

## PENDAHULUAN

Dalam era digital yang terus berkembang, sistem informasi penjualan telah menjadi elemen penting dalam operasional bisnis. Hal ini tidak terkecuali bagi *Mobile Cell*, sebuah toko penjualan perangkat seluler yang berlokasi di Kabupaten Tanah Toraja, Sulawesi Selatan. Kabupaten Tanah Toraja adalah daerah yang dikenal karena kekayaan budaya dan sejarahnya, dan bisnis seperti *Mobile Cell* memiliki peran krusial dalam memenuhi kebutuhan teknologi dan komunikasi masyarakat setempat[1].

*Mobile Cell* di Kabupaten Toraja masih menggunakan transaksi konvensional pada proses penjualannya. Metode konvensional lebih dipilih karena kesederhanaan dalam mengelolanya[2]. Cukup dengan sebuah buku besar untuk mencatat seluruh transaksi yang terjadi. Kekurangan dari transaksi konvensional yaitu proses rekapitulasi semua transaksi yang lambat dan pembuatan laporan yang memakan waktu. Kekurangan yang lain adalah potensi terjadinya kesalahan pencatatan juga lebih besar ketika data yang harus dimasukkan semakin banyak[3].

Untuk menghadapi persaingan yang semakin ketat dan meningkatkan efisiensi dalam manajemen penjualan serta inventaris produk, *Mobile Cell* merencanakan pengembangan Sistem Informasi Penjualan berbasis *Web*. Sistem ini akan membantu dalam mengelola transaksi penjualan, pelacakan stok produk, serta memberikan layanan yang lebih baik kepada pelanggan[4].

Penggunaan *Framework Laravel* sebagai kerangka kerja pengembangan menjadi pilihan utama dalam membangun sistem ini. *Framework Laravel* adalah salah satu alat pengembangan aplikasi *Web* yang paling populer dan memiliki fitur-fitur yang kuat untuk menciptakan aplikasi *Web* yang aman, skalabel, dan mudah dikelola[5].

Melalui penelitian ini, *Mobile Cell* berupaya untuk menghadirkan solusi teknologi yang dapat meningkatkan efisiensi operasional dan memberikan pengalaman pelanggan yang lebih baik[6]. Penelitian ini menguraikan dengan rinci tentang implementasi, fitur, dan manfaat dari Sistem Informasi Penjualan Perangkat Seluler berbasis *Web* di *Mobile Cell*, serta bagaimana sistem ini diharapkan dapat menguntungkan bisnis *Mobile Cell* di Kabupaten Tanah Toraja dan meningkatkan layanan kepada pelanggan setempat.

## STUDI PUSTAKA

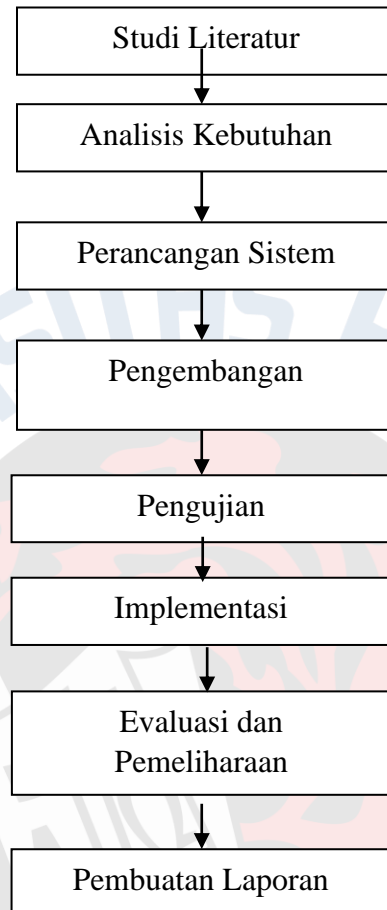
Pada penelitian yang berjudul "Rancang Bangun Sistem Informasi Penjualan Pada Toko Buku Kita Tasikmalaya Berbasis *Web* Menggunakan *Framework Laravel 8*" disebutkan bahwa *Framework Laravel* merupakan salah satu kerangka kerja dari bahasa pemrograman *PHP*. Kerangka kerja ini memiliki banyak fitur yang dapat membantu mempercepat proses pembuatan aplikasi. Menggunakan konsep *MVC* ( *Model View Controller* ) pemisahan antara tampilan (*frontend*) dengan *Controller* (*back-end*) menjadi salah satu fitur yang mempermudah dalam pembuatan aplikasi. Sehingga *Laravel* bisa menempati kerangka kerja *PHP* yang terpopuler. Toko Buku Kita adalah sebuah usaha yang mengkhususkan diri dalam penjualan buku. Saat ini, proses penjualan di Toko Buku Kita masih dilakukan secara manual, yang mengakibatkan beberapa masalah dalam penginputan data dan keterlambatan dalam pembuatan laporan. Oleh karena itu,

dibutuhkan sistem penjualan yang dapat mempermudah dan meningkatkan produktivitas staf di Toko Buku Kita[7]. Penelitian ini menyoroti kelebihan penggunaan *Framework Laravel* dalam pembangunan sistem informasi penjualan, menyebutkan bahwa kemudahan akses *plugin* dan pemisahan desain *database* dari desain tampilan adalah faktor yang mempermudah pengembangan sistem yang kompleks.

