

#Persiapan Tools yang diperlukan

1. Composer

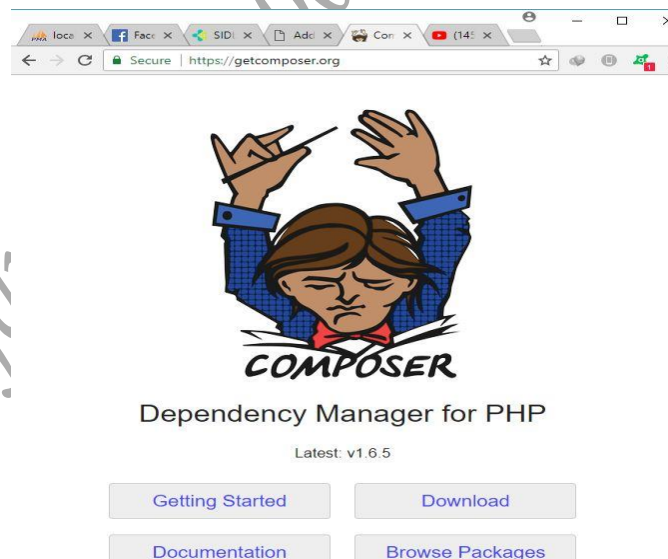
Composer adalah perangkat/software untuk mengelola dependensi di PHP. Composer mengijinkan Anda untuk menginstall dan mengelola modul-modul yang dibutuhkan dalam proses pengembangan aplikasi berbasis website.

➤ Pastikan XAMPP sudah terinstall

Sebelum menginstall Composer, pastikan Anda telah menginstall XAMPP. Anda bisa mendownloadnya di link ini <https://www.apachefriends.org/index.html>. Untuk proses instalasi XAMPP silakan tonton di video ini ya.

➤ Mendownload Composer

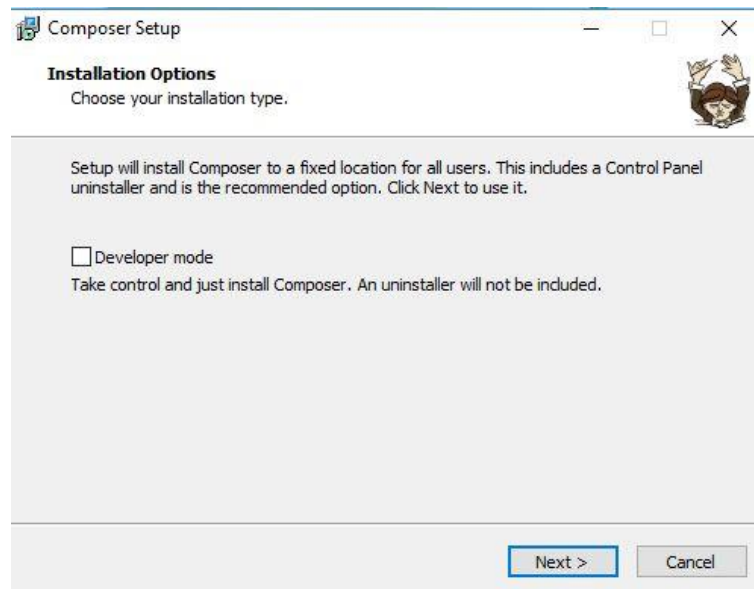
Setelah XAMPP, proses selanjutnya adalah mendownload Composer dari situs resminya di <https://getcomposer.org/>. Silakan didownload dulu Composernya ya. Pilih jenis file yang **Composer-setup.exe** ya. Bisa juga di unduh langsung di <https://getcomposer.org/Composer-Setup.exe>.



➤ Menginstall Composer di Windows

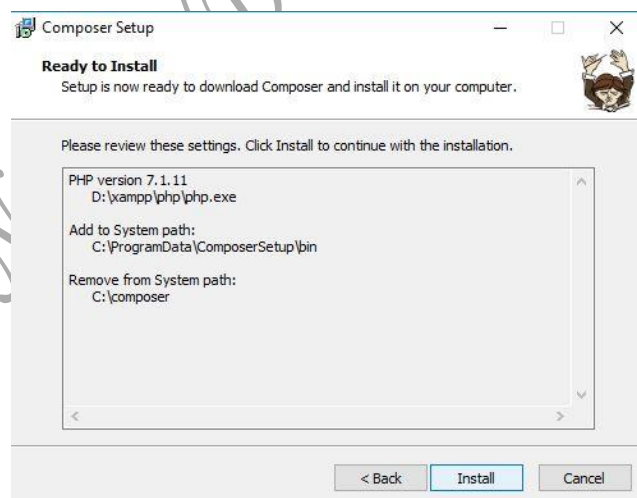
Setelah file, langkah berikutnya adalah menginstallnya:

Klik dua kali pada file **Composer-Setup.exe** sudah diunduh. Jendela konfirmasi akan muncul.



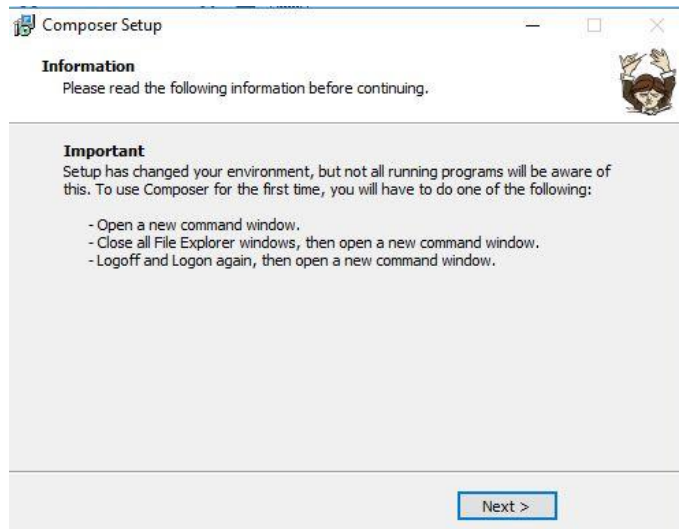
Install Composer

1. Klik **Next** untuk melakukan installasi.
2. Composer akan meminta lokasi file PHP berada. Biasanya ada dalam folder **C:\XAMPP\php\php.exe**. Klik **Browse** untuk mencari file PHP yang sudah diinstall saat menginstall XAMPP.
3. Lalu klik **Next** kembali. Jika muncul jendela **Proxy**, biarkan kosong saja.



4. Klik **Next** lagi. Sehingga muncul jendela Composer **Ready to install**.

5. Tunggu beberapa saat sampai proses instalasi selesai

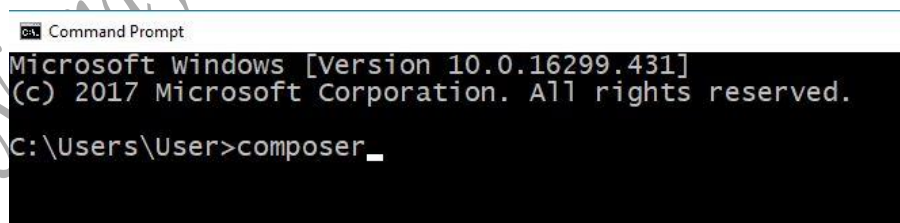


6. Kemudian klik **Next** lagi, lalu klik **Finish**. Anda telah berhasil menginstall Composer.

➤ **Mengetes Composer sudah berhasil diinstall atau belum**

Untuk mengetes Composer sudah berhasil diinstall atau belum, berikut langkah-langkahnya:

1. Pastikan XAMPP sudah diaktifkan (Apache) dengan mengklik tombol Start (ini tidak harus)
2. Buka **Start Up** program Windows Anda, ketik **CMD**
3. Klik program **Command Prompt**.
4. Lalu ketik **composer**



Maka akan tampil seperti gambar di bawah ini. Anda telah berhasil menginstall Composer.

```
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\User>composer

Composer version 1.6.5 2018-05-04 11:44:59

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.
  -vv|vvv, --verbose        Increase the verbosity of messages: 1 for normal output, 2 for more verbose ou
                             tput and 3 for debug

Available commands:
  about                Shows the short information about Composer.
  archive              Creates an archive of this composer package.
  browse               Opens the package's repository URL or homepage in your browser.
  check-platform-reqs Check that platform requirements are satisfied.
  clear-cache           Clears composer's internal package cache.
```

Instalasi Composer Berhasil

2. Git

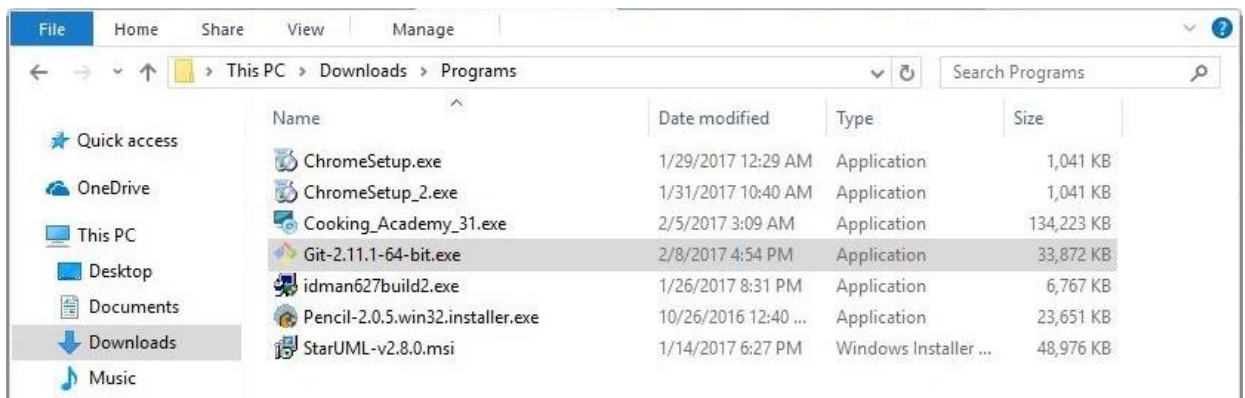
Instalasi Git di Windows memang tidak seperti di Linux yang ketik perintah langsung terinstal. Tapi dalam ritual tersebut, ada pilihan yang harus diperhatikan agar perintah **git** dapat dikenali di CMD.

➤ Download Git

Silahkan buka website resminya Git (git-scm.com). Kemudian unduh Git sesuai dengan arsitektur komputer kita. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.

➤ Langkah-langkah Install Git di Windows

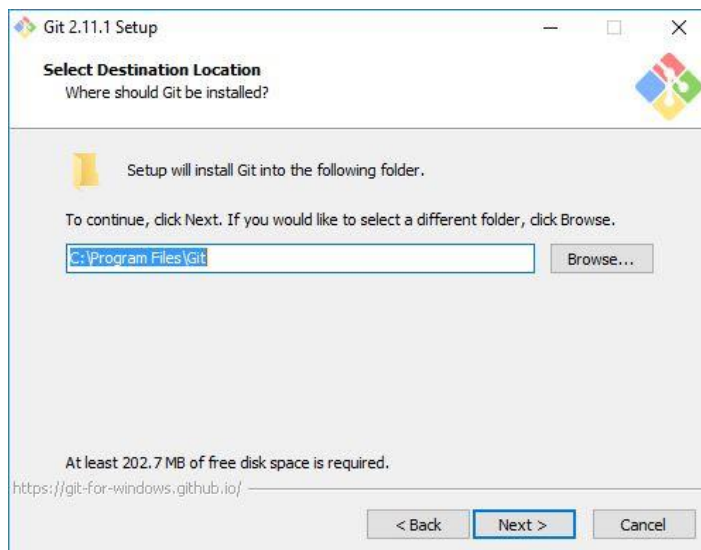
Baiklah, mari kita mulai ritual instalnya. Silahkan klik 2x file instaler Git yang sudah diunduh.



