



Jurusan Teknologi Informasi Politeknik Negeri Malang

Jobsheet-9: Pengantar dan Instalasi Laravel

Mata Kuliah Pemrograman Web Lanjut

Pengampu: Tim Ajar Pemrograman Web Lanjut

April 2020


Nama : Alfian Noufal Nasruddin

Kelas : TI-2A

NIM : 1841720059

No.absen : 07

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 menginstal, menghapus, dan memperbarui paket PHP. Unduh composer melalui https://getcomposer.org/, pilih Download untuk memulai proses unduh.</p> 

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

C:\Users\Asus FX>composer -v

Composer version 1.9.0 2019-08-02 20:55:32

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v|vv|vvv, --verbose    Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
                          3 for debug

Available commands:
```


- **Dulu saya sudah pernah instal composer bisa dilihat dengan mengetikan composer-v**

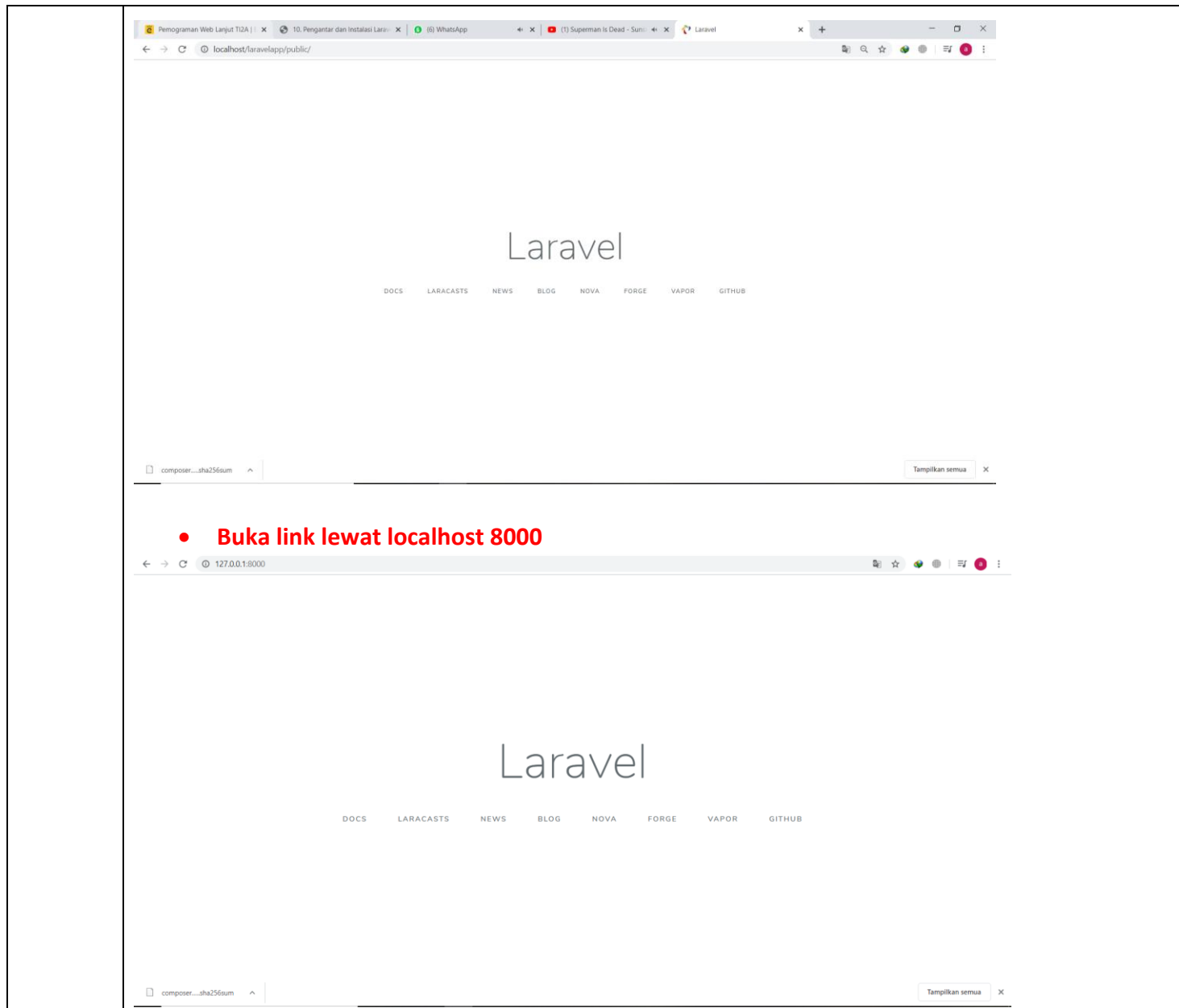
2

Terdapat 2 cara dalam menginstall Laravel,

1. Install Via Laravel Installer

Setelah proses instalasi Composer selesai, buka Command Prompt lalu tuliskan lokasi folder yang akan menampung folder kerja kita. Karena pada praktikum sebelumnya telah

	<p>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> <p>Tunggulah hingga proses penginstalan selesai. Setelah itu, Kita bisa mulai membuat project laravel dengan mengetikkan perintah:</p> <p style="text-align: center;">laravel new laravelapp</p> <p>Keterangan: laravelapp: merupakan folder kerja laravel, Kita dapat menggantinya sesuai dengan keinginan</p> <p>2. Install Via Composer Create-Project</p> <p>Selain