

**LAPORAN PRAKTIKUM
PEMROGRAMAN WEB LANJUT**

**Jobsheet - 9:
Pengantar dan Instalasi Laravel**



Disusun oleh:
Livia Yurike Khuril Maula
D4 TI – 2A / 16
NIM : 1841720025

**POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
PROGRAM STUDI D4 TEKNIK INFORMATIKA**

APRIL 2020


Topik

Mengenal Framework Laravel dan Proses Penginstallannya Tujuan

Mahasiswa diharapkan dapat:

1. Memahami framework Laravel
2. Memahami cara menginstall dan menjalankan Laravel

Praktikum – Bagian 1: Instalasi Laravel di Windows

Langkah	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 menginstall, menghapus, dan memperbarui paket PHP. Unduh composer melalui https://getcomposer.org/, pilih Download untuk memulai proses unduh</p> 
2	<p>Terdapat 2 cara dalam menginstall Laravel,</p> <ol style="list-style-type: none">1. Install Via Laravel Installer <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: <code>cd C:\xampp\htdocs</code>. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p style="text-align: center;">composer global require "laravel/installer"</p>

Jawab :

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs>composer global require "laravel/installer"

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): Loading from cache
  - 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
5 packages you are using are looking for funding.
Use the composer fund command to find out more!
```

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

Jawab :

```
C:\Windows\System32\cmd.exe

C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel laravelapp
Creating a "laravel/laravel" project at "../laravelapp"
Installing laravel/laravel (v5.5.28)
- Installing laravel/laravel (v5.5.28): Loading from cache
Created project in C:\xampp\htdocs\laravelapp
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 74 installs, 0 updates, 0 removals
- Installing kylekatarnls/update-helper (1.2.0): Loading from cache
- Installing symfony/thanks (v1.2.5): Loading from cache
- Installing symfony/polyfill-ctype (v1.15.0): Loading from cache
- Installing vlucas/phpdotenv (v2.6.2): Loading from cache
- Installing symfony/css-selector (v3.3.6): Loading from cache
- Installing tijsverkoyen/css-to-inline-styles (2.2.2): Loading from cache
- Installing symfony/polyfill-mbstring (v1.15.0): Loading from cache
- Installing symfony/var-dumper (v3.3.6): Loading from cache
- Installing symfony/routing (v3.3.6): Downloading (100%)
- Installing symfony/process (v3.3.6): Loading from cache
- Installing psr/log (1.1.3): Loading from cache
- Installing symfony/debug (v3.3.6): Loading from cache
- Installing symfony/polyfill-php72 (v1.15.0): Downloading (100%)
- Installing symfony/polyfill-intl-idn (v1.15.0): Downloading (100%)
- Installing paragonie/random-compat (v9.99.99): Downloading (100%)
- Installing psr/container (1.0.0): Downloading (100%)
- Installing symfony/event-dispatcher (v3.3.6): Downloading (100%)
- Installing symfony/http-foundation (v3.3.6): Downloading (100%)
- Installing symfony/http-kernel (v3.3.6): Downloading (100%)
- Installing symfony/finder (v3.3.6): Downloading (100%)
- Installing symfony/console (v3.3.6): Loading from cache
sebastian/global-state suggests installing ext-uopz (*)
phpunit/phpunit-mock-objects suggests installing ext-soap (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.5.5)
phpunit/phpunit suggests installing phpunit/php-invoker (^1.1)
phpunit/phpunit suggests installing ext-xdebug (*)
Package mtdowling/cron-expression is abandoned, you should avoid using it. Use dragonmantank/cron-expression instead.
Package jakub-ondierka/php-console-color is abandoned, you should avoid using it. Use php-parallel-lint/php-console-color instead.
Package jakub-ondierka/php-console-highlighter is abandoned, you should avoid using it. Use php-parallel-lint/php-console-highlighter instead.
Package phpunit/phpunit-mock-objects is abandoned, you should avoid using it. No replacement was suggested.
Writing lock file
Generating optimized autoload files
Carbon 1 is deprecated, see how to migrate to Carbon 2.
https://carbon.nesbot.com/docs/#api-carbon-2
You can run ".\vendor\bin\upgrade-carbon" to get help in updating carbon and other frameworks and libraries that depend on it.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover
Discovered Package: fideloper/proxy
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Package manifest generated successfully.
7 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php artisan key:generate
Application key [base64:fgvvoeWezRsw1KH1fhh8lQthN4xCCIMSF3CS1uN6RnA=] set successfully.