Pada penelitian yang berjudul “Implementasi *Framework Laravel* Dalam Perancangan Sistem Informasi Penjualan UKM Kota Surakarta” disebutkan bahwa *Website* dapat dibangun dengan mudah dan terjangkau. *Framework*, *bootstrap*, *CMS* dan lainnya memudahkan pengembangan *Website*. Salah satu keunggulannya adalah fleksibilitas dalam mengakses sistem. UKM Kota Surakarta masih banyak menggunakan transaksi konvensional pada proses penjualan[8]. Kekurangan dari transaksi konvensional yaitu proses transaksi yang lambat dan pembuatan laporan yang lama. Penelitian ini menyoroti bahwa penggunaan teknologi *Website*, *Framework*, dan *CMS* telah membantu Usaha Kecil Menengah di Kota Surakarta meningkatkan efisiensi dalam pengelolaan transaksi dan pelaporan bisnis mereka, mengatasi kekurangan dari metode konvensional yang lambat dan berisiko kesalahan.

Pada penelitian yang berjudul “Pengembangan Sistem Informasi Penjualan Barang Elektronik berbasis *Website* menggunakan *Framework Laravel* (Studi Kasus: UD Berkah Menuju Sukses Jombang)” disebutkan bahwa *Laravel* merupakan sebuah *MVC Web development Framework open source* yang banyak digunakan oleh pengembang *Website* karena ringan dan cepat dalam pembangunan pemrograman *Website* yang dinamis. *Laravel* memiliki fitur *Eloquent ORM* dan *Blade template* yang bisa mempercepat penulisan sebuah kode program sehingga dapat mempengaruhi efisiensi proses pengembangan sebuah sistem dengan mudah. UD Berkah Menuju Sukses saat ini hanya melakukan penjualan produk secara langsung melalui toko dan masih dianggap cukup sederhana. Hal ini disebabkan oleh kebutuhan pelanggan yang ingin membeli barang elektronik yang masih memerlukan bantuan karyawan toko untuk mendapatkan informasi detail atau spesifikasi barang yang dijual. Keterbatasan karyawan toko dalam memberikan detail tentang barang menjadi masalah yang signifikan, terutama karena produk yang dijual, terutama spare-part elektronik, selalu mengalami pembaruan dari pihak pemasok. Oleh karena itu, karyawan toko harus selalu memahami dengan baik detail-detail produk tersebut[9]. Berdasarkan penelitian terdahulu, ditemukan bahwa penggunaan kerangka kerja berbasis Web *Laravel* dan integrasi *API payment gateway Midtrans* dapat mempermudah pengembangan sistem informasi penjualan barang elektronik berbasis *Website*, meningkatkan efisiensi dalam penyampaian informasi produk, dan memungkinkan transaksi pembayaran yang lebih mudah, namun belum ada penekanan pada pengujian performa dan pengalaman pengguna yang cukup dalam, yang menjadi fokus penelitian ini.

