

IMPLEMENTASI *FRAMEWORK* LARAVEL 7.1 PADA SISTEM INFORMASI PENJUALAN *CONVENIENCE STORE* EMMI SHOP

Bandhu Nusantara ^{*1)}, Sri Yulianto Joko Prasetyo ²⁾

1. Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia
2. Teknik Informatika, Teknologi Informasi, Universitas Kristen Satya Wacana, Indonesia

Article Info

Kata Kunci: Laravel; Metode *Waterfall*; Sistem Informasi Penjualan; *Web*

Keywords: Laravel; *Sales Information System*; *Waterfall Method*; *Web*

Article history:

Received 15 October 2024
Revised 11 November 2024
Accepted 3 December 2025
Available online 1 March 2025

DOI :

<https://doi.org/10.29100/jupi.v10i1.5584>

* Corresponding author.

Bandhu Nusantara

E-mail address:

672018161@student.uksw.edu

ABSTRAK

Studi kasus ini meneliti Emmi Shop untuk menerapkan *framework* Laravel pada sistem informasi penjualan berbasis *web* menggunakan metode *waterfall*. Dalam perkembangan teknologi, sistem informasi penjualan mampu untuk meningkatkan efisiensi dan kenyamanan bagi penjual maupun pelanggan. Analisis kebutuhan, desain, pengkodean, dan pengujian adalah metode yang digunakan dalam penelitian ini. Analisis kebutuhan mengumpulkan persyaratan pengguna dan spesifikasi perangkat lunak yang diperlukan. Setelah tahap desain, struktur data, arsitektur perangkat lunak, dan antarmuka pengguna dibuat. Selanjutnya, tahap pengkodean menyelesaikan desain dan menghasilkan kode program yang berfungsi dengan baik. Sistem diuji untuk mengurangi kesalahan dan memastikan kualitas sistem informasi. Dengan menggunakan implementasi *framework* Laravel, Emmi Shop dapat memanfaatkan sistem informasi penjualan yang dapat meningkatkan efisiensi proses transaksi dan pengolahan data. Sistem informasi ini memberikan manfaat bagi pemilik usaha dan pelanggan. Hasil penelitian ini diharapkan dapat membantu dalam proses pengembangan sistem informasi penjualan berbasis *web* yang menggunakan metode *waterfall*.

ABSTRACT

With a case study conducted at the Emmi Shop, this research intends to deploy a Laravel framework in a web-based sales information system utilizing the waterfall method. In technological developments, sales information systems are able to increase efficiency and comfort for sellers and customers. The research methods used in this research include needs analysis, design, coding and testing. Requirements analysis is carried out to collect user needs and determine the required software specifications. The design stage produces data structures, software architecture and user interfaces. The coding stage translates the design into program code that runs well. System testing is done to check the information system's quality and minimize faults. Emmi Shop can leverage a sales information system that can boost the effectiveness of data processing and transaction operations by implementing the Laravel framework. Customers and business owners can both profit from this information system. It is envisaged that this research's findings would help with the waterfall method's construction of an online sales information system.

I. PENDAHULUAN

PERKEMBANGAN teknologi informasi menjadi sebuah hal yang tidak bisa dihindari pada masa kini yang memungkinkan implementasi sistem informasi pada berbagai sektor. Sistem informasi sudah diterapkan di berbagai sektor, seperti sektor pendidikan, keamanan, kesehatan, hingga sektor komersial. Sistem informasi adalah kombinasi dari orang-orang, teknologi informasi, dan kumpulan prosedur yang sudah diatur yang dapat digunakan untuk membuat keputusan dan menjalankan operasional perusahaan [1]. Dalam pemanfaatan sistem informasi, hal ini tidak dapat bisa dilepaskan dengan internet karena Internet menyediakan berbagai kebutuhan informasi yang dibutuhkan. Sistem informasi membantu menjaga keamanan bisnis dan mengorganisasikan berbagai data. Seiring perkembangan teknologi, pengolahan data yang cepat dan tepat menjadi sangat penting. Akibatnya, sistem informasi sangat penting bagi bisnis dalam menjalankan proses bisnisnya. Sistem di dalam suatu organisasi yang menggabungkan kebutuhan untuk pengolahan transaksi harian, membantu operasi manajemen dan kegiatan

strategis, dan memberikan laporan kepada pihak luar tertentu [2]. Penjualan, juga disebut penjualan, adalah bisnis menjual produk atau jasa. Secara umum, penjualan adalah kegiatan jual beli yang dilakukan oleh dua belah pihak atau lebih dengan menggunakan alat pembayaran yang sah [3].

