

LAPORAN TUGAS PEMROGRAMAN WEB LANJUT

Jobsheet 9 Pengantar dan Instalasi Laravel




Disusun Oleh :

Nama : Hunayn Risatayn
NIM : 1841720148
Kelas : TI-2B

Program Studi D-IV Teknik Informatika
Jurusan Teknologi Informasi
Politeknik Negeri Malang

Praktikum – Bagian 1: Instalasi Laravel di Windows

| Langkah | Keterangan |
|---------|--|
| 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> <div data-bbox="376 376 638 716"></div> <p>A Dependency Manager for PHP</p> <p>Latest: 1.10.1</p> <div data-bbox="279 853 740 963"><div>Getting Started</div><div>Download</div><div>Documentation</div><div>Browse Packages</div></div> <p>Terdapat 2 cara dalam menginstall Laravel,</p> <p>1. Install Via Laravel Installer</p> <p>Setelah proses instalasi Composer selesai, buka Command Prompt lalu tujulah lokasi folder yang akan menampung folder kerja kita. Karena pada praktikum sebelumnya telah menggunakan XAMPP, maka dapat menggunakan folder htdocs. Pindah ke direktori htdocs dengan ketik: cd C:\xampp\htdocs. Setelah berada di dalam folder, ketikkan perintah sebagai berikut:</p> <p style="text-align: center;">composer global require "laravel/installer"</p> <p>Tunggulah hingga proses penginstalan selesai. Setelah itu, Kita bisa mulai membuat project laravel dengan menetikkan perintah:</p> <p style="text-align: center;">laravel new laravelapp</p> <p>Keterangan: laravelapp: merupakan folder kerja laravel, Kita dapat menggantinya sesuai dengan keinginan</p> <p>2. Install Via Composer Create-Project</p> <p>Selain menggunakan cara pertama, Kita dapat melakukan penginstalan laravel dengan mengetikkan perintah berikut pada command prompt:</p> <p style="text-align: center;">composer create-project --prefer-dist laravel/laravel laravelapp</p> |

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\xampp\htdocs

C:\xampp\htdocs>composer global require "laravel/installer"
Changed current directory to C:/Users/User/AppData/Roaming/Composer
Using version ^3.0 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 (v5.0.7): Downloading (100%)
  - Installing symfony/polyfill-ctype (v1.15.0): Loading from cache
  - Installing symfony/filesystem (v5.0.7): Downloading (100%)
  - Installing psr/container (1.0.0): Downloading (100%)
  - Installing symfony/service-contracts (v2.0.1): Downloading (100%)
  - Installing symfony/polyfill-php73 (v1.15.0): Downloading (100%)
  - Installing symfony/polyfill-mbstring (v1.15.0): Downloading (100%)
  - Installing symfony/console (v5.0.7): Downloading (100%)
  - Installing ralouphie/getallheaders (3.0.3): Loading from cache
  - Installing psr/http-message (1.0.1): Loading from cache
  - Installing guzzlehttp/psr7 (1.6.1): Loading from cache
  - Installing guzzlehttp/promises (v1.3.1): Loading from cache
  - Installing guzzlehttp/guzzle (6.5.2): Loading from cache
  - Installing laravel/installer (v3.0.1): Downloading (100%)
symfony/service-contracts suggests installing symfony/service-implementation
symfony/console suggests installing symfony/event-dispatcher
symfony/console suggests installing symfony/lock
symfony/console suggests installing psr/log (For using the console logger)
guzzlehttp/psr7 suggests installing zendframework/zend-httphandler (Emit PSR-7 responses)
guzzlehttp/guzzle suggests installing psr/log (Required for using the Log middleware)
guzzlehttp/guzzle suggests installing ext-intl (Required for Internationalized Domain Name (IDN) support)
Writing lock file
Generating autoload files

C:\xampp\htdocs>
```

Administrator: Command Prompt

```
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/inflector (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/phpoption (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 (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/process (v5.0.7): Loading from cache
  - 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-foundation (v5.0.7): Downloading (100%)
  - Installing psr/event-dispatcher (1.0.0): Downloading (100%)
  - Installing symfony/event-dispatcher-contracts (v2.0.1): 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/collection (1.0.1): 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%)

Activate Windows
Go to Settings to activate Windows.
```

```
Administrator: Command Prompt
psy/psysh suggests installing ext-pcntl (Enabling the PCNTL extension makes PsySH a lot happier :))
psy/psysh suggests installing ext-pdo-sqlite (The doc command requires SQLite to work.)
psy/psysh suggests installing ext-posix (If you have PCNTL, you'll want the POSIX extension as well.)
filip/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
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
> @php artisan key:generate --ansi
Application key set successfully.
> 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.

