

# **LAPORAN TUGAS PEMROGRAMAN WEB LANJUT**

## **Pengantar dan Instalasi Laravel**



**Disusun Oleh :**

**ACHMAD RHEZA DARMAWAN / NIM : 1841720115**

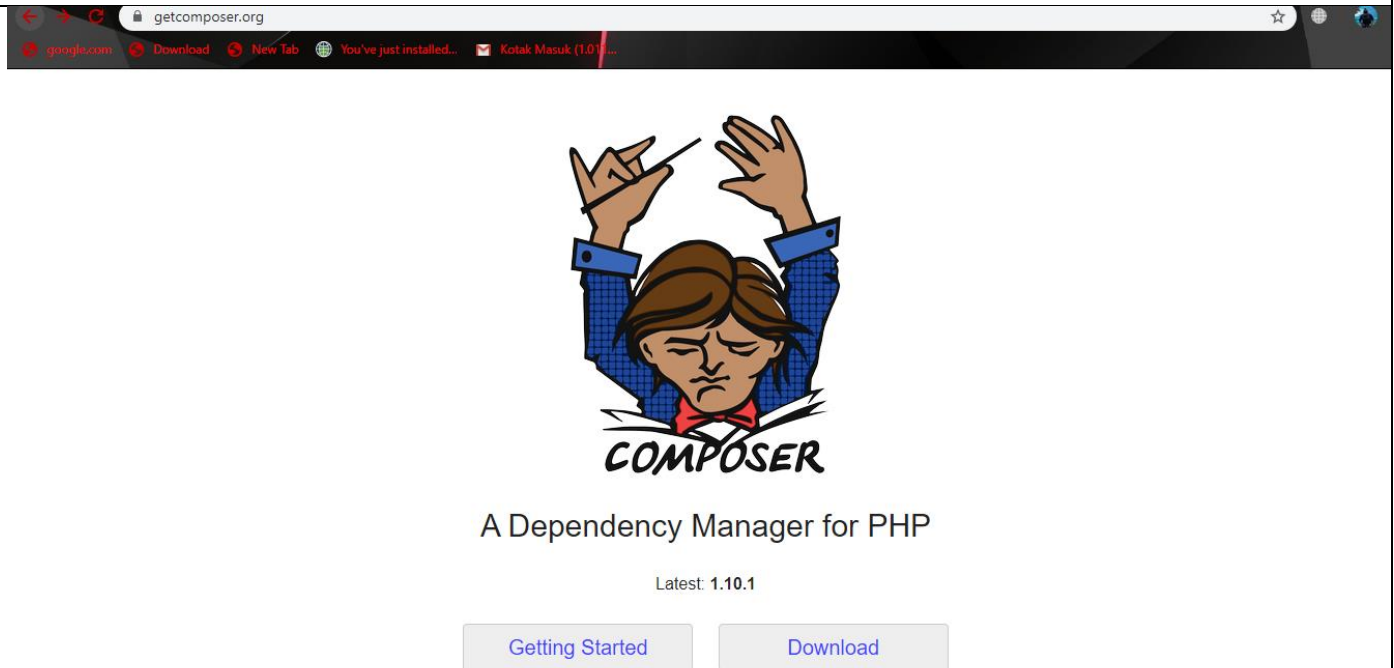
**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**Jl. Candi Panggung No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang,  
Jawa Timur 65141**

