

TUTORIAL APLIKASI CRUD SEDERHANA

Dengan laravel 5.5 dan MongoDB



Di susun oleh
Tri Febriani Pamungkas
trifeby15@gmail.com

SMK Negeri 10 Jakarta

Jl. Mayjen Soetoyo, Cawang, Kramatjati, RT.2/RW.9, Cawang, Kramatjati, Kota
Jakarta Timur, Daerah Khusus Ibukota Jakarta 13630

Daftar Isi

PENGERTIAN CRUD

PENGERTIAN LARAVEL

PENGERTIAN MONGO DB

TUTORIAL :

-Cara Install MongoDB di Ubuntu 16.04

-Cara menghubungkan Laravel ke database MongoDB

Langkah 1 :

buat Project Laravel baru

Langkah 2 :

Konfigurasi Database MongoDB

Langkah 3:

Instal Paket Laravel MongoDB

Langkah 4:

Define Providers dan akses Server MongoDB

Langkah 5:

Membuat Model dan Migration

Langkah 6:

Membuat Controller dan Auth

Langkah 7 :

Membuat View untuk CRUD di Laravel

Langkah 8 :

Membuat Route

PENGERTIAN CRUD

CRUD adalah akronim untuk Create, Read, Update, dan Delete. Operasi CRUD adalah manipulasi data dasar untuk database. Sering digunakan pada aplikasi-aplikasi pengolahan data yang kebanyakan menggunakan fungsi CRUD didalamnya . Fungsi ini digunakan untuk menambahkan data, menghapus data, serta mengupdate data.

- Create memasukkan atau membuat (row) sebuah data baru ke dalam database,
- Read membaca atau menampilkan suatu data yang tadinya berada didatabase,
- Update mengubah atau mengedit sebuah data dari database yang kemudian di edit menggunakan bahasa pemrograman Php berupa WEB,,
- Delete Menghapus data dari database.

PENGERTIAN LARAVEL

Laravel adalah kerangka kerja aplikasi web berbasis PHP yang open source, menggunakan konsep model–view–controller (MVC). Laravel berada dibawah lisensi MIT, dengan menggunakan GitHub sebagai tempat berbagi kode.

Desember 2013, Laravel menempati PHP framework terpopuler dan berada diatas PHP framework lain seperti Phalcon, Symfony2, CodeIgniter dan lainnya.

PENGERTIAN MONGO DB

MongoDB adalah perangkat lunak document-oriented database dengan lisensi open source. MongoDB masuk dalam kategori database NoSQL. Database NoSQL banyak digunakan dalam big-data dan aplikasi real-time.

Pertama kali dikembangkan oleh perusahaan asal New York City, 10gen (sekarang MongoDB Inc.) pada bulan Oktober 2007 sebagai bagian dari platform yang direncanakan sebagai produk jasa, perusahaan bergeser ke model pembangunan sumber terbuka pada tahun 2009, dengan 10gen menawarkan dukungan komersial dan layanan lainnya. Sejak itu, MongoDB telah diadopsi sebagai perangkat lunak backend oleh sejumlah situs dan layanan, termasuk Craigslist, eBay, Foursquare, SourceForge, dan The New York Times. MongoDB adalah sistem basis data NoSQL yang paling populer.

TUTORIAL

Cara Install MongoDB di Ubuntu 16.04

1. Import public key

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80  
--recv 2930ADAE8CAF5059EE73BB4B58712A2291FA4AD5
```

2. Buat file repository

```
echo "deb [ arch=amd64,arm64 ]  
https://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.6  
multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-  
3.6.list
```

3. Update repository

