

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

**Mata Kuliah Pemrograman Web Lanjut**



**Oleh :**

**ALAN PERDHANA TIMOR / NIM : 1841720187**

**KELAS TI 2A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

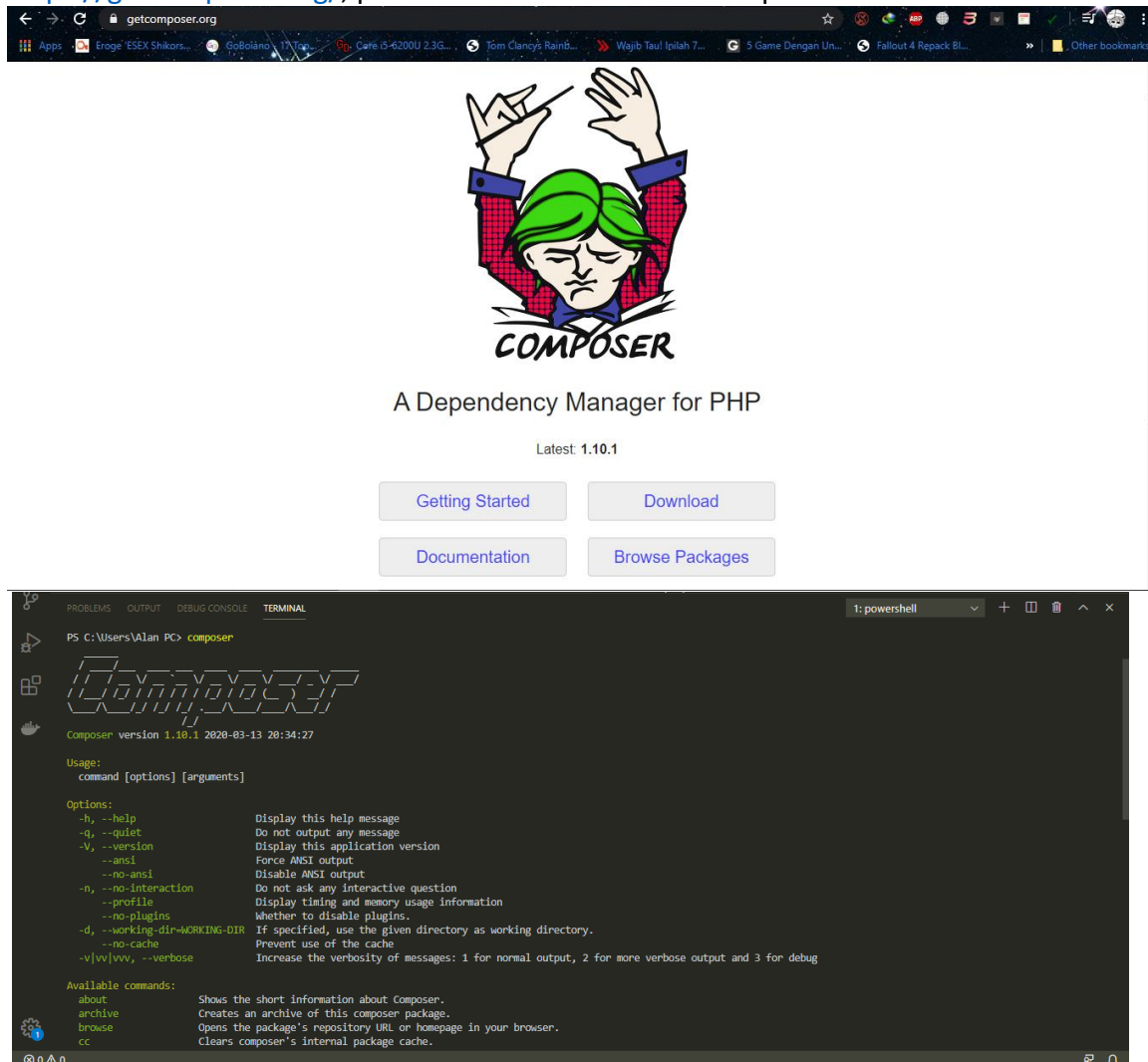
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2020**