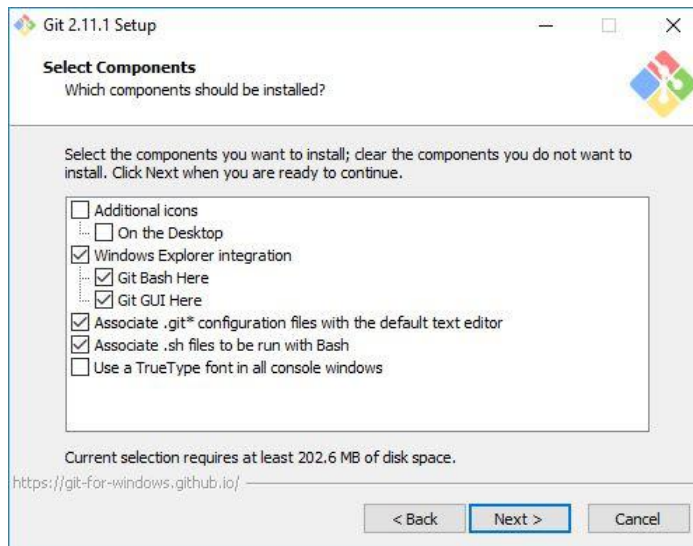
Maka akan muncul informasi lisensi Git, klik *Next >* untuk melanjutkan.



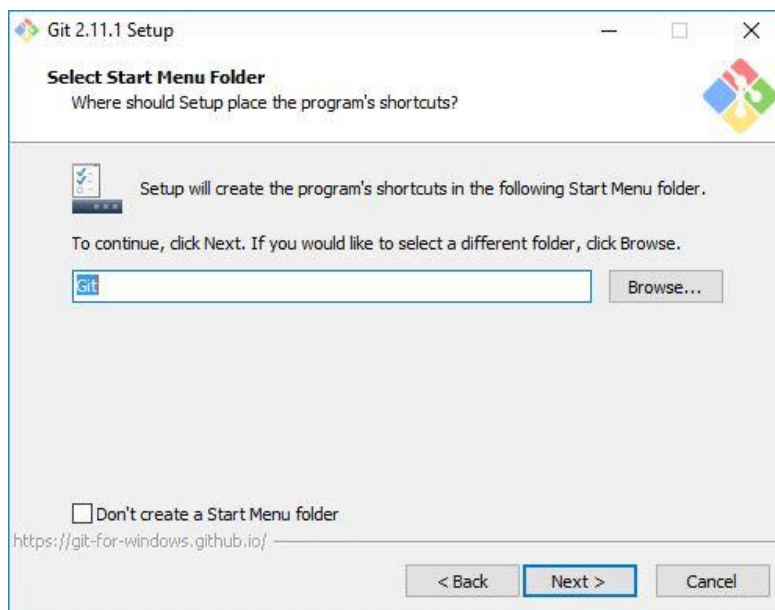
Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik *Next >*.



Selanjutnya pemilihan komponen, biarkan saja seperti ini kemudian klik *Next >*.



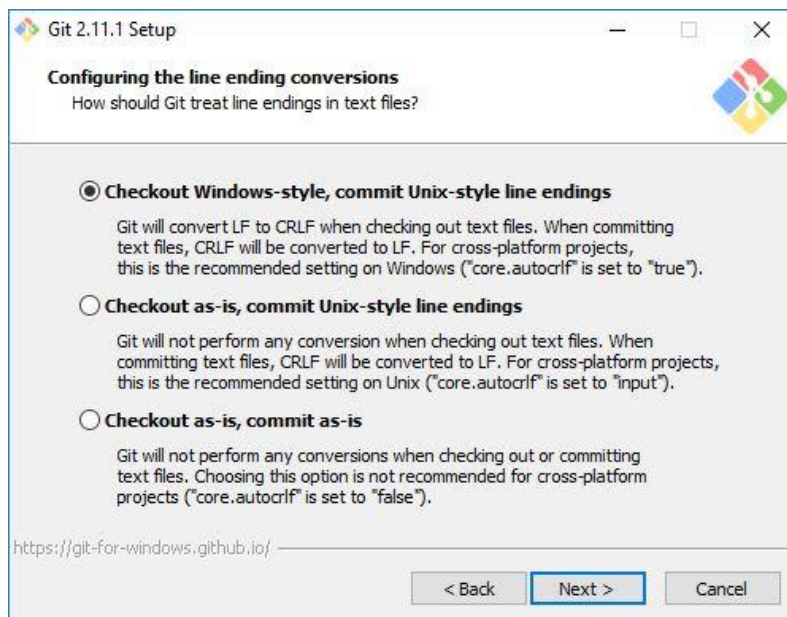
Selanjutnya pemilihan direktori start menu, klik *Next >*.



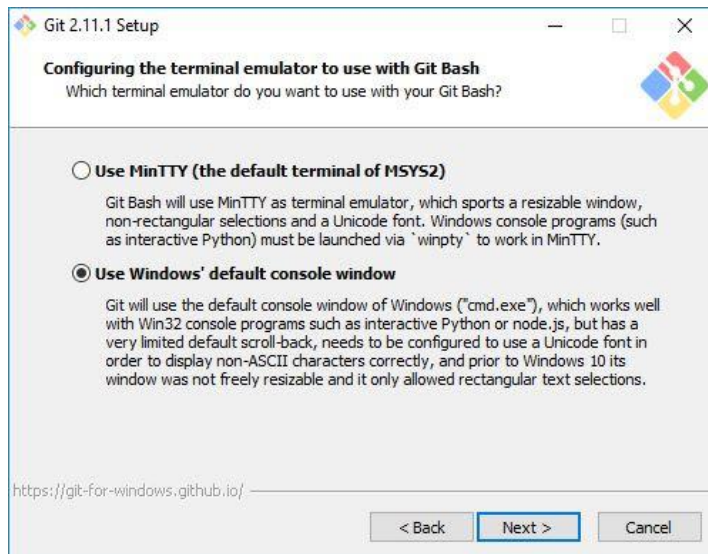
Selanjutnya pengaturan *PATH Environment*. Pilih yang tengah agar perintah *git* dapat di kenali di *Command Prompt (CMD)*. Setelah itu klik *Next >*.



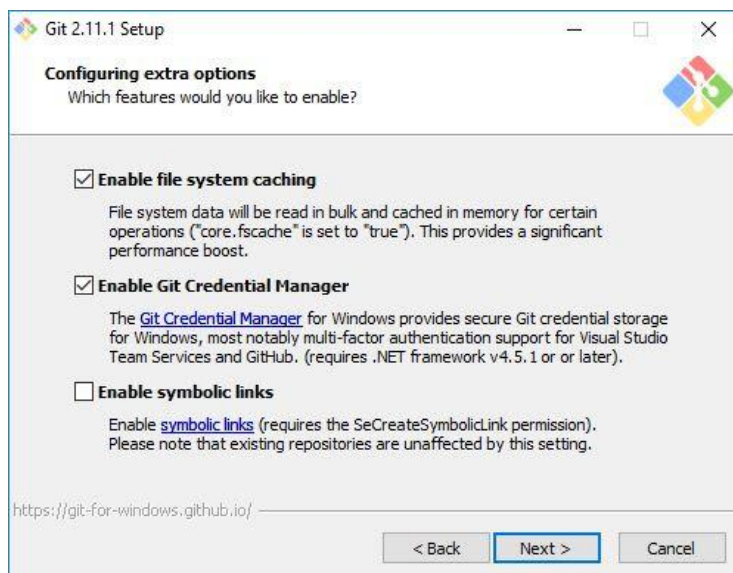
Selanjutnya konversi *line ending*. Biarkan saja seperti ini, kemudian klik *Next >*.



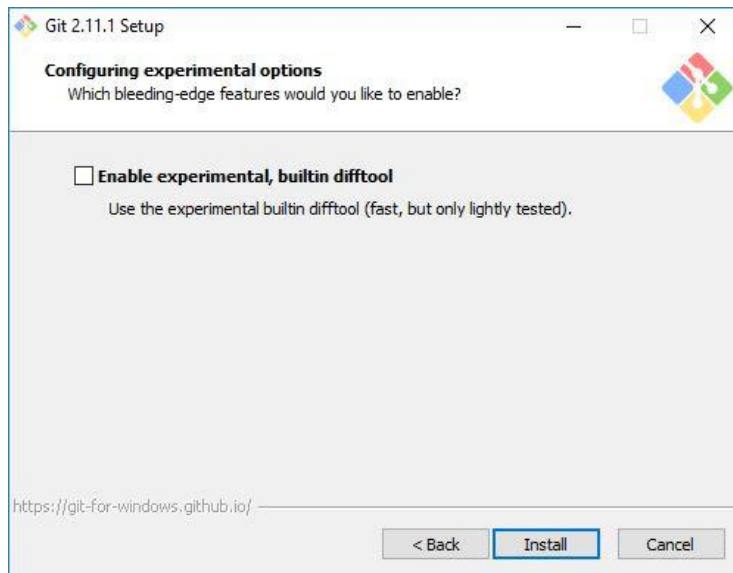
Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, kemudian klik *Next >*.



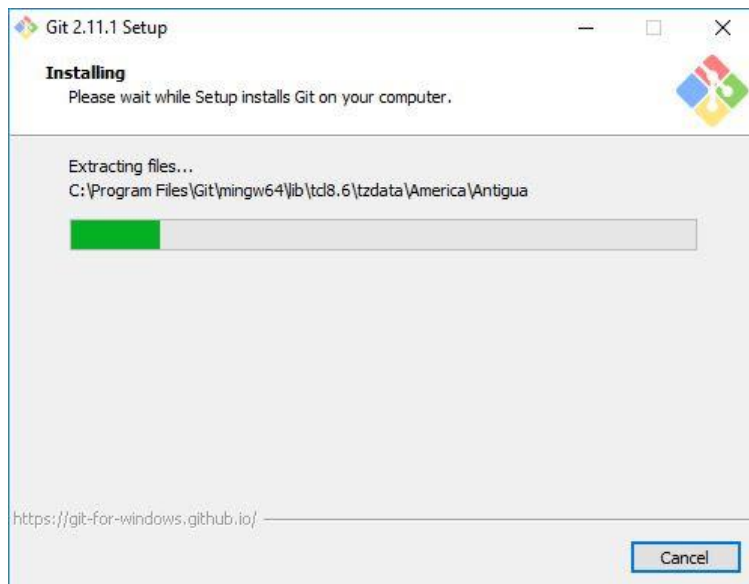
Selanjutnya pemilihan opsi ekstra. Klik saja *Next >*.



Selanjutnya pemilihan opsi ekspreimental, langsung saja klik *Install* untuk memaulai instalasi.



Tunggu beberapa saat, instalasi sedang dilakukan.



Setelah selesai, kita bisa langsung klik *Finish*.



Selamat, Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah `git --version`.

