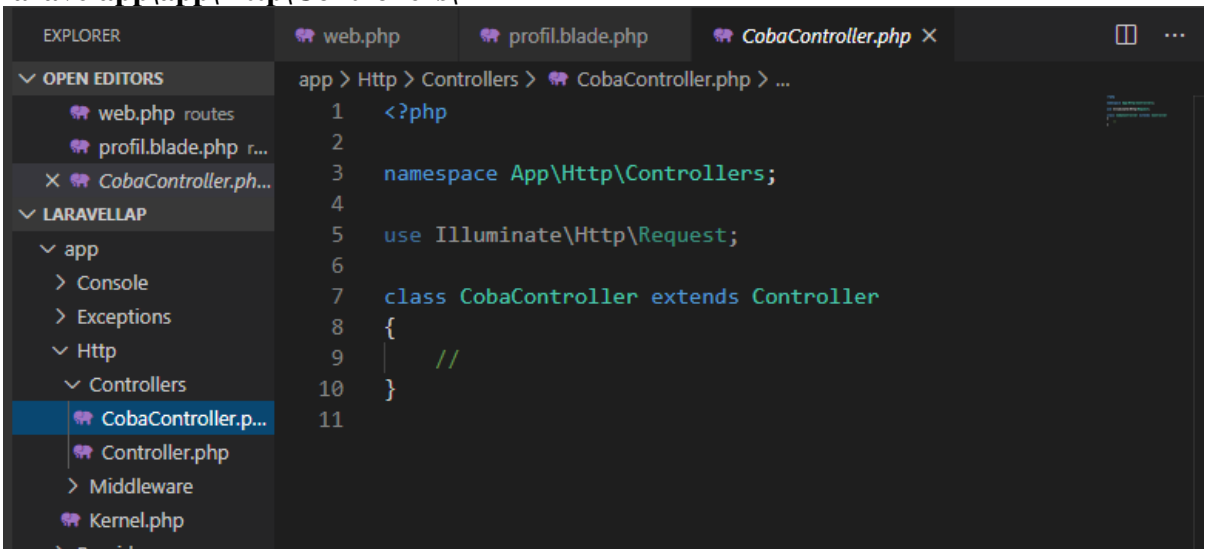
```
<!DOCTYPE html>
<html>
<head>
    <title>Profil Saya</title>
</head>
<body>
    <h1>Profil Saya</h1>
    <p>Perkenalkan Nama saya nama_saya, saya seorang .....
.</p>
</body>
</html>
```

A screenshot of a code editor interface. On the left is the 'EXPLORER' sidebar showing a file tree with folders like 'app', 'bootstrap', 'config', 'database', 'public', 'resources', 'js', 'lang', 'sass', and 'views'. The 'views' folder is expanded, showing 'profil.blade.php' and 'welcome.blade.php'. The main editor area shows the code for 'profil.blade.php' with line numbers 1 through 13. The code is: 1 <!DOCTYPE html>, 2 <html>, 3, 4 <head>, 5 <title>Profil Saya</title>, 6 </head>, 7, 8 <body>, 9 <h1>Profil Saya</h1>, 10 <p>Perkenalkan Nama saya nama\_saya, saya seorang ..... </p>, 11 </body>, 12, 13 </html>.

Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat <http://localhost:8000/profil>



## Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel

Langka h	Keterangan
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;"><b>php artisan make:controller CobaController</b></p>  <p>Maka akan terbentuk sebuah file dengan nama <b>CobaController.php</b> pada folder <b>laravelapp\app\Http\Controllers\</b></p>  <pre>1 &lt;?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 style="text-align: center;"><b>Route::get('profil', 'CobaController@profil');</b></p> <div style="background-color: #f0f0f0; padding: 10px; text-align: center;"><pre>28 Route::get('profil', 'CobaController@profil');</pre></div> <p>Langkah selanjutnya, tambahkan fungsi / method profil() pada class <b>CobaController</b> seperti berikut ini:</p>

