

**LAPORAN TUGAS 5 MEMBUAT WEBSITE FRONTPAGE
PEMANFAATAN ELOQUENT MODEL DAN DATABASE DINAMIS**



**Dosen Pengampu :
Ketut Agus Seputra, S.ST., M.T.**

**Disusun Oleh :
I Gede Gelgel Abdiutama ; 2115101014**

**MATA KULIAH TEKNOLOGI WEB
UNIVERSITAS PENDIDIKAN GANESHA
SINGARAJA
TA. 2022**

A. PENDAHULUAN

Laravel merupakan salah satu dari sekian banyak framework PHP yang dapat digunakan secara gratis. Laravel dikembangkan oleh programmer asal amerika yang bernama Taylor Otwell pada tahun 2011. Framework sendiri dapat diartikan sebagai kumpulan kode-kode program yang akan selalu digunakan pada setiap pembuatan aplikasi. Karena selalu digunakan maka kode-kode tersebut dikumpulkan dan disusun secara rapi pada folder-folder agar mudah digunakan dan jadilah sebuah framework. Laravel memiliki fitur Eloquent, yakni sebuah Object Relational Mapper (ORM) yang membuat interaksi controller dengan database menjadi lebih mudah. Untuk menggunakan fitur ini, maka setiap tabel dalam database wajib memiliki model yang merepresentasikannya sebagai sebuah object. Fitur dasar dari eloquent ini yakni create, read, update, delete serta relationalships.

B. MANFAAT DAN TUJUAN

1. Mahasiswa dapat memahami struktur Model
2. Mahasiswa dapat memahami Konsep Model, View dan Controller (MVC)
3. Mahasiswa dapat memahami Object Relational Mapper (ORM).
4. Mahasiswa dapat menggunakan ORM untuk manipulasi data Create Read Update Delete

C. PERMASALAHAN

Silakan buat sebuah frontpage sebuah website yang telah memanfaatkan eloquent model dan database, serta page telah dibuat dinamis.

Sumber video: <https://youtu.be/2--p4q6j0JU>

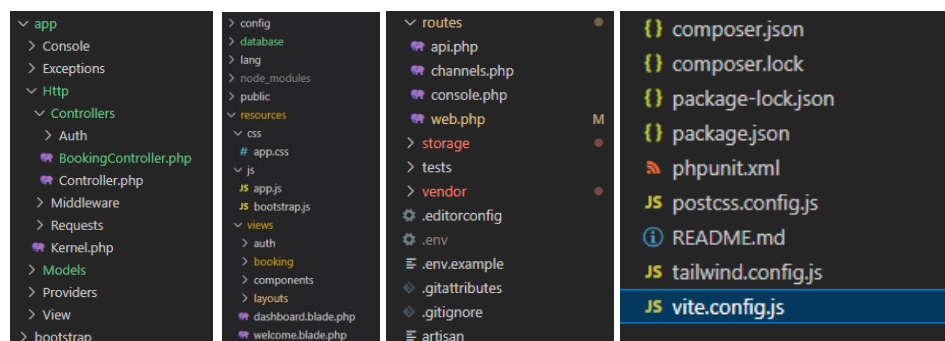
Ketentuan:

1. Tugas mandiri
2. Bisa menggunakan kasus yang telah dipilih kelompok, namun tetap dibuat mandiri
3. Keunikan produk setiap individu
4. Tugas dikumpulkan dalam PDF lengkap dengan narasi penjelasan produk serta hasil. Tugas ini akan terus dilanjutkan hingga pengembangan backpage, jadi tolong dipahami.

D. PEMBAHASAN

Pada tugas membuat website dengan implementasi eloquent model dan database secara dinamis ini, saya menggunakan project yang sudah dibuat sebelumnya yaitu “Trivili (Trip in Bali) Website Travel Agent Bali”. Website ini dibentuk berdasarkan hasil riset kecil mengenai pariwisata di daerah Bali yang masih sedikit terdapat website penyedia jasa pemandu wisata baik itu dari masyarakat lokal maupun luar. Didalam website ini nantinya akan berisi paket-paket wisata maupun hiburan di Bali bagi pengunjung untuk mempermudah dalam melakukan pemesanan dan review terkait destinasi-destinasi yang ada di Bali. Seperti yang kita ketahui, Bali merupakan sebuah provinsi di Indonesia yang beribukota di Denpasar. Provinsi Bali terletak di bagian barat Kepulauan Nusa Tenggara. Kini Bali sangat terkenal keramahan, kesenangan, kedamaian, dan keindahan yang dimiliki Bali, menjadikannya salah satu pulau paling populer di dunia. Bali juga dinobatkan sebagai destinasi populer tahun 2021 versi Tripadvisor. Dijuluki sebagai The Island of Gods, Bali memiliki beragam pilihan aktivitas yang bisa dicoba. Pariwisata di Bali sudah Go Internasional dan bahkan Bali lebih terkenal dari Indonesia. Website Trivili ini memiliki struktur dasar yang membentuknya didalam framework php laravel, yaitu MVC, ORM dan beberapa diantaranya sebagai berikut :

