



LAPORAN PRAKTIKUM

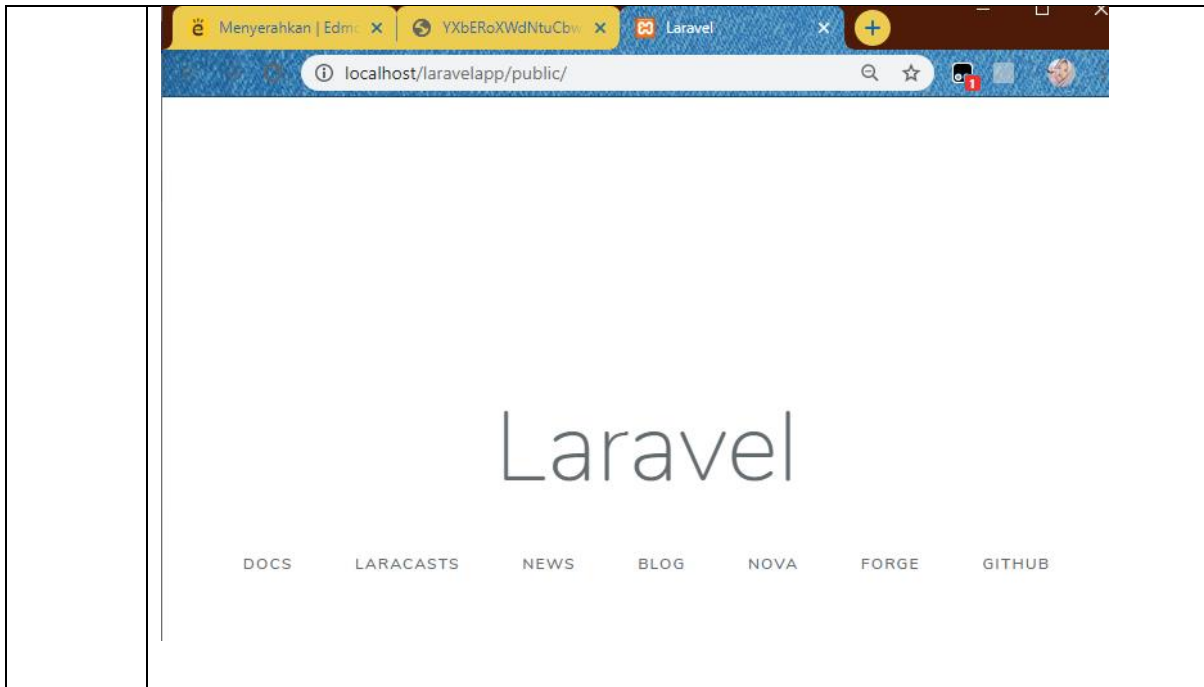
Web Lanjut

NAMA : Dina Lisuardi
NIM : 1941723004
Kelas : TI 2B
Isi Laporan : Jobsheet-9 : Pengantar dan Instalasi Laravel

