

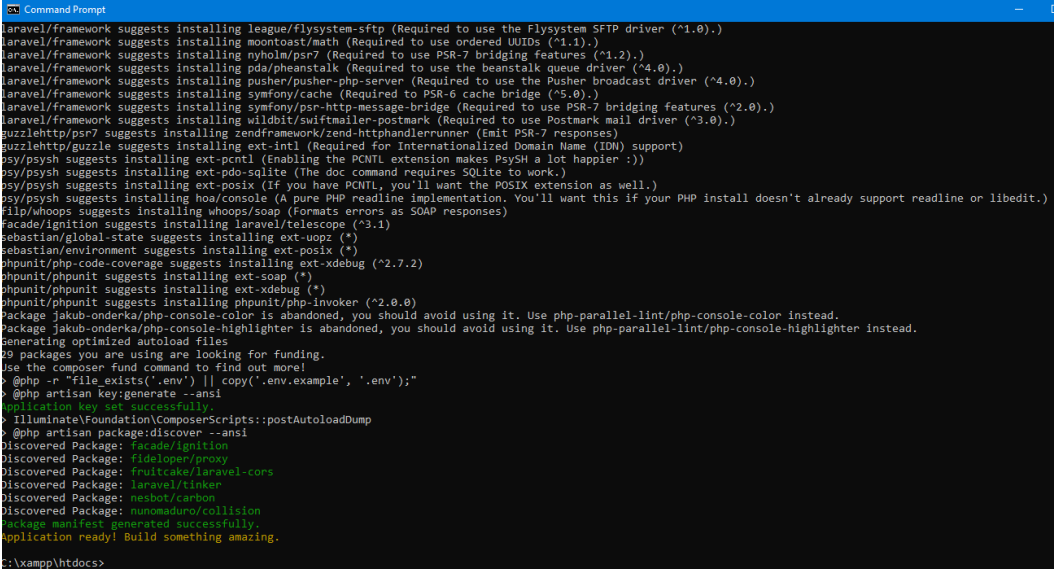
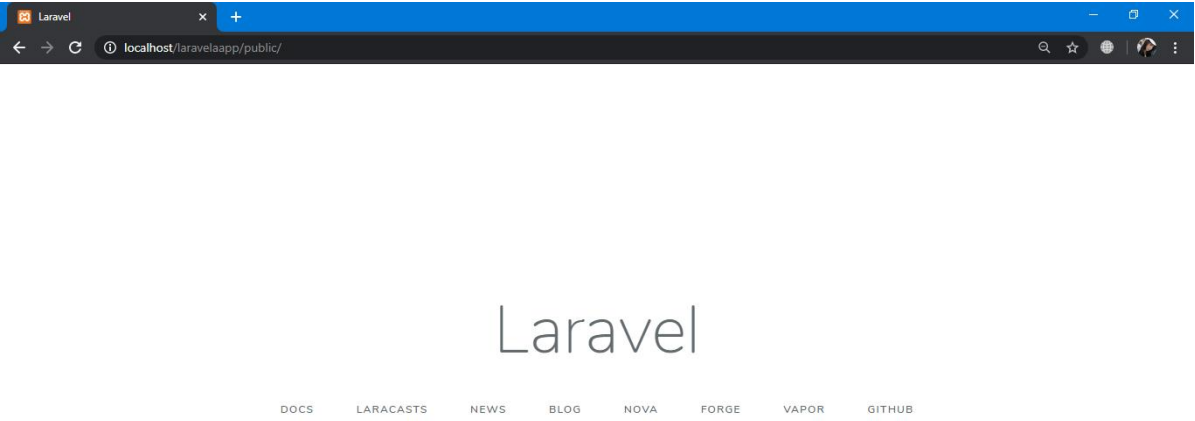


