

Modul 10 : *Framework* Laravel (1)

10.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- a. 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- b. 60 menit untuk penyampaian materi
- c. 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- d. 50 menit **pengayaan**

10.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Melakukan instalasi *framework* Laravel
2. Membuat website menggunakan *framework* Laravel

10.3 Dasar Teori

a) *Framework* Laravel

Laravel adalah sebuah *framework* aplikasi *web* yang sangat populer. Banyak *website* yang dibangun menggunakan *framework* ini. Laravel dipilih karena beberapa kelebihanannya, di antaranya yaitu fitur-fiturnya yang lengkap, seperti autentikasi, otorisasi, *command line interface*, dan *templating engine*. Seperti banyak *framework-framework* yang lain, Laravel dapat di-*download* dan digunakan secara gratis.

Framework Laravel menggunakan pola model-view-controller (MVC). Website dari *framework* Laravel beralamat di <https://www.laravel.com>. Pada *website* tersebut, tersedia segala sesuatu mengenai *framework* Laravel, mulai dari tutorial, dokumentasi, hingga informasi terbaru *framework* tersebut.

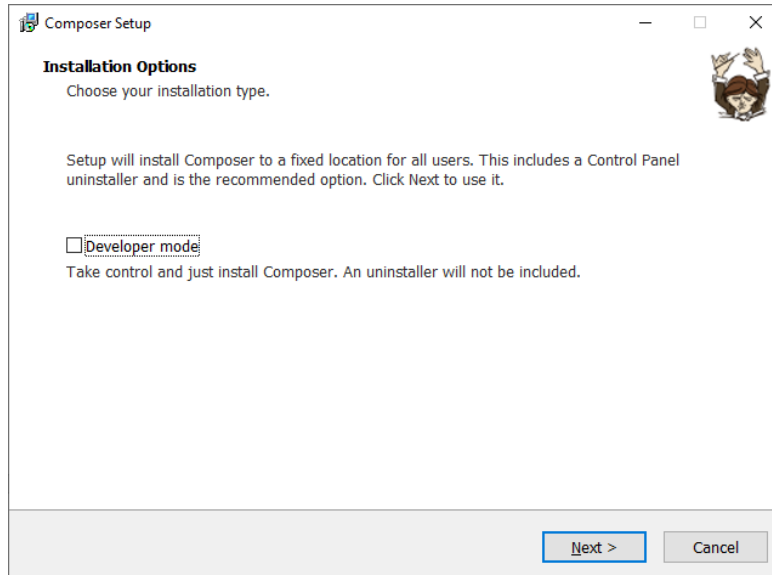
Pada saat modul praktikum ini ditulis, versi terbaru Laravel adalah versi 8 yang membutuhkan PHP versi 7.3 ke atas. Beberapa *file* dan direktori penting pada direktori project Laravel adalah sebagai berikut.

