**Laporan Proses Pengerjaan project Laravel : Tiket Pesawat**

Nama Anggota Kelompok 2 : 1. Bagus Kurniawan Hafizh

2. Azizah Senja Mega Sari

3. Gusti Aji Rizky I

4. Rafif Pradiyatma SW

Laravel adalah framework yang akan membantu Anda memaksimalkan penggunaan PHP dalam proses pengembangan website. Lengkapnya package yang dimiliki Laravel akan membuat PHP menjadi lebih powerful, cepat, aman, dan mudah digunakan.



Melalui artikel ini, kami akan menjelaskan bagaimana cara install Laravel di Windows. Sebelum melakukan instalasi Laravel pada Windows, pastikan Anda sudah menginstall beberapa aplikasi pendukung proses instalasi Laravel, yaitu:

* Aplikasi XAMPP
* Composer

**Cara Install Laravel Menggunakan Composer Windows**

Jika sudah menginstall XAMPP dan Composer, Anda bisa memulai proses instalasi Laravel di Windows. Berikut ini adalah cara install Laravel dengan Composer di Windows:

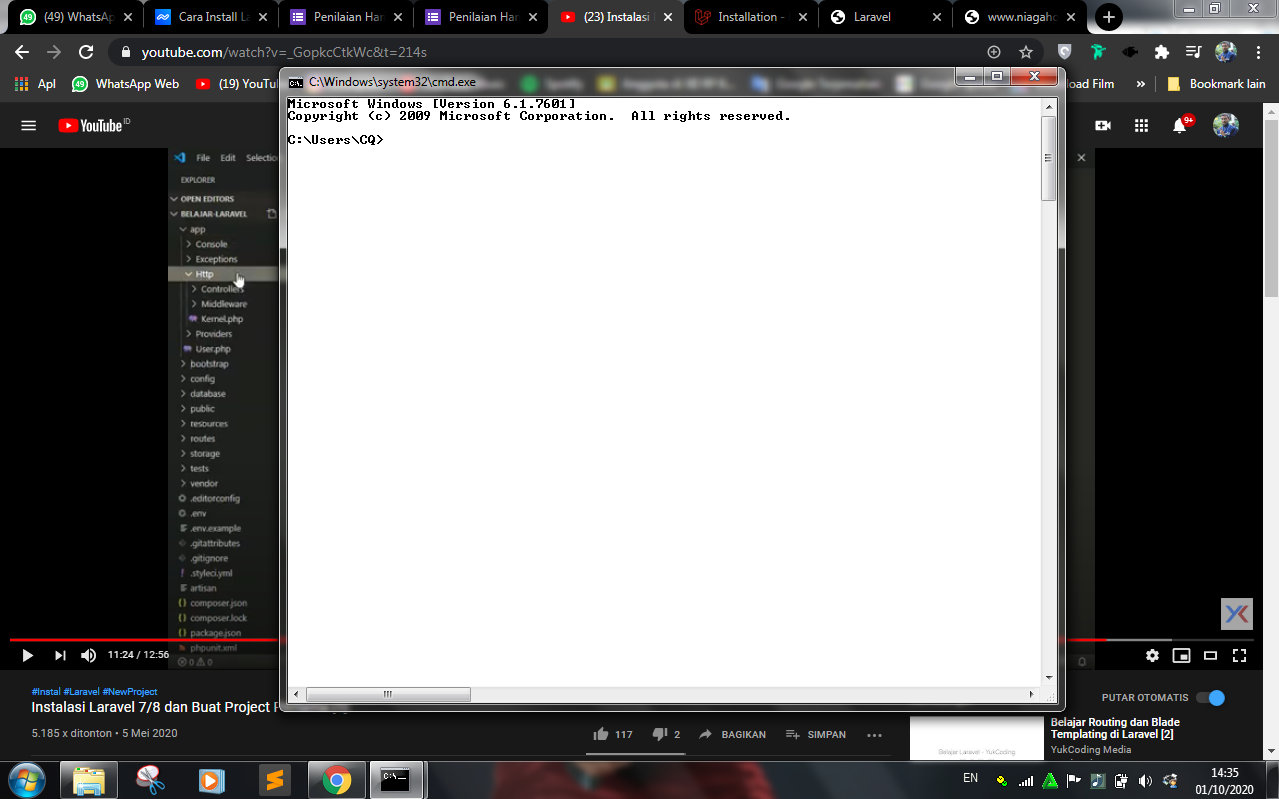
Cara install Laravel  terdiri dari empat langkah, yaitu:

1. Masuk Command Prompt
2. Masuk ke folder XAMPP
3. Mulai proses instalasi Laravel
4. Cek instalasi Laravel di web browser

Kami akan menjelaskan langkah-langkah tersebut satu per satu di bawah ini:

**1. Masuk Command Prompt**

Langkah pertama dalam install Laravel adalah masuk Command Prompt. Caranya adalah klik Win+R lalu ketik cmd dan klik **OK**



### ****2.**** ****Masuk ke Folder XAMPP****

Sebelum melakukan instalasi Laravel, arahkan Command Prompt atau terminal menuju direktori file server.  Lokasi file server pada XAMPP secara default berada pada direktori xampp/htdocs.  Masukan perintah ini pada jendela Command Prompt untuk masuk ke direktori htdocs.

cd \xampp\htdocs

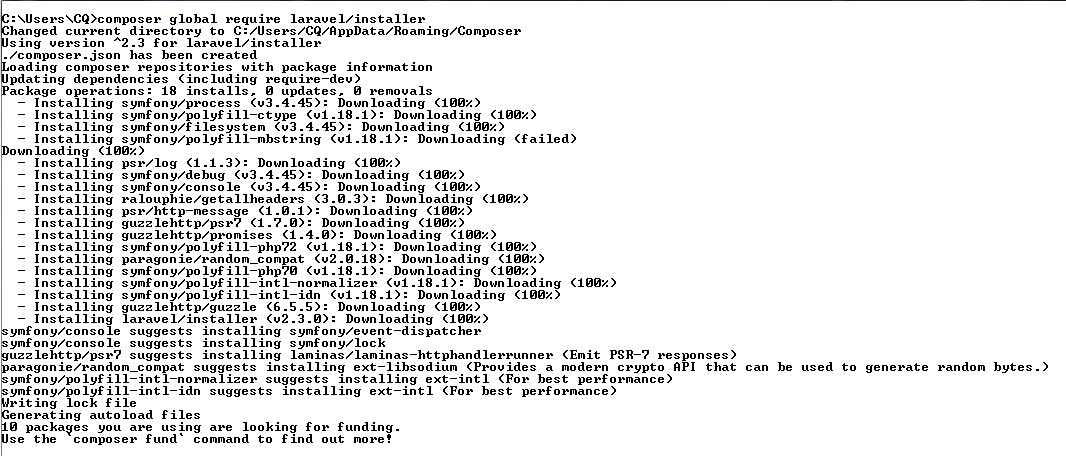


### ****3.**** ****Mulai Proses Install Laravel****

Selanjutnya jika sudah masuk direktori htdocs, Anda harus membuat request  untuk mengambil (serta menginstall) file Laravel yang telah disediakan dalam repositori Github. Gunakan perintah ini untuk melakukan request:

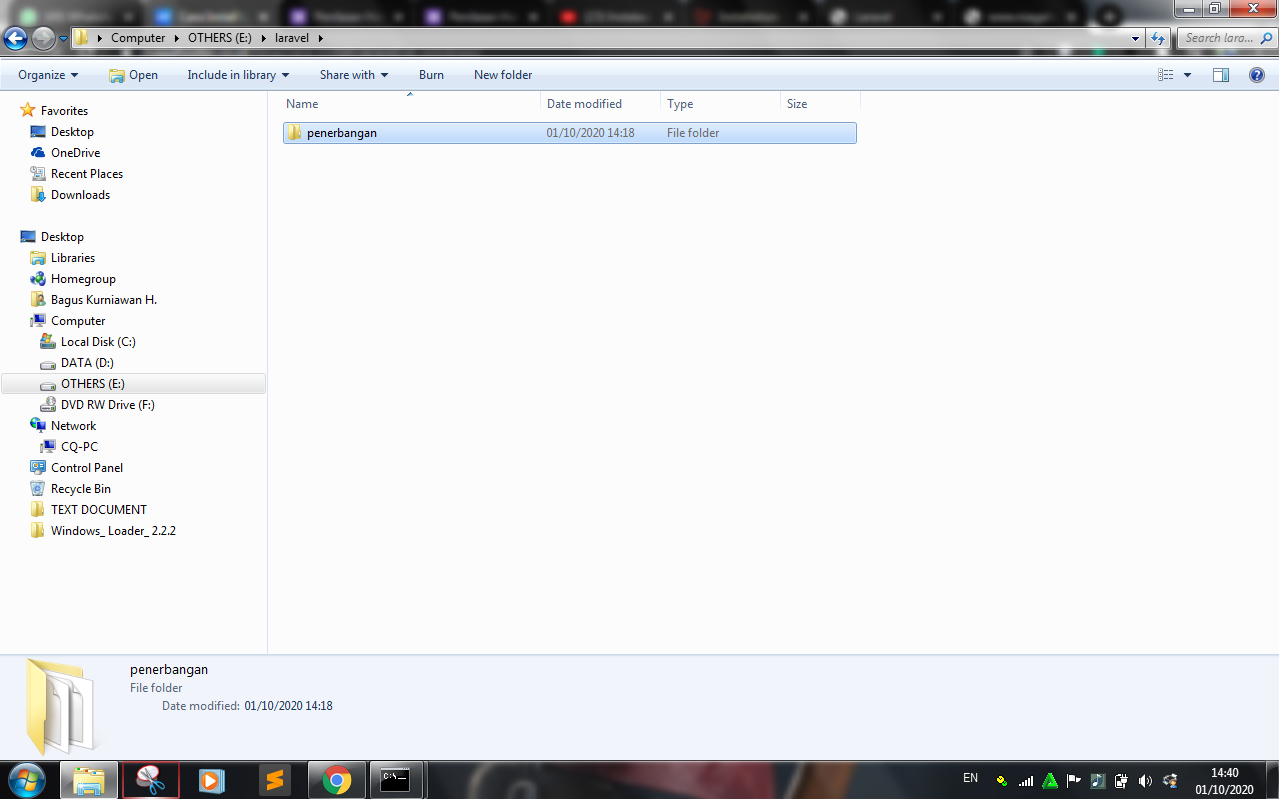
composer global require Laravel/installer

Jika perintah telah berhasil dimasukkan,  Composer akan mulai melakukan proses pengambilan data serta instalasi Laravel ke dalam direktori yang telah Anda tentukan. Pastikan bahwa koneksi internet dalam keadaan stabil agar tidak terjadi gangguan pada saat proses pengambilan data Laravel.



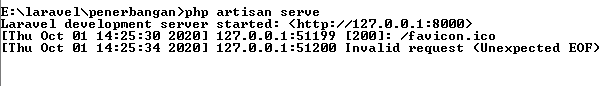
### ****4.**** ****Cek Instalasi Laravel di Web Browser****

Setelah proses download file Laravel selesai, nantinya akan ada folder baru pada direktori file server dengan nama sesuai nama project yang telah Anda tentukan sebelumnya pada folder /xampp/htdocs.

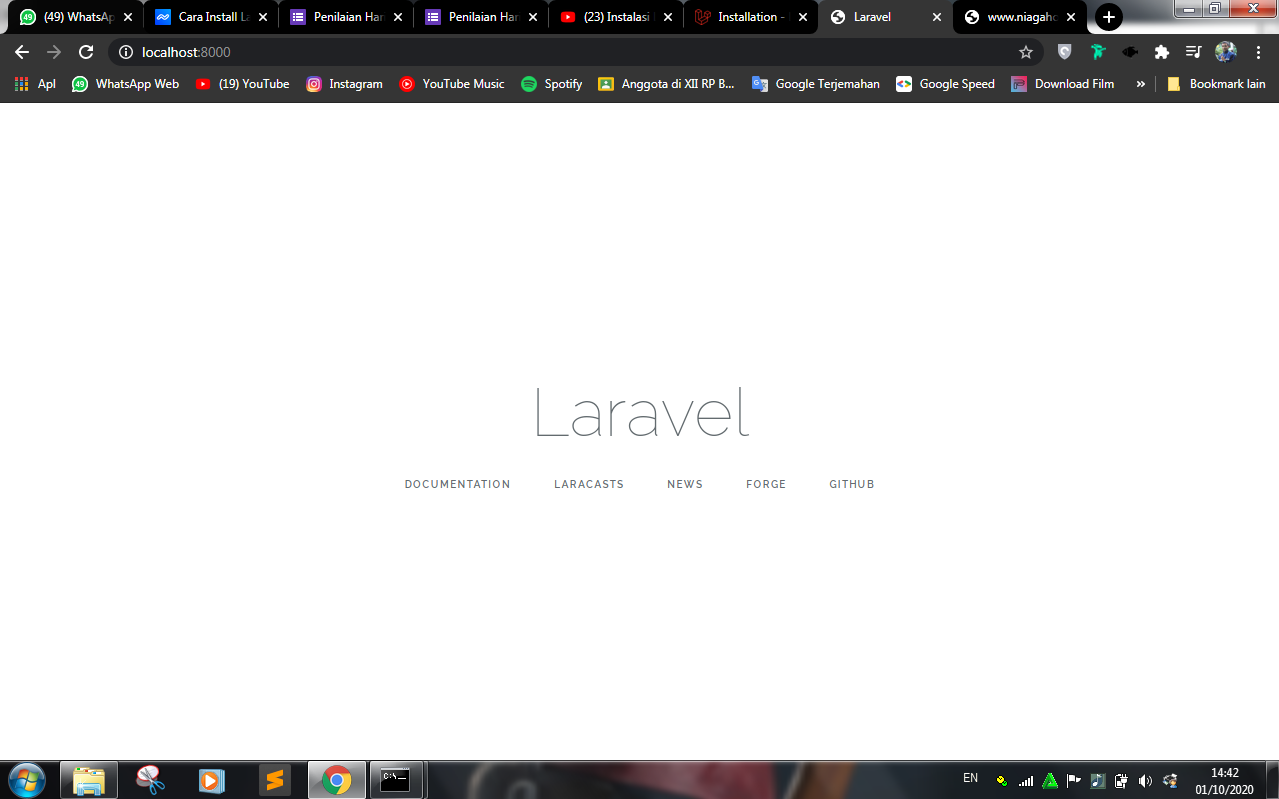


Untuk memastikan bahwa Laravel sukses terinstall dan siap untuk digunakan, arahkan Command Prompt atau Terminal menuju direktori yang telah Anda buat sebelumnya. Lalu, masukkan perintah berikut ke dalam Command Prompt atau Terminal:

Php artisan serve



Jika muncul tulisan Laravel development server started pada Command Prompt atau Terminal, langkah selanjutnya adalah membuka link yang telah disediakan oleh Laravel. Secara default, Anda akan diarahkan menuju alamat server,yaitu 127.0.0.1:8000. Nantinya, akan muncul  tampilan homepage dengan tulisan Laravel di bagian tengah seperti pada gambar di bawah ini:



Jika muncul seperti pada tampilan di atas, proses instalasi [**Laravel**](https://laravel.com/) Anda telah berhasil.

### Membuat koneksi PHP MySQL dengan PDO constructor

Karena **PDO** adalah fungsi/fitur PHP yang ditulis menggunakan objek, maka untuk membuat koneksi antara PHP dengan MySQL, kita menggunakan **PDO constructor**, yakni dengan membuat objek baru dari class PDO (menggunakan keyword: **new**).