## Praktikum – Bagian 1: Instalasi Laravel di Windows

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. Terdapat 2 cara dalam menginstall Laravel,  
**1. Install Via Laravel Installer**  
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: `cd C:\xampp\htdocs`. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:

**`composer global require "laravel/installer"`**

```
Administrator: Command Prompt

C:\xampp\htdocs> composer global require "laravel/installer"
Changed current directory to C:/Users/Alan PC/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-httpdierrunner (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>
C:\xampp\htdocs>jjj
```

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

**laravel new laravelapp**

```
Administrator: Command Prompt

[... ] [<packages>]...

C:\xampp\htdocs>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/inflexor (1.3.1): Loading from cache
- Installing doctrine/lexer (1.2.0): Loading from cache
- Installing dragonmantank/cron-expression (v2.3.0): Loading from cache
- Installing voku/portable-ascii (1.4.10): Loading from cache
- Installing symfony/polyfill-ctype (v1.15.0): Loading from cache
- Installing phpoption/phpoption (1.7.3): Loading from cache
- Installing vlucas/phpdotenv (v4.1.3): Loading from cache
- Installing symfony/css-selector (v5.0.7): 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 (v5.0.7): Loading from cache
- Installing symfony/routing (v5.0.7): Loading from cache
- Installing symfony/process (v5.0.7): Loading from cache
- Installing symfony/polyfill-php72 (v1.15.0): Loading from cache
- Installing symfony/polyfill-intl-idn (v1.15.0): Loading from cache
- Installing symfony/mime (v5.0.7): Loading from cache
- Installing symfony/polyfill-php73 (v1.15.0): Loading from cache
- Installing symfony/http-foundation (v5.0.7): Downloading (failed)
Download (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
```

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**

- |   |  |
|---|--|
| 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 <a href="http://localhost/laravelapp/public/">http://localhost/laravelapp/public/</a> Atau Kita dapat menggunakan perintah berikut di Command Prompt: |
|---|--|

## php artisan serve

```
PS C:\xampp\htdocs\laravelapp> php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Tue Apr  7 09:10:41 2020] 127.0.0.1:60516 [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

- 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.
 

```
web.php
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6   * Web Routes
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  Route::get('/', function () {
14      return view('welcome');
15  });
```
- 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"; });
```



