

JOBSHEET 9 PENGANTAR DAN INSTALASI LARAVEL PEMROGRAMAN WEB LANJUT



Oleh :
Inayati Machsus Izza Addin

1841720202

**Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang
2020**

Praktikum – Bagian 1: Instalasi Laravel di Windows

Langka h	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 https://getcomposer.org/, pilih Download untuk memulai proses unduh.</p>  <p>Jawaban:</p> <pre> C:\xampp\htdocs>composer Composer version 1.10.0 2020-03-10 14:08:05 Usage: command [options] [arguments] Options: -h, --help Display this help message -q, --quiet Do not output any message -V, --version Display this application version --ansi Force ANSI output --no-ansi Disable ANSI output -n, --no-interaction Do not ask any interactive questions --profile Display timing and memory usage information --no-plugins Whether to disable plugins. -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory. --no-cache Prevent use of the cache -v vv vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug </pre>
2	<p>Terdapat 2 cara dalam menginstall Laravel,</p> <p>1. Install Via Laravel Installer</p> <p>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: cd C:\xampp\htdocs. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p style="text-align: center;">composer global require "laravel/installer"</p>

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

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

Jawaban:

```
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.
C:\xampp\htdocs>
```

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/>

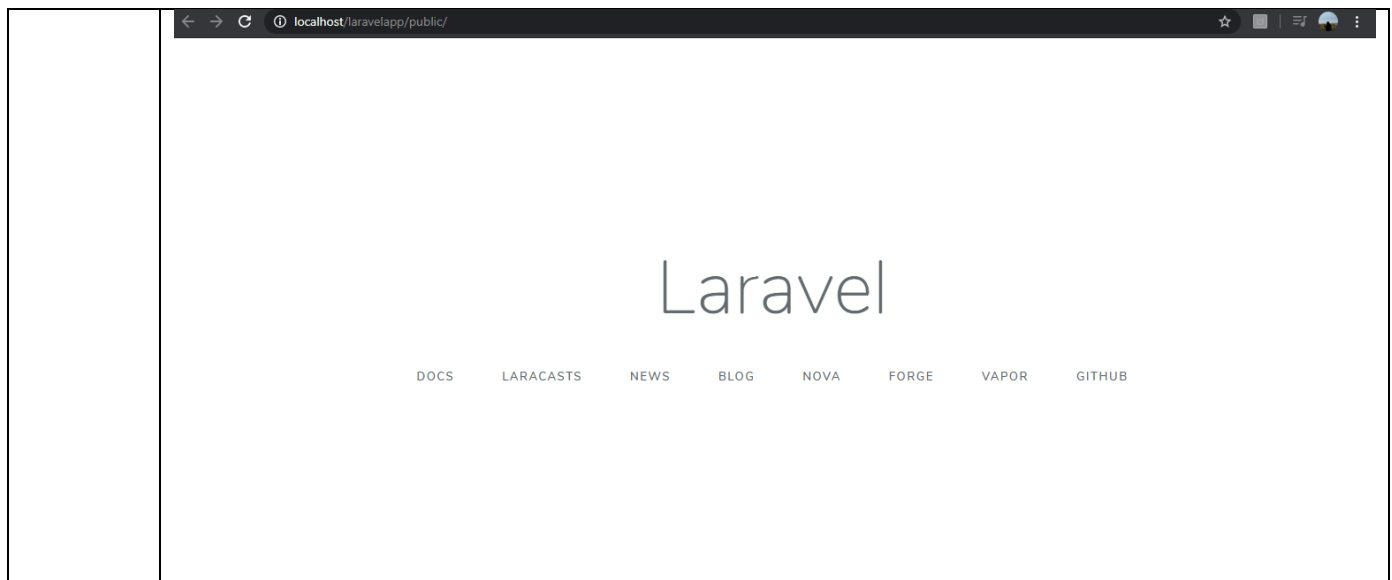
Atau Kita dapat menggunakan perintah berikut di Command Prompt:

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).

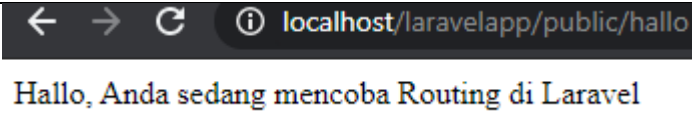
Jalankan dengan localhost:8000, dan pastikan tampilan di browser akan seperti berikut ini :

Jawaban:



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 laravelapp\routes\web.php Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut</p>  <pre> 1 <?php 2 3 /* 4 * Web Routes 5 * 6 * Here is where you can register web routes for your application. These 7 * routes are loaded by the RouteServiceProvider within a group which 8 * contains the "web" middleware group. Now create something great! 9 */ 10 11 Route::get('/', function () { 12 return view('welcome'); 13 }); 14 15 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.</p> <p>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> <p>Untuk melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</p> <p>Jawaban:</p>