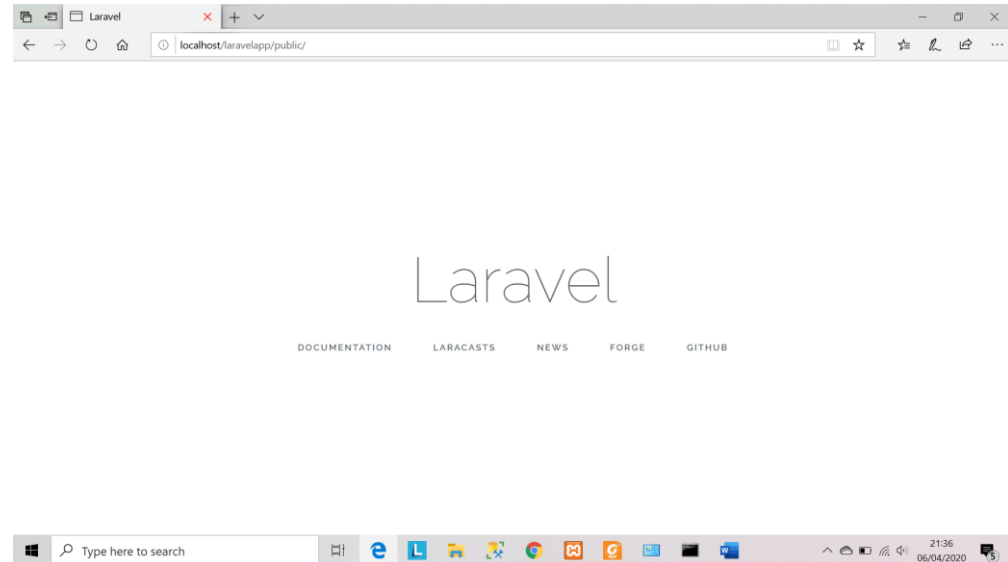
C:\xampp\htdocs>
```

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/>

Jawab :



Atau Kita dapat menggunakan perintah berikut di Command Prompt:

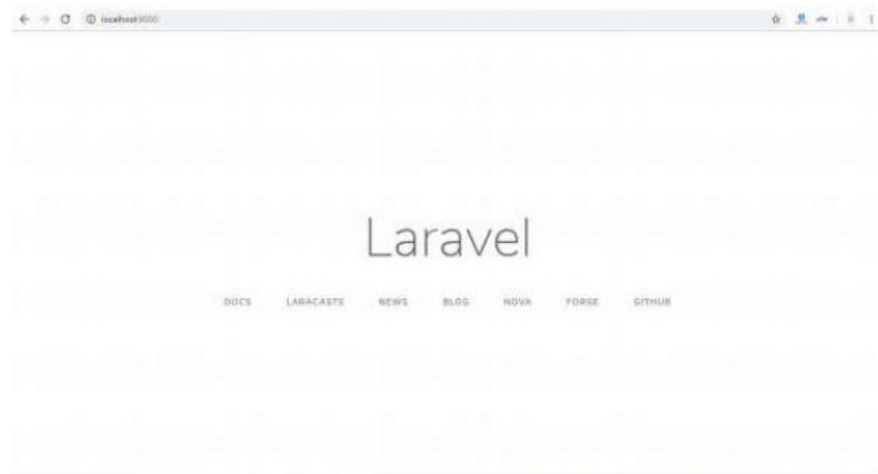
php artisan serve

Jawab :

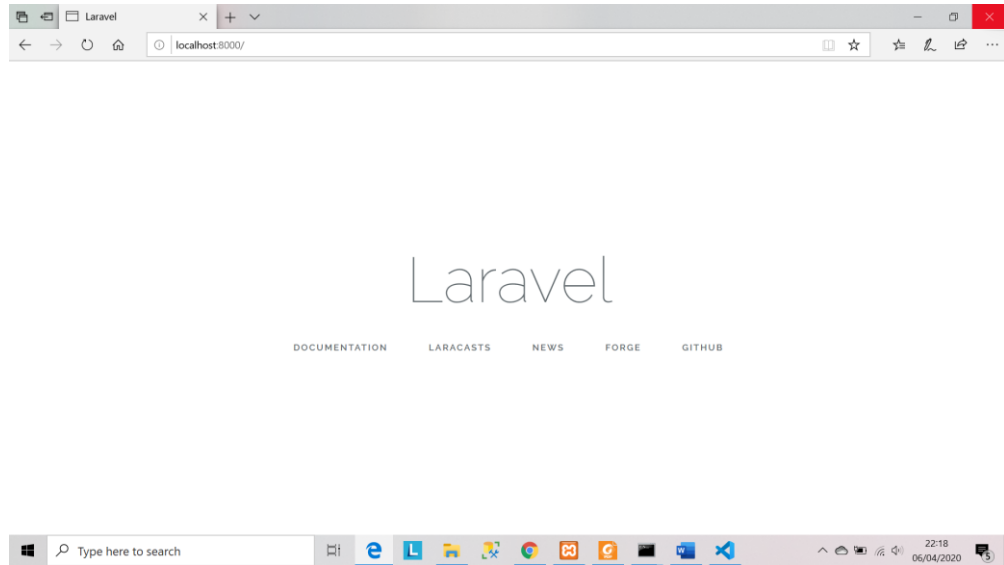
```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\laravelapp>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