Berikut adalah struktur dasar cara penulisan PDO constructor untuk database MySQL:

|  |  |
| --- | --- |
|  |  |

Variabel **$dbh** adalah variabel yang akan menjadi **object PDO**. Object ini biasa disebut sebagai **“Database Handler”**. Object inilah yang nantinya akan kita gunakan untuk menjalankan perintah-perintah PDO. Nama variabel boleh bebas, dan tidak harus **$dbh**.

Argumen dari constructor PDO terdiri dari 3 bagian, bagian pertama berisi **nama database server** (misalnya:mysql) kemudian diikuti dengan **alamat server** dan **nama database**, semuanya di dalam 1 string. Untuk argumen kedua diisi dengan **nama user**, dan argumen ketiga berisi **password user**.

Sebagai contoh, untuk masuk ke dalam MySQL server di komputer **localhost**, dengan user **root,** nama database: **tiketpenerbangan**, kode PDO yang digunakan adalah sebagai berikut:

|  |  |
| --- | --- |
|  |  |

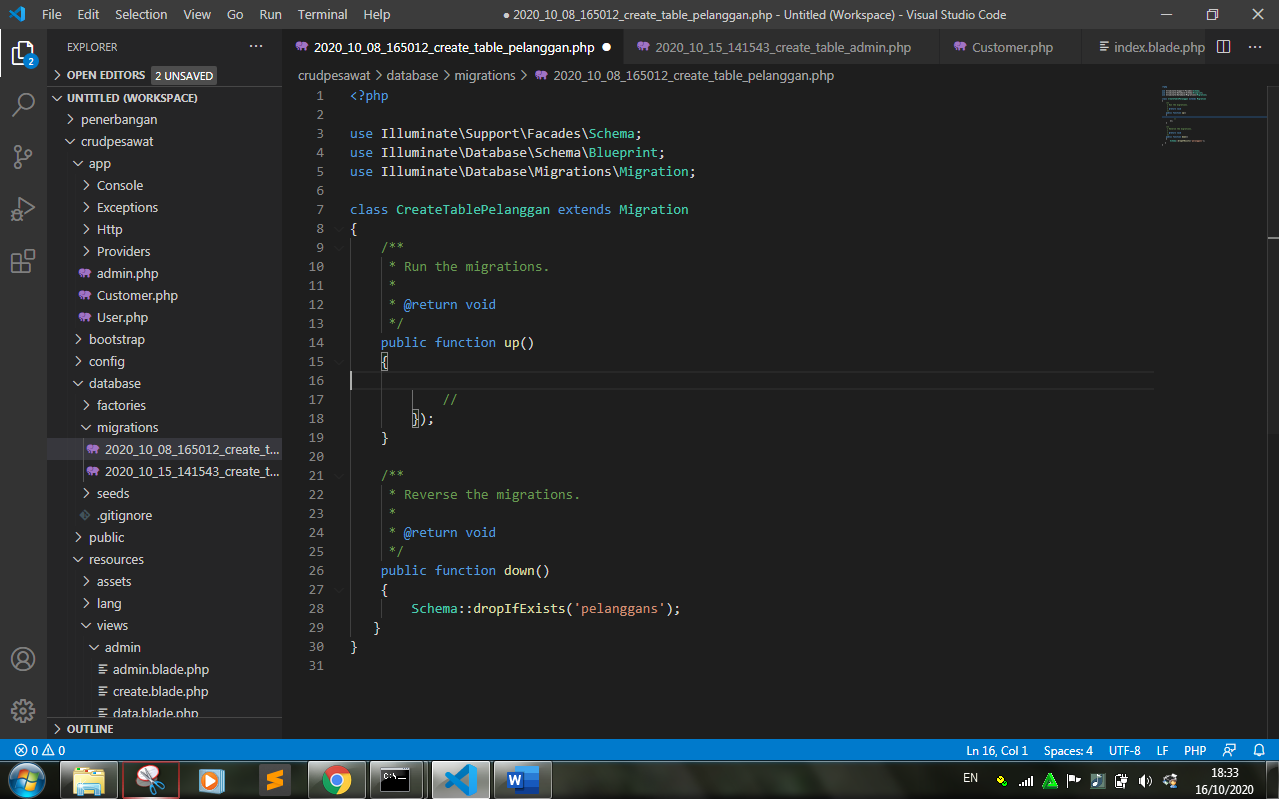
Setelah membuat koneksi PDO, untuk menghapus koneksi kita tinggal memberikan nilai “**null**” kepada variabel **“Database Handler”** sebagai berikut:

|  |  |
| --- | --- |
|  |  |

Sama seperti **mysqli**, proses penutupan koneksi ini bersifat opsional, dan PHP secara otomatis akan menghapus koneksi pada saat halaman PHP selesai dijalankan.

## 1. Koneksikan laravel dengan database di mysql.

Untuk tutorialnya bisa dilihat di artikel yang saya tulis sebelumnya di [Mengkoneksikan Antara Laravel Dengan Database Mysql atau MariaDB](https://medium.com/@diszaid/mengkoneksikan-antara-laravel-dengan-database-mysql-atau-mariadb-fd49bcb5ca90)



lalu tambahkan syntak dibawah ini pada public fuction up() diantara kurung kurawal.

        Schema::create('pelanggans', function (Blueprint $table) {

            $table->string('nama');

            $table->string('pemberangkatan');

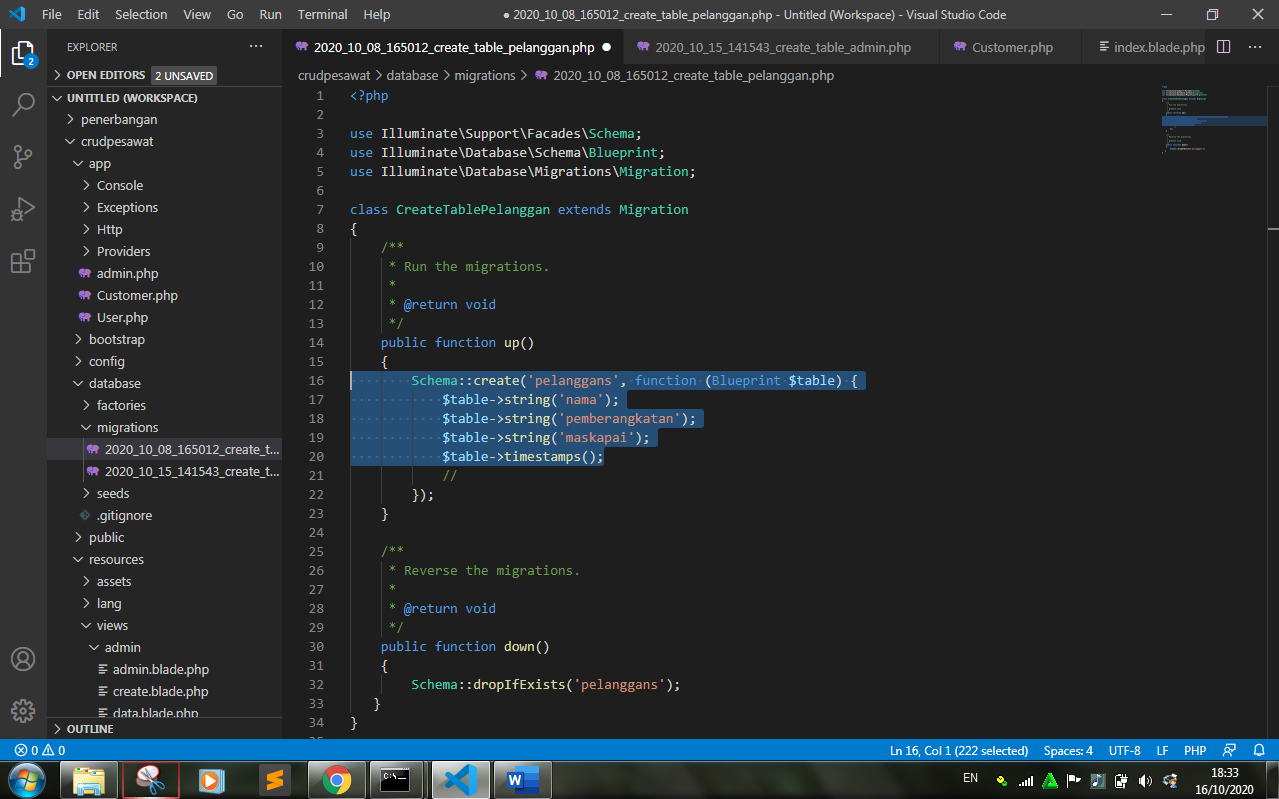
            $table->string('maskapai');

            $table->timestamps();

            //

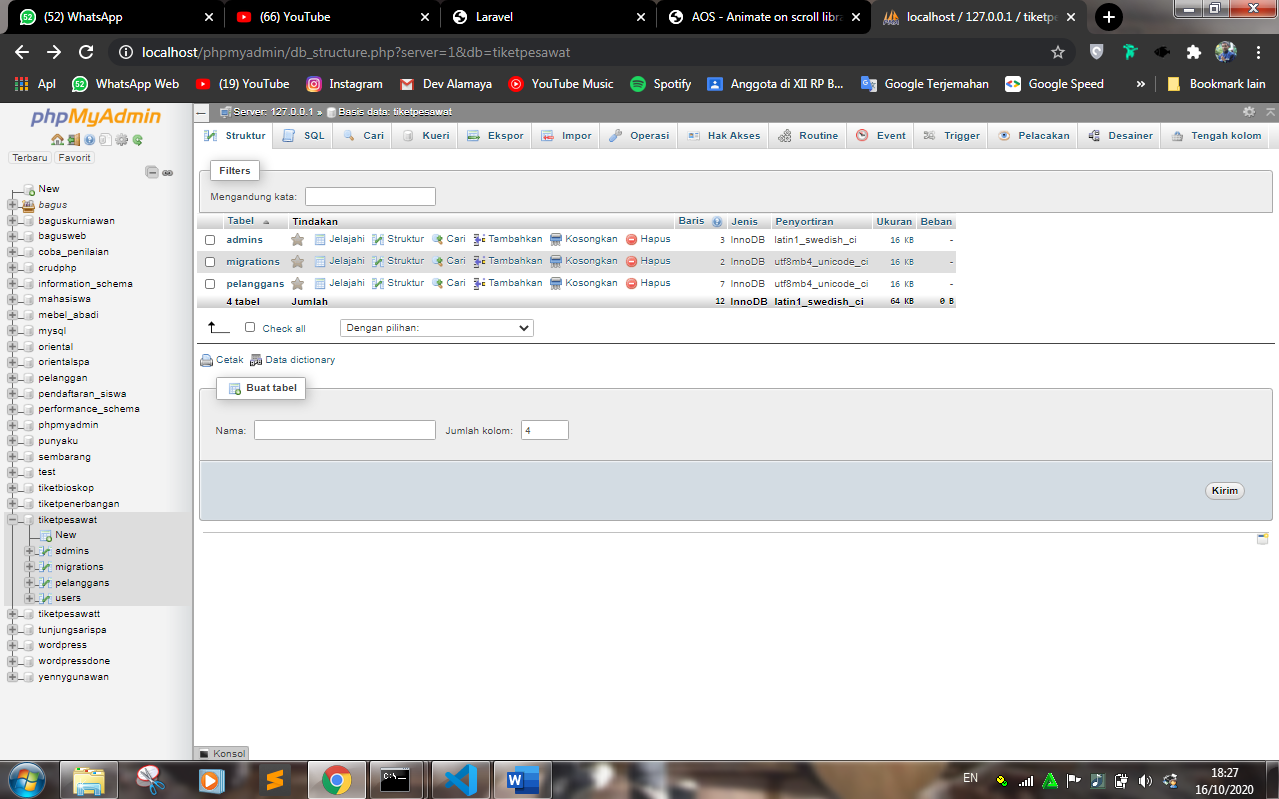
        });

jika sudah, maka akan tampil syntak keseluruhan seperti gambar dibawah ini :



setting default string length

setelahnya, cobalah delete semua tabel yang sudah dibuat dengan migrasi sebelumnya. Dengan mencentang kedua tabel tersebut lalu pilih drop.



jika sudah dihapus, cobalah migrate kembali dengan memasukan perintah seperti berikut :

php artisan migrate

tabel di database setelah migrate

## 2. Buat Model Dengan Nama User, Setting Route, Dan Tambah Controller

Setelahnya kita buat model User, tapi berhubung model user sudah tersedia, maka kita tinggal buka web.php yang berada di folder **route**.

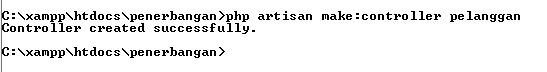
Tambahkan syntak sebagai berikut pada web.php

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

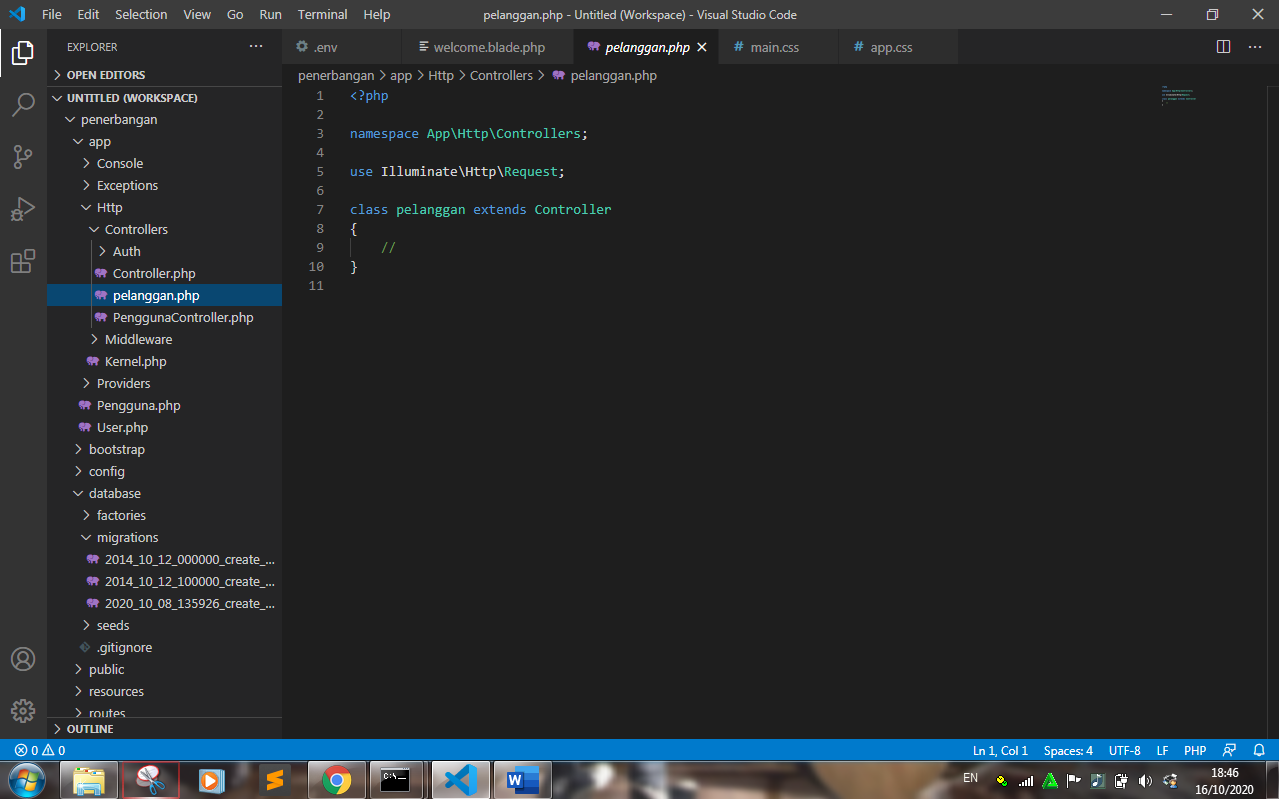
otomatis kita harus membuat controller dengan nama crud. Untuk membuatnya sangatlah mudah, masukan perintah dibawah ini.