Emmi Shop merupakan salah satu *convenience store* (toko kelontong) di Kecamatan Tenganan, Kabupaten Semarang yang menjual beraneka ragam kebutuhan sehari-hari seperti gas elpiji, beras, gula pasir, minyak goreng, telur ayam, dan sebagainya. Pengolahan data yang dilakukan oleh pengelola Emmi Shop saat ini masih menggunakan sistem manual dengan sehingga banyak menimbulkan permasalahan diantaranya: duplikasi data, kesulitan dalam mendeteksi pengeluaran barang dan kesulitan dalam melihat data penjualan. Belum adanya laporan yang terintegrasi secara akurat juga menjadi permasalahan dalam kegiatan operasional bisnis sehari-hari. Pada saat ini perkembangan bisnis *convenience store* semakin meningkat dan persaingan semakin ketat. Hal tersebut membuat pengelola perlu merancang teknologi informasi yang cepat dan akurat agar dapat meningkatkan efisiensi bisnis. Dengan adanya sistem informasi penjualan, maka pekerjaan dapat lebih teratur, terarah dan terintegrasi dengan data sehingga dapat mendukung operasional harian di dalam bisnis. Penjualan merupakan salah satu aktivitas bisnis yang paling penting bagi Emmi Shop. Untuk meningkatkan pelayanan terhadap konsumen maka sistem informasi pengolahan data menjadi sangat penting untuk diimplementasikan. Dalam penelitian sebelumnya, sistem informasi penjualan berhasil dibangun menggunakan framework Laravel. Dengan menggunakan framework ini, proses transaksi penjualan dan laporan dapat dilakukan dengan lebih cepat, tepat, dan efektif [4].

Laravel 7.1 adalah versi dari kerangka kerja PHP untuk pengembangan sistem informasi menggunakan *web*. Kelebihan yang dimiliki oleh Laravel adalah sintak yang lebih ekspresif dan elegan. Laravel memberikan kebebasan kepada developer untuk berkreasi dalam menyelesaikan persoalan berkaitan dengan pengembangan sistem informasi. Dalam proses pengembangannya, kode laravel disediakan dengan sederhana dan dapat menghadirkan fungsionalitas di setiap fitur. Pengembang dapat membangun aplikasi *full-stack* yang tanggung dalam kerangka PHP dan JavaScript. *Starter kit* Laravel dapat membuat pengembang menjadi lebih produktif dalam membangun sistem informasi. Selain hal tersebut Laravel dipilih karena memiliki solusi elegan untuk fitur-fitur umum yang dibutuhkan oleh semua aplikasi *web* modern. Hasil penelitian sebelumnya menghasilkan perangkat lunak website toko online yang menggunakan framework Laravel. Fungsinya adalah untuk meningkatkan pemasaran produk dan memudahkan pelanggan untuk membeli produk dengan cara yang lebih mudah dan cepat tanpa harus pergi ke toko langsung [5].

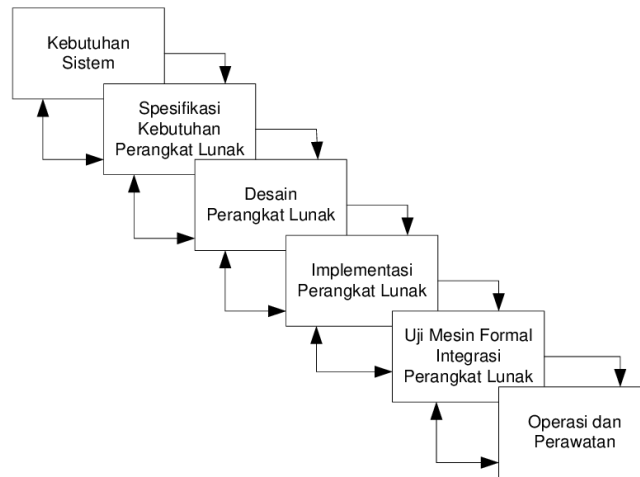
Dalam perkembangannya, sistem informasi memiliki sebuah peranan penting dalam pengolahan data dalam sebuah proses bisnis yang terjadi. Sebagai contoh dengan adanya berbagai sistem informasi penjualan, hal ini memungkinkan *customer* untuk dapat melakukan sebuah transaksi dengan internet tanpa harus datang langsung ke tempat atau ke toko. Hal itu tentu saja sangat memudahkan bagi penjual dalam melakukan transaksi penjualan. Dengan berbagai perkembangan yang ada, apabila sebuah usaha tidak dapat mengikuti proses perkembangan teknologi yang ada tentu saja akan mengakibatkan usaha tersebut menjadi kurang baik. Seperti halnya di Emmi Shop, yang sementara ini belum terdapat sebuah sistem informasi yang mendukung kegiatan transaksi. Dengan adanya sebuah sistem informasi penjualan yang akan dirancang dapat membantu proses pengolahan data informasi dan mempermudah seluruh proses transaksi yang ada. *Framework* Laravel memberikan fitur-fitur kuat dalam manajemen rute, pengguna, basis data, dan keamanan [6].