C:\xampp\htdocs>
```

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/> 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 :

```
C:\> Administrator: Command Prompt - php artisan serve
Microsoft Windows [Version 10.0.18362.720]
(c) 2019 Microsoft Corporation. All rights reserved.

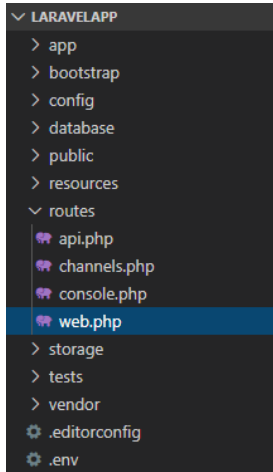
C:\WINDOWS\system32>cd C:\xampp\htdocs\laravelapp

C:\xampp\htdocs\laravelapp>php artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Apr  6 07:48:11 2020] 127.0.0.1:63544 [200]: /favicon.ico
[Mon Apr  6 07:50:26 2020] 127.0.0.1:63556 [200]: /favicon.ico
[Mon Apr  6 07:52:20 2020] 127.0.0.1:63560 [200]: /favicon.ico
```



Praktikum – Bagian 2: Mengenal dan membuat route

| Langkah | Keterangan |
|---------|---|
| 1 | 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 <code>laravelapp\routes\web.php</code> Berikut ini adalah perintah route untuk menampilkan halaman default Laravel tersebut |



```
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. These
11 | routes are loaded by the RouteServiceProvider within a group which
12 | contains the "web" middleware group. Now create something great!
13 |
14 */
15
16 Route::get('/', function () {
17     return view('welcome');
18 });
19
```

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>

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. These

11 | routes are loaded by the RouteServiceProvider within a group which

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 Routing di Laravel";

22 |});

23

10. Pengantar dan Instalasi Laravi ×

127.0.0.1:8000/hallo × +

← → ↺ ⓘ 127.0.0.1:8000/hallo

Hallo, Anda sedang mencoba Routing di Laravel

10. Pengantar dan Instalasi Laravi ×

localhost/laravelapp/public/hallo × +

← → ↺ ⓘ localhost/laravelapp/public/hallo

Hallo, Anda sedang mencoba Routing di Laravel

- 2

Selanjutnya, kita akan mencoba membuat route baru dengan menampilkan halaman view. Tambahkan route baru untuk halaman “profil” dengan menambahkan script berikut:

Route::get('profil', function () { return view('profil'); });

Buat file baru bernama **profil.blade.php** di dalam folder **laravelapp\resources\views**, ketikkan script berikut ini:

```

<!DOCTYPE html>
<html>

<head>
    <title>Profil Saya</title>
</head>

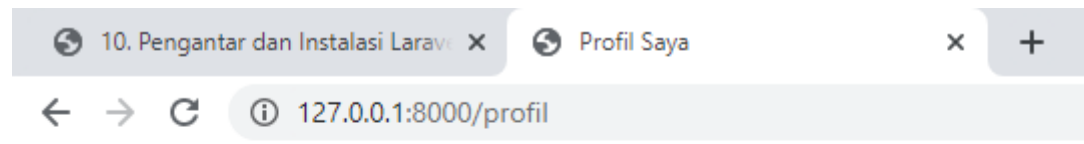
<body>
    <h1>Profil Saya</h1>
    <p>Perkenalkan Nama saya nama_saya, saya seorang .....
    .</p>

```

</body>

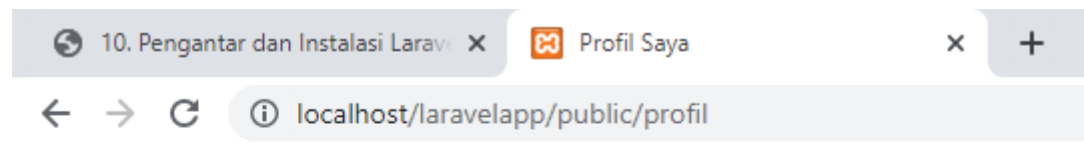
</html>