php artisan make:controller pelanggan

maka akan muncul tampilan sebagai berikut pada terminal atau cmd. Atinya bahwa kita berhasil menambahkan controller baru.



dimana file controller.? tepatnya ada pada folder **app->http->controller**. Ini adalah tampilan dari file controller crud.php



controller Pelanggan

**3. Membuat Model Customer**



**4. Membuat tampilan home**

Langkah selanjutnya adalah membuat kode di resources/welcome.blade.php yang merupakan halaman utama

<!doctype html>

<html lang="{{ app()->getLocale() }}">

    <head>

        <meta charset="utf-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1">

        <title>Tiket Pesawat</title>

        <!-- Fonts -->

        <link href="https://fonts.googleapis.com/css?family=Raleway:100,600" rel="stylesheet" type="text/css">

        <link href="https://unpkg.com/aos@2.3.1/dist/aos.css" rel="stylesheet">

        <script src="https://unpkg.com/aos@2.3.1/dist/aos.js"></script>

        <!-- Styles -->

        <style>

            html, body {

                background-color: #fff;

                color: #636b6f;

                font-family: 'Raleway', sans-serif;

                font-weight: 100;

                height: 100vh;

                margin: 0;

            }

            .full-height {

                height: 100vh;

            }

            .flex-center {

                align-items: center;

                display: flex;

                justify-content: center;

            }

            .position-ref {

                position: relative;

            }

            .top-right {

                position: absolute;

                right: 10px;

                top: 18px;

            }

            .content {

                text-align: center;

            }

            .title {

                font-size: 84px;

            }

            .links > a {

                color: #636b6f;

                padding: 0 25px;

                font-size: 12px;

                font-weight: 600;

                letter-spacing: .1rem;

                text-decoration: none;

                text-transform: uppercase;

            }

            .m-b-md {

                margin-bottom: 30px;

            }

            img.gambarpesawat {

                height: 100%;

                width: 32%;

            }

        </style>

    </head>

    <body>

        <div class="flex-center position-ref full-height">

            <div class="content">

                <div data-aos="zoom-in">

                <img class="gambarpesawat" src="/img/pesawat.png" alt="pesawat logo">

                </div>

                <div class="title m-b-md" data-aos="zoom-in-down">

                    TIKET PESAWAT

                </div>

                <div class="links">

                    <a href="{{ url('/customer/create') }}">BELI TIKET</a>

                    <a href="{{ url('/customer') }}">ADMIN</a>

                </div>

            </div>

        </div>

        <script>

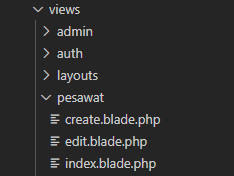
            AOS.init();

          </script>

    </body>

</html>

1. **Membuat folder pesawat dan mengisikan file create.blade.php, edit.blade.php, index.blade.php**



1. **Mengisikan coding pada model costumer**

**<**?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class customer extends Model

{

    protected $table = 'pelanggans';

    public $timestamps = false;

    protected $fillable = ['id','nama','pemberangkatan','arrive','maskapai'];

}

1. **Menuju Pelanggan Controller dan memasukan script ini**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\customer;

class Pelanggancontroller extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index()

    {

         $pelanggans = customer::all();

         return view('pesawat.index', ['pelanggans' => $pelanggans]);

    }

1. **Lalu menuju ke folder pesawat dan masuk ke file index.blade.php dan memasukkan script untuk mengambil table database dan juga tampil data**

@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row">

<div class="row">

        <div class="col-md-12">

            <div class="full-right">

                <h2>TIKET PESAWAT </h2>

            </div>

        </div>

    </div>

    <table class="table table-striped">

        <thead>

        <tr>

            <th>CUSTOMER ID :</th>

            <th>CUSTOMER NAME :</th>

            <th>LANDING FROM :</th>

            <th>ARRIVE :</th>

            <th>AIRLINES :</th>

        </tr>

        </thead>

        <tbody>

            @foreach ($pelanggans as $pelanggan)

            <tr>

            <td>{{ $pelanggan->id }}</td>

            <td>{{ $pelanggan->nama }}</td>

            <td>{{ $pelanggan->pemberangkatan }}</td>

            <td>{{ $pelanggan->arrive }}</td>

            <td>{{ $pelanggan->maskapai }}</td>

            <td>

                <a class="btn btn-primary" href="{{ url("/") }}">EDIT DATA</a>

            </td>

            <td>

                <form action="{{ url("/") }}" method="post">

                    <button onclick="myFunction()" class="btn btn-danger">DELETE</button>

                </form>

            </td>

            </tr>

            @endforeach

        </tbody>

    </table>

</div>

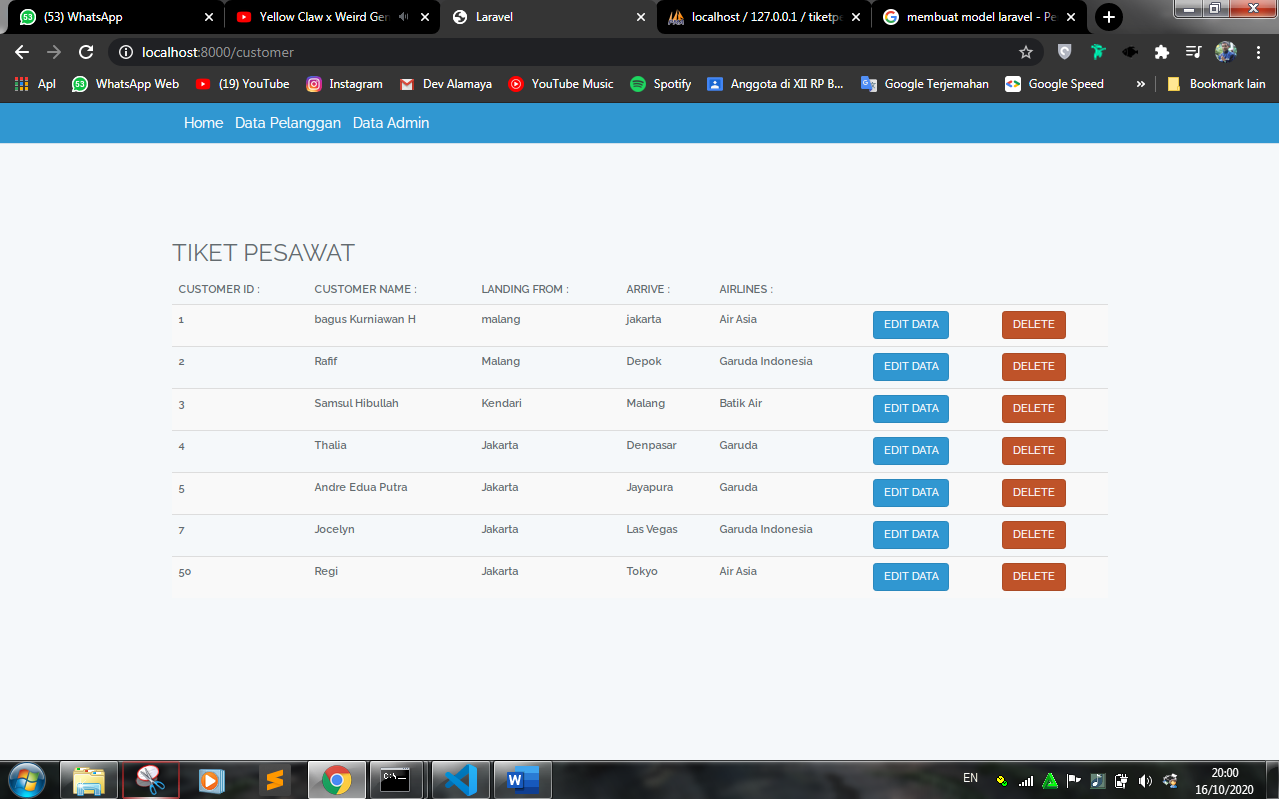
</div>

@endsection

1. **Lalu menuju ke folder routes dan click file web.php, dan masukkan script ini untuk mengambil halaman customer**

Route::resource('customer', 'PelangganController');

1. **Tampilan pada page customer**



Kelupaan 1 hal selanjutnya kita membuat tampilan header menjadi berwarna dan menambah kan menu di header

Pertama masuk ke folder layout dan click file app.blade.php



1. **Lalu menambahkan script dibawah tag button**

                   <!-- Branding Image -->

                    <ul class="nav navbar-nav">

                        <li><a class="navbar-brand"  style="color: white;" href="{{ url('/') }}">Home</a></li>

                    </ul>

                </div>

                <div class="collapse navbar-collapse" id="app-navbar-collapse">

                <ul class="nav navbar-nav">

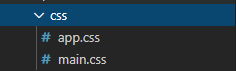
            <li> <a class="navbarbrand"style="color: white;" href="{{ url('/customer') }}">

Data Pelanggan</a></li>

                        <li><a class="navbar-brand"  style="color: white;" href="{{ url('/admin') }}">Data Admin</a></li>

                    </ul>

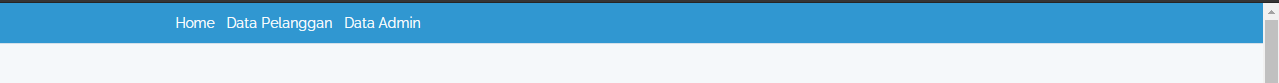
1. **Selanjutnya masuk ke folder public dan pilih app.css**



1. **Lalu memasukkan styling css berikut untuk mengganti warna pada header**



1. **Maka header akan terlihat seperti ini**



**MEMBUAT SIMPAN DATA**

* 1. Langkah berikut nya adalah membuat simpan data yang pertama adalah masuk ke create.blade.php dan masukkan script untuk membuat form dan juga memanggil table database

@extends('layouts.app')

@section('content')

    <div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

            <h2 class="display-6">TAMBAH DATA BARU</h2>

        </div>

        </div>

    </div>

    <div class="container">

    <div class="row">

        <style>

          </style>

        <div class="col-md-8 offsett-sm-2">

        <form action="{{ url("/customer") }}" method="post">

            {!! csrf\_field() !!}

            <div  class="form-group">

                <label for="id">ID Pengguna :</label>

                <input class="from-control" type="text" name="id" style="margin: 20px 90px;">

            </div>

            <div  class="form-group">

                <label for="nama">Nama :</label>

                <input class="from-control" type="text" name="nama" style="margin: 20px 137px;">

            </div>

            <div  class="form-group">

                <label for="pemberangkatan">Pemberangkatan :</label>

                <input class="from-control" type="text" name="pemberangkatan" style="margin: 20px 60px;">

            </div>

            <div  class="form-grup">

                <label for="arrive">Tujuan :</label>

                <input class="from-control" type="text" name="arrive" style="margin: 20px 128px;">

            </div>

            <div  class="form-grup">

                <label for="maskapai">Nama Maskapai :</label>

                <input class="from-control" type="text" name="maskapai" style="margin: 20px 71px;">

            </div>

            <button onclick="myFunction()" type="submit" class="btn btn-primary" style="margin: 20px 0px;

            padding: 5px 160px;">SAVE</button>

            <input type="reset" value="Reset" style="padding: 5px 50px;

            margin: 0px 40px;

            background-color: #bf5329;

            border-color: #aa4a24;

            color:white;

            border-radius: 5px;">

        </form>

        </div>

    </div>

    </div>

    <script>

        function myFunction() {

          alert("Data Berhasil Tersimpan");

        }

        </script>

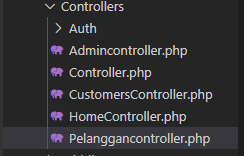
@endsection

* 1. Langkah berikut nya adalah membuat route di web.php untuk menampilkan page create data

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

Route::post('/customer', 'PelangganController@store');

* 1. Berikut nya kita masuk ke controller pelanggan di PelangganController.php



* 1. Memasukkan script pada controller pada public function create() yang kosong kita isi dengan script ini

    public function create()

    {

        return view('pesawat.create');

    }

* 1. Lalu kita menuju ke public function store(request $request) dan disni kita memanggil atribut table pelanggans dan juga id pada form yang ada di file create.blade.php

        $pelanggans = new customer ([

            'id'=> $request-> input('id'),

            'nama'=> $request-> input('nama'),

            'pemberangkatan'=> $request-> input('pemberangkatan'),

            'arrive'=> $request-> input('arrive'),

            'maskapai'=> $request-> input('maskapai')

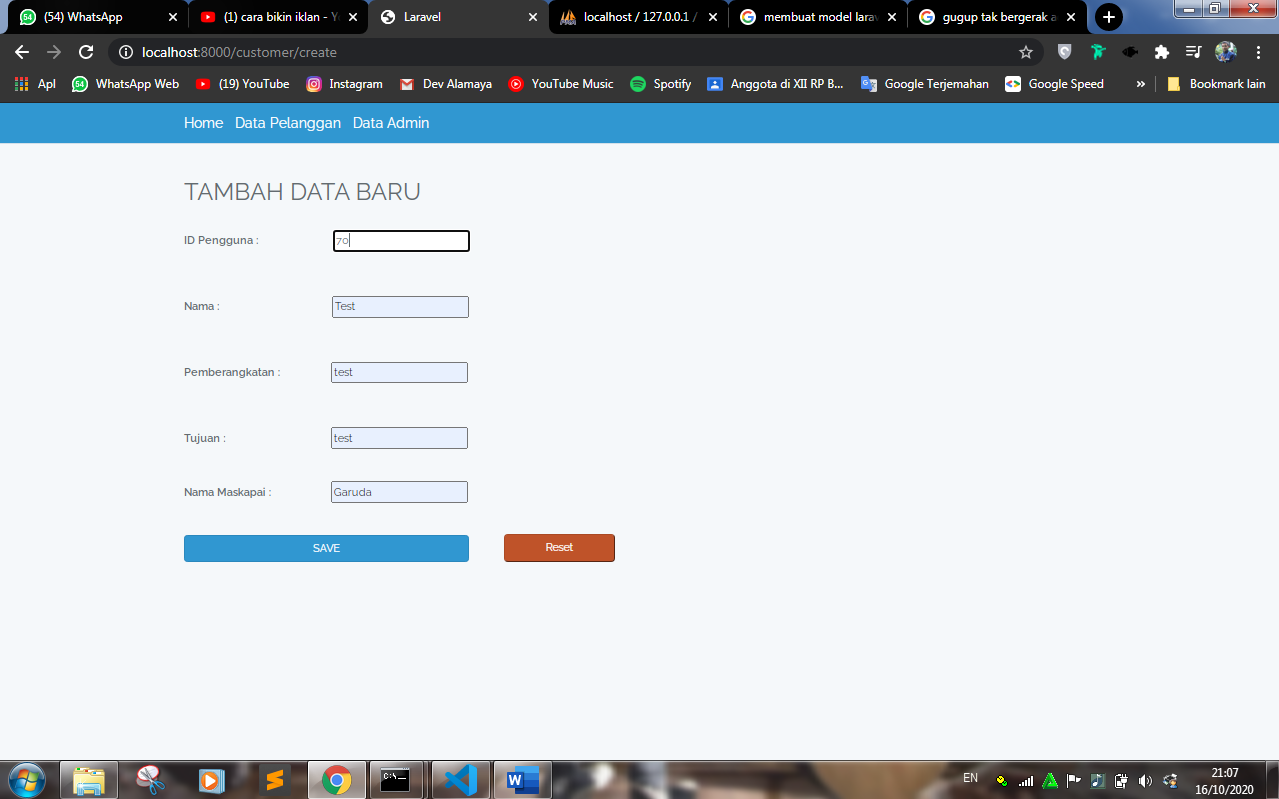
        ]);