menggunakan cara pertama, Kita dapat melakukan penginstalan laravel dengan mengetikkan perintah berikut pada command prompt:</p> <p style="text-align: center;">composer create-project --prefer-dist laravel/laravel laravelapp</p> <pre>C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel laravelapp</pre> <ul style="list-style-type: none"> • Saya menggunakan cara ke dua yaitu dengan create project
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/ 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). Jalankan dengan localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :</p>  <ul style="list-style-type: none"> • buka lewat link localhost



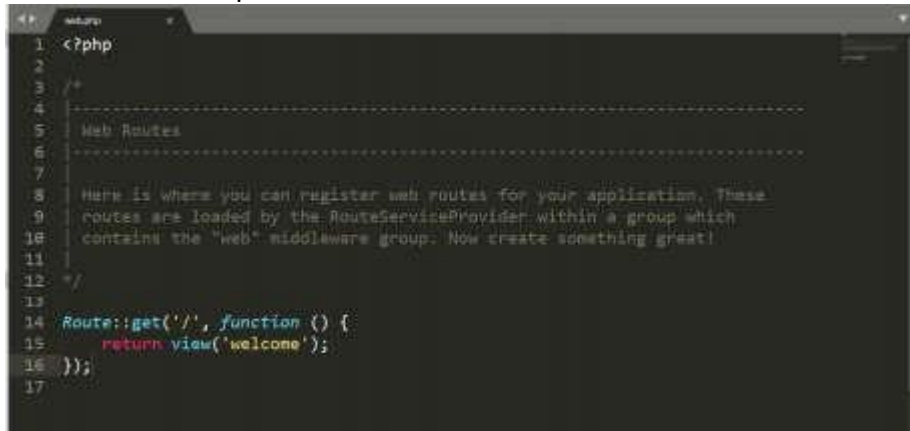
Bagian 2: Mengenal dan membuat route

Langkah	Keterangan
---------	------------

Praktikum –

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 `laravelapp\routes\web.php` Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut



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

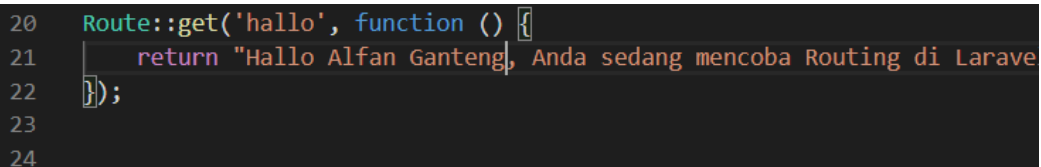
```
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
```

2

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 :

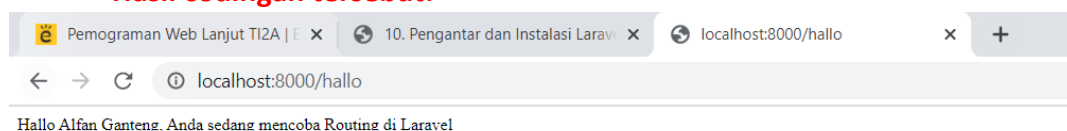
```
Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; });
```

Untuk melihat hasilnya, pada browser, ketikkan alamat <http://localhost:8000/hallo>



```
20 Route::get('hallo', function () {
21     return "Hallo Alfian Ganteng, Anda sedang mencoba Routing di Larave
22 });
23
24
```

- **Hasil codingan tersebut:**



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'); });
```