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Development>git --version
git version 2.11.1.windows.1

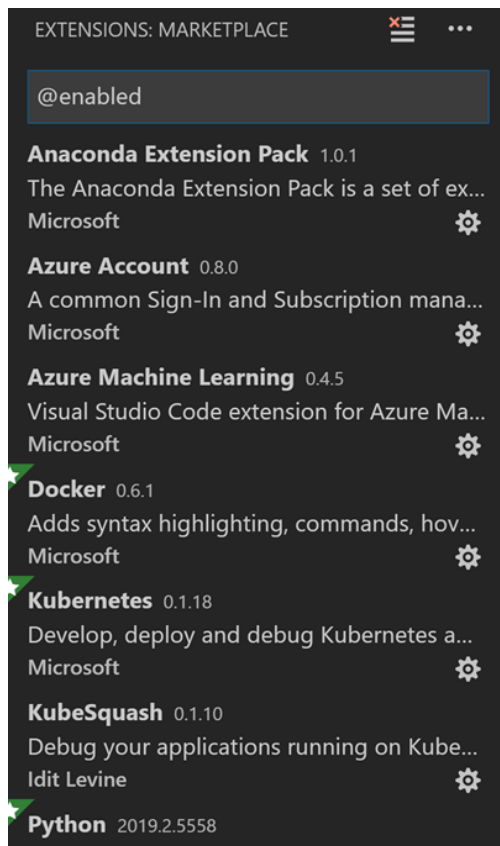
C:\Users\Development>
```

3. Visual studio Code

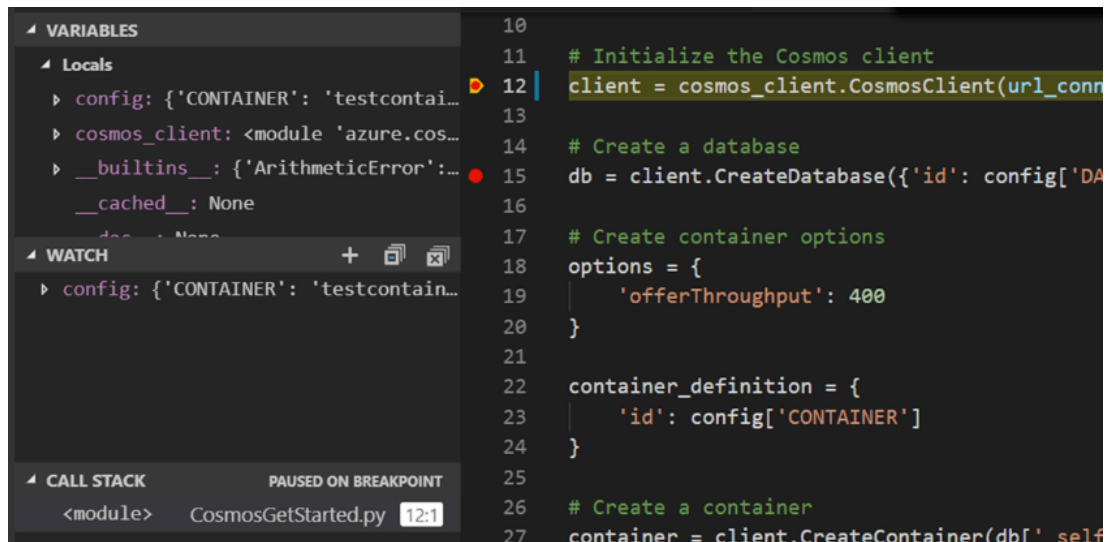
Microsoft Visual Studio Code adalah one-stop shop yang memungkinkan kita fokus pada proses pengembangan dan melupakan tools baru. beberapa fitur Visual Studio Code:

- **Cross platform** – tersedia di macOS, Linux dan Windows artinya Anda dapat bekerja pada sistem operasi manapun tanpa khawatir belajar coding tools yang sama untuk sistem yang berbeda-beda.
- **Lightweight** – tak perlu menunggu lama untuk memulai. Anda mengontrol sepenuhnya bahasa, tema, debugger, commands dan lain-lainnya sesuai keinginan. Ini dapat dilakukan melalui extentions untuk bahasa populer

seperti python, node.js, java dan lain-lainnya di Visual Studio Code Marketplace.



- **Powerful editor** – memfungsikan fitur untuk source code editing yang sangat produktif, seperti membuat code snippets, IntelliSense, auto correct, dan formatting.
- **Code Debugging** – salah satu fitur terkeren yang ditawarkan Visual Studio Code, adalah membantu Anda melakukan *debug* pada kode dengan cara mengawasi kode, variabel, call stack dan expression yang mana saja.



- **Source control** – Visual Studio Code memiliki integrated source control termasuk Git support in-the-box dan penyedia source code control lainnya di pasaran. Ini meningkatkan siklus rilis proyek Anda secara signifikan.
- **Integrated terminal** – Tiada lagi *multiple windows* dan alt-tabs. Anda dapat melakukan command-line task sekejap dan membuat banyak terminal di dalam editor.

4. Cara Menginstall Visual Studio Code di Windows

Visual Studio Code merupakan salah satu text editor yang paling populer dikalangan Web developer diseluruh dunia. Para web developer itu yang mengembangkan aplikasi web menggunakan ASP.NET, Node.js, HTML, CSS, Less, Sass, dan JSON. Seperti editor pada umumnya VSCode memiliki fitur *syntax coloring* dan *bracket matching*. Bahasa pemrograman yang mendukung fitur tadi adalah Batch, C++, Closure, Coffee Script, DockerFile, F#, Go, Jade, Java, HandleBars, Ini, Lua, Makefile, Markdown, Objective-C, Perl, PHP, PowerShell, Python, R, Razor, Ruby, SQL, Visual Basic, dan XML.

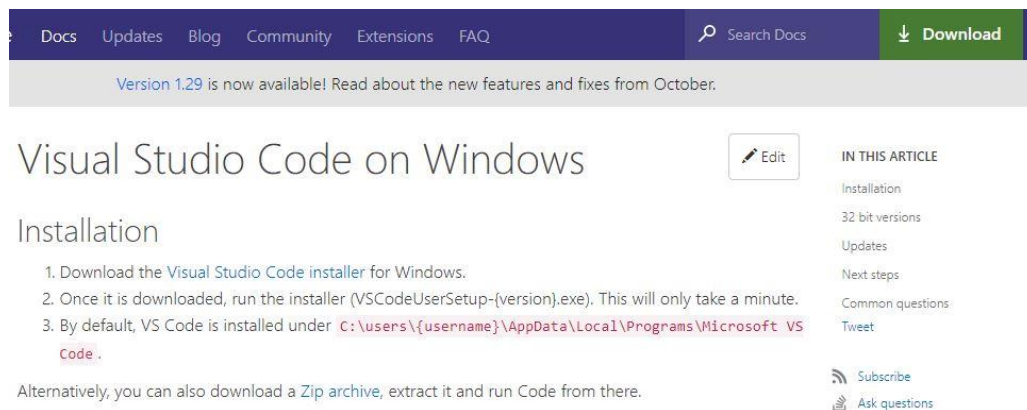
Spesifikasi yang dibutuhkan:

- **Hardware**, Perangkat keras rekomendasi yang dibutuhkan yaitu memiliki processor 1.6 GHz or faster processor, dan minimal memiliki 1 GB RAM.
- **Sistem operasi**, Sistem operasi di windows minimal kalau memakai windows 7 wajib Menginstal [.NET Framework 4.5.2](#) untuk sistem operasi lain

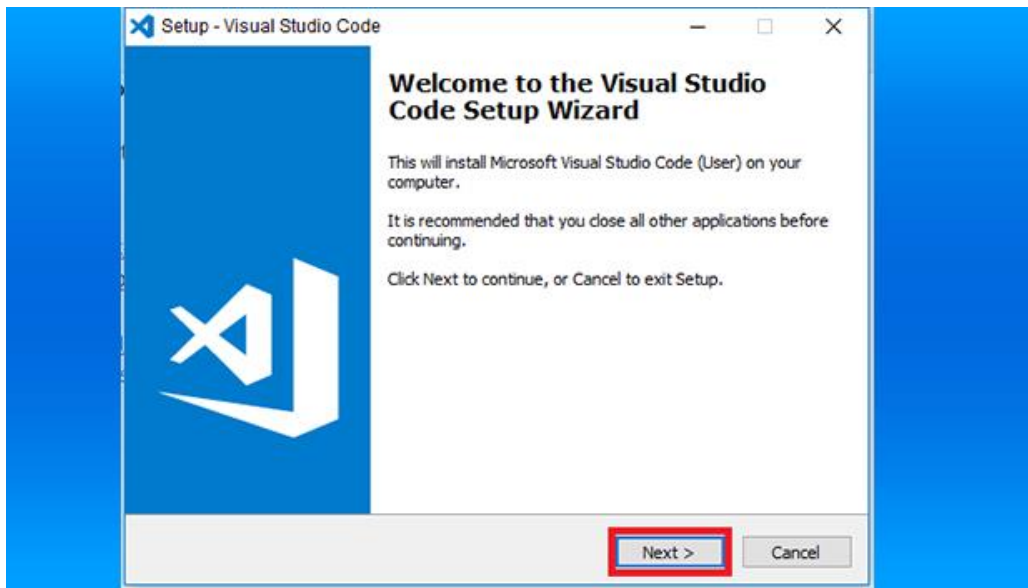
yaitu: OS X Yosemite, Windows 7, 8.0, 8.1 and 10 (32-bit and 64-bit), Linux (Debian): Ubuntu Desktop 14.04, Debian 7, Linux (Red Hat): Red Hat Enterprise Linux 7, CentOS 7, Fedora 23

Selanjutnya jika sudah disesuaikan spesifikasinya kita lanjut mendownload terlebih dahulu VSCode nya. Untuk operasi Windows link Download nya silahkan dibuka [disini](#).

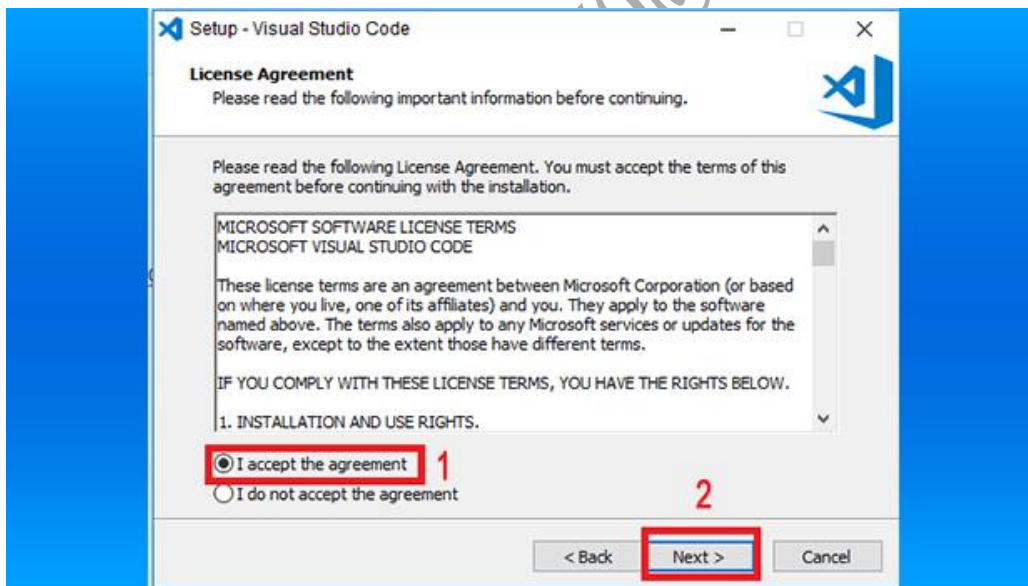
Jika sudah di buka silahkan klik download di pojok kanan atas atau juga bisa dengan mengklik “Download the [Visual Studio Code installer](#) for Windows.” seperti gambar di bawah ini.



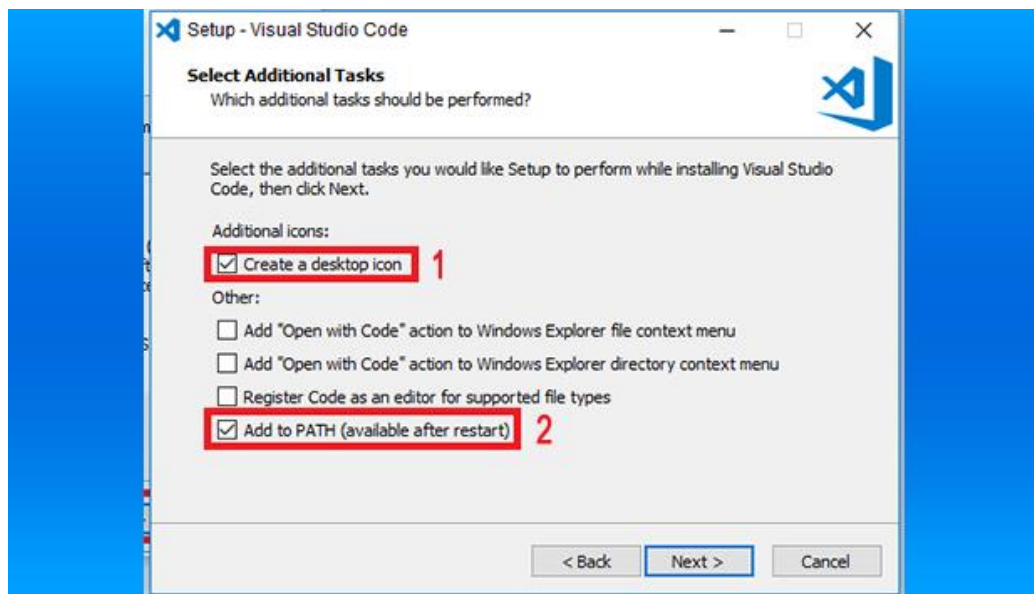
Jika sudah klik/buka Aplikasi VSCode yang sudah di download, Kemudian ikuti kotak merah pada gambar klik *Next*.



Kemudian pilih “*I accept the agreement*” untuk menyetujui kebijakan dari VSCode lalu klik *Next*.



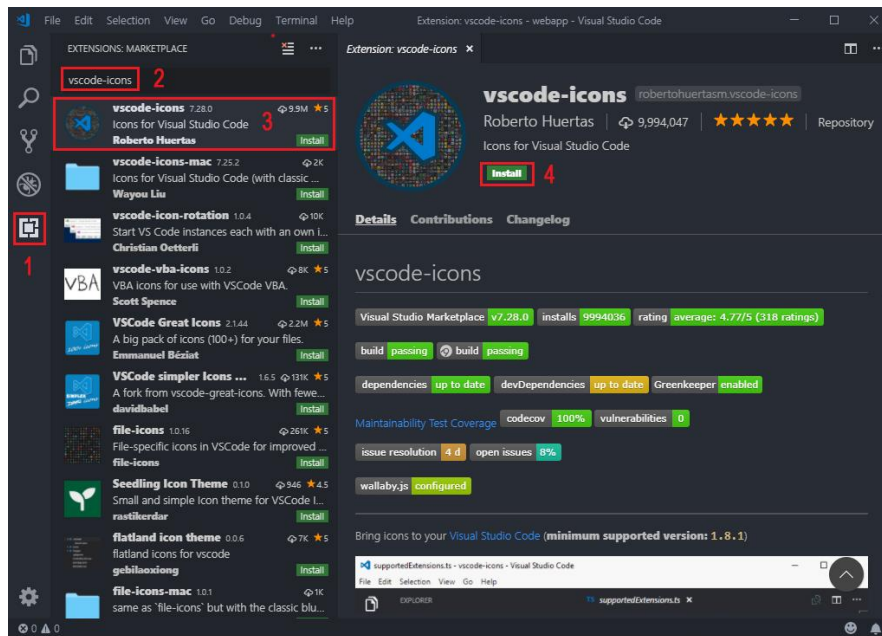
Kemudian klik *Next* lagi itu hanya penempatan direktori file programnya, lalu *Next* untuk membuat folder nya, Kemudian Ceklis “*Create Desktop Icon*” jika ingin membuat shortcut VSCode nya kemudian ceklis “*Add to PATH (available after restart)*” jika sudah klik *Next*.



kemudian Klik *Install* tunggu sampai “*Setup has finished installing Visual Studio Code on your komputer*” . kemudian klik *Finish* seperti gambar dibawah ini.



Silahkan anda jalankan VS code yang baru saja selesai anda install



#Membuat Fungsi CRUD Pada Laravel

1. Install Laravel

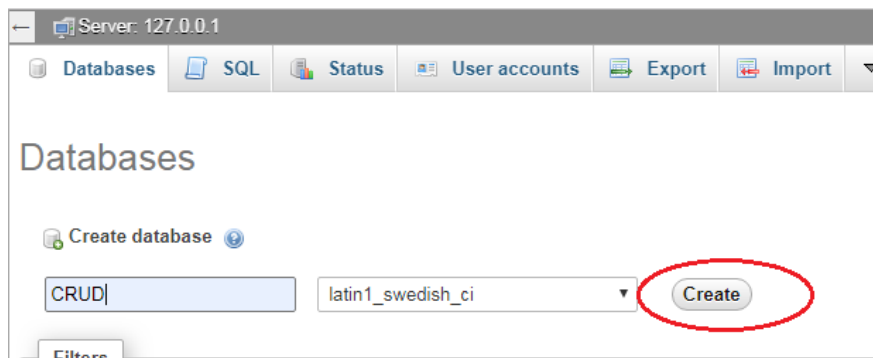
Setelah semua tools yang dibutuhkan terinstall maka pada commandprompt jalankan perintah ***composer create-project --prefer-dist Laravel/Laravel CRUDmhs***