        $pelanggans->save();

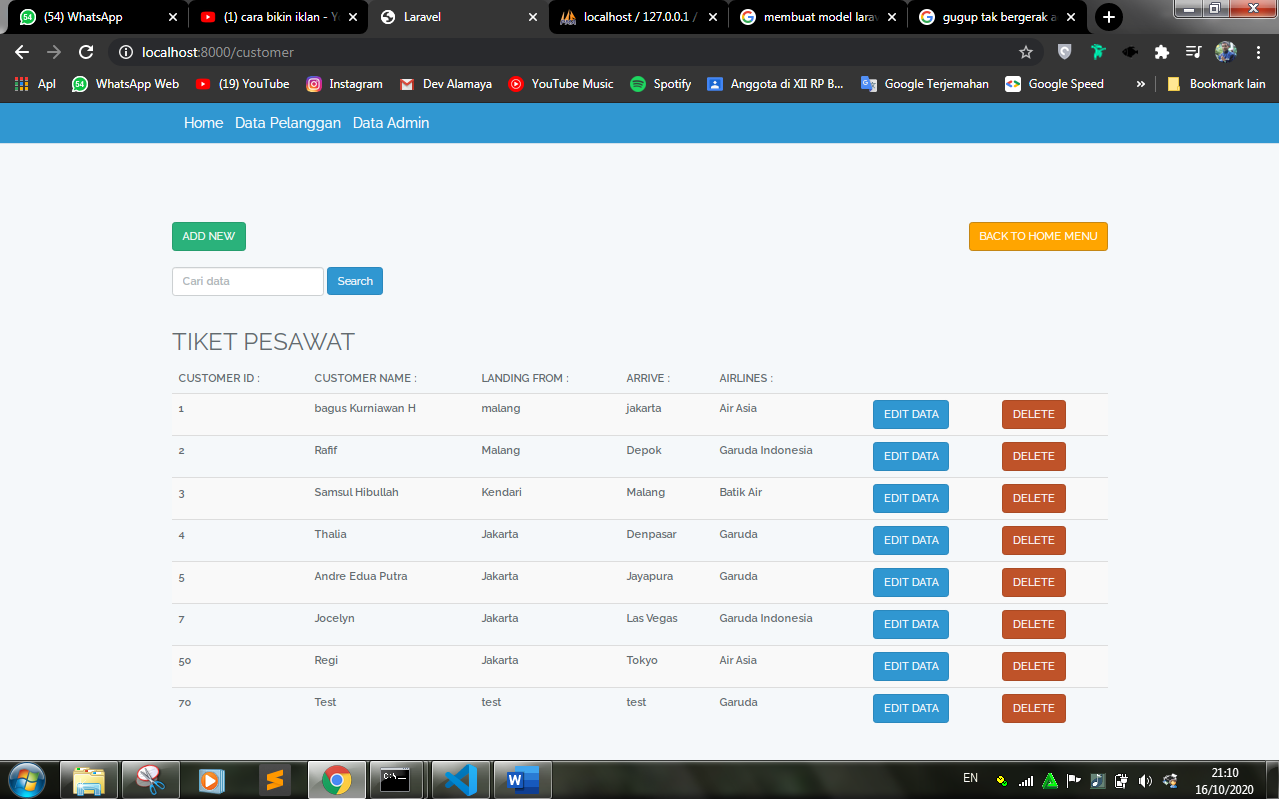
        return redirect('/customer');

    }

* 1. Kita mencoba untuk menginput data pada form tersebut



* 1. Data berhasil masuk



**MEMBUAT EDIT DATA**

1. Langkah selanjutnya adalah membuat edit data pada bagian pertama kita masuk ke folder pesawat dan pilih file edit.blade.php dan memasukkan form yang ada di create.blade.php namun ada perubahan

@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

            <h2 class="display-6">EDIT DATA</h2>

        </div>

    </div>

</div>

    <div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

        <form action="{{ url("/customer/{$pelanggans->id}") }}" method="post">

            {{ method\_field('PATCH') }}

            {!! csrf\_field() !!}

            <div  class="form-group">

                <label for="id">ID Pengguna :</label>

                <input value="{{ $pelanggans->id }}" class="from-control" type="text" name="id" style="margin: 20px 90px;">

            </div>

            <div  class="form-group">

                <label for="nama">Nama :</label>

                <input value="{{ $pelanggans->nama }}" class="from-control" type="text" name="nama" style="margin: 20px 137px;">

            </div>

            <div  class="form-group">

                <label for="pemberangkatan">Pemberangkatan :</label>

                <input value="{{ $pelanggans->pemberangkatan }}" class="from-control" type="text" name="pemberangkatan" style="margin: 20px 60px;">

            </div>

            <div  class="form-grup">

                <label for="arrive">Tujuan :</label>

                <input value="{{ $pelanggans->arrive }}" class="from-control" type="text" name="arrive" style="margin: 20px 128px;">

            </div>

            <div  class="form-grup">

                <label for="maskapai">Nama Maskapai :</label>

                <input value="{{ $pelanggans->maskapai }}" class="from-control" type="text" name="maskapai" style="margin: 20px 71px;">

            </div>

            <button type="submit" onclick="myFunction()" class="btn btn-primary">SAVE</button>

        </form>

        </div>

    </div>

    </div>

    <script>

        function myFunction() {

          alert("Data Berhasil Di Edit");

        }

        </script>

@endsection

1. Lalu kita masuk ke file web.php dan memasukkan route untuk memanggil halaman edit data pelanggan

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

Route::patch('/pelanggans/{id}/update', 'PelangganController@update');

1. Langkah berikut nya kita masuk ke PelangganController.php dan memasukkan script berikut Pada public function edit dan public function update

   public function edit($id)

    {

        $pelanggans = customer::find($id);

        return view('pesawat.edit', compact('pelanggans'));

    }

    public function update(Request $request, $id)

    {

        $pelanggans = customer::find($id);

        $pelanggans->id = $request->input('id');

        $pelanggans->nama = $request->input('nama');

        $pelanggans->pemberangkatan = $request->input('pemberangkatan');

        $pelanggans->arrive = $request->input('arrive');

        $pelanggans->maskapai = $request->input('maskapai');

        $pelanggans->save();

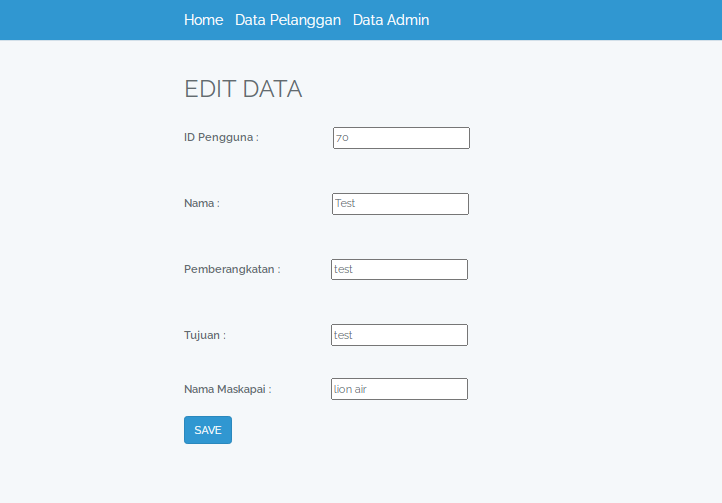
        return redirect('/customer');

    }

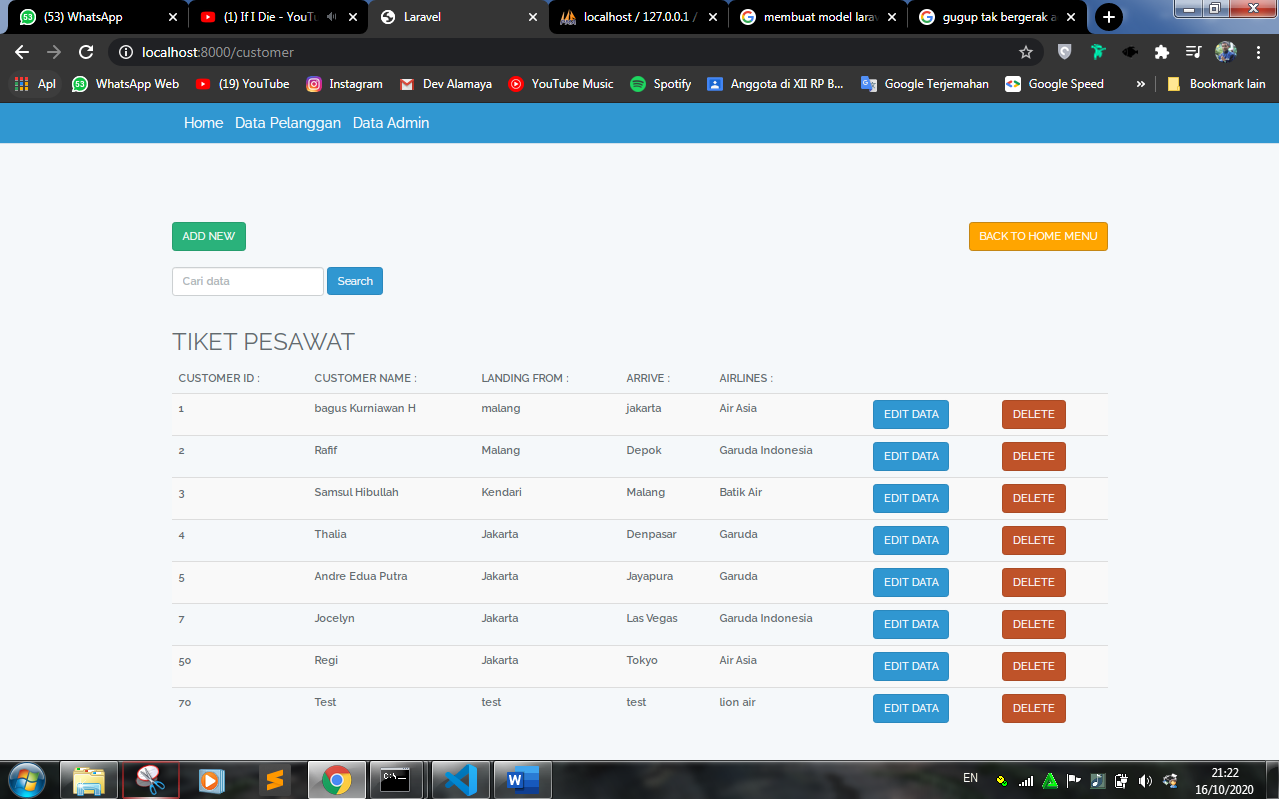
1. Langkah berikut nya adalah mengisi link pada button dengan script yang akan melakukan link menuju update form

<a class="btn btn-primary" href="{{ url("pelanggans/{$pelanggan->id}/edit") }}">EDIT DATA</a>

1. Mencoba meng edit data dengan mengganti nama maskapai dari garuda menjadi lion air



1. Data berhasil di edit



**MEMBUAT DELETE DATA**

1. Langkah selanjutnya adalah membuat delete data, untuk membuat delete data ini pertama kita menuju PelangganController.php dan memasukkan script pada public function destroy Untuk melakukan delete data pelanggan

    public function destroy($id)

    {

        $pelanggans = customer::find($id);

        $pelanggans -> delete();

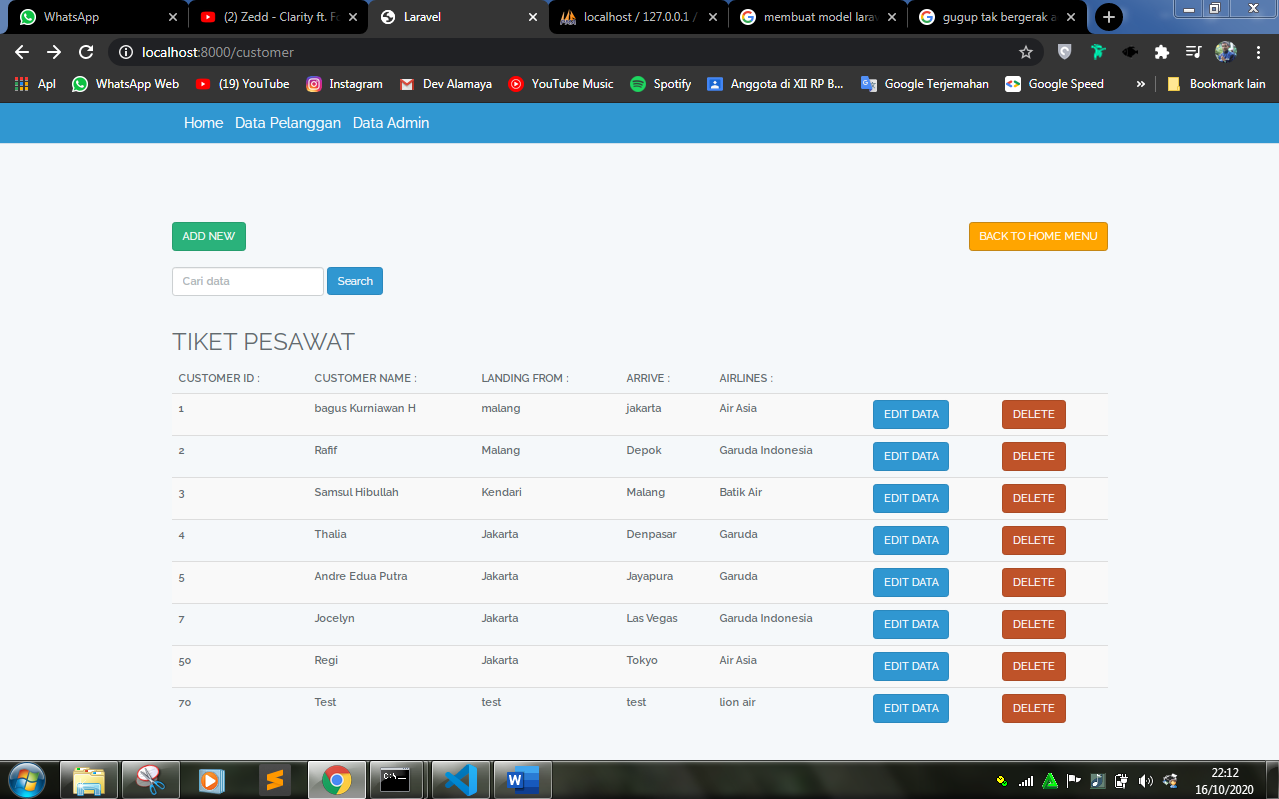
        return redirect('/customer');