Framework Laravel dipilih karena memiliki sintaksis yang sederhana, ekspresif dan kode yang *clean*. *Framework* Laravel berkontribusi dalam membangun aplikasi *full-stack* yang tangguh dalam PHP. Library Laravel dapat mempercepat proses pembuatan aplikasi [7]. Laravel memiliki solusi elegan untuk fitur-fitur umum yang dibutuhkan oleh semua aplikasi *web* modern. Karena *syntax* Laravel yang bersih dan berfungsi serta *library* yang banyak dan mudah digunakan, *Framework* Laravel membantu *programmer* membangun website dengan cepat [8]. Hal ini sesuai dengan kebutuhan oleh bisnis Emmi Shop yang memerlukan aplikasi dengan pengujian terbaik. Model *waterfall* digunakan dalam pengembangan sistem. Model *waterfall* digunakan untuk *project* yang kompleks dan menggunakan pendekatan struktur yang berurutan pada pengembangan perangkat lunak. Selain itu, menggunakan *framework* Laravel mudah dan melakukan hosting dengan biaya yang relatif murah [9]. *Framework* Laravel dipilih karena *open source*, banyak digunakan, dan memiliki banyak komunitas [10].

II. METODE PENELITIAN

A. Metode Penelitian

Dalam membangun perangkat lunak, model air terjun, atau model air terjun, digunakan secara sistematis dan berurutan [11]. Analisis, desain, kode, pengujian, dan pemeliharaan model *waterfall* adalah bagian dari proses pembuatannya. Salah satu keunggulan model pengembangan *waterfall* adalah bahwa itu mudah digunakan dan dapat digunakan dalam proses pengembangan perangkat lunak. Perancangan yang menggunakan metode *waterfall* dalam pembuatan sistem informasi digunakan untuk pembuatan perangkat lunak, yang terdiri dari [12]:

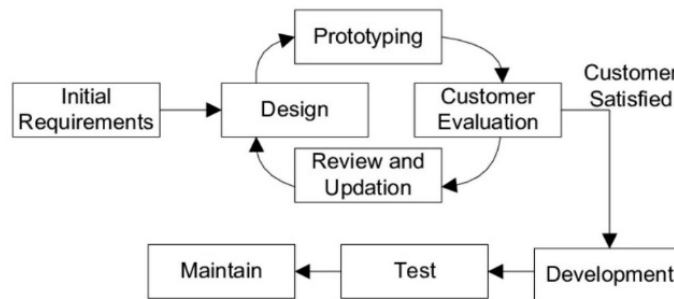


Gambar 1. Tahapan Penelitian Model *Waterfall* [11]

Karena metode *waterfall* tidak memungkinkan banyak revisi jika terjadi kesalahan dalam prosesnya, sulit untuk kembali lagi dan mengubah sesuatu yang tidak terdokumentasi dengan baik dalam tahap konsep sebelumnya setelah aplikasi di tahap pengujian.

B. Metode Perancangan Sistem

Pada Gambar 2 menggambarkan alur dari metode penelitian ini dengan menggunakan metode *Waterfall* sebagai berikut:



Gambar 2. Metode Perancangan Sistem

1. Analisis Kebutuhan

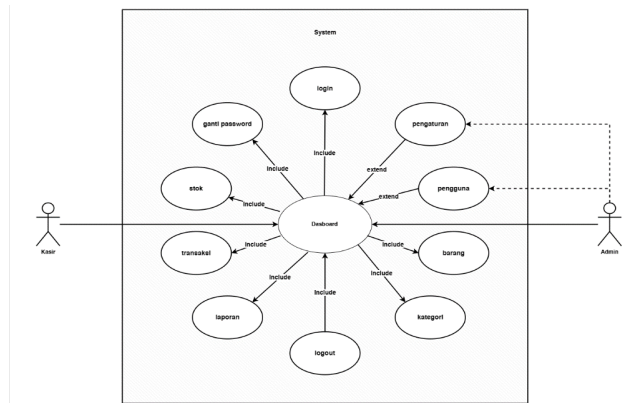
Pengumpulan kebutuhan yang diperkuat dilakukan melalui tahap analisis kebutuhan perangkat lunak. Untuk mengetahui kebutuhan perangkat lunak, seperti yang dibutuhkan oleh pengguna, spesifikasi kebutuhan perangkat lunak pada tahap ini harus didokumentasikan. Tujuan dari analisis kebutuhan adalah untuk mengumpulkan hal-hal yang diinginkan pengguna dan menemukan kebutuhan yang diperlukan untuk mengembangkan perangkat lunak. Setelah kebutuhan sistem dan perangkat lunak didokumentasikan, pengguna dapat menilai apakah kebutuhan tersebut telah dipenuhi atau tidak.