```
24 Route::get('profil', function () {  
25     return view('profil');  
26 });  
27
```



Profil Saya

Perkenalkan Nama saya nama_saya, saya seorang



Profil Saya

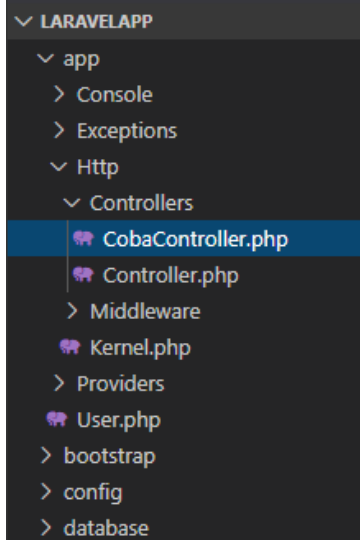
Perkenalkan Nama saya nama_saya, saya seorang

Praktikum – Bagian 3: Cara Menggunakan Controller pada Framework Laravel

| 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.</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> |


```
C:\xampp\htdocs\laravelapp>php artisan make:controller CobaController
Controller created successfully.

C:\xampp\htdocs\laravelapp>
```



```
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      //
10 }
11
```

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:

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class CobaController extends Controller
```

```
{
```

```
    //
```

```
    public function profil()
```

```
    {
```

```
        return view('profil');
```

```

}
}

23
24 Route::get('profil', 'CobaController@profil');
25

```

```

app > Http > Controllers > CobaController.php > PHP Intelephense > CobaController > profil
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class CobaController extends Controller
8  {
9      //
10     public function profil()
11     {
12         return view('profil');
13     }
14 }
15

```

10. Pengantar dan Instalasi Laravel x Profil Saya x +

127.0.0.1:8000/profil

Profil Saya

Perkenalkan Nama saya nama_saya, saya seorang

Praktikum – Bagian 4: Memberikan Data Controller kepada View

| 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.</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> <p>Ketikkan script berikut dengan memberikan method / fungsi index pada laravelapp\app\Http\Controllers\BiodataController.php</p> <pre><?php</pre> |

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class BiodataController extends Controller
```

```
{
```

```
    //
```

```
    public function index()
```

```
    {
```

```
        $nama = 'Hunayn Risatayn'; //ubah dengan nama kalian
```

```
        return view('biodata', ['nama' => $nama]);
```

```
    }
```

```
}
```

```
C:\xampp\htdocs\laravelapp>php artisan make:controller BiodataController  
Controller created successfully.
```

```
C:\xampp\htdocs\laravelapp>
```

```
25
```

```
26     Route::get('biodata', 'BiodataController@index');
```

```
27
```

```
app > Http > Controllers > BiodataController.php > PHP Intelephense > BiodataController
```