## METODE PENELITIAN

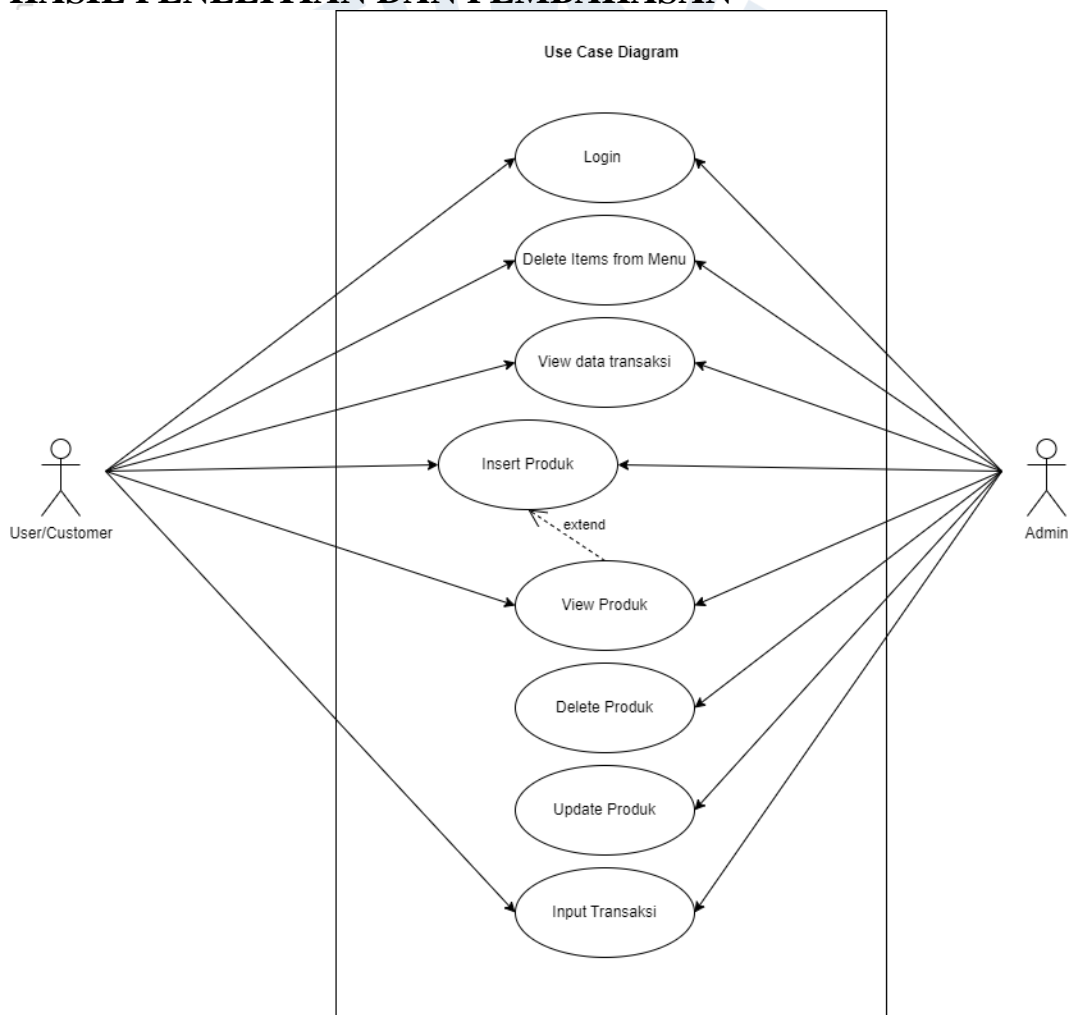


Gambar 1. Tahapan Penelitian

Terdapat delapan tahapan pada penelitian ini dimulai dari studi literatur hingga pembuatan laporan. Tahapan penelitian pada Gambar 1 dapat dijelaskan sebagai berikut. Tahapan pertama adalah studi literatur untuk memahami sistem informasi penjualan berbasis Web yang menggunakan *Laravel* sebagai *framework*, serta riset terkini tentang implementasi sistem informasi penjualan di industri serupa. Tahap berikutnya adalah analisis kebutuhan untuk memahami kebutuhan spesifik *Mobile Cell*, identifikasi kebutuhan fungsional dan non-fungsional sistem, serta evaluasi kendala dan tantangan yang mungkin dihadapi selama pengembangan. Tahap ketiga adalah perancangan sistem untuk membuat desain arsitektur sistem informasi penjualan menggunakan *Laravel*, rancang basis data untuk menyimpan informasi penjualan dan inventaris produk, serta pilih teknologi dan alat yang akan digunakan selama pengembangan. Tahap keempat adalah pengembangan untuk implementasikan sistem informasi penjualan berbasis Web menggunakan *Laravel*, lakukan uji coba secara berkala selama proses pengembangan untuk mengidentifikasi dan memperbaiki bug atau masalah

potensial. Tahap kelima adalah pengujian untuk melakukan pengujian fungsional dan pengujian kinerja untuk memastikan sistem berjalan sesuai dengan kebutuhan, ajak pihak terkait di *Mobile Cell* untuk menguji sistem dan memberikan umpan balik. Tahap keenam adalah implementasi untuk terapkan sistem informasi penjualan berbasis Web di lingkungan produksi *Mobile Cell*, lakukan pelatihan kepada tim yang akan menggunakan sistem baru. Tahap ketujuh adalah evaluasi dan pemeliharaan untuk evaluasi kinerja sistem setelah implementasi, berikan pemeliharaan rutin dan pembaruan jika diperlukan, monitor kinerja sistem secara terus-menerus dan tanggapilah perubahan kebutuhan atau masalah yang muncul. Tahap kedelapan adalah buat dokumentasi lengkap tentang sistem informasi penjualan, termasuk petunjuk penggunaan, struktur basis data, dan kode sumber.