NAMA : QORINDA YULVARISMA
KELAS : TI-2A
NIM : 1841720084

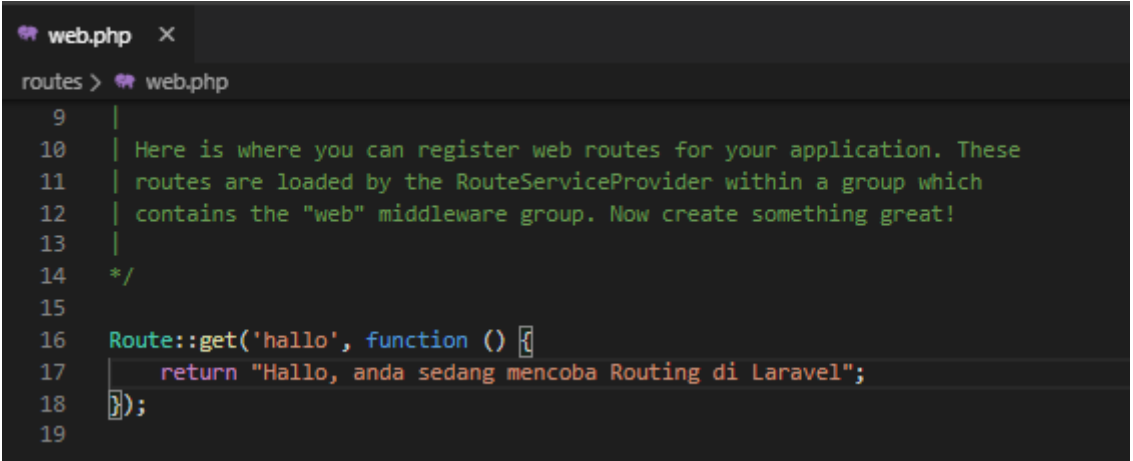
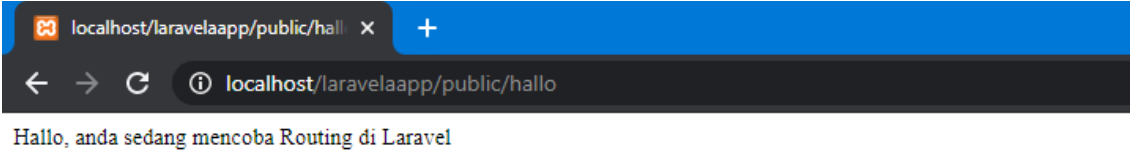
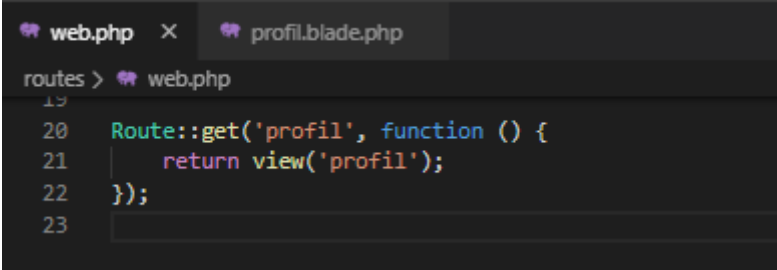
LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT

Praktikum – Bagian 1 : Instalasi Laravel di Windows

Langkah	Keterangan
1	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.
2	<p>Terdapat 2 cara dalam menginstall Laravel,</p> <p>1. Install Via Laravel Installer</p> <p>Setelah proses instalasi Composer selesai, buka Command Prompt lalu tujuhan 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>composer global require "laravel/installer"</p> <pre>cmd Command Prompt Microsoft Windows [Version 10.0.18363.720] (c) 2019 Microsoft Corporation. All rights reserved. C:\Users\TOSHIBA>cd C:\xampp\htdocs C:\xampp\htdocs>composer global require "laravel/installer" Changed current directory to C:/Users/TOSHIBA/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): 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 6 packages you are using are looking for funding. Use the composer fund command to find out more! C:\xampp\htdocs></pre>

	<p>Tunggulah hingga proses penginstalan selesai. Setelah itu, Kita bisa mulai membuat project laravel dengan mengetikkan perintah:</p> <p>laravel new laravelapp</p> <p>Keterangan:</p> <p>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>composer create-project --prefer-dist laravel/laravel laravelapp</p>
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 <code>http://localhost/laravelaapp/public/</code> Atau Kita dapat menggunakan perintah berikut di Command Prompt:</p> <p>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 <code>localhost:8000</code>, dan pastikan tampilan di browser akan seperti berikut ini :</p> 

