USER MANUAL

Perjalanan: Aplikasi Antar Jemput Pengguna Berbasis Website

Arya Pangestu (1301190144)

Zahra Fadiah Putri (1301194212)

Diaz Tiyasya Putra (1301194120)

Syafiq Muhammad Arrazzak (1301190377)

A. Deskripsi Produk

Pada revolusi industri 4.0 seperti sekarang ini dunia digital akan semakin memiliki pengaruh yang sangat besar bagi masyarakat luas dalam hal memenuhi kebutuhan sehari-hari contohnya seperti kebutuhan untuk berpergian kemanapun yang dibutuhkan. Maka dari itu dalam hal untuk memenuhi kebutuhan tersebut. Aplikasi Perjalanan hadir untuk membantu pengguna dalam menemukan alternatif untuk mengantarkan pengguna ke mana pun yang pengguna mau.

Perangkat lunak kami diberi nama Perjalanan ini yang diharapkan mampu diterima di kalangan masyarakat luas yang membutuhkan jasa transportasi. Aplikasi ini diharapkan mampu menghubungkan *driver* dengan penumpangnya sehingga membantu masyarakat untuk berpergian menggunakan jasa driver online. Aplikasi ini dapat meringankan pekerjaan *driver* untuk mencari penumpang dan juga sebaliknya penumpang juga dengan mudah mencari *driver* untuk bepergian kemanapun.

Website **Perjalanan** memiliki beberapa fitur yaitu:

a. Fitur Registrasi (Driver)

   Pengguna harus registrasi terlebih dahulu jika belum memiliki akun, dengan memasukkan username, email, password, plat nomor, jenis kendaraan, model kendaraan.

b. Fitur Registrasi (Passenger)

   Pengguna harus registrasi terlebih dahulu jika belum memiliki akun, dengan memasukkan username, email dan password.

c. Fitur Login (Semua Role)

   Fitur ini adalah fitur yang digunakan agar pengguna bisa masuk ke halaman beranda. Jika pengguna sudah memiliki akun maka bisa langsung login dengan memasukkan username dan password.

d. Fitur Lihat Dashboard (Semua Role)

   Fitur ini adalah fitur yang dimiliki oleh semua role. Fitur ini berguna untuk para pengguna melihat halaman dashboard mereka setelah melakukan login sesuai dengan role mereka yang dimana pada dashboard ini mereka dapat melihat seluruh fitur lain pada role mereka.

e. Fitur Lihat User Detail(Admin)

   Role Admin dapat melihat detail para pengguna lain seperti data diri, perjalanan yang sudah dilakukan, dan lain lain. Tetapi fitur ini tidak berlaku untuk sesama

role Admin

f. Fitur Edit User Status(Admin)

   Admin dapat mengaktifkan dan menonaktifkan status dari pengguna lain. fitur ini memungkinkan untuk admin seperti memblok akses penggunaan dari pengguna yang menurut admin mengganggu atau melakukan kesalahan.

g. Fitur Ride Status (Driver)

   Driver dapat mengaktifkan dan menonaktifkan status berkendaranya sesuai dengan kebutuhan dari driver itu sendiri, jika ingin mendapatkan pesanan maka fitur ini diaktifkan dan jika tidak ingin mendapat pesanan maka fitur ini dapat dinonaktifkan.

h. Fitur Memilih Kendaraan (Passenger)

   Passenger dapat memilih jenis kendaraan yang ingin mereka pesan (motor/mobil). Jika mereka memilih motor maka pesanan akan masuk kedalam list pesanan driver yang menggunakan kendaraan motor.

i. Fitur Memilih Lokasi (Passenger)

   Setelah memilih kendaraan Passenger diminta untuk memasukan lokasi mereka berupa titik jemput dan titik tujuan mereka.

j. Fitur Perjalanan (Passenger & Driver)

   Passenger dapat melakukan fitur ini setelah mereka memasukkan jenis kendaraan dan lokasi mereka. fitur ini akan menampilkan jarak, waktu tempuh, dan biaya yang harus dikeluarkan oleh passenger. Dan sebaliknya Driver dapat menggunakan fitur ini jika sudah mendapat pesanan dari Passenger.

k. Fitur Riwayat Perjalanan (Passenger & Driver)

   Setelah Melakukan Perjalanan Passenger dan Driver dapat melihat detail dari Perjalanan yang sudah dilakukan seperti lokasi, biaya yang dikeluarkan, dan lain lain.

l. Fitur Menambah Ulasan (Passenger)

   Jika Passenger sudah melakukan perjalanan mereka dapat menambahkan ulasan kepada Driver atas perjalanan yang sudah dilakukan, apakah itu buruk atau tidaknya perjalanan yang sudah dilakukan.

m. Fitur Melihat Ulasan (Passenger & Driver)

   Jika Passenger sudah menambahkan ulasan, maka Passenger maupun Driver dapat melihat ulasan yang telah diberikan.

B. Source Code

a. Admin

1. AdminController

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\User;
use Illuminate\Http\Request;

class AdminController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman dashboard
admin
    public function index()
    {
        return view('admin.dashboard.index', [
            "title" => "Dashboard Admin",
            "total_passenger" => User::all()->where('role',
1)->count(),
            "total_driver" => User::all()->where('role',
2)->count(),
        ]);
    }
}
```

2. AdminDriverController

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use App\Models\User;
use App\Models\Driver;
use App\Models\Ride;

class AdminDriverController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman list/data
```

```
driver
    public function index()
    {
        return view('admin.dashboard.list-pengemudi', [
            "title" => "List-pengemudi Admin",
            "users" => User::all(),
            "rides" => Ride::all()
        ]);
    }

    // Method yang digunakan untuk halaman detail sesuai driver
yang dipilih
    public function showDetail($id)
    {
        return view('admin.dashboard.detailList-pengemudi', [
            "title" => "List-pengemudi Admin",
            "rides" => Ride::where('driver_id',
$id)->where('status', 1)->get()
        ]);
    }

    // Method yang digunakan untuk halaman view map sesuai detail
yang dipilih
    public function showView($id)
    {
        return view('admin.dashboard.viewList-pengemudi', [
            "title" => "List-pengemudi Admin",
            "ride" => Ride::where('id', $id)->first()
        ]);
    }

    // Method yang digunakan untuk mengubah status driver
    public function updateDriverStatus($id)
    {
        if (User::where('id', $id)->first()->status === 1) {
            $update = array('status' => 0);
        } else {
            $update = array('status' => 1);
        }
        User::where('id', $id)->update($update);
        return;
    }
```

```
}
```

3. AdminPassengerController

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Controllers\Controller;
use App\Models\Passenger;
use App\Models\Ride;
use App\Models\User;
use Illuminate\Http\Request;

class AdminPassengerController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman list/data
penumpang
    public function index()
    {
        return view('admin.dashboard.list-penumpang', [
            "title" => "List-penumpang Admin",
            "users" => User::all(),
            "rides" => Ride::all()
        ]);
    }

    // Method yang digunakan untuk halaman detail sesuai penumpang
yang dipilih
    public function showDetail($id)
    {
        return view('admin.dashboard.detailList-penumpang', [
            "title" => "List-penumpang Admin",
            "rides" => Ride::where('passenger_id',
$id)->where('status', 1)->get()
        ]);
    }

    // Method yang digunakan untuk halaman view map sesuai detail
yang dipilih
    public function showView($id)
    {
        return view('admin.dashboard.viewList-penumpang', [
```

```
            "title" => "List-penumpang Admin",
            "ride" => Ride::where('id', $id)->first()
        ]);
    }

    // Method yang digunakan untuk mengubah status penumpang
    public function updatepassengerStatus($id)
    {
        if (User::where('id', $id)->first()->status === 1) {
            $update = array('status' => 0);
        } else {
            $update = array('status' => 1);
        }
        User::where('id', $id)->update($update);
        return;
    }
}
```

b. Driver

1. DriverController

```
<?php

namespace App\Http\Controllers\Driver;

use App\Http\Controllers\Controller;
use App\Models\Driver;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class DriverController extends Controller
{
    // Method yang digunakan untuk menampilkan dashboard driver
    public function index()
    {
        return view('driver.dashboard.index', [
            "title" => "Dashboard Driver",
            "driver" => Driver::where('user_id',
Auth::user()->id)->first(),
            "total_ride" => Ride::all()->where('driver_id',
Auth::user()->id)->count(),
```