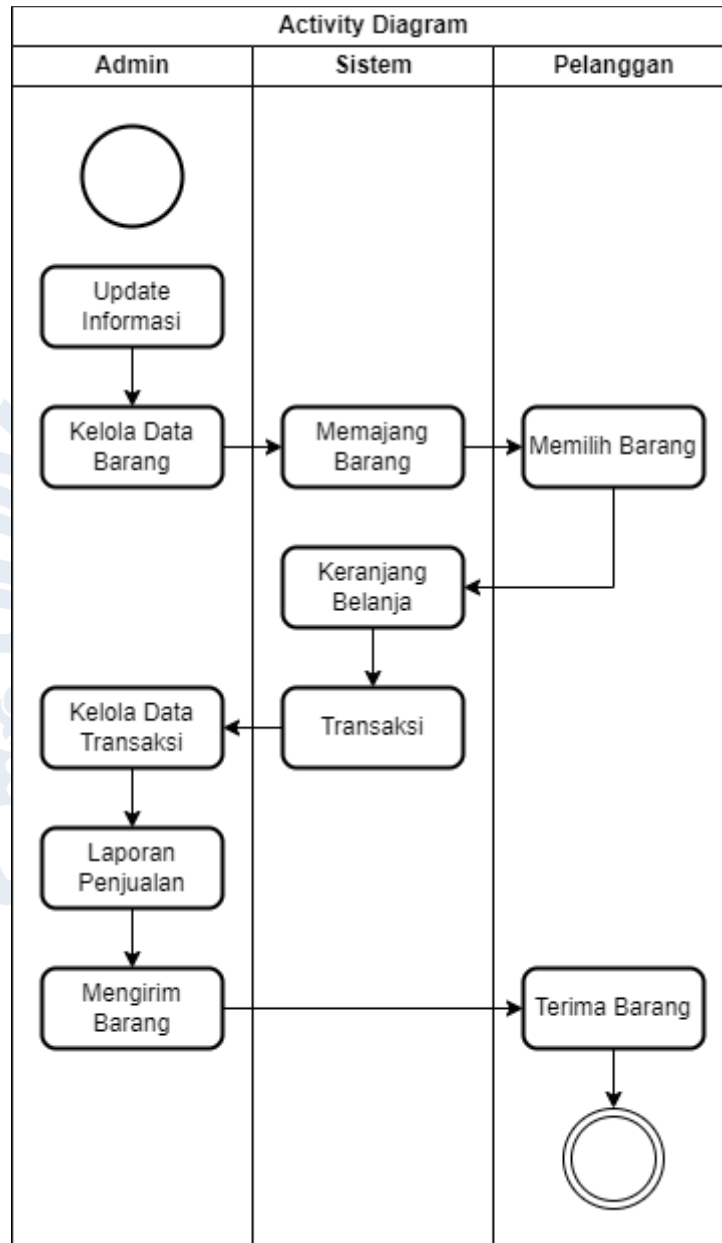
## HASIL PENELITIAN DAN PEMBAHASAN



Gambar 2. Use case Diagram

Diagram *use case* pada gambar 2 adalah suatu bentuk representasi visual yang digunakan untuk menggambarkan interaksi antara pengguna dan sistem yang sedang direncanakan atau dibangun[10]. Diagram ini memiliki tiga aktor (*user/customer*, *administrator*, dan sistem) dan beberapa kasus pengguna (*use case*)

yang mereka dapat melakukan. Diagram ini menunjukkan bahwa sistem memungkinkan *user/customer* dan *administrator* untuk melakukan berbagai tindakan dalam sistem, seperti *login*, menghapus item dari menu, melihat data transaksi, menambahkan produk, melihat produk, menghapus produk, dan memperbarui produk.