```
sudo apt-get update
```

4. Install MongoDB

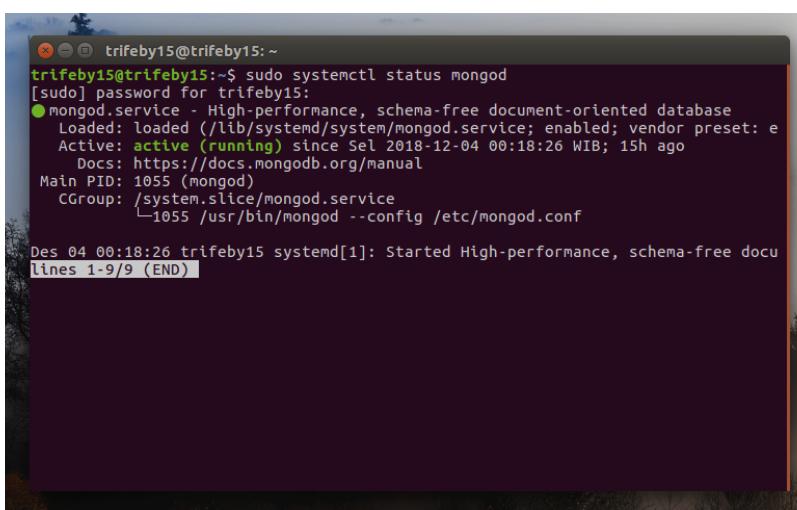
```
sudo apt-get install mongodb-org
```

5. Jalankan service mongod

```
sudo systemctl start mongod
```

6. Cek status service

```
sudo systemctl status mongod
```



```
trifeby15@trifeby15:~$ sudo systemctl status mongod  
[sudo] password for trifeby15:  
● mongod.service - High-performance, schema-free document-oriented database  
  Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset: e  
  Active: active (running) since Sel 2018-12-04 00:18:26 WIB; 15h ago  
    Docs: https://docs.mongodb.org/manual  
    Main PID: 1055 (mongod)  
      CGroup: /system.slice/mongod.service  
              └─1055 /usr/bin/mongod --config /etc/mongod.conf  
  
Des 04 00:18:26 trifeby15 systemd[1]: Started High-performance, schema-free docu  
lines 1-9/9 (END)
```

7. Auto-start service pada saat boot

```
sudo systemctl enable mongod
```

Cara menghubungkan Laravel ke database MongoDB

Konfigurasi Database MongoDB dalam Aplikasi Laravel

Masuk ke direktori

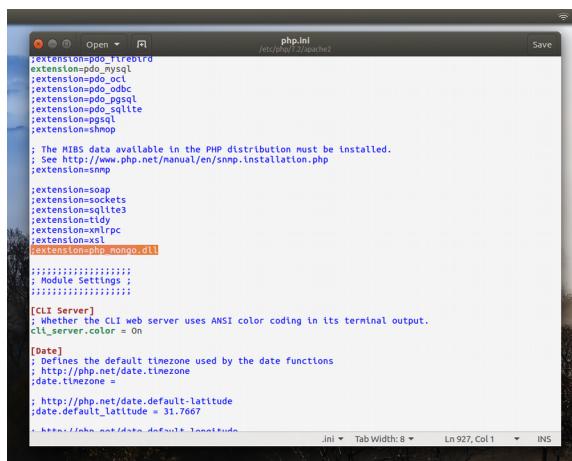
```
cd /etc/php/7.2/apache2
```

ketikan

```
sudo gedit php.ini
```

setelah terbuka file php.ini, find kata .dll dan masukan baris berikut

```
extension=php_mongodb.dll
```



save, dan restart server dengan mengetikan

```
sudo service apache2 restart
```

selanjutnya ketik

```
sudo apt-get install php-mongodb
```

Langkah 1 :

buat Project Laravel baru dengan mengetik perintah berikut.