1. Struktur Folder

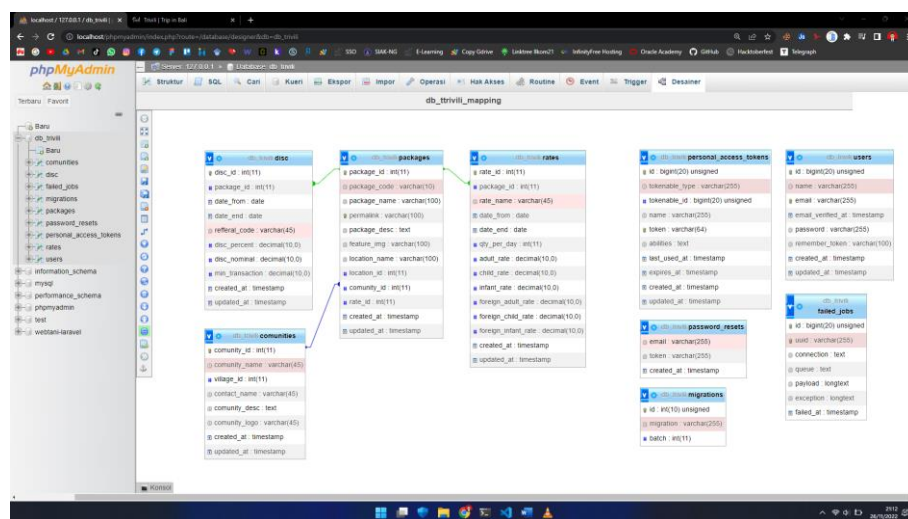


Website ini menggunakan framework php laravel, dengan struktur folder standar yang dimana kita mengolah sesuai struktur folder yang telah diberikan. Contoh saya pada website ini, jika saya menaruh file-file yang sifatnya publik seperti asset gambar, css, js, saya menaruhnya pada folder

public agar file tersebut dapat dilihat oleh orang lain. Sebaliknya, jika saya memiliki file-file yang sifatnya private maka saya tidak akan menaruhnya di folder public, sehingga orang lain tidak dapat melihatnya. Untuk file-file kebutuhan website yang sifatnya tidak boleh orang lain ketahui, maka saya menyimpannya sesuai ketentuan dari struktur folder laravel.

2. ORM (Object Relation Mapping)

• Relation Mapping



• Foreign Key

Foreign Key Name	Referenced Table	Column	Referenced Column
communities	'db_trivili', 'communities'	<input type="checkbox"/> package_id	
		<input type="checkbox"/> package_code	
		<input type="checkbox"/> package_name	
		<input type="checkbox"/> permalink	
		<input type="checkbox"/> package_desc	
		<input type="checkbox"/> feature_img	
		<input type="checkbox"/> location_name	
		<input type="checkbox"/> location_id	
		<input checked="" type="checkbox"/> community_id	community_id
		<input type="checkbox"/> rate_id	
		<input type="checkbox"/> created_at	
		<input type="checkbox"/> updated_at	

Foreign Key Name	Referenced Table	Column	Referenced Column
package	`db_trivili`, `packages`	<input type="checkbox"/> rate_id	
		<input checked="" type="checkbox"/> package_id	package_id
		<input type="checkbox"/> rate_name	
		<input type="checkbox"/> date_from	
		<input type="checkbox"/> date_end	
		<input type="checkbox"/> qty_per_day	
		<input type="checkbox"/> adult_rate	
		<input type="checkbox"/> child_rate	
		<input type="checkbox"/> infant_rate	
		<input type="checkbox"/> foreign_adult_rate	
		<input type="checkbox"/> foreign_child_rate	
		<input type="checkbox"/> foreign_infant_r...	
		<input type="checkbox"/> created_at	
		<input type="checkbox"/> updated_at	

Foreign Key Name	Referenced Table	Column	Referenced Column
disc	`db_trivili`, `packages`	<input type="checkbox"/> disc_id	
		<input checked="" type="checkbox"/> package_id	package_id
		<input type="checkbox"/> date_from	
		<input type="checkbox"/> date_end	
		<input type="checkbox"/> refferal_code	
		<input type="checkbox"/> disc_percent	
		<input type="checkbox"/> disc_nominal	
		<input type="checkbox"/> min_transaction	
		<input type="checkbox"/> created_at	
		<input type="checkbox"/> updated_at	

Website ini menggunakan sebuah database sql untuk menyimpan data seperti akun pengguna, item-item paket wisata yang ada didalam website serta merujuk juga pada proses transaksi yang ada pada website ini. Dalam pembuatan database sql saya menggunakan MariaDB melalui XAMPP phpmyadmin dan juga menggunakan bantuan MYSQL Workbench untuk membuat foreign key agar setiap tabel pada database ini memiliki relasi atau hubungan yang baik antar tabel, sehingga dapat membentuk sebuah database sql yang tepat digunakan untuk pengembangan website ini.

3. Model

Model merupakan bagian yang bertugas untuk menyiapkan, mengatur, memanipulasi, dan mengorganisasikan data yang ada di database dalam sebuah website. Pada website ini terdapat 4 buah model yang akan bertugas menangani data pada website.

- **Community**

```
app > Models > Community.php > PHP Intelephense > Community
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 1 reference|0 implementations
9 class Community extends Model
10 {
11     use HasFactory;
12     2 references
13     public $primaryKey = 'community_id';
14     4 references
15     protected $table = "communities";
16     0 references
17     protected $fillable = [
18         'community_name', 'village_id', 'contact_name', 'community_desc', 'community_logo'
19     ];
20     0 references|0 overrides
21     public function packages() {
22         return $this->hasMany(Packages::class, 'community_id', 'community_id');
23     }
24 }
```

Model Community bertugas untuk menghubungkan database pada website agar data pada database di beberapa kolom tabel communities dapat di return ke view pada website kita.

