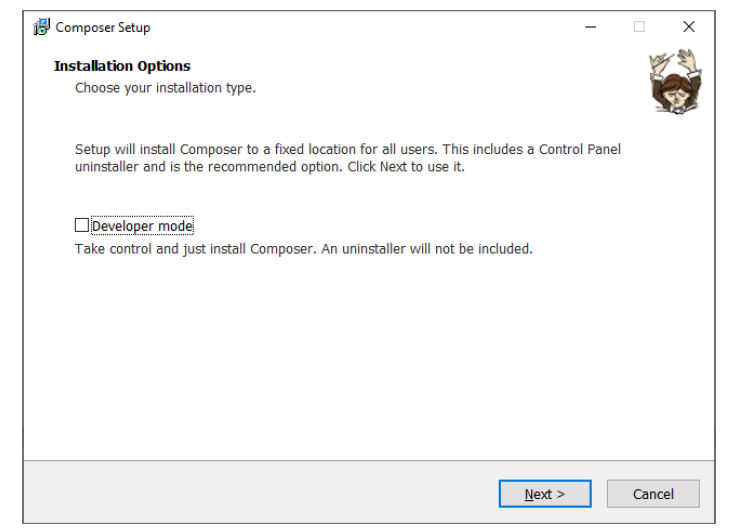
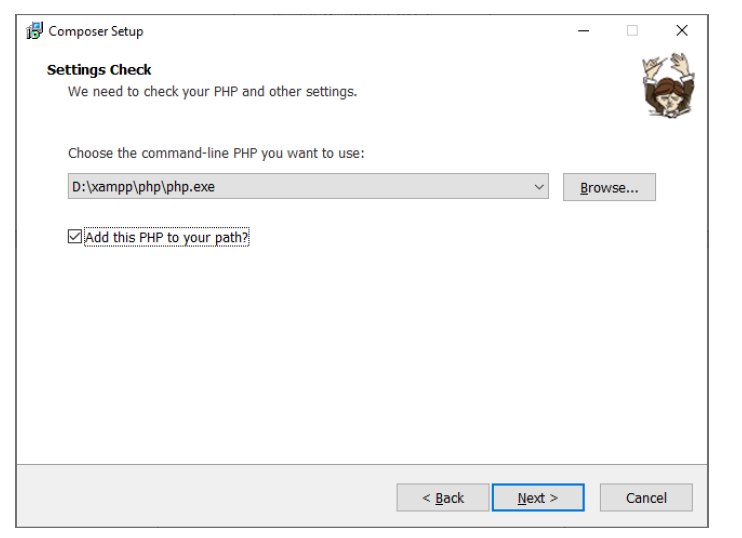
**LATIHAN 1**

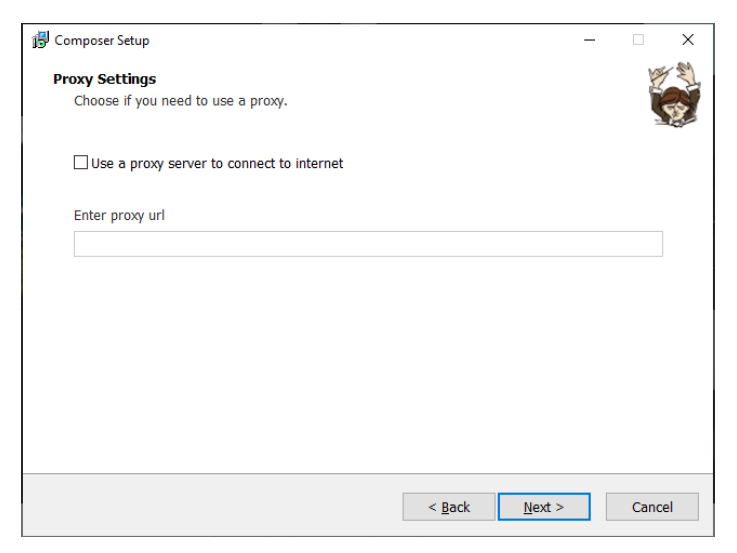
1. **Soal**
2. **Instalasi Composer**
   1. Instalasi *framework* Laravel dapat menggunakan Composer, sebuah *dependency* *manager* untuk PHP. *Download* dan lakukan instalasi Composer melalui Alamat <https://getcomposer.org/Composer-Setup.exe>.



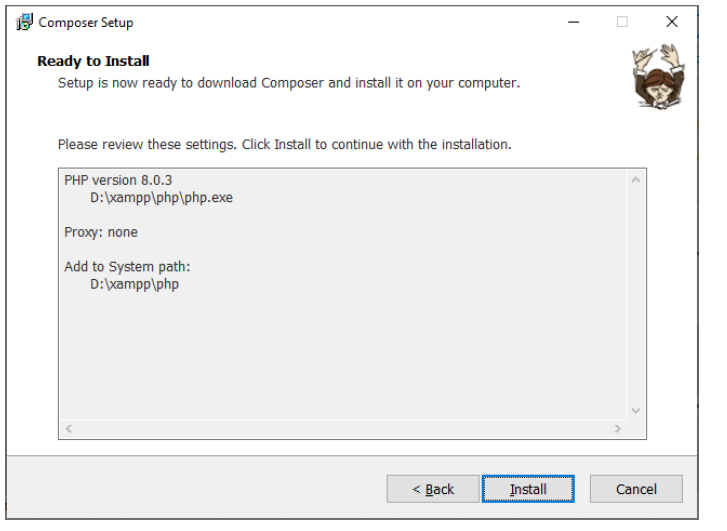
* 1. *Browse* dan pilih *file* “php.exe” yang ada pada direktori instalasi XAMPP. Centang pilihan “Add this PHP to your path?”. Klik “Next”.



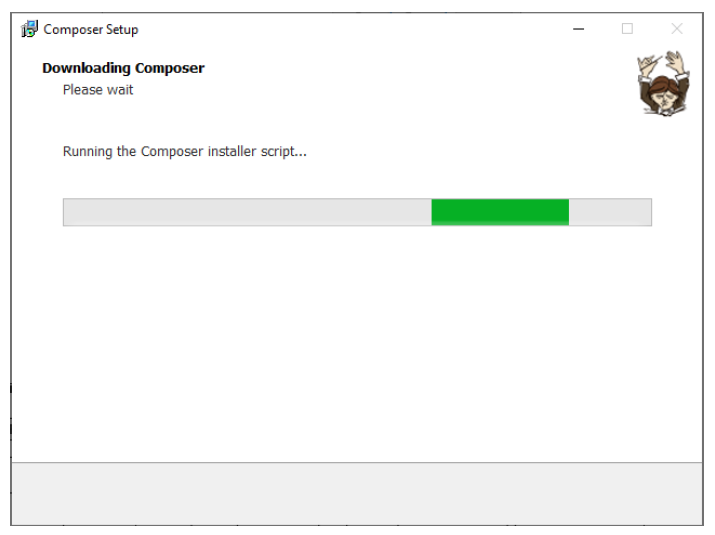
* 1. Kosongi pilihan “Use a proxy…” Klik “Next”



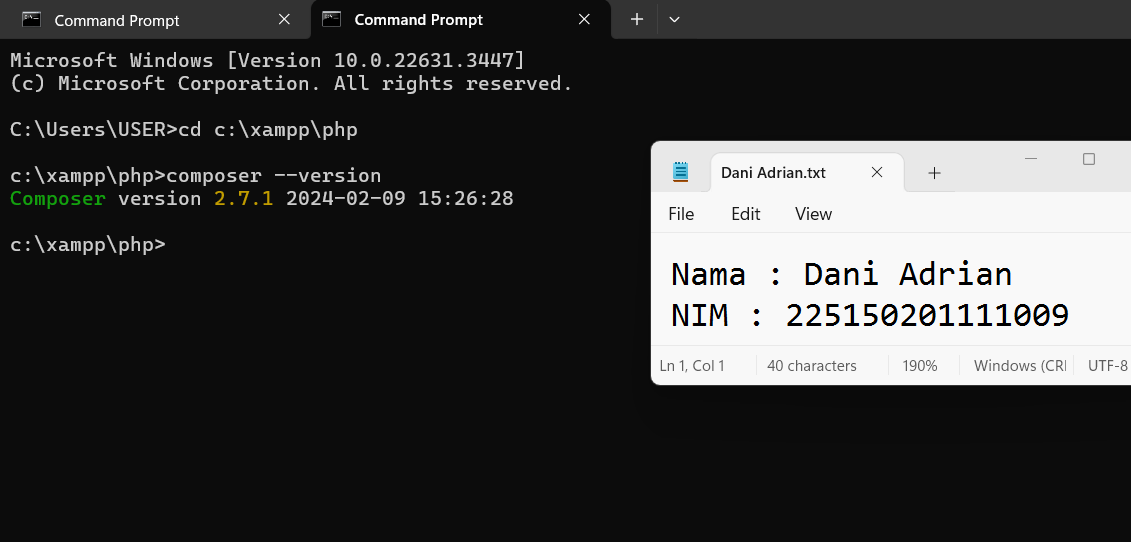
* 1. Klik “Install”.