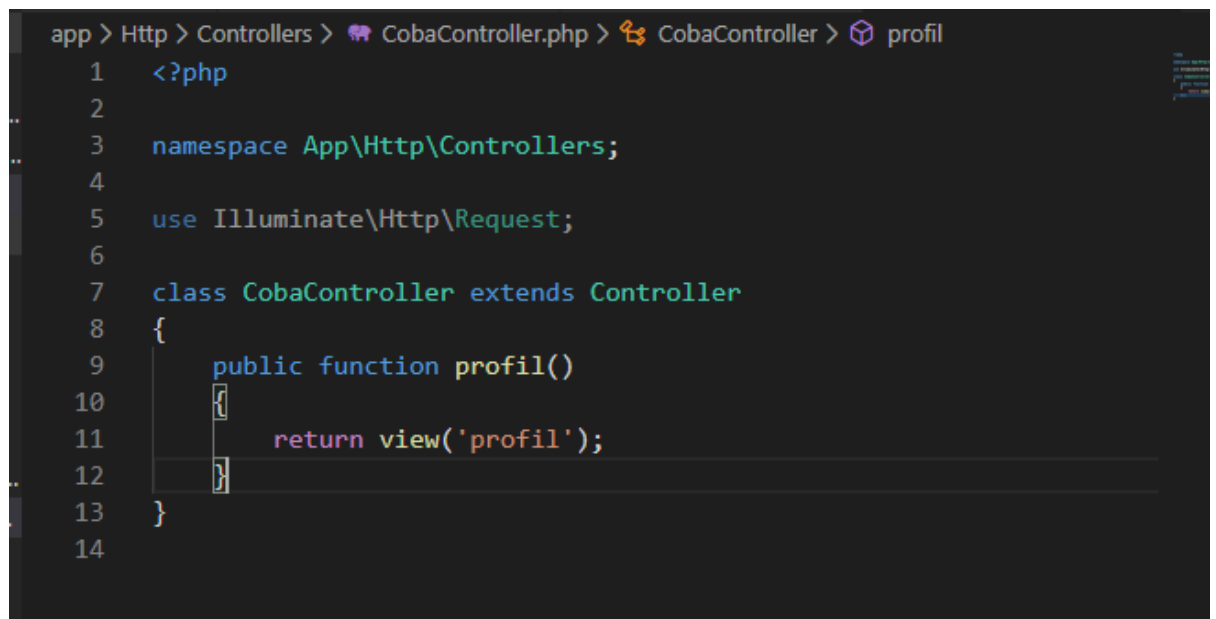
```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

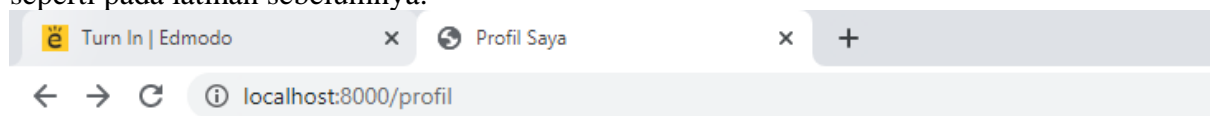
```
class CobaController extends Controller
```

```
{  
    public function profil()  
    {  
        return view('profil');  
    }  
}
```



```
app > Http > Controllers > CobaController.php > CobaController > profil  
1  <?php  
2  
3  namespace App\Http\Controllers;  
4  
5  use Illuminate\Http\Request;  
6  
7  class CobaController extends Controller  
8  {  
9      public function profil()  
10     {  
11         return view('profil');  
12     }  
13 }  
14
```

Ketikkan alamat <http://localhost:8000/profil>, dan Kita akan mendapatkan hasil yang sama seperti pada latihan sebelumnya.



## Profil Saya

Perkenalkan Nama saya nama\_saya, saya seorang .....

- 3 Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:

```
Route::get('profil', function () { return view('profil'); });
```



```

25 Route::get('profil', function () {
26     return view('profil');
27 });
28

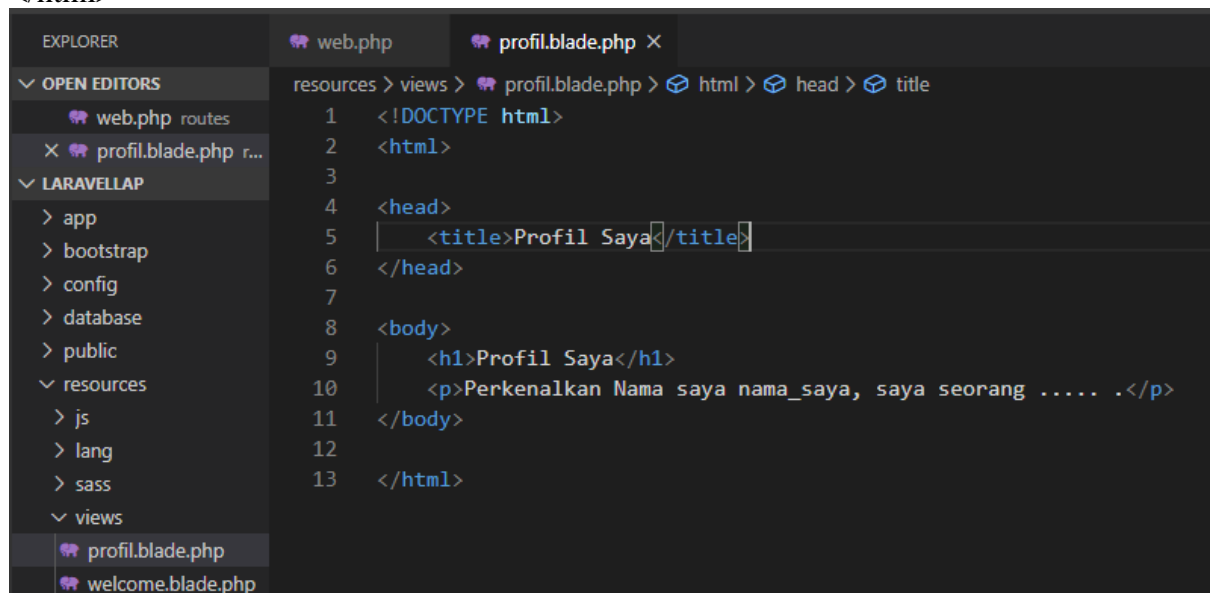
```

Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:

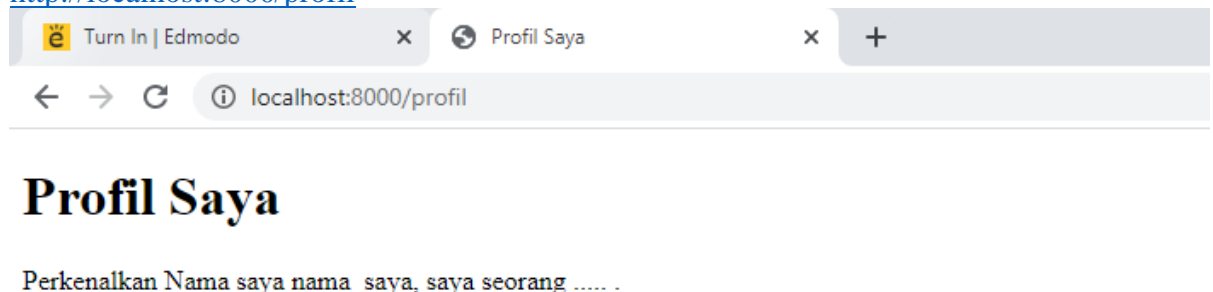
```