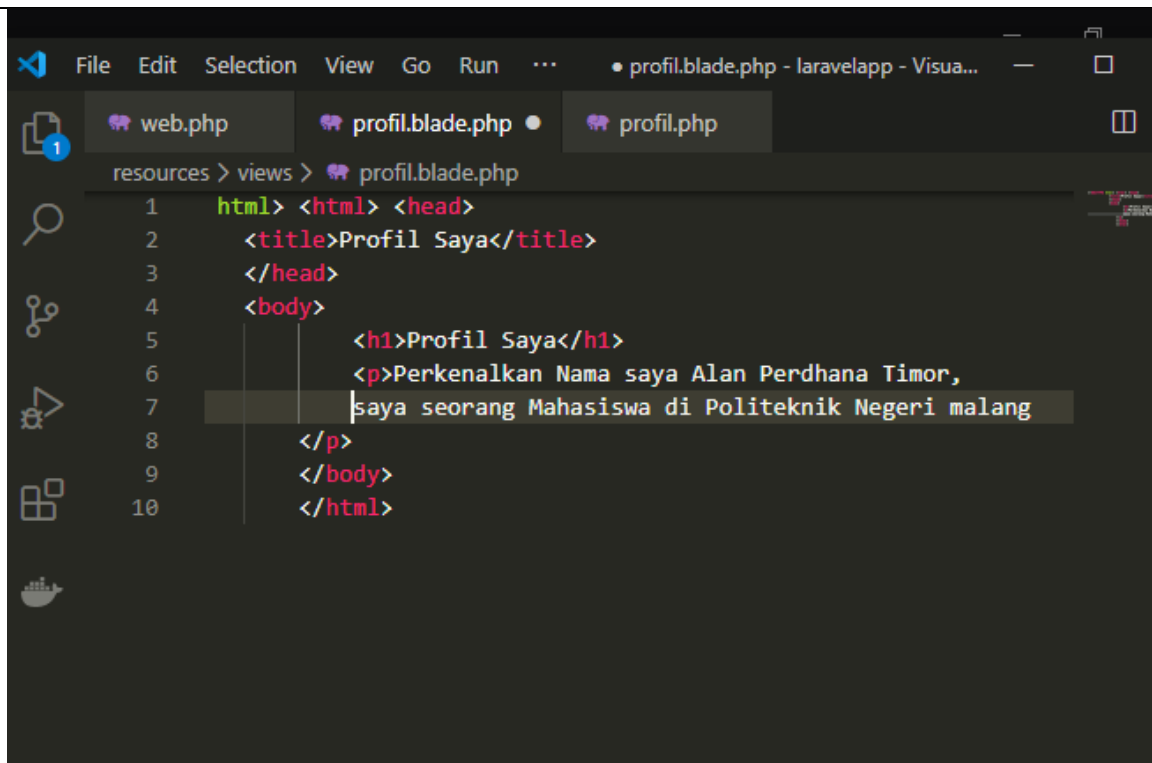
---

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

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

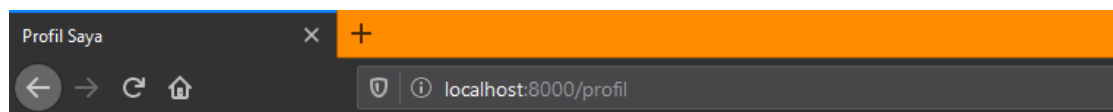
```
Route::get('profil', function () { return view('profil'); });
```

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



```
1  <html> <html> <head>
2    <title>Profil Saya</title>
3  </head>
4  <body>
5    <h1>Profil Saya</h1>
6    <p>Perkenalkan Nama saya Alan Perdhana Timor,
7    saya seorang Mahasiswa di Politeknik Negeri malang
8  </p>
9  </body>
10 </html>
```

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



## Profil Saya

Perkenalkan Nama saya Alan Perdhana Timor, saya seorang Mahasiswa di Politeknik Negeri malang

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

- 1 ada 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. 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. 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

```
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\**

```
File Edit Selection View Go Run ... CobaController.php - laravelapp - Vis...
web.php profil.blade.php CobaController.php x profil.php
app > Http > Controllers > CobaController.php > ...
1 k?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class CobaController extends Controller
8 {
9     //
10 }
11
```

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

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

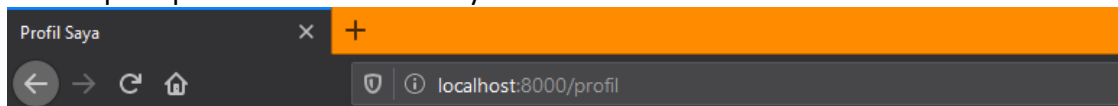
```
Route::get('profil', 'CobaController@profil');
```

```
web.php x profil.blade.php CobaController.php profil.php
routes > web.php
9 |
10 | Here is where you can register web routes for your application.
11 | routes are loaded by the RouteServiceProvider within a group
12 | contains the "web" middleware group. Now create something great.
13 |
14 | */
15 |
16 | Route::get('/', function () {
17 |     return view('welcome');
18 | });
19 |
20 | Route::get('hallo', function () {
21 |     return "Hallo, Anda sedang mencoba Routing di Laravel";
22 | });
23 |
24 | Route::get('profil', 'CobaController@profil');
```

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