* 1. Tunggu hingga proses instalasi selesai.



1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
|  |  |

1. **Penjelasan**

Compose telah berhasil di install

**LATIHAN 2**

1. **Soal**
2. **Instalasi Laravel**
   1. Setelah Composer berhasil di-*install*, instalasi Laravel dapat dilakukan sekaligus dengan pembuatan *project* Laravel menggunakan Composer. Pertama-tama, pastikan bahwa Composer dapat diakses pada *command prompt* atau *console* menggunakan perintah “composer” hingga menghasilkan tampilan seperti berikut.
   2. Arahkan direktori aktif pada *command prompt* atau *console* ke direktori “htdocs” pada direktori instalasi XAMPP. Lalu, jalankan perintah:

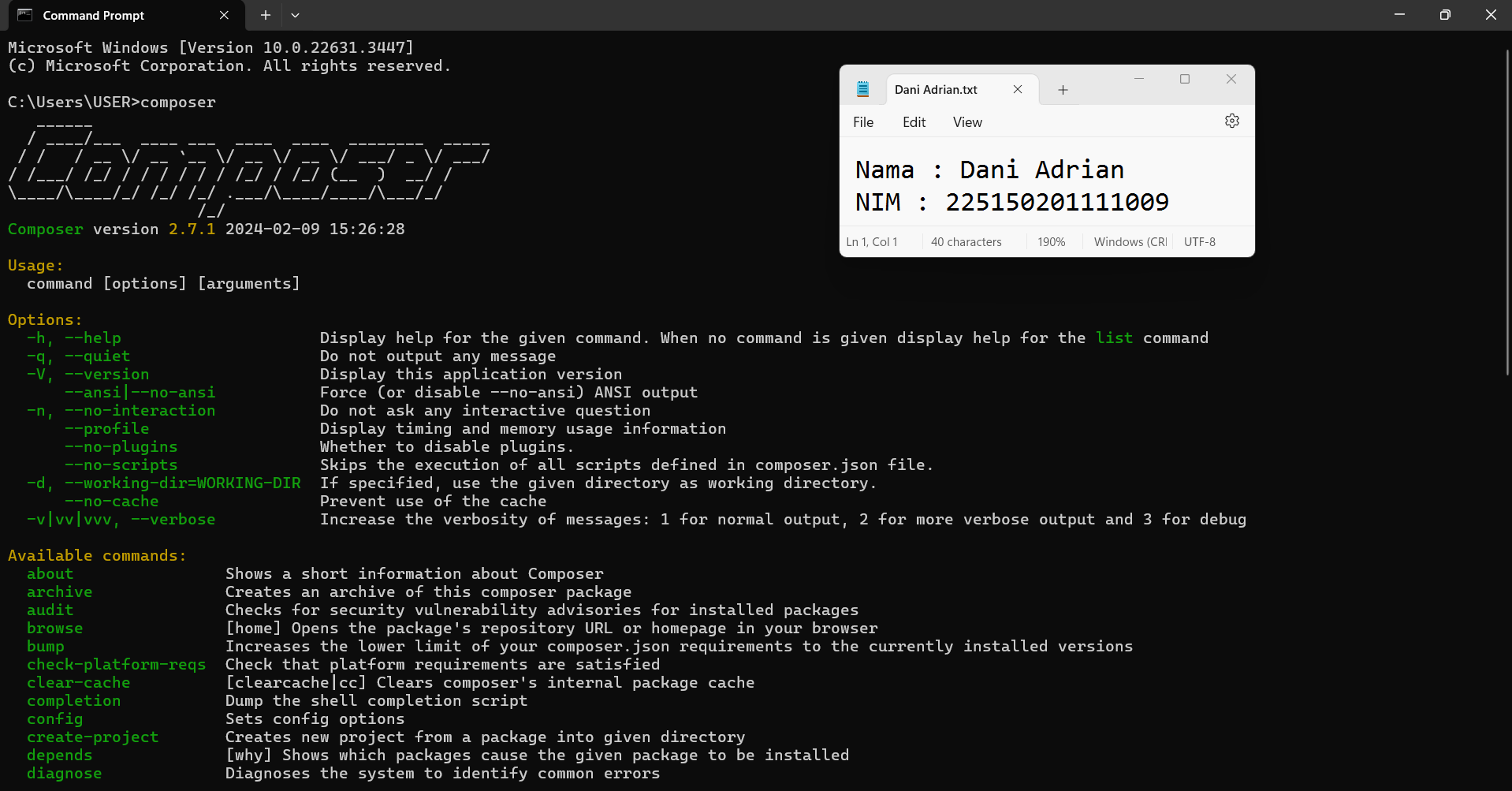
|  |
| --- |
| composer create project laravel/laravel Laravel-app |

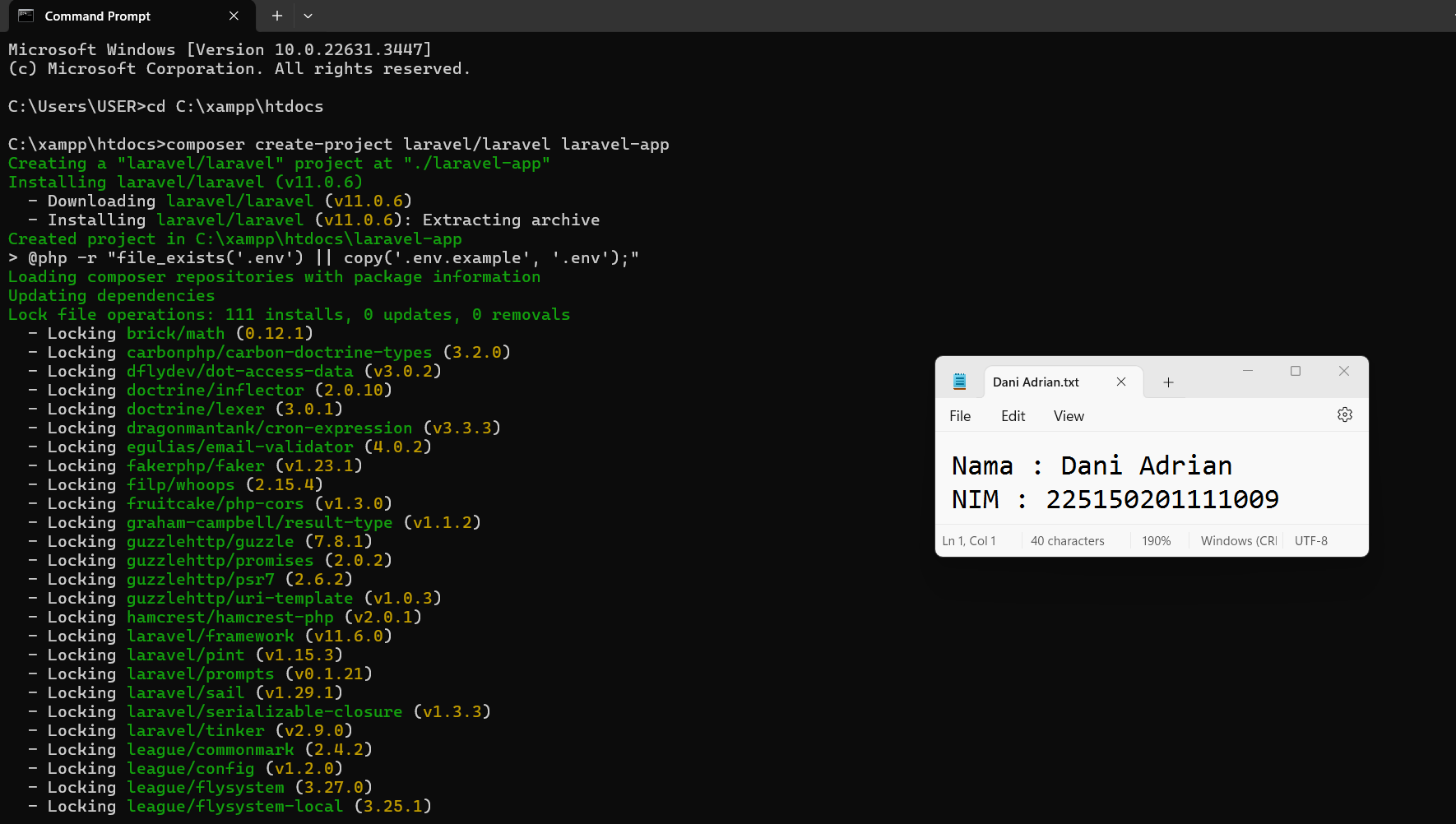
* 1. Jalankan perintah berikut menggunakan command prompt atau console dengan hak akses administrator:

|  |
| --- |
| echo 127.0.0.1 laravel-app.test >> C:\Windows\System32\drivers\etc\hosts |