Praktikum – Bagian 2: Mengenal dan membuat route

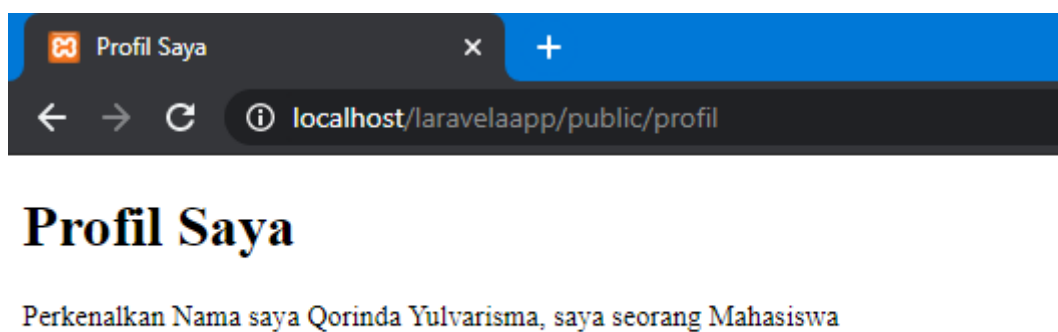
Langkah	Keterangan
1	Setiap request yang datang pada laravel akan diarahkan melalui sebuah route. Route ini yang akan menentukan respon apa yang akan dikerjakan untuk membalas request tersebut. Halaman homepage default Laravel seperti yang ditampilkan pada gambar di Bagian 1 adalah hasil dari script yang terletak pada file laravelapp\routes\web.php Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut
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>  <pre>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('hallo', function () { 17 return "Hallo, anda sedang mencoba Routing di Laravel"; 18 }); 19 </pre> <p>Untuk melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</p>  <p>localhost/laravelaapp/public/hallo</p> <p>Hallo, 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> <p>Route::get('profil', function () { return view('profil'); });</p>  <pre>19 20 Route::get('profil', function () { 21 return view('profil'); 22 }); 23 </pre> <p>Buat file baru bernama profil.blade.php di dalam folder laravelapp\routes\views\, ketikkan script berikut ini:</p>

```

web.php  profil.blade.php X
resources > views > profil.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Profil Saya</title>
5  </head>
6  <body>
7  |   <h1>Profil Saya</h1>
8  |   <p>Perkenalkan Nama saya Qorinda Yulvarisma, saya seorang Mahasiswa</p>
9  </body>
10 </html>
11

```

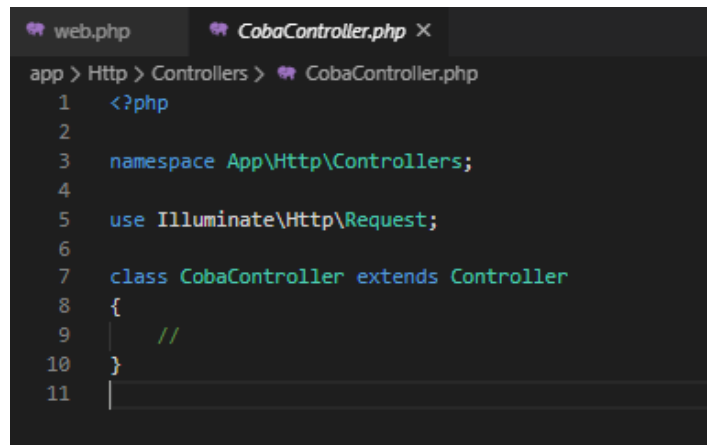
Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat <http://localhost:8000/profil>



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> <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> <pre> C:\Windows\System32\cmd.exe Microsoft Windows [Version 10.0.18363.720] (c) 2019 Microsoft Corporation. All rights reserved. C:\xampp\htdocs\laravelaapp>php artisan make:controller CobaController Controller created successfully. C:\xampp\htdocs\laravelaapp> </pre>

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



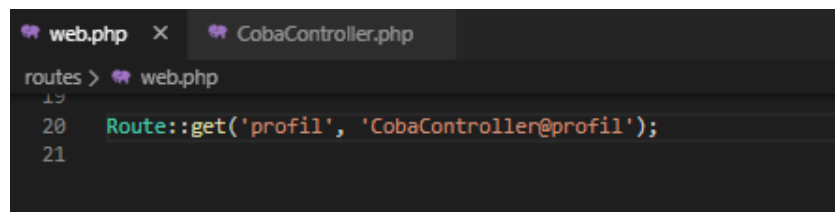
The screenshot shows a code editor with two tabs: 'web.php' and 'CobaController.php'. The 'CobaController.php' tab is active, showing the following code:

```
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 }
11
```

2

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

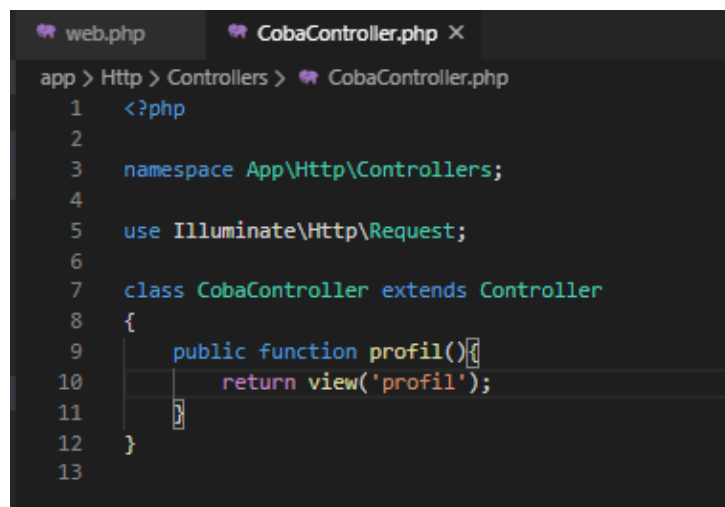
Route::get('profil', 'CobaController@profil');