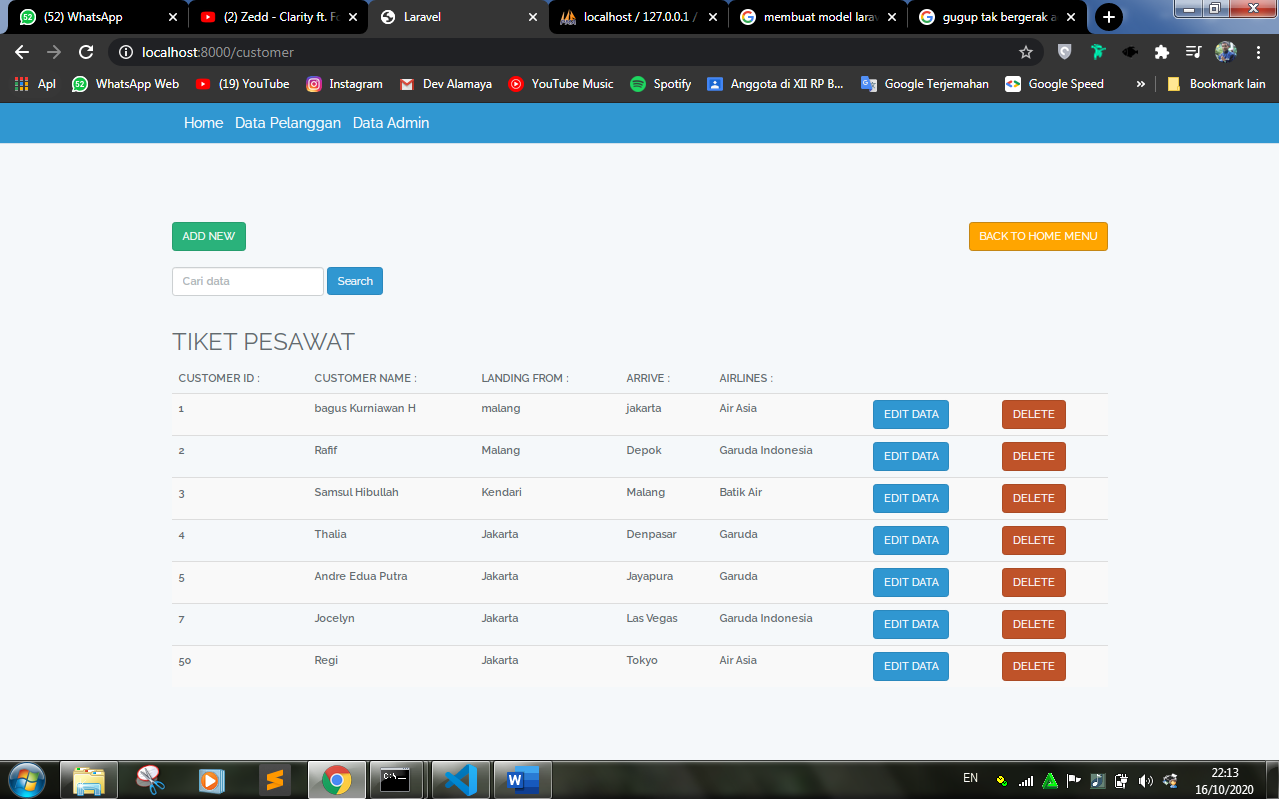
    }

1. Langkah selanjutnya adalah membuat route pada web.php

Route::delete('/pelanggans/{id}', 'PelangganController@destroy');

1. Maka tampilan delete akan seperti ini





**MEMBUAT CARI DATA**

1. Langkah selanjutnya adalah membuat cari data langkah pertama adalah membuat form dan juga name variable keyword

{{--  cari  --}}

    <form action="{{ url("/customer") }}" class="form-inline" method="GET" style="margin: 20px 0px">

        <div class="form-group mx-sm-3 mb-2">

        <input value="{{Request::get('keyword') }}" name="keyword" type="text" class="form-control" placeholder="Cari data">

        </div>

        <button type="submit" class="btn btn-primary mb-2">Search</button>

    </form>

{{--  /cari  --}}

2. Selanjutnya menuju ke PelangganController.php dan memasukkan script pencarian data di function index dengan menambahkan parameter request dan if

    public function index(Request $request)

    {

        $keyword = $request->get('keyword');

         $pelanggans = customer::all();

        if ($keyword) {

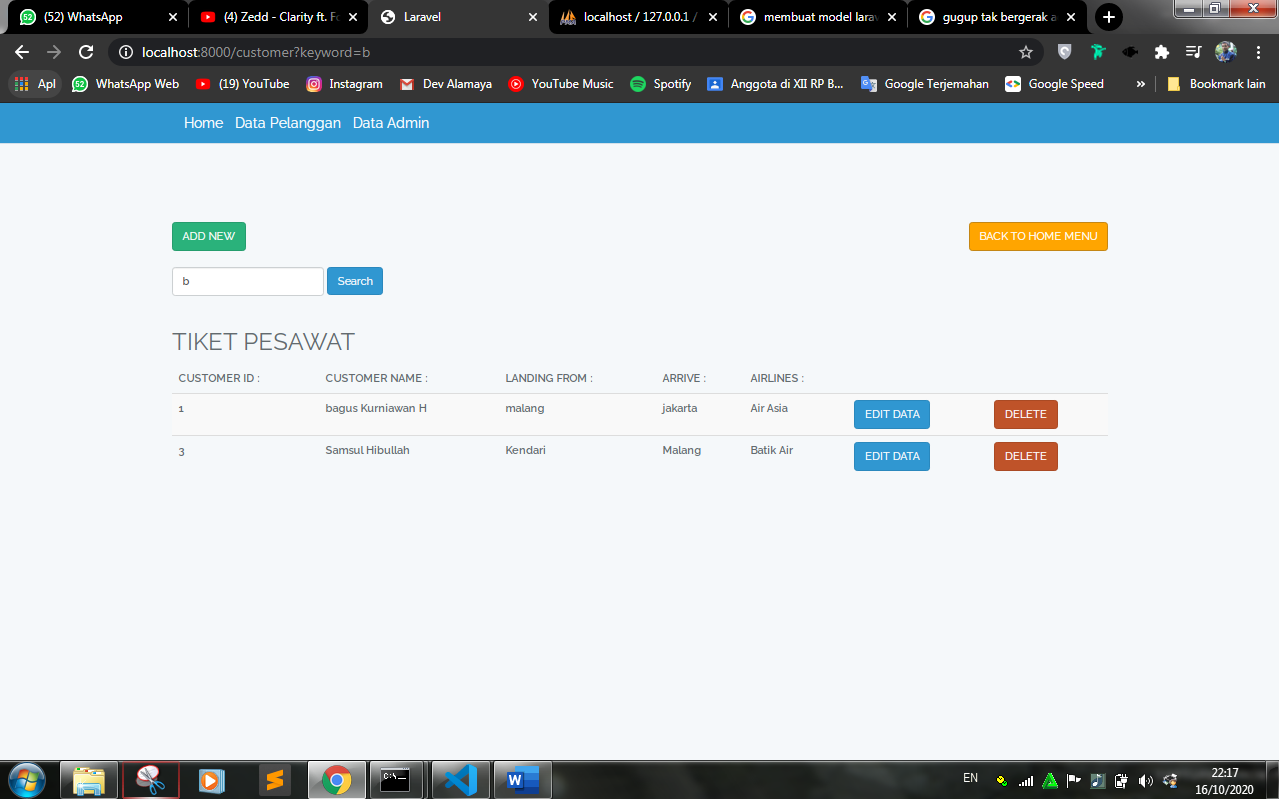
            $pelanggans = customer::where("nama","LIKE","%$keyword%")->get();

        }

         return view('pesawat.index', ['pelanggans' => $pelanggans]);

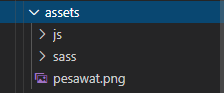
    }

1. Tampilan pencarian data



1. Memasukkan Gambar untuk home page

Pertama pastikan gambar masuk dalam 1 folder yaitu di folder assets



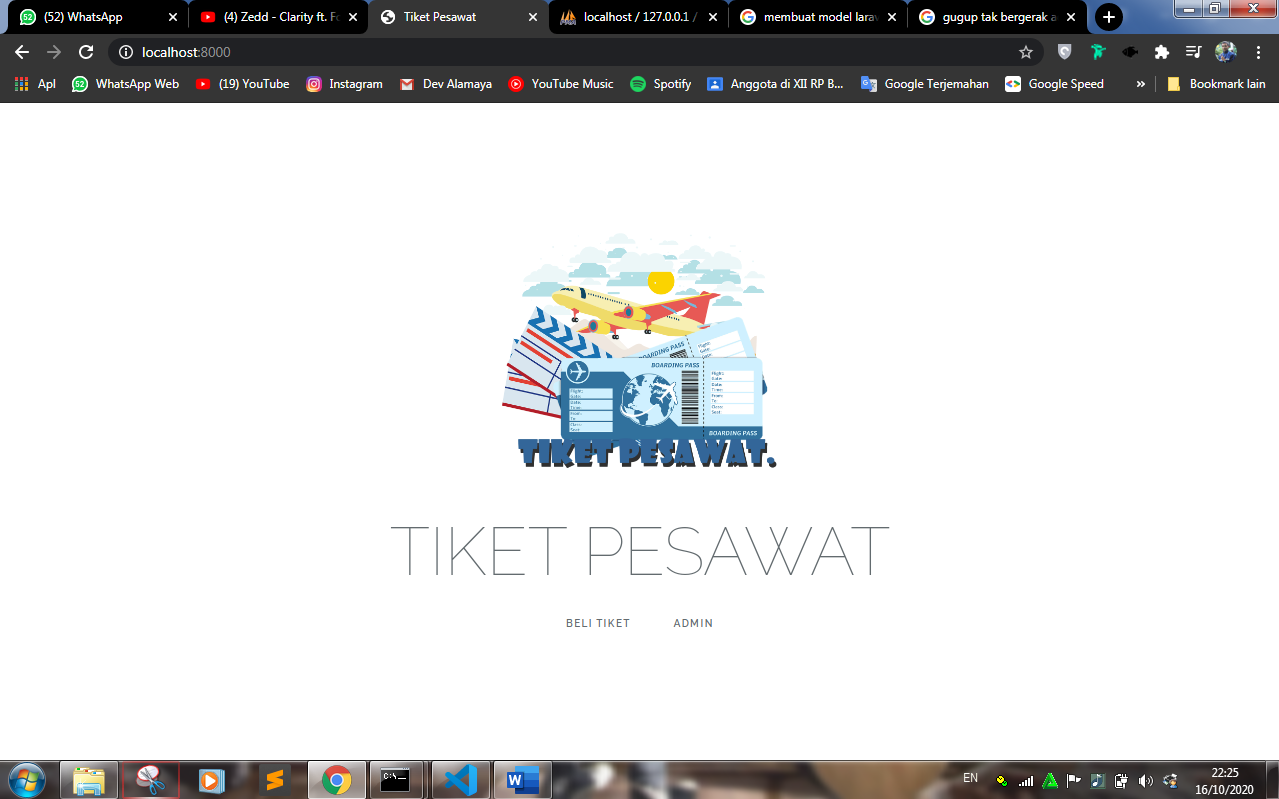
1. Langkah selanjutnya adalah memasukkan gambar ke website dengan script berikut

 <img class="gambarpesawat" src="/img/pesawat.png" alt="pesawat logo">

1. Beri styling untuk mengatur ukuran gambar tersebut



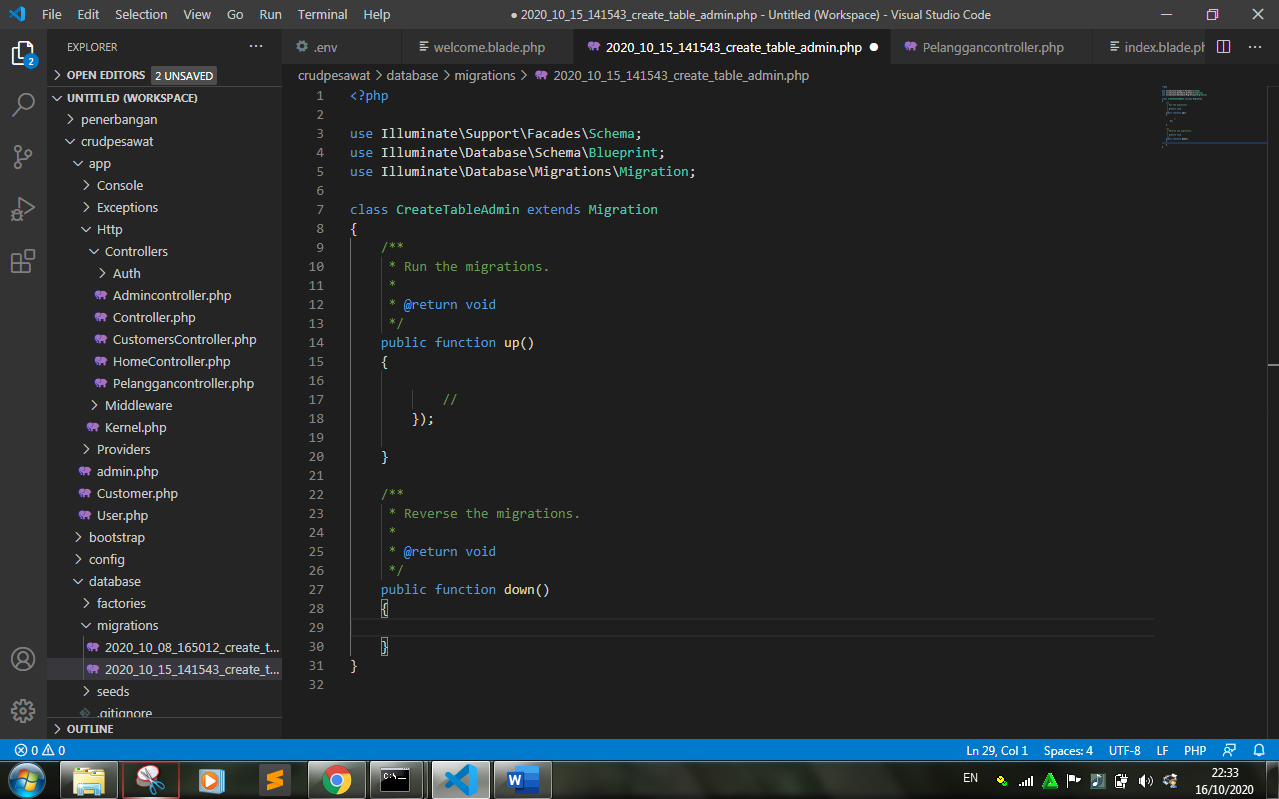
1. Maka tampilan home page akan seperti ini



**MEMBUAT MODEL CONTROLLER DAN VIEW PADA ADMIN**

## Koneksikan laravel dengan database di mysql.

Untuk tutorialnya bisa dilihat di artikel yang saya tulis sebelumnya di [Mengkoneksikan Antara Laravel Dengan Database Mysql atau MariaDB](https://medium.com/@diszaid/mengkoneksikan-antara-laravel-dengan-database-mysql-atau-mariadb-fd49bcb5ca90)



1. lalu tambahkan syntak dibawah ini pada public fuction up() diantara kurung kurawal.

    public function up()

    {

        Schema::create('admin', function (Blueprint $table) {

            $table->string('id\_admin');

            $table->string('nm\_admin');

            $table->string('telepon');