Gambar 3. Activity Diagram

Selanjutnya pada Gambar 3 merupakan proses perancangan dan implementasi sistem informasi penjualan Perangkat Selular berbasis Web di *Mobile Cell* menggunakan *Framework Laravel* dimulai dengan *user* melakukan *login* ke aplikasi. Setelah *login*, *user* dapat melihat daftar produk Perangkat Selular yang

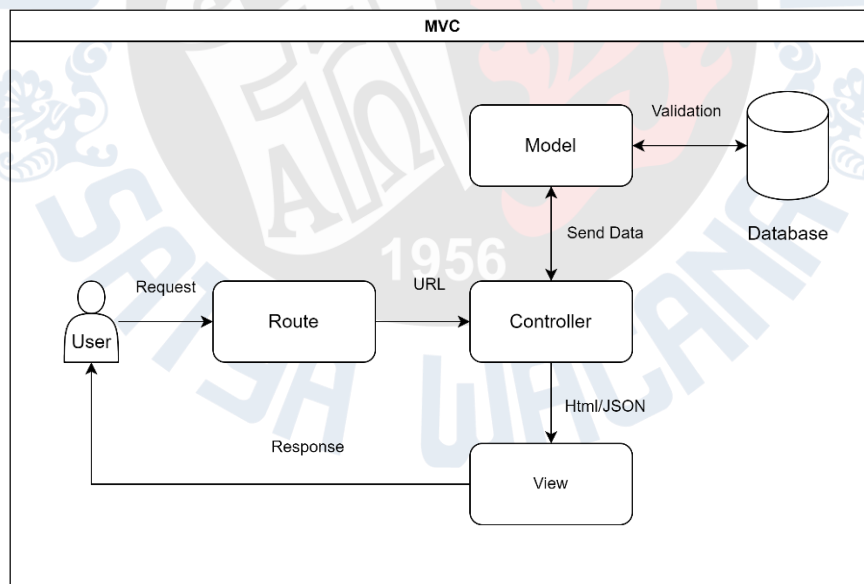
tersedia dan melakukan pembelian dengan menambahkan produk ke dalam keranjang belanja. Setelah selesai memilih produk, *user* dapat melakukan *Checkout* dan melakukan pembayaran melalui metode yang tersedia.

Setelah pembuatan *Use case* dan *Activity Diagram*, selanjutnya tahap pengembangan dan pengujian. Tahap pertama untuk membuat *Website* pada penelitian ini, *install Laravel* dengan cara menggunakan *Composer* sebelum melakukan konfigurasi *Laravel*. *Composer* adalah paket manajer *PHP* yang digunakan untuk mengelola dependensi *PHP*[11]. *Laravel* dapat dibuat dengan menggunakan *Composer* dengan perintah seperti pada Kode program 1.

#### Kode Program 1. Install *Laravel*

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

*Laravel Framework* menerapkan pola *MVC (Model-View-Controller)*, yang merupakan pola arsitektur yang umum digunakan dalam pengembangan aplikasi web. *Model* mengatur data dan logika aplikasi, *View* mengatur tampilan aplikasi, dan *Controller* mengatur interaksi antara *Model* dan *View*. Pola *MVC* mempermudah pengembangan aplikasi web dengan memisahkan logika aplikasi dari tampilan, yang memungkinkan pengembang untuk mengubah tampilan aplikasi tanpa mengubah logika aplikasi[12]. Namun, ada perbedaan dalam pola *MVC* yang digunakan pada *Laravel* dengan yang lainnya. Hal tersebut dapat dilihat pada gambar 4.



Gambar 4. Pola *MVC* pada *Framework Laravel*

Perbedaan terdapat pada *Route*. *Laravel* menggunakan *routing* yang menjembatani antara *request* dari *user* dan *controller*. Sehingga *Controller* tidak langsung menerima *request* tersebut. Hal ini memungkinkan untuk mempermudah pengaturan *routing* dan mengurangi kompleksitas dalam pengaturan *controller*.

Kode Program 2 merupakan kode program dari model *Product*. Model di *Laravel* adalah *class* yang digunakan untuk menginteraksi dengan tabel *database*. Model ini adalah *class* yang digunakan untuk menginteraksi dengan tabel *database Products*. Model ini menggunakan *Eloquent ORM (Object-Relational Mapper)* sebagai alat untuk menginteraksi dengan *database*. Pada baris dua, model menggunakan *trait HasFactory* untuk membuat *factory* untuk membuat record baru, serta *Sluggable* untuk membuat *slug* untuk atribut *name*. Model ini memiliki beberapa *method* dan relasi yang digunakan untuk menangani data produk. Pada baris empat sampai sebelas, *method* mengembalikan konfigurasi *sluggable* untuk atribut *slug*. Pada baris lima belas sampai tujuh belas, mengembalikan relasi dengan model *Category*. Baris lainnya merupakan *method* lain yang digunakan model *class Product*.

