




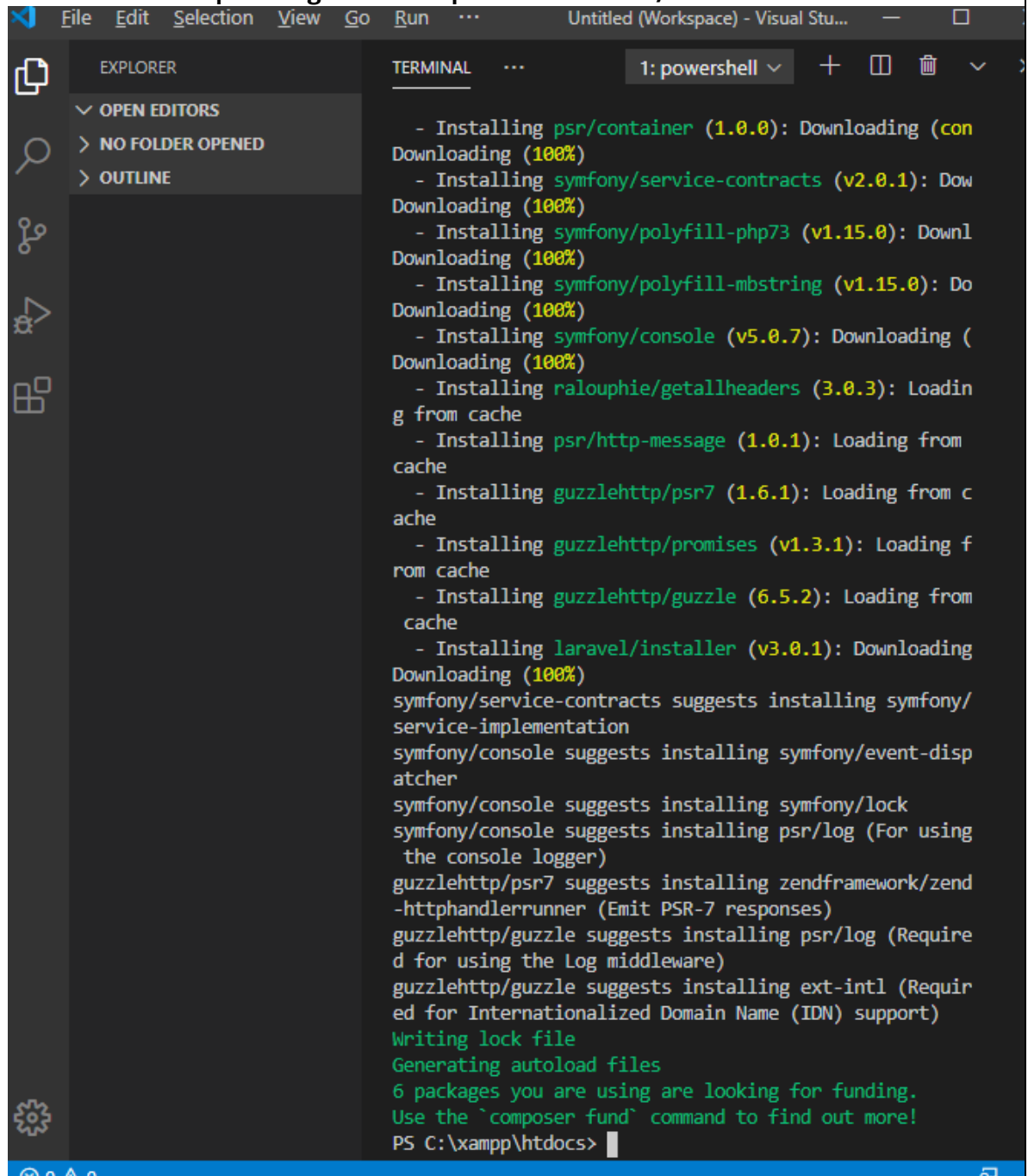
**Nama : Agsal Fairrohmad A.P**  
**No : 04/TI2A**

**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 <a href="https://getcomposer.org/">https://getcomposer.org/</a>, pilih Download untuk memulai proses unduh.</p> 
2	<p>Terdapat 2 cara dalam menginstall Laravel, <b>1. 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</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:

