

LAPORAN PRAKTIKUM  
PEMROGRAMAN WEB LANJUT

Jobsheet 9

Pengantar dan Instalasi Laravel



NAMA : SILVI INDAH NOVITA SARI  
NIM : 1841720006  
KELAS : 2B – TI

D-IV TEKNIK INFORMATIKA  
TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
2019

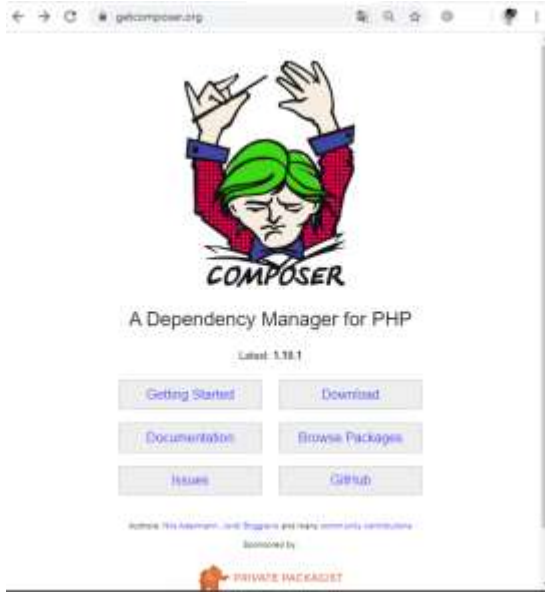


06 April 2020

Topik : Mengetahui Framework Laravel dan Proses Penginstallannya

Tujuan: Mahasiswa diharapkan dapat: 1. Memahami framework Laravel 2. Memahami cara menginstall dan menjalankan Laravel

### Praktikum – Bagian 1: Instalasi Laravel di Windows

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,</p> <ol style="list-style-type: none"><li>1. Install Via Laravel Installer Setelah proses instalasi Composer selesai, buka Command Prompt lalu tuliskan 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: <code>composer global require "laravel/installer"</code></li></ol>

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd C:\xampp\htdocs

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

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

laravel new laravelapp

```
C:\xampp\htdocs>laravel new laravelapp
```

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:

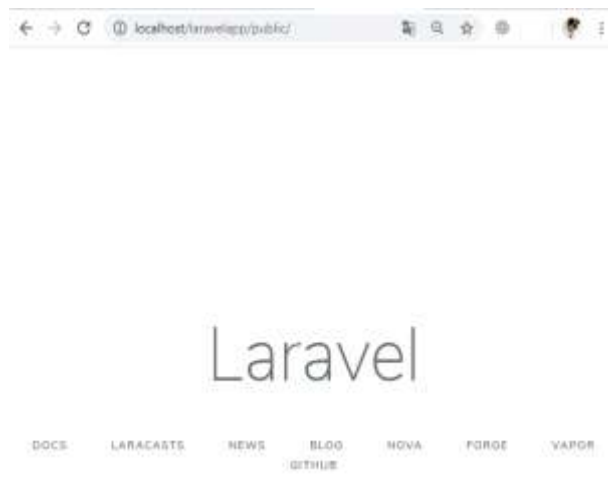
```
composer create-project --prefer-dist laravel/laravel laravelapp
```

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

php artisan serve

```
C:\xampp\htdocs>php artisan serve
```

Melalui WEB



	<p>Melalui command prompt</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 localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :</p>
--	--

## Praktikum – Bagian 2: Mengenal dan membuat route Langkah

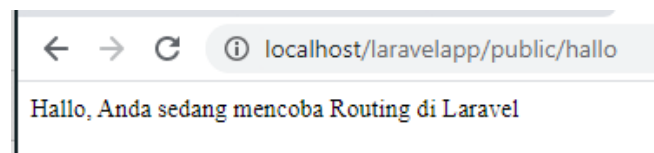
1	<p>Setiap request yang datang pada laravel akan diarahkan melalui sebuah route. Route ini yang akan menentukan respon apa yang akan dikerjakan untuk membalas request tersebut. Halaman homepage default Laravel seperti yang ditampilkan pada gambar di Bagian 1 adalah hasil dari script yang terletak pada file laravelapp\routes\web.php Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut</p> 
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>

```

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
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba Routing di Laravel";
22 });
23