<!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>

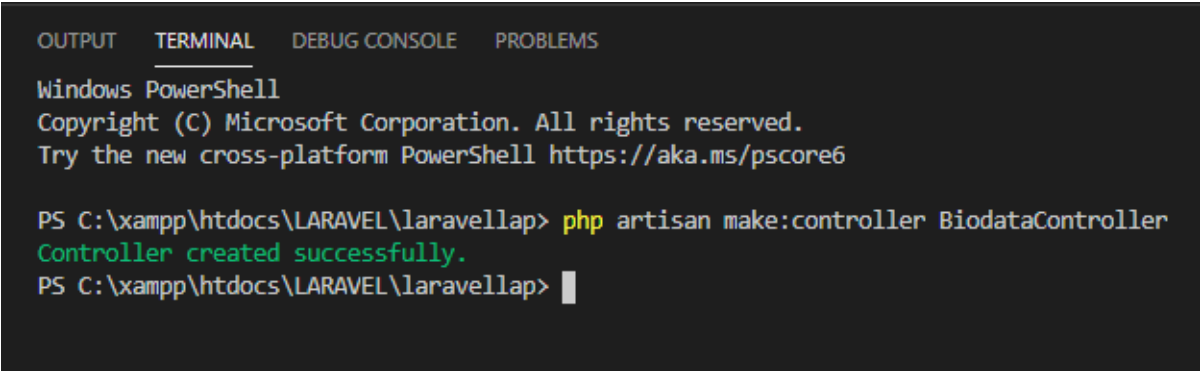
```



Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat <http://localhost:8000/profil>



## Praktikum – Bagian 4: Memberikan Data Controller kepada View

Langka h	Keterangan
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> <p style="text-align: center;"><b>Route::get('biodata', 'BiodataController@index');</b></p> <pre>29 Route::get('biodata', 'BiodataController@index'); 30</pre> <p>Buat controller baru dengan nama <b>BiodataController.php</b></p> <p style="text-align: center;"><b>php artisan make:controller BiodataController</b></p>  <p>Ketikkan script berikut dengan memberikan method / fungsi index pada <b>laravelapp\app\Http\Controllers\BiodataController.php</b></p> <pre>&lt;?php namespace App\Http\Controllers; use Illuminate\Http\Request;  class BiodataController extends Controller {     public function index() {         \$nama = 'Sugono Galih Aprianto'; <i>//ubah dengan nama kalian</i>         return view('biodata', ['nama' =&gt; \$nama]);     } }</pre>

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class BiodataController extends Controller
8  {
9      public function index()
10     {
11         $nama = 'Dina Risky Alin Saputri'; //ubah dengan nama kalian
12         return view('biodata', ['nama' => $nama]);
13     }
14 }
15

```

2

Buat file view dengan nama **biodata.blade.php** pada **laravelapp/resources/views/**

```

<!DOCTYPE html>
<html>
<head>
    <title>Biodata</title>
</head>
<body>
    <h1>Biodata</h1>
    <p>Nama : {{ $nama }}</p>
</body>
</html>

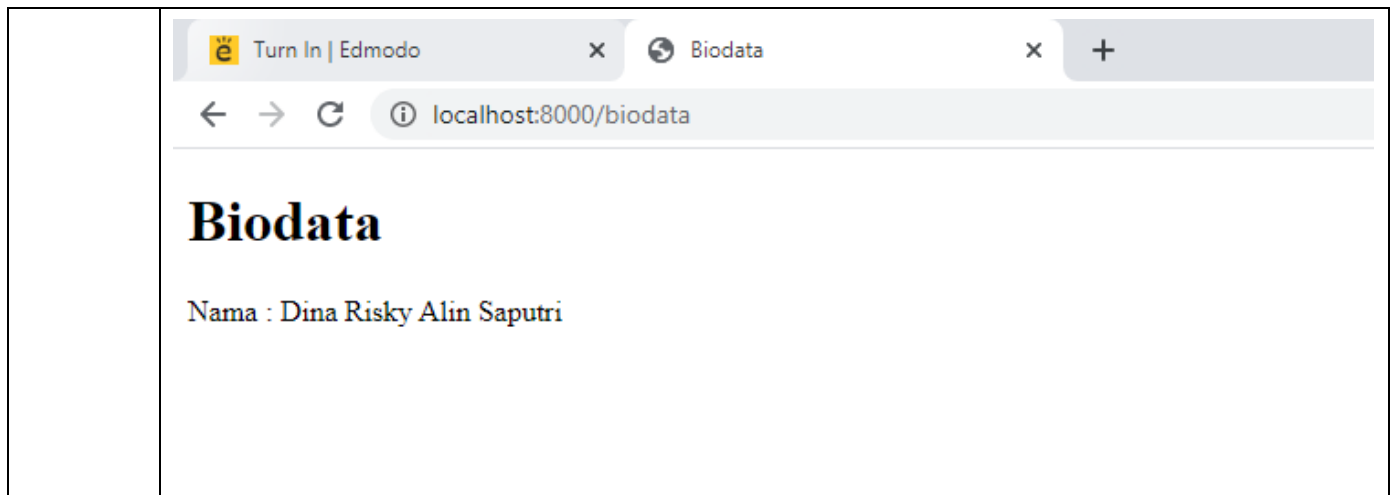
```

```

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 </body>
12
13 </html>

