

LAPORAN PRAKTIKUM PEMROGRAMAN WEB LANJUT

Jobsheet-9

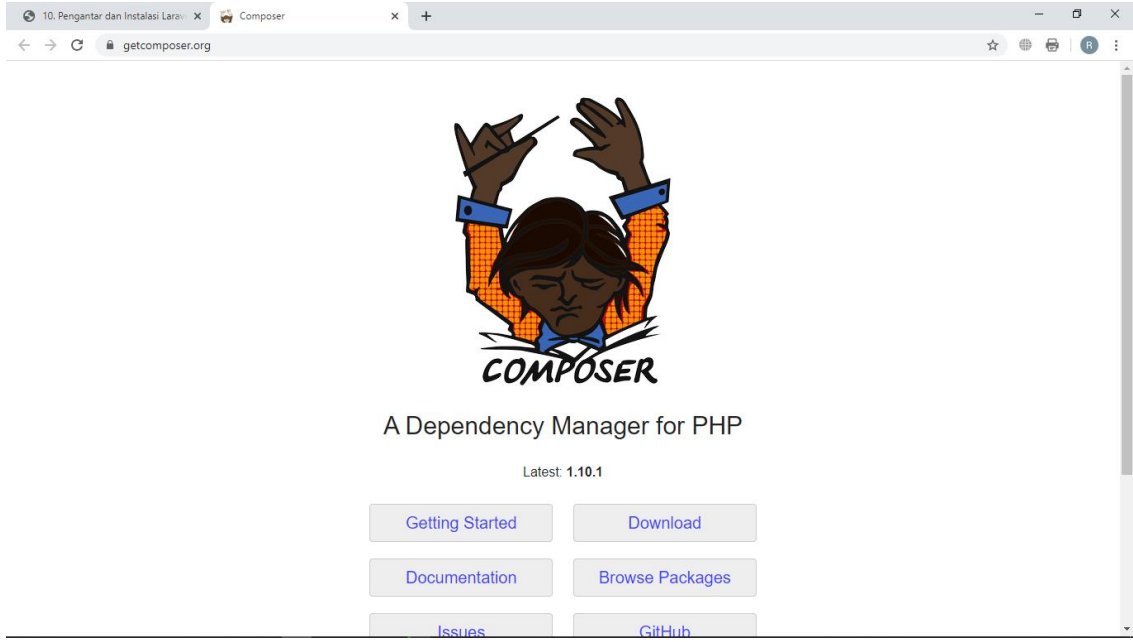


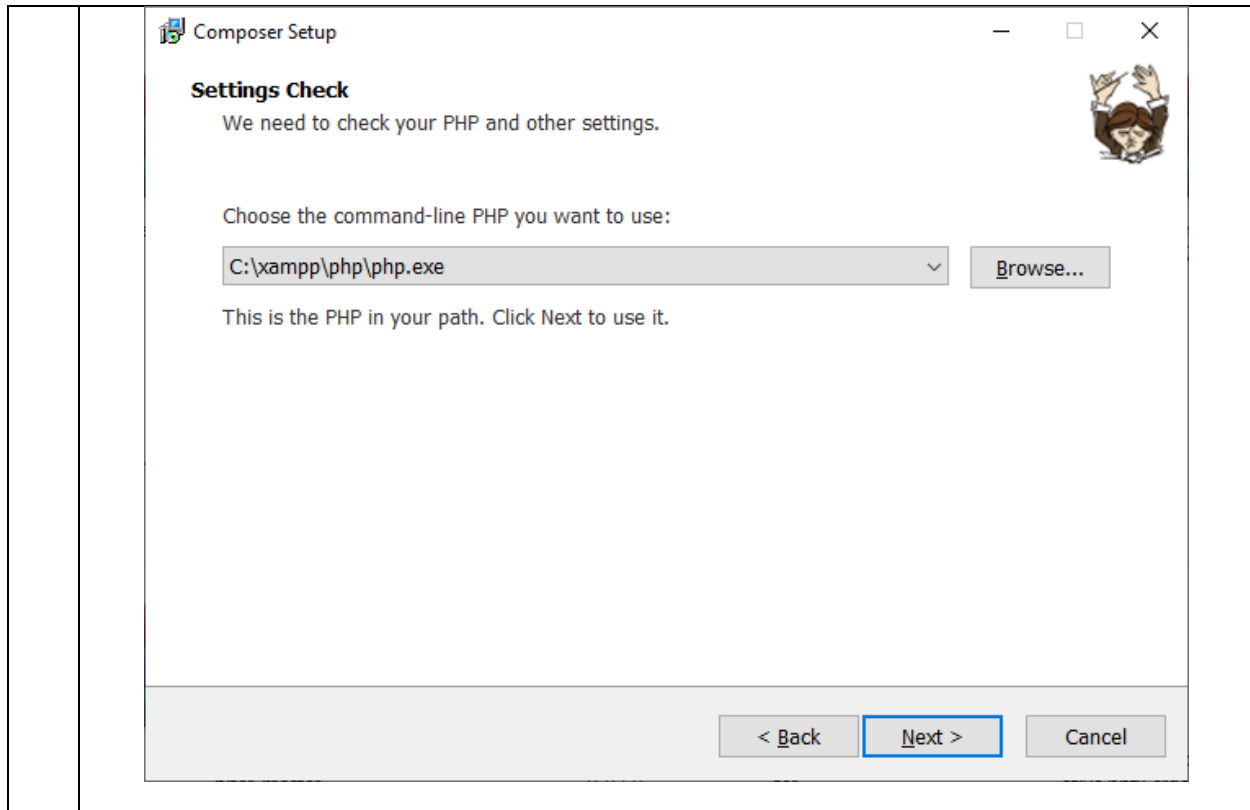
Oleh :