**2020**

## Praktikum 1

1



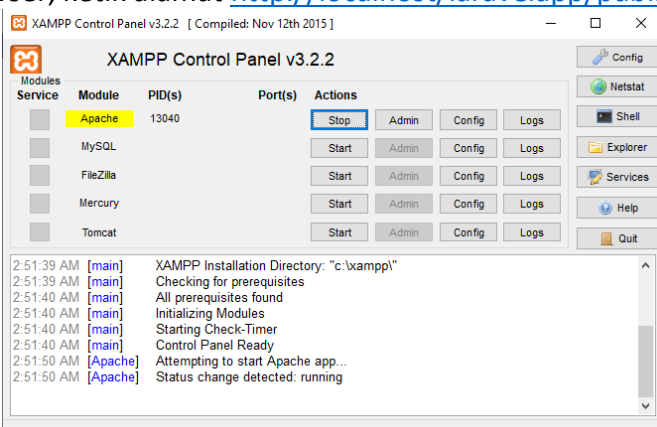
2

### Install Via Laravel Installer

```
Select C:\Windows\System32\cmd.exe
laravel/framework suggests installing ext-memcached (Required to use the memcache cache driver.)
laravel/framework suggests installing ext-pcntl (Required to use all features of the queue worker.)
laravel/framework suggests installing ext-posix (Required to use all features of the queue worker.)
laravel/framework suggests installing ext-redis (Required to use the Redis cache and queue drivers (^4.0|^5.0).)
laravel/framework suggests installing league/flysystem-aws-s3-v3 (Required to use the Flysystem S3 driver (^1.0).)
laravel/framework suggests installing league/flysystem-azure (Required to use the Flysystem Azure driver (^1.0).)
laravel/framework suggests installing league/flysystem-cached-adapter (Required to use the Flysystem cache (^1.0).)
laravel/framework suggests installing league/flysystem-sftp (Required to use the Flysystem SFTP driver (^1.0).)
laravel/framework suggests installing moontoast/math (Required to use ordered UUIDs (^1.1).)
laravel/framework suggests installing nyholm/psr7 (Required to use PSR-7 bridging features (^1.2).)
laravel/framework suggests installing pda/pheanstalk (Required to use the beanstalk queue driver (^4.0).)
laravel/framework suggests installing pusher/pusher-php-server (Required to use the Pusher broadcast driver (^4.0).)
laravel/framework suggests installing symfony/cache (Required to use PSR-6 cache bridge (^5.0).)
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to use PSR-7 bridging features (^2.0).)
laravel/framework suggests installing wildcard/swiftmailer-postmark (Required to use Postmark mail driver (^3.0).)
guzzlehttp/guzzle suggests installing zendframework/zend-httpdierrunner (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :)
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psy/psysh suggests installing hoar/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
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 (^3.1)
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: fidoper/proxy
Discovered Package: fruitcake/laravel-logs
Discovered Package: laravel/tinker
Discovered Package: mezzio/carbon
Discovered Package: nunomaduro/collision
Package manifest generated successfully.
Application ready! Build something amazing.
```

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/>





# Laravel

[DOCS](#) [LARACASTS](#) [NEWS](#) [BLOG](#) [NOVA](#) [FORGE](#) [VAPOR](#) [GITHUB](#)



# Laravel

[DOCS](#) [LARACASTS](#) [NEWS](#) [BLOG](#) [NOVA](#) [FORGE](#) [VAPOR](#) [GITHUB](#)

C:\Windows\System32\cmd.exe - php artisan serve

```
C:\xampp\htdocs\laravelapp>php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Sat Apr  4 03:04:29 2020] 127.0.0.1:51583 [200]: /favicon.ico
[Sat Apr  4 03:06:30 2020] 127.0.0.1:51603 [200]: /favicon.ico
```

## Praktikum 2

- 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 x
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6   |-----
7   | Web Routes
8   |-----
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
```

- 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>

```
web.php x
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6   |-----
7   | Web Routes
8   |-----
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
22      Routing di Laravel";
23  });
24
```



- 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:

```
web.php ×
routes > web.php
1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----|
7 | Web Routes
8 |-----|
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
22     Routing di Laravel";
23 });
24
25 Route::get('profil', function () {
26     return view('profil');
27 });
```

```
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 Achmad Rheza, saya seorang Mahasiswa TI-2A NIM 1841720115 No Absen 02</p>
9 </body>
10 </html>
```

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



## Profil Saya

Perkenalkan Nama saya Achmad Rheza, saya seorang Mahasiswa TI-2A NIM 1841720115 No Absen 02

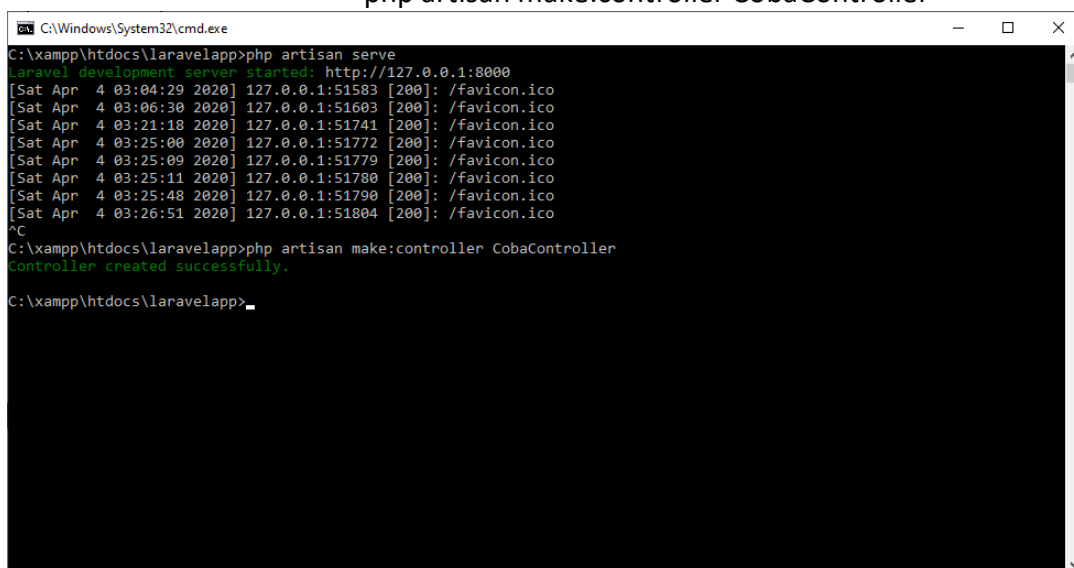
### Praktikum 3

- 1 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.

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.

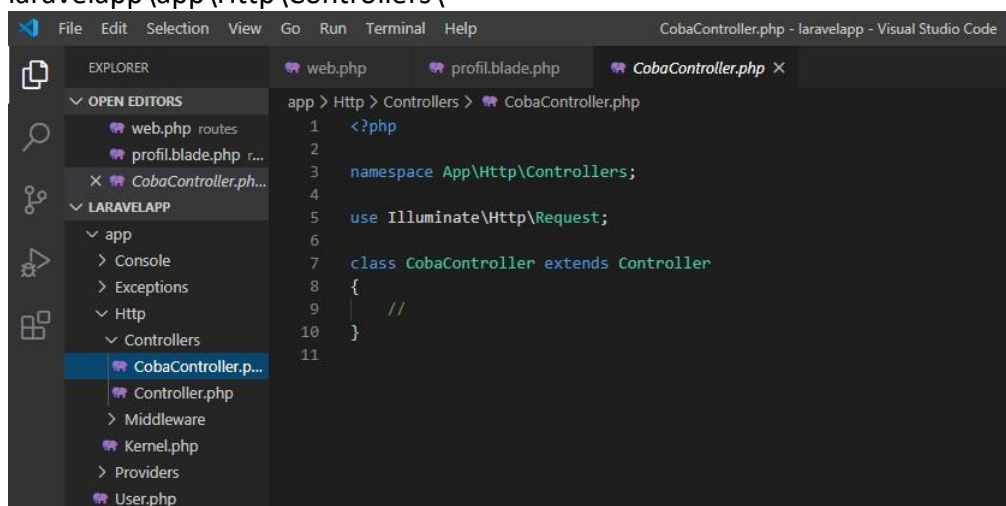
Kita gunakan cara yang kedua, pada latihan kali ini kita akan membuat Controller dengan nama CobaController dengan menuliskan di command prompt / terminal :

php artisan make:controller CobaController