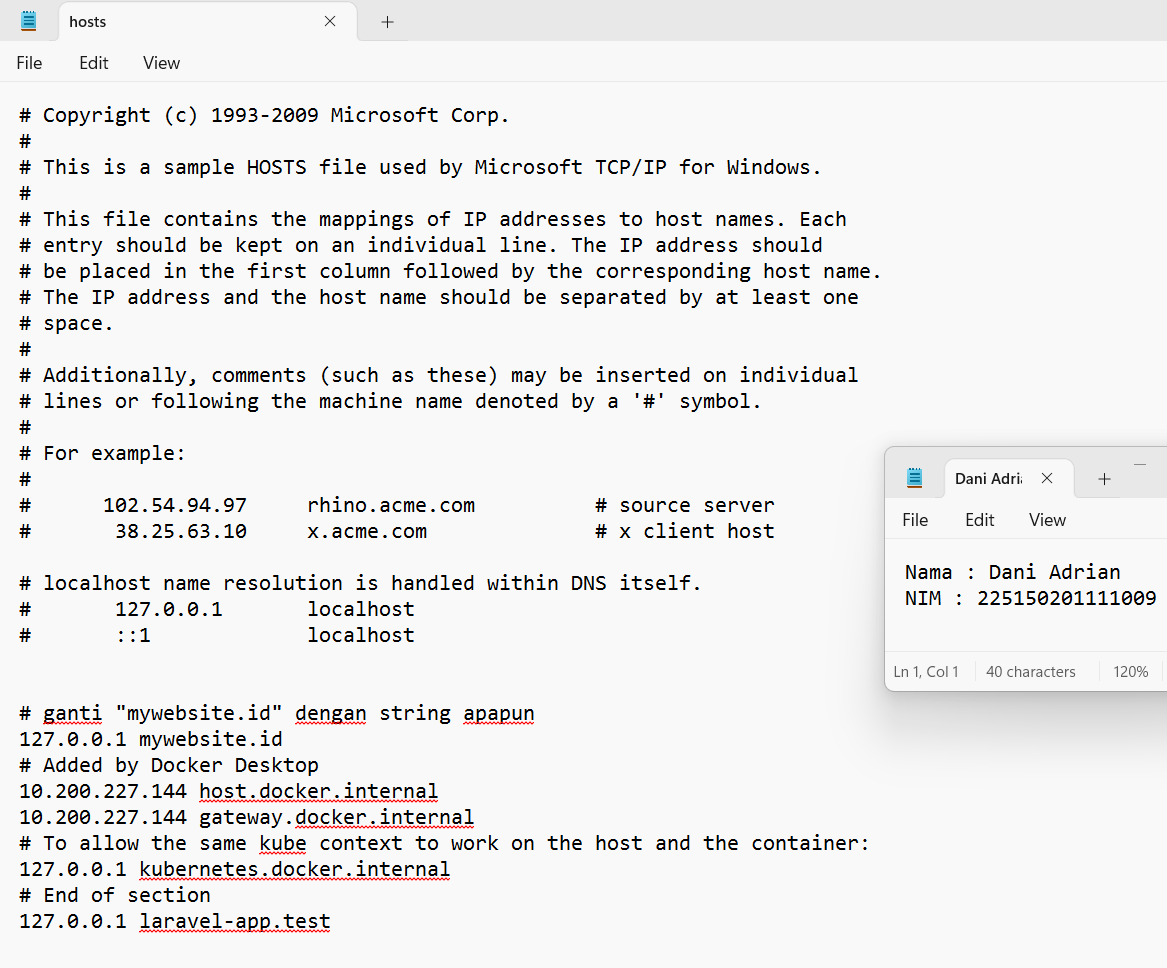
Jelaskan apa yang terjadi setelah perintah di atas dieksekusi dan mengapa perintah tersebut dijalankan.

1. **Screenshoot**









1. **Syntax**

|  |  |
| --- | --- |
| 1 |  |

1. **Penjelasan**

Ketika perintah tersebut dieksekusi. Maka, perintah tersebut akan langsung menuliskan 127.0.0.1 laravel-app.test *(Only Write (Unread))* pada direktori C:\Windows\System32\drivers\etc\hosts dengan tujuan file hosts.

**LATIHAN 3**

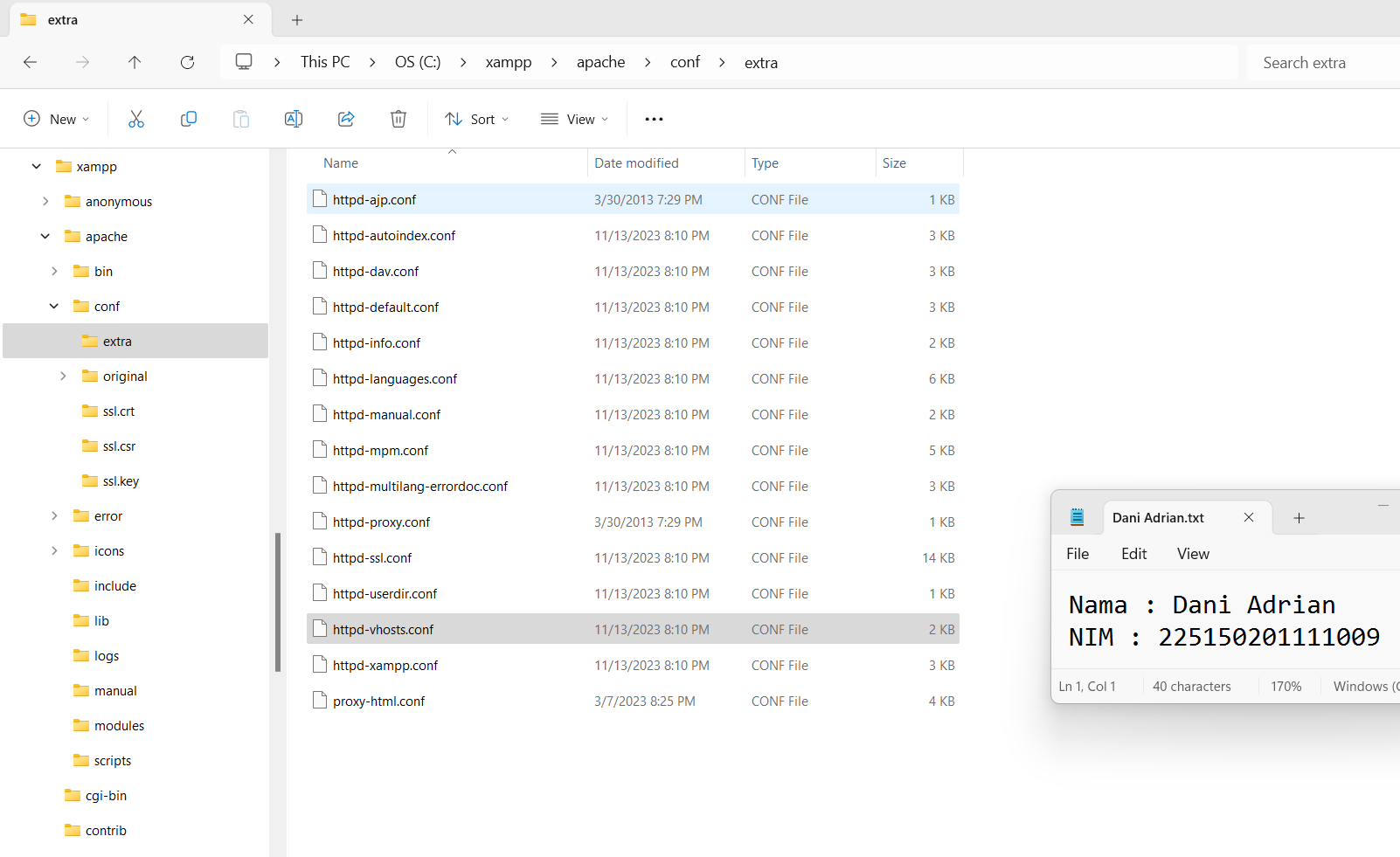
1. **Soal**
2. Buka file **“<direktori XAMPP>**\apache\conf\extra\httpd-vhosts.conf”. Tambahkan baris berikut pada akhir file tersebut:

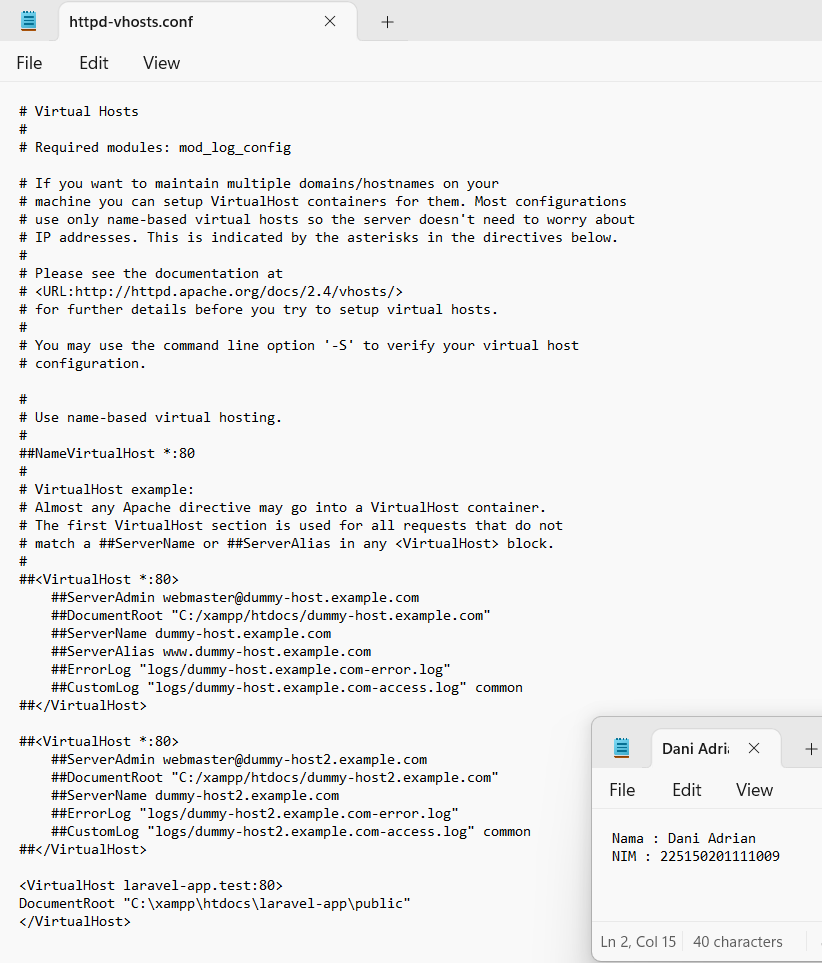
|  |  |
| --- | --- |
|  | <VirtualHost laravel-app.test:80> |
|  | DocumentRoot **"<direktori XAMPP>**/htdocs/laravel-app/public" |
|  | </VirtualHost> |

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.

1. **Screenshoot**

****



1. **Syntax**

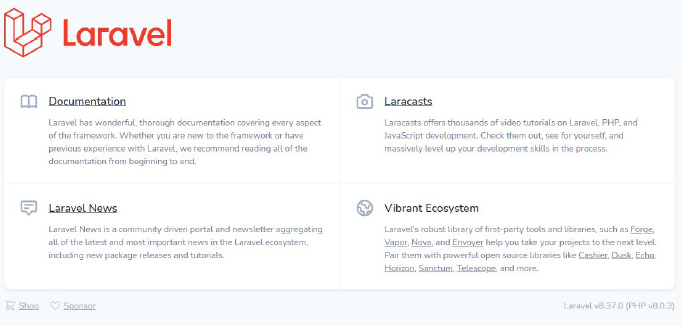
|  |  |
| --- | --- |
| 1 |  |

1. **Penjelasan**

Kode tersebut ditambahkan agar keluaran halaman dari pranala laravel-app.test sama dengan keluaran dari direktori C:\xampp\htdocs\laravel-app\public. Jangan lupa untuk melakukan restart pada XAMPP Control Panel agar bisa menghasilkan keluaran yang sesuai dan maksimal.

**LATIHAN 4**

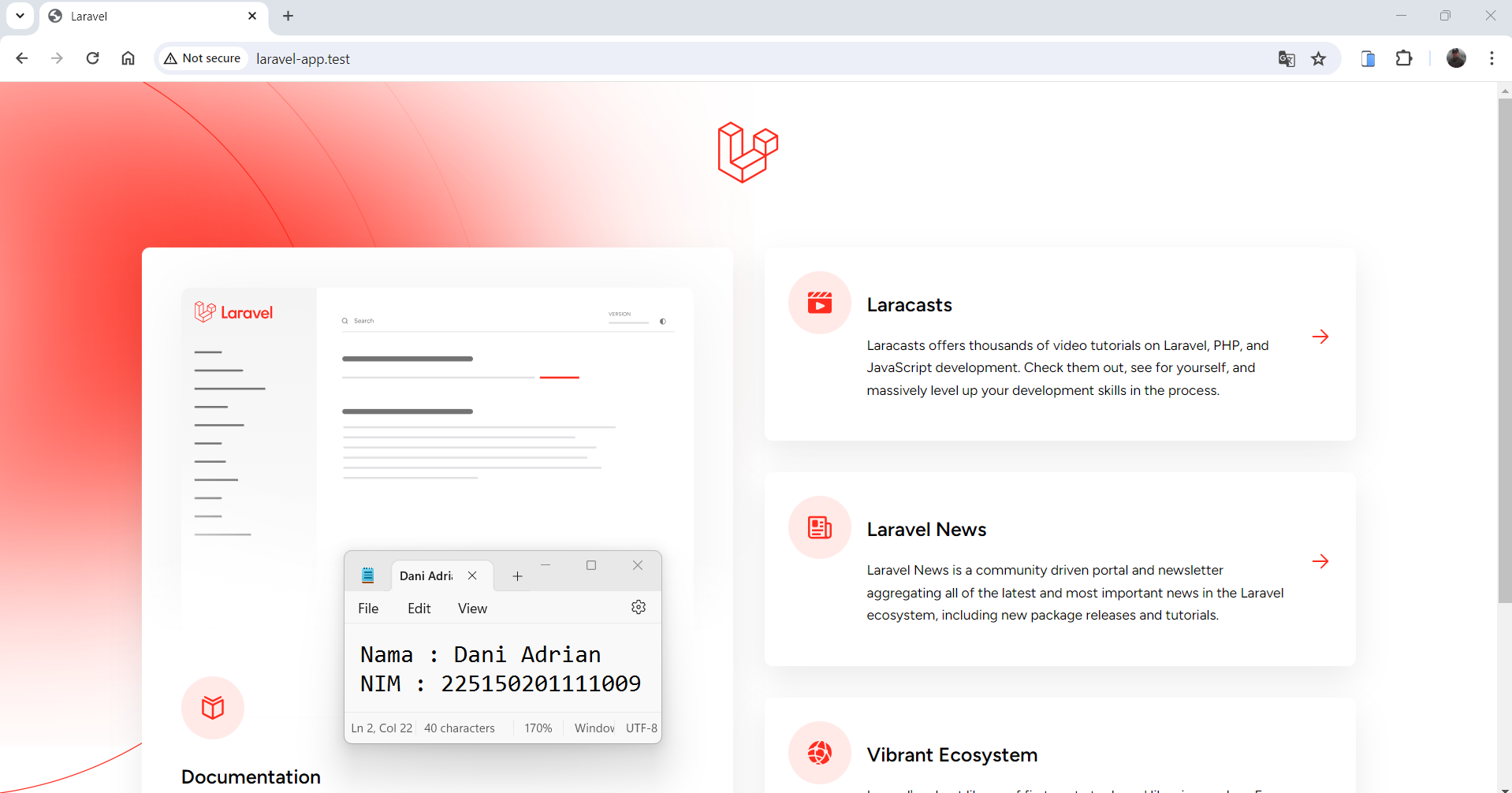
1. **Soal**
2. Buka alamat “laravel-app.test” pada *browser*. Tampilan seperti berikut akan muncul. Ini menandakan bahwa Laravel telah berhasil di-*install*.



Jika gagal, silakan coba beberapa hal berikut:

* 1. Pastikan tidak ada kesalahan penulisan.
  2. *Stop* lalu *start* Apache *web server*.

1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1 |  |

1. **Penjelasan**

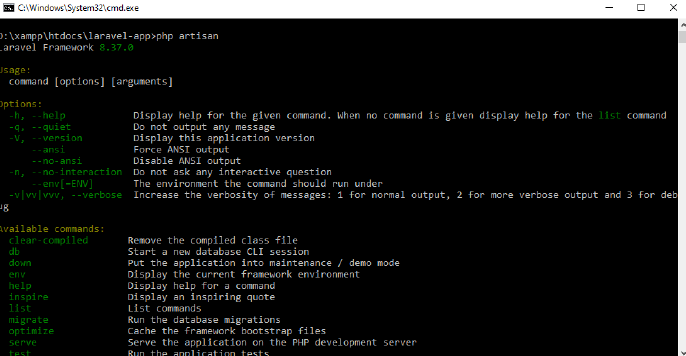
laravel-app.test akan mengeluarkan output halaman defaultnya laravel (Bukan halaman XAMPP).