Rahmad Alfian Maskuri / NIM : 1841720048 / TI – 2A

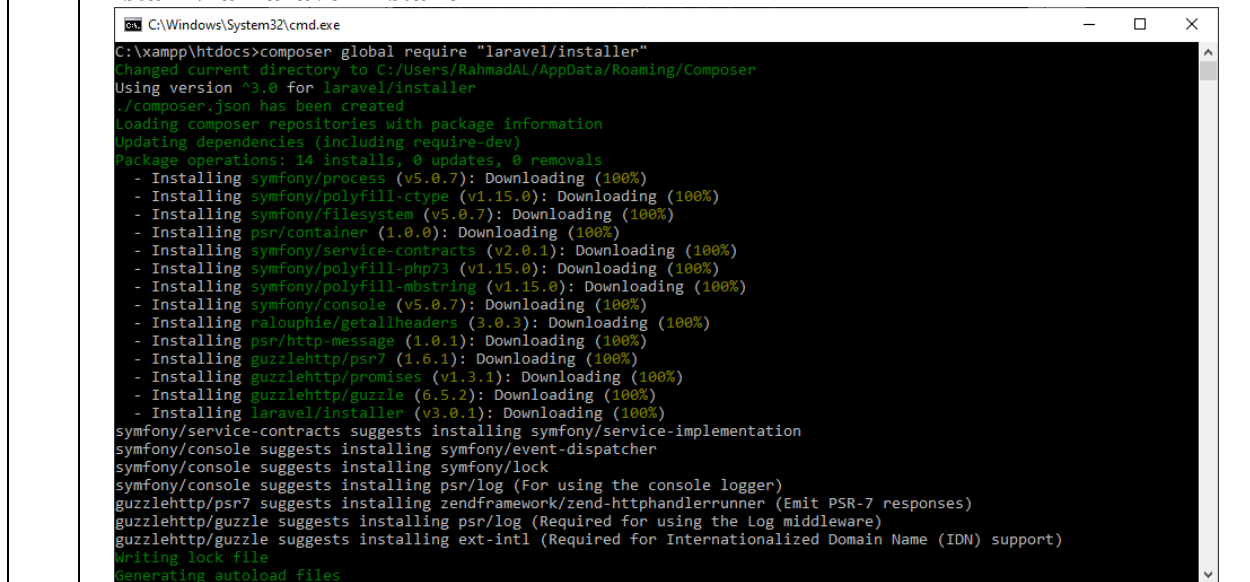
**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG**

Praktikum – Bagian 1: Instalasi Laravel di Windows

No	Langkah Percobaan
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>  A screenshot of a web browser showing the Composer website. The browser has two tabs: '10. Pengantar dan Instalasi Laravi...' and 'Composer'. The address bar shows 'getcomposer.org'. The website features a cartoon character with a distressed expression, wearing an orange shirt and blue sleeves, with the word 'COMPOSER' written below. Underneath the character, it says 'A Dependency Manager for PHP' and 'Latest: 1.10.1'. At the bottom, there are six buttons: 'Getting Started', 'Download', 'Documentation', 'Browse Packages', 'Issues', and 'GitHub'. <p> A Dependency Manager for PHP Latest: 1.10.1 Getting Started Download Documentation Browse Packages Issues GitHub</p>