```
        ]);
    }

    // Method yang digunakan untuk mengubah status ride driver
    public function updateDriverStatus($id)
    {
        if (Driver::where('user_id', $id)->first()->ride_status
=== 1) {
            $update = array('ride_status' => 0);
        } else {
            $update = array('ride_status' => 1);
        }
        Driver::where('user_id', $id)->update($update);
        return;
    }
}
```

2. DriverHistoryController

```
<?php

namespace App\Http\Controllers\Driver;

use App\Http\Controllers\Controller;
use App\Models\Review;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class DriverHistoryController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman riwayat
driver
    public function index()
    {
        return view('driver.dashboard.history', [
            "title" => "Riwayat Driver",
            "histories" => Ride::where('driver_id',
Auth::user()->id)
                ->where('status', 1)
                ->get()
        ]);
```

```
    }

    // Method yang digunakan untuk menampilkan halaman vie detail
dari riwayat yang dipilih
    public function showView($id)
    {
        return view('driver.dashboard.viewHistory', [
            "title" => "Riwayat Driver",
            "ride" => Ride::where('id', $id)->first()
        ]);
    }
}
```

3. DriverPerjalananController

```
<?php

namespace App\Http\Controllers\Driver;

use App\Http\Controllers\Controller;
use App\Models\Driver;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class DriverPerjalananController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman perjalanan
driver
    public function index()
    {
        return view('driver.dashboard.perjalanan', [
            "title" => "Perjalanan Driver",
            "rides" => Ride::where('vehicle_type',
Auth::user()->driver->vehicle->vehicle_type)
                ->where('status', 0)
                ->where('driver_id', null)
                ->get(),
            "perjalanan_ride" => Ride::where('vehicle_type',
Auth::user()->driver->vehicle->vehicle_type)
                ->where('status', 0)
                ->where('driver_id', Auth::user()->id)
                ->first(),
```

```
            "driver" => Driver::where('user_id',
Auth::user()->id)->first()
        ]);
    }

    // Method yang digunakan untuk menampilkan view map pada
perjalanan yang dipilih
    public function showView($id)
    {
        return view('driver.dashboard.viewPerjalanan', [
            "title" => "Perjalanan Driver",
            "ride" => Ride::where('id', $id)->first()
        ]);
    }

    // Method yang digunakan untuk mengubah status Perjalanan
berhasil diambil
    public function updateRideDriver($id)
    {
        $update = array('driver_id' => Auth::user()->id);
        Ride::where('id', $id)->update($update);
        return redirect('/driver/perjalanan')->with('alert',
'Perjalanan berhasil diambil');
    }

    // Method yang digunakan untuk mengubah status Selamat Anda
telah menyelesaikan perjalanan
    public function updateRideStatus($id)
    {
        $update = array('status' => 1);
        Ride::where('id', $id)->update($update);

        return redirect('/driver/perjalanan')->with('alert',
'Selamat Anda telah menyelesaikan perjalanan');
    }
}
```

c. Passenger

1. PassengerController

```
<?php
```

```php
namespace App\Http\Controllers\Passenger;

use App\Http\Controllers\Controller;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class PassengerController extends Controller
{
    // Method yang digunakan untuk halaman dashboard penumpang
    public function index()
    {
        return view('passenger.dashboard.index', [
            "title" => "Dashboard Passenger",
            "total_ride" => Ride::all()->where('passenger_id',
Auth::user()->id)->count(),
        ]);
    }
}
```

2. PassengerHistoryController

```php
<?php

namespace App\Http\Controllers\Passenger;

use App\Http\Controllers\Controller;
use App\Models\Review;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class PassengerHistoryController extends Controller
{
    // Method yang digunakan untuk menampilkan halamana riwayat
penumpang
    public function index()
    {
        return view('passenger.dashboard.history', [
            "title" => "Riwayat Passenger",
            "histories" => Ride::where('passenger_id',
Auth::user()->id)
                ->where('status', 1)
```

```
                ->get()
        ]);
    }

    // Method yang digunakan untuk view map pada riwayat yang
telah dipilih
    public function showView($id)
    {
        return view('passenger.dashboard.viewHistory', [
            "title" => "Riwayat Passenger",
            "ride" => Ride::where('id', $id)->first()
        ]);
    }

    // Method yang digunakan untuk memasukan ulasan pada table
reviews sesuai riwayat yang telah dipilih
    public function storeUlasan(Request $request, $id)
    {
        $validated = $request->validate([
            'rate' => 'required',
            'review' => 'required',
        ]);
        $review = Review::create($validated);

        $temp['review_id'] = $review->id;
        Ride::where('id', $id)->update($temp);
        return redirect('/passenger/history')->with('alert',
'Ulasan berhasil ditambahkan!');
    }
}
```

3. PassengerPemesananController

```
<?php

namespace App\Http\Controllers\Passenger;

use App\Http\Controllers\Controller;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class PassengerPemesananController extends Controller
```

```
{
    // Method yang digunakan untuk menampilkan halaman pemesanan
penumpang
    public function index()
    {
        if (Ride::where('passenger_id', Auth::user()->id)->first()
== null) {
            $temp =  -1;
        } else {
            $temp =  Ride::where('passenger_id',
Auth::user()->id)->first()->value('status');
        }
        return view('passenger.dashboard.pemesanan', [
            "title" => "Pemesanan Passenger",
            "status" => $temp,

        ]);
    }

    // Method yang digunakan untuk menyimpan posisi titik jemput
dan tujuan serta jenis kendaraan ke dalam table ride
    public function store(Request $request)
    {
        $validated = $request->validate([
            'vehicle_type' => 'required',
            'pick_up_form_latitude' => 'required',
            'pick_up_form_longitude' => 'required',
            'drop_to_latitude' => 'required',
            'drop_to_longitude' => 'required',
            'amount' => 'required',
        ]);
        $validated['passenger_id'] = Auth::user()->id;

        Ride::create($validated);

        return redirect('/passenger/perjalanan')->with('alert',
'Pesanan berhasil ditambahkan, Silakan tunggu driver Anda');
    }
}
```

4. PassengerPerjalananController

```
<?php
```

```php
namespace App\Http\Controllers\Passenger;

use App\Http\Controllers\Controller;
use App\Models\Ride;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class PassengerPerjalananController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman perjalanan penumpang
    public function index()
    {
        if (
            Ride::where('passenger_id', Auth::user()->id)
            ->where('status', 0)
            ->first() == null
        ) {
            $temp =  null;
        } else {
            $temp =  Ride::where('passenger_id', Auth::user()->id)
                ->where('status', 0)
                ->first();
        }
        return view('passenger.dashboard.perjalanan', [
            "title" => "Perjalanan Passenger",
            "ride" => $temp
        ]);
    }
}
```

D. HomeController

```php
<?php

namespace App\Http\Controllers;

use App\Models\Driver;
use App\Models\Ride;
use App\Models\User;
```

```php
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;

class HomeController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman dashboard
sesuai dengan role
    public function index()
    {
        if (Auth::user()->role == 0) {
            return view('admin.dashboard.index', [
                "title" => "Dashboard Admin",
                "total_passenger" => User::all()->where('role',
1)->count(),
                "total_driver" => User::all()->where('role',
2)->count(),
            ]);
        }

        if (Auth::user()->role == 1) {
            return view('passenger.dashboard.index', [
                "title" => "Dashboard Passenger",
                "total_ride" => Ride::all()->where('passenger_id',
Auth::user()->id)->count(),
            ]);
        }

        if (Auth::user()->role == 2) {
            return view('driver.dashboard.index', [
                "title" => "Dashboard Driver",
                "driver" => Driver::where('user_id',
Auth::user()->id)->first(),
                "total_ride" => Ride::all()->where('driver_id',
Auth::user()->id)->count(),
            ]);
        }
    }
}
```

E. LoginController

```php
<?php
```

```php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

//  Import Auth facade
use Illuminate\Support\Facades\Auth;

class LoginController extends Controller
{
    // Method yang digunakan untuk menampilkan halaman login
    public function index()
    {
        return view('Login.index',  [
            "title" => "login"
        ]);
    }

    // Method yang digunakan untuk melakukan login atau melakukan
Authenticating Users
    public function authenticate(Request $request)
    {
        // Melakukan validasi email dan password
        $credentials = $request->validate([
            // email harus terisi dan formatnya email
            'email' => ['required', 'email'],
            // password harus terisi
            'password' => ['required'],
        ]);