- **Packages**

```
app > Models > Packages.php > PHP Intelephense > Packages > rates
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 4 references|0 implementations
9 class Packages extends Model
10 {
11     use HasFactory;
12     2 references
13     public $primaryKey = 'packages_id';
14     4 references
15     protected $table = "packages";
16     0 references
17     protected $fillable = [
18         'package_code', 'package_name', 'package_desc', 'feature_img', 'location_id', 'community_id', 'rate_id', 'rate_name', 'date_from', 'date_end', 'qty_per_day', 'adult_rate', 'child_rate',
19         'infant_rate', 'foreign_adult_rate', 'foreign_child_rate', 'foreign_infant_rate'
20     ];
21     0 references|0 overrides
22     public function community () {
23         return $this->belongsTo(Community::class, 'community_id', 'community_id');
24     }
25     2 references|0 overrides
26     public function rates () {
27         return $this->belongsTo(Rates::class, 'rate_id', 'rate_id');
28     }
29 }
30
31
```

Model Packages bertugas untuk menghubungkan database pada website agar data pada database di beberapa kolom tabel packages dapat di return ke view pada website kita. Model Packages ini dapat juga dikatakan sebagai poin utamanya, karena setiap tabel yang ada pada database yang berkaitan dengan package akan saling terhubung kedalamnya, sehingga membentuk foreign key yang menjadi referensi dari setiap tabel.

- **Rates**

```
app > Models > Rates.php > PHP Intellisense > Rates > packages
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 /**
9  * @property int $rate_id
10  */
11 class Rates extends Model
12 {
13     use HasFactory;
14
15     /**
16      * The primary key for the model.
17      *
18      * @var string
19      */
20     protected $primaryKey = 'rate_id';
21
22     /**
23      * The table associated with the model.
24      *
25      * @var string
26      */
27     protected $table = 'rates';
28
29     /**
30      * The attributes that are mass assignable.
31      *
32      * @var array<int, string>
33      */
34     protected $fillable = [
35         'package_id', 'rate_name', 'date_from', 'date_end', 'qty_per_day', 'adult_rate', 'child_rate', 'infant_rate', 'foreign_adult_rate', 'foreign_child_rate', 'foreign_infant_rate'
36     ];
37
38     /**
39      * The attributes that should be hidden for serialization.
40      *
41      * @var array<int, string>
42      */
43     protected $hidden = [
44         'password', 'remember_token'
45     ];
46
47     /**
48      * The attributes that should be cast.
49      *
50      * @var array<string, string>
51      */
52     protected $casts = [
53         'email_verified_at' => 'datetime',
54     ];
55 }
```

Model Rates bertugas untuk menghubungkan database pada website agar data pada database di beberapa kolom tabel rates dapat di return ke view pada website kita.

- **User**

```
app > Models > User.php > ...
1 <?php
2
3 namespace App\Models;
4
5 // use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Sanctum\HasApiTokens;
10
11 /**
12  * @property int $id
13  */
14 class User extends Authenticatable
15 {
16     use HasApiTokens, HasFactory, Notifiable;
17
18     /**
19      * The attributes that are mass assignable.
20      *
21      * @var array<int, string>
22      */
23     protected $fillable = [
24         'name',
25         'email',
26         'password',
27     ];
28
29     /**
30      * The attributes that should be hidden for serialization.
31      *
32      * @var array<int, string>
33      */
34     protected $hidden = [
35         'password',
36         'remember_token',
37     ];
38
39     /**
40      * The attributes that should be cast.
41      *
42      * @var array<string, string>
43      */
44     protected $casts = [
45         'email_verified_at' => 'datetime',
46     ];
47 }
```

Model User bertugas untuk menghubungkan database pada website agar data-data user seperti informasi untuk login, registrasi, lupa kata sandi dan verifikasi email pada database dapat di return ke view pada website kita.

4. View

View merupakan bagian yang bertugas untuk menampilkan informasi website yang sudah dibentuk dalam bentuk Graphical User Interface (GUI). Semua baris kode yang berkaitan dengan yang pengguna bisa lihat nanti akan di eksekusi pada view ini, sehingga dalam framework laravel view adalah tempat kita melakukan pembuatan desain halaman atau page pada website sehingga dapat dikelola oleh controller sebagai otak dari website kita ataupun model sebagai pengelola database nya yang nantinya akan dipanggil oleh route sebagai tempat proses penjaluran dari setiap halaman web yang dipanggil. Dalam website triviali ini terdapat beberapa view dasar yang membentuk website ini, dan juga pada semua view juga telah diatur responsive web design nya sehingga tampilan pada desktop, tablet maupun handphone dapat berjalan dengan baik dan tampilan website tetap terjaga.