2 Install Via Laravel Installer

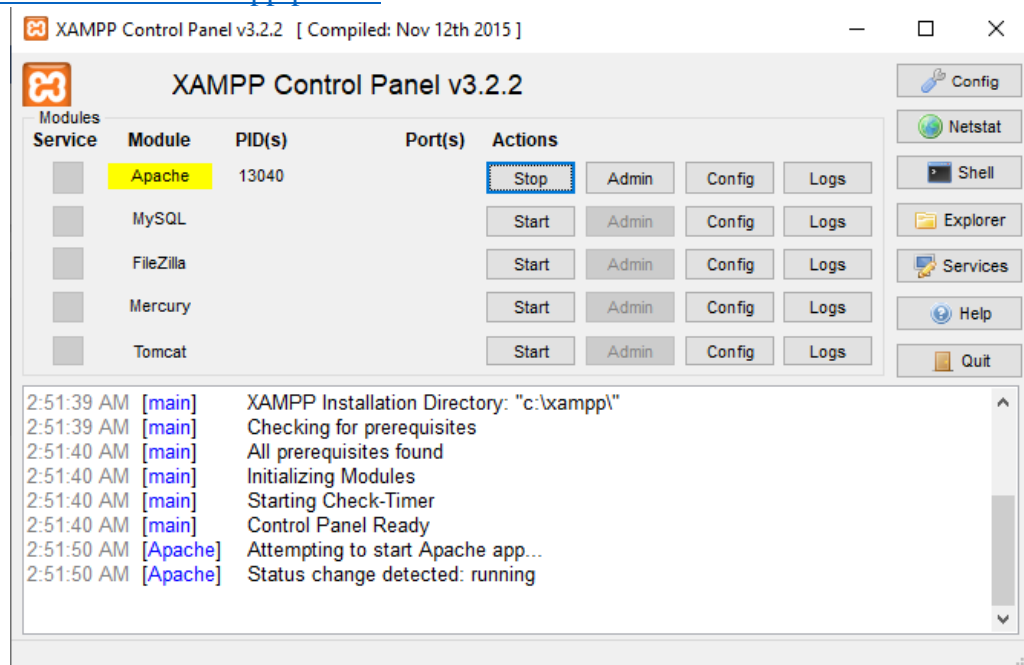


```
C:\Windows\System32\cmd.exe
C:\xampp\htdocs>laravel new laravelapp
Crafting application...
Loading composer repositories with package information
Installing dependencies (including require-dev) from lock file
Package operations: 94 installs, 0 updates, 0 removals
- Installing doctrine/infector (1.3.1): Downloading (100%)
- Installing doctrine/lexer (1.2.0): Downloading (100%)
- Installing dragonmantank/cron-expression (v2.3.0): Downloading (100%)
- Installing voku/portable-ascii (1.4.10): Downloading (100%)
- Installing symfony/polyfill-ctype (v1.15.0): Loading from cache
- Installing phpoption/phpooption (1.7.3): Downloading (100%)
- Installing vlucas/phpdotenv (v4.1.3): Downloading (100%)
- Installing symfony/css-selector (v5.0.7): Downloading (100%)
- Installing tijsverkoyen/css-to-inline-styles (2.2.2): Downloading (connecting..Downloading (100%)
- Installing symfony/polyfill-mbstring (v1.15.0): Loading from cache
- Installing symfony/var-dumper (v5.0.7): Downloading (100%)
- Installing symfony/routing (v5.0.7): Downloading (100%)
- Installing symfony/http-foundation (v5.0.7): Downloading (100%)
- Installing symfony/polyfill-php72 (v1.15.0): Downloading (100%)
- Installing symfony/polyfill-intl-idn (v1.15.0): Downloading (100%)
- Installing symfony/mime (v5.0.7): Downloading (100%)
- Installing symfony/polyfill-php73 (v1.15.0): Loading from cache
- Installing symfony/http-kernel (v5.0.7): Downloading (100%)
- Installing psr/event-dispatcher (1.0.0): Downloading (100%)
- Installing symfony/event-dispatcher-contracts (v2.0.1): Downloading (connecting..Downloading (100%)
- Installing symfony/event-dispatcher (v5.0.7): Downloading (100%)
- Installing psr/log (1.1.3): Downloading (100%)
- Installing symfony/error-handler (v5.0.7): Downloading (100%)
- Installing symfony/http-kernel (v5.0.7): Downloading (100%)
- Installing symfony/finder (v5.0.7): Downloading (100%)
- Installing psr/container (1.0.0): Loading from cache
- Installing symfony/service-contracts (v2.0.1): Loading from cache
- Installing symfony/console (v5.0.7): Loading from cache
- Installing symfony/polyfill-iconv (v1.15.0): Downloading (100%)
- Installing egulias/email-validator (2.1.17): Downloading (100%)
- Installing swiftmailer/swiftmailer (v6.2.3): Downloading (100%)
- Installing ramsey/collecton (0.8.1): Downloading (100%)
- Installing brick/math (0.8.14): Downloading (100%)
- Installing ramsey/uuid (4.0.1): Downloading (100%)
- Installing psr/simple-cache (1.0.1): Downloading (100%)
- Installing opis/closure (3.5.1): Downloading (100%)
- Installing symfony/translation-contracts (v2.0.1): Downloading (100%)
- Installing symfony/translation (v5.0.7): Downloading (100%)
- Installing nesbot/carbon (2.32.2): Downloading (100%)
- Installing monolog/monolog (2.0.2): Downloading (100%)
- Installing league/flysystem (1.0.66): Downloading (100%)
```

```
C:\Windows\System32\cmd.exe
voku/portable-ascii suggests installing ext-intl (Use intl for transliterator.transliterate() support)
symfony/var-dumper suggests installing ext-intl (To show region name in time zone dump)
symfony/routing suggests installing doctrine/annotations (For using the annotation loader)
symfony/routing suggests installing symfony/config (For using the all-in-one router or any loader)
symfony/routing suggests installing symfony/expression-language (For using expression matching)
symfony/routing suggests installing symfony/yaml (For using the YML loader)
symfony/polyfill-intl-idn suggests installing ext-intl (For best performance)
symfony/event-dispatcher suggests installing symfony/dependency-injection
symfony/http-kernel suggests installing symfony/browser-kit
symfony/http-kernel suggests installing symfony/config
symfony/http-kernel suggests installing symfony/dependency-injection
symfony/service-contracts suggests installing symfony/service-implementation
symfony/console suggests installing symfony/lock
egulias/email-validator suggests installing ext-intl (PHP Internationalization Libraries are required to use the SpoofChecking validation)
swiftmailer/swiftmailer suggests installing ext-intl (Needed to support internationalized email addresses)
swiftmailer/swiftmailer suggests installing true/punycode (Needed to support internationalized email addresses, if ext-intl is not installed)
ramsey/uuid suggests installing ext-gmp (Enables faster math with arbitrary-precision integers using GMP.)
ramsey/uuid suggests installing ext-uuid (Enables the use of PeclUuidTimeGenerator and PeclUuidRandomGenerator.)
ramsey/uuid suggests installing paragonie/random-lib (Provides RandomLib for use with the RandomLibAdapter)
ramsey/uuid suggests installing ramsey/uuid-doctrine (Allows the use of Ramsey\Uuid\Uuid as Doctrine field type.)
symfony/translation suggests installing symfony/config
symfony/translation suggests installing symfony/yaml
monolog/monolog suggests installing aws/aws-sdk-php (Allow sending log messages to AWS services like DynamoDB)
monolog/monolog suggests installing doctrine/couchdb (Allow sending log messages to a CouchDB server)