**LATIHAN 5**

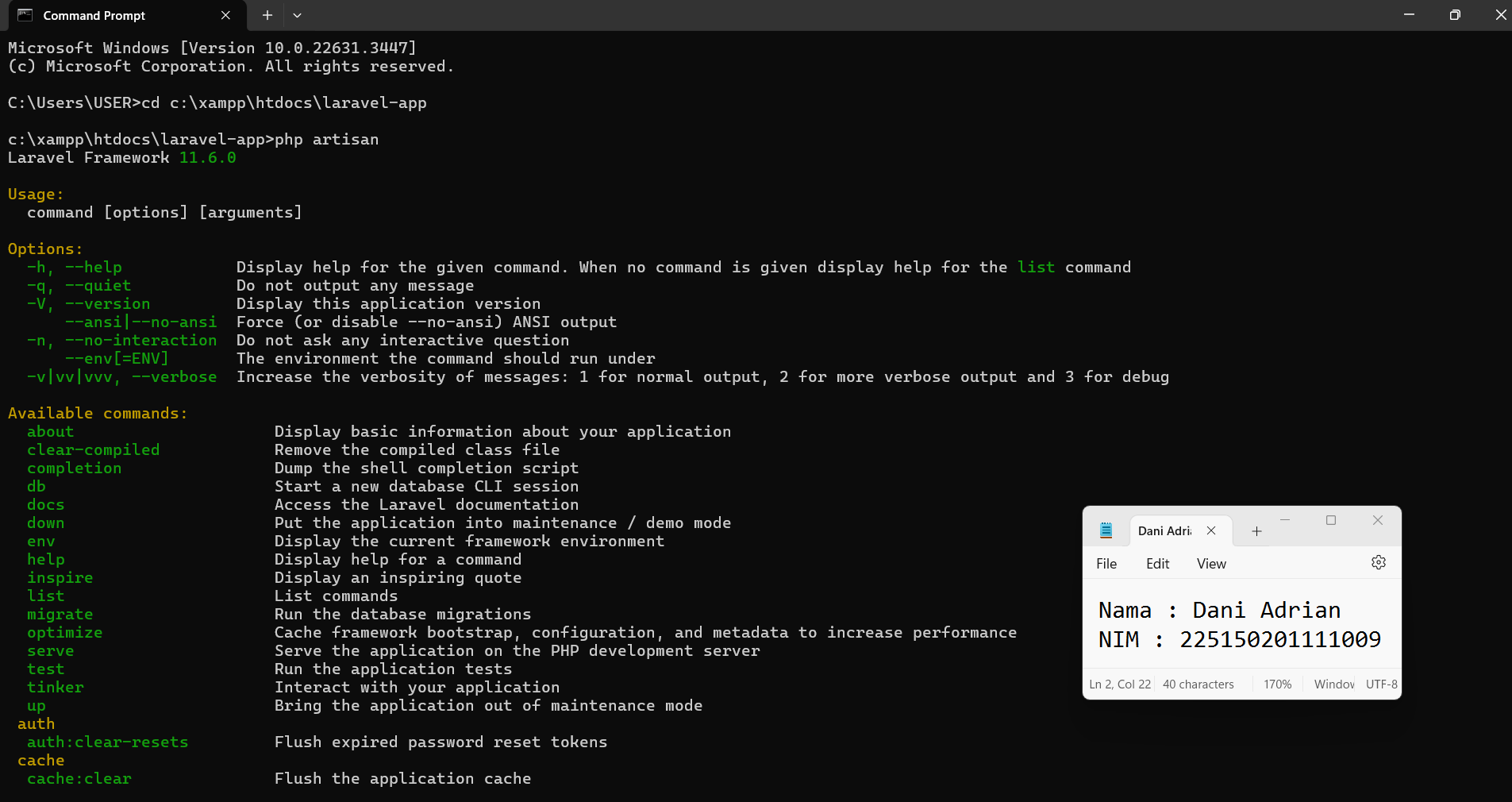
1. **Soal**
2. **Artisan**
   1. Buka command prompt atau console dengan direktori aktif pada direktori project **(<direktori XAMPP>\**htdocs\laravel-app). Sesuaikan “<direktori XAMPP>” dengan direktori XAMPP pada komputer masing-masing. Pastikan juga bahwa file “php.exe” telah berada *path* sehingga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

|  |
| --- |
| php artisan |

* 1. Jika berhasil, maka tampilannya adalah seperti berikut.



1. **Screenshoot**



1. **Syntax**

|  |  |
| --- | --- |
| 1 |  |

1. **Penjelasan**

Berhasil dieksekusi

**LATIHAN 6**

1. **Soal**
2. **Controller**
   1. Jalankan perintah berikut untuk membuat controller baru dengan nama GreetController.

|  |  |
| --- | --- |
|  | php artisan make:controller GreetController |

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

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

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

|  |  |
| --- | --- |
|  | < h1>Hello, {{ $name }}!</h1> |

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

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

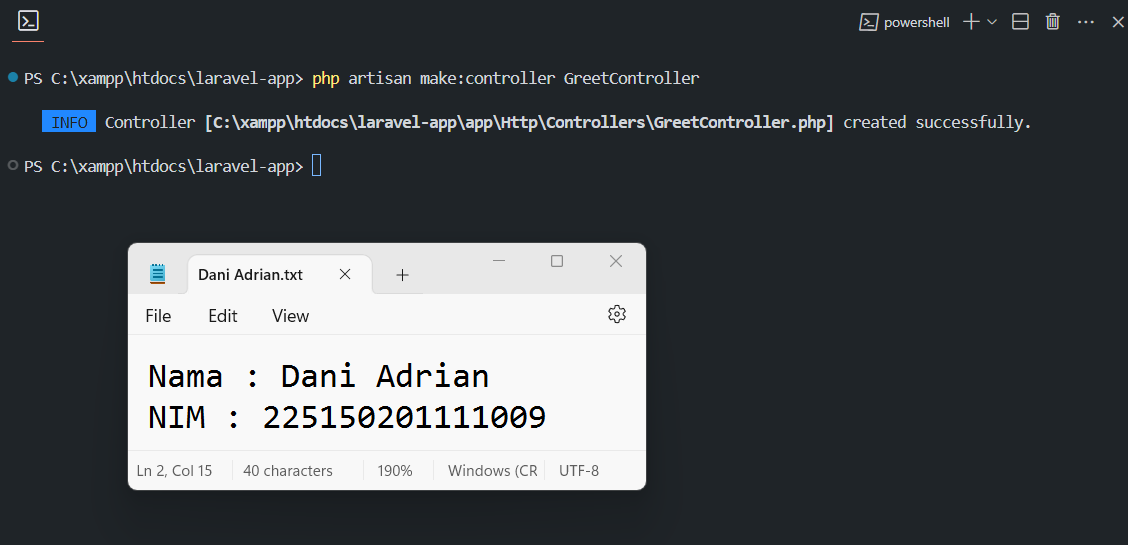
* 1. Buka URL “http://laravel-app.test/hello/FILKOM” di *browser* dan amati *output* yang dihasilkan.

