

---

**PRAKTIKUM PEMROGRAMAN WEB**

BAB : XIII FRAMEWORK LARAVEL 1  
KELAS : PTI – C  
NAMA : AULIYAA ZULFA  
NIM : 235150600111035  
ASISTEN : TAUFIK HIDAYAT  
AZHARY MUNIR ABDILLAH  
TGL PRAKTIKUM : 21/11/2024

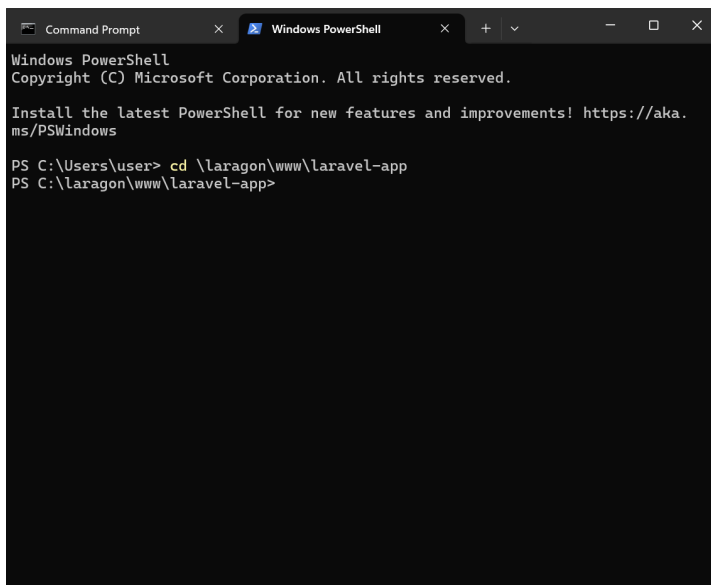
---

4. Buka file “<direktori XAMPP>\apache\conf\extra\httpd-vhosts.conf”. Tambahkan baris berikut pada akhir file tersebut:

<pre>&lt;VirtualHost laravel-app.test:80&gt;  DocumentRoot "&lt;direktori XAMPP&gt;/htdocs/laravel-app/public"  &lt;/VirtualHost&gt;</pre>
--

Sesuaikan “<direktori XAMPP>” dengan lokasi instalasi XAMPP pada komputer masing-masing.

Jelaskan apa yang terjadi setelah penambahan kode di atas dan mengapa kode tersebut dibuat.

**Screenshot****penjelasan**

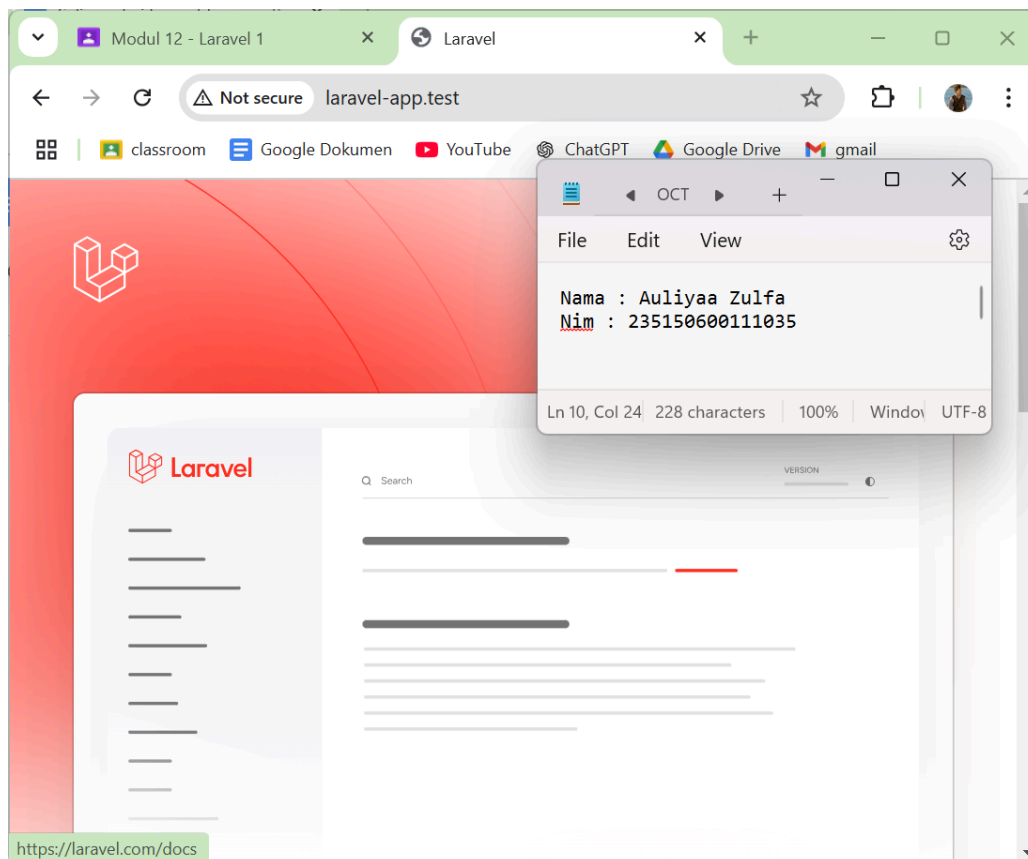
<VirtualHost laravel-app.test:80> Kode ini mendefinisikan sebuah Virtual Host untuk domain laravel-app.test pada port 80 (HTTP standar). Virtual Host memungkinkan Anda mengakses proyek Laravel melalui URL khusus, bukan menggunakan localhost/direktori-proyek