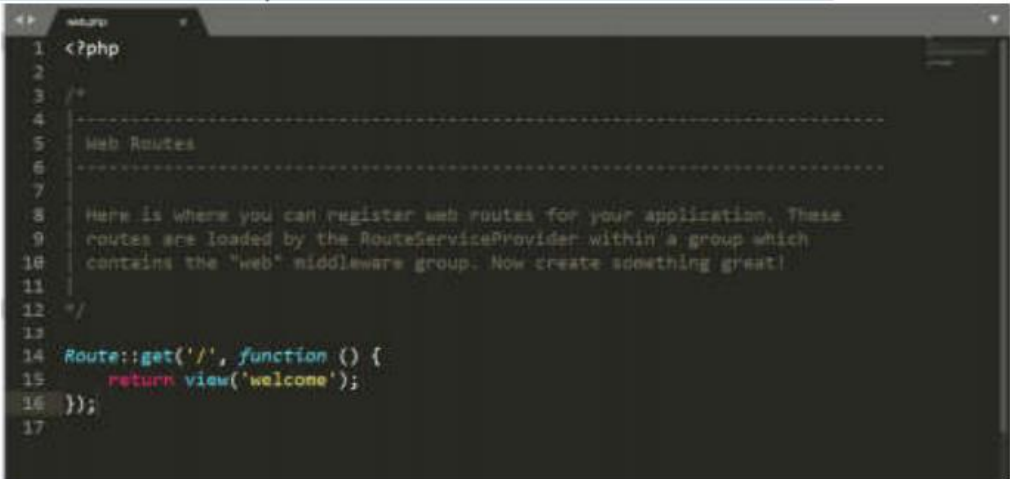
Kita dapat menjalankan aplikasi LARAVEL di folder manapun dengan menggunakan perintah di atas sehingga tidak harus selalu berada di dalam folder htdocs (XAMPP). Jalankan dengan localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :



Jawab :



Praktikum – Bagian 2: Mengenal dan membuat route

Langkah	Keterangan
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>  <pre>1 <?php 2 3 /* 4 * Web Routes 5 * 6 * Here is where you can register web routes for your application. These 7 * routes are loaded by the RouteServiceProvider within a group which 8 * contains the "web" middleware group. Now create something great! 9 */ 10 11 12 13 14 Route::get('/', function () { 15 return view('welcome'); 16 }); 17</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 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>