```

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'); });

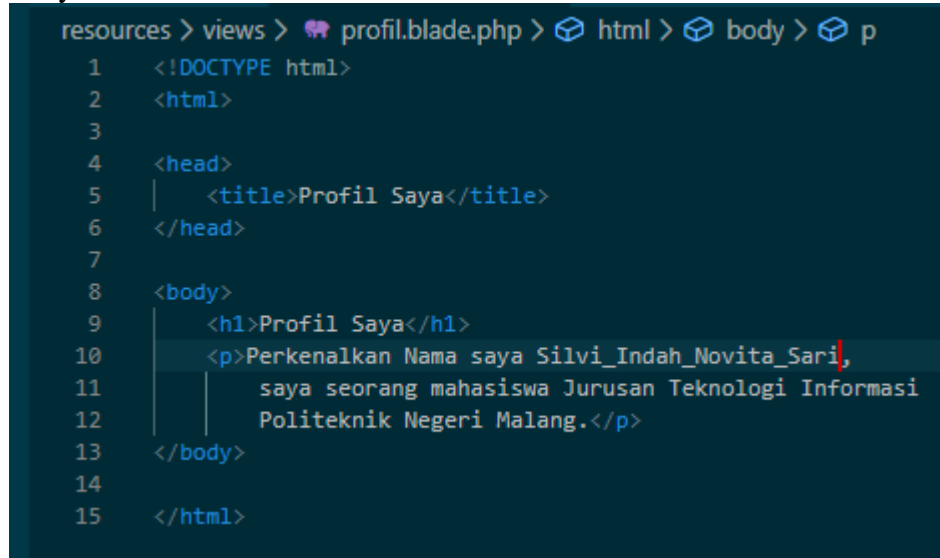
```

web.php X
routes > web.php > #Function#f95d5695
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
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba Routing di Laravel";
22 });
23
24 Route::get('profil', function () {
25     return view('profil');
26 });
27

```

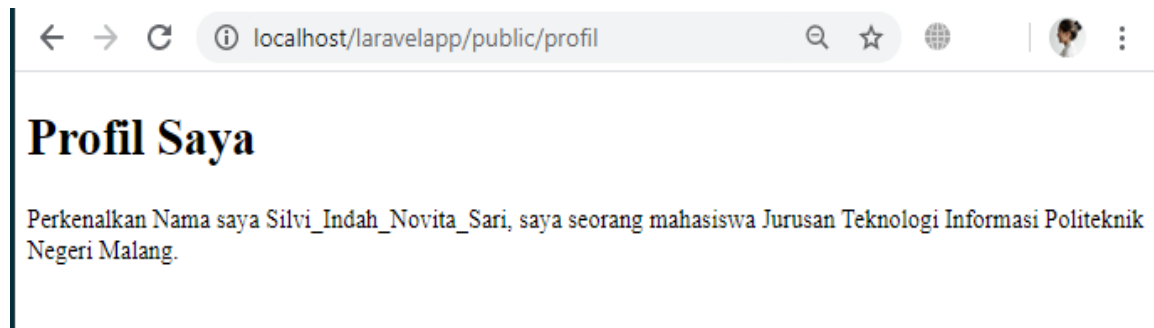
Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:

```
<!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>
```



```
resources > views > profil.blade.php > html > body > p
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 | <title>Profil Saya</title>
6 </head>
7
8 <body>
9 | <h1>Profil Saya</h1>
10 | <p>Perkenalkan Nama saya Silvi_Indah_Novita_Sari,
11 |   saya seorang mahasiswa Jurusan Teknologi Informasi
12 |   Politeknik Negeri Malang.</p>
13 </body>
14
15 </html>
```

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



### 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 :

php artisan make:controller CobaController

```
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\

```
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');

```
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
```