2. Desain

Tahap desain berfokus pada desain perangkat lunak, termasuk struktur data, arsitektur perangkat lunak, antarmuka pengguna, dan prosedur pengkodean. Setelah analisis kebutuhan perangkat lunak selesai, tahap desain dilakukan dengan menerjemahkan kebutuhan perangkat lunak ke dalam bentuk desain. Selanjutnya, desain didokumentasikan dan dimasukkan ke dalam konfigurasi perangkat lunak. Tahap ini menggunakan Unified Modeling Language (UML), yang mendefinisikan perangkat lunak yang akan dibuat. UML dibuat untuk membantu pengembang membuat perangkat lunak.

a. Use Case Diagram

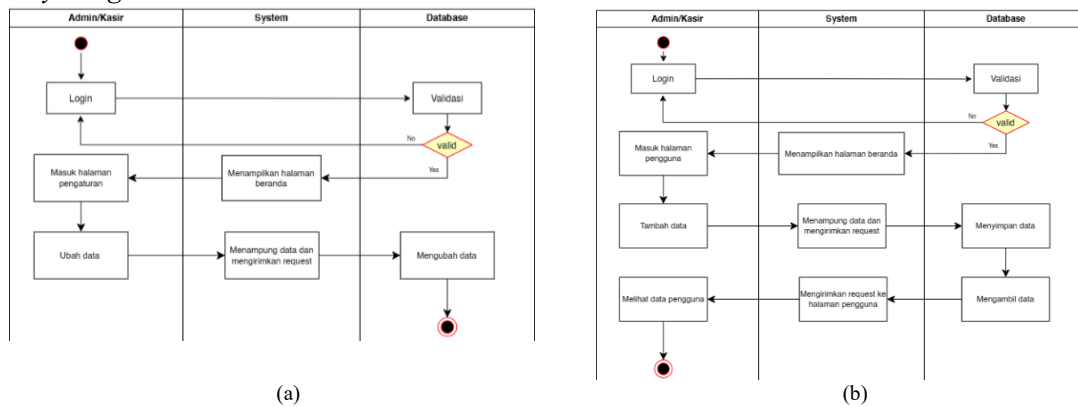
Representasi sematis yang dihasilkan tetap sederhana dan dimaksudkan untuk membuat informasi yang diberikan lebih mudah dibaca [13]



Gambar 3. Diagram Usecase

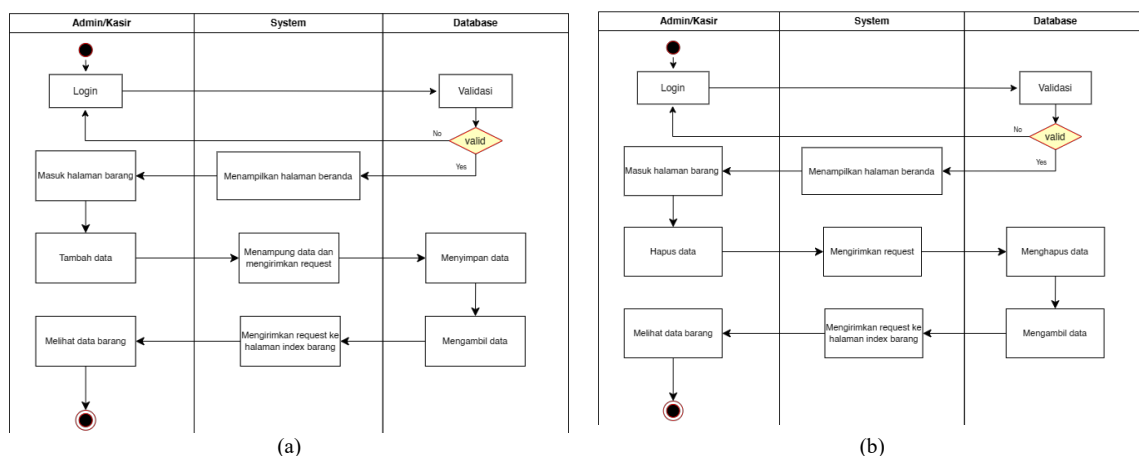
Gambar 3 menunjukkan dua orang yang berinteraksi dengan sistem aplikasi ini: admin (pengelola sistem) dan kasir (pengguna). Masing-masing dari mereka memiliki peran yang berbeda terhadap sistem, dan admin memiliki kemampuan untuk berinteraksi dengan sistem dalam berbagai cara seperti *login*, *logout* serta mengakses semua fungsi yang ada. Sedangkan kasir tidak dapat mengakses fungsi pengaturan dan *user*.

b. Activity Diagram



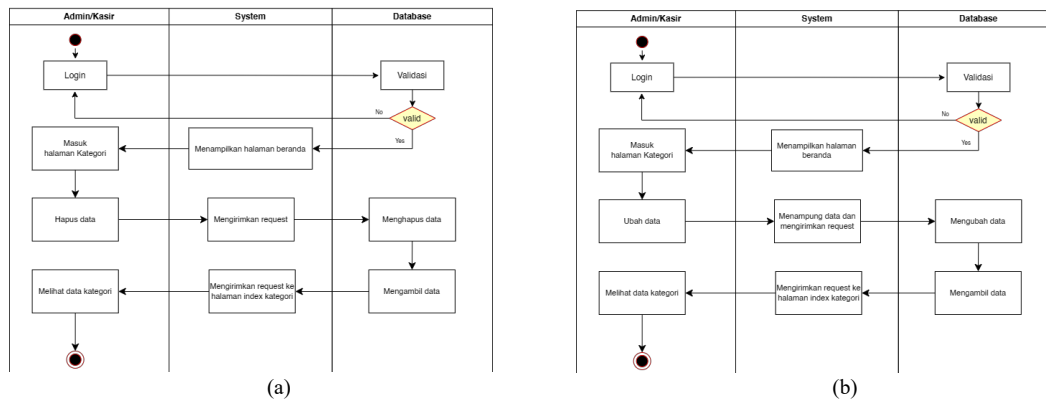
Gambar 4. Activity Diagram: (a) pengaturan.update, (b) pengguna.create