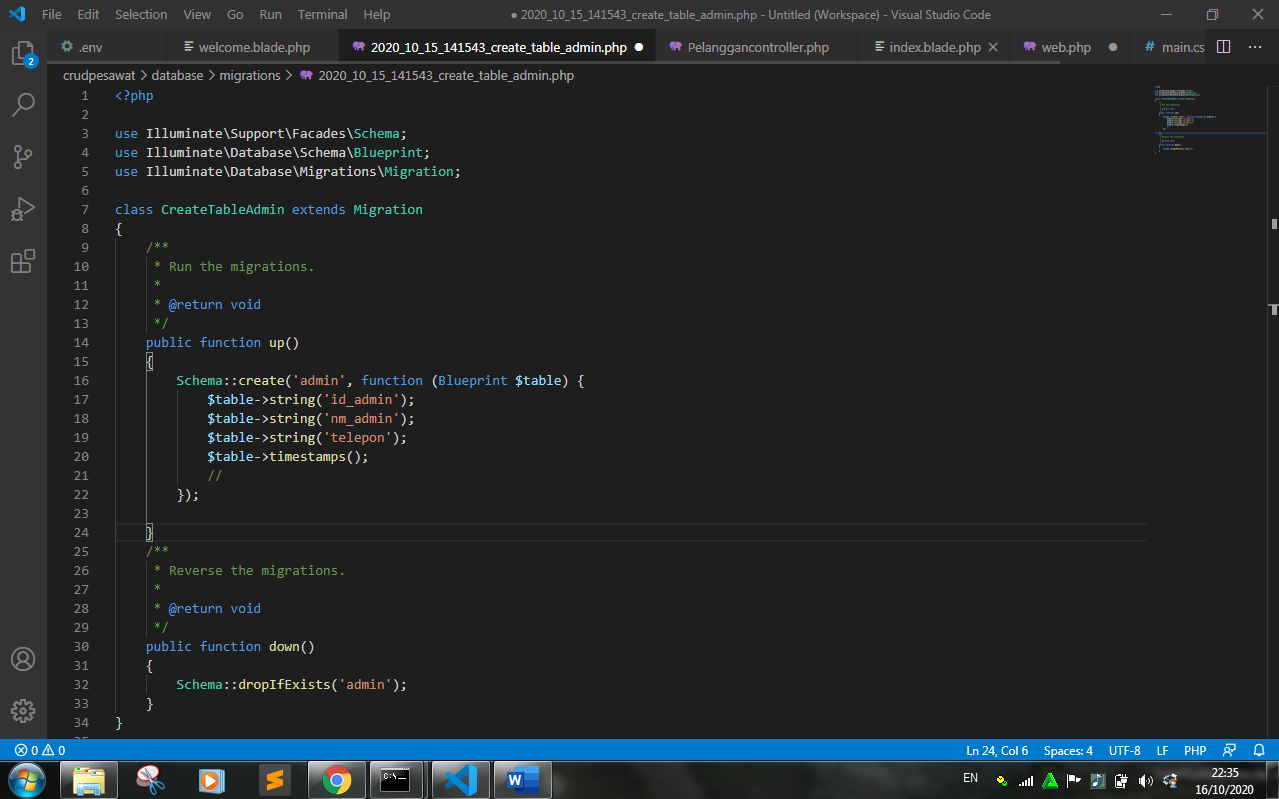
            $table->timestamps();

            //

        });

    }

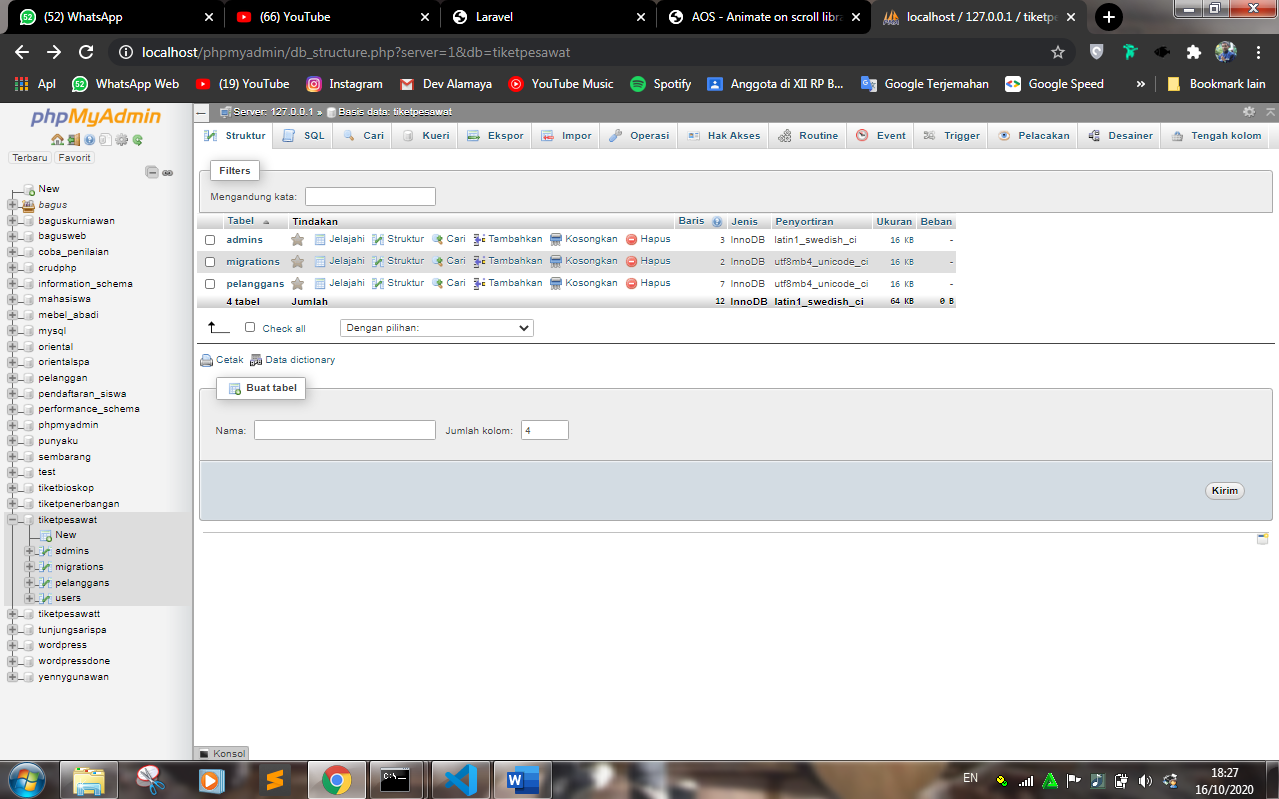
1. jika sudah, maka akan tampil syntak keseluruhan seperti gambar dibawah ini :



1. cobalah migrate kembali dengan memasukan perintah seperti berikut :

php artisan migrate

1. Maka tampilan table sekarang seperti ini



1. Setelahnya kita buat model User, tapi berhubung model user sudah tersedia, maka kita tinggal buka web.php yang berada di folder **route**. Tambahkan syntak sebagai berikut pada web.php

Route::resource('admin', 'AdminController');

Route::get('/admin', 'AdminController@index')->name('admin');

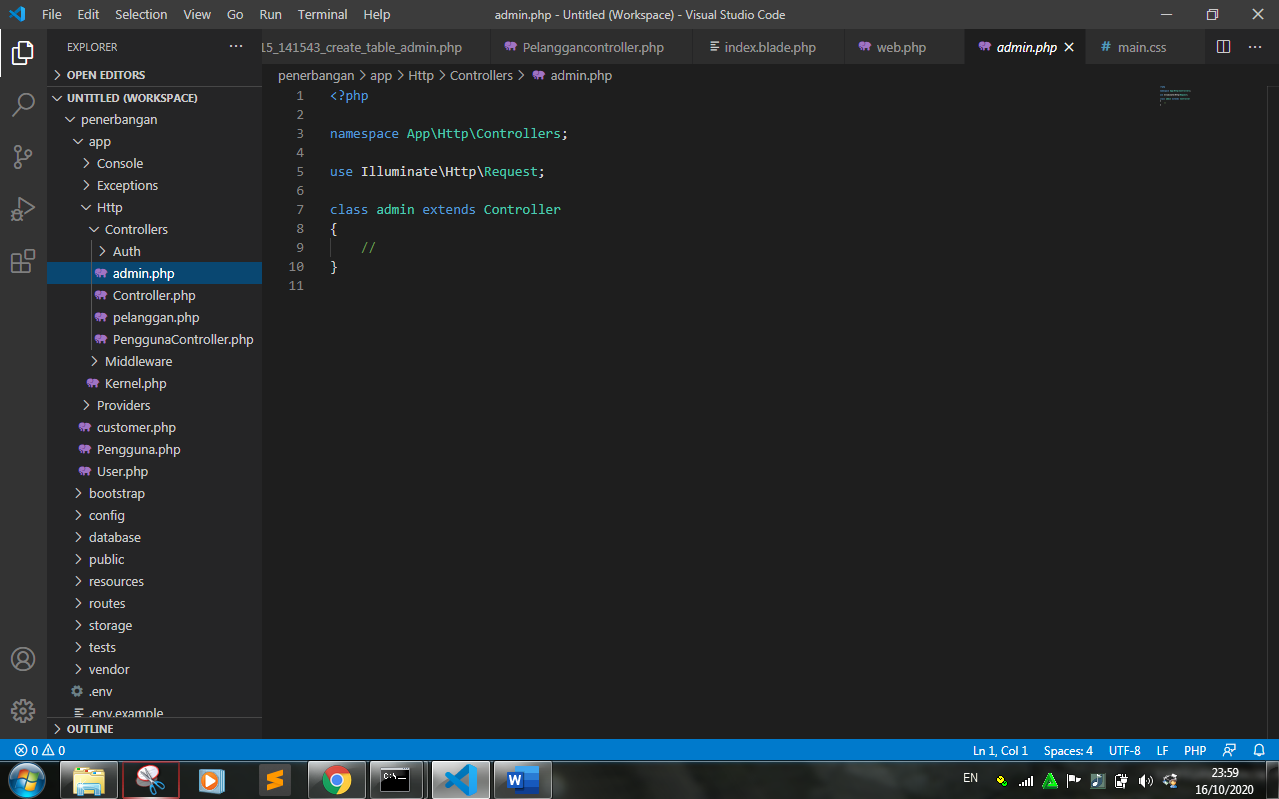
1. otomatis kita harus membuat controller dengan nama crud. Untuk membuatnya sangatlah mudah, masukan perintah dibawah ini.

php artisan make:controller admin

1. maka akan muncul tampilan sebagai berikut pada terminal atau cmd. Atinya bahwa kita berhasil menambahkan controller baru.



1. dimana file controller.? tepatnya ada pada folder **app->http->controller**. Ini adalah tampilan dari file controller admin.php

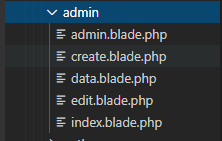


1. Membuat Model admin



1. Membuat folder dan tampilan admin

Membuat folder pesawat dan mengisikan file create.blade.php, edit.blade.php, admin.blade.php



1. Mengisikan coding pada model admin

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class admin extends Model

{

    protected $table = 'admins';

    public $timestamps = false;

    protected $fillable = ['id','nm\_admin','callnumber'];

}

1. Menuju Admin Controller dan memasukan script ini

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

use App\Admin;

class Admincontroller extends Controller

{

    /\*\*

     \* Display a listing of the resource.

     \*

     \* @return \Illuminate\Http\Response

     \*/

    public function index(Request $request)

    {

        $admins = admin::all();

        return view('admin.admin', ['admins' => $admins]);

    }

1. Lalu menuju ke folder admin dan masuk ke file admin.blade.php dan memasukkan script untuk mengambil table database dan juga tampil data

@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row">

<a href="{{ url("/") }}" class="btn" style="color: rgb(255, 255, 255);background-color: orange;border-color: #c48614;float: right;color: #fff;">BACK TO HOME MENU</a>

{{--  cari  --}}

    <form action="{{ url("/admin") }}" class="form-inline" method="GET">

        <div class="form-group mx-sm-3 mb-2">

        <input value="{{Request::get('keyword') }}" name="keyword" type="text" class="form-control" placeholder="Cari data">

        </div>

        <button type="submit" class="btn btn-primary mb-2">Search</button>

    </form>

{{--  /cari  --}}

<div class="row">

        <div class="col-md-12">

            <div class="full-right">

                <h2>ADMIN TIKET PESAWAT</h2>

                <a href="{{ url("admin/create") }}" class="btn btn-success">ADD NEW</a>

            </div>

        </div>

    </div>

    <table class="table table-striped">

        <thead>

        <tr>

            <th>ADMIN ID :</th>

            <th>ADMIN NAME :</th>

            <th>TELEPHONE</th>

        </tr>

        </thead>

        <tbody>

            @foreach ($admins  as $admin)

            <tr>

            <td>{{ $admin->id }}</td>

            <td>{{ $admin->nm\_admin }}</td>

            <td>{{ $admin->callnumber }}</td>

           <td>

                <a class="btn btn-primary"href="{{ url("/") }}">EDIT DATA</a>

            </td>

            <td>

                <form action="{{ url("/") }}" method="post">

                {!! csrf\_field() !!}

                {{ method\_field('DELETE') }}

                    <button class="btn btn-danger">DELETE</button>

                </form>

            </td>

        </tr>

            @endforeach

        </tbody>

    </table>

</div>

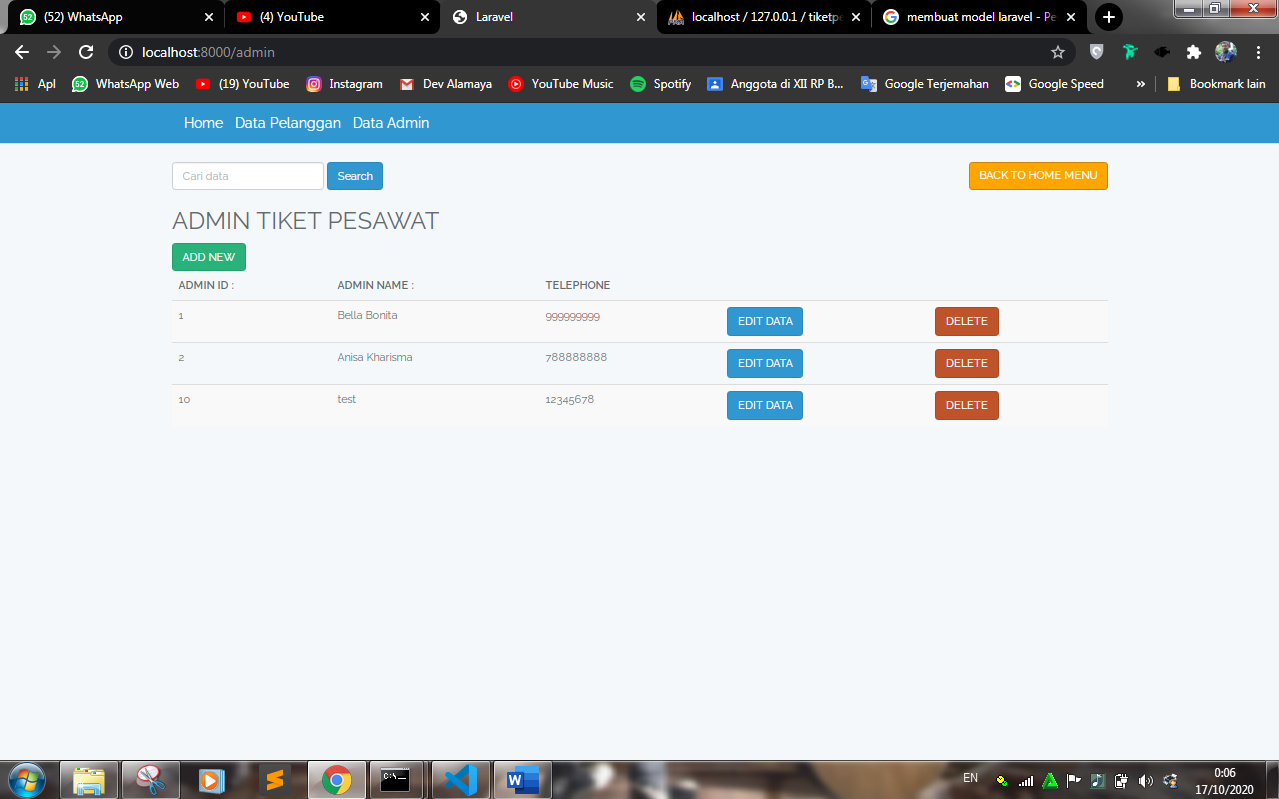
</div>

@endsection

1. Lalu menuju ke folder routes dan click file web.php, dan masukkan script ini untuk mengambil halaman customer

Route::resource('admin', 'AdminController');