### Kode Program 2. Model *Product*

```

1 class Product extends Model{
2     use HasFactory, Sluggable;
3     protected $guarded = ['id', 'updated_at', 'created_at'];
4     public function sluggable(): array{
5         return [
6             'slug' => [
7                 'source' => 'name',
8                 'onUpdate' => true
9             ]
10        ];
11    }
12    public function getStatusAttribute(): string{...}
13    public function scopeActive($query){...}
14    public function scopeHasQuantity($query){...}
15    public function category(){
16        return $this->belongsTo(Category::class);
17    }
18    public function tags(){...}
19    public function media(): MorphMany{...}
20    public function firstMedia(): MorphOne{...}
21    public function reviews(){...}
22    public function approvedReviews(){...}
23    public function ratings(){...}
24    public function rate(){...}
25 }

```

Setelah model, kode program 3 merupakan *Controller Product*. *Controller* ini menggunakan model produk untuk menangani permintaan dari pengguna untuk menampilkan halaman tampilan produk berdasarkan *slug* yang diberikan. Pada baris lima, *Controller* ini memiliki *method show* yang digunakan untuk menampilkan detail produk dengan *slug* tertentu. Baris enam *method* ini menggunakan *with* untuk mengambil data relasi media, *category*, dan *tags* dari model produk. *Method* ini mengambil produk dari *database* dengan melakukan query dengan *method where* untuk mencari produk berdasarkan *slug* yang diberikan. Pada baris delapan dan sembilan, *Controller* juga menggunakan *withCount* dan *withAvg* untuk mengambil jumlah dan rata-rata nilai dari relasi *approvedReviews*. Selain itu, metode ini menggunakan *active* dan *hasQuantity* untuk mengambil produk yang aktif dan memiliki jumlah stok lebih dari 0 seperti yang ditunjukkan pada baris sepuluh sampai dua belas. Terakhir, baris enam belas

dan tujuh belas merupakan *method* mengembalikan *view frontend.Product.show* dengan mengirimkan produk dan relasi lainnya sebagai variabel.

### Kode Program 3. *Controller Product*

```
1 use App\Models\Product;
2 use Illuminate\Http\Request;
3
4 class ProductController extends Controller{
5     public function show($slug){
6         $Product = Product::with('media', 'category', 'tags')
7             ->where('slug', $slug)
8             ->withCount('media','approvedReviews')
9             ->withAvg('approvedReviews', 'rating')
10            ->active()
11            ->hasQuantity()
12            ->firstOrFail();
13
14        ...
15
16        return view('frontend.Product.show', compact('Product',
17            'relatedProducts'));
18    }
19 }
```

Setelah pembuatan model dan *Controller* produk langkah selanjutnya adalah pembuatan *Route* agar web dapat berjalan dengan baik. Kode program 4 merupakan kode program *Route* yang telah didefinisikan dalam *Framework Laravel*.

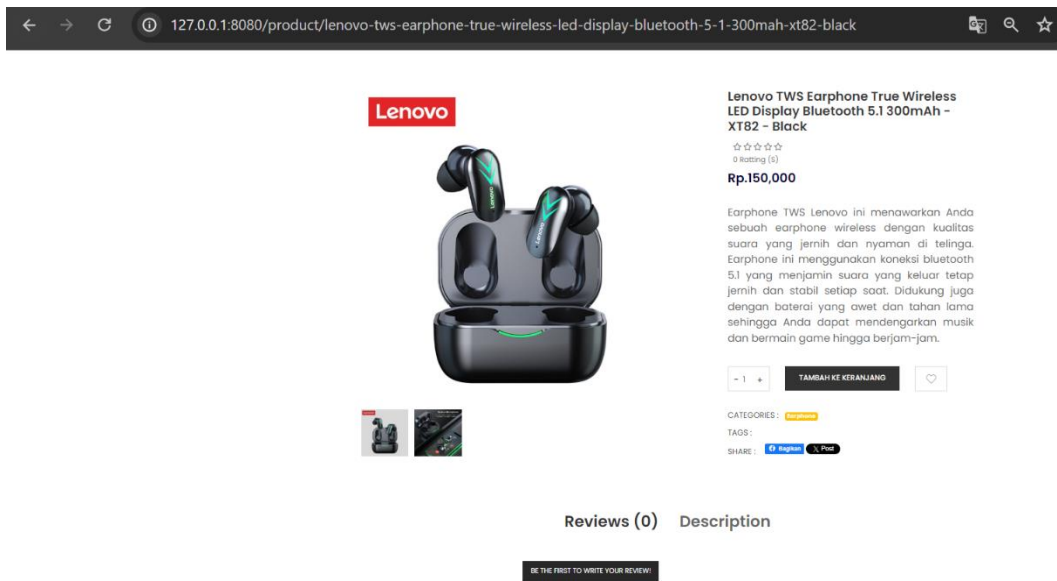
### Kode Program 4. *Route Product*

```
1 Route::get('product/{slug}',
2     [...\Controllers\ProductController::class, 'show']->name('product.show'));
3
```

Pada baris satu, *method* yang digunakan untuk mendefinisikan *method HTTP GET* dengan parameter yang akan digunakan untuk mengambil data produk berdasarkan *slug* yang diberikan dan menjadi bagian dari *URL* yang akan ditampilkan kepada *user*. Baris dua menunjukkan *Controller* yang digunakan untuk mengambil data produk berdasarkan *slug*. *ProductController* adalah nama *Controller* dan *show* adalah *method* yang digunakan.