```
web.php profil.blade.php CobaController.php profil.php
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         return view('profil');
11     }
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 Alan Perdhana Timor, saya seorang Mahasiswa di Politeknik Negeri malang

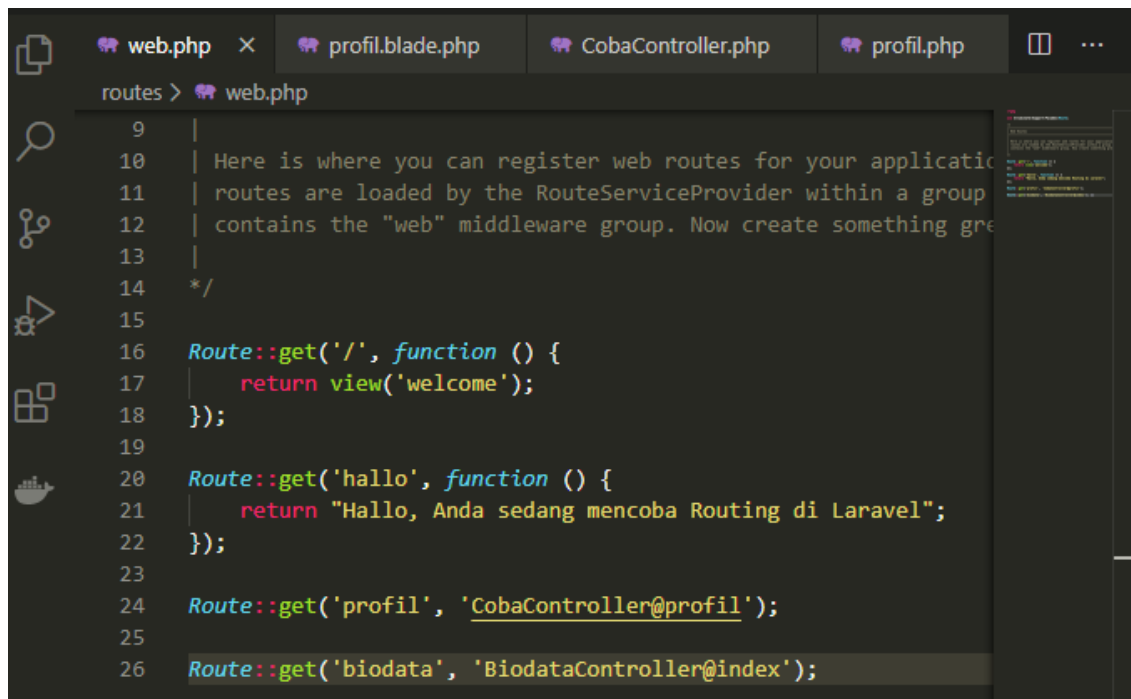


## Praktikum – Bagian 4: Memberikan Data Controller kepada View

- 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

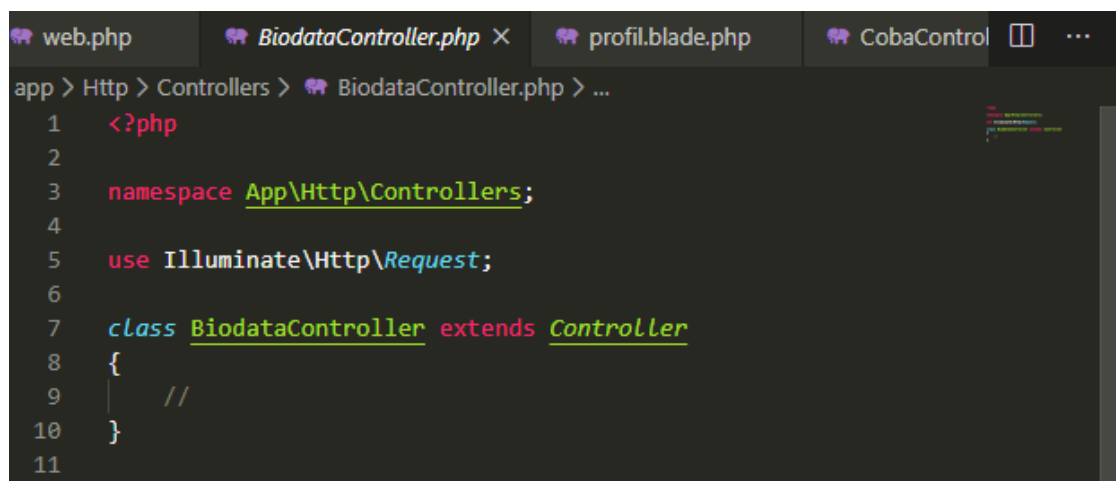
```
Route::get('biodata', 'BiodataController@index');
```



```
web.php x profil.blade.php CobaController.php profil.php ...
routes > web.php
9