```
composer create-project --prefer-dist laravel/laravel bihero
```

Langkah 2 :

Konfigurasi Database MongoDB

buka file .env

```
//.env
```

```
MONGO_DB_HOST=127.0.0.1
MONGO_DB_PORT=27017
MONGO_DB_DATABASE=mongocrud
MONGO_DB_USERNAME=
MONGO_DB_PASSWORD=
```

Tambahkan koneksi mongodb baru pada file config>>database.php

```
//database.php
```

```
'connections' => [
    ...
    'mongodb' => [
        'driver'    => 'mongodb',
        'host'      => env('MONGO_DB_HOST', 'localhost'),
        'port'      => env('MONGO_DB_PORT', 27017),
        'database'  => env('MONGO_DB_DATABASE'),
        'username'  => env('MONGO_DB_USERNAME'),
        'password'  => env('MONGO_DB_PASSWORD'),
        'options'   => []
    ],
]
```

Langkah 3: Instal Paket Laravel MongoDB

Instal paket jenssegers/mongodb, ketikan direktori project

```
composer require jenssegers/mongodb
```

Langkah 4: Define Providers dan akses Server MongoDB

temukan providers difile config>>app.php dan daftarkan MongodbServiceProvider.

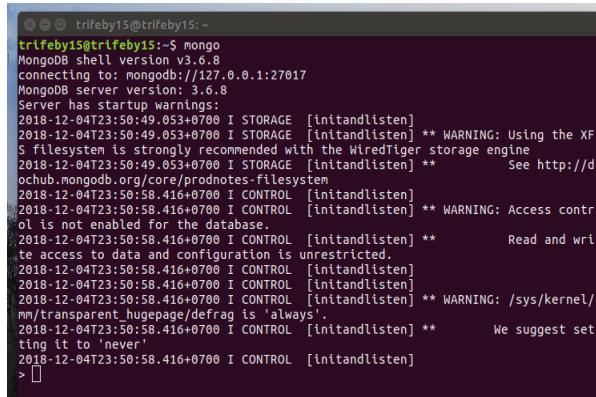
```
'providers' => [
    Jenssegers\Mongodb\MongodbServiceProvider::class,
]
```

Ketik perintah

```
mongo
```

Perintah mongo adalah program *client* untuk membuka *Shell* MongoDB.

Kemudian akan muncul



```
trifeby15@trifeby15:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
MongoDB server version: 3.6.8
Server has startup warnings:
2018-12-04T23:50:49.053+0700 I STORAGE  [initandlisten]
2018-12-04T23:50:49.053+0700 I STORAGE  [initandlisten] ** WARNING: Using the XFS
filesystem is strongly recommended with the WiredTiger storage engine
2018-12-04T23:50:49.053+0700 I STORAGE  [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten]
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten] ** WARNING: Access contr
ol is not enabled for the database.
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten] ** Read and wri
te access to data and configuration is unrestricted.
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten]
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten]
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/
mm/transparent_hugepage/defrag is 'always'.
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten] ** We suggest set
ting it to 'never'
2018-12-04T23:50:58.416+0700 I CONTROL  [initandlisten]
> □
```

setelah itu ketikan perintah

```
use bihero
```

perintah tersebut akan membuat database baru bernama bihero.

Langkah 5: Membuat Model dan Migration

Ketik perintah berikut di terminal

```
php artisan make:model Bihero -m
```

Kemudian masuk ke dalam file Bihero.php dan tambahkan kode berikut

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;
use Jenssegers\Mongodb\Eloquent\Model as Eloquent;

class Bihero extends Eloquent
{
    protected $connection = 'mongodb';
    protected $collection = 'biheros';

    protected $fillable =
        [
            'image', 'nama', 'asal', 'lahirwafat', 'penetapan',
            'ringkasan', 'penjelasan', 'created_at'
        ];
}
```

setelah itu masuk ke dalam file migrations dan pada setiap file didalamnya ubahlah

```
use Illuminate\Database\Schema\Blueprint;
```

menjadi

```
use Jenssegers\Mongodb\Schema\Blueprint;
```

selanjutnya ke file

migrations>>2018_11_06_084200_create_biheroes_table.php masukan data table sama seperti di model.

```
2018_11_06_084200_create_biheroes_table.php
1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Jenssegers\Mongodb\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateBiheroesTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('biheroes', function (Blueprint $table)
17         {
18             $table->increments('_id');
19             $table->string('image');
20             $table->string('nama');
21             $table->string('asal');
22             $table->string('lahirwafat');
23             $table->string('penetapan');
24             $table->string('ringkasan');
25             $table->text('penjelasan');
26             $table->timestamps();
27         });
28     }
29
30     /**
31      * Reverse the migrations.
32      */
33 }
```