```

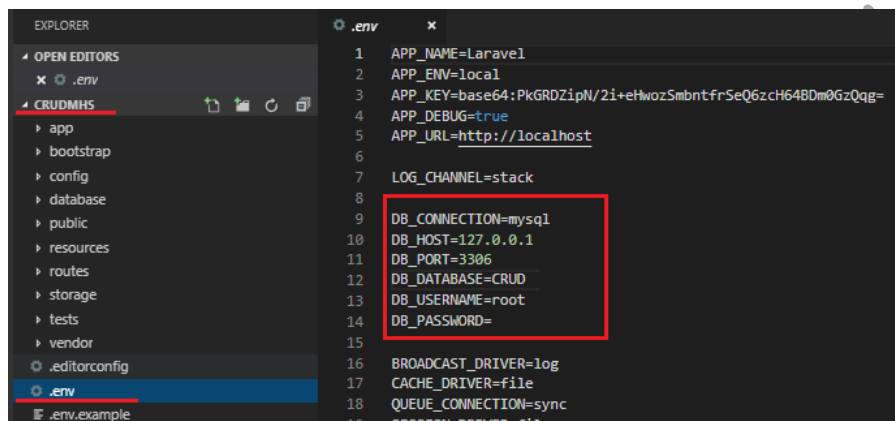
C:\>composer create-project --prefer-dist laravel/laravel CRUDmhs
Installing laravel/laravel (v5.8.3)
- Installing laravel/laravel (v5.8.3): Downloading (100%)
Created project in CRUDmhs
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 76 installs, 0 updates, 0 removals
- Installing symfony/polyfill-ctype (v1.11.0): Loading from cache
- Installing phoption/phoption (1.5.0): Downloading (100%)

```

Buatlah database yang akan kita gunakan terlebih dahulu, buatlah database dengan nama ***CRUD***

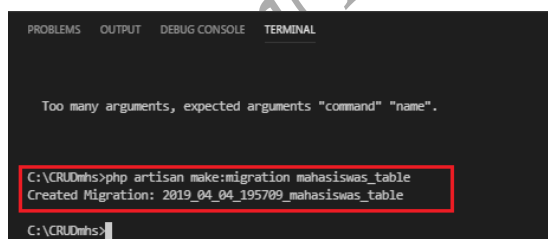


Bukalah kode editor dan pastikan anda aktif pada direktori yang baru dibuat, kemudian buka file .ENV, kemudian buatlah pengaturan tentang koneksi database

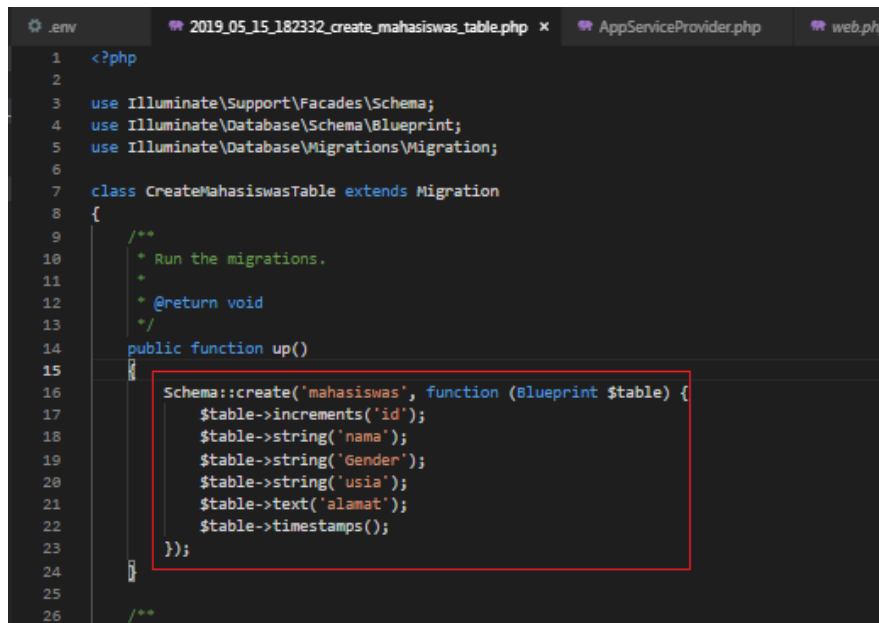


2. Membuat migrasi database

Jalankan perintah **php artisan make:migration create_mahasiswas_table** pada commandprompt untuk membuat migrasi database

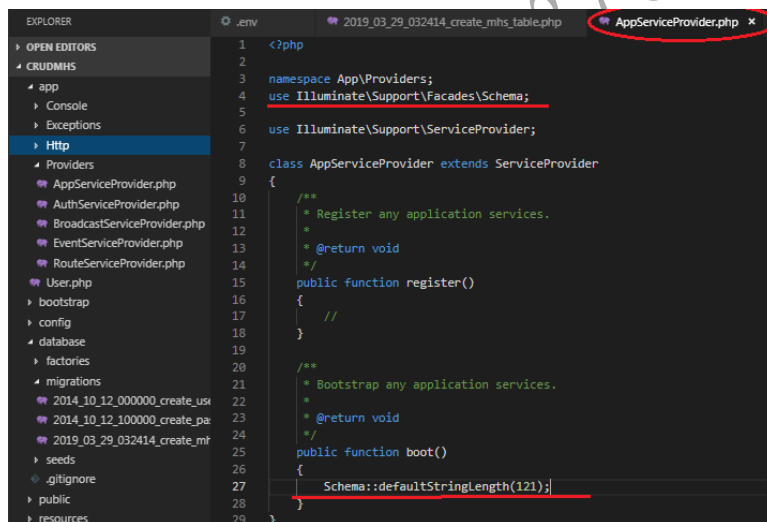


Bukalah file **xxx_xx_create_mhs_table.php** yang baru saja dibuat dan terdapat didalam folder **databases/migrations/ xxx_xx_create_mahasiswa_table.php**, lalu tambahkan sourcecode pada bagian **public function up()**. Perhatikan gambar dibawah ini



```
1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateMahasiswasTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('mahasiswas', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('nama');
19             $table->string('Gender');
20             $table->string('usia');
21             $table->text('alamat');
22             $table->timestamps();
23         });
24     }
25
26     /**
```

Untuk mencegah terjadinya error saat migrasi table, maka ubahlah **file** **AppServiceProvider.php** yang terdapat didalam folder **app/providers/** **AppServiceProvider.php**



```
1 <?php
2
3 namespace App\Providers;
4 use Illuminate\Support\Facades\Schema;
5
6 use Illuminate\Support\ServiceProvider;
7
8 class AppServiceProvider extends ServiceProvider
9 {
10     /**
11      * Register any application services.
12      *
13      * @return void
14      */
15     public function register()
16     {
17         //
18     }
19
20     /**
21      * Bootstrap any application services.
22      *
23      * @return void
24      */
25     public function boot()
26     {
27         Schema::defaultStringLength(121);
28     }
29 }
```

Selanjutnya jalankan perintah **php artisan migrate** untuk membuat migrasi tabel mhs dan strukturnya

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\CRUDmhs>php artisan migrate
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2019_04_04_195709_mahasiswas_table
Migrated: 2019_04_04_195709_mahasiswas_table

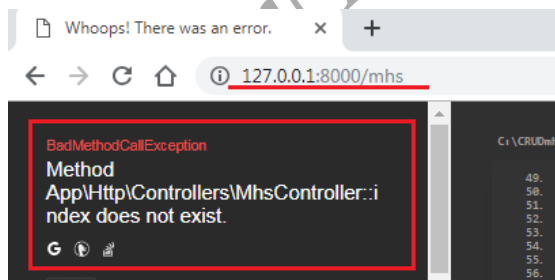
C:\CRUDmhs>
```

3. Membuat route

Untuk mengaktifkan fitur login maka Jalankan perintah **php artisan make:auth**, pada **commandprompt** untuk mengaktifkan fungsi login. Selanjutnya bukalah file **app/routes/web.php**, kemudian buatlah routing baru dengan nama mhs

```
1 <?php
2
3 /*
4 |-----
5 | Web Routes
6 |-----
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('welcome');
16 });
17
18 Auth::routes();
19 Route::get('/home', 'HomeController@index')->name('home');
20 Route::resource('mhs', 'MahasiswaController');
21
22
```