10 | Here is where you can register web routes for your application.
11 | routes are loaded by the RouteServiceProvider within a group
12 | contains the "web" middleware group. Now create something great.
13 |
14 | */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba Routing di Laravel";
22 });
23
24 Route::get('profil', 'CobaController@profil');
25
26 Route::get('biodata', 'BiodataController@index');
```

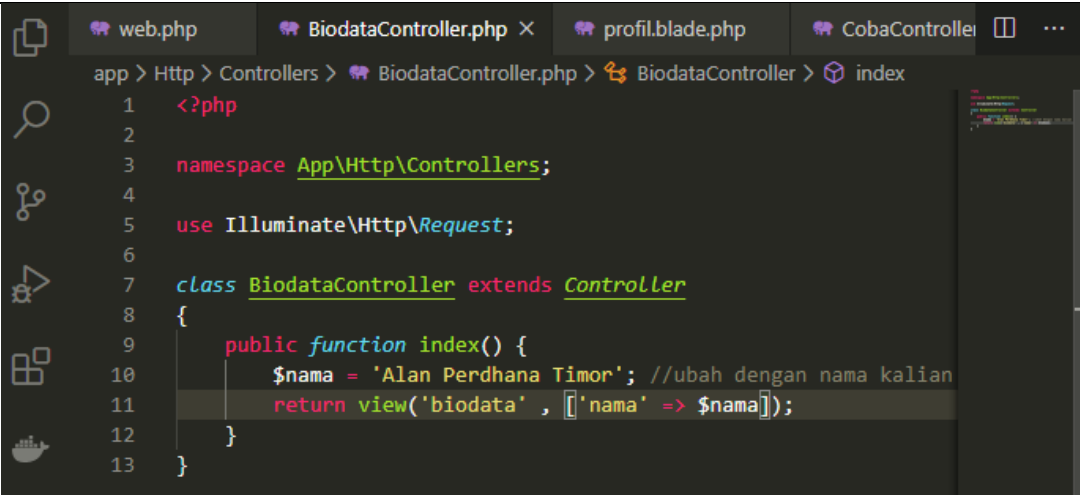
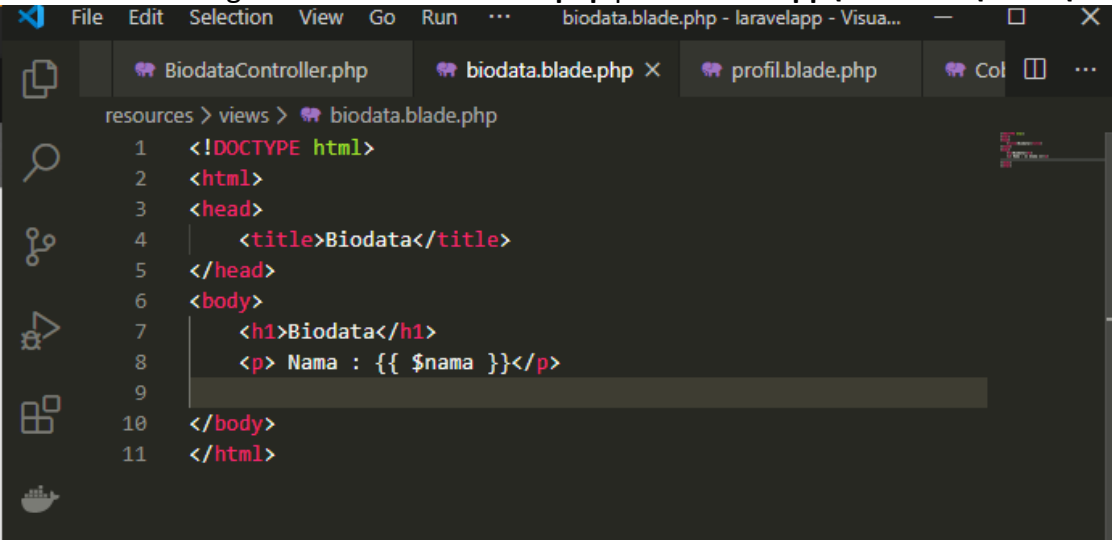
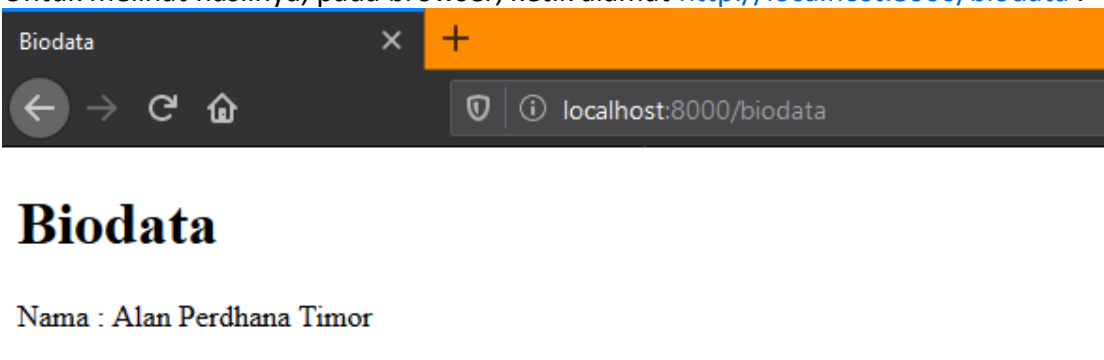
Buat controller baru dengan nama **BiodataController.php**

```
php artisan make:controller BiodataController
```