- Welcome

```
resources > views > welcome.blade.php > ...
1 <@csrf >
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}" class="scroll-smooth">
3 <head>
4 <meta charset="utf-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1">
6 <meta name="csrf-token" content="{{ csrf_token() }}">
7 <title>triviali | Trip in Bali</title>
8 <link rel="preconnect" href="https://fonts.googleapis.com">
9 <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
10 <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=block" rel="stylesheet">
11 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css" integrity="sha512-xh6O/CkQqPOFDBvE8PfCV1SPr4Cv0eK5jSP78xq000LnXf25fW7255N8l6Baqz8a9s" crossorigin="anonymous">
12 <link rel="shortcut icon" href="img/favicon-triviali.png">
13 <@vite(['resources/css/app.css', 'resources/js/app.js'])>
14 </head>
15 <body>
16 <@auth>
17 <@include('layouts.navigation')>
18 </@auth>
19 <div class="absolute w-full flex items-center pt-3 pb-3">
20 <div class="flex items-center justify-between relative">
21 <div class="px-4 flex items-center cursor-pointer">
22 <img alt="application logo" />
23 </div>
24 </div>
25 <@if (Route::has('login'))>
26 <div class="absolute right-0 pr-6">
27 <div>
28 <@if (Route::has('login'))>
29 <@if (Route::has('register'))>
30 <@if (Route::has('login'))>
31 <@if (Route::has('register'))>
32 <@if (Route::has('login'))>
33 <@if (Route::has('register'))>
34 <@if (Route::has('login'))>
35 <@if (Route::has('register'))>
36 <@if (Route::has('login'))>
37 <@if (Route::has('register'))>
38 <@if (Route::has('login'))>
39 <@if (Route::has('register'))>
40 <@if (Route::has('login'))>
41 <@if (Route::has('register'))>
42 <@if (Route::has('login'))>
43 <@if (Route::has('register'))>
44 <@if (Route::has('login'))>
45 <@if (Route::has('register'))>
46 <@if (Route::has('login'))>
47 <@if (Route::has('register'))>
```

Welcome adalah view untuk menampilkan tampilan awal pada saat website pertama kali diakses oleh pengunjung web yang menampilkan isi singkat dari website, pengunjung juga dapat melakukan proses login maupun register pada tampilan ini.

- **Dashboard**

```
sources / view / dashboard.html.php > @ x app-layout
1 <!-- app-layout -->
2 <!-- Name Section Start -->
3 <section id="name" class="pt-36 bg-slate-100">
4 </section>
5 <!-- Name Section End -->
6
7 <!-- Pricing Section Start -->
8 <section id="pricing" class="pt-36 pb-16 bg-slate-100">
9 </section>
10 <!-- Pricing Section End -->
11
12 <!-- Gallery Section Start -->
13 <section id="gallery" class="pt-36 pb-32 bg-slate-200">
14 </section>
15 <!-- Gallery Section End -->
16
17 <!-- About Section Start -->
18 <section id="about" class="pt-36 pb-32 bg-slate-300">
19 </section>
20 <!-- About Section End -->
21
22 <!-- Partner Section Start -->
23 <section id="partner" class="pt-36 pb-32 bg-slate-200">
24 </section>
25 <!-- Partner Section End -->
26
27 <!-- Contact Section Start -->
28 <section id="contact" class="pt-36 pb-32 bg-slate-100">
29 </section>
30 <!-- Contact Section End -->
31
32 <!-- Footer Section Start -->
33 <footer class="bg-dark pt-24 pb-12">
34 </footer>
35 <!-- Footer Section End -->
36
37 <!-- Toggle Return Home Start -->
38 <a href="#" class="fixed hidden justify-center items-center z-[999] bottom-3 right-3 p-3 h-12 w-12 transition duration-300 ease-in-out bg-primary rounded-full text-white text-center
39 hover:animate-pulse id="btn-return">
40 </a>
41 <!-- Toggle Return Home End -->
42
43 </x-app-layout>
44
```

Dashboard adalah view untuk menampilkan tampilan setelah view welcome, ini ditampilkan jika pengunjung telah mengeksekusi tombol Get Started yang ada pada view welcome, sehingga pengunjung akan diarahkan ke tampilan Dashboard yang memuat banyak informasi mengenai website.

- **Booking**

```
sources / view / booking > @ x app-layout
1 <!-- app-layout -->
2 <main class="py-6 bg-slate-100">
3 <div class="max-w-7xl mx-auto lg:px-8">
4 <div class="flex items-center justify-between border-b border-gray-200 pt-6 pb-6 lg:px-4">
5 <!-- Component Checker -->
6 <div class="flex items-center justify-between border-b border-gray-200 pt-6 pb-6 lg:px-4">
7 <div class="text-2xl font-bold tracking-tight text-gray-900 lg:text-2xl pb-1">Paket Destinasi Trivili/Nil</div>
8 <div class="flex items-center">
9 <div data-range="date" class="lg:flex justify-center items-center">
10 </div>
11 <div>
12 <button type="button"
13 class="inline-block px-4 py-3 border border-slate-300 text-primary font-medium text-xs leading-tight uppercase rounded-md hover:bg-red-400 hover:text-white
14 focus:outline-none focus:ring-0 transition duration-150 ease-in-out">
15 href="#paket-wisata" <Cek Tiket> </button>
16 <button type="button"
17 class="inline-block px-4 py-3 border border-slate-300 text-primary font-medium text-xs leading-tight uppercase rounded-md hover:bg-red-400 hover:text-white
18 focus:outline-none focus:ring-0 transition duration-150 ease-in-out">
19 href="#slide-over-title" <Keranjang> </button>
20 </div>
21 </div>
22 <!-- Component Input Informal Pengunjung -->
23 <div class="justify-center items-center" id="paket-wisata">
24 </div>
25 <!-- Component Home -->
26 <div class="lg:flex">
27 </div>
28 </div>
29 </div>
30 </main>
31 </x-app-layout>
32
```

Booking adalah view untuk menampilkan tampilan proses booking/pesanan tiket. Jika pengunjung menekan tombol booking atau pesanan, maka akan diarahkan ke halaman ini, sehingga pengunjung akan dapat melihat paket-paket destinasi yang tersedia dan langsung melakukan pemesanan.