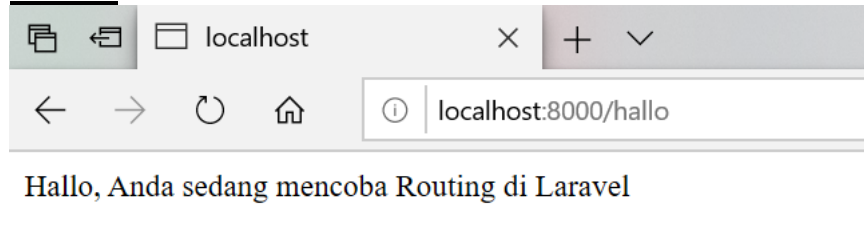
Jawab :

```
web.php x
routes > web.php > ...
1  <?php
2
3  /*
4  |-----
5  | Web Routes
6  |-----
7  |
8  | Here is where you can register web routes for your application. These
9  | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('hallo', function () {
15     return "Hallo, Anda sedang mencoba Routing di Laravel";
16 });
17
```

Untuk melihat hasilnya, pada browser, ketikkan alamat

<http://localhost:8000/hallo>

Jawab :



Hallo, Anda sedang mencoba Routing di Laravel

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'); });

Jawab :

```
17
18 Route::get('profil', function () {
19     return view('profil');
20 });
21
```

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

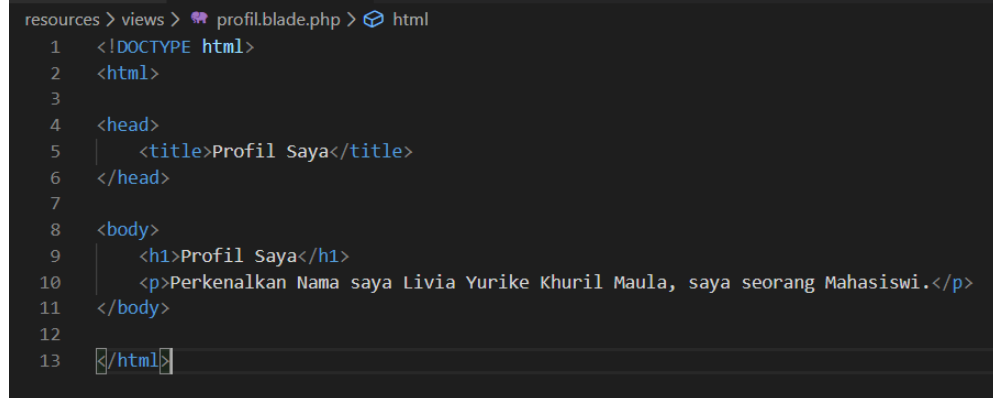
```
<!DOCTYPE html>
<html>
<head>
    <title>Profil Saya</title>
</head>
<body>
```

```

<h1>Profil Saya</h1>
<p>Perkenalkan Nama saya nama_saya, saya seorang .....
</p>
</body>
</html>

```

Jawab :



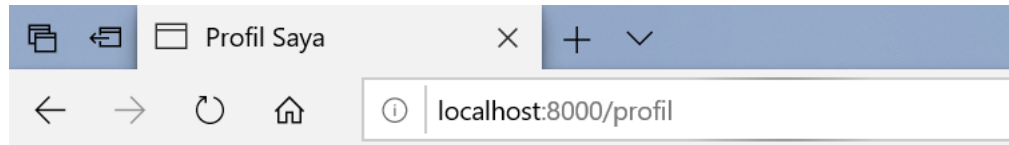
```

resources > views > profil.blade.php > html
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 Livia Yurike Khuril Maula, saya seorang Mahasiswi.</p>
11 </body>
12
13 </html>

```

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

Jawab :



Profil Saya

Perkenalkan Nama saya Livia Yurike Khuril Maula, saya seorang Mahasiswi.

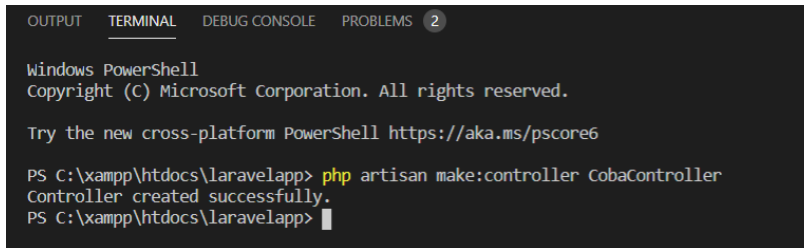
Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel

Langkah	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>

Kita gunakan cara yang kedua, pada latihan kali ini kita akan membuat Controller dengan nama CobaController dengan menuliskan di command prompt / terminal :

php artisan make:controller CobaController

Jawab :