Gambar 4 (a) menjelaskan aktivitas validasi pengguna ketika akan mengubah data dan mengirimkan *request* kepada sistem. Gambar 4 (b) menunjukkan aktivitas *create* data pada sistem. Sistem secara otomatis dapat menyimpan menampilkan hasil *create* data di halaman.



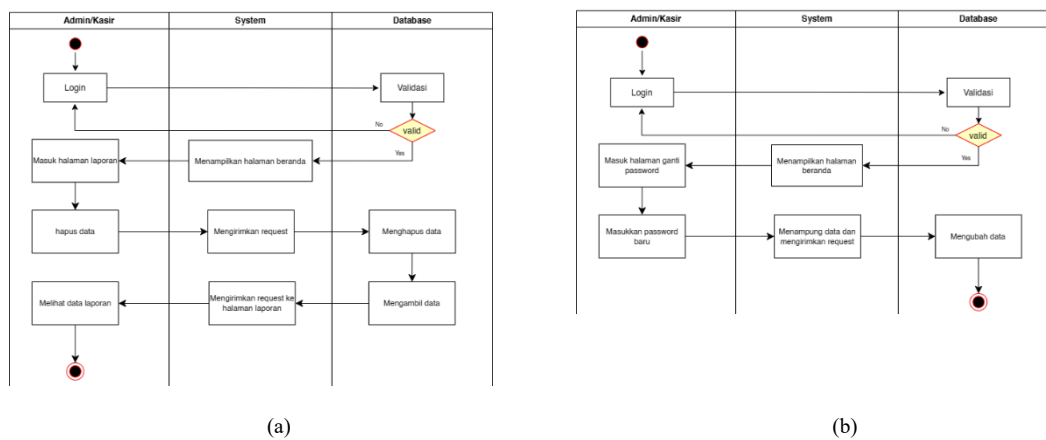
Gambar 5. Activity Diagram: (a) barang.create, (b) barang.delete

Gambar 5 (a) menjelaskan tentang fitur sistem *admin* atau kasir ketika hendak menambahkan barang baru di sistem. Pada gambar 5 (b), kasir juga dapat menghapus data barang dan secara otomatis sistem akan melakukan *update* pada tampilan manifes barang.



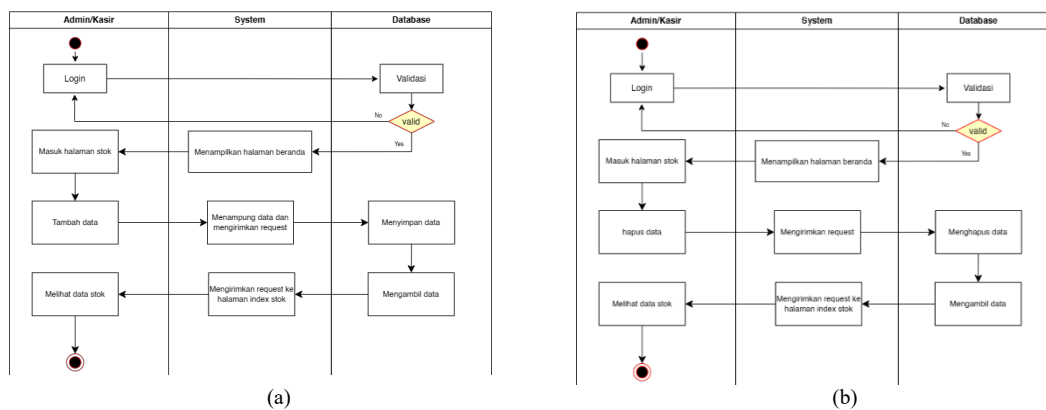
Gambar 6. Activity Diagram: (a) kategori.delete, (b) kategori.update

Gambar 6 (a) menjelaskan tentang aktivitas kasir ketika akan menghapus kategori barang. Pada gambar 6 (b), setelah melakukan validasi sistem akan melakukan *update* data yang telah diubah oleh *admin*.