Silahkan anda buka pada browser anda dan perhatikan apa yang terjadi

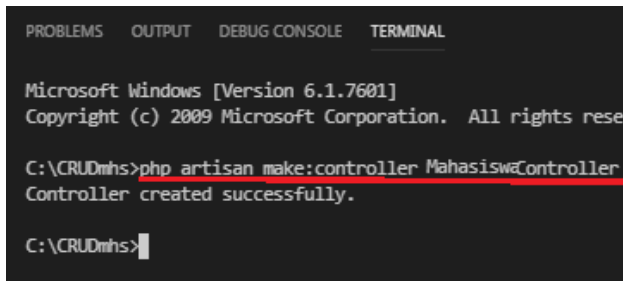


Error tersebut terjadi karena kita belum membuat file controller dengan nama MhsController,

4. Membuat controller

Sekarang buatlah file tersebut pada **commandprompt** dengan menjalankan perintah **php artisan make:controller MahasiswaController --resource** Jika berhasil maka

didalam **folder app/Http/controllers** akan tercipta file baru dengan nama **MahasiswaController**



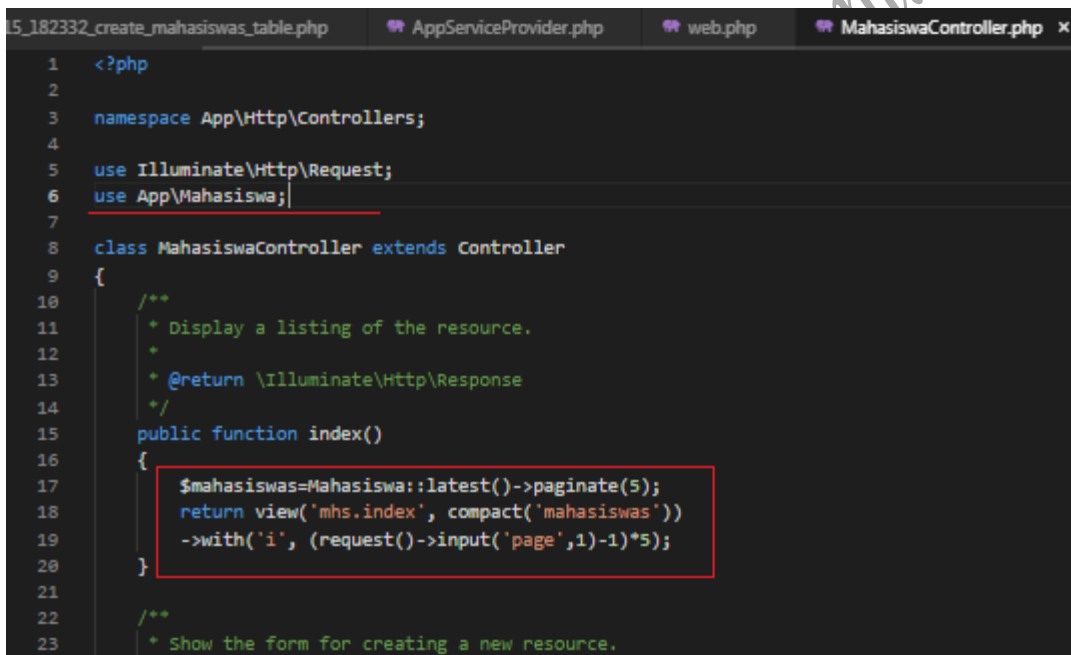
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\CRUDmhs>php artisan make:controller MahasiswaController
Controller created successfully.

C:\CRUDmhs>
```

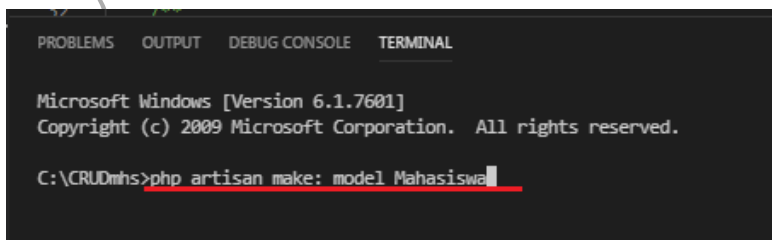
Bukalah file **MahasiswaController** yang baru saja kita buat, dan tambahkan sebuah fungsi baru dengan nama index



```
182332_create_mahasiswas_table.php AppServiceProvider.php web.php MahasiswaController.php x
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Mahasiswa;
7
8 class MahasiswaController extends Controller
9 {
10     /**
11      * Display a listing of the resource.
12      *
13      * @return \Illuminate\Http\Response
14      */
15     public function index()
16     {
17         $mahasiswas=Mahasiswa::latest()->paginate(5);
18         return view('mhs.index', compact('mahasiswas'))
19             ->with('i', (request()->input('page',1)-1)*5);
20     }
21
22     /**
23      * Show the form for creating a new resource.
```

5. Membuat model

Pada commandprompt Jalankan perintah **php artisan make:model Mahasiswa**



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

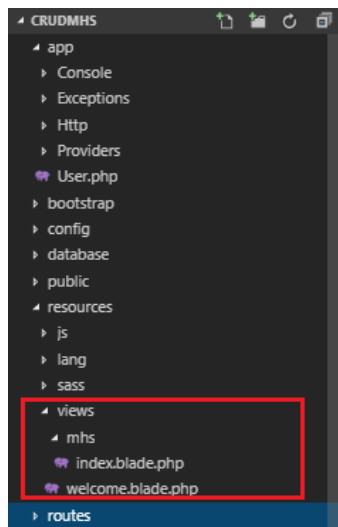
C:\CRUDmhs>php artisan make:model Mahasiswa
```

Selanjutnya tambahkan perintah protected \$fillable pada model mahasiswa


```
.env 2019_05_15_182332_create_m... Mahasiswa.php x AppServiceProvider.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Mahasiswa extends Model
8 {
9     protected $fillable = ['nama', 'Gender', 'usia', 'alamat'];
10 }
11
```

6. Menampilkan data

Buatlah folder baru dengan nama **mhs** didalam **folder resources/views/mhs**, kemudian buatlah file dengan nama **index.blade.php** didalam folder mhs yang baru saja dibuat



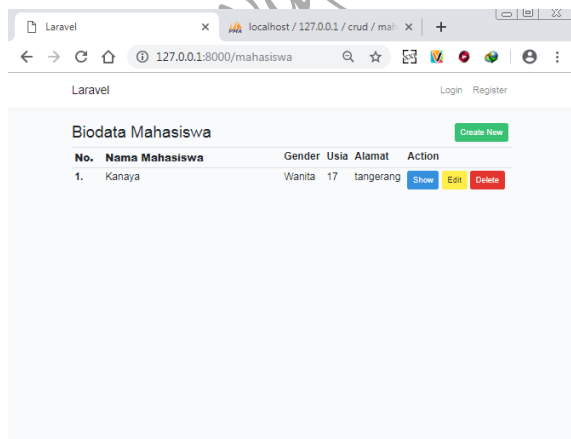
Masukan sourcecode berikut didalam **file index.blade.php**

```

1 @extends('layouts.app')
2 @section('content')
3     <div class="container">
4         <div class="row">
5             <div class="col-md-10">
6                 <h3>Biodata Mahasiswa</h3>
7             </div>
8             <div class="col-sm-2">
9                 <a class="btn btn-sm btn-success" href="#">Create New</a>
10            </div>
11        </div>
12        @if ($message = Session::get('success'))
13            <div class="alert alert-success">
14                <p>{{ $message }}</p>
15            </div>
16        @endif
17
18        <table class="table table-hover table-sm">
19            <tr>
20                <th width="50px"><b>No.</b></th>
21                <th width="300px"><b>Nama Mahasiswa</b></th>
22                <th>Gender</th>
23                <th>Usia</th>
24                <th>Alamat</th>
25                <th width="180px">Action</th>
26            </tr>
27
28            @foreach ($mahasiswas as $item)
29                <tr>
30                    <td><b>{{++$i}}</b></td>
31                    <td>{{ $item->nama }}</td>
32                    <td>{{ $item->Gender }}</td>
33                    <td>{{ $item->usia }}</td>
34                    <td>{{ $item->alamat }}</td>
35                    <td>
36                        <form action="#" method="post">
37                            <a class="btn btn-sm btn-primary" href="#">Show</a>
38                            <a class="btn btn-sm btn-warning" href="#">Edit</a>
39
40                            @csrf
41                            @method('DELETE')
42
43                            <button type="submit" class="btn btn-sm btn-danger">Delete</button>
44                        </form>
45                    </td>
46                </tr>
47            @endforeach
48        </table>
49    @endsection
50    {!! $mahasiswas->links() !!}
51</div>

```

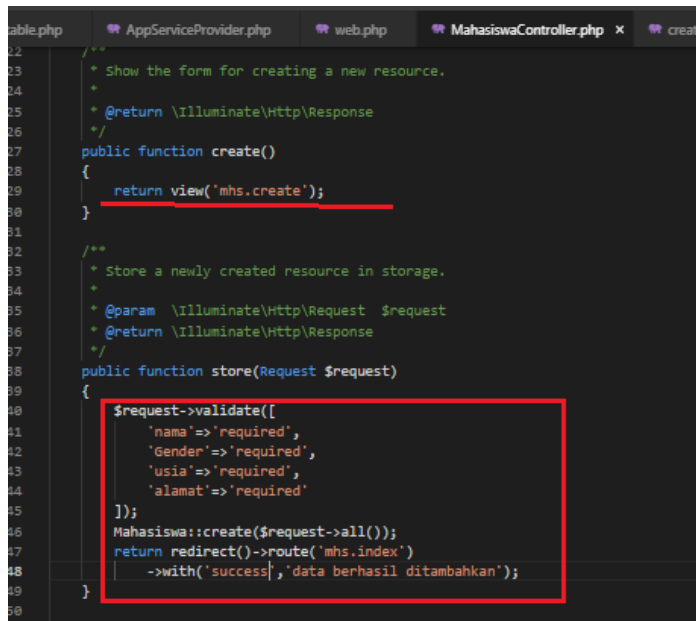
Silahkan anda jalankan browser anda dan perhatikan perubahan yang terjadi
(pastikan anda memiliki dummy data yang akan ditampilkan pada halaman)



Pada halaman mahasiswa diatas, sudah berhasil menampilkan data yang ada didalam database dan juga sudah kita sudah berhasil membuat *button create, show, edit serta delete data, akan tetapi tombol-tombol tersebut belum berfungsi sebagaimana mestinya*, oleh karena itu maka tahap selanjutnya kita akan membuat fungsi-fungsi tersebut

7. Membuat Fungsi Create

Untuk membuat fungsi create, silahkan anda buka file *MahasiswaController*, kemudian pada bagian fungsi create



```
22
23  /**
24   * Show the form for creating a new resource.
25   *
26   * @return \Illuminate\Http\Response
27   */
28  public function create()
29  {
30      return view('mhs.create');
31  }
32
33  /**
34   * Store a newly created resource in storage.
35   *
36   * @param \Illuminate\Http\Request $request
37   * @return \Illuminate\Http\Response
38   */
39  public function store(Request $request)
40  {
41      $request->validate([
42          'nama'=>'required',
43          'Gender'=>'required',
44          'usia'=>'required',
45          'alamat'=>'required'
46      ]);
47      Mahasiswa::create($request->all());
48      return redirect()->route('mhs.index')
49          ->with('success','data berhasil ditambahkan');
50  }
```

Kemudian didalam *folder mhs*, buatlah file baru dengan nama *create.blade.php*, dan masukan source code berikut

```

1  @extends('layouts.app')
2  @section('content')
3      <div class="container">
4          <div class="row">
5              <div class="col-lg-12">
6                  <h3>New Biodata</h3>
7              </div>
8          </div>
9      </div>
10
11  @if ($errors->any())
12      <div class="alert alert-danger">
13          <strong>Whoops !!</strong> terjadi kesalahan input data.<br>
14          <ul>
15              @foreach ($errors as $error)
16                  <li>{{ $error }}</li>
17              @endforeach
18          </ul>
19      </div>
20  @endif
21
22  <form action="{{ route('mhs.store') }}" method="post">
23      @csrf
24      <div class="row">
25          <div class="col-sm-8">
26              <strong>Nama :</strong>
27              <input type="text" name="nama" class="form-control" placeholder="Nama">
28          </div>
29          <div class="col-sm-8">
30              <strong>Gender :</strong>
31              <input type="text" name="Gender" class="form-control" placeholder="Gender">
32          </div>
33          <div class="col-sm-8">
34              <strong>Usia :</strong>
35              <input type="text" name="usia" class="form-control" placeholder="Usia">
36          </div>
37          <div class="col-sm-8">
38              <strong>Alamat :</strong>
39              <textarea class="form-control" placeholder="Alamat" name="alamat" rows="4" cols="80"></textarea>
40          </div>
41
42          <div class="col-md-12">
43              <a href="{{ route('mhs.index') }}" class="btn btn-sm btn-success">Back</a>
44              <button type="submit" class="btn btn-sm btn-primary">Submit</button>
45          </div>
46      </div>
47  </form>
48  </div>
49  @endsection

```

Pada file index.blade.php, silahkan anda ubah **link button create**, seperti dibawah ini

```

1  @extends('layouts.app')
2  @section('content')
3      <div class="container">
4          <div class="row">
5              <div class="col-md-10">
6                  <h3>Biodata Mahasiswa</h3>
7              </div>
8              <div class="col-sm-2">
9                  <a class="btn btn-sm btn-success" href="{{ route('mhs.create') }}">Create New</a>
10              </div>
11          </div>

```

Silahkan anda coba **refresh browser**, dan jalankan **fungsi button create**

The screenshot shows a web browser window with the URL `localhost / 127.0.0.1 / crud / mah`. The page title is "Laravel" and it includes "Login" and "Register" links. The main heading is "New Biodata". The form contains the following fields:

- Nama :** A text input field with the placeholder "Nama".
- Jenis Kelamin :** A text input field with the placeholder "Jenis_Kelamin".
- Usia :** A text input field with the placeholder "Usia".
- Alamat :** A larger text input field with the placeholder "Alamat".