DocumentRoot "<direktori XAMPP>/htdocs/laravel-app/public"

Baris ini menentukan lokasi root directory aplikasi Laravel, yaitu di folder public. Laravel membutuhkan folder public sebagai titik masuk (entry point) untuk memproses permintaan HTTP.

saya menggunakan Laragon yang memiliki fitur otomatis untuk membuat Virtual Host. sehingga saya tidak perlu mengedit file httpd-vhosts.conf atau hosts.

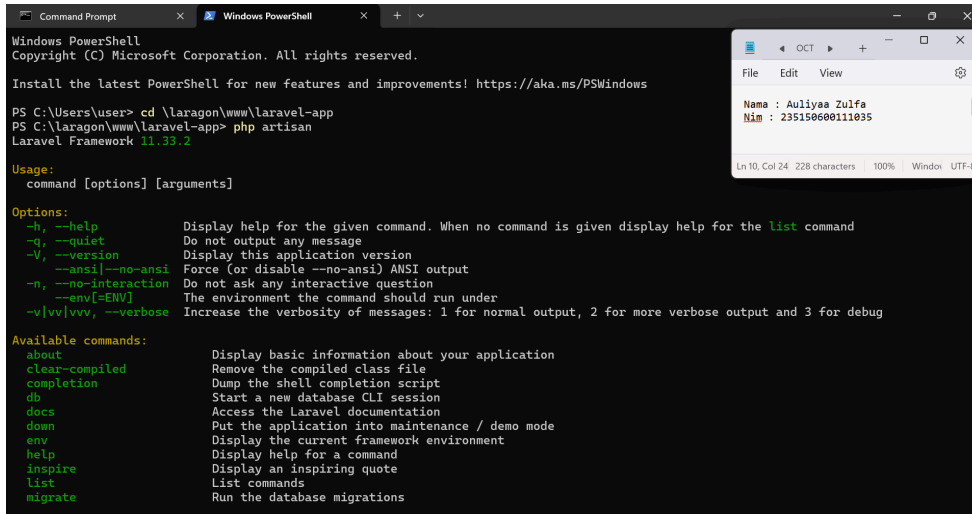
5. Buka alamat "laravel-app.test" pada browser. Tampilan seperti berikut akan muncul. Ini menandakan bahwa Laravel telah berhasil di-install.



#### d. artisan

1. Buka command prompt atau console dengan direktori aktif pada direktori project (<direktori XAMPP>\htdocs\laravel-app). Sesuaikan "<direktori XAMPP>" dengan direktori XAMPP pada komputer masing-masing. Pastikan juga bahwa file "php.exe" telah berada path sehinggaperintah "php" dapat diakses di mana saja.
2. Jalankan perintah berikut untuk memastikan bahwa akses ke Artisan dapat dilakukan.