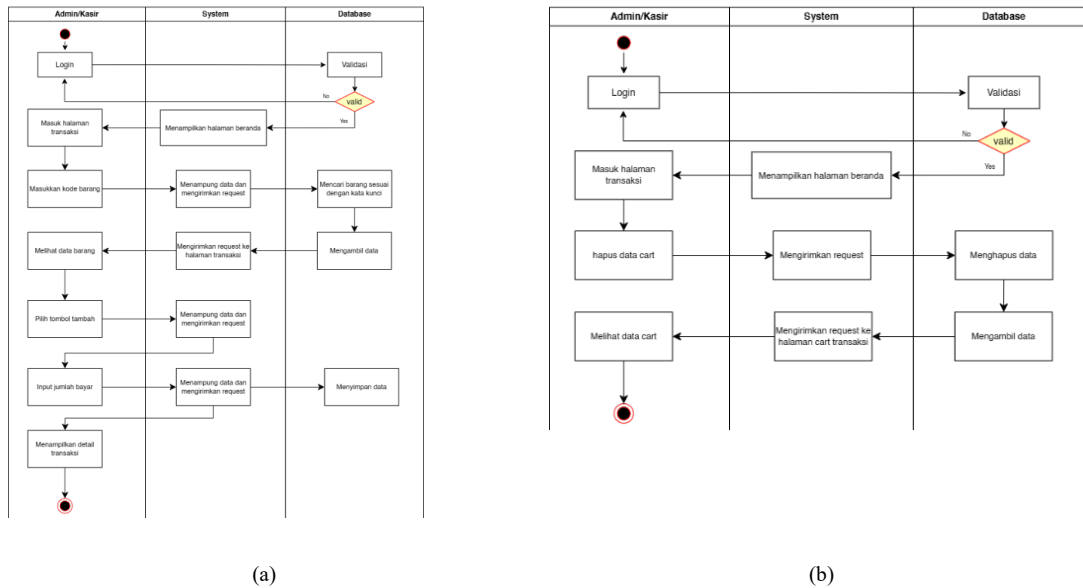
Gambar 7. Activity Diagram: (a) laporan.delete, (b) password.update

Gambar 7 (a), menjelaskan aktivitas sitem ketika akan hendak menghapus data laporan transaksi. Sistem juga memungkinkan dapat mengubah kata sandi pada profil pengguna seperti ditunjukkan pada gambar 7 (b).



Gambar 8. Activity Diagram: (a) stok.create, (b) stok.delete

Gambar 8 (a) menjelaskan aktivitas pengguna dalam membuat daftar stok barang di *inventory*. Selain menambah, sistem juga memungkinkan pengguna dalam menghapus stok barang di dalam *inventory* seperti ditunjukkan pada gambar 8 (b).



Gambar 9. Activity Diagram: (a) transaksi.create, (b) transaksi.delete-cart

Gambar 9 (a) menjelaskan proses alur bagaimana transaksi dikendalikan oleh admin dan diproses oleh sistem. Sistem juga memungkinkan admin dalam menghapus data transaksi jika terdapat kesalahan seperti ditunjukkan pada gambar 9 (b). Dengan menu pengolahan data transaksi yang tersedia dalam sistem, karyawan dapat mengolah data tanpa membuka aplikasi tambahan [14].

3. Pengkodean

Penerjemahan desain sistem ke dalam perangkat lunak dimulai dengan tahap desain. Sehingga desain dapat diterjemahkan dengan benar, pembuatan kode dapat dilakukan secara mekanis jika desain dilakukan dengan benar. Tahap ini berakhir dengan pembuatan sistem informasi yang sesuai dengan desain yang sudah dibuat.

4. Pengujian

Tahapan pengujian yang dilakukan antara lain:

a. Functionality Test

Tujuan dari pengujian ini adalah untuk memastikan bahwa semua fungsi dan fitur sistem informasi bekerja sebagaimana mestinya sesuai dengan spesifikasi yang telah ditentukan.

b. User Test

Pengujian ini dilakukan oleh pengguna sistem informasi yang mungkin untuk menilai antarmuka pengguna, kemudahan penggunaan, dan responsivitas sistem.

c. Performance Test

Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan sistem informasi dalam hal respons waktu, kecepatan akses, dan ketahanan terhadap beban pengguna yang besar.

d. Compatibility Test

Pengujian ini mengevaluasi kemampuan sistem informasi untuk berfungsi dengan baik di berbagai platform dan perangkat yang berbeda, seperti berbagai browser web dan perangkat seluler.

III. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah berupa purwarupa aplikasi kasir Emmi Shop yang memanfaatkan Laravel 7.1 sebagai kerangka kerja program.

A. Implementasi Arsitektur Model Laravel

Model berupa data dan logika bisnis. Model bertanggung jawab untuk berinteraksi dengan database dan merangkum operasi terkait data. Model berfungsi sebagai perantara antara aplikasi dan database, menyediakan antarmuka berorientasi objek untuk bekerja dengan data. ORM (*Object-Relational Mapping*) Laravel menyederhanakan interaksi database dalam Model. Dengan mendefinisikan model Eloquent, Operasi CRUD (*Create, Read, Update, Delete*) pada catatan database. Model menentukan hubungan antar model yang berbeda, sehingga memudahkan penanganan asosiasi data yang kompleks.