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



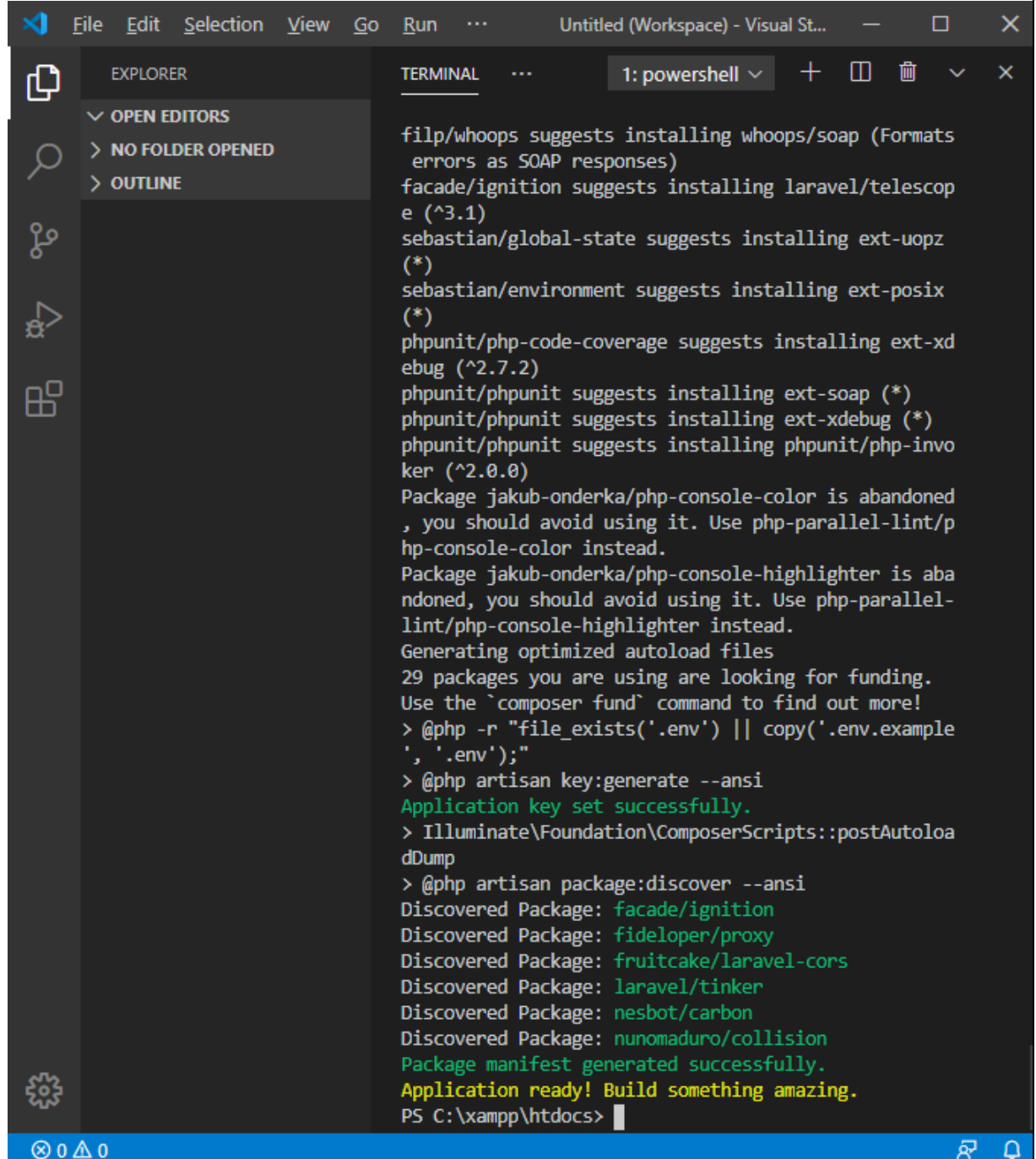
```
File Edit Selection View Go Run ... Untitled (Workspace) - Visual Stu...
EXPLORER
  OPEN EDITORS
  NO FOLDER OPENED
  OUTLINE
TERMINAL
  1: powershell
- Installing psr/container (1.0.0): Downloading (con
Downloading (100%)
- Installing symfony/service-contracts (v2.0.1): Dow
Downloading (100%)
- Installing symfony/polyfill-php73 (v1.15.0): Downl
Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Do
Downloading (100%)
- Installing symfony/console (v5.0.7): Downloading (
Downloading (100%)
- Installing ralouphie/getallheaders (3.0.3): Loadin
g from cache
- Installing psr/http-message (1.0.1): Loading from
cache
- Installing guzzlehttp/psr7 (1.6.1): Loading from c
ache
- Installing guzzlehttp/promises (v1.3.1): Loading f
rom cache
- Installing guzzlehttp/guzzle (6.5.2): Loading from
cache
- Installing laravel/installer (v3.0.1): Downloading
Downloading (100%)
symfony/service-contracts suggests installing symfony/
service-implementation
symfony/console suggests installing symfony/event-disp
atcher
symfony/console suggests installing symfony/lock
symfony/console suggests installing psr/log (For using
the console logger)
guzzlehttp/psr7 suggests installing zendframework/zend
-httphandlerunner (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Require
d for using the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Require
d 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!
PS C:\xampp\htdocs>
```