At the bottom of the form are two buttons: a green "Back" button and a blue "Submit" button.

Silahkan anda coba tambahkan data baru, jika berhasil maka anda diarahkan ke halaman mahasiswa.index

The screenshot shows the "Biodata Mahasiswa" page. It features a "Create New" button in the top right corner. Below it is a green success message box that says "data baru telah berhasil dibuat". The main content is a table with the following data:

No.	Nama Mahasiswa	Gender	Usia	Alamat	Action
1.	Belai	laki-laki	20	Jl. Kemuning	Show Edit Delete
2.	alfa	perempuan	19	Jakarta Barat	Show Edit Delete
3.	elzio	laki-laki	16	jakarta	Show Edit Delete
4.	Kanaya	perempuan	17	tangerang	Show Edit Delete

8. Membuat Fungsi Read

Buatlah file baru dengan nama *detail.blade.php* didalam *folder mhs*

Kemudian bukalah file *detail.blade.php* tersebut dan masukan *sourcecode* dibawah ini.

```
1 @extends('layouts.app')
2 @section('content')
3 <div class="container">
4     <div class="row">
5         <div class="col-md-12">
6             <h3>Detail Mahasiswa</h3>
7             <hr>
8         </div>
9     </div>
10    <div class="row">
11        <div class="col-md-12">
12            <div class="form-group">
13                <strong>Nama :</strong> {{ $mahasiswa-> nama }}
14            </div>
15        </div>
16    </div>
17    <div class="row">
18        <div class="col-md-12">
19            <div class="form-group">
20                <strong>Jenis Kelamin :</strong> {{ $mahasiswa-> Gender }}
21            </div>
22        </div>
23    </div>
24    <div class="row">
25        <div class="col-md-12">
26            <div class="form-group">
27                <strong>Usia :</strong> {{ $mahasiswa-> usia }}
28            </div>
29        </div>
30    </div>
31    <div class="row">
32        <div class="col-md-12">
33            <div class="form-group">
34                <strong>Alamat :</strong> {{ $mahasiswa-> alamat }}
35            </div>
36        </div>
37    </div>
38    <div class="row">
39        <div class="col-md-12">
40            <a href="{{ route('mhs.index') }}" class="btn btn-sm btn-success">back</a>
41        </div>
42    </div>
43 </div>
44
45 @endsection
```

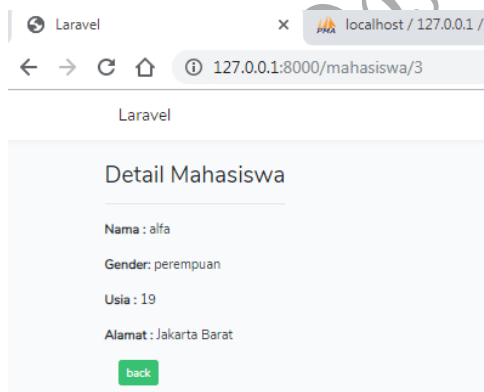
Didalam file *MahasiswaController*, dibagian *function show*, masukan sourcecode berikut


```
ServiceProvider.php  web.php  MahasiswaController.php x  create.blade.php
56  /*
57  public function show($id)
58  {
59      $mahasiswas = Mahasiswa::find($id);
60      return view('mhs.detail', compact('mahasiswas'));
61  }
62
63  /**
```

Pada file **index.blade.php**, ubahlah sourcecode sesuai gambar dibawah ini, dan buatlah link untuk button show

```
ServiceProvider.php  web.php  MahasiswaController.php  create.blade.php  Mahasiswa.php  index.blade.php x
28  @foreach ($mahasiswas as $mahasiswa)
29      <tr>
30          <td><b>{{ ++$i }}</b></td>
31          <td>{{ $mahasiswa->nama }}</td>
32          <td>{{ $mahasiswa->Gender }}</td>
33          <td>{{ $mahasiswa->usia }}</td>
34          <td>{{ $mahasiswa->alamat }}</td>
35          <td>
36              <form action="#" method="post">
37                  <a class="btn btn-sm btn-primary" href="{{ route('mhs.show', $mahasiswa->id) }}">Show</a>
38                  <a class="btn btn-sm btn-warning" href="#">Edit</a>
39              </form>
40          </td>
41      </tr>
42  </foreach>
43  @csrf
44  @method('DELETE')
```

Silahkan anda **refresh browser** anda, kemudian klik **tombol show**, jika berhasil maka tampilan pada halam web akan seperti gambar dibawah ini



9. Membuat Fungsi Edit dan Update

Buatlah file baru dengan nama **edit.blade.php** didalam **folder mhs**, lalu bukalah file tersebut dan masukan sourcecode dibawah ini.

```

1  @extends('layouts.app')
2  @section('content')
3  <div class="container">
4      <div class="row">
5          <div class="col-lg-12">
6              <h3>Edit Data</h3>
7          </div>
8      </div>
9  </div>
10
11  @if ($errors->any())
12      <div class="alert alert-danger">
13          <strong>Whoops !!</strong> terjadi kesalahan input data.<br>
14          <ul>
15              @foreach ($errors as $error)
16                  <li>{{$error}}</li>
17              @endforeach
18          </ul>
19      </div>
20  @endif
21
22  <form action="{{route('mhs.update', $mahasiswas->id)}}" method="post">
23  @csrf
24  @method('PUT')
25      <div class="row">
26          <div class="col-sm-8">
27              <strong>Nama :</strong>
28              <input type="text" name="nama" class="form-control" value="{{ $mahasiswas->nama }}">
29          </div>
30          <div class="col-sm-8">
31              <strong>Jenis Kelamin :</strong>
32              <input type="text" name="Gender" class="form-control" value="{{ $mahasiswas->Gender }}">
33          </div>
34          <div class="col-sm-8">
35              <strong>Usia :</strong>
36              <input type="text" name="usia" class="form-control" value="{{ $mahasiswas->usia }}">
37          </div>
38          <div class="col-sm-8">
39              <strong>Alamat :</strong>
40              <textarea class="form-control" name="alamat" rows="4" cols="80">{{ $mahasiswas->alamat }}</textarea>
41          </div>
42
43          <div class="col-md-12">
44              <a href="{{route('mhs.index')}}" class="btn btn-sm btn-success">Back</a>
45              <button type="submit" class="btn btn-sm btn-primary">Submit</button>
46          </div>
47      </div>
48  </form>
49  </div>
50  @endsection

```

Didalam file **MahasiswaController**, dibagian **function edit dan update**, masukan sourcecode berikut

```

67  * @return \Illuminate\Http\Response
68  */
69  public function edit($id)
70  {
71      $mahasiswas = Mahasiswa::find($id);
72      return view('mhs.edit', compact('mahasiswas'));
73  }
74
75  /**
76   * Update the specified resource in storage.
77   *
78   * @param \Illuminate\Http\Request $request
79   * @param int $id
80   * @return \Illuminate\Http\Response
81   */
82  public function update(Request $request, $id)
83  {
84      $request->validate([
85          'nama'=>'required',
86          'Gender'=>'required',
87          'usia'=>'required',
88          'alamat'=>'required'
89      ]);
90      $mahasiswas = Mahasiswa::find($id);
91      $mahasiswas->nama = $request->get('nama');
92      $mahasiswas->Gender = $request->get('Gender');
93      $mahasiswas->usia = $request->get('usia');
94      $mahasiswas->alamat = $request->get('alamat');
95      $mahasiswas->save();
96      return redirect()->route('mhs.index')
97          ->with('success', 'Data berhasil di edit..!php');
98  }
99

```

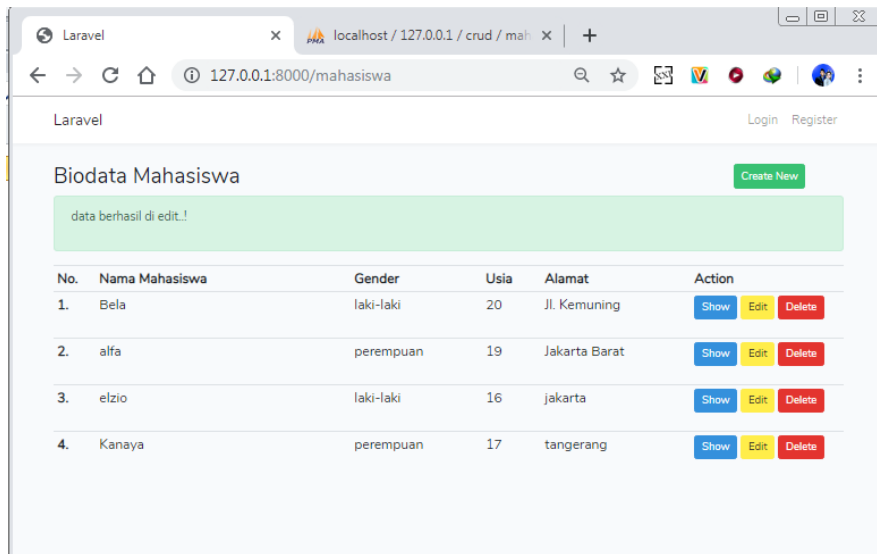
Pada file **index.blade.php**, dan buatlah link untuk button edit

```

pServiceProvider.php  web.php  index.blade.php x  MahasiswaController.php  create.blade.php  Mal
28  @foreach ($mahasiswas as $mahasiswa)
29      <tr>
30          <td><b>{{++$i}}.</b></td>
31          <td>{{ $mahasiswa->nama }}</td>
32          <td>{{ $mahasiswa->Gender }}</td>
33          <td>{{ $mahasiswa->usia }}</td>
34          <td>{{ $mahasiswa->alamat }}</td>
35      <td>
36          <form action="#" method="post">
37              <a class="btn btn-sm btn-primary" href="{{ route('mhs.show', $mahasiswa->id) }}">Show</a>
38              <a class="btn btn-sm btn-warning" href="{{ route('mhs.edit', $mahasiswa->id) }}">Edit</a>
39          </form>
40      @csrf
41      @method('DELETE')

```

Silahkan anda **refresh browser** anda, kemudian **klik tombol edit**, jika berhasil maka diarahkan pada halaman edit data, silahkan anda ubah, kemudian simpan perubahan



10. Membuat Fungsi Delete

Didalam file **MahasiswaController**, dibagian **function destroy**, masukan sourcecode berikut

```

106 public function destroy($id)
107 {
108     $mahasiswas =Mahasiswa::find($id);
109     $mahasiswas->delete();
110     return redirect()->route('mhs.index')
111         ->with('success', 'Data berhasil dihapus');
112 }
113