Pada function down()

ubahlah

```
public function down()
```

```
{
```

```
    Schema::dropIfExists('stuff1s');
```

```
}
```

menjadi

```
public function down()
```

```
{
```

```
    Schema::table('biheroes', function (Blueprint  
$collection) {
```

```
        $collection->drop();
```

```
    });
```

```
}
```

hal ini dilakukan agar dapat migration data ke MongoDB.

Kemudian migrasikan dengan mengetik perintah berikut.

```
php artisan migrate:refresh
```

Langkah 6: Membuat Controller dan Auth

Ketik perintah berikut di terminal

```
php artisan make:controller BiheroController
```

jika ingin membuat controller dengan tambahan fungsi untuk CRUD secara otomatis bisa ditambahkan -r setelah penulisan nama controller.

Masukan koding berikut untuk membuat CRUD-nya.

```
<?php

namespace App\Http\Controllers;

use \App\Bihero;
use PDF;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class BiheroController extends Controller
{
    public function index()
    {
        $data['bihero'] = \App\Bihero::All();
        return view('bihero.index', $data);
    }

    public function review($id)
    {
        $data['bihero'] = Bihero::where('_id', '=', $id)->get();
        return view('bihero.review', $data);
    }

    public function create()
    {
        $data['bihero'] = \App\Bihero::All();
        return view('bihero.create');
    }

    public function table()
    {
        $data['bihero'] = \App\Bihero::All();
        return view('bihero.table')->with($data);
    }

    public function save(Request $r)
    {
        $file = $r->file('image');
        $gambar = $file->getClientOriginalName();
        $r->file('image')->move(public_path('upload'), $gambar);
        $bihero = new Bihero;
        $bihero->image = $gambar;
        $bihero->nama = $r->input('nama');
        $bihero->asal = $r->input('asal');
        $bihero->lahirwafat = $r->input('lahirwafat');
        $bihero->penetapan = $r->input('penetapan');
        $bihero->ringkasan = $r->input('ringkasan');
        $bihero->penjelasan = $r->input('penjelasan');

        $bihero->save();
    }
}
```

```

        return redirect(url('bihero'));
    }

    public function edit($id)
    {
        $data['bihero'] = \App\Bihero::find($id);
        return view('bihero.edit')->with($data);
    }

    public function update(Request $r)
    {
        $edit = Bihero::find($r->input('id'));

        $edit->nama = $r->input('nama');
        $edit->asal = $r->input('asal');
        $edit->lahirwafat = $r->input('lahirwafat');
        $edit->penetapan = $r->input('penetapan');
        $edit->ringkasan = $r->input('ringkasan');
        $edit->penjelasan = $r->input('penjelasan');

        $edit->save();

        return redirect(url('bihero'));
    }

    public function delete(Request $r)
    {
        $delete = Bihero::where('_id', $r->iditem)->delete();

        return redirect(url('bihero'));
    }

    public function readpdf()
    {

        $bihero['bihero'] = \App\Bihero::All();
        $pdf = PDF::loadview('bihero.report',$bihero);
        return $pdf->stream();
    }
}

```

Ketik perintah berikut di terminal

```
php artisan make:auth
```

perintah tersebut digunakan untuk membuat login dan dapat juga menjadi user role di laravel .

Langkah 7 : Membuat View untuk CRUD di Laravel

Untuk menyambungkan bootstrap dengan project download terlebih dahulu file di <https://getbootstrap.com/> setelah terpasang klik tombol Get started samping tombol download kemudian ikuti instruksi tersebut.