<pre>php artisan</pre>
------------------------



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\User> cd \laragon\www\laravel-app
PS C:\laragon\www\laravel-app> php artisan
Laravel Framework 11.33.2

Usage:
  command [options] [arguments]

Options:
  -h, --help            Display help for the given command. When no command is given display help for the list command
  -q, --quiet           Do not output any message
  -V, --version          Display this application version
  --ansi|--no-ansi      Force (or disable --no-ansi) ANSI output
  -n, --no-interaction  Do not ask any interactive question
  --env[=ENV]           The environment the command should run under
  -v|vv|vvv, --verbose  Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about                Display basic information about your application
  clear-compiled        Remove the compiled class file
  completion           Dump the shell completion script
  db                   Start a new database CLI session
  docs                 Access the Laravel documentation
  down                 Put the application into maintenance / demo mode
  env                  Display the current framework environment
  help                 Display help for a command
  inspire              Display an inspiring quote
  list                 List commands
  migrate              Run the database migrations
  
```

#### d. controller

1. Jalankan perintah berikut untuk membuat controller baru dengan nama GreetController.

	<code>php artisan make:controller GreetController</code>
--	--

2. Edit kode tersebut (app/Http/Controllers/GreetController.php) sehingga menjadi seperti berikut.

	<pre> &lt;?php  namespace App\Http\Controllers;  use Illuminate\Http\Request;  class GreetController extends Controller {     public function greet(\$name) {         return view('greet', ['name' =&gt; \$name]);     } } </pre>
--	---

#### e. view dengan blade

1. Buat file baru “resources/views/greet.blade.php” dengan isi kode berikut.

	<code>&lt;h1&gt;Hello, {{ \$name }}!&lt;/h1&gt;</code>
--	--

## f. Routing

1. Buka file “routes/web.php”. Tambahkan kode berikut.