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

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

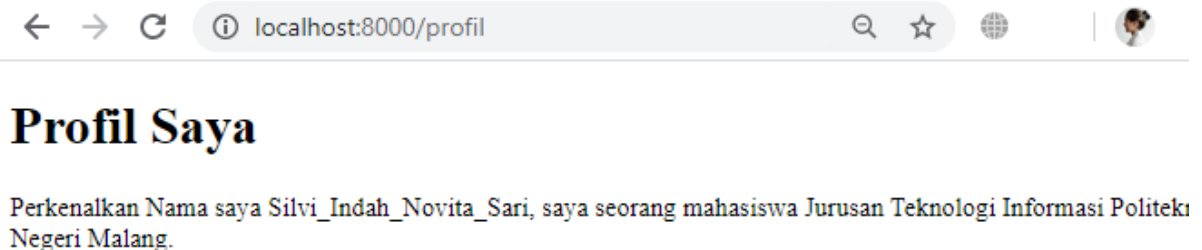
```
class CobaController extends Controller {    public function profil()    {        return
view('profil');    } }
```

```

app > Http > Controllers > CObaController.php > PHP Intelephense
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.



#### 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');
```

```
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.
```

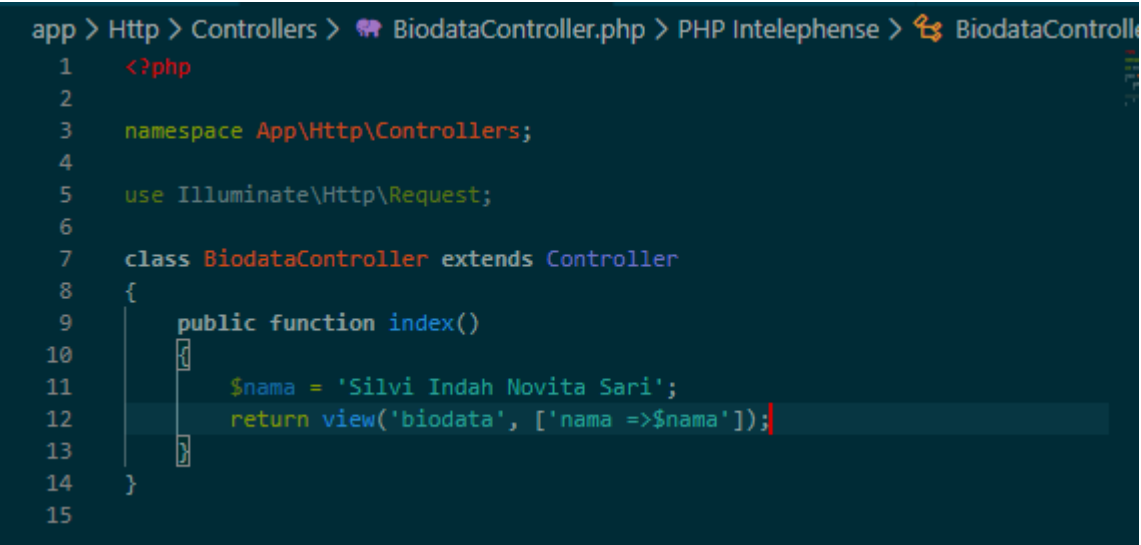
```
C:\xampp\htdocs\laravelapp>
```



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

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

class BiodataController extends Controller { public function index() { $nama
= 'Sugono Galih Aprianto'; //ubah dengan nama kalian return view('biodata',
['nama' => $nama]); } }
```



```
app > Http > Controllers > BiodataController.php > PHP Intelephense > 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     {
11         $nama = 'Silvi Indah Novita Sari';
12         return view('biodata', ['nama' => $nama]);
13     }
14 }
15
```

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