- **App**

```

1 <!DOCTYPE html>
2 <html lang="">{{ str_replace('_', '-', app().getLocale()) }}</html><!-- scroll-smooth -->
3 <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <meta name="csrf-token" content="{{ csrf_token() }}">
7     <title>Trivili | Trip In Bali</title>
8     <link rel="preconnect" href="https://fonts.googleapis.com">
9     <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
10    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display=swap" rel="stylesheet">
11    <link href="stylesheet" href="https://cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css" integrity="sha512-xh0oUxOxUjxQw3DzKozLqZS7YnW3cuyk/jA/R6+yo/hpeVj9UEtS7dEocYMepBfiWNqF8imPgCV6FR71ngWU?">
12    <link href="shortcuts icon" href="img/favicon-trivili.png">
13    @if('resources/css/app.css', 'resources/js/app.js')
14        <script src='../path/to/floabit/dist/datepicker.js'></script>
15        <script src='http://wpak.com/flooditapi.9.3/dist/datepick.js'></script>
16        <script src='http://www.googleapis.com/js/lib/googleapis.js&#x2F;j&#x2F;q&#x2F;js&#x2F;min.js'></script>
17        <script src='http://api.lksgroup.co.id/api/trilokasipg.js'></script>
18        <script src='http://api.lksgroup.co.id/api/trilokasipg.js'></script>
19    @endif
20    <script>
21        var renderCreatedWidgetlokasi( "provinsi","kabupaten","kecamatan","kelurahan");
22        $(document).ready(function(){
23            $('#search').click(function(){
24                //
25            });
26            $('#nama').keyup(function(){
27                //
28            });
29            $('#email').keyup(function(){
30                //
31            });
32            $('#alamat').keyup(function(){
33                //
34            });
35            $('#provinsi').keyup(function(){
36                //
37            });
38            $('#regency').keyup(function(){
39                //
40            });
41            $('#pus').keyup(function(){
42                //
43            });
44            $('#kampung').click(function(){
45                //
46            });
47        });
48    </script>
49 </head>
50 <body>
51     <div class="layouts.navigation">
52         {{ $slot }}
53     </div>
54 </body>
55 </html>

```

App adalah view yang difungsikan sebagai kerangka dasar dari view yang lainnya (untuk menampung semua view lainnya). Artinya App ini akan bertugas sebagai dasarnya saja yang akan menampung seperti body dari website maupun navigation, sehingga dalam view ini kita hanya tinggal memanggil view lainnya kedalamnya, ini akan mengurangi redundansi pada kodingan kita, sehingga kodingan yang dibuat berkali-kali disetiap view nya dapat diminimalisir dengan membuat 1 view khusus untuk tampilan tersebut dan dipanggil setiap kali dibutuhkan saja, sehingga akan menghemat baris kode yang kita buat pada website.

- **Navigation**

```
resources > views > layouts > navigation.blade.php > ...
1 <div x-data="{ open: false }" class="bg-slate-100 border-gray-100">
2 <!-- Primary Navigation Menu -->
3 <header class="w-full mx-auto px-4 sm:px-6 lg:px-8">
4 <div class="flex justify-between h-16">...
83 </div>
84
85 <div :class="{ 'block': open, 'hidden': !open }" class="hidden sm:hidden">...
127 </div>
128 </header>
129 </div>
130
```

Navigation adalah view untuk menampilkan tampilan header sekaligus navigasi bar yang ada pada website, dikemas menjadi 1 agar dapat dipanggil secara berulang oleh view app, sehingga tidak dibuat berkali-kali di setiap view pada saat membuat tampilan halaman baru.

5. Controller

Controller merupakan bagian yang bertugas untuk menghubungkan serta mengatur model dan view agar dapat saling terhubung. Website ini menggunakan beberapa controller untuk mengolah setiap proses dari tindakan yang terjadi pada website, contoh pada kasus ini adalah website trivilin yang memerlukan controller untuk mengatur jalannya pemesanan tiket. Adapun beberapa controller yang telah dibuat yaitu :

- **Main Controller**

```
app > Http > Controllers > Controller.php > ...
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Foundation\Auth\Access\AuthorizesRequests;
6  use Illuminate\Foundation\Bus\DispatchesJobs;
7  use Illuminate\Foundation\Validation\ValidatesRequests;
8  use Illuminate\Routing\Controller as BaseController;
9
10 class Controller extends BaseController
11 {
12     use AuthorizesRequests, DispatchesJobs, ValidatesRequests;
13 }
14
```

Main controller merupakan controller default yang telah disediakan otomatis pada saat kita menggunakan framework php laravel.