1. Tampilan pada page customer



**MEMBUAT SIMPAN DATA**

1. Langkah berikut nya adalah membuat simpan data yang pertama adalah masuk ke create.blade.php dan masukkan script untuk membuat form dan juga memanggil table database

@extends('layouts.app')

@section('content')

    <div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

            <h2 class="display-6">TAMBAH DATA BARU</h2>

        </div>

        </div>

    </div>

    <div class="container">

    <div class="row">

        <style>

          </style>

        <div class="col-md-8 offsett-sm-2">

        <form action="{{ url("/admin") }}" method="post">

            {!! csrf\_field() !!}

            <div  class="form-group">

                <label for="id">ID Admin :</label>

                <input class="from-control" type="text" name="id" style="margin: 20px 90px;">

            </div>

            <div  class="form-group">

                <label for="nm\_admin">Nama Admin :</label>

                <input class="from-control" type="text" name="nm\_admin" style="margin: 20px 137px;">

            </div>

            <div  class="form-group">

                <label for="callnumber">Telepon :</label>

                <input class="from-control" type="text" name="callnumber" style="margin: 20px 60px;">

            </div>

            <button type="submit" class="btn btn-primary" style="margin: 20px 0px;

            padding: 5px 160px;">SAVE</button>

            <input type="reset" value="Reset" style="padding: 5px 50px;

            margin: 0px 40px;

            background-color: #bf5329;

            border-color: #aa4a24;

            color:white;

            border-radius: 5px;">

        </form>

        </div>

    </div>

    </div>

     <script>

        function myFunction() {

          alert("Data Berhasil Tersimpan");

        }

        </script>

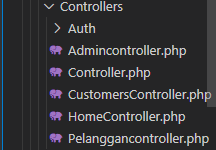
@endsection

1. Langkah berikut nya adalah membuat route di web.php untuk menampilkan page create data

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

Route::post('/admin', 'AdminController@store');

1. Berikut nya kita masuk ke controller admin di AdminController.php



1. Memasukkan script pada controller pada public function create() yang kosong kita isi dengan script ini

    public function create()

    {

        return view('admin.create');

    }

1. Lalu kita menuju ke public function store(request $request) dan disni kita memanggil atribut table admins dan juga id pada form yang ada di file create.blade.php

 public function store(Request $request)

    {

        $admins = new admin ([

            'id'=> $request-> input('id'),

            'nm\_admin'=> $request-> input('nm\_admin'),

            'callnumber'=> $request-> input('callnumber')

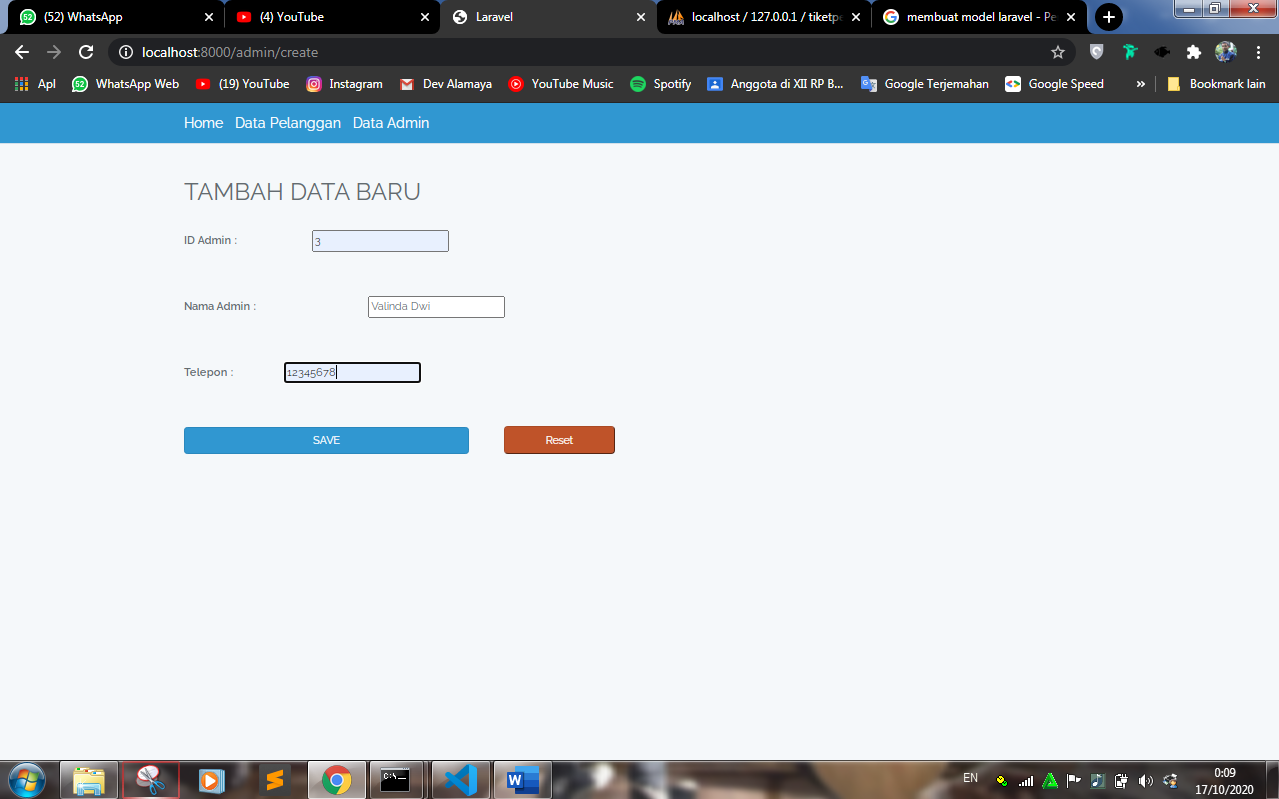
        ]);

        $admins->save();

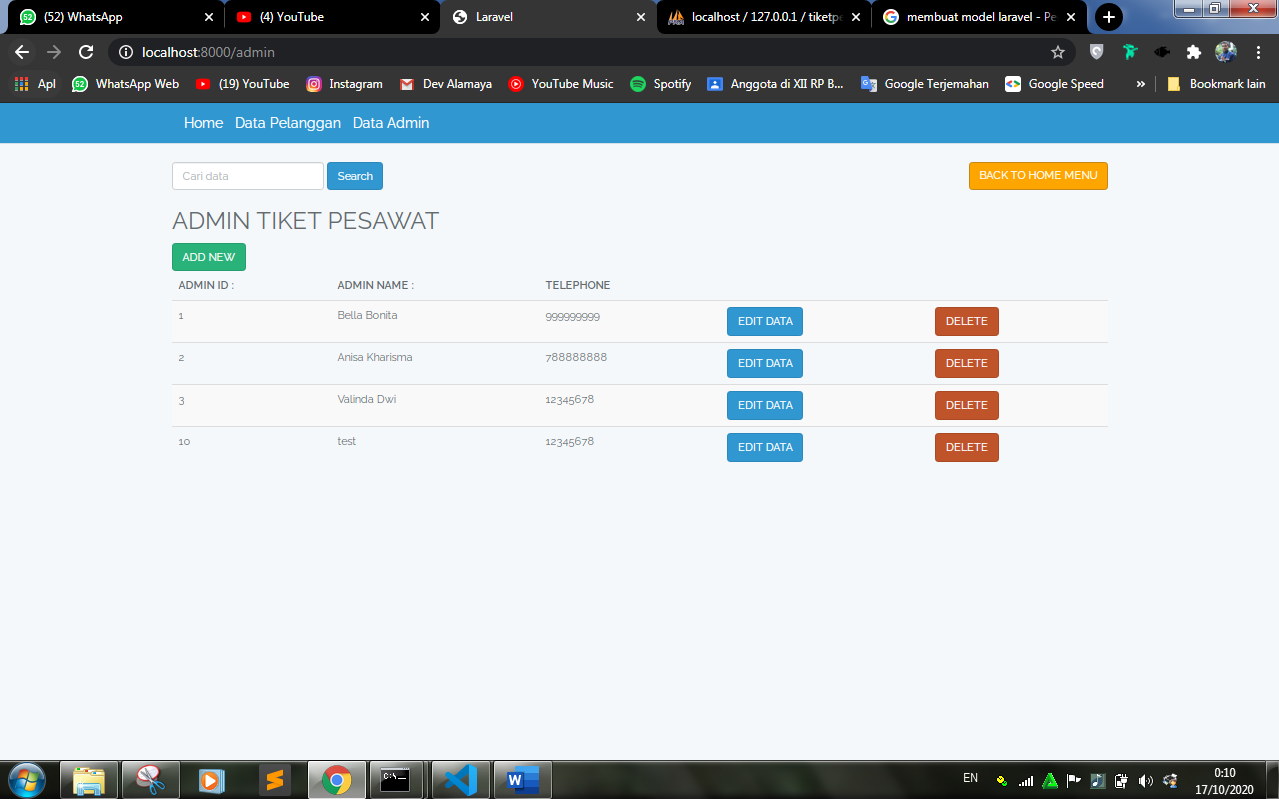
        return redirect('/admin');

    }

1. Kita mencoba untuk menginput data pada form tersebut



1. Data berhasil masuk



**MEMBUAT EDIT DATA**

1. Langkah selanjutnya adalah membuat edit data pada bagian pertama kita masuk ke folder pesawat dan pilih file edit.blade.php dan memasukkan form yang ada di create.blade.php namun ada perubahan

@extends('layouts.app')

@section('content')

<div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

            <h2 class="display-6">EDIT DATA</h2>

        </div>

    </div>

</div>

    <div class="container">

    <div class="row">

        <div class="col-md-8 offsett-sm-2">

        <form action="{{ url("/admin/{$admins->id}") }}" method="post">

            {{ method\_field('PATCH') }}

            {!! csrf\_field() !!}

            <div  class="form-group">

                <label for="id">ID Pengguna :</label>

                <input value="{{ $admins->id }}" class="from-control" type="text" name="id" style="margin: 20px 90px;">

            </div>

            <div  class="form-group">

                <label for="nm\_admin">Nama :</label>

                <input value="{{ $admins->nm\_admin }}" class="from-control" type="text" name="nm\_admin" style="margin: 20px 137px;">

            </div>

            <div  class="form-group">

                <label for="callnumber">Pemberangkatan :</label>

                <input value="{{ $admins->callnumber }}" class="from-control" type="text" name="callnumber" style="margin: 20px 60px;">

            </div>

            <button type="submit" onclick="myFunction()" class="btn btn-primary">SAVE</button>

        </form>

        </div>

    </div>

    </div>

    <script>

        function myFunction() {

          alert("Data Berhasil Di Edit");

        }

        </script>

@endsection

1. Lalu kita masuk ke file web.php dan memasukkan route untuk memanggil halaman edit data admin

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

Route::patch('/admins/{id}/update', 'AdminController@update');

1. Langkah berikut nya kita masuk ke AdminController.php dan memasukkan script berikut Pada public function edit dan public function update

    public function edit($id)

    {

        $admins = Admin::find($id);

        return view('admin.edit', compact('admins'));

    }

    public function update(Request $request, $id)

    {

        $admins = admin::find($id);

        $admins->id = $request->input('id');

        $admins->nm\_admin = $request->input('nm\_admin');

        $admins->callnumber = $request->input('callnumber');

        $admins->save();

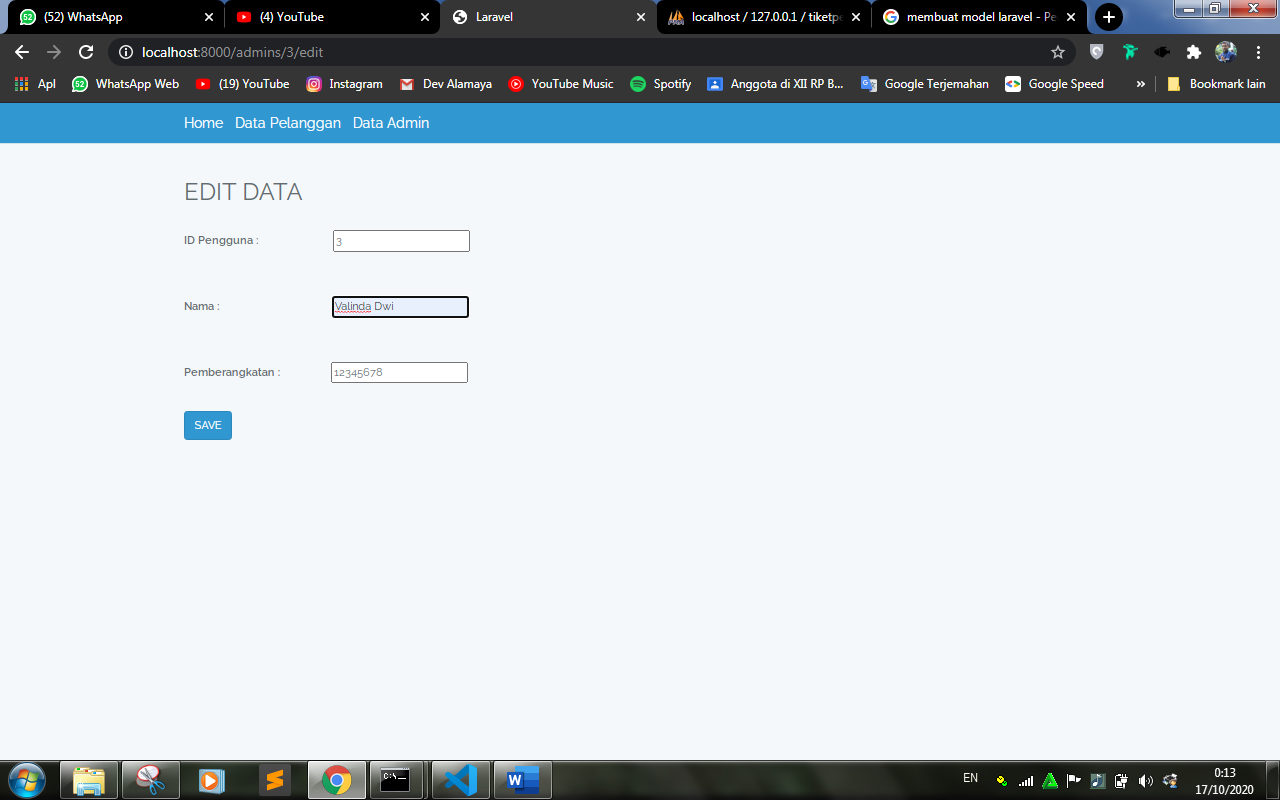
        return redirect('/admin');