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

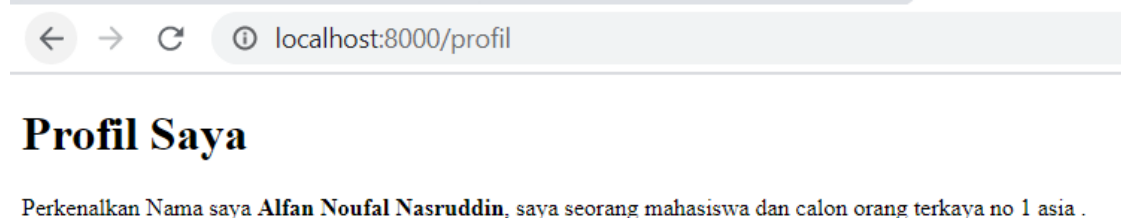
Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:

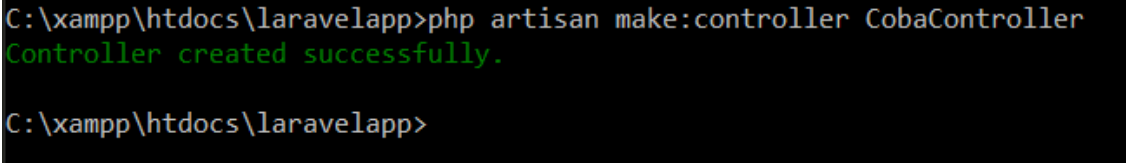
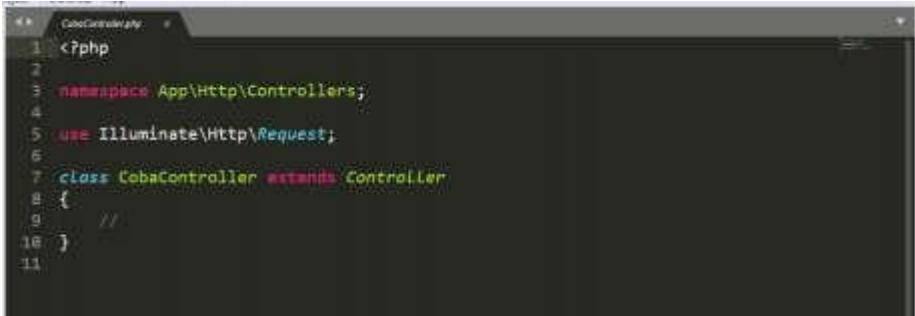
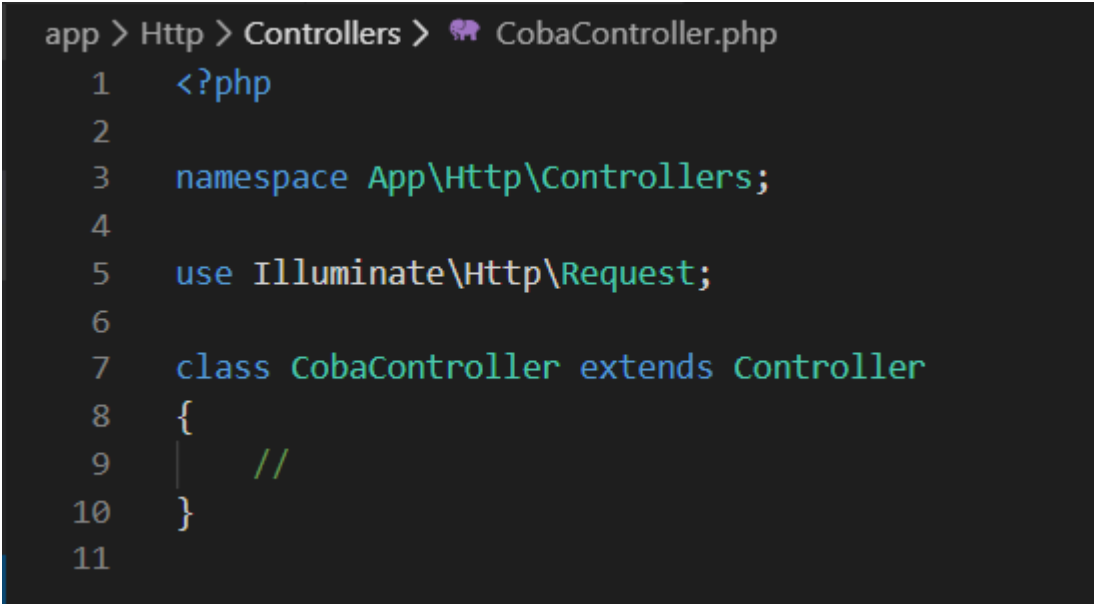
```
<!DOCTYPE html>
<html>
<head>
    <title>Profil Saya</title>
</head>
<body>
    <h1>Profil Saya</h1>
    <p>Perkenalkan Nama saya nama_saya, saya seorang .....
.</p>
</body>
</html>
```

```
resources > views > profil.blade.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Profil Saya</title>
5 </head>
6 <body>
7     <h1>Profil Saya</h1>
8     <p>Perkenalkan Nama saya <b>Alfan Noufal Nasruddin</b>, saya seorang mahasiswa dan calon orang terkaya no 1 asia
9 .</p>
10 </body>
11 </html>
12
```