monolog/monolog suggests installing elasticsearch/elasticsearch (Allow sending log messages to an Elasticsearch server via official client)
monolog/monolog suggests installing ext-amqp (Allow sending log messages to an AMQP server (1.0+ required))
monolog/monolog suggests installing ext-mongodb (Allow sending log messages to a MongoDB server (via driver))
monolog/monolog suggests installing graylog2/gelf-php (Allow sending log messages to a Graylog2 server)
monolog/monolog suggests installing mongodb/mongodb (Allow sending log messages to a MongoDB server (via library))
monolog/monolog suggests installing php-amqplib/php-amqplib (Allow sending log messages to an AMQP server using php-amqplib)
monolog/monolog suggests installing php-console/php-console (Allow sending log messages to Google Chrome)
monolog/monolog suggests installing rollbar/rollbar (Allow sending log messages to Rollbar)
monolog/monolog suggests installing rufin/elastica (Allow sending log messages to an Elastic Search server)
league/flysystem suggests installing league/flysystem-aws-s3-v2 (Allows you to use S3 storage with AWS SDK v2)
league/flysystem suggests installing league/flysystem-aws-s3-v3 (Allows you to use S3 storage with AWS SDK v3)
league/flysystem suggests installing league/flysystem-azure (Allows you to use Windows Azure Blob storage)
league/flysystem suggests installing league/flysystem-cached-adapter (Flysystem adapter decorator for metadata caching)
league/flysystem suggests installing league/flysystem-eventable-flysystem (Allows you to use EventableFlysystem)
league/flysystem suggests installing league/flysystem-rackspace (Allows you to use Rackspace Cloud Files)
league/flysystem suggests installing league/flysystem-sftp (Allows you to use SFTP server storage via phpseclib)
league/flysystem suggests installing league/flysystem-webdav (Allows you to use WebDAV storage)
league/flysystem suggests installing league/flysystem-ziparchive (Allows you to use ZipArchive adapter)
league/flysystem suggests installing spatie/flysystem-dropbox (Allows you to use Dropbox storage)
league/flysystem suggests installing srmlive/flysystem-dropbox-v2 (Allows you to use Dropbox storage for PHP 5 applications)
laravel/framework suggests installing aws/aws-sdk-php (Required to use the SQS queue driver, DynamoDb failed job storage and SES mail driver (~3.0).)
laravel/framework suggests installing doctrine/dbal (Required to rename columns and drop SQLite columns (~2.6).)
```

```
Select C:\Windows\System32\cmd.exe
laravel/framework suggests installing ext-memcached (Required to use the memcache cache driver.)
laravel/framework suggests installing ext-pcntl (Required to use all features of the queue worker.)
laravel/framework suggests installing ext-posix (Required to use all features of the queue worker.)
laravel/framework suggests installing ext-redis (Required to use the Redis cache and queue drivers (^4.0|^5.0).)
laravel/framework suggests installing league/flysystem-aws-s3-v3 (Required to use the Flysystem S3 driver (^1.0).)
laravel/framework suggests installing league/flysystem-azure (Required to use the Flysystem Azure driver (^1.0).)
laravel/framework suggests installing league/flysystem-ftp (Required to use the Flysystem FTP driver (^1.0).)
laravel/framework suggests installing nyholm/psr7 (Required to use PSR-7 bridging features (^1.2).)
laravel/framework suggests installing pdm/psr7 (Required to use the psr7 queue driver (^4.0).)
laravel/framework suggests installing symfony/cache (Required to use PSR-6 cache bridge (^5.0).)
laravel/framework suggests installing symfony/http-message-bridge (Required to use PSR-7 bridging features (^2.0).)
laravel/framework suggests installing wilddbt/swiftmailer-postmark (Required to use Postmark mail driver (^3.0).)
guzzlehttp/psr7 suggests installing zendframework/zend-httpdierrunner (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
psr/psr suggests installing ext-pcntl (Enabling the PCNTL extension makes Psr7 a lot happier :)
psr/psr suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psr/psr suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
psr/psr suggests installing hoa/console (A pure PHP readline implementation. You'll want this if your PHP install doesn't already support readline or libedit.)
filp/whoops suggests installing whoops/soap (Formats errors as SOAP responses)
facade/ignition suggests installing laravel/telescope (^3.1)
sebastian/global-state suggests installing ext-uopz (*)
sebastian/environment suggests installing ext-posix (*)
phpunit/php-code-coverage suggests installing ext-xdebug (^2.7.2)
phpunit/phpunit suggests installing ext-soap (*)
phpunit/phpunit suggests installing ext-xdebug (*)
phpunit/phpunit suggests installing phpunit/php-invoker (^2.0.0)
Package jakub-onderka/php-console-color is abandoned, you should avoid using it. Use php-parallel-lint/php-console-color instead.
Package jakub-onderka/php-console-highlighter is abandoned, you should avoid using it. Use php-parallel-lint/php-console-highlighter instead.
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 ready! Build something amazing.
> 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.
```