- **BookingController**

```
app > Http > Controllers > BookingController.php > PHP Intelephense > BookingController > index
2
3  namespace App\Http\Controllers;
4
5  use App\Models\Packages;
6  use App\Models\Rates;
7  use Illuminate\Http\Request;
8
9  class BookingController extends Controller
10 {
11     public function index(){
12         // $x=12345;
13         // $y=67890;
14         // $hasil=$x*$y;
15         // $nama='-';
16         // $email='-';
17         // $alamat='-';
18         // $provinsi='-';
19         // $negara='-';
20         // $pos='-';
21         // return view('booking.booking', compact('hasil','x','y','nama','email','alamat','provinsi','negara','pos'));
22         // return view('booking.booking');
23
24         $packages=Packages::all();
25         $rates=Rates::all();
26         return view('booking.booking', compact('packages','rates'));
27     }
28 }
29
```

BookingController merupakan controller yang bertugas untuk mengatur jalannya pemanggilan data yang ada pada database kita yang akan dimasukkan kedalam view booking. Dalam controller ini, kita menghubungkan antara Model dan juga Controller, Model Packages dan

Rates yang didalamnya telah berisi algoritma pemrograman untuk mengolah database akan dipanggil ke dalam controller ini yang pada gambar dapat dilihat semua isi dari model Package dan Rates akan dipanggil kedalam controller, kemudian akan direturn beberapa variabel yang dibutuhkan dengan bantuan syntax compact. Didalam view booking kita dapat melakukan pemilihan kolom pada tabel mana saja yang akan kita sisipkan didalam view ini. Pada view booking, ada beberapa elemen dari database yang disisipkan yaitu nama paket, harga paket, deskripsi paket, lokasi paket, dan community dari paket tersebut. Sehingga tampilan view nya akan tampil secara dinamis.

6. Route

```
routes > web.php > ...
1  <?php
2
3  use App\Http\Controllers\BookingController;
4  use Illuminate\Support\Facades\Route;
5
6  /*
7  |-----
8  | Web Routes
9  |-----
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 */
16
17 Route::get('/', function () {
18     return view('welcome');
19 });
20
21 Route::get('/dashboard', function () {
22     return view('dashboard');
23 }->middleware(['auth', 'verified']->name('dashboard'));
24
25 Route::get('booking',[BookingController::class,'index']->name('booking'));
26
27 require __DIR__.'/auth.php';
```

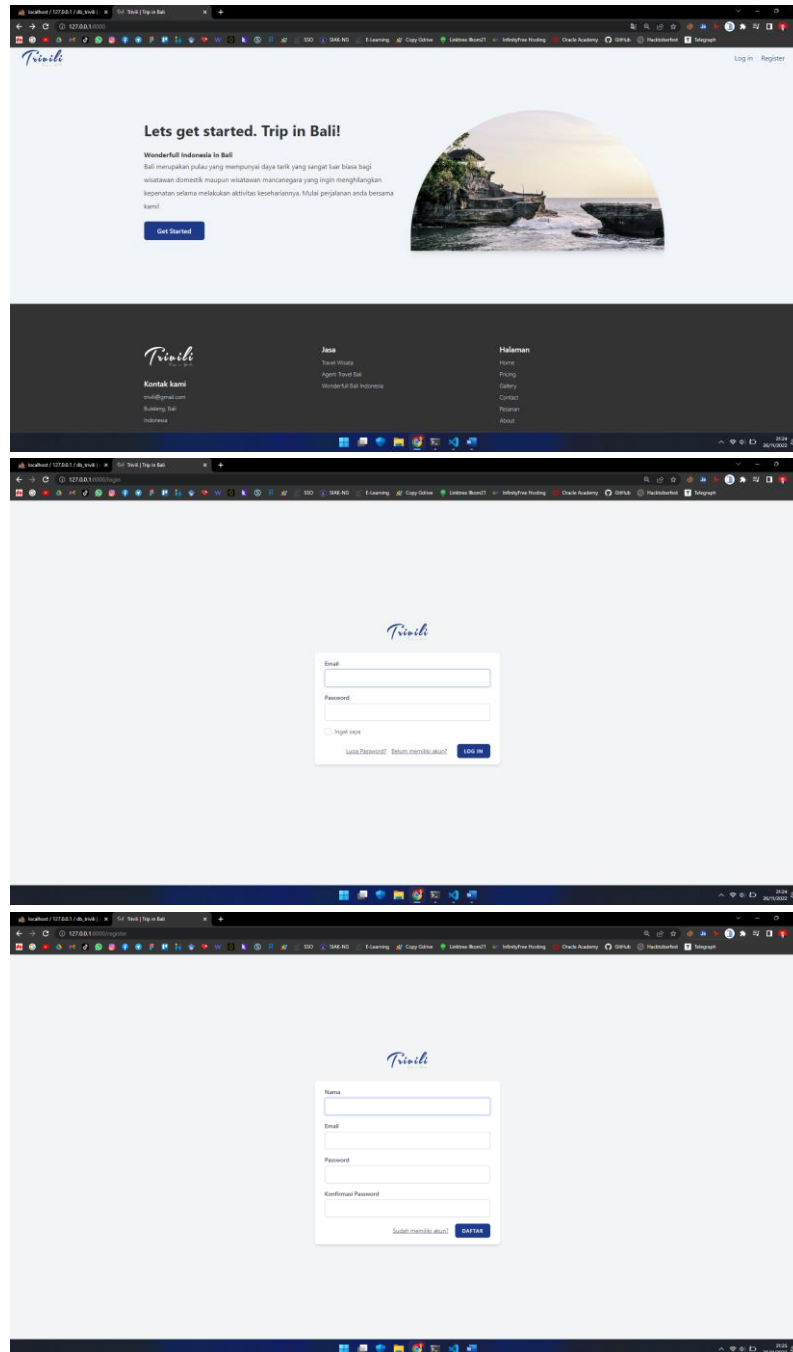
Route merupakan bagian bertugas untuk menentukan controller yang menampung view-view yang ada dan method/fungsi yang akan dieksekusi. Secara default pada framework laravel telah dibuatkan standar route yang siap untuk kita langsung gunakan, pada website ini saya hanya menggunakan route web.php saja, sehingga saya akan menambahkan beberapa route/melakukan routing pada web.php. Terdapat 3 route pada website ini yaitu pertama route untuk menampilkan view welcome ketika