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

The image shows a screenshot of the Visual Studio Code editor. The top menu bar includes File, Edit, Selection, View, Go, Run, and other options. The Explorer sidebar on the left shows 'OPEN EDITORS' with 'NO FOLDER OPENED' and 'OUTLINE'. The main area is a PowerShell terminal window titled '1: powershell'. It displays the output of a Laravel installation command, including suggestions for installing various packages like whoops/soap, facade/ignition, and laravel/telescope. It also shows the generation of autoload files and the discovery of installed packages like facade/ignition, fideloper/proxy, and laravel/tinker. The terminal ends with 'Application ready! Build something amazing.' and the prompt 'PS C:\xampp\htdocs>'.

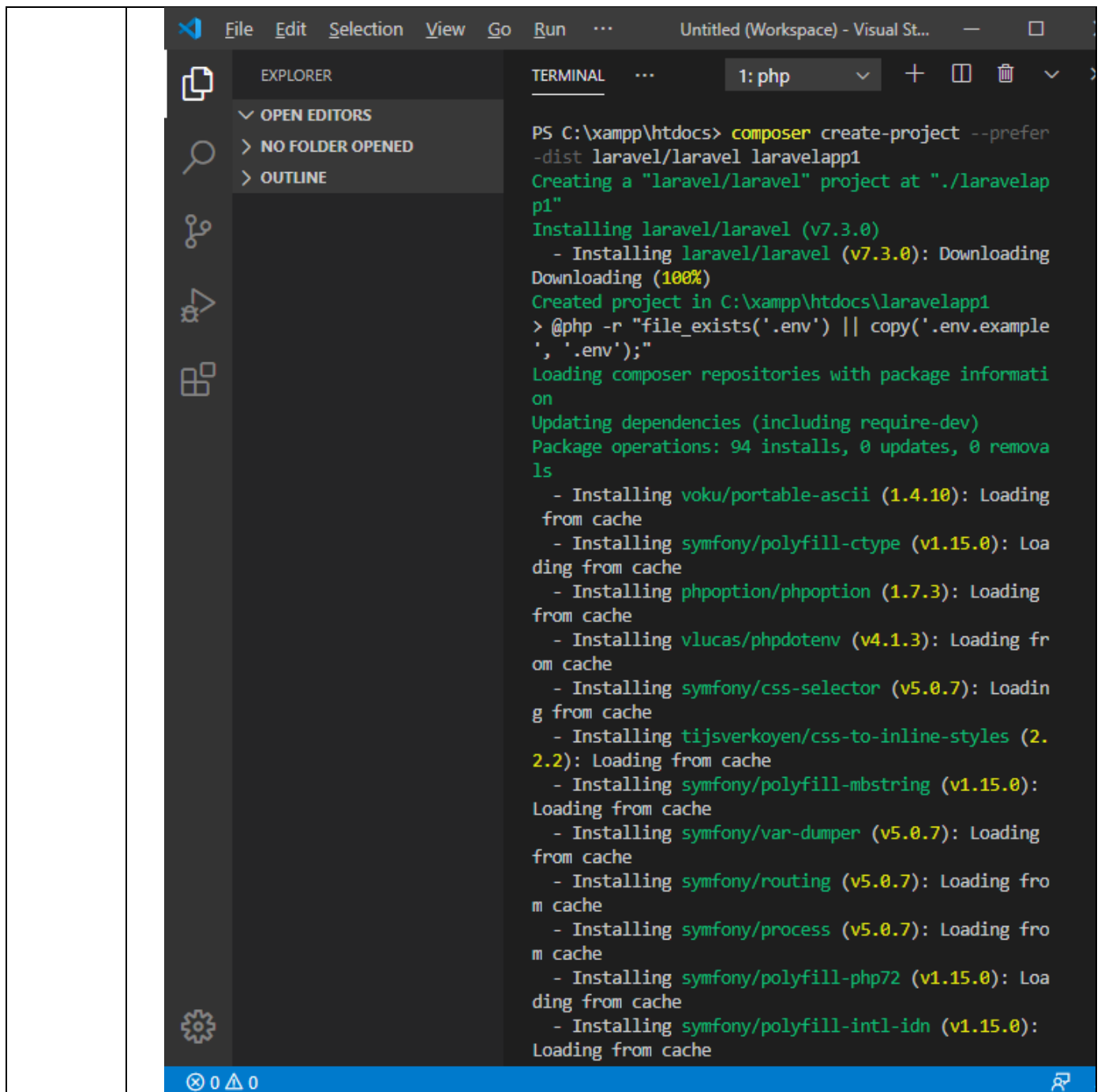
```
File Edit Selection View Go Run ... Untitled (Workspace) - Visual St...
EXPLORER
  OPEN EDITORS
    NO FOLDER OPENED
    OUTLINE
  SEARCH
  SOURCE CONTROL
  RUN AND DEBUG
  SETTINGS
  SETTINGS
  0 0 0

TERMINAL 1: powershell
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
facade/ignition suggests installing laravel/telescope (^3.1)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.7.2)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0.0)
Package jakub-onderka/php-console-color is abandoned, you should avoid using it. Use php-parallel-lint/php-console-color instead.
Package jakub-onderka/php-console-highlighter is abandoned, you should avoid using it. Use php-parallel-lint/php-console-highlighter instead.
Generating optimized autoload files
29 packages you are using are looking for funding.
Use the `composer fund` command to find out more!
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
> @php artisan key:generate --ansi
Application key set successfully.
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Discovered Package: facade/ignition
Discovered Package: fideloper/proxy
Discovered Package: fruitcake/laravel-cors
Discovered Package: laravel/tinker
Discovered Package: nesbot/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.
PS C:\xampp\htdocs>
```