- 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/>





Laravel

[DOCS](#) [LARACASTS](#) [NEWS](#) [BLOG](#) [NOVA](#) [FORGE](#) [VAPOR](#) [GITHUB](#)

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 :



Laravel

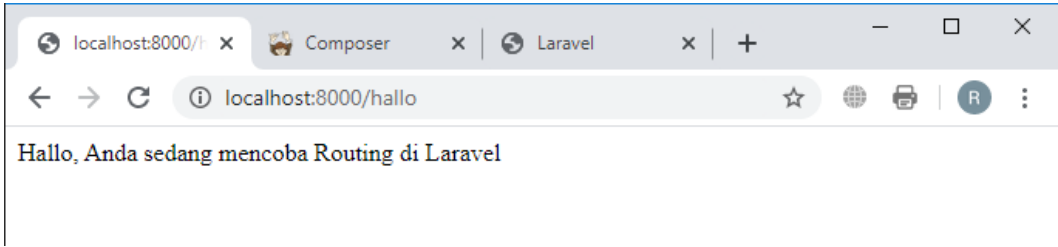
[DOCS](#) [LARACASTS](#) [NEWS](#) [BLOG](#) [NOVA](#) [FORGE](#) [VAPOR](#) [GITHUB](#)

Talking: Rahmad Alf

```
C:\Windows\System32\cmd.exe - php artisan serve
C:\xampp\htdocs\laravelapp>php artisan serve
laravel development server started: http://127.0.0.1:8000
[Sat Apr  4 03:04:29 2020] 127.0.0.1:51583 [200]: /favicon.ico
[Sat Apr  4 03:06:30 2020] 127.0.0.1:51603 [200]: /favicon.ico
```

Praktikum – Bagian 2: Mengenal dan membuat route

No	Langkah Percobaan
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 use Illuminate\Support\Facades\Route; 4 5 /* 6 ----- 7 Web Routes 8 ----- 9 10 Here is where you can register web routes for your application. 11 routes are loaded by the RouteServiceProvider within a group 12 contains the "web" middleware group. Now create something great. 13 14 */ 15 16 Route::get('/', function () { 17 return view('welcome'); 18 }); 19</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 : Route::get('hallo', function () { return "Hallo, Anda sedang mencoba Routing di Laravel"; }); Untuk melihat hasilnya, pada browser, ketikkan alamat http://localhost:8000/hallo</p>