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

• **Hasil:**



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> <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>php artisan make:controller CobaController</p> <ul style="list-style-type: none"> • Membuat controller dengan artisan  <p>Maka akan terbentuk sebuah file dengan nama CobaController.php pada folder laravelapp\app\Http\Controllers\</p>  

2

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

```
Route::get('profil', 'CobaController@profil');  
24 Route::get('profil', 'CobaController@profil');|  
25  
26
```

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

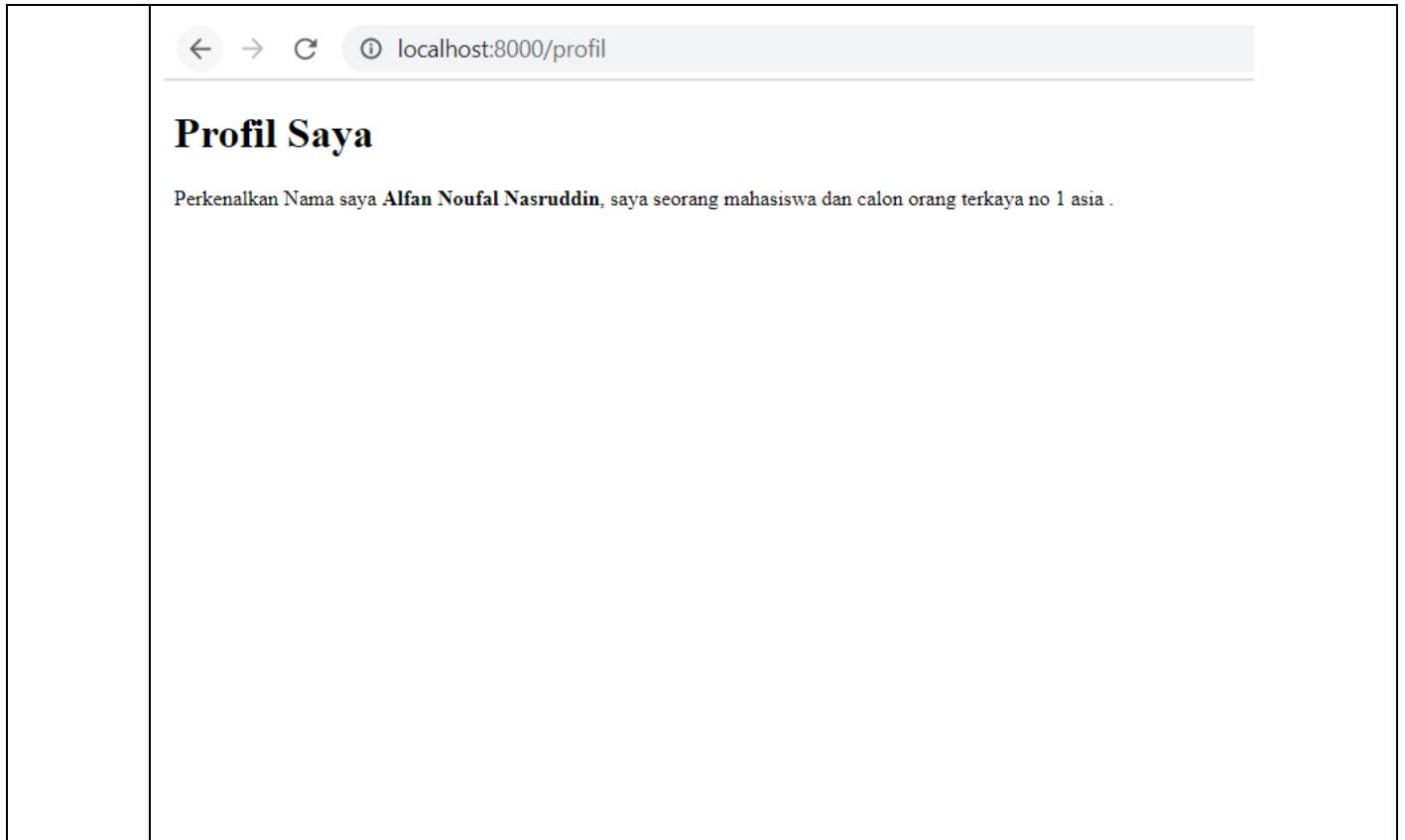
```
<?php  
namespace App\Http\Controllers; use  
Illuminate\Http\Request;  
  
class CobaController extends Controller  
{  
    public function profil()  
    {  
        return view('profil');  
    }  
}
```