        // Digunakan untuk menangani upaya autentikasi dari
formulir "login" aplikasi
        if (Auth::attempt($credentials)) {
            // Membuat ulang sesi pengguna untuk mencegah fiksasi
sesi
            $request->session()->regenerate();

            // Simpan nama route sesuai role user
            if (Auth::user()->role == 0 && Auth::user()->status ==
1) {
                $temp = '/dashboard';
            } else if (Auth::user()->role == 1 &&
Auth::user()->status == 1) {
```

```
                $temp = '/passenger';
            } else if (Auth::user()->role == 2 &&
Auth::user()->status == 1) {
                $temp = '/driver';
            } else {
                $temp = '';
            }

            if ($temp != '') {
                // Redirect route sesuai role user
                return redirect($temp);
            } else {
                // jika role tidak ada maka akan logout untuk
menghancurkan session
                LoginController::_logout($request);
            }
        }

        // jika login gagal/user atau password user tidak tersedia
maka akan kembali/back view
        // dengan mengirim pesan alert loginError yang berisi
Login failed!
        return back()->with('loginError', 'Login failed!');
    }


    // Method yang digunakan untuk mengeluarkan pengguna dari
aplikasi dengan memanggil _logout
    public function logout(Request $request)
    {
        // Ini akan menghapus informasi otentikasi dari sesi
pengguna sehingga permintaan berikutnya tidak diautentikasi.
        LoginController::_logout($request);

        // Setelah mengeluarkan pengguna maka akan mengarahkan
pengguna ke root aplikasi
        return redirect('/');
    }

    // Method yang digunakan untuk mengeluarkan pengguna dari
aplikasi
    private function _logout(Request $request)
```

```
    {
        // Ini akan menghapus informasi otentikasi dari sesi
pengguna sehingga permintaan berikutnya tidak diautentikasi.
        Auth::logout();

        // invalidate sesi pengguna
        $request->session()->invalidate();

        // membuat ulang token CSRF user
        $request->session()->regenerateToken();
    }
}
```

F. RegisterController

```
<?php

namespace App\Http\Controllers;

use App\Models\Driver;

use App\Models\Passenger;

use Illuminate\Http\Request;

use App\Models\User;

use App\Models\Vehicle;

use Illuminate\Support\Facades\Hash;

class RegisterController extends Controller

{

   // Method yang digunakan untuk menampilkan halaman registrasi

   public function index()

   {

       return view('register.index',  [
```

```
        "title" => "register"

    ]);

}



// Method yang digunakan untuk menyimpan data ke table users

public function storePassenger(Request $request)

{
    $validated = $request->validate([
        'name' => 'required|max:255',
        'email' => 'required|email|unique:users',
        'password' => 'required',
        'terms' => 'required',
    ]);
    $validated['password']=Hash::make($validated['password']);

    $validated['role'] = 1;

    $user = User::create($validated);

    Passenger::create(['user_id' => $user->id]);


    return redirect('/login');

}

// Method yang digunakan untuk menyimpan data ke table drivers

public function storeDriver(Request $request)

{
    $validated = $request->validate([

        'name' => 'required|max:255',
```

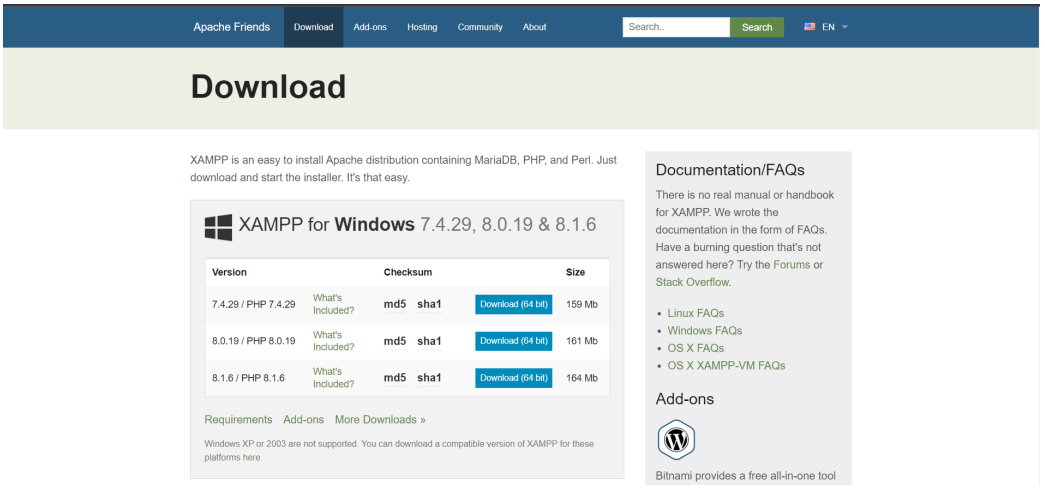
```
            'email' => 'required|email|unique:users',
            'password' => 'required',
            'terms' => 'required',
            'vehicle_type' => 'required',
            'model' => 'required',
            'plat' => 'required',
        ]);

        $validated['password']=Hash::make($validated['password']);


        $datVehicle['vehicle_type'] = $validated['vehicle_type'];

        unset($validated['vehicle_type']);


        $datVehicle['model'] = $validated['model'];

        unset($validated['model']);


        $datVehicle['plat'] = $validated['plat'];

        unset($validated['plat']);


        $vehicle = Vehicle::create($datVehicle);


        $validated['role'] = 2;

        $user = User::create($validated);