File/Direktori	Keterangan

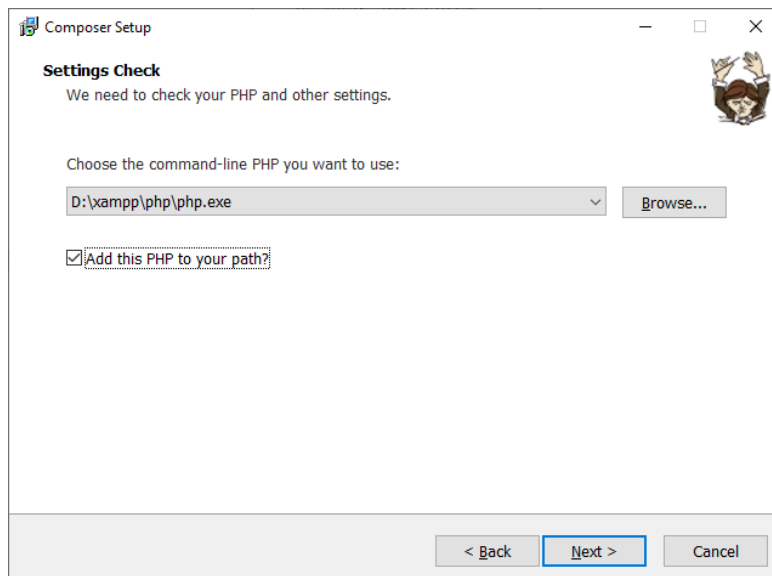
10.4 Prosedur Praktikum

a) Instalasi Composer

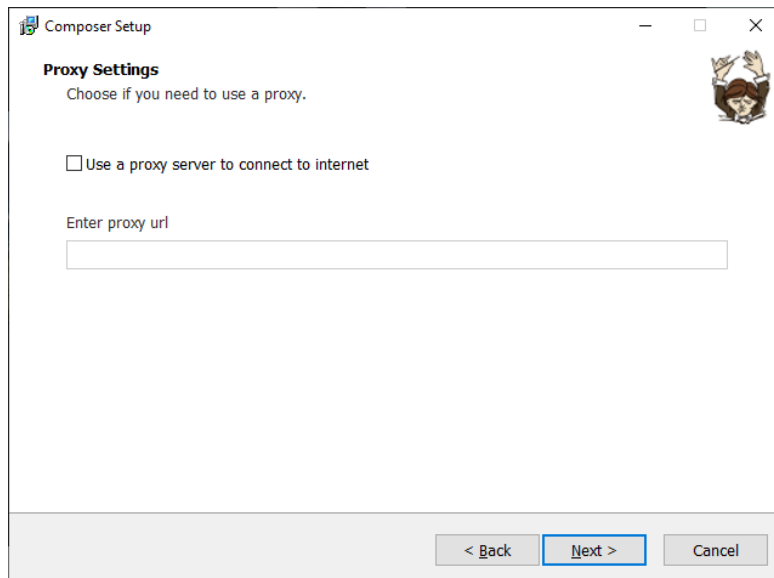
1. Instalasi *framework* Laravel dapat dilakukan menggunakan Composer, sebuah *dependency manager* untuk PHP. *Download* dan lakukan instalasi Composer melalui alamat <https://getcomposer.org/Composer-Setup.exe>.



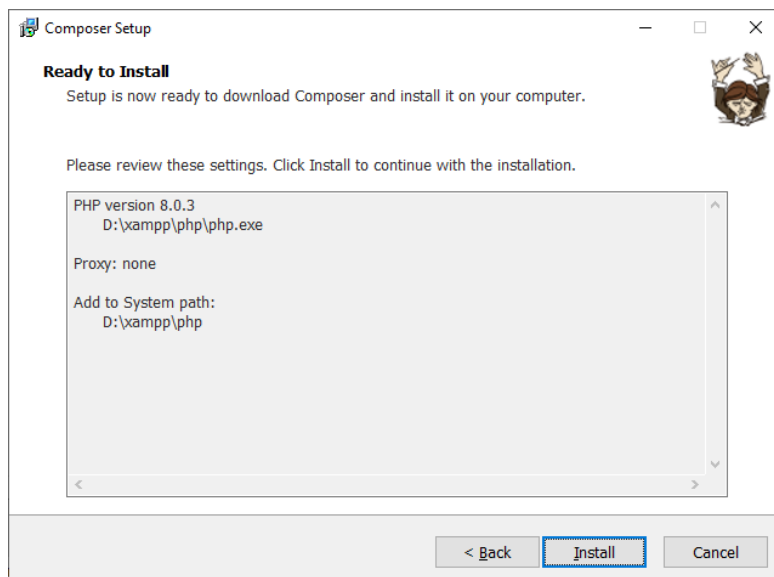
2. *Browse* dan pilih file "php.exe" yang ada pada direktori instalasi XAMPP. Centang pilihan "Add this PHP to your path?". Klik "Next".