	
3	<p>Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:</p> <p style="text-align: center;">Route::get('profil', function () { return view('profil'); });</p> <p>Buat file baru bernama profil.blade.php di dalam folder laravelapp\resources\views\, ketikkan script berikut ini:</p> <pre><!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></pre> <p>Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat http://localhost:8000/profil</p> <p>Jawaban:</p> 

Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel

Langka h	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>

php artisan make:controller CobaController

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



```
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');

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

```
<?php
```

```
namespace App\Http\Controllers;
```

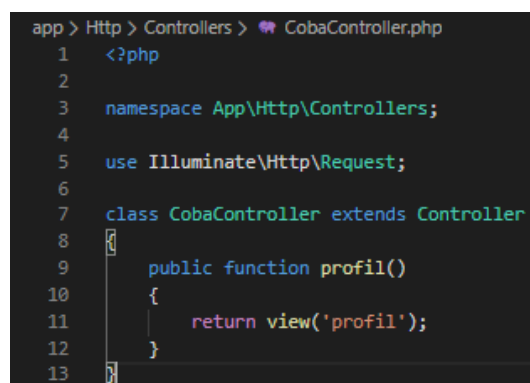
```
use Illuminate\Http\Request;
```

```
class CobaController extends Controller
```

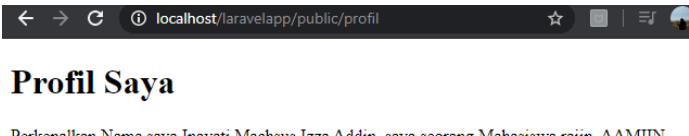
```
{
    public function profil()
    {
        return view('profil');
    }
}
```

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

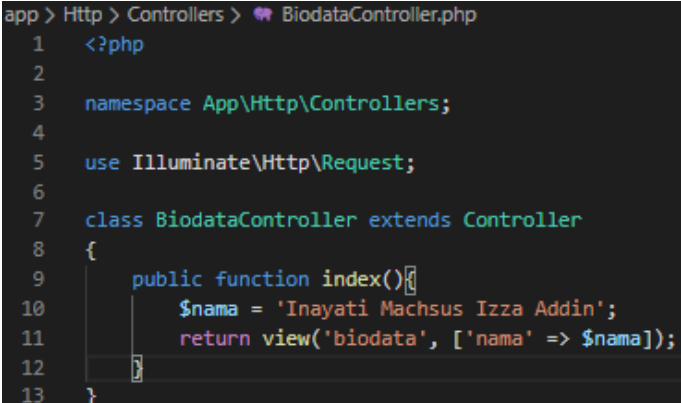
Jawaban:



```
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 }
```

	
--	---

Praktikum – Bagian 4: Memberikan Data Controller kepada View

Langka h	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. Buat route baru untuk halaman biodata</p> <p style="text-align: center;">Route::get('biodata', 'BiodataController@index');</p> <p>Buat controller baru dengan nama BiodataController.php</p> <p style="text-align: center;">php artisan make:controller BiodataController</p> <p>Ketikkan script berikut dengan memberikan method / fungsi index pada laravelapp\app\Http\Controllers\BiodataController.php</p> <pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index(){ \$nama = 'Inayati Machsus Izza Addin'; return view('biodata', ['nama' => \$nama]); } }</pre> <p>Jawaban:</p> 

2	<p>Buat file view dengan nama biodata.blade.php pada laravelapp/resources/views\</p> <pre> <!DOCTYPE html> <html> <head> <title>Biodata</title> </head> <body> <h1>Biodata</h1> <p>Nama : {{ \$nama }}</p> </body> </html> </pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat http://localhost:8000/biodata .</p> <p>Jawaban:</p> 
---	--

Praktikum – Bagian 5: Memberikan Data Array Kepada View

Langka h	Keterangan
1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre> <?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index(){ \$nama = 'Inayati Machsus Izza Addin'; \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); } } </pre> <p>Jawaban:</p>


```

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 = 'Inayati Machsus Izza Addin';
11         $materi = ["Web Design", "Web Programming", "Digital Marketing", "Gra
12         return view('biodata', ['nama' => $nama, 'materi' => $materi]);
13     }
14 }

```

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

```

<!DOCTYPE html>
<html>
<head>
    <title>Biodata</title>
</head>
<body>
    <h1>Biodata</h1>
    <p>Nama : {{ $nama }}</p>
    <p>Materi Mengajar</p>
    <ul>
        @foreach($materi as $datamateri)
            <li>{{ $datamateri }}</li>
        @endforeach
    </ul>
</body>
</html>

```

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

Jawaban:

← → ↻ ⓘ localhost/laravelapp/public/biodata

Biodata

Nama : Inayati Machsus Izza Addin

Materi Mengajar

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