```
// ...  
use App\Http\Controllers\GreetController;  
// ...  
Route::get('/hello/{name}',  
[GreetController::class, 'greet']);
```

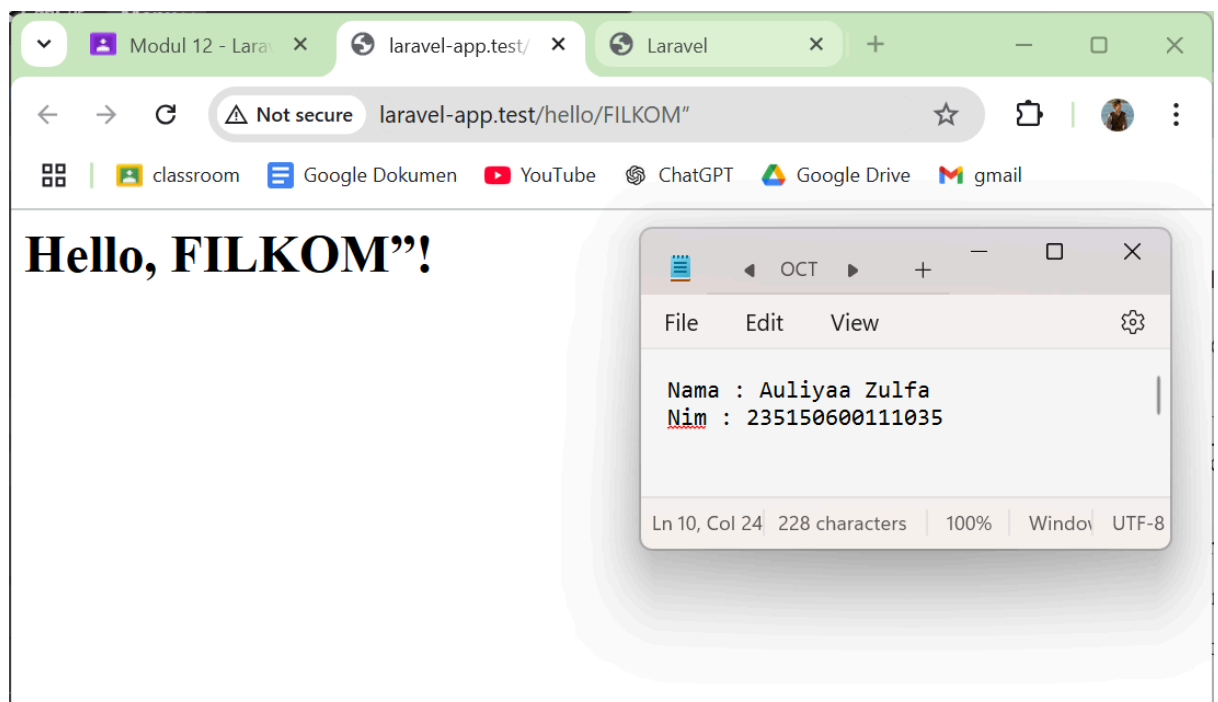
2. Buka URL “http://laravel-app.test/hello/FILKOM” di browser dan amati output yang dihasilkan.

Jelaskan yang dilakukan oleh fungsi greet() pada GreetController.

Jelaskan kode yang ditulis pada file “greet.blade.php”.

Jelaskan kode yang ditulis pada file “routes/web.php”.

## Screenshot



1. Fungsi greet(): Menerima parameter dari URL dan meneruskannya ke view.
2. File greet.blade.php: Menampilkan pesan personalisasi menggunakan data dari controller.

3. File routes/web.php: Mendefinisikan route untuk memetakan URL ke fungsi di controller.

### g. Blade Template

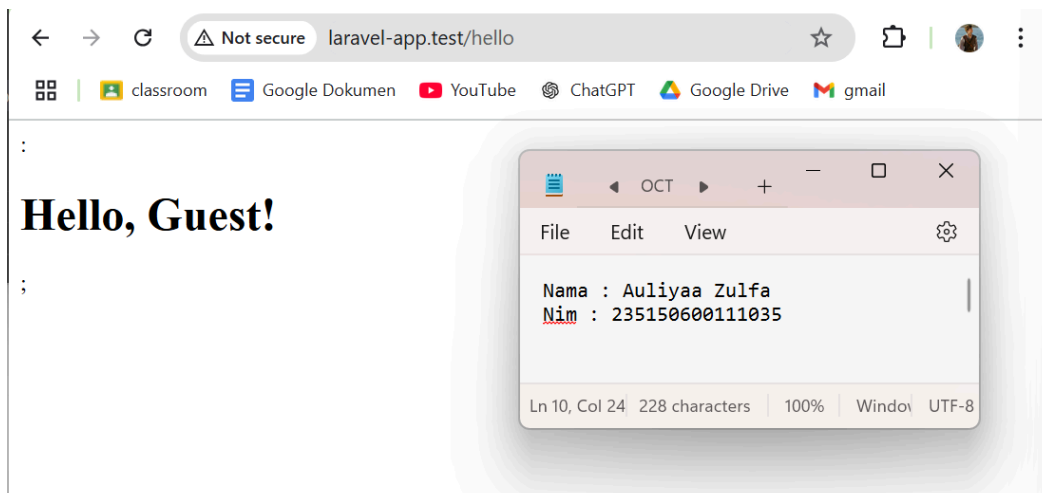
1. Tambahkan baris berikut pada file “routes/web.php”.

	<code>Route::view('/hello', 'greet');</code>
--	--

2. Ubah file “greet.blade.php” menjadi seperti berikut.

	<pre>@isset(\$name) :  &lt;h1&gt;Hello, {{ \$name }}!&lt;/h1&gt;  @else:  &lt;h1&gt;Hello, Guest!&lt;/h1&gt;  @endisset;</pre>
--	--

3. Buka URL “http://laravel-app.test/hello” di browser. Jelaskan perubahan yang dibuat pada dua file di atas. Bandingkan dengan URL “http://laravel-app.test/hello/FILKOM” yang diakses sebelumnya.



1. Perubahan pada routes/web.php:

Penambahan route dengan Route::view untuk memuat view secara langsung tanpa controller. Digunakan untuk kasus sederhana tanpa logika tambahan.

2. Perubahan pada greet.blade.php:

Penambahan pengecekan dengan @isset untuk menampilkan pesan default jika \$name tidak tersedia. Meningkatkan fleksibilitas dan mencegah kesalahan (undefined variable).

3. Perbandingan:

URL /hello: Menggunakan Route::view, tanpa parameter, menampilkan Hello, Guest!

URL /hello/FILKOM: Menggunakan route dengan parameter, menampilkan Hello, FILKOM!.