```

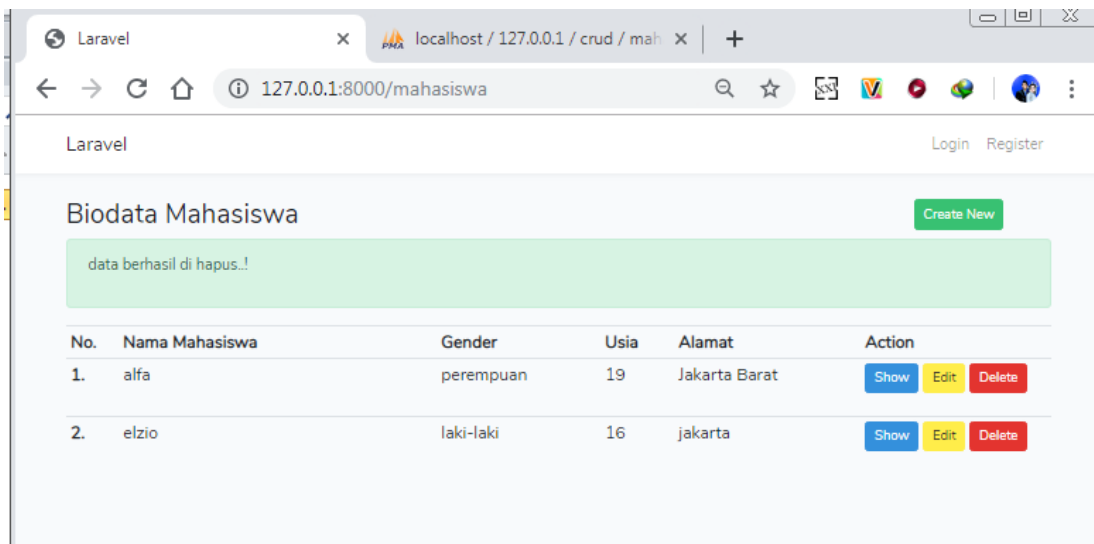
Pada file **index.blade.php**, dan buatlah link untuk button delete

```

28 @foreach ($mahasiswas as $mahasiswa)
29 <tr>
30 <td><b>{{++$i}}.</b></td>
31 <td>{{ $mahasiswa->nama}}</td>
32 <td>{{ $mahasiswa->Gender}}</td>
33 <td>{{ $mahasiswa->usia}}</td>
34 <td>{{ $mahasiswa->alamat}}</td>
35 <td>
36 <form action="{{route('mhs.destroy', $mahasiswa->id)}}" method="post">
37 <a class="btn btn-sm btn-primary" href="{{ route('mhs.show', $mahasiswa->id) }}">Show</a>
38 <a class="btn btn-sm btn-warning" href="{{ route('mhs.edit', $mahasiswa->id) }}">Edit</a>
39

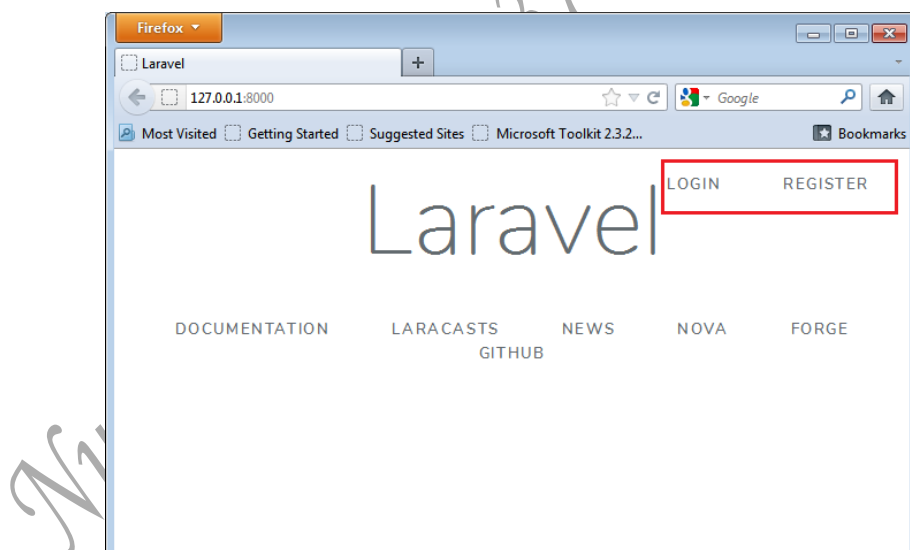
```

Silahkan anda refresh browser anda, kemudian klik tombol delete, jika berhasil maka diarahkan pada halaman index, silahkan anda ubah, kemudian simpan perubahan



Mengaktifkan Fitur Login

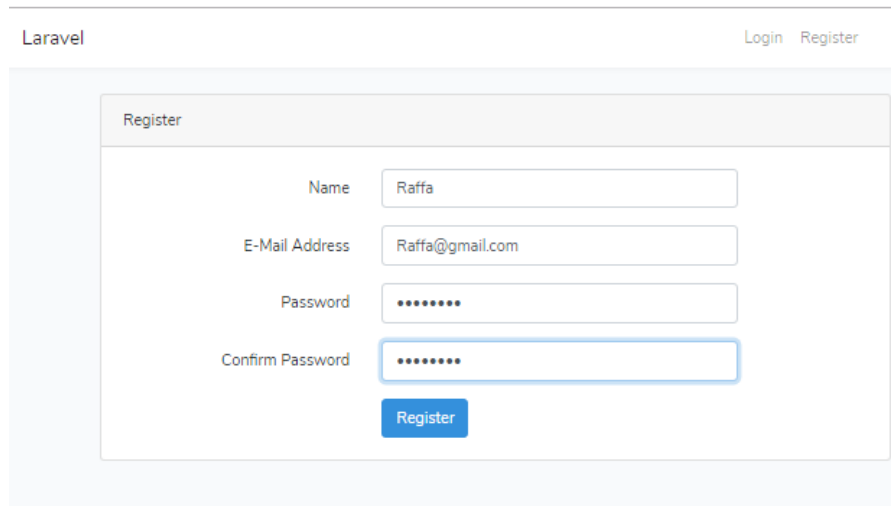
Pada tahap sebelumnya kita telah mengaktifkan fitur login dengan menjalankan perintah ***php artisan make:auth*** pada ***commandprompt***, sehingga ketika kita akses halaman yang tampil pada framework akan seperti pada gambar dibawah ini:



Pada tampilan diatas kita hanya baru mengaktifkan halaman login saja belum mengaktifkan fitur yang lainya seperti forgot password

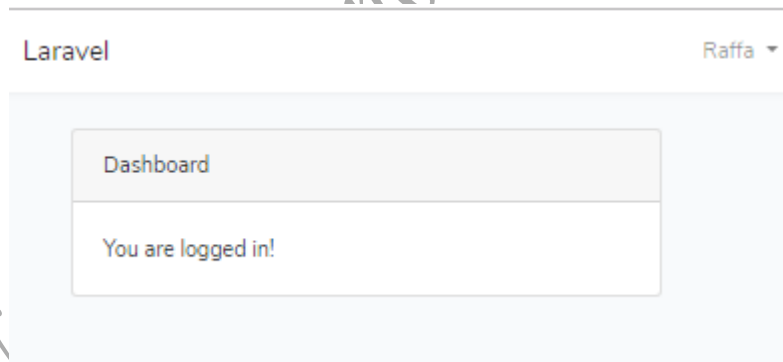
1. Mengaktifkan Fitur Register User

Silahkan anda jalankan browser anda, lalu klik menu register dan isilah form registrasi user yang tersedia



The screenshot shows the Laravel web application interface. At the top, there is a navigation bar with the 'Laravel' logo on the left and 'Login' and 'Register' links on the right. The 'Register' link is highlighted. Below the navigation bar, there is a 'Register' form. The form contains four input fields: 'Name' with the value 'Raffa', 'E-Mail Address' with the value 'Raffa@gmail.com', 'Password' with masked characters '*****', and 'Confirm Password' with masked characters '*****'. A blue 'Register' button is located at the bottom of the form.

Jika proses register berhasil maka tampilan halaman akan menunjukkan bahwa anda telah berhasil login, seperti dibawah ini



The screenshot shows the Laravel web application interface after a successful login. The navigation bar now displays the 'Laravel' logo on the left and the user's name 'Raffa' with a dropdown arrow on the right. Below the navigation bar, there is a 'Dashboard' section. The dashboard contains a message that says 'You are logged in!'.

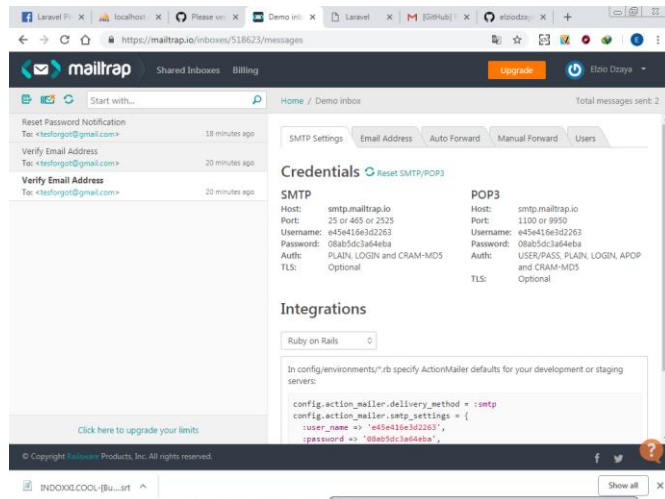
2. Mengaktifkan Fitur Forgot Password

Didalam laravel 5.8 terdapat fitur baru, yaitu mailtrap yang digunakan untuk verifikasi email, baik saat register ataupun saat lupa password ketika akan login

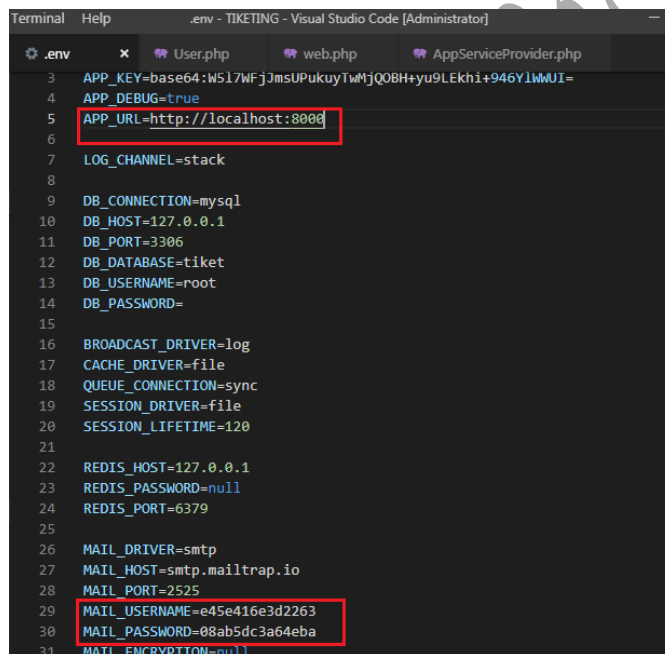
Silahkan anda buka browser anda dan masuk kedalam web mailtrap melalui url <https://mailtrap.io>

Silahkan login melalui *akun gmail* atau bisa juga melalui *akun github* anda (buat akun baru jika belum ada)

Kemudian kopi username dan password yang ada pada akun mailtrap anda

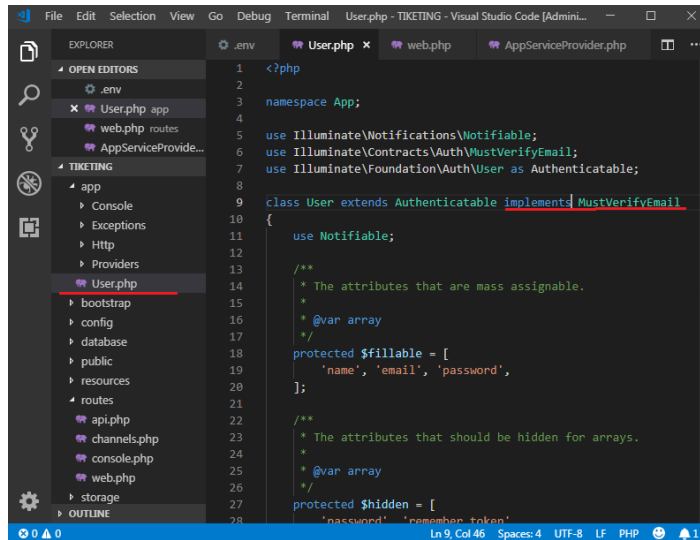


Kemudian bukalah file .env pada folder project anda, dan salinlah file username dan password yang berasal dari halaman mailtrap.io tersebut