## 2. Install Via Composer Create-Project

Selain menggunakan cara pertama, Kita dapat melakukan penginstalan laravel dengan mengetikkan perintah berikut pada command prompt:

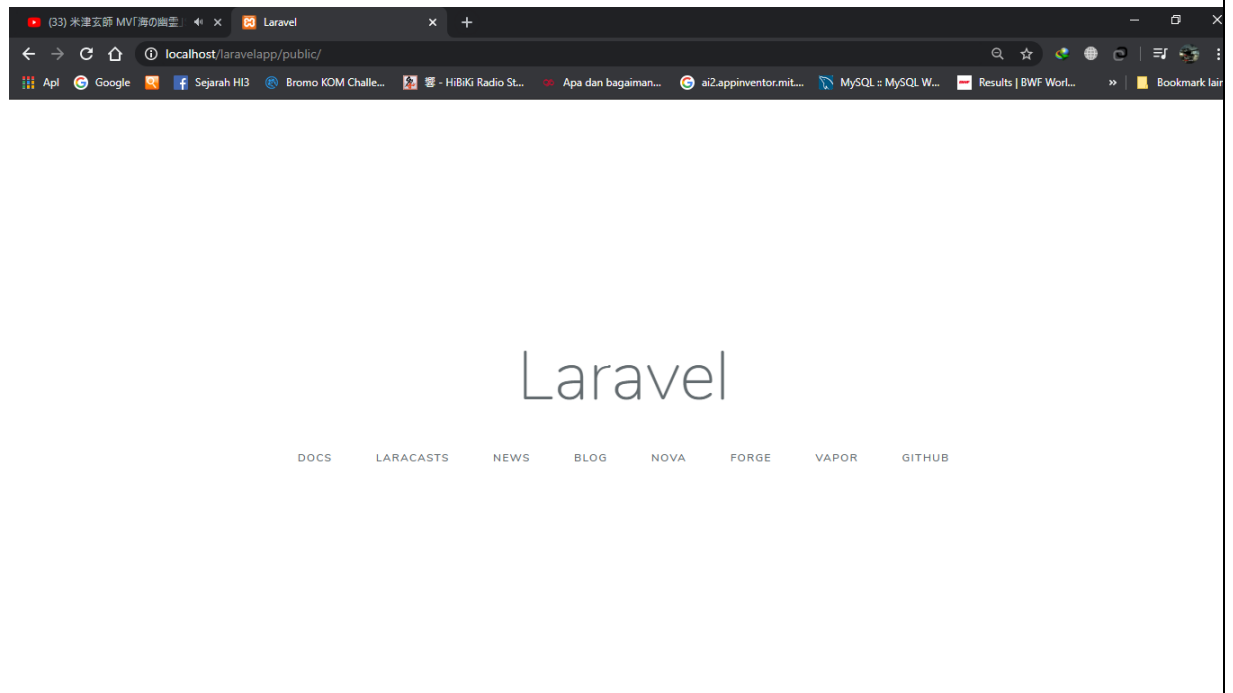
```
composer create-project --prefer-dist laravel/laravel laravelapp
```