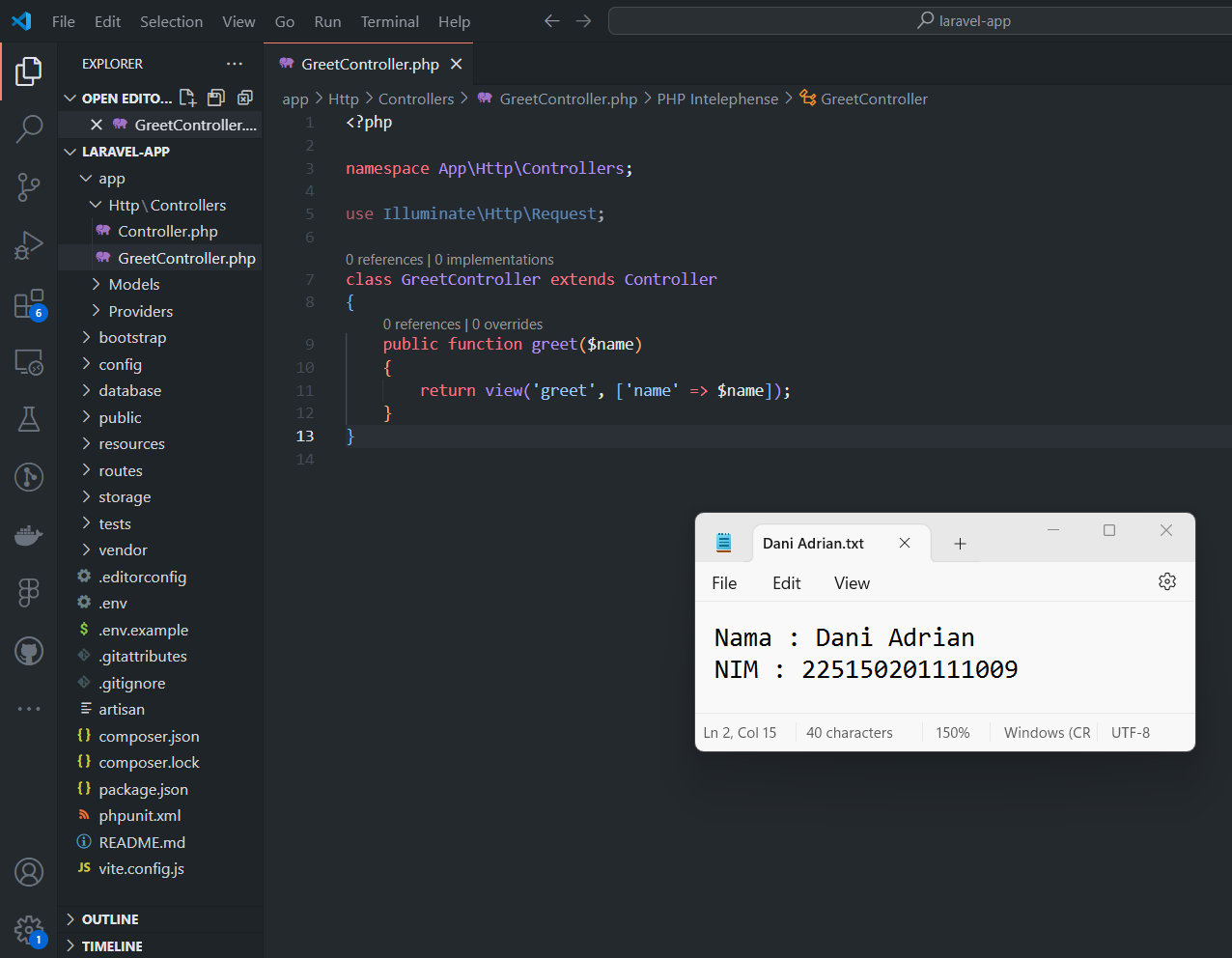
Jelaskan yang dilakukan oleh fungsi greet() pada GreetController.

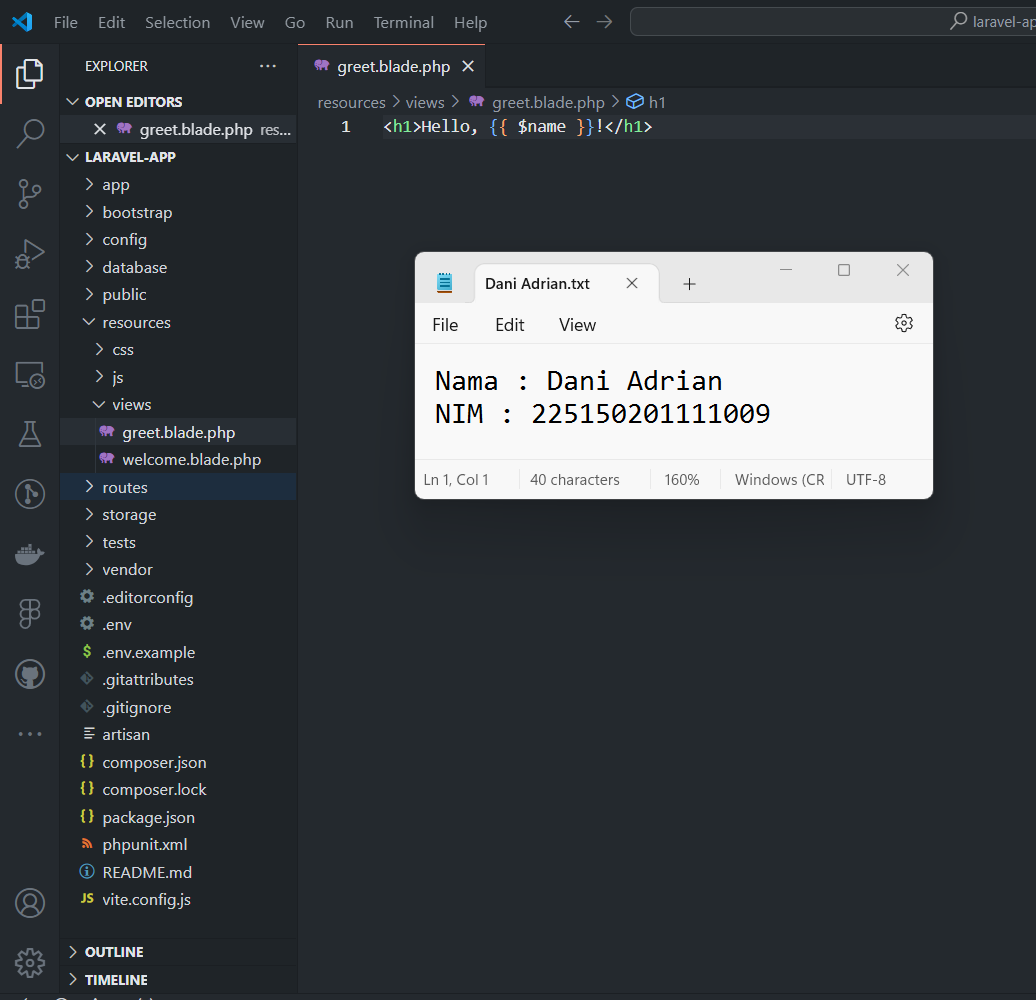
Jelaskan kode yang ditulis pada *file* “greet.blade.php”.

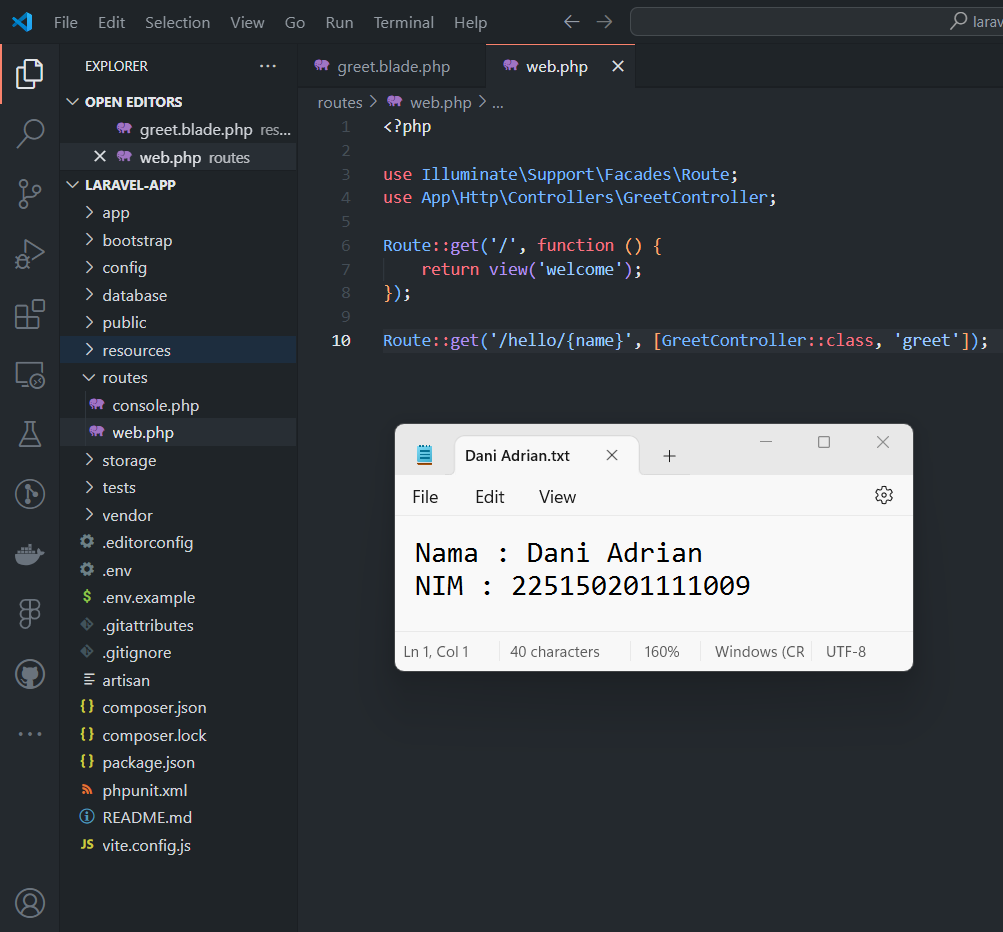
Jelaskan kode yang ditulis pada *file* “routes/web.php”.