pengunjung pertama kali mengakses website kita. Kemudian route untuk menampilkan view dashboard, namun disini terdapat middleware yang mengiringi jalannya route ini, artinya view ini akan terlihat atau dapat diakses ketika pengunjung telah melakukan register atau login yang akan di cek informasi penggunaannya pada database. Dan terakhir terdapat route untuk menampilkan view booking yang terdapat pada BookingController jika pengunjung mengeksekusi button yang telah di set akan mengarahkan ke tampilan booking, sehingga view yang telah direturn pada BookingController akan di panggil pada route ini dan ditampilkan ke website sehingga pengunjung dapat melihat halaman untuk melakukan pemesanan/booking tiket.

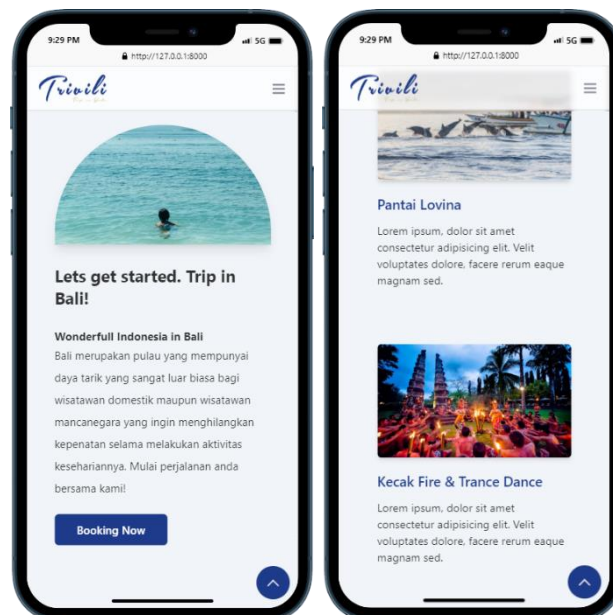
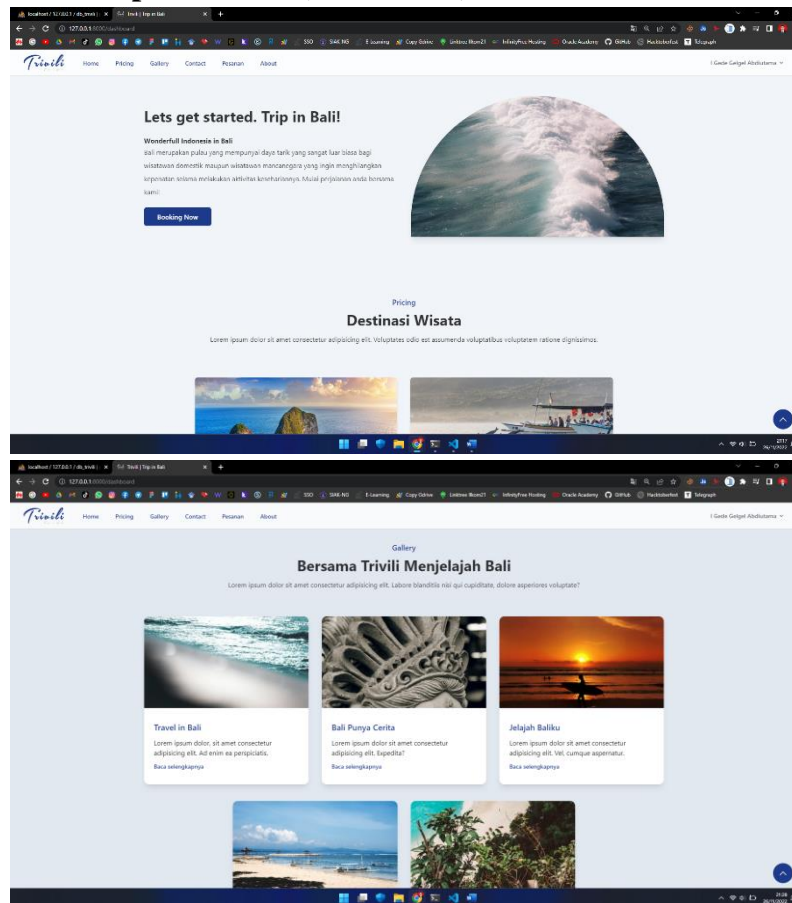
7. Antarmuka output

Pada antarmuka output, diperoleh beberapa hasil output pada website trivili ini pada seluruh tampilan yang ada.