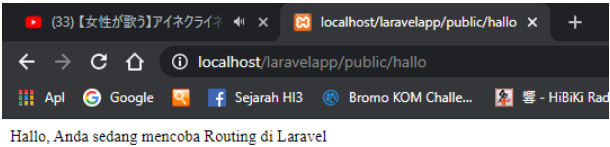
3

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 <http://localhost/laravelapp/public/> Atau Kita dapat menggunakan perintah berikut di Command Prompt:

**php artisan serve**



## 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 <b>laravelapp\routes\web.php</b> Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut</p>  <pre>1 &lt;?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</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 <b>laravelapp\routes\web.php</b>, tambahkan isinya dengan script di bawah ini :</p> <p><b>Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; });</b></p> <p>Untuk melihat hasilnya, pada browser, ketikkan alamat <a href="http://localhost/laravelapp/public/hallo">localhost/laravelapp/public/hallo</a></p> 

3

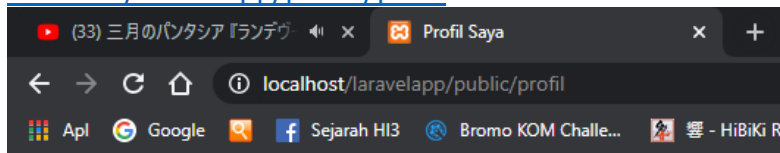
Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:

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

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

```
<!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>
```

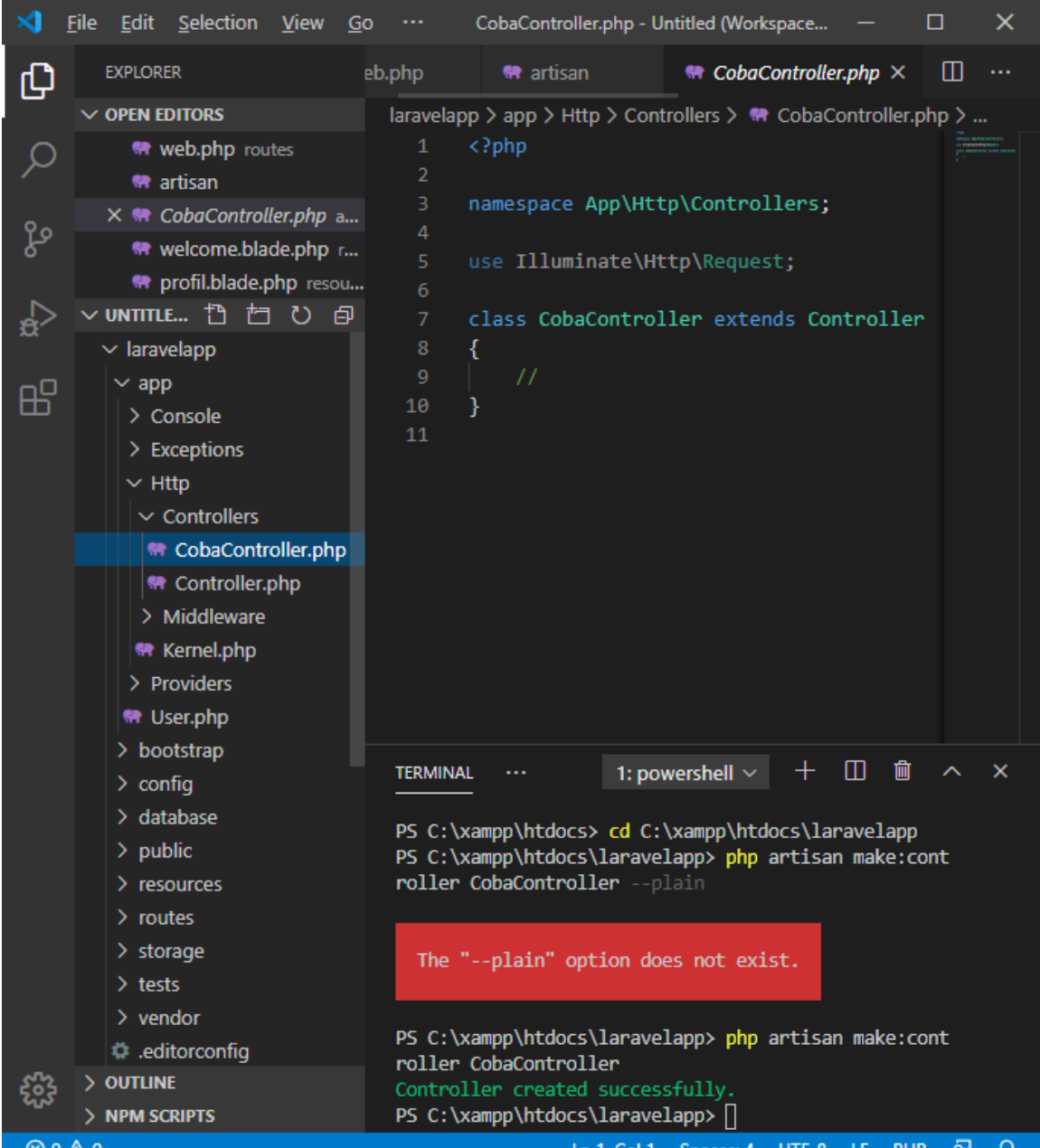
Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat [localhost/laravelapp/public/profil](http://localhost/laravelapp/public/profil)