	<div data-bbox="378 195 1313 1054"> <pre> web.php x routes > web.php 1 <?php 2 3 use Illuminate\Support\Facades\Route; 4 5 /* 6 ----- 7 Web Routes 8 ----- 9 10 Here is where you can register web routes for your application. 11 routes are loaded by the RouteServiceProvider within a group 12 contains the "web" middleware group. Now create something great! 13 14 */ 15 16 Route::get('/', function () { 17 return view('welcome'); 18 }); 19 20 Route::get('hallo', function () { 21 return "Hallo, Anda sedang mencoba 22 Routing di Laravel"; 23 }); 24 </pre> </div> <div data-bbox="321 1089 1372 1333">  </div>
3	<p>Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut: <code>Route::get('profil', function () { return view('profil'); });</code>. Buat file baru bernama <code>profil.blade.php</code> di dalam folder <code>laravelapp/resources/views/</code>, ketikkan script berikut ini:</p>

```

web.php ×
routes > web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4
5  /*
6  |-----
7  | Web Routes
8  |-----
9  |
10 | Here is where you can register web routes for your application.
11 | routes are loaded by the RouteServiceProvider within a group
12 | contains the "web" middleware group. Now create something great.
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba
22     Routing di Laravel";
23 });
24
25 Route::get('profil', function () {
26     return view('profil');
27 });

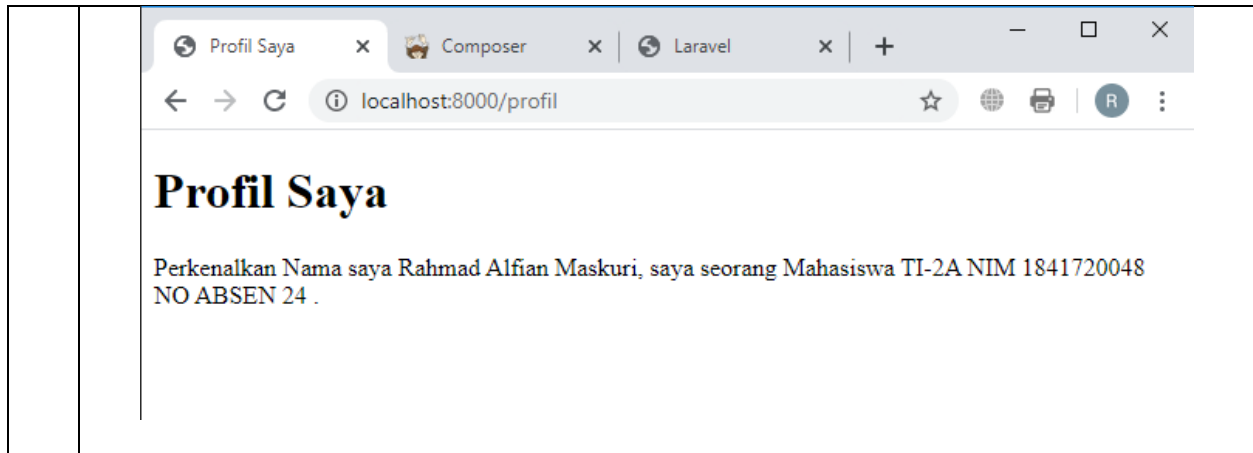
```

```

web.php  profil.blade.php ×
resources > views > profil.blade.php
1  <!DOCTYPE html>
2
3  <html>
4  <head>
5  <title>Profil Saya</title>
6
7  </head>
8  <body>
9  <h1>Profil Saya</h1>
10 <p>Perkenalkan Nama saya Rahmad Alfian Maskuri, saya seorang Mahasiswa TI-2A NIM 1841720048 NO ABSEN 24
11 </p>
12 </body>
13 </html>
14

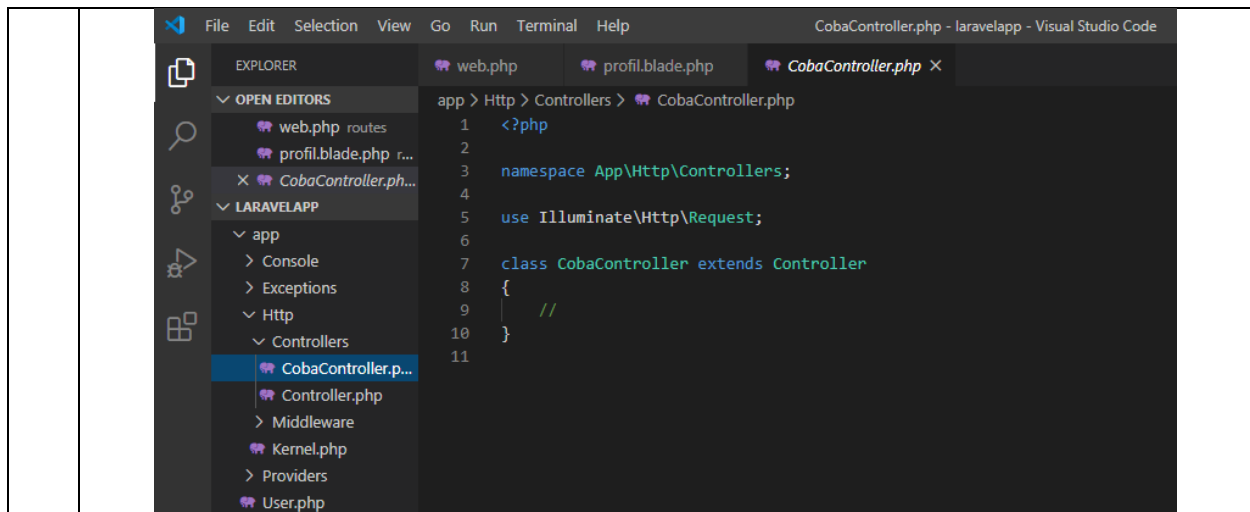
```