Sebelum membuat tampilan pada setiap form, kita terlebih dahulu membuat file master pada direktori resources\views\layouts untuk tampilannya ada master.blade.php, header.blade.php, footer.blade.php. Isi file tersebut adalah koding dari template bootstrap yang kita pasang. Untuk templatnya bisa didownload di

<https://startbootstrap.com/template-categories/landing-pages/>

Buat folder baru pada direktori
resources\views\
yaitu bihero, kemudian buatlah file untuk layout CRUD-nya terdiri dari 6 file.

1. create.blade.php

```
@extends('layouts.master')

@section('content')

<body style="background-color: #f9fbe7">

<div class="container" style="margin-bottom:3%;>

    <form method="post" action="{{url('bihero/save')}}" enctype="multipart/form-data"
style="padding-top: 30px; margin-bottom: 3%;">

        {{csrf_field()}}

        <div class="form-row">
            <div class="form-group col-md-6">
                <label for="nama">Nama</label>
                <input type="text" class="form-control" name="nama" placeholder="nama">
            </div>

            <div class="form-group col-md-6">
                <label class="control-label">Foto</label>
```

```
<div class="custom-file">
    <input type="file" class="custom-file-input" name="image">
        <label class="custom-file-label" for="validatedCustomFile">pilih
        gambar...</label>
    </div>
</div>

<div class="form-group">
    <label for="asal">Asal</label>
    <input type="text" class="form-control" name="asal" placeholder="asal">
</div>

<div class="form-group">
    <label for="lahirwafat">Lahirwafat</label>
    <input type="text" class="form-control" name="lahirwafat"
placeholder="lahirwafat">
</div>

<div class="form-row">
    <div class="form-group col-md-6">
        <label for="penetapan">Penetapan</label>
        <input type="text" class="form-control" name="penetapan"
placeholder="penetapan">
    </div>
    <div class="form-group col-md-6">
        <label for="ringkasan">Ringkasan</label>
        <input type="text" class="form-control" name="ringkasan"
placeholder="ringkasan">
    </div>
</div>

<div class="form-group">
```

```

<label for="penjelasan">Penjelasan</label>

<textarea class="form-control" rows="3" type="text" class="form-control"
name="penjelasan"></textarea>

</div>

<div class="text-right" style="margin-top: 2%;>
    <button type="submit" class="btn btn-info">POST</button>
</div>

</form>
</div>
</body>
@endsection

```

2. index.blade.php

```

@extends('layouts.master')
@section('content')
<body style="background-color: #f9fbe7">
<div class="container">
<section class="content">
<div class="row">
    <div class="col-lg-12">
        <h2 class="my-5"> </h2>
    </div>
    <div class="row">
        @foreach($bihero as $key)
            <div class="col-lg-4 col-sm-6 text-center mb-4">
                <a href="{{url('bihero/review', [$key->id])}}>
                    
<div class="container" style="margin-top:2%;">
<form method="post" action="{{url('bihero/update')}}">
{{csrf_field()}}
<input class="form-control" name="id" type="hidden"
value="{{\$bihero->id}}">
<div class="form-row">
<div class="form-group col-md-6">
<label for="nama">Nama</label>
<input type="text" class="form-control"
name="nama" value="{{\$bihero->nama}}">
</div>
<div class="form-group col-md-6">
<label class="control-label">Foto</label>
<div class="custom-file">