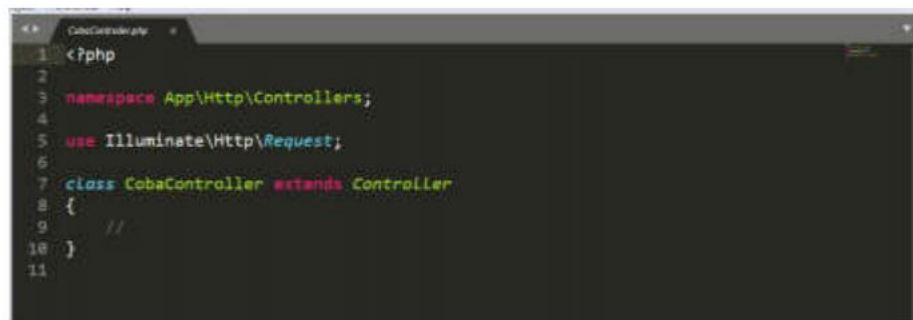
```
OUTPUT  TERMINAL  DEBUG CONSOLE  PROBLEMS 2

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

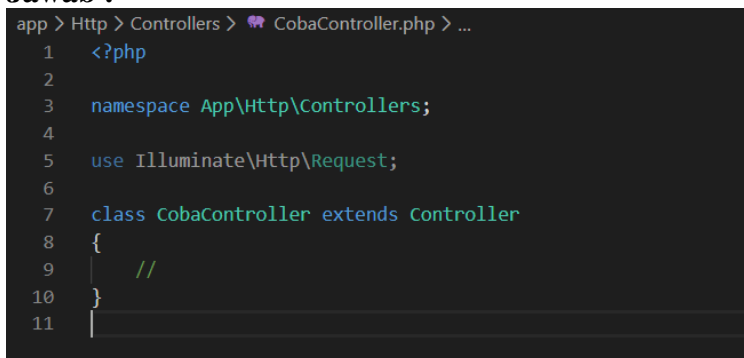
PS C:\xampp\htdocs\laravelapp> php artisan make:controller CobaController
Controller created successfully.
PS C:\xampp\htdocs\laravelapp>
```

Maka akan terbentuk sebuah file dengan nama **CobaController.php** pada folder **laravelapp\app\Http\Controllers**



```
CobaController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CobaController extends Controller
8 {
9     //
10 }
11
```

Jawab :



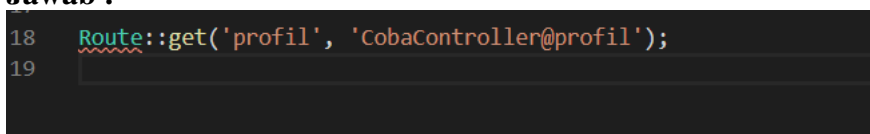
```
app > Http > Controllers > CobaController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CobaController extends Controller
8 {
9     //
10 }
11
```

2

Ubah route halaman 'profil' pada praktikum sebelumnya menjadi seperti berikut

Route::get('profil', 'CobaController@profil');

Jawab :



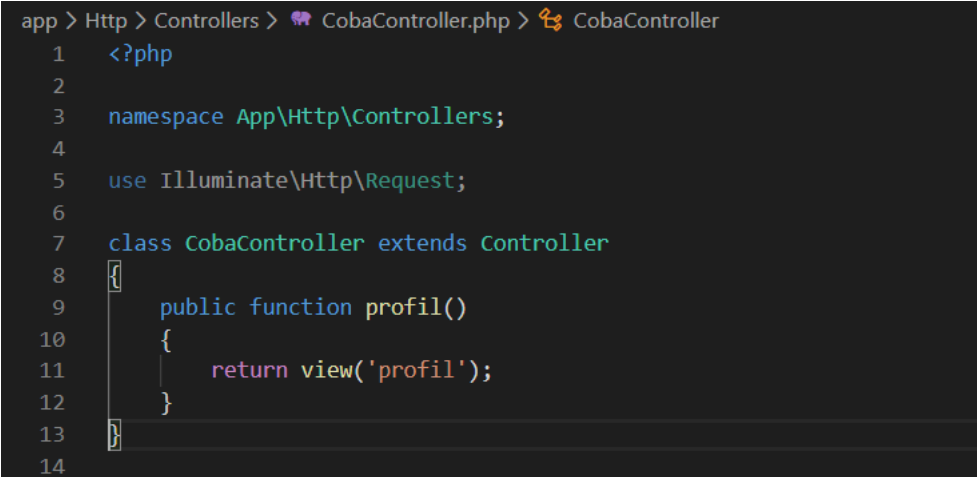
```
18 Route::get('profil', 'CobaController@profil');
19
```