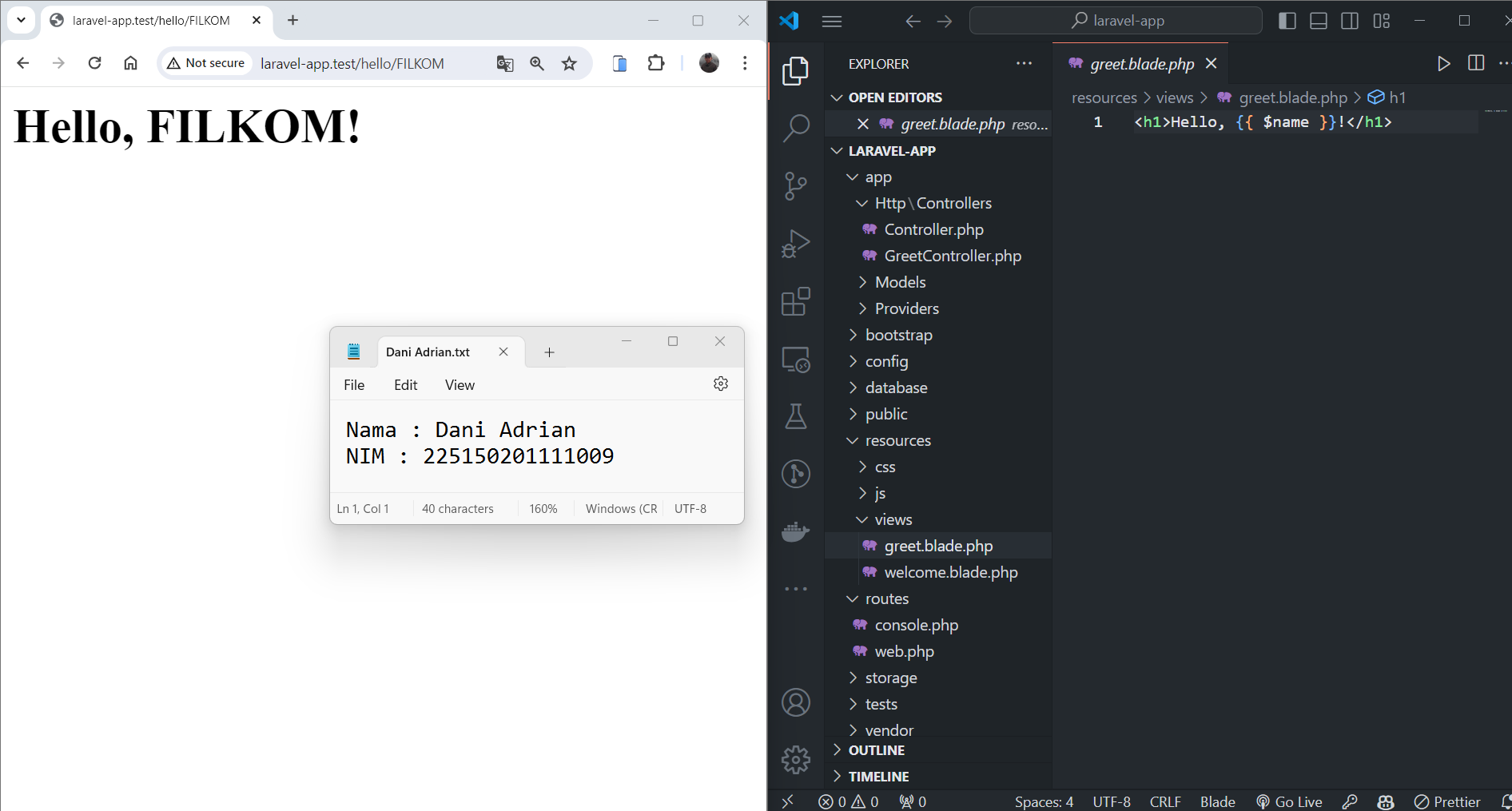
1. **Screenshoot**











1. **Syntax**

|  |  |
| --- | --- |
| GreetController.php | |
| 1. | <?php |
| 2. |  |
| 3. | namespace App\Http\Controllers; |
| 4. |  |
| 5. | use Illuminate\Http\Request; |
| 6. |  |
| 7. | class GreetController extends Controller |
| 8. | { |
| 9. | public function greet($name) |
| 10. | { |
| 11. | return view('greet', ['name' => $name]); |
| 12. | } |
| 13. | } |

|  |  |
| --- | --- |
| greet.blade.php | |
| 1. | < h1>Hello, {{ $name }}!</h1> |

|  |  |
| --- | --- |
| web.php | |
| 1. | // ... |
| 2. | use App\Http\Controllers\GreetController; |
| 3. | // ... |
| 4. | Route::get('/hello/{name}', [GreetController::class, 'greet']); |

1. **Penjelasan**

Function greet($name) pada GreetController.php digunakan untuk menampilkan output keluaran sesuai dengan kode pada greet.blade.php yaitu Hello, $name!. Variabel name ini selain digunakan pada greet.blade.php, juga digunakan pada pranala yang akan diketik. Contohnya, laravel-app.test/hello/filkom yang akan mengeluarkan keluaran Hello, filkom!.

greet.blade.php digunakan sebagai kode sisipan atau skrip teks yang sebenarnya. Hello, $name! sebagai keluaran yang pasti melalui kode GreatController.php dan web.php.

web.php digunakan untuk melakukan rute atau penujuan pada pranala halaman yang diinginkan. Pada /hello/{name}. Maka dengan pranala laravel-app.test/hello/{name} akan menjalankan kode pada GreetController.php (Penggunaan use membantu agar kode pada GreatController.php dapat dengan teliti dan mudah ditangkap).

**LATIHAN 7**

1. **Soal**
2. **Blade template**
   1. Tambahkan baris berikut pada *file* “routes/web.php”.

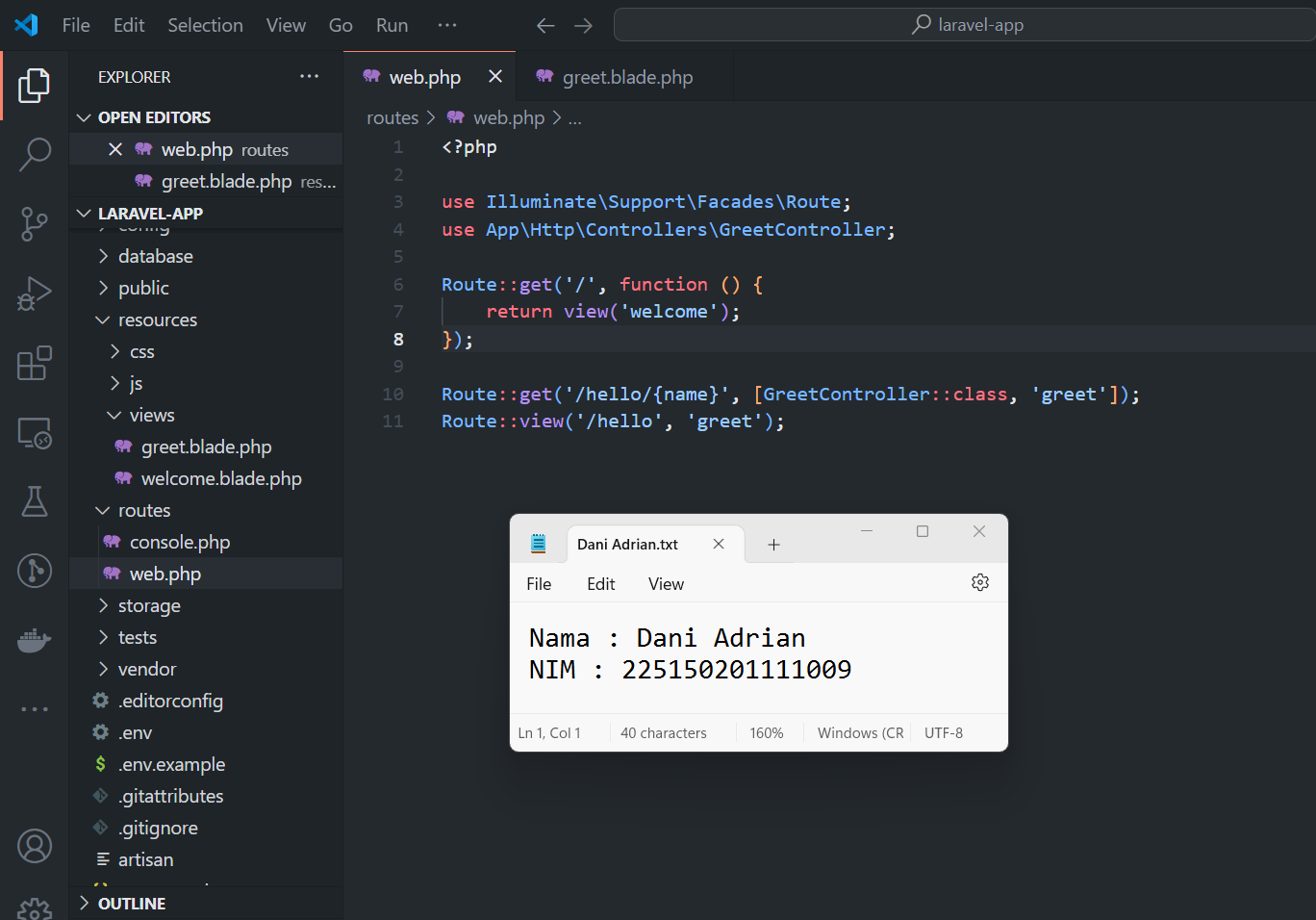
|  |  |
| --- | --- |
|  | Route::view('/hello', 'greet'); |