        Driver::create(['user_id' => $user->id, 'vehicle_id' =>
$vehic
```

C. Instalasi lokal

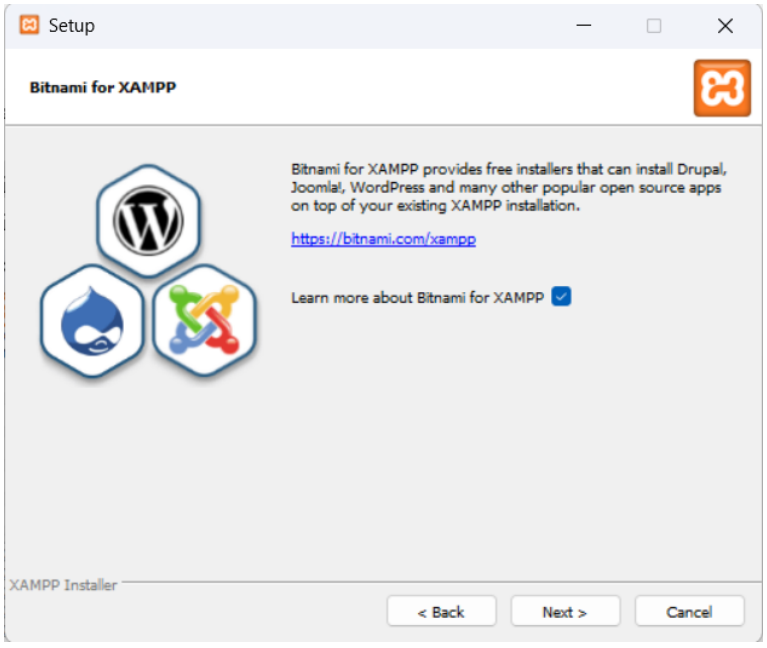
A. Set Up XAMPP

1. Download XAMPP

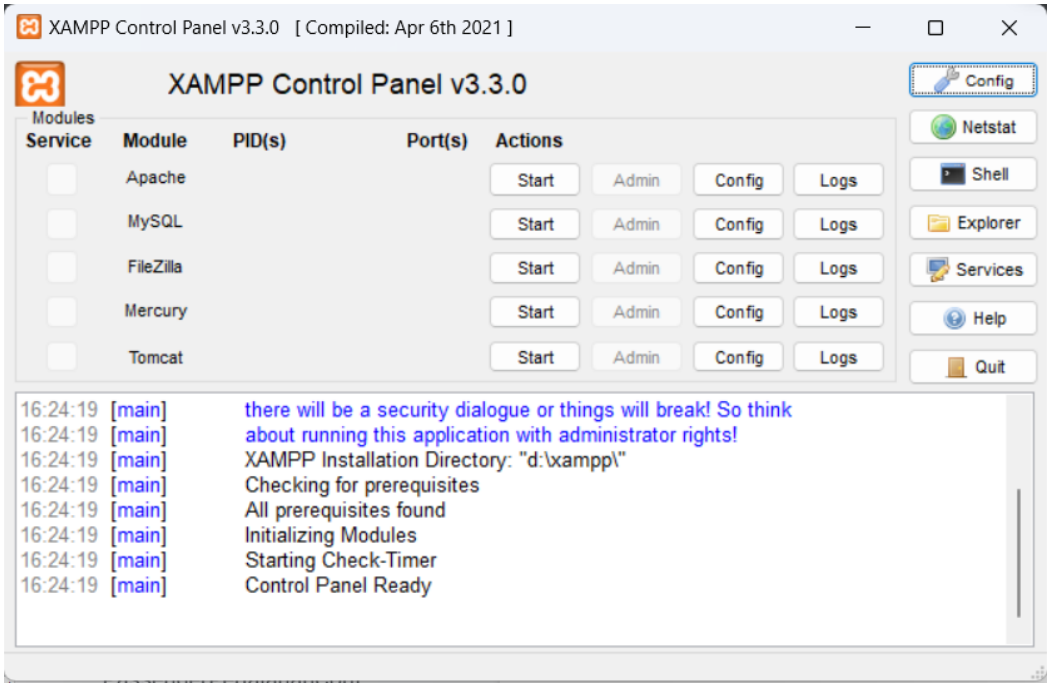
   XAMPP dapat diunduh pada link berikut ini namun jangan lupa sesuaikan dengan OS yang digunakan dan download versi terbaru: https://www.apachefriends.org/download.html



2. Install XAMPP



klik Next, kemudian tunggu sampai instalasi selesai dan klik finish. Berikut tampilan awal saat XAMPP dijalankan



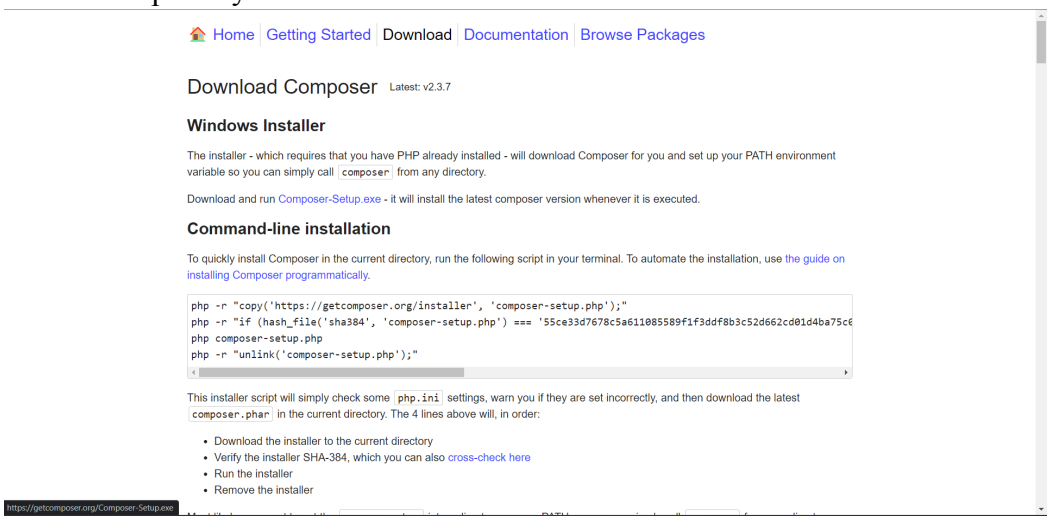
B. Set Up Composer

1. Download Composer

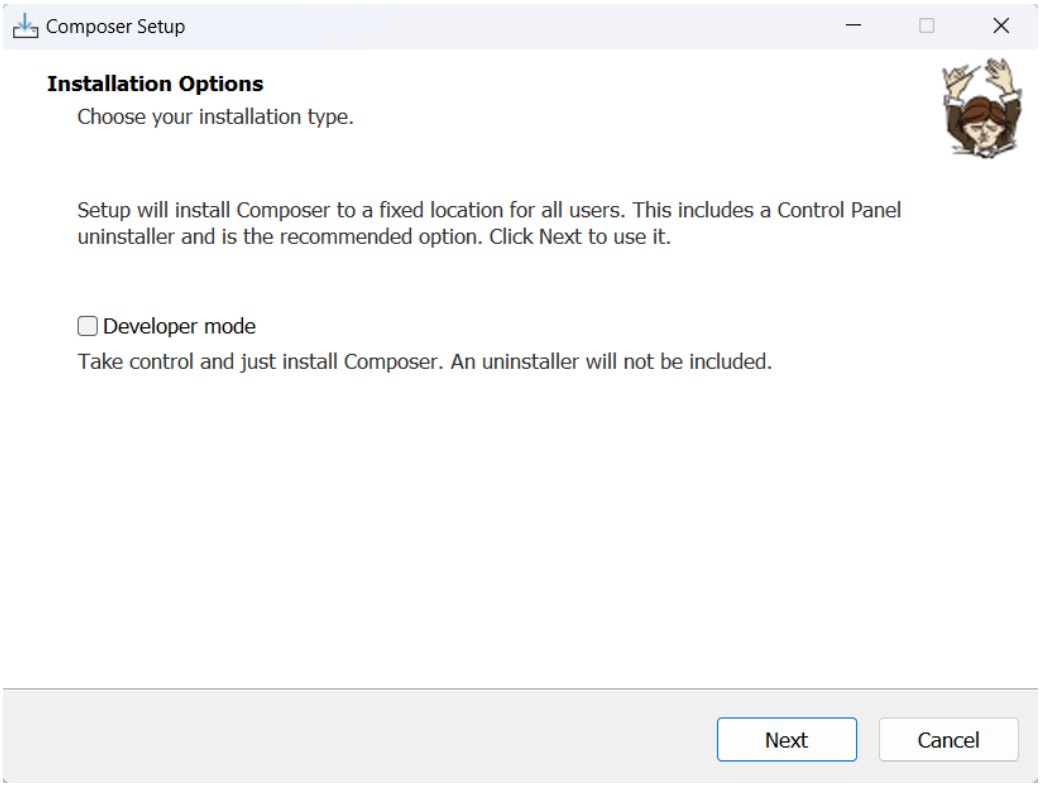
   Composer dapat didownload pada link berikut ini namun jangan lupa sesuaikan dengan OS yang digunakan :

   https://getcomposer.org/download/

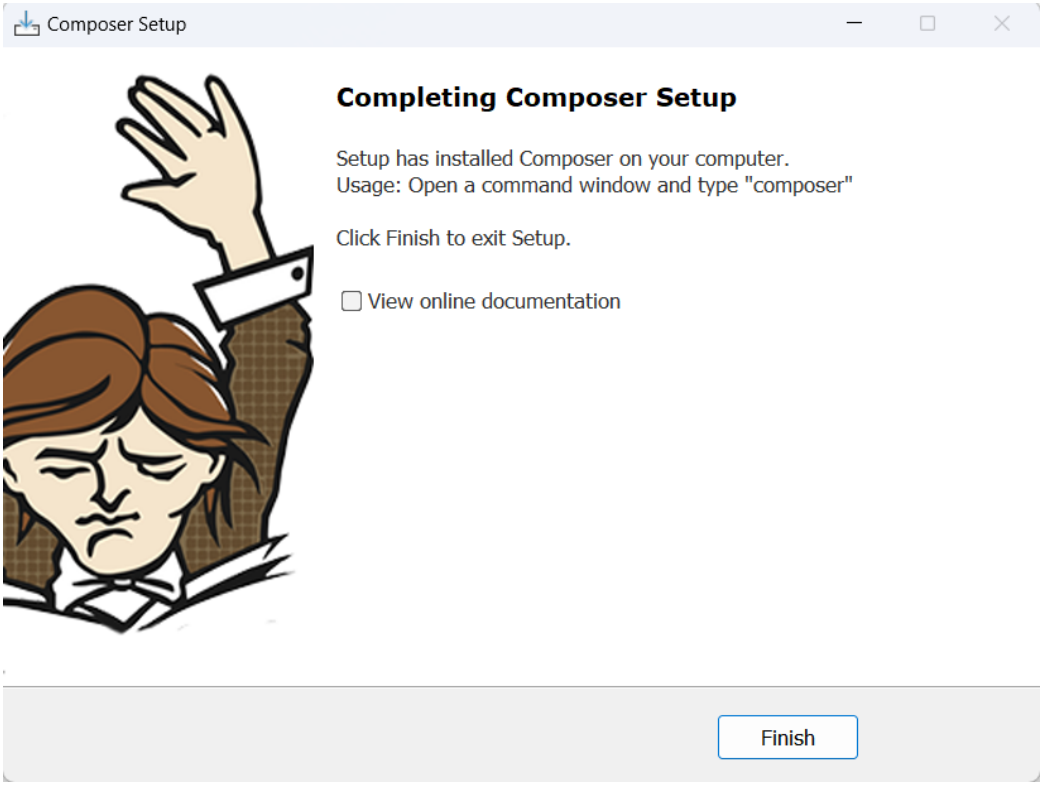
   Berikut tampilannya:



2. Install Composer