Bagian 1

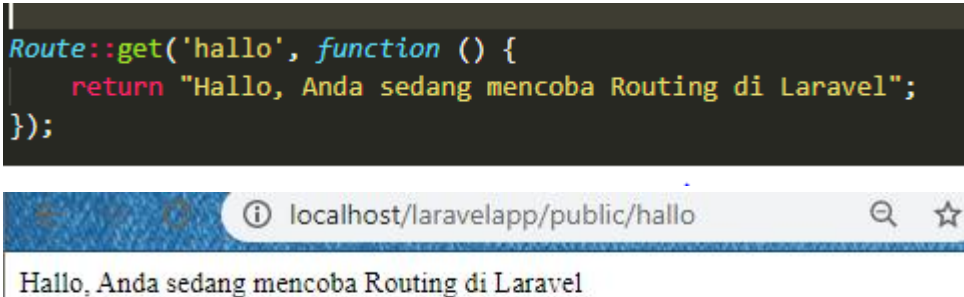
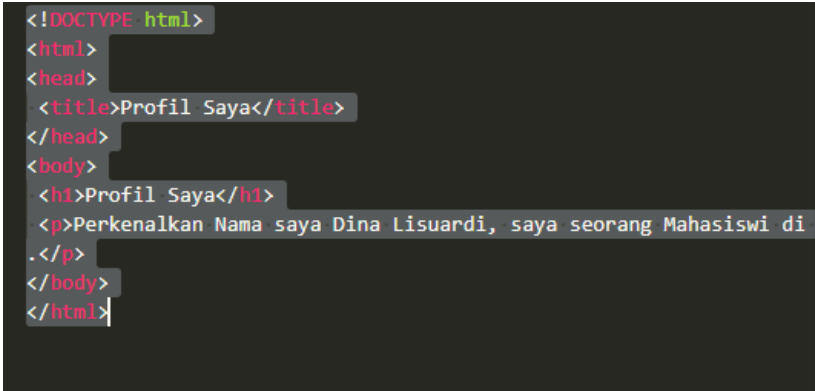
Langkah	Keterangan
1	<p>Install Via Laravel Installer Setelah proses instalasi Composer selesai, buka Command Prompt lalu tuliskan lokasi folder yang akan menampung folder kerja kita. Karena pada praktikum sebelumnya telah 2 menggunakan XAMPP, maka dapat menggunakan folder htdocs. Pindah ke direktori htdocs dengan ketik:</p> <p>cd C:\xampp\htdocs.</p> <p>Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p>composer global require "laravel/installer"</p> <pre>c:\xampp2\htdocs>composer global require "laravel/installer" Changed current directory to C:/Users/lenovo/AppData/Roaming/Composer Using version ^2.3 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 (v4.4.7): Downloading (100%) - Installing symfony/polyfill-ctype (v1.15.0): Downloading (100%) - Installing symfony/filesystem (v4.4.7): Downloading (100%) - Installing psr/container (1.0.0): Downloading (100%) - Installing symfony/service-contracts (v1.1.8): Downloading (100%) - Installing symfony/polyfill-php73 (v1.15.0): Downloading (100%) - Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%) - Installing symfony/console (v4.4.7): Downloading (100%)</pre> <p>laravel new laravelapp</p>

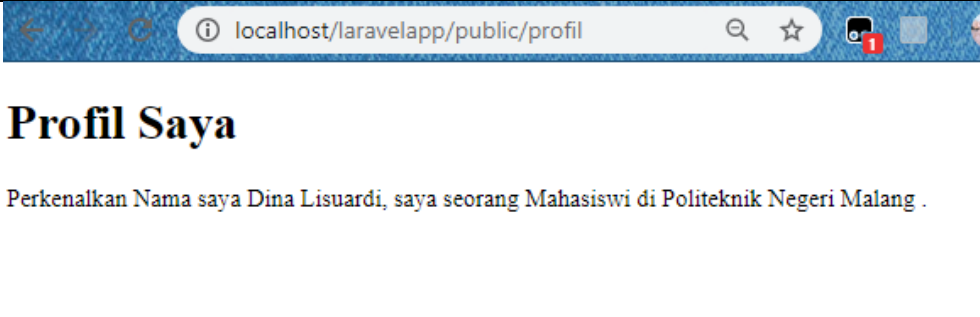
	<pre>c:\xampp2\htdocs>laravel new laravelapp</pre> <p>In NewCommand.php line 108:</p> <p>Application already exists!</p> <pre>new [--dev] [--auth] [-f --force] [--] [<name>]</pre> <pre>c:\xampp2\htdocs>_</pre> <p>Install Via Composer Create-Project</p> <pre>c:\xampp2\htdocs>composer create-project --prefer-dist laravel/laravel laravelapp Installing laravel/laravel (v5.8.35)</pre> <p>[InvalidArgumentException] Project directory laravelapp/ is not empty.</p> <pre>create-project [-s --stability STABILITY] [--prefer-source] [--prefer-dist] [--repository REPO SITORY] [--repository-url REPOSITORY-URL] [--dev] [--no-dev] [--no-custom-installers] [--no-sc ripts] [--no-progress] [--no-secure-http] [--keep-vcs] [--remove-vcs] [--no-install] [--ignore -platform-reqs] [--] [<package>] [<directory>] [<version>]</pre> <pre>c:\xampp2\htdocs></pre>
2	<p>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/</p> <p>Atau Kita dapat menggunakan perintah berikut di Command Prompt:</p> <p>php artisan serve</p> <pre>c:\xampp2\htdocs>cd laravelapp</pre> <pre>c:\xampp2\htdocs\laravelapp>php artisan server</pre> <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>