```

```
        <input type="file" class="custom-file-input"
name="image">
            <label class="custom-file-label"
for="validatedCustomFile">pilih gambar...</label>
        </div>
    </div>
</div>
<div class="form-group">
    <label for="asal">Asal</label>
    <input type="text" class="form-control"
name="asal" value="{{ $bihero->asal }}>
</div>
<div class="form-group">
    <label for="lahirwafat">Lahirwafat</label>
    <input type="text" class="form-control"
name="lahirwafat" value="{{ $bihero->lahirwafat }}>
</div>
<div class="form-row">
    <div class="form-group col-md-6">
        <label for="penetapan">Penetapan</label>
        <input type="text" class="form-control"
name="penetapan" value="{{ $bihero->penetapan }}>
    </div>
    <div class="form-group col-md-6">
        <label for="ringkasan">Ringkasan</label>
        <input type="text" class="form-control"
name="ringkasan" value="{{ $bihero->ringkasan }}>
    </div>
</div>
<div class="form-group">
    <label for="penjelasan">Penjelasan</label>
    <textarea class="form-control" rows="3"
type="text" class="form-control" name="penjelasan">{{ $bihero->penjelasan }}</textarea>
```

```

        </div>
        <div class="text-right" style="margin-top: 2%;>
            <button type="submit" class="btn btn-primary">UPDATE</button>
        </div>
    </form>
</div>
</body>
@endsection

```

4. review.blade.php

```

@extends('layouts.master')
@section('content')
<body style="background: #f9fbe7">
<div class="container">
    <div class="row">
        <div class="col-lg-12">
            @if(isset($bihero))
                @foreach($bihero as $key)
                    <h1 class="mt-4">{{$key->nama}}</h1>
                    <hr>
                    <p>Posted on {{$key->created_at}}</p>
                    <hr>
                    
                    <hr>
                <div class="row">
                    <div class="col-12">
                        <p align="justify" class="text-capitalize" style="line-height: 1.5;">{{$key->penjelasan}}</p>
                    </div>
                </div>
            @endif
        </div>
    </div>
</div>

```

```
<hr>
@endforeach
@endif
</body>
<hr>
```

5. **table.blade.php**

```
@extends('layouts.master')

@section('content')
<body style="background: #f9fbe7">
<style type="text/css">
    .tabel {
        margin-right: 1%;
        margin-left: 1%;
    }
</style>
<div class="table" style="padding-top: 30px; margin-bottom: 3%;>
    <div class="row">
        <div class="col-lg-12">
            <a href="{{route('readpdf')}}"><button class="btn btn-success" class="text-right" style="margin-bottom: 1%; color: #fff">CETAK</button>
            </a>
            <table class="table table-hover">
                <thead style="background-color: #fff8e1 ; color: black;">
                    <tr>
                        <th><b>Tanggal Posting</b></th>
                        <th><b>Nama</b></th>
                        <th><b>Asal</b></th>
                        <th><b>Lahir Wafat</b></th>
```

```

<th><b>Penetapan</b></th>
<th>Ringkasan</th>
<th style="text-align: center;"><b>Aksi</b></th>
</tr>
</thead>
<tbody>
@foreach($bihero as $key)
<tr>
<td>{{$key->created_at}}</td>
<td>{{$key->nama}}</td>
<td>{{$key->asal}}</td>
<td>{{$key->lahirwafat}}</td>
<td>{{$key->penetapan}}</td>
<td>{{$key->ringkasana}}</td>
<td>
<center>
<form action="{{route('delete')}}" method="POST" onclick="return confirm('Hapus Data ?')">
{{csrf_field()}}
<input type="hidden" name="iditem" value="{{$key->id}}>
<button type="submit" class="btn btn-danger">hapus</button>
</form>
<a href="{{route('edit', [$key->id])}}><button type="submit" class="btn btn-warning" style="color: #fff; margin-top: 2%; ">edit</button>
</a>
</center>
</td>
</tr>
@endforeach
</tbody>