The screenshot shows a code editor with two tabs: 'web.php' and 'CobaController.php'. The 'web.php' tab is active, showing the following code:

```
routes > web.php
19
20 Route::get('profil', 'CobaController@profil');
21
```

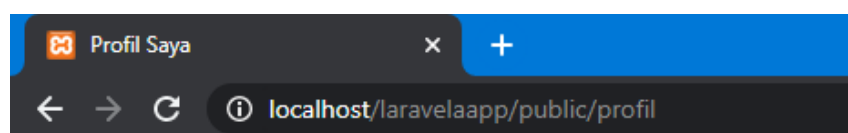
Langkah selanjutnya, tambahkan fungsi / method profil() pada class CobaController seperti berikut ini:



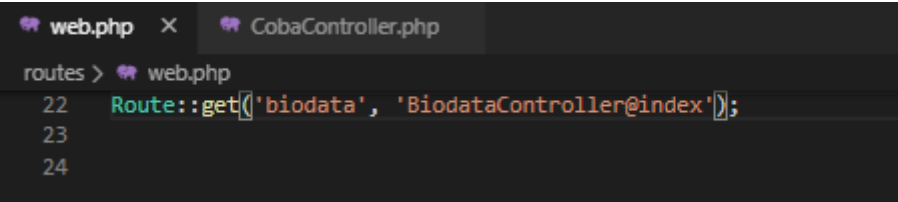
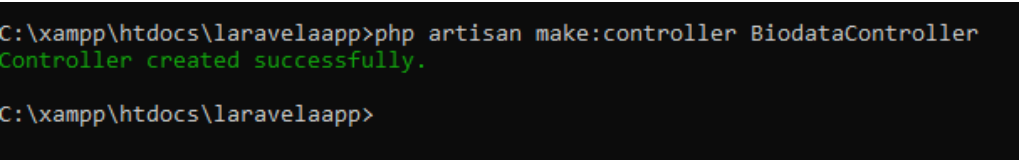

The screenshot shows a code editor with two tabs: 'web.php' and 'CobaController.php'. The 'CobaController.php' tab is active, showing the following code:

```
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      public function profil()
10     {
11         return view('profil');
12     }
13 }
```

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

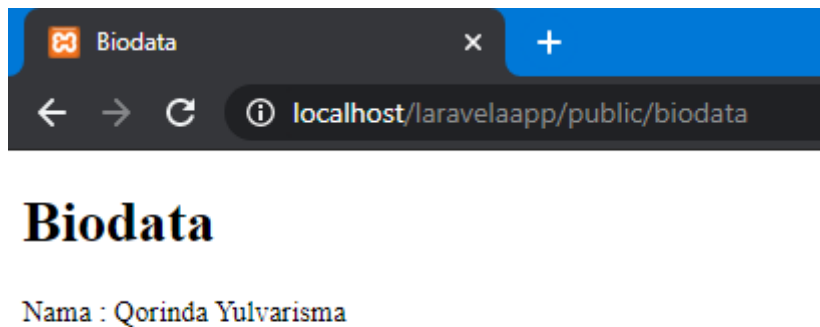
Langkah	Keterangan
1	<p>Dalam sebuah aplikasi, sangat jarang Kita temukan isi dari file view yang bersifat statis. Pada umumnya, view dipakai untuk menampilkan data, contohnya adalah data yang berasal dari database. Namun, karena kita belum sampai pada pembahasan database, maka pada latihan kali ini kita akan mencoba menampilkan data dari variabel.</p> <p>Buat route baru untuk halaman biodata</p> <p>Route::get('biodata', 'BiodataController@index');</p>  <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> 
2	Buat file view dengan nama biodata.blade.php pada laravelapp\resources\views\

```

web.php  BiodataController.php  biodata.blade.php X
resources > views > biodata.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <title>Biodata</title>
5  </head>
6  <body>
7  |   <h1>Biodata</h1>
8  |   <p>Nama : {{ $nama }}</p>
9  |
10 |
11 </body>
    </html>

```

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



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> web.php BiodataController.php X biodata.blade.php 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 public function index(){ 10 \$nama = 'Qorinda Yulvarisma'; 11 \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; 12 return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); 13 } 14 } 15 </pre>
2	Tambahkan isi file view biodata.blade.php menjadi seperti berikut:

```
web.php  BiodataController.php  biodata.blade.php x
resources > views > biodata.blade.php
1  <!DOCTYPE html>
2  <html lang="en">
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
16 </body>
17 </html>
```

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