Bagian 2

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> web.php routes > web.php 1 <?php 2 3 /* 4 ----- 5 Web Routes 6 ----- 7 8 Here is where you can register web routes for your application. The 9 routes are loaded by the RouteServiceProvider within a group which 10 contains the "web" middleware group. Now create something great! 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 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>melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</p>  <pre>Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; });</pre> <p>The browser screenshot shows the URL <code>localhost/laravelapp/public/hallo</code> and the output text: <code>Hallo, Anda sedang mencoba Routing di Laravel</code>.</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>Route::get('profil', function () { return view('profil'); });</p> <p>Buat file baru bernama profil.blade.php di dalam folder laravelapp\routes\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 Dina Lisuardi, saya seorang Mahasiswi di .</p> </body> </html></pre> <p>Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat http://localhost/laravelapp/public/profil</p>

	
--	--

Bagian 3

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. 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 :</p> <p>php artisan make:controller</p> <pre>c:\xampp2\htdocs\laravelapp>php artisan make:controller CobaController Controller created successfully.</pre> <p>CobaController Maka akan terbentuk sebuah file dengan nama CobaController.php pada folder laravelapp\app\Http\Controllers\</p> <pre>1 <?php 2 3 namespace App\Http\Controllers; 4 5 use Illuminate\Http\Request; 6 7 class CobaController extends Controller 8 { 9 // 10 } 11</pre>

2

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

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

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

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

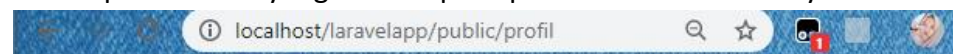
```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

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

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



Profil Saya

Perkenalkan Nama saya Dina Lisuardi, saya seorang Mahasiswi di Politeknik Negeri Malang .

Bagian 4

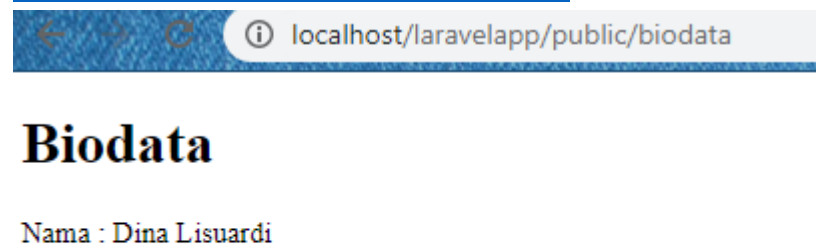
Langkah	Keterangan
1	<p>Dalam sebuah aplikasi, sangat jarang Kita temukan isi dari file view yang bersifat statis. Pada umumnya, view dipakai untuk menampilkan data, contohnya adalah data yang berasal dari database. Namun, karena kita belum sampai pada pembahasan database, maka pada latihan kali ini kita akan mencoba menampilkan data dari variabel. Buat route baru untuk halaman biodata</p> <pre>Route::get('biodata', 'BiodataController@index');</pre> <pre>Route::get('profil', 'BiodataController@index');</pre> <p>Buat controller baru dengan nama BiodataController.php php artisan make:controller BiodataController</p> <pre>c:\xampp2\htdocs\laravelapp>php artisan make:controller BiodataController Controller created successfully.</pre> <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 = 'Dina Lisuardi'; //ubah dengan nama kalian return view('biodata', ['nama' => \$nama]); } }</pre>
2	<p>Buat file view dengan nama biodata.blade.php pada laravelapp\resources\views\</p>

```

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

```

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

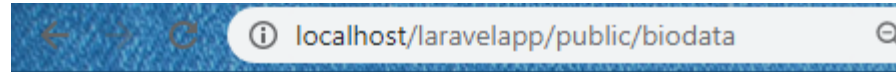


Bagian 5

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

	<pre> <?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller { public function index() { \$nama = 'Dina Lisuardi'; //ubah dengan nama kalian \$materi = ["Web Design", "Web Programming", "Digital Marketing","Graphic Design"]; return view('biodata' , ['nama' => \$nama, 'materi' => \$materi]); } </pre>
2	<p>Tambahkan isi file view biodata.blade.php menjadi seperti berikut:</p> <pre> <!DOCTYPE html> <html> <head> <title>Biodata</title> </head> <body> <h1>Biodata</h1> <p>Nama : {{ \$nama }}</p> <p>Materi Mengajar</p> @foreach(\$materi as \$datamateri) {{ \$datamateri }} @endforeach </body> </html> </pre>

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



Biodata

Nama : Dina Lisuardi

Materi Mengajar

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