```

```
</table>
</div>
</div>
</div>
</body>
@endsection
```

6. report.blade.php

```
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8"/>
        <title>Cetak</title>
    <body>
        <style type="text/css">
            .tg {border-collapse:collapse; border-
spacing:0; border-color:#ccc; width: 100%; }

            .tg td{font-family:Arial;font-
size:12px;padding:10px 5px; border-style:solid; border-
width:1px; overflow:hidden; word-break: normal; border-
color:#ccc;color:#333;background-color:#fff; }

            .tg th{font-family:Arial;font-size:14px;font-
weight: normal; padding:10px 5px; border-style:solid; border-
width:1px; overflow:hidden; word-break: normal; border-
color:#ccc;color:#333;background-color:#f0f0f0; }

            .tg .tg-3wr7{font-weight:bold;font-size:12px;font-
family:"Arial", Helvetica, sans-serif !important;;text-
align:center}

            .tg .tg-ti5e{font-size:10px;font-family:"Arial",
Helvetica, sans-serif !important;;text-align:center}

            .tg .tg-rv4w{font-size:10px;font-family:"Arial",
Helvetica, sans-serif !important; }

        </style>
```

```

<div style="font-family:Arial; font-size:12px;">
    <center><b><h2>DATA PAHLAWAN</b></h2></center>
</div>
<br>
<table class="tg">
    <tr>
        <th class="tg-3wr7">Tanggal Posting<br></th>
        <th class="tg-3wr7">Nama<br></th>
        <th class="tg-3wr7">Asal<br></th>
        <th class="tg-3wr7">Lahir Wafat<br></th>
        <th class="tg-3wr7">Penetapan<br></th>
        <th class="tg-3wr7">Ringkasan<br></th>
    </tr>
    @foreach ($bihero as $key)
        <tr>
            <td class="tg-rv4w" width="20%">{{ $key->created_at }}</td>
            <td class="tg-rv4w" width="20%">{{ $key->nama }}</td>
            <td class="tg-rv4w" width="20%">{{ $key->asal }}</td>
            <td class="tg-rv4w" width="20%">{{ $key->lahirwafat }}</td>
            <td class="tg-rv4w" width="10%">{{ $key->penetapan }}</td>
            <td class="tg-rv4w" width="20%">{{ $key->ringkasan }}</td>
        </tr>
    @endforeach
</table>
</body>
</head>
</html>

```

Langkah 8 : Membuat Route

Route adalah fitur yang digunakan untuk mendaftarkan semua URL yang bisa diakses oleh pengguna aplikasi berdasarkan respon dari *HTTP* verb. Berikut penulisan routes.

```
Route::get('/', function () {
    return view('welcome');
});

Auth::routes();

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

Route::get('/bihero', 'BiheroController@index');

Route::get('/bihero/review/{id}', 'BiheroController@review');

Route::get('/bihero/create', 'BiheroController@create')-
>name('create');

Route::get('/bihero/table', 'BiheroController@table')-
>name('table');

Route::post('/bihero/save', 'BiheroController@save');

Route::get('/bihero/edit/{id}', 'BiheroController@edit')-
>name('edit');

Route::post('/bihero/update', 'BiheroController@update');

Route::post('/bihero/delete/action', 'BiheroController@delete')-
>name('delete');

Route::get('/bihero/pdf', 'BiheroController@readpdf')-
>name('readpdf');
```

Langkah 8 : Jalankan Project

Setelah sudah megikuti hingga langkah ke 7 tiba saat untuk mencoba jalankan project dengan masuk ke direktori project dan mengetikan perintah pada terminal

```
php artisan serve
```

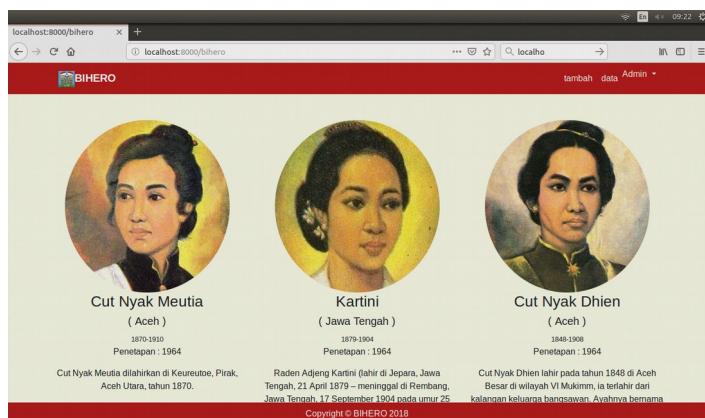
tunggu hingga ada bacaan seperti ini

```
trifeby15@trifeby15: /var/www/bihero
trifeby15@trifeby15:~$ cd /var/www/bihero
trifeby15@trifeby15:/var/www/bihero$ php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
```