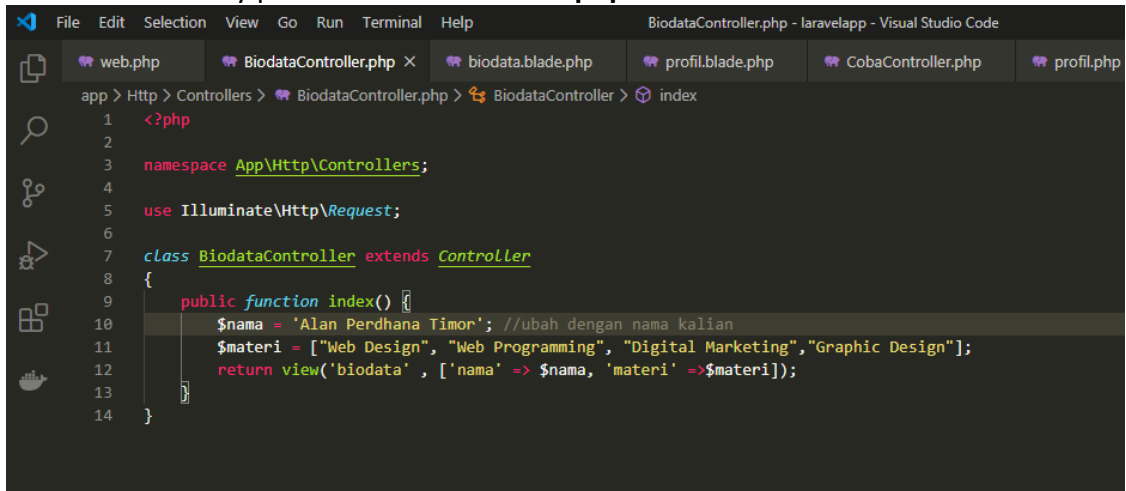


```
web.php BiodataController.php x profil.blade.php CobaControl ...
app > Http > Controllers > BiodataController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class BiodataController extends Controller
8 {
9     //
10 }
11
```