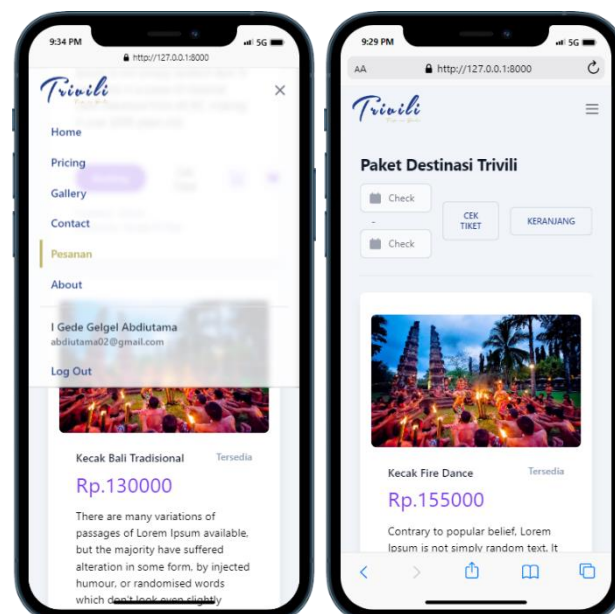
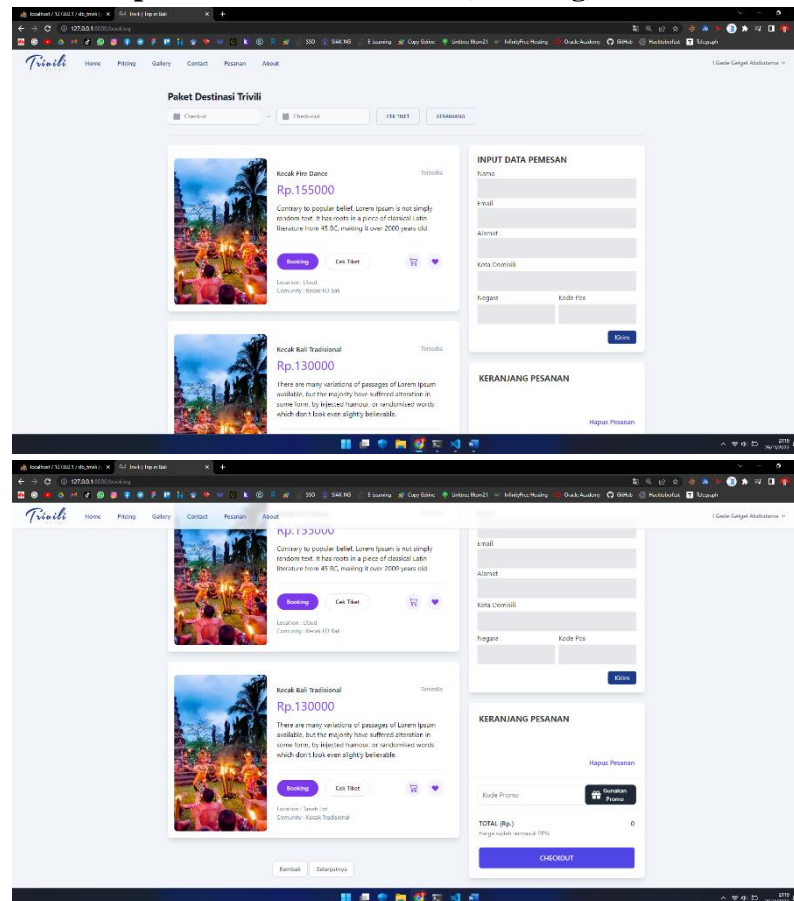
- **Tampilan awal (Welcome)**



• Tampilan Home (Dashboard)



• Tampilan Menu Pemesanan (Booking)



E. KESIMPULAN

Dalam pembuatan website trivili dengan studi kasus tari kecak ini, menggunakan bahasa peromgraman web framework php laravel. Pada proses pembuatannya, tetap mengikuti aturan atau kerangka kerja dari framework php laravel ini yaitu MVC. Ini adalah sebuah model kerangka pemisah aplikasi web berdasarkan komponen-komponennya, seperti manipulasi data, controller, dan antarmuka interface. Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data didalam halaman web dan lain-lain. View adalah bagian yang mengatur tampilan ke pengguna. Dan Controller merupakan bagian yang menjembatani model dan view sehingga dapat saling terhubung sesuai dengan yang dibuat. Dari permasalahan yang diberikan, ide saya adalah membuat suatu website dengan nama Trivili yang dapat menampung beberapa destinasi-destinasi wisata contohnya pada studi kasus ini yaitu tari kecak, sehingga pengguna yang menggunakan web ini dapat melihat beberapa destinasi wisata yang ada di bali hanya dalam 1 website saja dan rencana untuk kedepannya ini akan digunakan sebagai project hingga tugas akhir pada matakuliah Teknologi Web ini.

F. LINK PROJECT

Website Trivili ini dikembangkan dan asset kode programnya dipublish secara publik dengan bantuan GitHub. Berikut linknya :

- URL GitHub Personal : <https://github.com/odetv>
- URL GitHub Project Tekweb : <https://github.com/odetv/PROJECT-TEKNOWEB>
- URL GitHub Project Tugas ini : <https://github.com/odetv/trivili-laravel>