Tuliskan profil kalian masing-masing. Untuk menampilkan hasilnya, ketikkan alamat <http://localhost:8000/profil>



Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel

No	Langkah Percobaan
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> <p style="text-align: center;">php artisan make:controller CobaController</p>  <p>Maka akan terbentuk sebuah file dengan nama CobaController.php pada folder laravelapp\app\Http\Controllers\</p>



- 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:


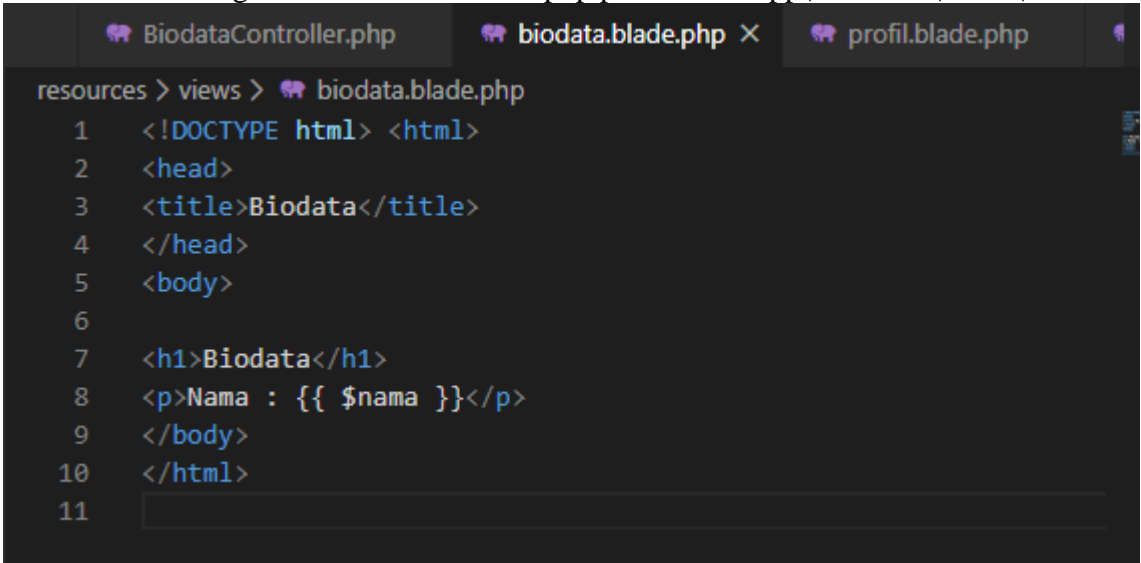
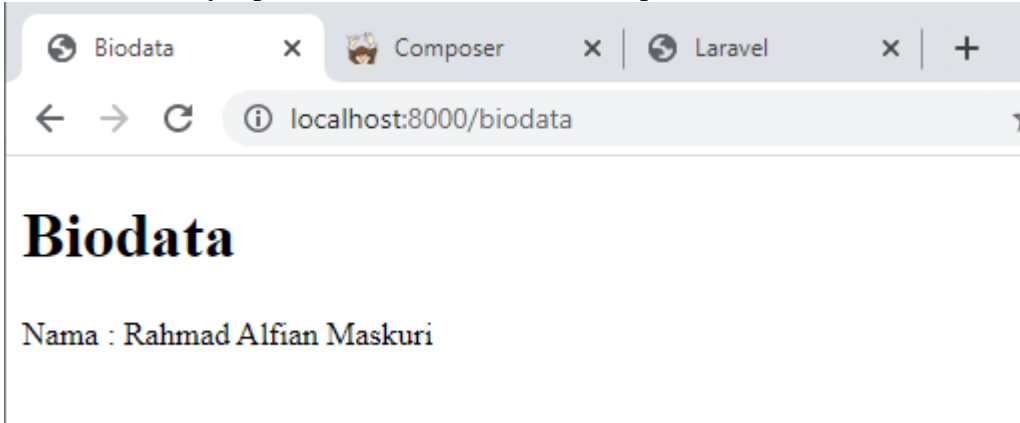
```
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
20 Route::get('hallo', function () {
21     return "Hallo, Anda sedang mencoba
22     Routing di Laravel";
23 });
24
25 Route::get('profil', 'CobaController@profil');
26
```

```
web.php • profil.blade.php • CobaController.php •
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         return view('profil');
11     }
12
13 }
14
```