klik Next, kemudian pilih versi php, kemudian tunggu sampai instalasi selesai. Berikut tampilan saat composer berhasil diinstall, lalu klik finish:



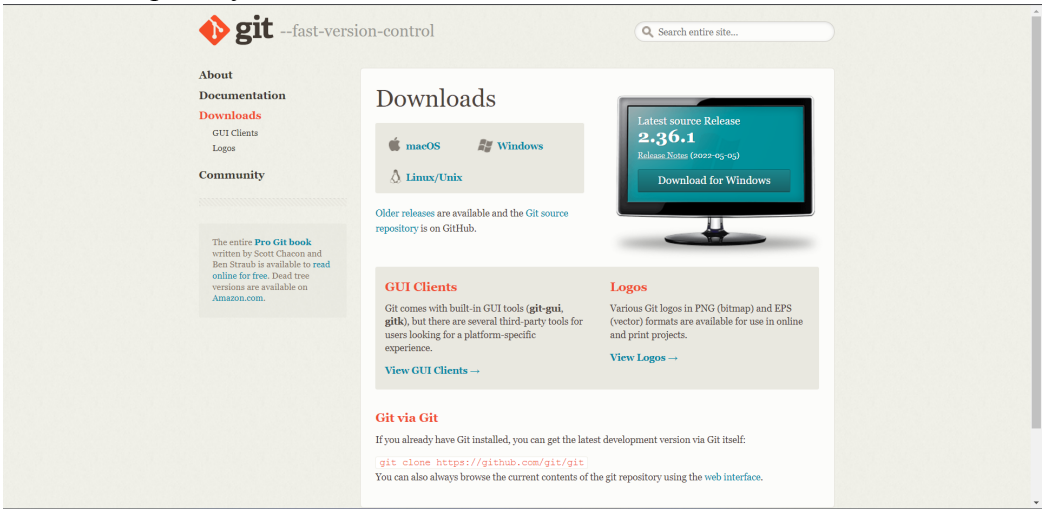
C. Set Up Git

1. Download Git

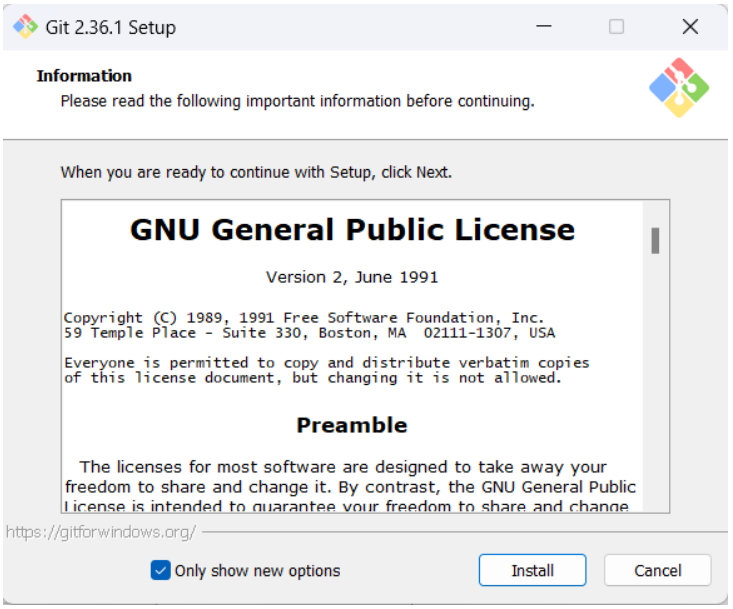
Git dapat didownload pada link berikut ini namun jangan lupa sesuaikan dengan OS yang digunakan :

https://git-scm.com/downloads

Berikut tampilannya:

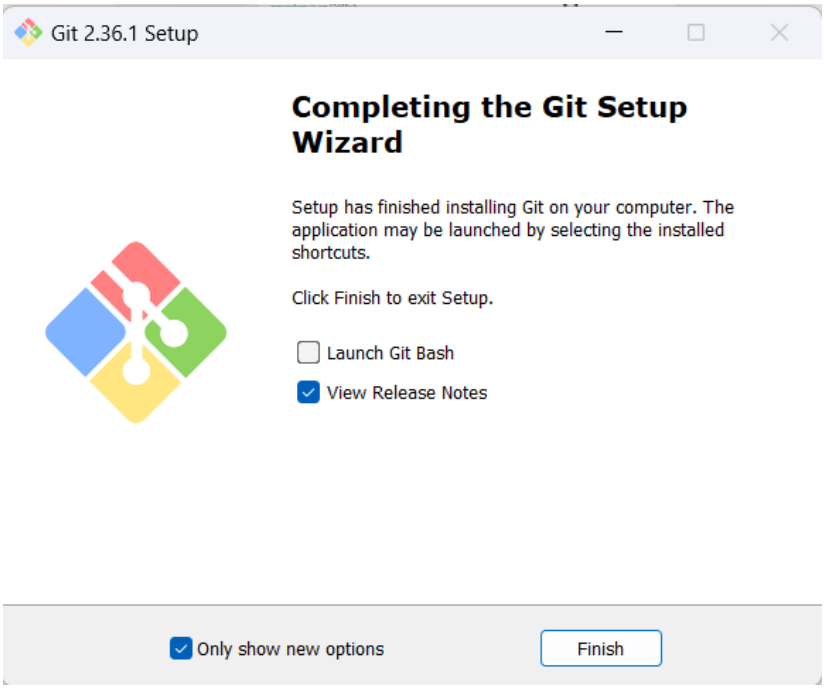


2. Install Git



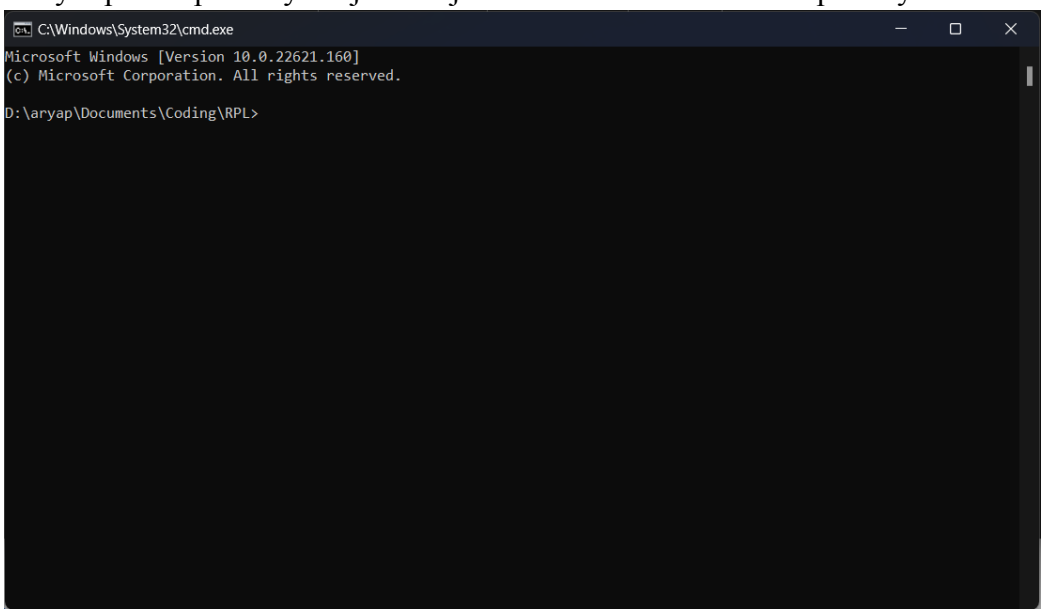
klikInstall, kemudian tunggu sampai instalasi selesai.

Berikut tampilan saat Git berhasil diinstall, lalu klik finish:



D. Jalankan Project Perjalanan Laravel

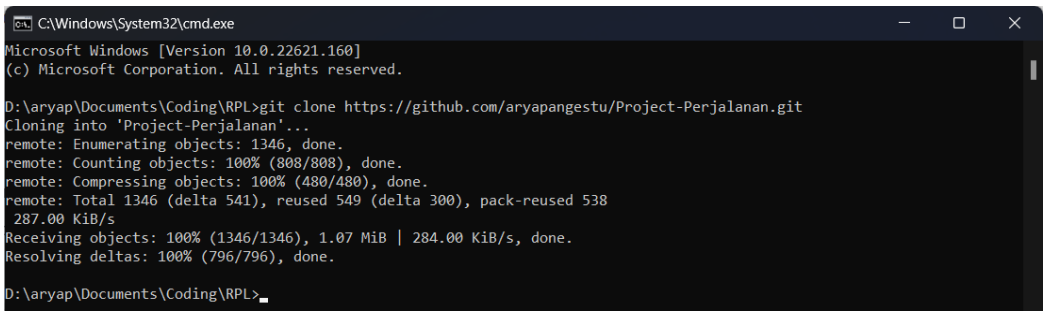
1. Buka CMD, ganti Directory menjadi Directory yang akan digunakan untuk menyimpan Repository Project-Perjalanan Laravel. Berikut tampilannya:



2. Clone the github repo dengan masukkan perintah berikut pada CMD:

```
git clone https://github.com/aryapangestu/Project-Perjalanan.git
```

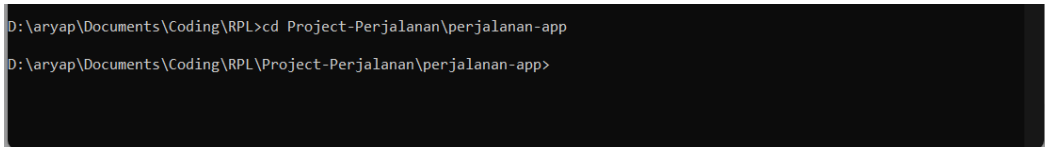
Berikut tampilannya:



3. Masuk direktori project perjalanan-app, dengan perintah berikut:

```
cd Project-Perjalanan\perjalanan-app
```

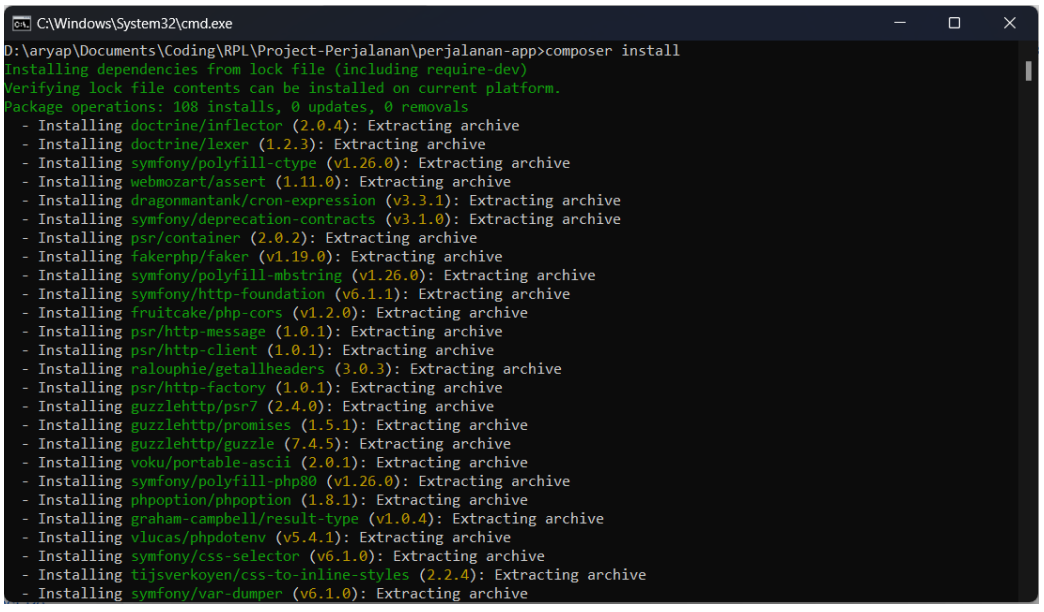
Berikut tampilannya:



4. Instal project dependencies:

```
composer install
```

Berikut tampilannya:



5. Salin .env.example ke .env atau cukup ganti namanya:

If linux:

```
cp .env.example .env
```

If Windows:

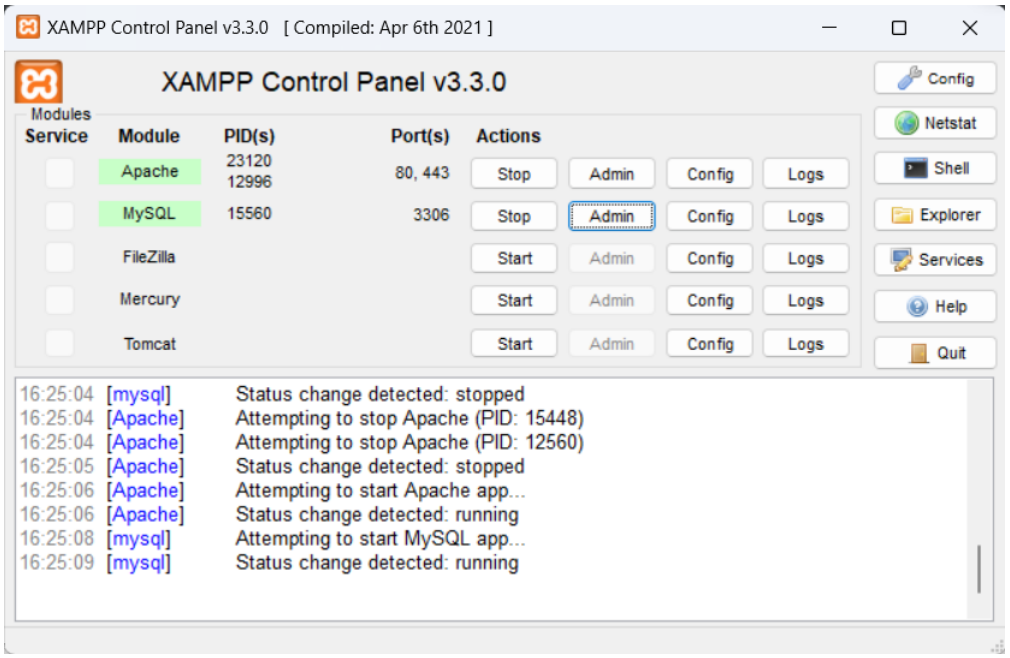
```
copy .env.example .env
```

Berikut tampilannya:

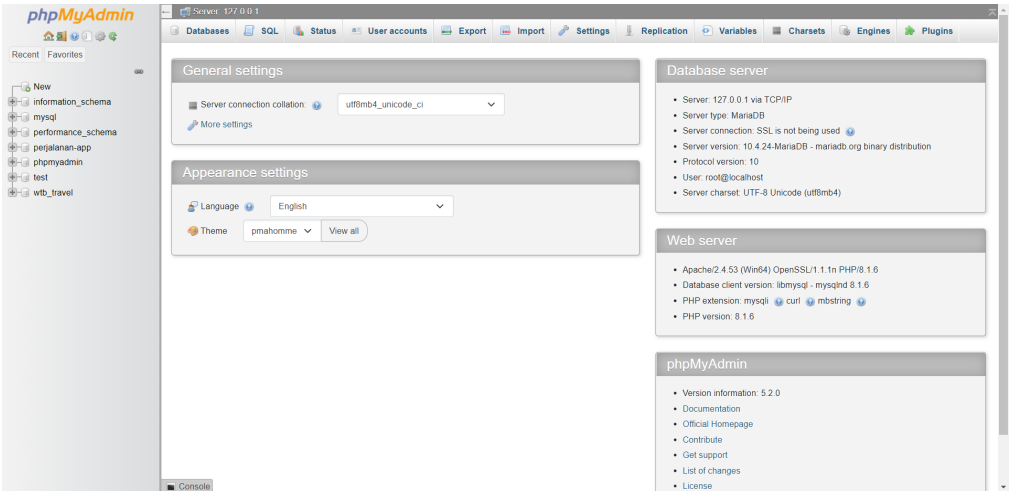
```
D:\aryap\Documents\Coding\RPL\Project-Perjalanan\perjalanan-app>copy .env.example .env
        1 file(s) copied.