Langkah selanjutnya, tambahkan fungsi / method profil() pada class CobaController seperti berikut ini:

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

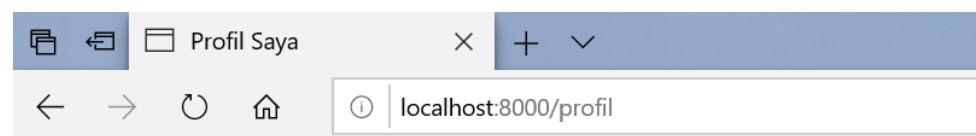
class CobaController extends Controller
{
    public function profil()
    {
        return view('profil');
    }
}
```

Jawab :



```
app > Http > Controllers > CobaController.php > CobaController
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 Livia Yurike Khuril Maula, saya seorang Mahasiswi.

Praktikum – Bagian 4: Memberikan Data Controller kepada View

Langkah	Keterangan
1	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.

Buat route baru untuk halaman biodata

Route::get('biodata', 'BiodataController@index');

Jawab :

```
20 Route::get('biodata', 'BiodataController@index');
21
```

Buat controller baru dengan nama **BiodataController.php**

php artisan make:controller BiodataController

Jawab :

```
controller created successfully.
PS C:\xampp\htdocs\laravelapp> php artisan make:controller BiodataController
Controller created successfully.
PS C:\xampp\htdocs\laravelapp>
```

Ketikkan script berikut dengan memberikan method / fungsi index pada **laravelapp\app\Http\Controllers\BiodataController.php**

```
<?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]);
    }
}
```

Jawab :

```
app > Http > Controllers > BiodataController.php > BiodataController > index
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 = 'Livia Yurike Khuril Maula'; //ubah dengan nama kalian
13         return view('biodata', ['nama' => $nama]);
14     }
15 }
16
```

2

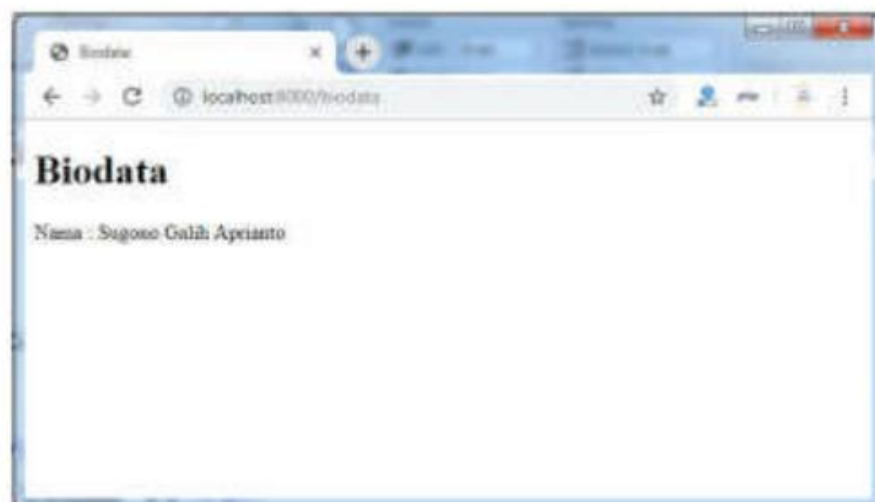
Buat file view dengan nama biodata.blade.php pada **laravelapp\resources\views**

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

Jawab :

```
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 </body>
12
13 </html>
```

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



	<p>Jawab :</p>  <p>The screenshot shows a web browser window with the title 'Biodata'. The address bar displays 'localhost:8000/biodata'. The main content of the page features the word 'Biodata' in a large, bold, black serif font. Below it, the text 'Nama : Livia Yurike Khuril Maula' is displayed in a smaller, black serif font.</p>
--	--