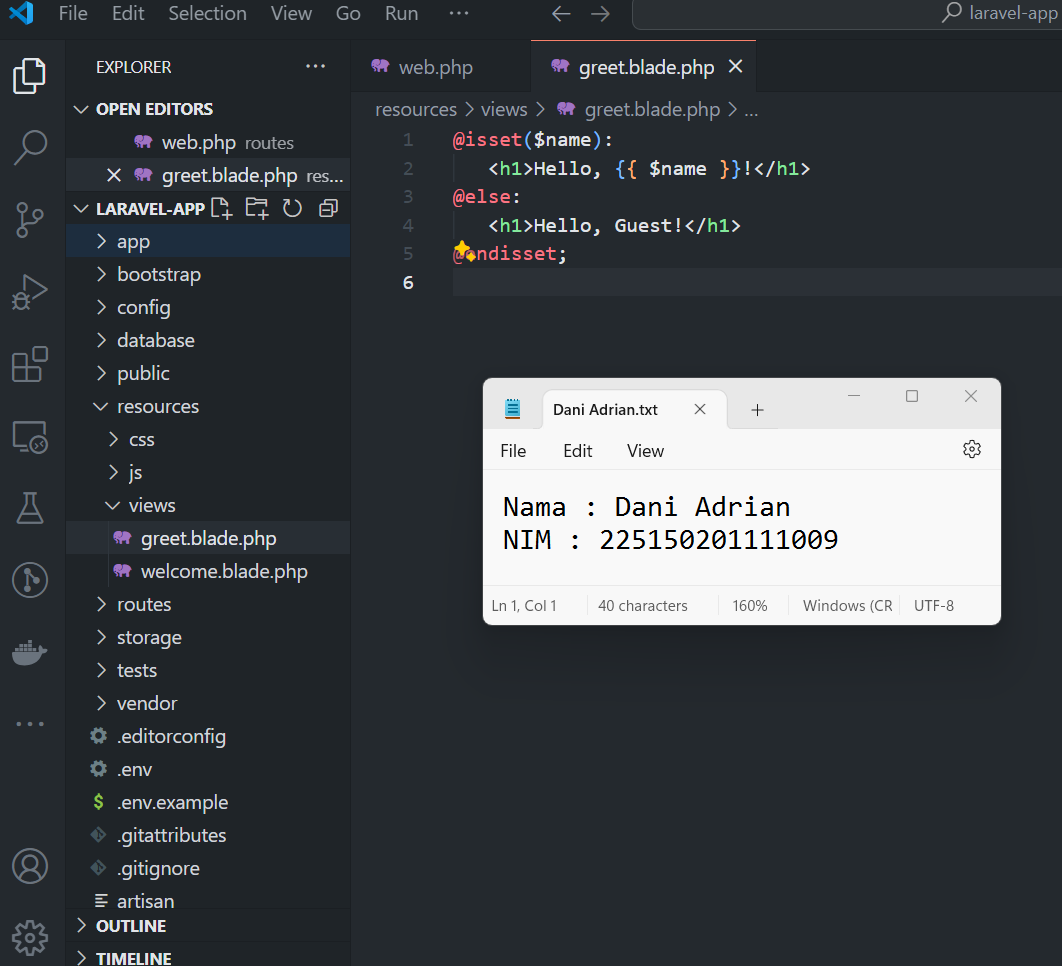
* 1. Ubah file “greet.blade.php” menjadi seperti berikut.

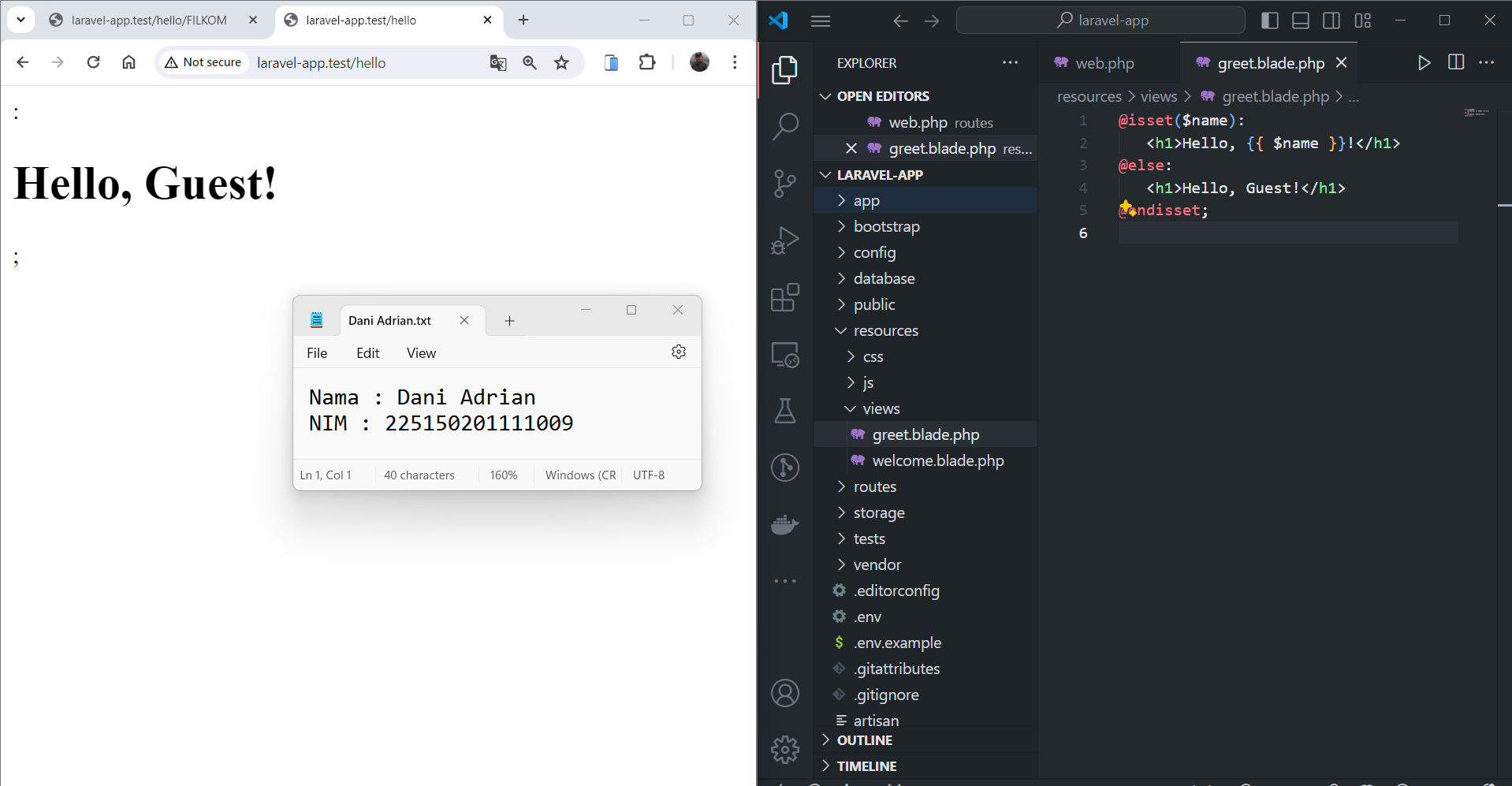
|  |  |
| --- | --- |
|  | @isset($name): |
|  | <h1>Hello, {{ $name }}!</h1> |
|  | @else: |
|  | <h1>Hello, Guest!</h1> |
|  | @endisset; |

* 1. 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. **Screenshoot**







1. **Syntax**

|  |  |
| --- | --- |
| web.php | |
| 1. | Route::view('/hello', 'greet'); |

|  |  |
| --- | --- |
| greet.blade.php | |
|  | @isset($name): |
|  | <h1>Hello, {{ $name }}!</h1> |
|  | @else: |
|  | <h1>Hello, Guest!</h1> |
|  | @endisset; |

1. **Penjelasan**

laravel-app.test/hello dan laravel-app.test/hello/filkom bisa berbeda hanya gara-gara if else pada greet.blade.php. Bila ada variabel name akan mengeluarkan luaran Hello, $name! dan bila tidak ada variabel name ataupun apapun akan mengeluarkan luaran Hello, Guest!. web.php kemudian menyediakan rute yaitu laravel-app.test/hello (mengoutputkan yang tidak ada variabel namenya) dan laravel-app.test/hello/$name (mengoutputkan yang ada variabel namenya).