```

Untuk melihat hasilnya, pada browser, ketik alamat <http://localhost:8000/biodata> .



### Praktikum – Bagian 5: Memberikan Data Array Kepada View

Langka h	Keterangan
1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada <b>BiodataController.php</b></p> <pre> &lt;?php namespace App\Http\Controllers; use Illuminate\Http\Request;  class BiodataController extends Controller {     public function index() {         \$nama = 'Sugono Galih Aprianto'; <i>//ubah dengan nama kalian</i>         \$materi = ["Web Design", "Web Programming", "Digital                     Marketing", "Graphic Design"];         return view('biodata', ['nama' =&gt; \$nama, 'materi' =&gt;             \$materi]);     } } </pre>

	 <pre> web.php BiodataController.php x biodata.blade.php CobaController.php app &gt; Http &gt; Controllers &gt; BiodataController.php &gt; BiodataController &gt; index 1  &lt;?php 2 3  namespace App\Http\Controllers; 4 5  use Illuminate\Http\Request; 6 7  class BiodataController extends Controller 8  { 9      public function index() 10     { 11         \$nama = 'Dina Risky Alin Saputri'; //ubah dengan nama kalian 12         // return view('biodata', ['nama' =&gt; \$nama]); 13         \$materi = ["Web Design", "Web Programming", "Digital 14         Marketing", "Graphic Design"]; 15         return view('biodata', ['nama' =&gt; \$nama, 'materi' =&gt; 16         \$materi]); 17     } 18 } 19 </pre>
2	<p>Buat file view dengan nama <b>biodata.blade.php</b> pada <b>laravelapp\resources\views\</b></p> <pre> &lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt;     &lt;title&gt;Biodata&lt;/title&gt; &lt;/head&gt; &lt;body&gt;     &lt;h1&gt;Biodata&lt;/h1&gt;     &lt;p&gt;Nama : {{ \$nama }}&lt;/p&gt;     &lt;ul&gt;         @foreach(\$materi as \$datamateri)             &lt;li&gt;{{ \$datamateri }}&lt;/li&gt;         @endforeach     &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt; </pre>

```
web.php x BiodataController.php biodata.blade.php x CobaController.php
resources > views > biodata.blade.php > html > body
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Biodata</title>
6 </head>
7
8 <body>
9   <h1>Biodata</h1>
10  <p>Nama : {{ $nama }}</p>
11  <p>Materi Mengajar</p>
12  <ul>
13    @foreach($materi as $datamateri) <li>{{ $datamateri }}</li>
14    @endforeach
15  </ul>
16
17 </body>
18
19 </html>
```

Untuk melihat hasilnya, pada browser, ketik alamat <http://localhost:8000/biodata> .