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

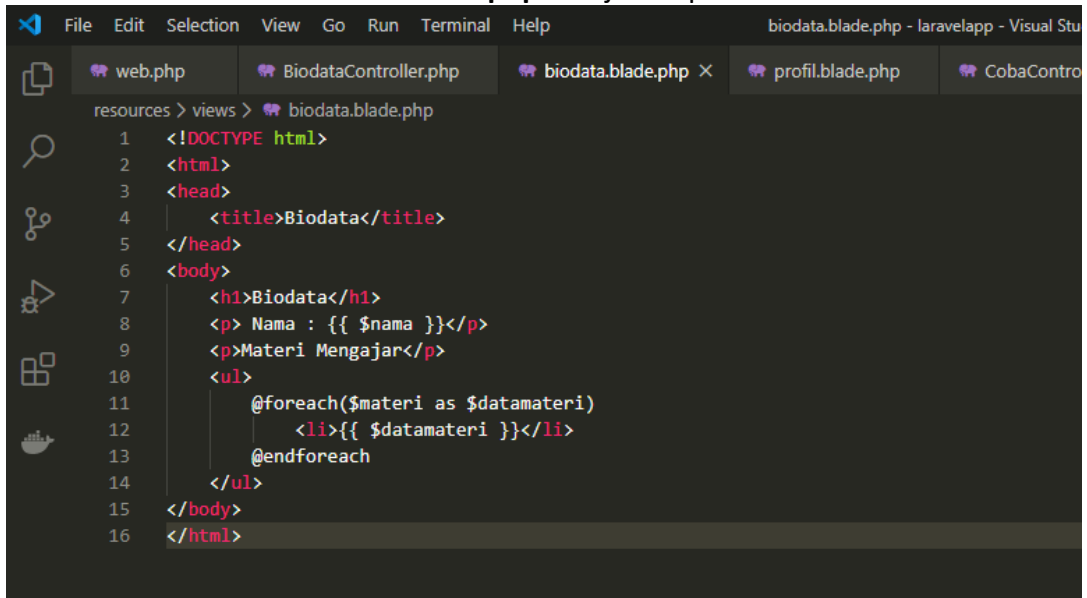
	 <pre>1 &lt;?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class BiodataController extends Controller 8 { 9     public function index() { 10         \$nama = 'Alan Perdhana Timor'; //ubah dengan nama kalian 11         return view('biodata', ['nama' =&gt; \$nama]); 12     } 13 }</pre>
2	<p>Buat file view dengan nama <b>biodata.blade.php</b> pada <b>laravelapp/resources/views/</b></p>  <pre>1 &lt;!DOCTYPE html&gt; 2 &lt;html&gt; 3 &lt;head&gt; 4     &lt;title&gt;Biodata&lt;/title&gt; 5 &lt;/head&gt; 6 &lt;body&gt; 7     &lt;h1&gt;Biodata&lt;/h1&gt; 8     &lt;p&gt; Nama : {{ \$nama }}&lt;/p&gt; 9 10 &lt;/body&gt; 11 &lt;/html&gt;</pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat <a href="http://localhost:8000/biodata">http://localhost:8000/biodata</a> .</p>  <p><b>Biodata</b></p> <p>Nama : Alan Perdhana Timor</p>

