

# **LAPORAN PEMROGRAMAN WEB LANJUT**

## **JOBSHEET 9**

### **PENGANTAR DAN INSTALASI LARAVEL**

**Oleh:**

**KINANTI PERMATA PUTRI**

**NIM. 1841720022**

**TI-2A**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**APRIL 2020**

## 1. 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,

### a) **Install Via Laravel Installer**

Setelah proses instalasi Composer selesai, buka Command Prompt lalu tujuilah 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"**

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\xampp\htdocs>composer global require "laravel/installer"
```

```
C:\Windows\System32\cmd.exe
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!
```

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

**laravel new laravelapp**

```
C:\Windows\System32\cmd.exe
C:\xampp\htdocs>laravel new laravelapp
```

```
C:\Windows\System32\cmd.exe
laravel/framework suggests installing symfony/psr-http-message-bridge (Required to use PSR-7 bridging features (^2.0).)
laravel/framework suggests installing wildbit/swiftmailer-postmark (Required to use Postmark mail driver (^3.0).)
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (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 hoa/console (A pure PHP readline implementation. You'll want this if your PHP install does n'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 (*)
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-ondarka/php-console-color is abandoned, you should avoid using it. Use php-parallel-lint/php-console-color instead.
Package jakub-ondarka/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
> Illuminate\Foundation\ComposerScripts::postAutoloadDump
> @php artisan package:discover --ansi
Application ready! Build something amazing.
```

Keterangan:

**laravelapp**: merupakan folder kerja laravel, Kita dapat menggantinya sesuai dengan keinginan

b) **Install Via Composer Create-Project**

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

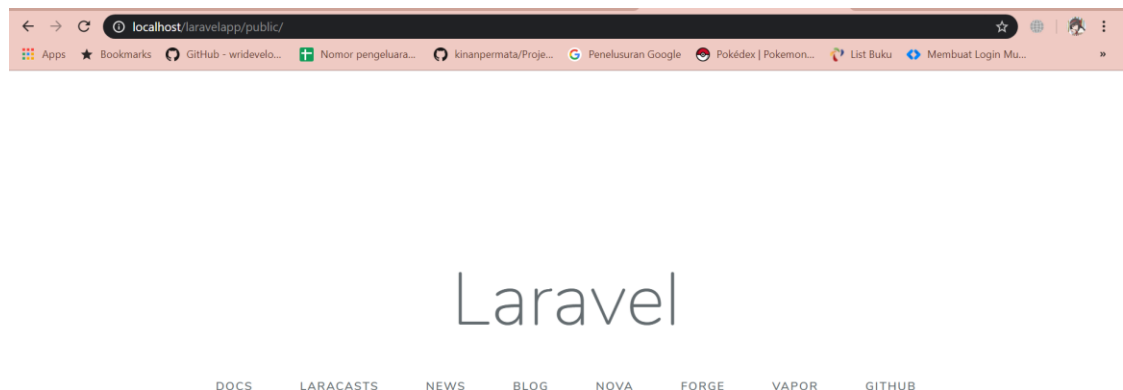
**composer create-project --prefer-dist laravel/laravel laravelapp**

- 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**

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:



2. 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  |-----
7  | Web Routes
8  |-----
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('/', function () {
17     return view('welcome');
18 });
19

```

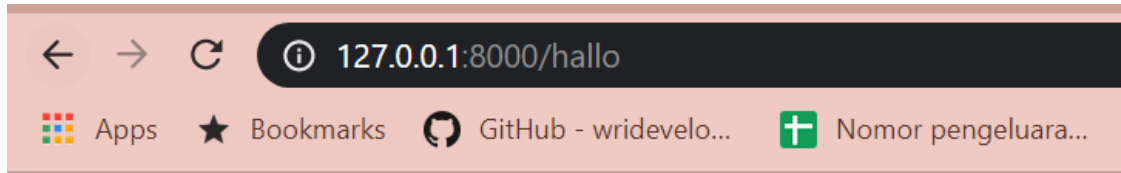
- 2) Kita dapat menuliskan perintah baru untuk membuat route, selain mengembalikan file view sebagai response, Kita dapat mengembalikan response berupa string atau teks biasa. Buka file **laravelapp/routes/web.php**, tambahkan isinya dengan script di bawah ini :

```

web.php ×
routes > web.php > #Function#f95be6b5
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. 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

```

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



Hallo, Anda sedang mencoba Routing di Laravel

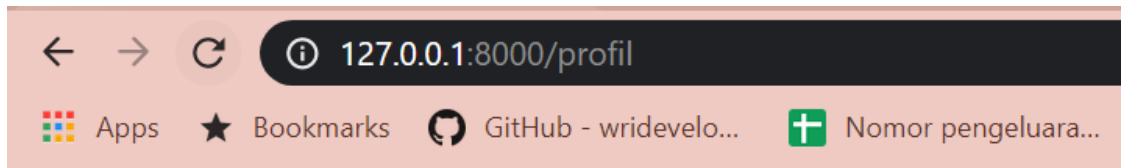
- 3) Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:

```
routes > web.php > #Function#f95d4f14
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. 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
20  Route::get('profil', function () {
21      return view('profil');
22  });
```

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

```
resources > views > profil.blade.php > html
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 Kinanti, saya seorang Mahasiswa</p>
9  </body>
10 </html>
```

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



# Profil Saya

Perkenalkan Nama saya Kinanti, saya seorang Mahasiswa

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

- 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.