D:\aryap\Documents\Coding\RPL\Project-Perjalanan\perjalanan-app>
```

6. Buat database baru pada MySQL

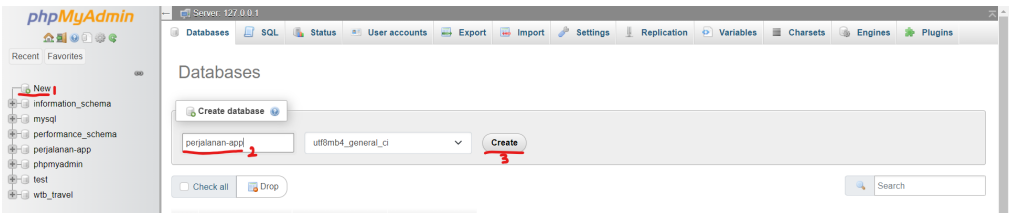
   a. Nyalakan Xampp module Apache dan MySQL



   b. Kemudian klik Admin pada MySQL, maka akan menuju ke http://localhost/phpmyadmin/



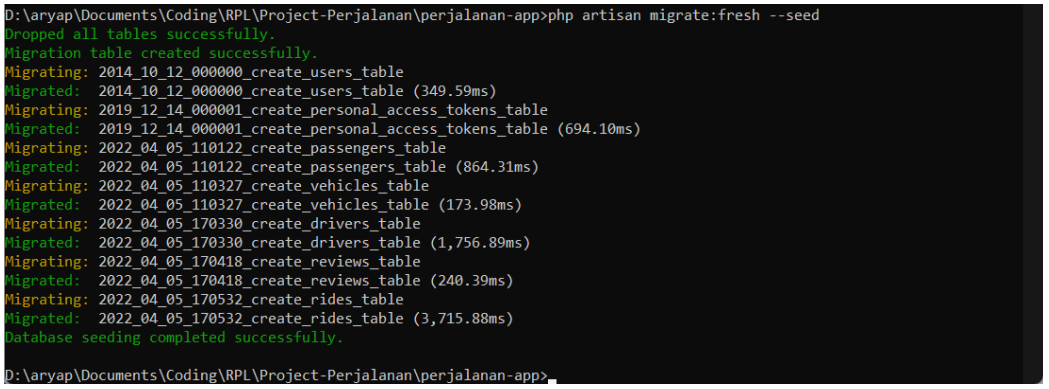
   c. Klik new, kemudian buat empty Database dengan nama perjalanan-app



7. Buat tabel ke dalam database menggunakan migrasi Laravel dan seeder:

```
php artisan migrate:fresh --seed
```

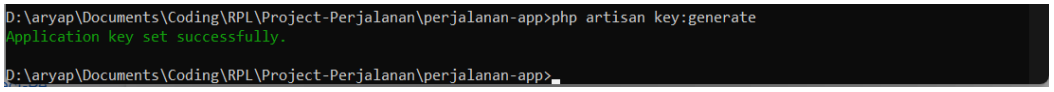
Berikut tampilannya:



8. Buat application key:

```
php artisan key:generate
```

Berikut tampilannya:



9. Mulai server laravel:

```
php artisan serve
```

Jika css/js tidak berkerja:

```
php -S localhost:8000 -t public
```

Berikut tampilannya:

10. Buka web Perjalanan dengan url yang telah tertulis di CMD:

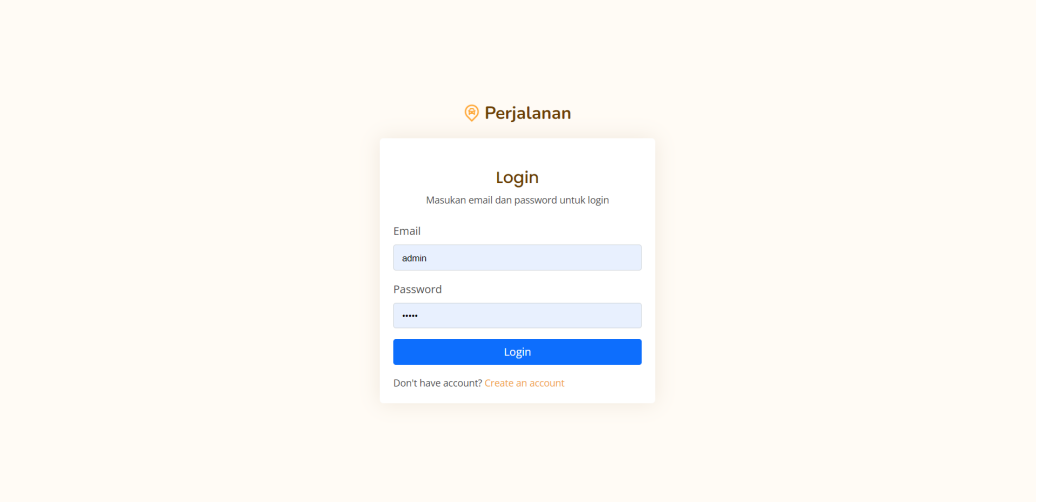
    Jika menggukan `php artisan serve:`

    http://127.0.0.1:8000

    Jika menggukan `php -S localhost:8000 -t public:`

    http://localhost:8000

    Berikut tampilannya:



11. Login atau buat akun

    Akun test:

    a. Admin

       email: testadmin@example.com

       password: password

    b. Passenger

       email: testpassenger@example.com

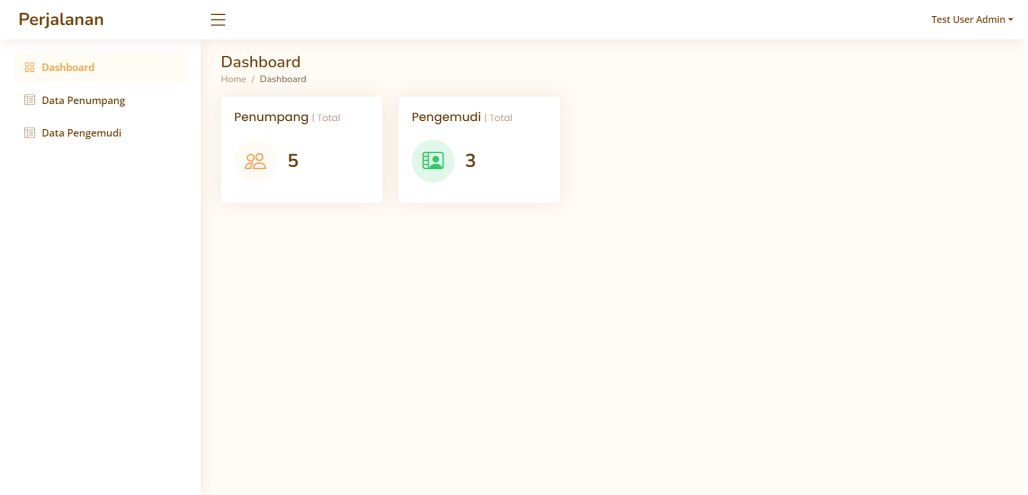
       password: password

    c. Driver

       email: testdriver@example.com

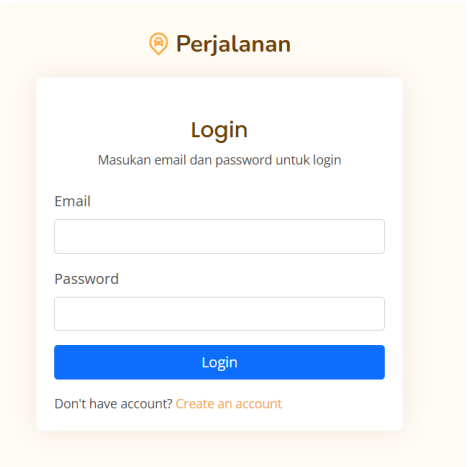
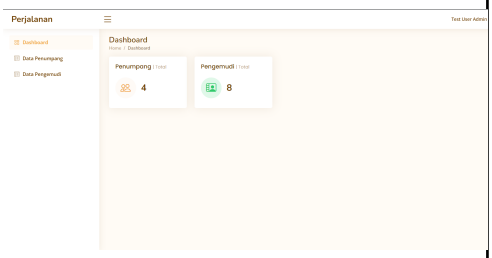
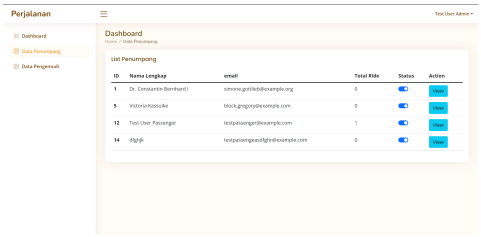
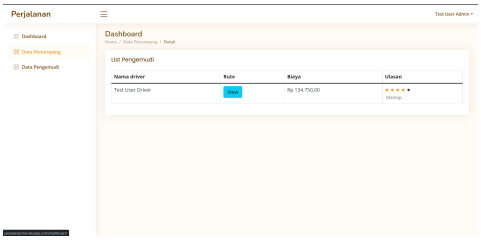
       password: password

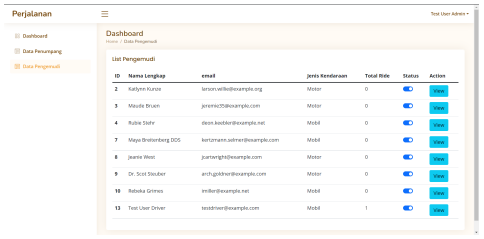
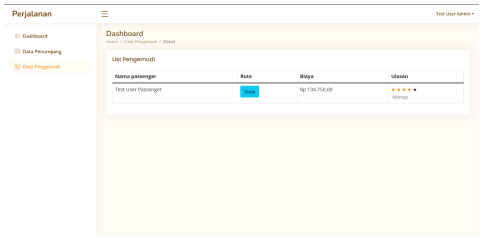
    Berikut tampilannya:



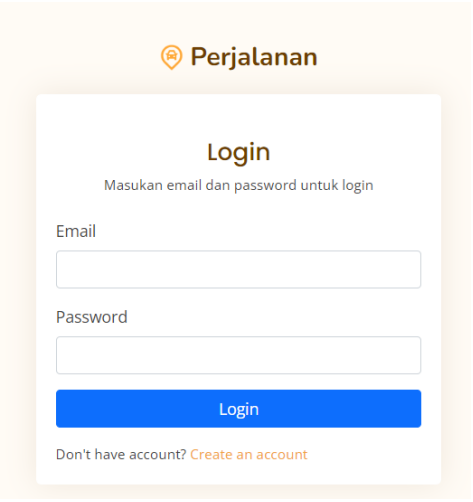
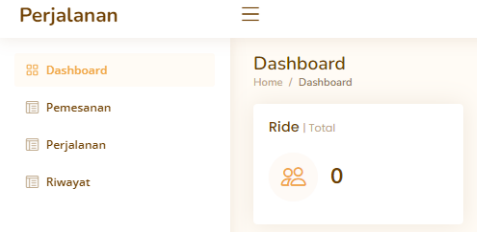
D. Cara Penggunaan

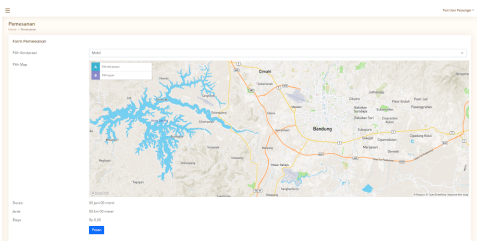
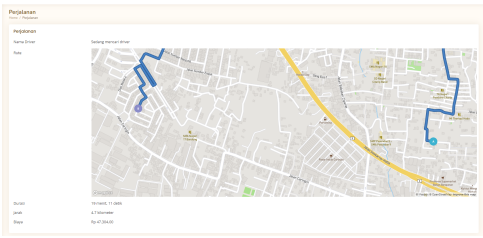
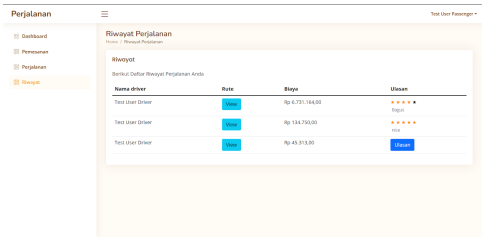
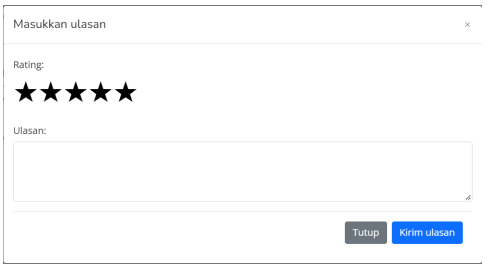
a. Admin

Nama	Screen	Cara Penggunaan
Login		Pada halaman login ini Anda dapat gunakan agar bisa masuk ke halaman beranda dengan memasukkan email dan password dengan role admin.
Dashboard		Pada halaman dashboard Anda dapat gunakan untuk melihat informasi jumlah penumpang dan driver yang terdaftar.
Data Penumpang		Pada halaman data penumpang Anda dapat gunakan untuk melihat list penumpang yang terdaftar dan dapat melihat nama, email, jumlah perjalanan, mengedit status, dan melihat list perjalanan yang telah dilakukan penumpang.
Detail Data Penumpang		Pada halaman detail data penumpang Anda dapat melihat riwayat yang telah dilakukan penumpang yang berupa nama driver, rute, biaya, dan ulasan.

Data Pengemudi		Pada halaman data pengemudi Anda dapat gunakan untuk melihat list pengemudi yang terdaftar dan dapat melihat nama, email kenis kendaraan, total ride, mengedit status, dan melihat list perjalanan yang telah dilakukan pengemudi.
Detail Data Pengemudi		Pada halaman detail data pengemudi Anda dapat melihat riwayat yang telah dilakukan pengemudi yang berupa nama pengemudi, rute, biaya, dan ulasan.

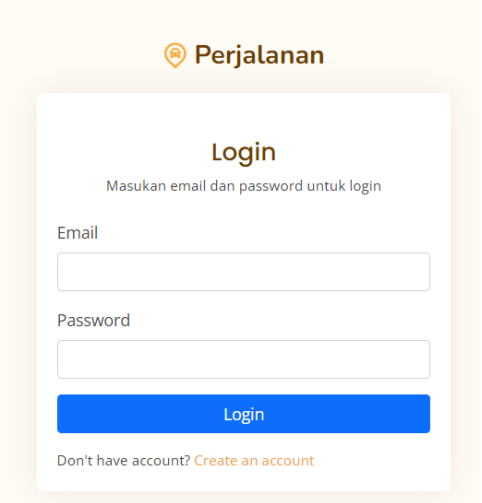
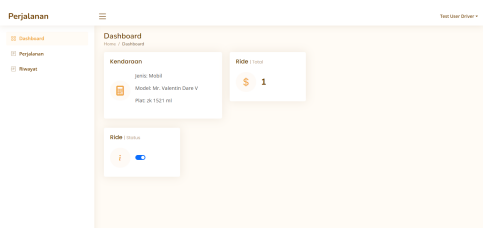
b. Passenger

Nama	Screen	Cara Penggunaan
Login		Pada halaman login ini Anda dapat gunakan agar bisa masuk ke halaman beranda dengan memasukkan email dan password dengan role passenger.
Dashboard		Pada halaman dashboard, Anda dapat melihat informasi ride yang telah dilakukan

Pemesanan		Pada halaman pemesanan, Anda dapat melakukan pemesanan dengan cara memilih kendaraan yang diinginkan, memilih rute sehingga app perjalanan menampilkan informasi mengenai jarak dan biaya yang dikalkulasikan .
Perjalanan		Pada halaman perjalanan, setelah melakukan pemesanan maka Aplikasi akan mencarikan driver di halaman Perjalanan, jika sudah diambil oleh driver maka akan menampilkan nama driver yang menerima.
History Perjalanan		Pada halaman Riwayat, Anda dapat melihat informasi mengenai ride yang telah dilakukan dan rute map yang dilalui.
Beri Ulasan		Pada halaman Riwayat, Anda dapat memberi ulasan terhadap driver yang telah melakukan perjalanan berupa rating dan review.

c. Driver

Nama	Screen	Cara Penggunaan

Login		Pada halaman login ini anda dapat menggunakannya untuk masuk ke halaman dashboard driver dengan memasukan email dan password driver
Dashboard		Pada halaman dashboard ini driver dapat melihat total ride yang sudah dilakukan dan mengubah ride status
Perjalanan		Pada halaman ini driver dapat memilih pesanan yang akan diambil
Riwayat		Jika sudah memilih pesanan dan melakukan perjalanan maka akan tertampil di riwayat beserta dengan ulasan dari penumpang