```
1  <?php
```

```
2
```

```
3  namespace App\Http\Controllers;
```

```
4
```

```
5  use Illuminate\Http\Request;
```

```
6
```

```
7  class BiodataController extends Controller
```

```
8  {
```

```
9      //
```

```
10     public function index()
```

```
11     {
```

```
12         $nama = 'Hunayn Risatayn'; //ubah dengan nama kalian
```

```
13         return view('biodata', ['nama' => $nama]);
```

```
14     }
```

```
15 }
```

```
16
```

Buat file view dengan nama **biodata.blade.php** pada **laravelapp/resources/views**

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

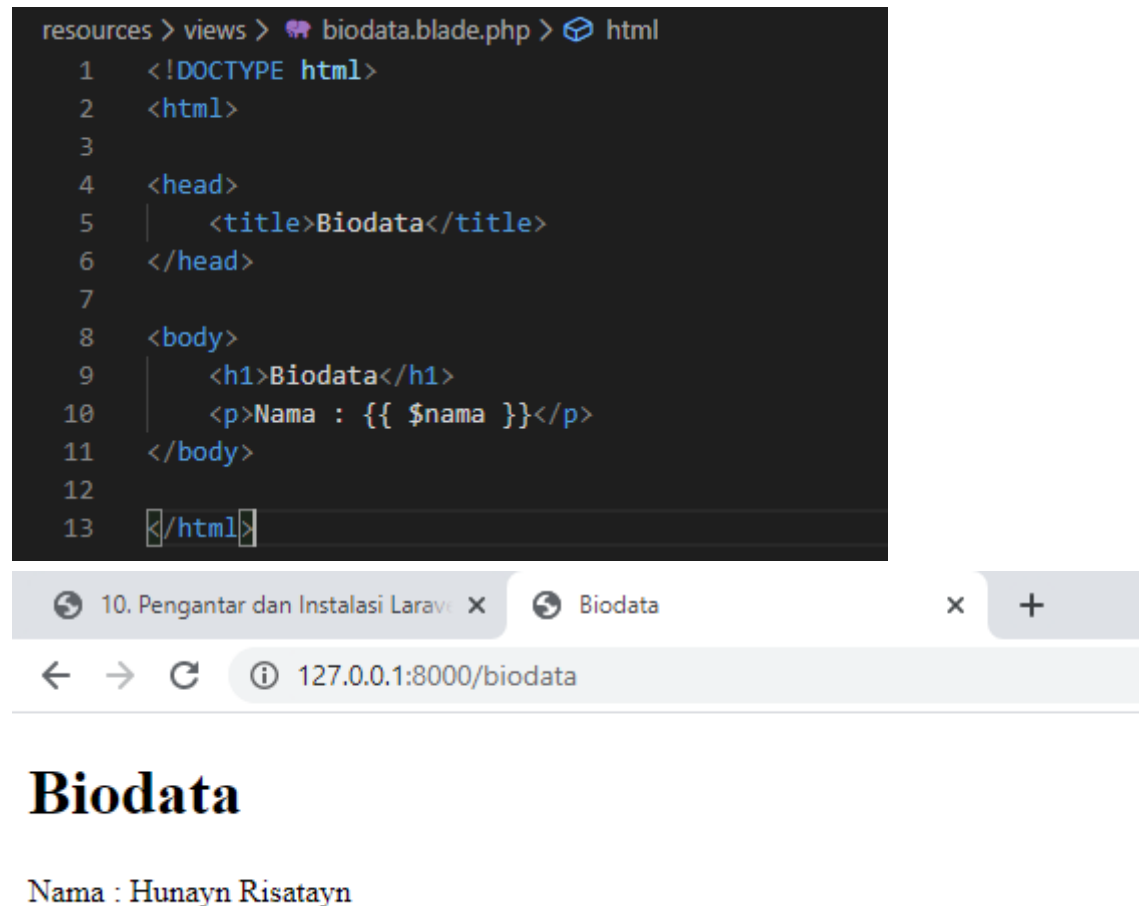
```
    <title>Biodata</title>
```