```
<!DOCTYPE html> <html> <head> <title>Biodata</title> </head> <body>
<h1>Biodata</h1> <p>Nama : {{ $nama }}</p> </body> </html>
```



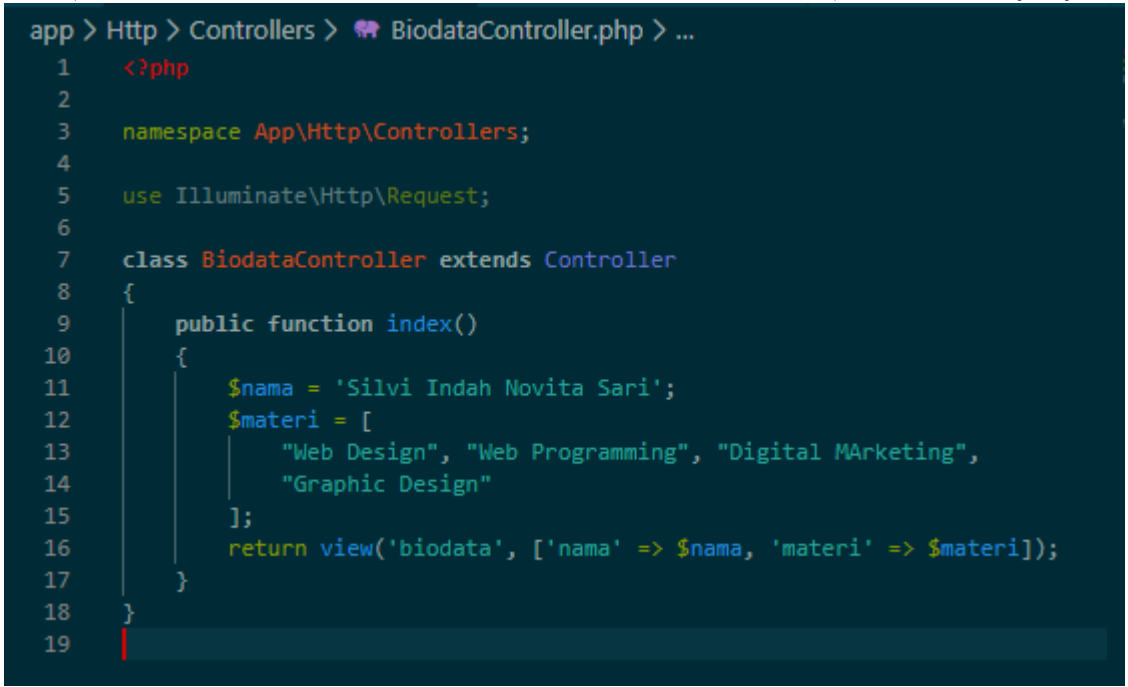
```
resources > views > biodata.blade.php > html > body > p
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Biodata</title>
6  </head>
7
8  <body>
9      <h1>Biodata</h1>
10     <p>Nama : {{ $nama }}</p>
11 </body>
12
13 </html>
```

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

# Biodata

Nama : Silvi Indah Novita Sari

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

1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre>&lt;?php namespace App\Http\Controllers; use Illuminate\Http\Request;  class BiodataController extends Controller { public function index() { \$nama = 'Sugono Galih Aprianto'; //ubah dengan nama kalian \$materi = ["Web Design", "Web Programming", "Digital Marketing","Graphic Design"]; return view('biodata', ['nama' =&gt; \$nama, 'materi' =&gt; \$materi]); } }</pre> 
2	<p>Tambahkan isi file view biodata.blade.php 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 Mengajar&lt;/p&gt; &lt;ul&gt; @foreach(\$materi as \$datamateri) &lt;li&gt;{{ \$datamateri }}&lt;/li&gt; @endforeach &lt;/ul&gt; &lt;/body&gt; &lt;/html&gt;</pre>

```
resources > views > biodata.blade.php > html > body > ul
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5  |   <title>Biodata</title>
6  </head>
7
8  <body>
9  |   <h1>Biodata</h1>
10 |   <p>Nama : {{$nama}}</p>
11 |   <p>Materi Mengajar </p>
12 |   <ul>
13 |       @foreach ($materi as $datamateri)
14 |           <li>{{$datamateri}}</li>
15 |       @endforeach
16 |   </ul>
17 </body>
18
19 </html>
```

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

## Biodata

Nama : Silvi Indah Novita Sari

Materi Mengajar

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