Kode Program 1. Kode Program *Model*

```

1. class Produk extends Model
2. {
3.     use HasFactory;
4.     protected $table = 'produk';
5.     protected $primaryKey = 'id_produk';
6.     protected $guarded = [];
7. }
    
```

B. Implementasi Arsitektur View Laravel : Rendering User Interface

View mewakili *User Interface* (UI) aplikasi. *View* bertanggung jawab untuk menampilkan data dan menerima komentar pengguna. *View* ditulis menggunakan mesin template *Blade*, yang menyediakan sintaksis yang bersih dan ringkas. *View* di Laravel adalah *file* terpisah yang berisi markup HTML beserta kode PHP yang disematkan. *View* dapat menerima data dari *Controller* dan menampilkannya menggunakan sintaks *Blade*. Tampilan juga mendukung kondisional, *loop*, dan parsial, sehingga memudahkan pembuatan UI modular dan interaktif.

Kode Program 2. Kode Program *View*

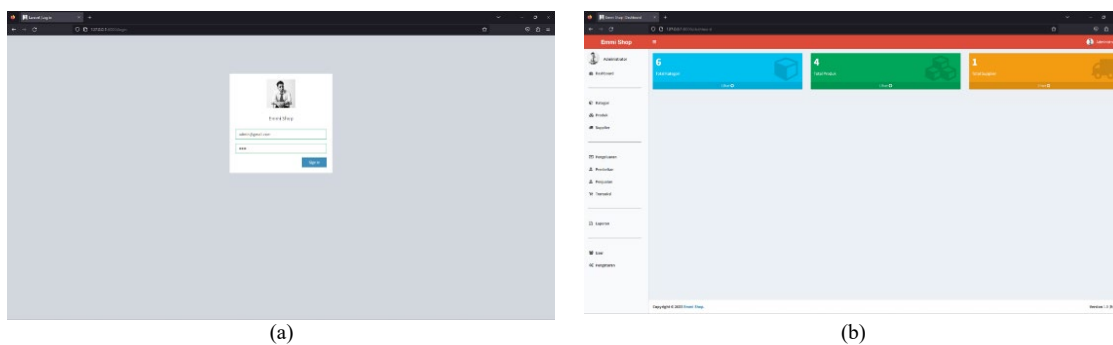
```

1. @section('content')
2. <div class="row">
3.     <div class="col-lg-12">
4.         <div class="box">
5.             <div class="box-header with-border">
6.                 <div class="btn-group">
7.                     <button onclick="addForm('{{ route('produk.store') }}')" 8.class="btn btn-
success btn-xs btn-flat"><i class="fa fa-plus-circle"></i> Tambah</button>
9.                     <button onclick="deleteSelected('{{ route('produk.delete_selected') }}')"
class="btn btn-danger btn-xs btn-flat"><i class="fa fa-trash"></i> Hapus</button>
10.                    <button onclick="cetakBarcode('{{ route('produk.cetak_barcode') }}')"
class="btn btn-info btn-xs btn-flat"><i class="fa fa-barcode"></i> Cetak Barcode</button>
11.                </div>
12.            </div>
13.            <div class="box-body table-responsive">
14.                <form action="" method="post" class="form-produk">
15.                    @csrf
16.                    <table class="table table-stiped table-bordered">
17.                        <thead>
18.                            <th width="5%">
19.                                <input type="checkbox" name="select_all" id="select_all">
20.                @includeIf('produk.form')
21.            @endsection
    
```

Kode Program 2 merupakan suatu template *view* pada *framework* Laravel yang menunjukkan suatu tabel daftar produk. Pada baris awal, terdapat *directive Blade* `@section('content')`, yang menunjukkan bahwa isi dari *view* tersebut akan disisipkan ke dalam *section* dengan nama *content*. Berikutnya, ada sebagian elemen HTML yang digunakan untuk menunjukkan tampilan tabel daftar produk, di antaranya tombol untuk menambah produk baru, menghapus produk yang dipilih, serta mencetak *barcode* untuk produk yang dipilih. Pada akhirnya, di dalam kode tersebut pula terdapat sebuah *directive Blade* `@includeIf('produk.form')`, yang akan memasukkan isi dari *file view* *produk.form.blade.php* ke dalam *view* tersebut. File-File tersebut berisi sebuah model form untuk menambah.

1) *View Login dan Dashboard*

Saat aplikasi ini berjalan, halaman *login* adalah antarmuka awal untuk admin dan kasir. Untuk masuk ke halaman *dashboard* admin dan kasir, Anda perlu memasukkan *email* dan *password* yang sudah dicatat oleh sistem.