- 1 Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variable data array pada **BiodataController.php**



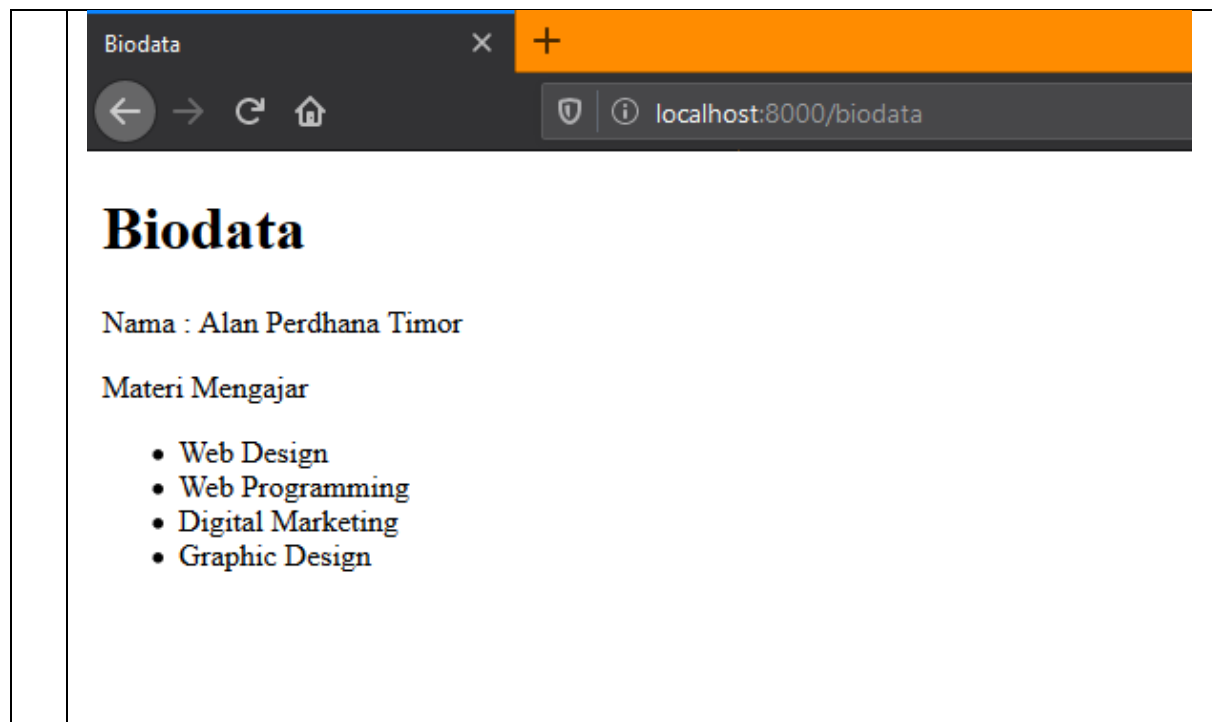
```
File Edit Selection View Go Run Terminal Help BiodataController.php - laravelapp - Visual Studio Code
web.php BiodataController.php x biodata.blade.php profil.blade.php CobaController.php profil.php
app > Http > Controllers > BiodataController.php > BiodataController > index
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class BiodataController extends Controller
8 {
9     public function index() {
10         $nama = 'Alan Perdana Timor'; //ubah dengan nama kalian
11         $materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"];
12         return view('biodata', ['nama' => $nama, 'materi' => $materi]);
13     }
14 }
```

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



```
File Edit Selection View Go Run Terminal Help biodata.blade.php - laravelapp - Visual Studio
web.php BiodataController.php biodata.blade.php x profil.blade.php CobaContro
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>
```

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



Biodata



localhost:8000/biodata

# Biodata

Nama : Alan Perdhana Timor

Materi Mengajar

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