Dan tambahkan juga perintah/code pada baris ke 5 menjadi *APP_URL =http://localhost:8000*

Selanjutnya bulaklah model User yang terdapat didalam folder App/User.php lalu ubah/tambahkan perintah pada baris ke 9 menjadi ***class User extends Authenticatable implements MustVerifyEmail***

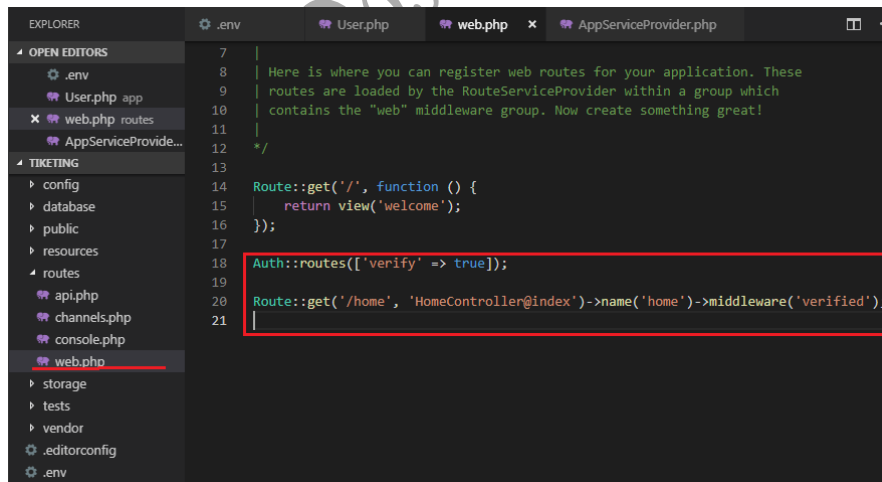


```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Notifications\Notifiable;
6 use Illuminate\Contracts\Auth\MustVerifyEmail;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8
9 class User extends Authenticatable implements MustVerifyEmail
10 {
11     use Notifiable;
12
13     /**
14      * The attributes that are mass assignable.
15      *
16      * @var array
17      */
18     protected $fillable = [
19         'name', 'email', 'password',
20     ];
21
22     /**
23      * The attributes that should be hidden for arrays.
24      *
25      * @var array
26      */
27     protected $hidden = [
28         'password', 'remember_token',
29     ];
30 }
```

Bukalah file web.php pada folder routes/web.php, kemudian ubahlah code pada baris ke 18-20 menjadi

Auth::routes(['verify' => true]);

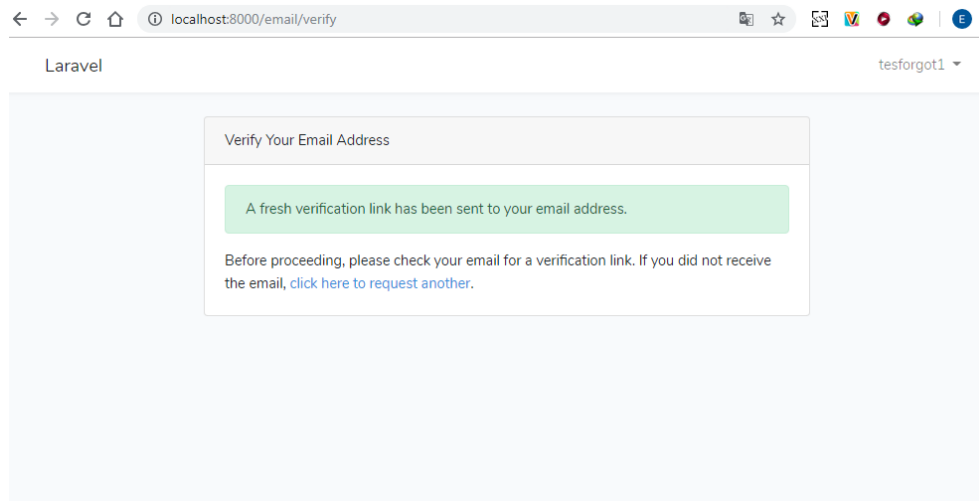
Route::get('/home', 'HomeController@index')->name('home')->middleware('verified');



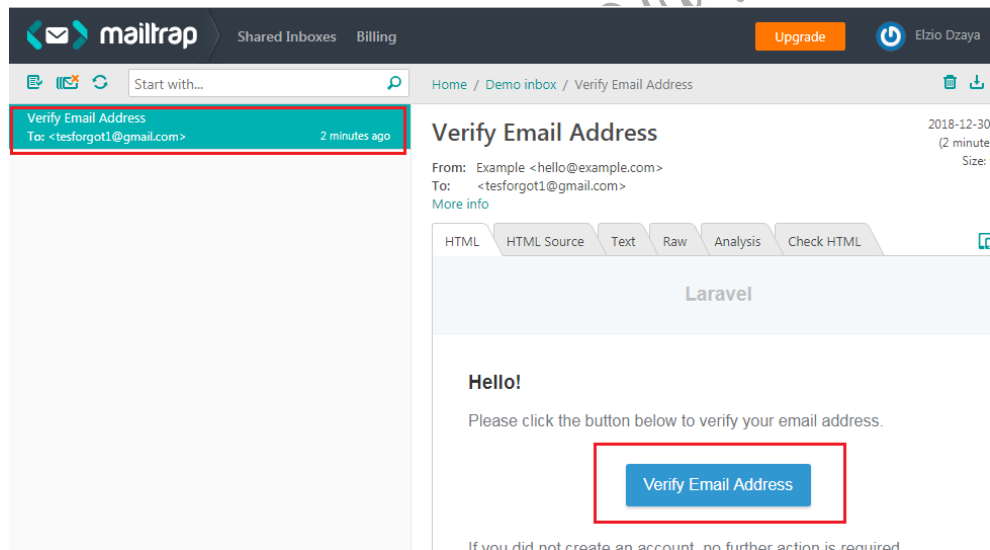
```
7 |
8 | Here is where you can register web routes for your application. These
9 | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 | */
13 |
14 Route::get('/', function () {
15     return view('welcome');
16 });
17 |
18 Auth::routes(['verify' => true]);
19 |
20 Route::get('/home', 'HomeController@index')->name('home')->middleware('verified');
21 |
```

Kemudian simpanlah semua perubahan pada file .Env, User.php dan web.php

setelah itu jalankan server dan bukalah aplikasi anda, silahkan register akun baru dan lakukan verifikasi email akun tersebut



Lakukanlah verifikasi email melalui mailtrap.io



Seteah anda mengklik tombol verify email address, maka anda akan dapat login pada aplikasi tersebut

Silahkan anda logout terlebih dahulu, kemudian masuk kedalam halaman login, setelah itu gunakanlah *fitur forgot password*, maka anda akan masuk kedalam halaman *reset password*, masukan *alamat email dari akun* yang baru anda buat setelah itu *klik tombol send password reset*

Reset Password

We have e-mailed your password reset link!

E-Mail Address

[Send Password Reset Link](#)

Jika berhasil maka pada web mailtrap, akan terdapat notifikasi reset password seperti gambar dibawah ini:

mailtrap Shared Inboxes Billing Upgrade Elzio Dzay

Start with...

Home / Demo inbox / Reset Password Notifica...

Reset Password Notification
To: <tesforgot@gmail.com> a few seconds ago

Verify Email Address
To: <tesforgot@gmail.com> 2 minutes ago

Verify Email Address
To: <tesforgot@gmail.com> 2 minutes ago

Reset Password Notification
From: Example <hello@example.com>
To: <tesforgot@gmail.com>
More info

HTML HTML Source Text Raw Analysis Check HTML

Laravel

Hello!

You are receiving this email because we received a password reset request for your account.

[Reset Password](#) ✓

If you did not request a password reset, no further action is required.

Ketika anda **klik tombol reset password**, maka anda akan diarahkan kedalam halaman reset password, untuk kemudian **mengisi form reset password**

Reset Password

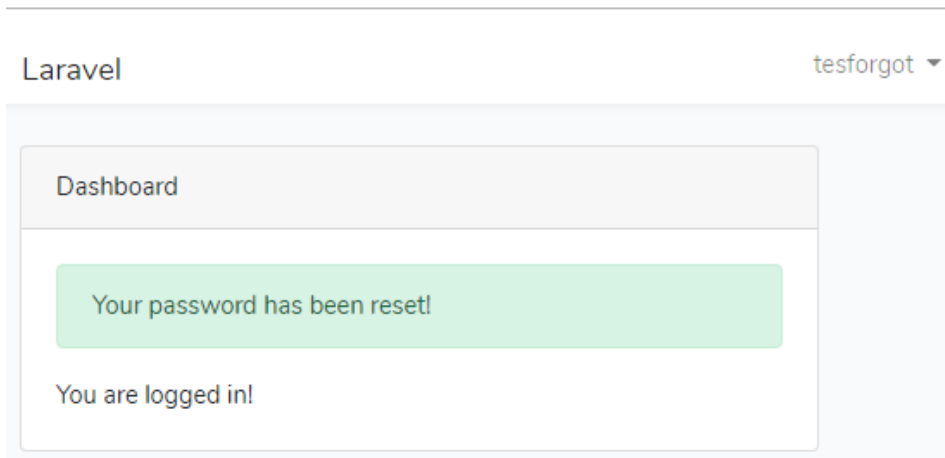
E-Mail Address

Password

Confirm Password

[Reset Password](#)

Jika berhasil maka akan muncul konfirmasi pada halaman tersebut seperti pada gambar dibawah ini:



3. Mencegah Akses Halaman Tanpa Login

Fungsi ini digunakan agar ketika user langsung mengakses halaman melalui URL, akan dialihkan ke halaman login terlebih dahulu

Pada *file web.php*, ubahlah source code seperti dibawah ini

```
1  <?php
2
3  /*
4  |-----
5  | Web Routes
6  |-----
7  |
8  | Here is where you can register web routes for your application. These
9  | routes are loaded by the RouteServiceProvider within a group which
10 | contains the "web" middleware group. Now create something great!
11 |
12 */
13
14 Route::get('/', function () {
15     return view('auth/login');
16 });
17
18 Auth::routes(['verify' => true]);
19 Route::group(['middleware' => 'auth'], function () {
20     Route::get('/home', 'HomeController@index')->name('home')->middleware('verified');
21     Route::resource('mhs', 'MahasiswaController');
22 });
23
24
```

Perintah pada line 15 berfungsi agar halaman yang pertama kali ditampilkan pada *halaman web adalah halaman login*

Perintah **route group** pada **line 20** berfungsi untuk melakukan verifikasi ketika akan mengakses halaman yang terdapat didalam **route group** tersebut

Bukalah file **app.blade.php** yang terdapat didalam **folder resources/views/layouts**, kemudian tambahkan sourcecode, untuk membuat menu data, sehingga user yang berhasil login dapat mengakses data mahasiswa melalui menubar yang tersedia

```
.env 2019_05_15_182332_create_mahasiswas_table.php AppServiceProvider.php web.php app.blade.php index.blade.php
46 @if (Route::has('register'))
47     <li class="nav-item">
48         <a class="nav-link" href="{{ route('register') }}">{{ __('Register') }}</a>
49     </li>
50 @endif
51 @else
52     <li class="nav-item">
53         <a class="nav-link" href="{{ route('mhs.index') }}">Data</a>
54     </li>
55     <li class="nav-item dropdown">
56         <a id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-toggle="dropdown">
57             {{ Auth::user()->name }} <span class="caret"></span>
58         </a>
```

Silahkan anda refresh browser anda dan silahkan anda coba akses melalui url: <http://127.0.0.1:8000/mhs>, jika berhasil maka anda akan diarahkan ke halaman login, jika anda berhasil login maka tampilan akan seperti gambar dibawah ini