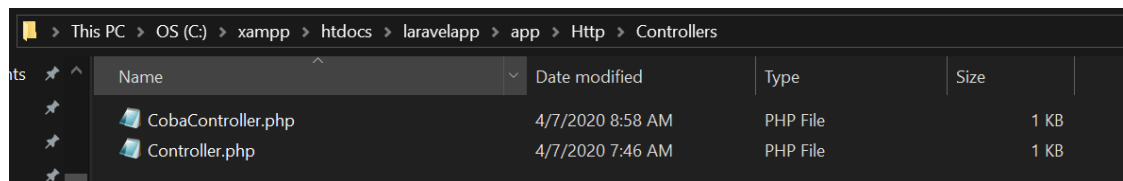
Kita gunakan cara yang kedua, pada latihan kali ini kita akan membuat Controller dengan nama CobaController dengan menuliskan di command prompt / terminal:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

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\**



Name	Date modified	Type	Size
CobaController.php	4/7/2020 8:58 AM	PHP File	1 KB
Controller.php	4/7/2020 7:46 AM	PHP File	1 KB

```
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
```

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

```
16 Route::get('hallo', function () {
17     return "Hallo, Anda sedang mencoba Routing di Laravel";
18 });
19
20 Route::get('profil', 'CobaController@profil');
```

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

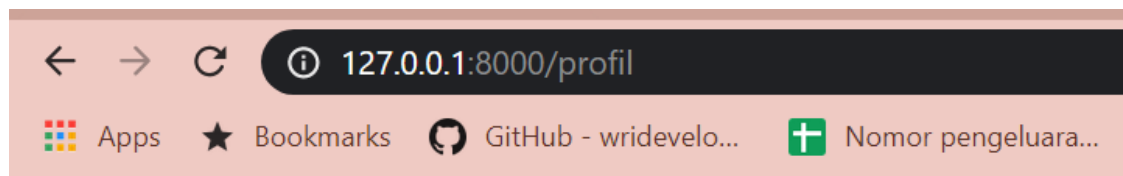


```

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 Kinanti, saya seorang Mahasiswa

#### 4. 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

```

16 Route::get('hallo', function () {
17     return "Hallo, Anda sedang mencoba Routing di Laravel";
18 });
19
20 Route::get('biodata', 'BiodataController@index');

```

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

```

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

```

File Explorer view of the Controllers directory:

Name	Date modified	Type	Size
BiodataController.php	4/7/2020 9:11 AM	PHP File	1 KB
CobaController.php	4/7/2020 9:02 AM	PHP File	1 KB
Controller.php	4/7/2020 7:46 AM	PHP File	1 KB

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

```

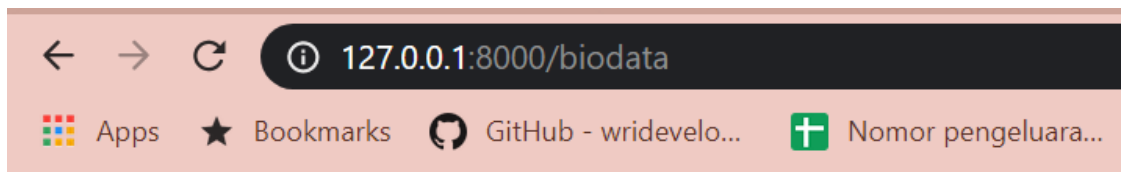
app > Http > Controllers > BiodataController.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class BiodataController extends Controller
8  {
9      public function index() {
10         $nama = 'Kinanti Permata Putri';
11         return view('biodata' , ['nama' => $nama]);
12     }
13 }
14

```

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

```
resources > views > biodata.blade.php > html
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>
```

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



# Biodata

Nama : Kinanti Permata Putri

## 5. Praktikum – Bagian 5: Memberikan Data Array Kepada View

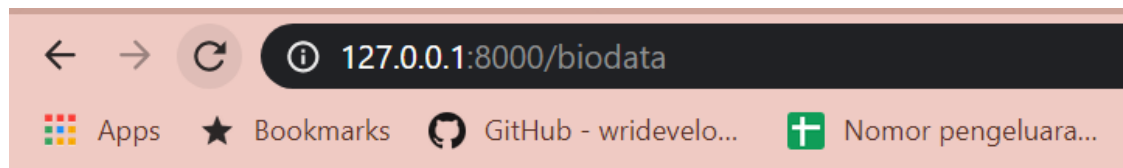
- 1) Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada **BiodataController.php**

```
app > Http > Controllers > BiodataController.php > PHP Intelephense > 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 = 'Kinanti Permata Putri';
11         $materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"];
12         return view('biodata', ['nama' => $nama, 'materi' => $materi]);
13     }
14 }
15
```

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

```
resources > views > biodata.blade.php > html > body
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> .



# Biodata

Nama : Kinanti Permata Putri

Materi Mengajar

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