```
</head>
```

```
<body>
  <h1>Biodata</h1>
  <p>Nama : {{ $nama }}</p>
</body>

</html>
```

Untuk melihat hasilnya, pada browser, ketik alamat <http://localhost:8000/biodata>



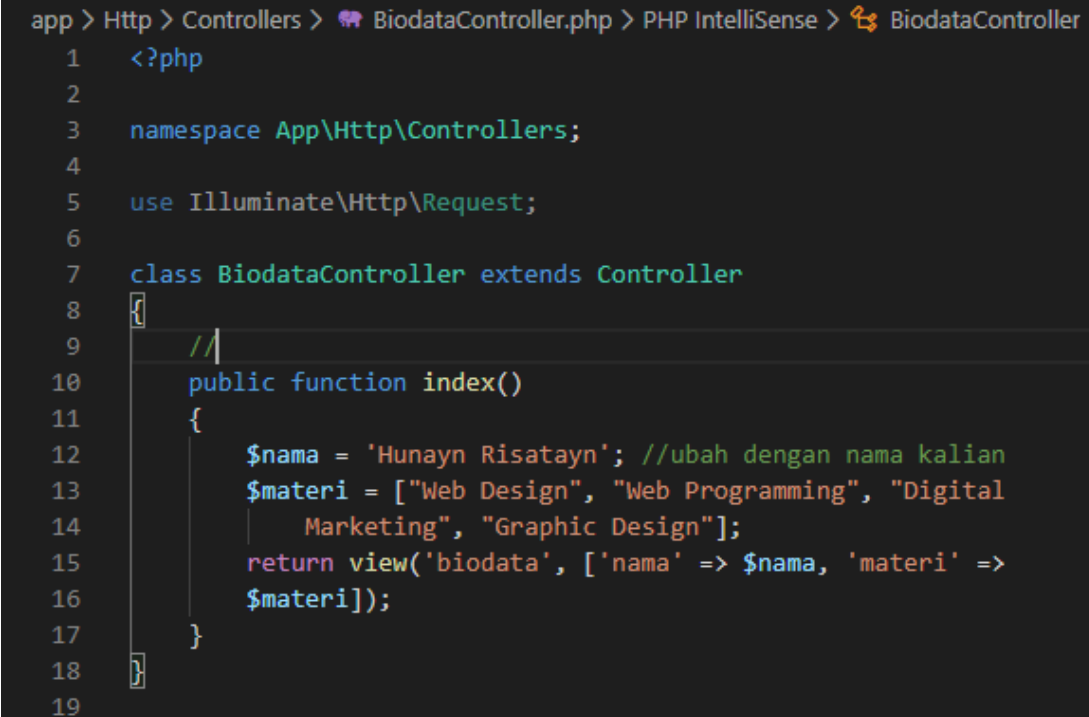
Praktikum – Bagian 5: Memberikan Data Array Kepada View

| Langkah | Keterangan |
|---------|---|
| 1 | <p>Pada bagian ini, Kita coba untuk memberikan data array kepada view. Tambahkan variabel data array pada BiodataController.php</p> <pre><?php namespace App\Http\Controllers; use Illuminate\Http\Request; class BiodataController extends Controller</pre> |

```

{
    //
    public function index()
    {
        $nama = 'Hunayn Risatayn'; //ubah dengan nama kalian
        $materi = ["Web Design", "Web Programming", "Digital
            Marketing", "Graphic Design"];
        return view('biodata', ['nama' => $nama, 'materi' =>
            $materi]);
    }
}

```



```

app > Http > Controllers > BiodataController.php > PHP IntelliSense > BiodataController
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class BiodataController extends Controller
8  {
9      //
10     public function index()
11     {
12         $nama = 'Hunayn Risatayn'; //ubah dengan nama kalian
13         $materi = ["Web Design", "Web Programming", "Digital
14             Marketing", "Graphic Design"];
15         return view('biodata', ['nama' => $nama, 'materi' =>
16             $materi]);
17     }
18 }
19

```

Tambahkan isi file view **biodata.blade.php** menjadi seperti berikut:

```

<!DOCTYPE html>
<html>

<head>
    <title>Biodata</title>
</head>

<body>
    <h1>Biodata</h1>
    <p>Nama : {{ $nama }}</p>
    <p>Materi Mengajar</p>
    <ul>
        @foreach($materi as $datamateri)

```

```
<li>{{ $datamateri }}</li>
@endforeach
</ul>
</body>

</html>
```

```
resources > views > biodata.blade.php > html
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Biodata</title>
6  </head>
7
8  <body>
9      <h1>Biodata</h1>
10     <p>Nama : {{ $nama }}</p>
11     <p>Materi Mengajar</p>
12     <ul>
13         @foreach($materi as $datamateri)
14             <li>{{ $datamateri }}</li>
15         @endforeach
16     </ul>
17 </body>
18
19 </html>
```

Untuk melihat hasilnya, pada browser, ketik alamat <http://localhost:8000/biodata>