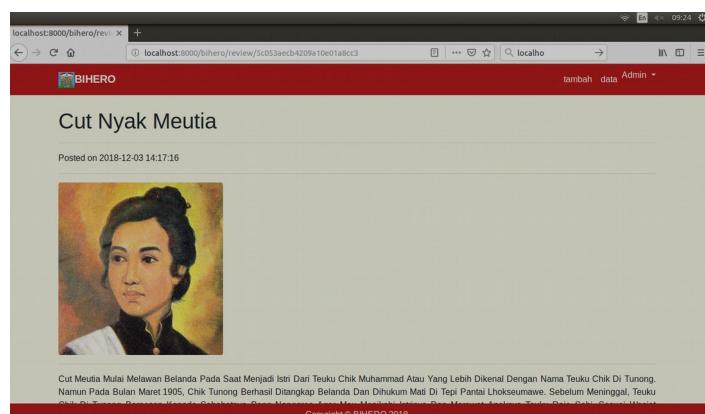
jika sudah ada kemudian buka browser dan ketikan

localhost:8000

selanjutnya silahkan register atau jika kalian telah membuat role silahkan login untuk role admin dan akan tampak tampilan-tampilannya seperti berikut (sudah ditambahkan sebelumnya) :



(index.blade.php)



(review.blade.php)

localhost:8000/bihero/create

BIHERO

tambah data Admin

Nama **Foto**

Asal

Lahirwafat

Penetapan **Ringkasan**

Penjelasan

POST

Copyright © BIHERO 2018

(create.blade.php)

localhost:8000/bihero/table

CETAK

Tanggal Posting **Nama** **Asal** **Lahir Wafat** **Penetapan** **Ringkasan** **Aksi**

2018-12-03 14:17:16	Cut Nyak Meutia	Aceh	1870-1910	1964	Cut Nyak Meutia dilahirkan di Keureuteo, Pirak, Aceh Utara, tahun 1870.	<input type="button" value="hapus"/> <input type="button" value="edit"/>
2018-12-03 14:19:45	Kartini	Jawa Tengah	1879-1904	1964	Raden Adjeng Kartini (lahir di Jepara, Jawa Tengah, 21 April 1879 – meninggal di Rembang, Jawa Tengah, 17 September 1904 pada umur 25 tahun).	<input type="button" value="hapus"/> <input type="button" value="edit"/>
2018-12-06 02:21:31	Cut Nyak Dhien	Aceh	1848-1908	1964	Cut Nyak Dhien lahir pada tahun 1848 di Aceh Besar di wilayah VI Mukimm, ia terlahir dari kalangan keluarga bangsawan. Ayahnya bernama Teuku Nanta Seutia, seorang uleebalang, yang juga mempunyai keturunan dari Datuk Makhudum Sati.	<input type="button" value="hapus"/> <input type="button" value="edit"/>

Copyright © BIHERO 2018

(table.blade.php)

localhost:8000/bihero/edit

BIHERO

tambah data Admin

Nama **Foto**

Asal

Lahirwafat

Penetapan **Ringkasan**

Penjelasan

UPDATE

Copyright © BIHERO 2018

(edit.blade.php)

Daftar Pustaka

<https://www.codepolitan.com/tutorial-membuat-crud-php-dengan-mysql-59897c72d8470>

<https://id.wikipedia.org/wiki/MongoDB>

<https://id.wikipedia.org/wiki/Laravel>

<https://www.petanikode.com/tutorial-dasar-mongodb/>