- **Mengetik code**

```
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    public function profil()  
11    {  
12        return view('profil');  
13    }  
14 }  
15
```

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

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');
```

- **Membuat route baru**

```
26 Route::get('biodata', 'BiodataController@index');
27
28
```

Buat controller baru dengan nama **BiodataController.php**
php artisan make:controller BiodataController

- **Buat controller**

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

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]);
        }
    }
```

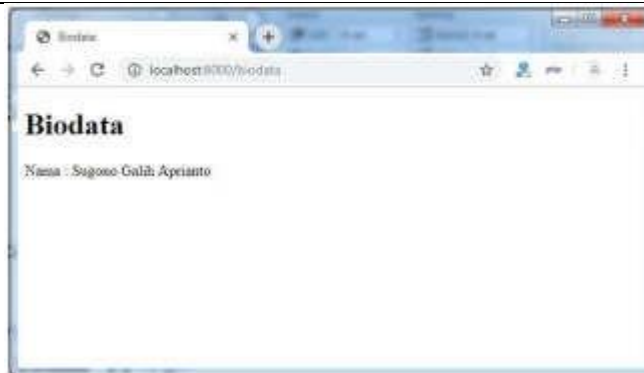
- **Membuat fungsi pada biodatacontroller**

Praktikum –

	<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 \$nama = 'Alfan Noufal Nasruddin'; //ubah dengan nama kalian 12 return view('biodata' , ['nama' => \$nama]); 13 } 14 15 } 16</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>

```
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  </body>
10 </html>
11 |
```

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



← → ↻ ⓘ localhost:8000/biodata

Biodata

Nama : Alfian Noufal Nasruddin

Praktikum – Bagian 5: Memberikan Data Array Kepada View

Langkah	Keterangan
---------	------------

Praktikum –

1

Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada **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
            $materi = ["Web Design", "Web Programming", "Digital
Marketing","Graphic Design"];
            return view('biodata' , ['nama' => $nama, 'materi' =>
                $materi]);
        }
    }
```

```
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         $nama = 'Alfan Noufal Nasruddin'; //ubah dengan nama kalian
12         $materi = ["Web Design", "Web Programming", "Digital Marketing","Graphic Design"];
13         return view('biodata' , ['nama' => $nama, 'materi' => $materi]);
14     }
15 }
16
17 }
18
```

2

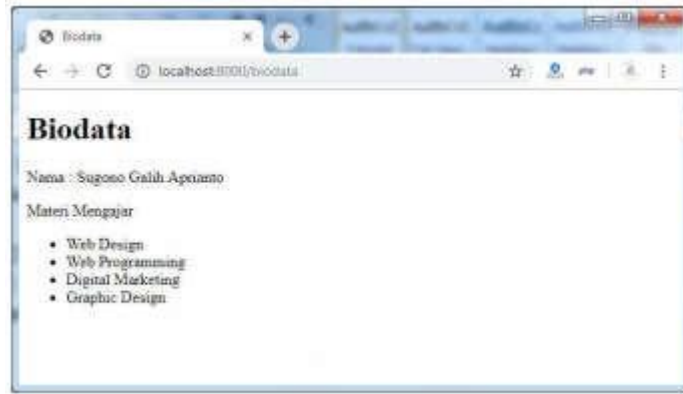
Tambahkan isi file view **biodata.blade.php** menjadi seperti berikut:

```
<!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>
```

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 : {{ $nama }}</p>
9      <p>Materi Mengajar</p>
10     <ul>
11         @foreach($materi as $datamateri)
12             <li>{{ $datamateri }}</li>
13         @endforeach
14     </ul>
15 </body>
16 </html>
17
```

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



← → ↻ ⓘ localhost:8000/biodata

Biodata

Nama : Alfian Noufal Nasruddin

Materi Mengajar

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

-- Selamat Mengerjakan --