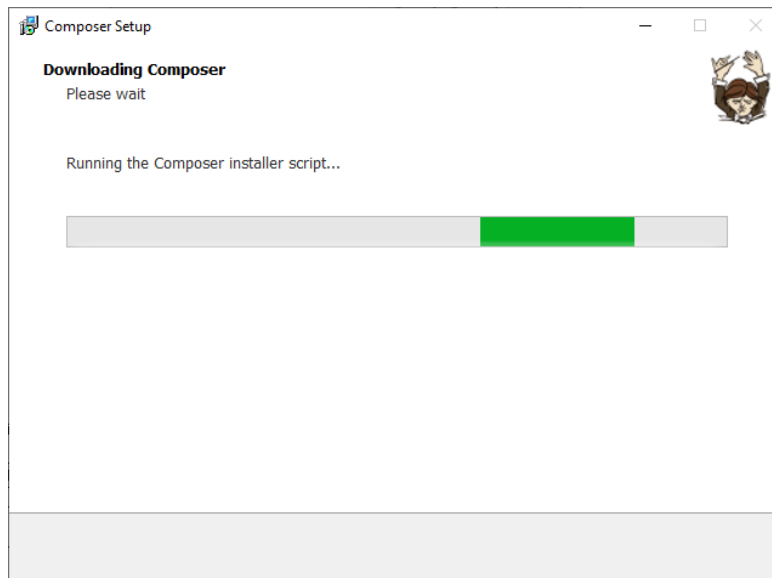
3. Kosongi pilihan "Use a proxy..." Klik "Next"



4. Klik "Install".



5. Tunggu hingga proses instalasi selesai.



b) 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.

```
C:\Users\Randy>composer

Composer version 2.0.12 2021-04-01 10:14:59

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 question
      --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

Available commands:
  about      Shows the short information about Composer.
  archive    Creates an archive of this composer package.
```

- 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
```

Perintah di atas akan mengunduh Laravel sekaligus membuat project Laravel dengan nama “laravel-app”. Tunggu hingga proses *download* selesai. Setelah selesai, akan ada direktori baru, yaitu “laravel-app”, sesuai dengan nama *project* yang diberikan.

3. 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.

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

```
1 <VirtualHost laravel-app.test:80>
2     DocumentRoot "<direktori XAMPP>/htdocs/laravel-app/public"
3 </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.

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



[Documentation](#)

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.



[Laracasts](#)

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.



[Laravel News](#)

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.



[Vibrant Ecosystem](#)

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), and [Envoyer](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

[Shop](#) [Sponsor](#)

Laravel v8.37.0 (PHP v8.0.3)

Jika gagal, silakan coba beberapa hal berikut:

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

c) 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
```

Jika berhasil, maka tampilannya adalah seperti berikut.

```
C:\Windows\System32\cmd.exe

D:\xampp\htdocs\laravel-app>php artisan
Laravel Framework 8.37.0

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           Force ANSI output
      --no-ansi        Disable 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:
  clear-compiled  Remove the compiled class file
  db             Start a new database CLI session
  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
  optimize      Cache the framework bootstrap files
  serve        Serve the application on the PHP development server
  test         Run the application tests
```

d) **Controller**

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

```
php artisan make:controller GreetController
```

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

```
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         return view('greet', ['name' => $name]);
11     }
12 }
```

e) **View dengan Blade**

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

```
1  <h1>Hello, {{ $name }}!</h1>
```

f) **Routing**

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

```
1  // ...
2  use App\Http\Controllers\GreetController;
3
4  // ...
5  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`”.

g) *Blade Template*

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

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

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

2	<code>@isset(\$name):</code>
3	<code><h1>Hello, {{ \$name }}!</h1></code>
4	<code>@else:</code>
5	<code><h1>Hello, Guest!</h1></code>
6	<code>@endisset;</code>

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.