Praktikum – Bagian 5: Memberikan Data Array Kepada View

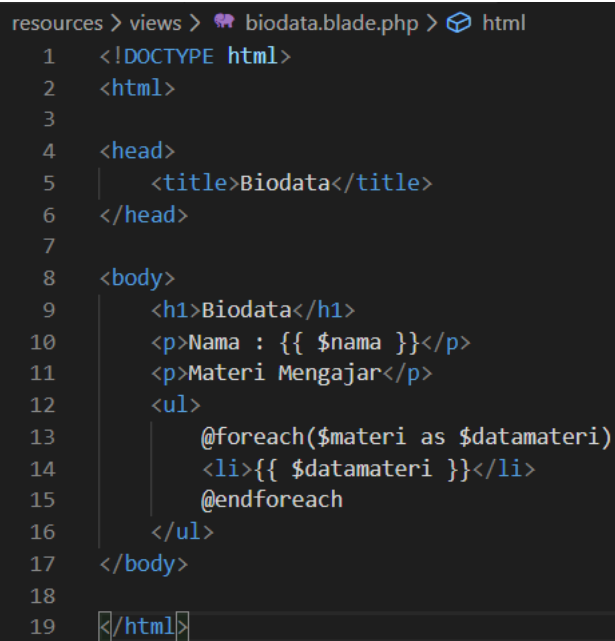
Langkah	Keterangan
1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index() { \$nama = 'Sugono Galih Aprianto'; <i>//ubah dengan nama kalian</i> \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); } }</pre> <p>Jawab :</p>  <p>The screenshot shows a code editor with the following PHP code in BiodataController.php:</p> <pre>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 = 'Livia Yurike Khuril Maula'; <i>//ubah dengan nama kalian</i> 13 \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; 14 return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); 15 } 16 } 17</pre>

2

Tambahkan isi file view **biodata.blade.php** menjadi seperti berikut:

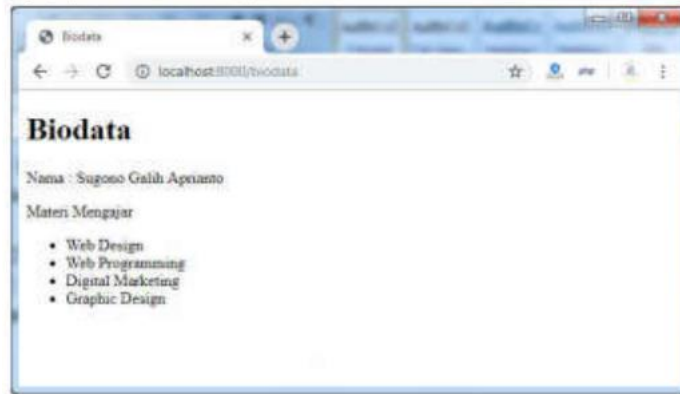
```
<!DOCTYPE html>
<html>
<head>
    <title>Biodata</title>
</head>
<body>
    <h1>Biodata</h1>
    <p>Nama : {{ $nama }}</p>
    <p>Materi Mengajar</p>
    <ul>
        @foreach($materi as $datamateri)
            <li>{{ $datamateri }}</li>
        @endforeach
    </ul>
</body>
</html>
```

Jawab :

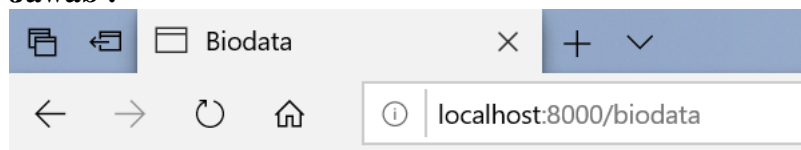


```
resources > views > biodata.blade.php > html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Biodata</title>
6  </head>
7
8  <body>
9      <h1>Biodata</h1>
10     <p>Nama : {{ $nama }}</p>
11     <p>Materi Mengajar</p>
12     <ul>
13         @foreach($materi as $datamateri)
14            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

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



Jawab :



Biodata

Nama : Livia Yurike Khuril Maula

Materi Mengajar

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