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

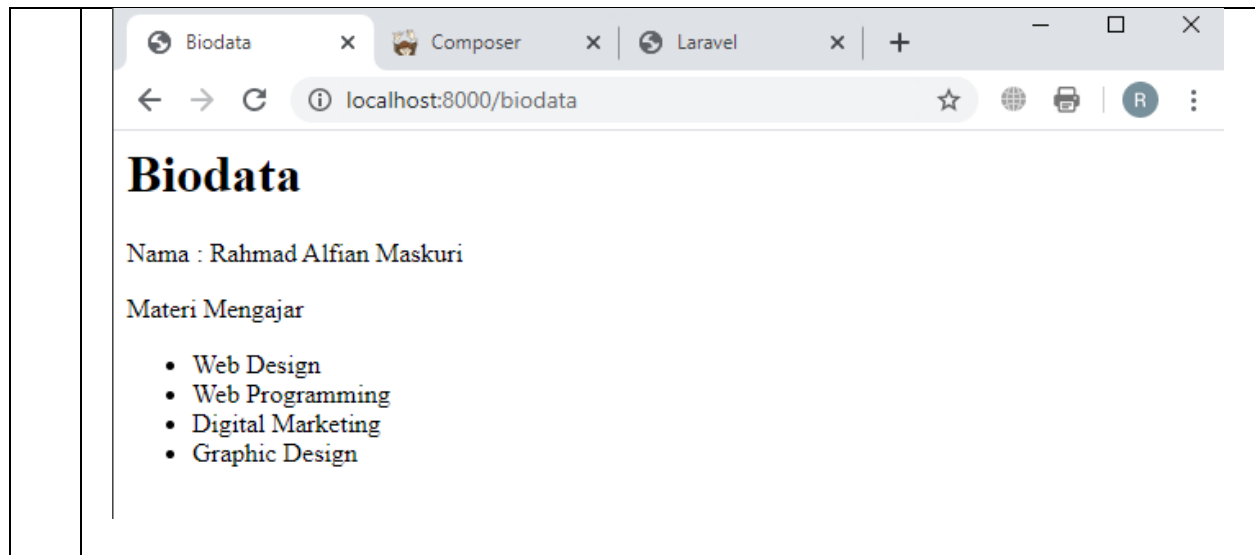
Praktikum – Bagian 4: Memberikan Data Controller kepada View

No	Langkah Percobaan
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.</p> <p>Buat route baru untuk halaman biodata</p> <pre>Route::get('biodata', 'BiodataController@index');</pre> <p>Buat controller baru dengan nama BiodataController.php</p> <pre>php artisan make:controller BiodataController</pre> <pre>C:\xampp\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> web.php BiodataController.php × biodata.blade.php 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 = 'Rahmad Alfian Maskuri'; //ubah dengan nama kalian 11 return view('biodata',['nama' => \$nama]); 12 } 13 14 } 15 </pre>
2	<p>Buat file view dengan nama biodata.blade.php pada laravelapp\resources\views\</p>  <pre> BiodataController.php biodata.blade.php × profil.blade.php resources > views > biodata.blade.php 1 <!DOCTYPE html> <html> 2 <head> 3 <title>Biodata</title> 4 </head> 5 <body> 6 7 <h1>Biodata</h1> 8 <p>Nama : {{ \$nama }}</p> 9 </body> 10 </html> 11 </pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat http://localhost:8000/biodata .</p>  <p>Biodata</p> <p>Nama : Rahmad Alfian Maskuri</p>

Praktikum – Bagian 5: Memberikan Data Array Kepada View

No	Langkah Percobaan
1	<p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre data-bbox="313 449 1382 963">web.php BiodataController.php × biodata.blade.php 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 = 'Rahmad Alfian Maskuri'; 11 \$materi = ["Web Design", "Web Programming", "Digital Marketing", "Graphic Design"]; 12 return view('biodata', ['nama' => \$nama, 'materi' => \$materi]); 13 } 14 } 15 16 17</pre>
2	<p>Tambahkan isi file view biodata.blade.php menjadi seperti berikut:</p> <pre data-bbox="313 1014 1390 1709">web.php BiodataController.php biodata.blade.php × resources > views > biodata.blade.php 1 <html> 2 <head> 3 <title>Biodata</title> 4 5 </head> 6 <body> 7 <h1>Biodata</h1> 8 <p>Nama : {{ \$nama }}</p> <p>Materi Mengajar</p> 9 10 @foreach(\$materi as \$datamateri) 11 {{ \$datamateri }} 12 @endforeach 13 14 </body> 15 16 </html> 17</pre> <p>Untuk melihat hasilnya, pada browser, ketik alamat http://localhost:8000/biodata .</p>



Biodata

Nama : Rahmad Alfian Maskuri

Materi Mengajar

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