    }

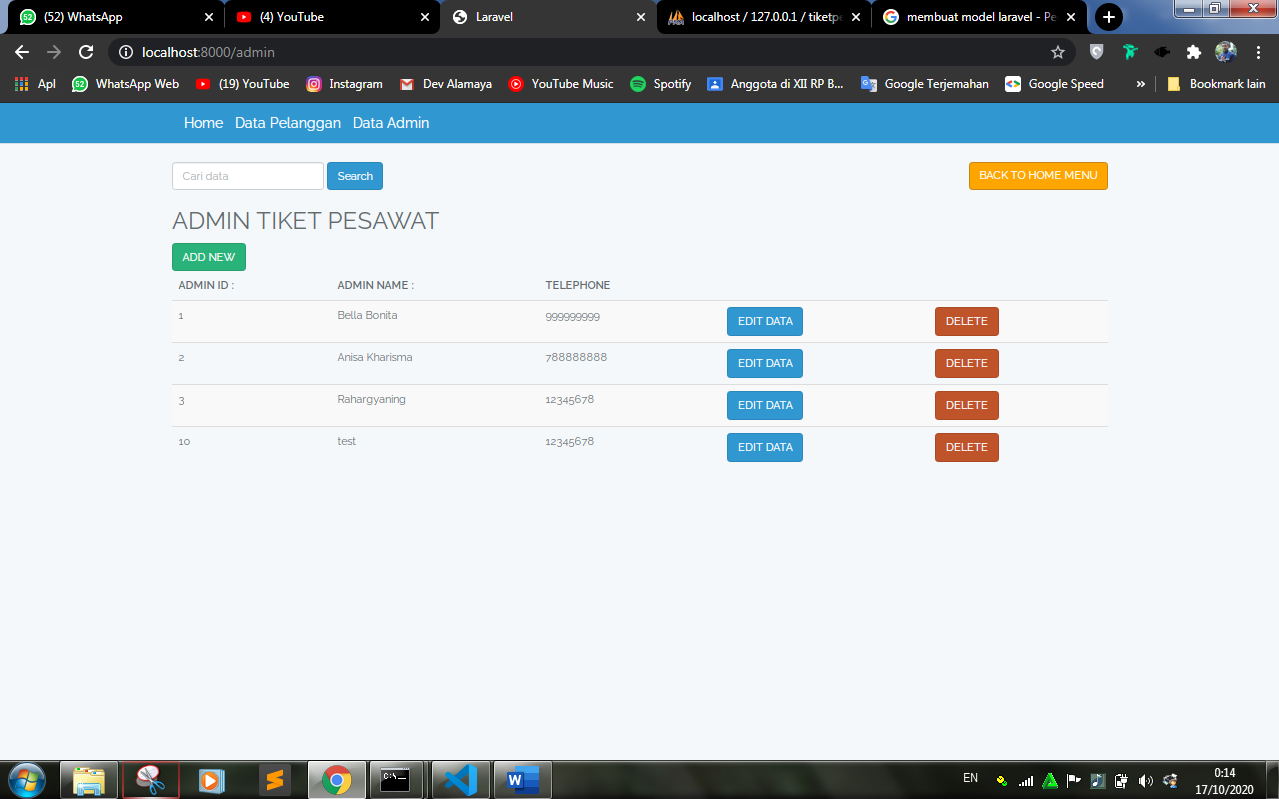
1. Langkah berikut nya adalah mengisi link pada button dengan script yang akan melakukan link menuju update form

<a class="btn btn-primary" href="{{ url("admins/{$admin->id}/edit") }}">EDIT DATA</a>

1. Mencoba meng edit data dengan mengganti nama admin dari Valinda menjadi Rahargyaning



1. Data berhasil di edit



**MEMBUAT DELETE DATA**

1. Langkah selanjutnya adalah membuat delete data, untuk membuat delete data ini pertama kita menuju AdminController.php dan memasukkan script pada public function destroy Untuk melakukan delete data admin

    public function destroy($id)

    {

        $admins = admin::find($id);

        $admins -> delete();

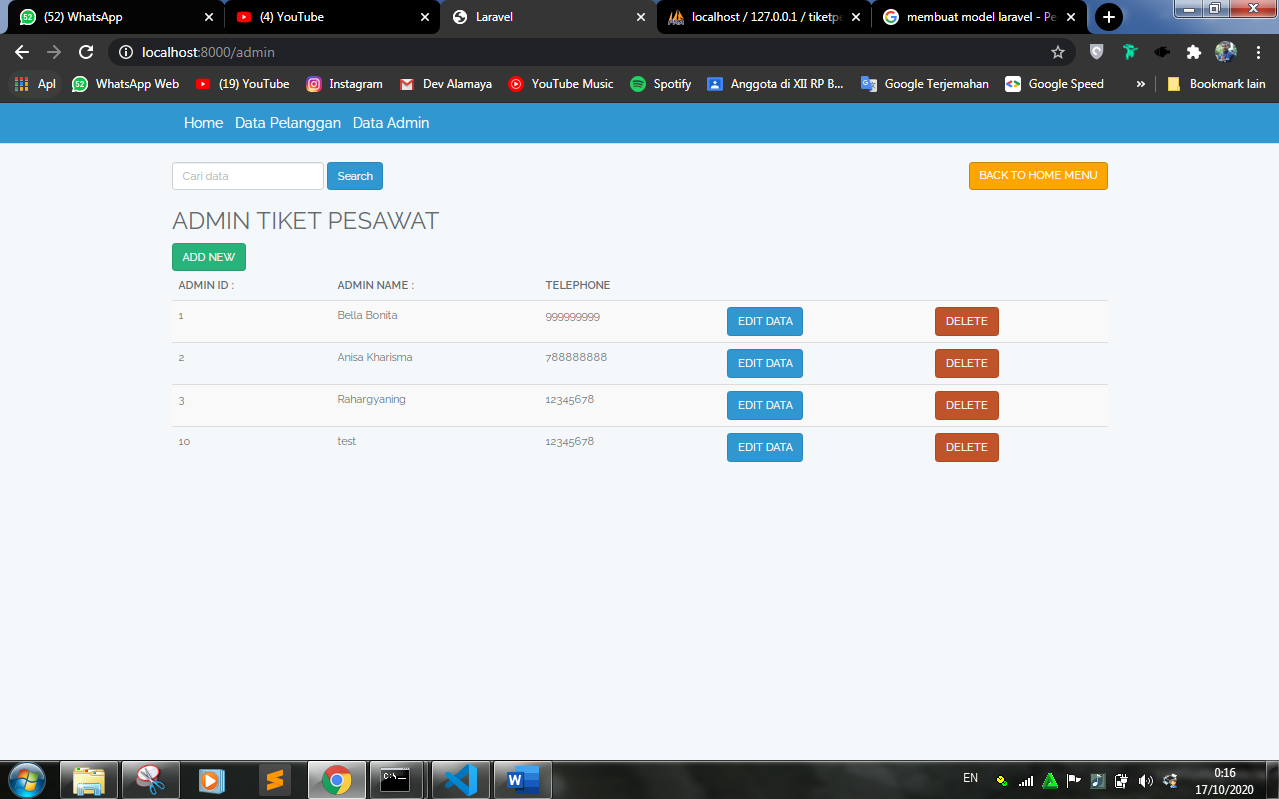
        return redirect('/admin');

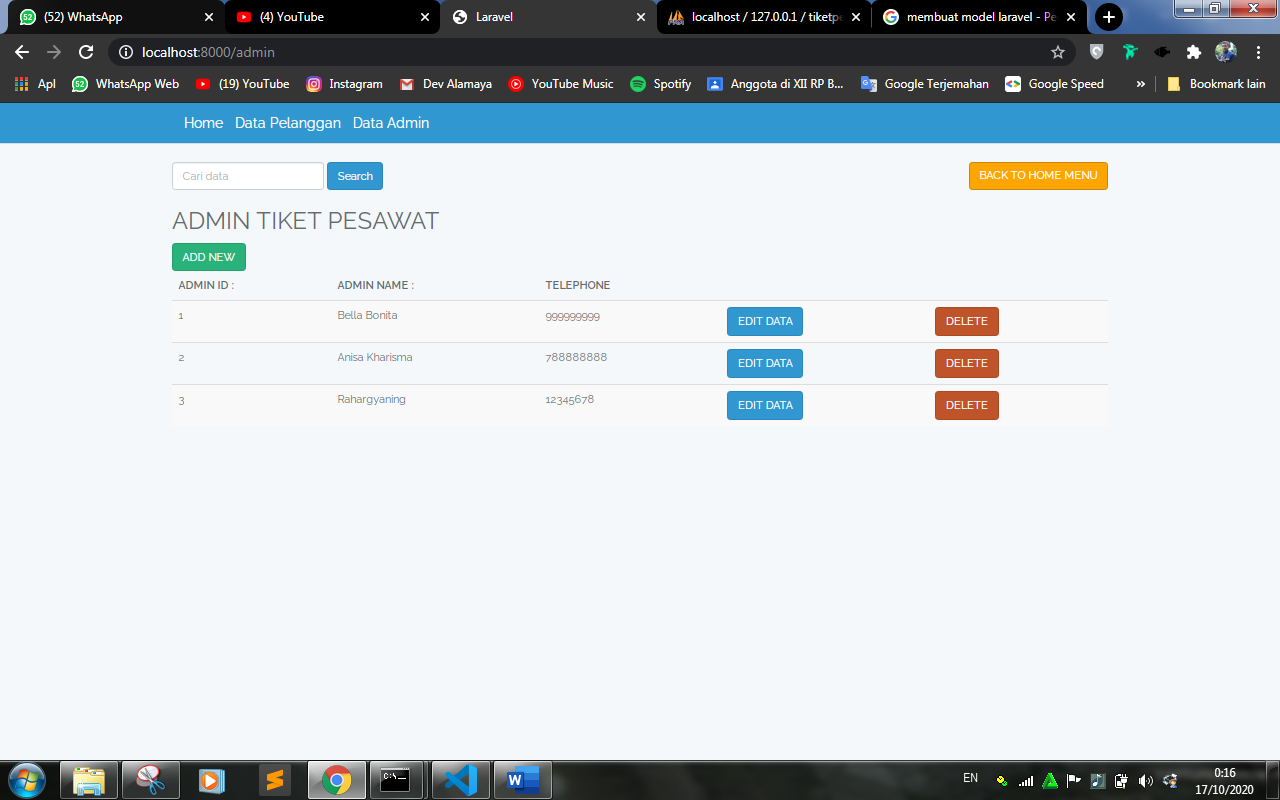
    }

1. Langkah selanjutnya adalah membuat route pada web.php

Route::delete('/admins/{id}', 'AdminController@destroy');

1. Maka tampilan delete akan seperti ini





**MEMBUAT CARI DATA**

1. Langkah selanjutnya adalah membuat cari data langkah pertama adalah membuat form dan juga name variable keyword

{{--  cari  --}}

    <form action="{{ url("/admin") }}" class="form-inline" method="GET">

        <div class="form-group mx-sm-3 mb-2">

        <input value="{{Request::get('keyword') }}" name="keyword" type="text" class="form-control" placeholder="Cari data">

        </div>

        <button type="submit" class="btn btn-primary mb-2">Search</button>

    </form>

{{--  /cari  --}}

2. Selanjutnya menuju ke AdminController.php dan memasukkan script pencarian data di function index dengan menambahkan parameter request dan if

        public function index(Request $request)

    {

        $keyword = $request->get('keyword');

        $admins = admin::all();

       if ($keyword) {

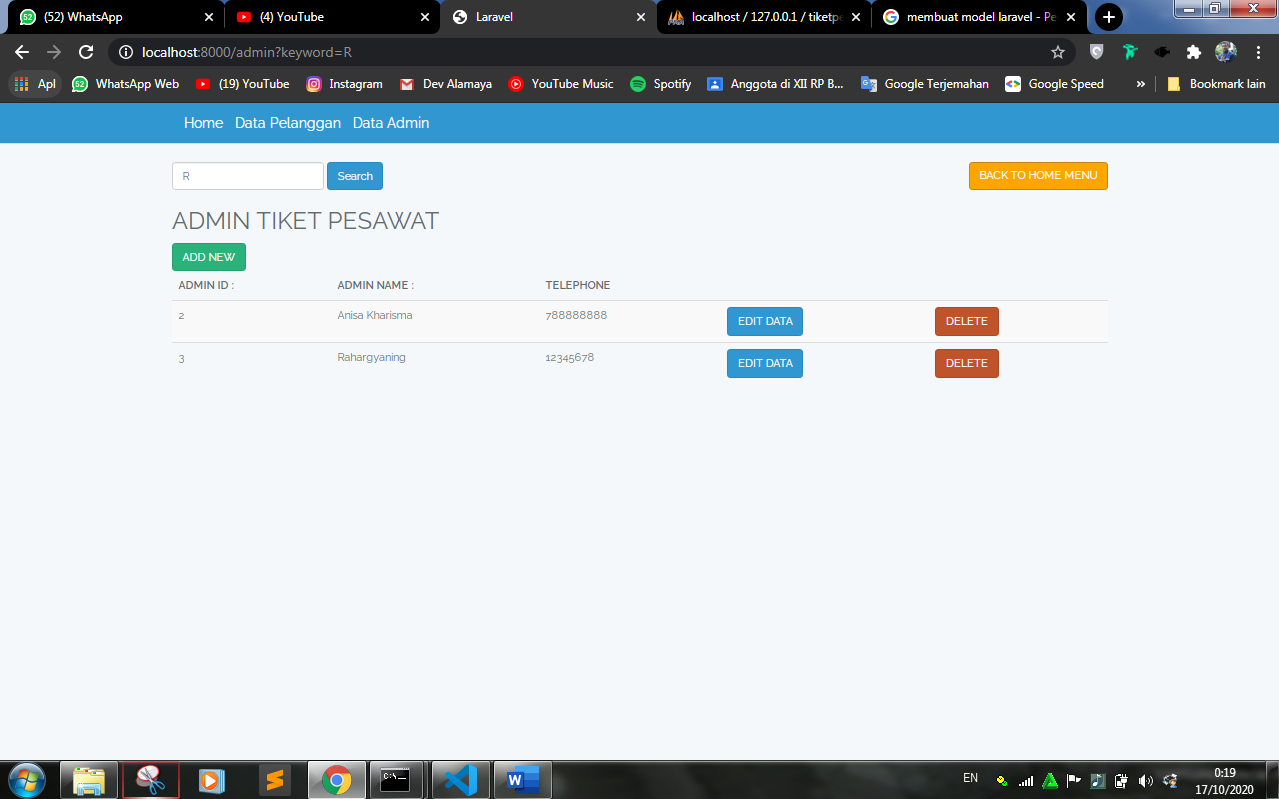
           $admins = admin::where("nm\_admin","LIKE","%$keyword%")->get();

       }

        return view('admin.admin', ['admins' => $admins]);

    }

3., Maka Tampilan pencarian Admin akan seperti berikut



Sistem pemesanan tiket penerbangan menggunakan database untuk menyimpan data-data pemesanan tiket pesawat secara komputerisasi sehingga kemungkinan terjadinya terulangnya data yang sama dalam sebuah data base sangat kecil. Dan juga dapat memudahkan pelanggan dalam hal memasukan data pemesanan dari manual ke komputerisasi sehingga pekerjaan yang sebelumnya lebih efisien dan terdata dengan baik, memudahkan pelanggan dalam mengatur system pemesanan. Rancang bangun aplikasi pemesanan tiket penerbangan belum maksimal sehingga kedepannya dapat dikembangan lagi dengan berbasis web.

KELOMPOK 2 :

* Bagus Kurniawan Hafizh (Programmer, Laporan, Design Logo)
* Azizah Senja Mega Sari (Upload File)
* Gusti Aji Rizky (Laporan)
* Rafif Pradyatma Surya Wahyudi (Programmer)