Tahap terakhir dalam pengembangan *Website* adalah menguji *Route* yang di buat menggunakan *Google* dan *Xampp*. Pada *Google*, *request* dapat diterima dalam bentuk *HTTP request*, yang dapat berupa *method GET, POST, PUT, DELETE*, dan lainnya. *Xampp* digunakan untuk membuat koneksi ke *database MySQL*[13]. Dengan pengujian ini, mengonfirmasi bahwa setiap *Route* yang digunakan oleh aplikasi dapat diakses dan memberikan respons yang benar. Hal ini membantu memastikan ketersediaan dan keandalan *Laravel* untuk di implementasikan pada fitur lainnya. Pada gambar 5. melihat pengujian menggunakan *Google*.





Gambar 5. Hasil Pengujian pada *Google*

Dari hasil pengujian *request* data produk *response* berupa bentuk *HTML* dan sesuai dengan hasil yang diharapkan untuk di implementasikan dengan fitur lainnya. *Return value* yang dihasilkan berisi atribut data produk pada suatu tabel.

Salah satu keunggulan pengembangan aplikasi web menggunakan *Laravel* adalah penggunaan pola arsitektur *MVC* (*Model-View-Controller*) yang mempermudah pengembangan dan pemeliharaan aplikasi[14]. *Laravel* menerapkan pola *MVC* dengan cara yang berbeda dari yang lain, seperti yang telah dijelaskan. Hal ini mempermudah pengaturan *routing* dan mengurangi kompleksitas dalam pengaturan *controller*.

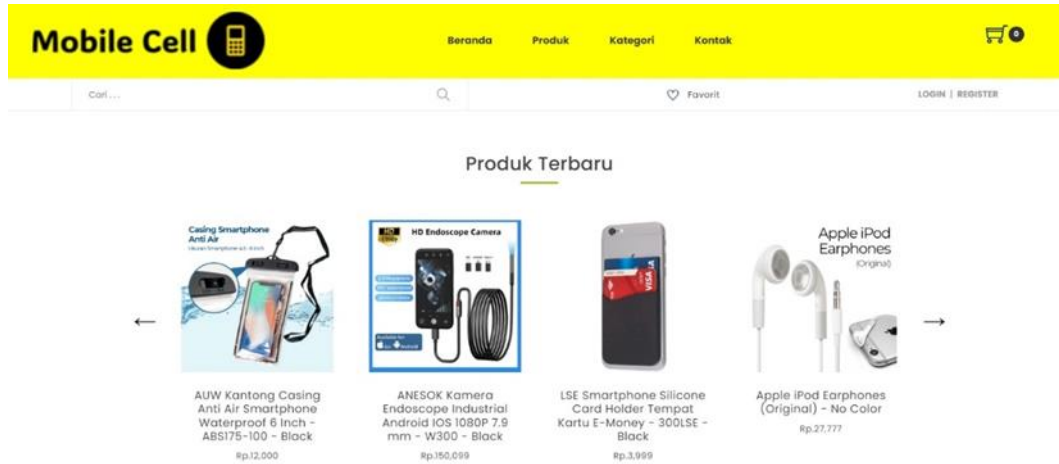
Implementasi *user interface* (*UI*) *Laravel* adalah proses mengatur tampilan dan interaksi pengguna dengan aplikasi web menggunakan *Framework Laravel*[15]. *User interface Laravel* adalah bagian dari pola arsitektur *MVC* (*Model-View-Controller*), yang memisahkan logika aplikasi dari tampilan. Implementasi *UI Laravel* menggunakan *blade template engine*, *HTML*, *CSS*, dan *JavaScript* dalam membuat tampilan dan interaksi pengguna[16]. Contoh *user interface* ditunjukkan pada gambar 6 dan 7.

The screenshot shows a 'Register' form with the following fields and a button:

- Username:** A text input field with a blue border and a cursor.
- Email Address:** A text input field.
- Password:** A text input field.
- Confirm Password:** A text input field.
- Register:** A blue button with white text.