## **Profil Saya**

Perkenalkan Nama saya Agsal, saya seorang Programmer .

## 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 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> 



2

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

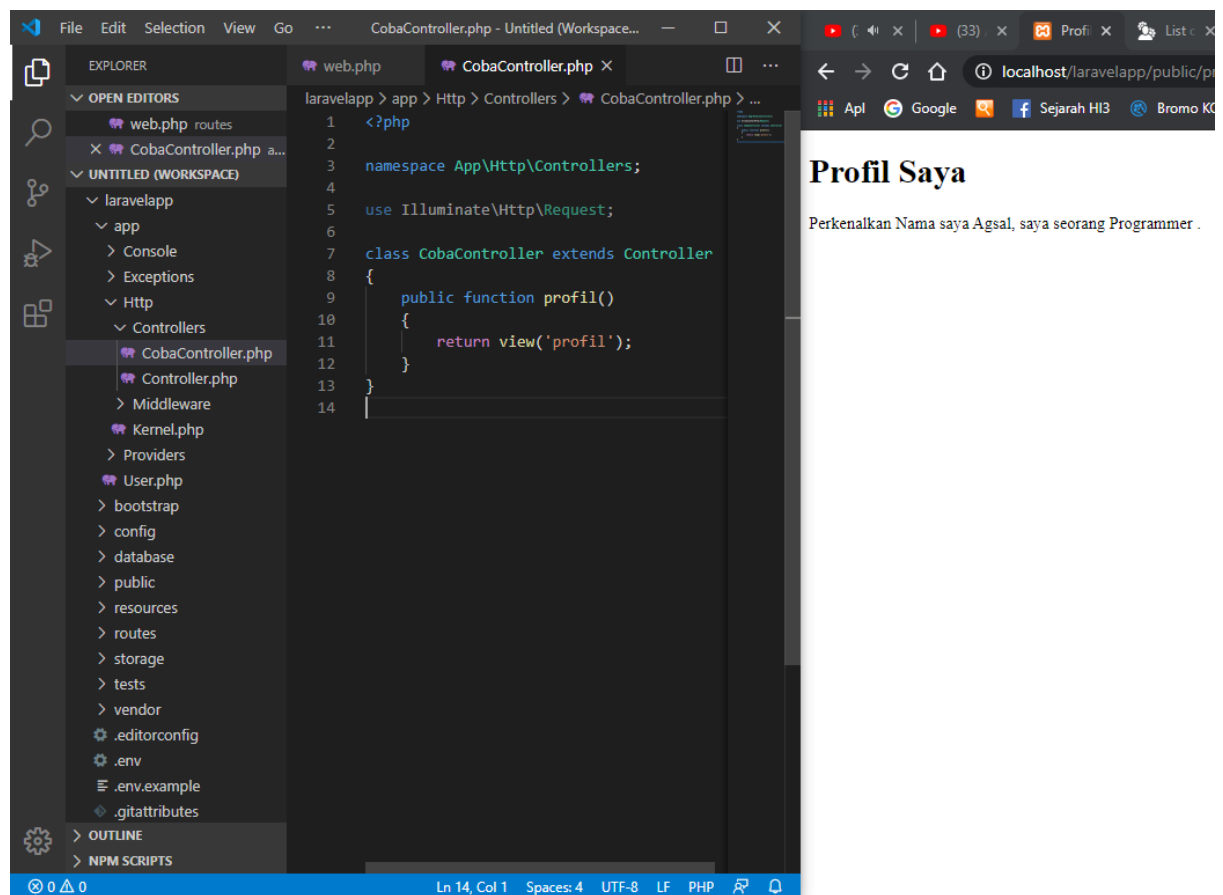
```
Route::get('profil', 'CobaController@profil');
```