```
C:\xampp\htdocs\laravelapp>php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Sat Apr 4 03:04:29 2020] 127.0.0.1:51583 [200]: /favicon.ico
[Sat Apr 4 03:06:30 2020] 127.0.0.1:51603 [200]: /favicon.ico
[Sat Apr 4 03:21:18 2020] 127.0.0.1:51741 [200]: /favicon.ico
[Sat Apr 4 03:25:00 2020] 127.0.0.1:51772 [200]: /favicon.ico
[Sat Apr 4 03:25:09 2020] 127.0.0.1:51779 [200]: /favicon.ico
[Sat Apr 4 03:25:11 2020] 127.0.0.1:51780 [200]: /favicon.ico
[Sat Apr 4 03:25:48 2020] 127.0.0.1:51790 [200]: /favicon.ico
[Sat Apr 4 03:26:51 2020] 127.0.0.1:51804 [200]: /favicon.ico
^C
C:\xampp\htdocs\laravelapp>php artisan make:controller CobaController
Controller created successfully.
C:\xampp\htdocs\laravelapp>
```

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



- 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:

```

16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba
22     Routing di Laravel";
23 });
24
25 Route::get('profil', 'CobaController@profil');
26

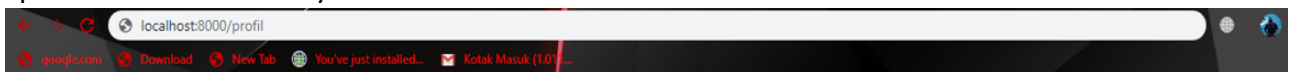
```

```

web.php • profil.blade.php • CobaController.php •
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 }
14

```

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



## Profil Saya

Perkenalkan Nama saya Achmad Rheza, saya seorang Mahasiswa TI-2A NIM 1841720115 No Absen 02

## Praktikum 4

- 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');
```

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

Buat controller baru dengan nama BiodataController.php

php artisan make:controller BiodataController

```
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

```
app > Http > Controllers > BiodataController.php
1  <?php
2  namespace App\Http\Controllers;
3  use Illuminate\Http\Request;
4  class BiodataController extends Controller {
5      public function index() {
6          $nama = 'Achmad Rheza Darmawan'; //ubah dengan nama kalian
7          return view('biodata', ['nama' => $nama]);
8      }
9  }
```

- 2 Buat file view dengan nama biodata.blade.php pada laravelapp/resources/views\

```
BiodataController.php  biodata.blade.php  profil.blade.php
resources > views > biodata.blade.php
1  <!DOCTYPE html> <html>
2  <head>
3  <title>Biodata</title>
4  </head>
5  <body>
6
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>



### Biodata

Nama : Achmad Rheza Darmawan



## Praktikum 5

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

```
app > Http > Controllers > BiodataController.php
1  <?php
2  namespace App\Http\Controllers;
3  use Illuminate\Http\Request;
4  class BiodataController extends Controller {
5      public function index() {
6          $nama = 'Achmad Rheza Darmawan'; //ubah dengan nama kalian
7          $materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]
8          return view('biodata', ['nama' => $nama 'materi' => $materi]);
9      }
10 }
```

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

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

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