Gambar 6. Tampilan Halaman *Register*

Gambar 6 merupakan tampilan halaman *Register* pada sistem penjualan perangkat seluler di *Mobile Cell*. *User* dapat membuat akun dengan mengisi formulir yang tersedia. *User* harus mengisi data diri seperti *username*, *email address*, *password*, dan *confirm password*. Setelah mengisi formulir, *user* dapat menekan tombol menu *Register* untuk membuat akun.



Gambar 7. Tampilan Halaman Beranda

Gambar 8 merupakan tampilan halaman beranda. Setelah berhasil login, user akan diarahkan ke halaman beranda yang berisi daftar produk Perangkat Seluler yang tersedia. User dapat melihat detail produk, seperti harga, spesifikasi, dan gambar. User dapat menambahkan produk ke dalam keranjang belanja dengan menekan tombol "Tambah ke keranjang" pada halaman beranda.

Pengujian sistem adalah proses untuk memastikan bahwa sistem perangkat lunak berfungsi sesuai dengan spesifikasi yang telah ditetapkan dan memenuhi kebutuhan pengguna[17]. Metode yang di gunakan adalah metode *black box*. Metode ini berfokus pada sisi fungsionalitas dari perangkat lunak, khususnya pada *input* dan *output* aplikasi. Pengujian ini dilakukan untuk menemukan kesalahan atau kekurangan dalam sistem, sehingga dapat diperbaiki sebelum diimplementasikan secara penuh.

Tabel 1. Hasil Pengujian

Fungsi	Deskripsi	Hasil yang di harapkan	Hasil Uji
Halaman <i>Register</i>	<i>Customer</i> melakukan Registrasi Akun	Dapat melakukan pembuatan akun dan data masuk ke <i>database</i>	Berhasil
Halaman <i>Login</i>	<i>Customer</i> melakukan <i>Login</i> Akun	Dapat melihat halaman profil dan data masuk ke <i>database</i>	Berhasil

Halaman Beranda	<i>Customer</i> memilih menu Beranda	Masuk ke halaman beranda dan melihat beberapa produk	Berhasil
Halaman Profil	<i>Customer</i> memilih menu Profil	Masuk ke halaman Profil	Berhasil
Halaman Keranjang Barang	<i>Customer</i> memilih menu Tambahkan ke Keranjang pada halaman Beranda dan masuk ke halaman Keranjang Barang	Masuk ke halaman Keranjang Barang yang memperlihatkan daftar produk yang di tambahkan	Berhasil
Halaman Detail Pembayaran	<i>Customer</i> memilih menu <i>Checkout</i>	Menampilkan halaman <i>Checkout</i> dan detail Pembayaran	Berhasil
Halaman Bayar	<i>Customer</i> memilih menu Bayar	Menampilkan halaman Bayar, melakukan pembayaran <i>online</i> , dan laporan transaksi	Berhasil
Halaman Order	<i>Customer</i> memilih menu Order	Masuk ke halaman Order untuk melihat riwayat pemesanan yang di simpan di <i>database</i>	Berhasil
Halaman Detail Pemesanan	<i>Customer</i> memilih menu Detail pada halaman Order	Menampilkan detail pemesanan	Berhasil

Tabel 1 menunjukkan hasil pengujian metode *Black Box Testing* pada beberapa halaman aplikasi web. Dari tabel tersebut, dapat disimpulkan bahwa semua halaman aplikasi berhasil diuji dan menghasilkan hasil yang diharapkan. Hal ini dapat disimpulkan dari persentase hasil pengujian yang semua berada pada 100% valid dalam setiap fungsinya.

## SIMPULAN DAN SARAN

Berdasarkan hasil penelitian dan pembahasan, sistem informasi penjualan perangkat seluler yang dikembangkan pada *Mobile Cell* telah melalui beberapa tahapan analisis dan implementasi. *Laravel* digunakan sebagai *Framework* untuk pengembangan aplikasi web penjualan perangkat seluler. *Laravel* menerapkan pola arsitektur *MVC (Model-View-Controller)*, yang mempermudah pengembangan dan pemeliharaan aplikasi. *Laravel* menggunakan *routing* yang menjembatani antara *request* dari *user* dan *controller*, yang mempermudah pengaturan *routing* dan mengurangi kompleksitas dalam pengaturan *controller*. *Laravel* menggunakan *Eloquent ORM (Object-Relational Mapper)* sebagai alat untuk menginteraksi dengan *database*.

Pengujian sistem dilakukan untuk memastikan bahwa fungsi-fungsi utama sistem berjalan sesuai dengan spesifikasi dan kebutuhan pengguna. Hasil pengujian menunjukkan bahwa semua fungsi utama sistem telah berhasil diuji dan berjalan dengan baik sesuai harapan. Hal ini mengindikasikan bahwa sistem telah siap untuk diimplementasikan secara penuh.