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

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

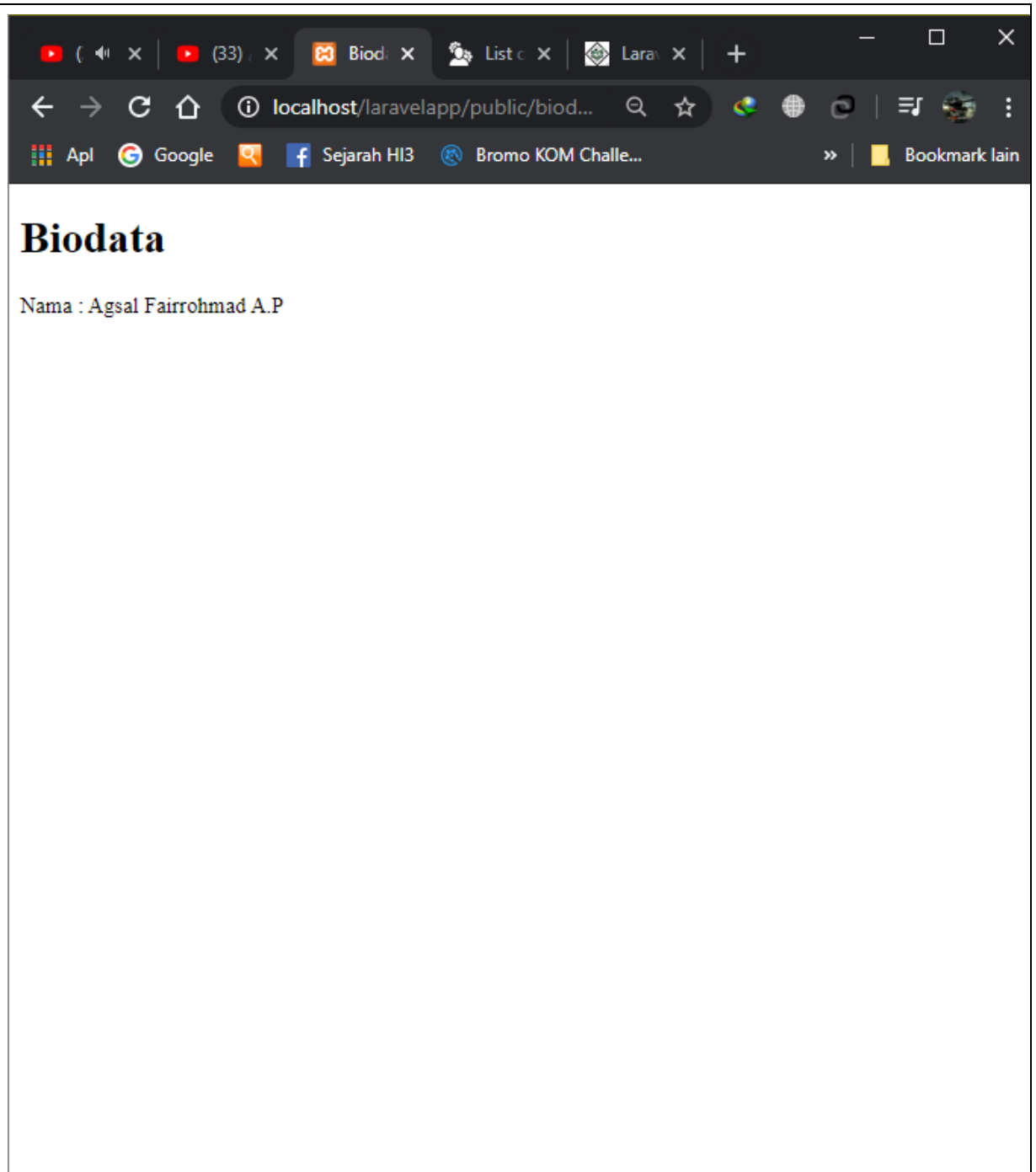
class CobaController extends Controller
{
    public function profil()
    {
        return view('profil');
    }
}
```

Ketikkan alamat [localhost/laravelapp/public/profil](http://localhost/laravelapp/public/profil), dan Kita akan mendapatkan hasil yang sama seperti pada latihan sebelumnya.



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

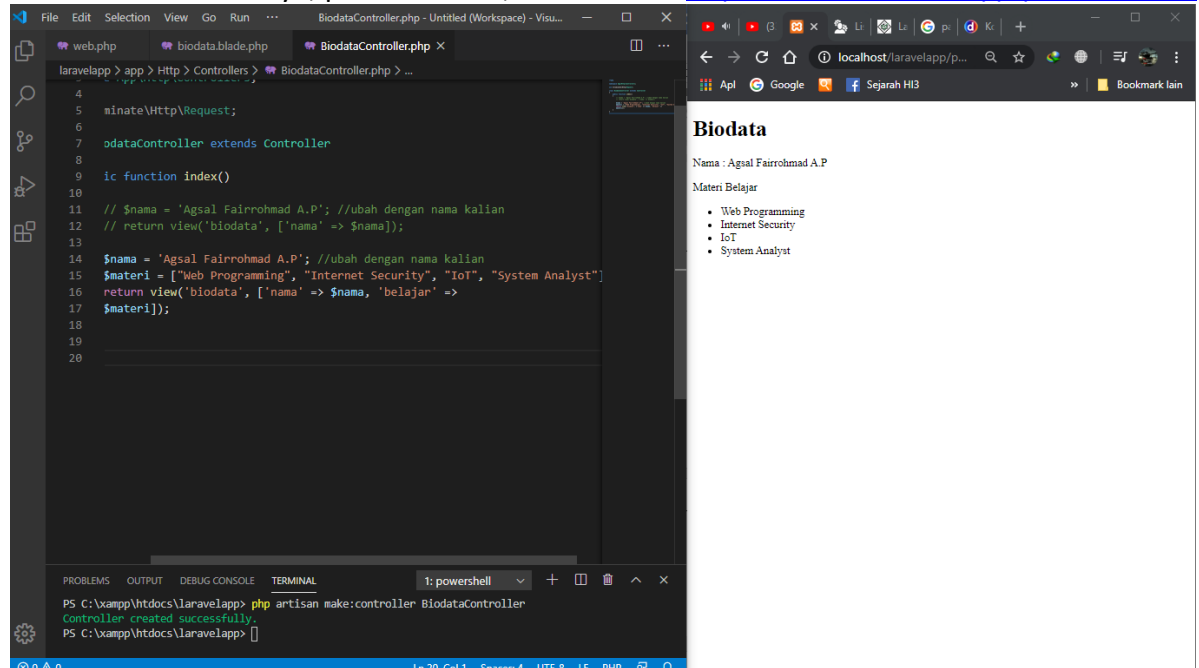
Langkah	Keterangan
1	<p>Dalam sebuah aplikasi, sangat jarang Kita temukan isi dari file view yang bersifat statis. Pada umumnya, view dipakai untuk menampilkan data, contohnya adalah data yang berasal dari database. Namun, karena kita belum sampai pada pembahasan database, maka pada latihan kali ini kita akan mencoba menampilkan data dari variabel.</p> <p>Buat route baru untuk halaman biodata</p> <pre>Route::get('biodata', 'BiodataController@index');</pre> <p>Buat controller baru dengan nama <b>BiodataController.php</b></p> <pre>php artisan make:controller BiodataController</pre> <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 = 'Agsal Fairrohmad A.P'; //ubah dengan nama kalian         return view('biodata', ['nama' =&gt; \$nama]);     } }</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;/body&gt;  &lt;/html&gt;</pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat <a href="http://localhost/laravelapp/public/biodata">http://localhost/laravelapp/public/biodata</a></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 <b>BiodataController.php</b></p> <pre>&lt;?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class BiodataController extends Controller {     public function index()     {         \$nama = 'Agsal Fairrohmad A.P'; //ubah dengan nama kalian         \$materi = ["Web Programming", "Internet Security", "IoT",                   "System Analyst"];          return view('biodata', ['nama' =&gt; \$nama, 'belajar' =&gt;                                 \$materi]);     } }</pre>
2	<p>Tambahkan isi file view <b>biodata.blade.php</b> menjadi seperti berikut:</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;p&gt;Materi Belajar&lt;/p&gt;     &lt;ul&gt;         @foreach(\$belajar as \$databelajar)             &lt;li&gt;{{ \$databelajar }}&lt;/li&gt; @endforeach     &lt;/ul&gt; &lt;/body&gt;  &lt;/html&gt;</pre>

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



**-- Selamat Mengerjakan --**