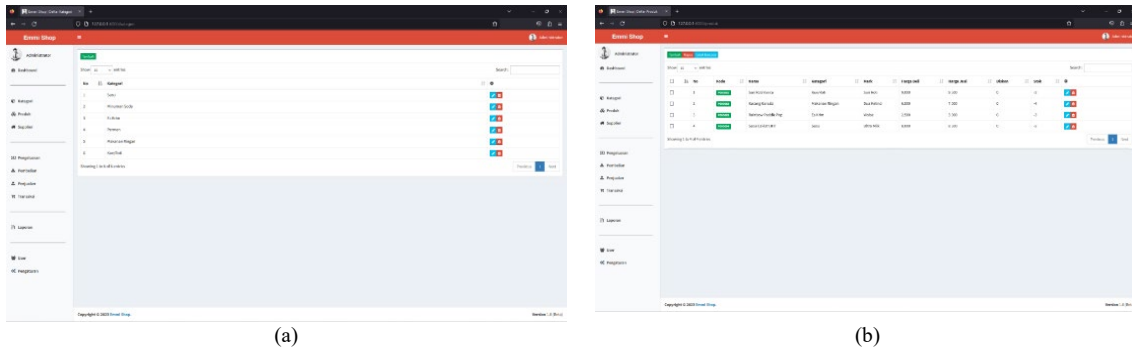


Gambar 10. Hasil Halaman (a) *Login*, (b) *Dashboard*

Gambar 10 menunjukkan tampilan *dashboard* administrator dan kasir setelah masuk. Masing-masing aktor menjalankan peran yang berbeda dalam sistem, dan administrator dapat berinteraksi dengan sistem dengan berbagai cara seperti *login*, *logout* serta mengakses semua fungsi yang ada. Sedangkan kasir tidak dapat mengakses fungsi pengaturan dan *user*.

2) View Kategori dan Produk

Berikut merupakan halaman kategori yang berfungsi untuk *add*, *edit* dan *delete* data kategori.

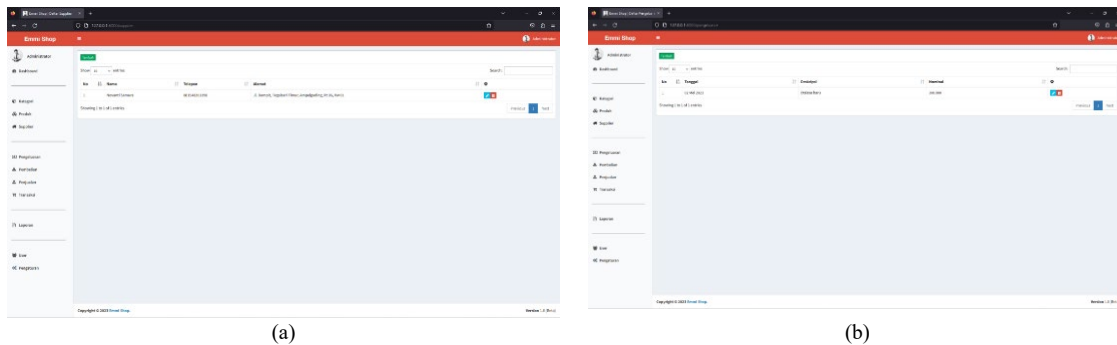


Gambar 11. Hasil Halaman (a) Kategori, (b) Produk

Pada halaman produk berfungsi untuk menampilkan daftar produk yang telah dibuat serta dapat melakukan edit, cetak *barcode* dan hapus berdasarkan data yang dipilih.

3) View Suplier dan Pengeluaran

Berikut merupakan halaman *supplier* yang berfungsi untuk *add*, *edit* dan *delete* data pemasok.

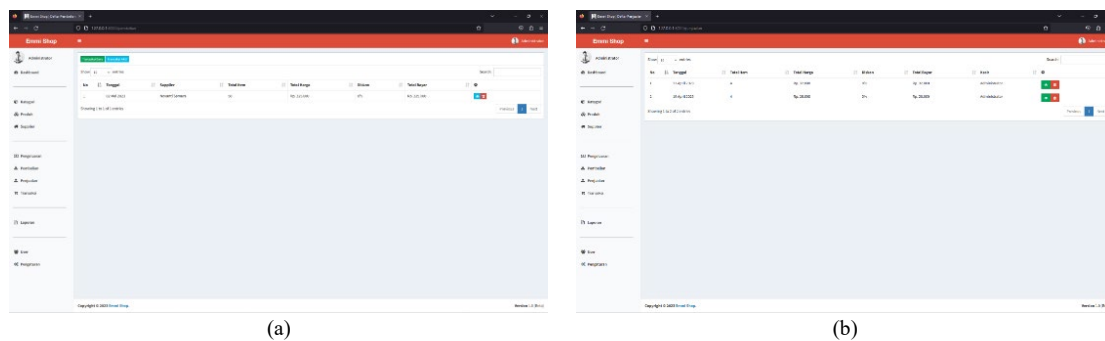


Gambar 12. Antarmuka: (a) *Supplier*, (b) Pengeluaran

Halaman pengeluaran digunakan untuk untuk *add*, *edit* dan *delete* semua data transaksi pengeluaran.

4) View Pembelian

Berikut pada halaman pembelian berfungsi untuk *add*, *edit* dan *delete* data pembelian barang dari *supplier*.

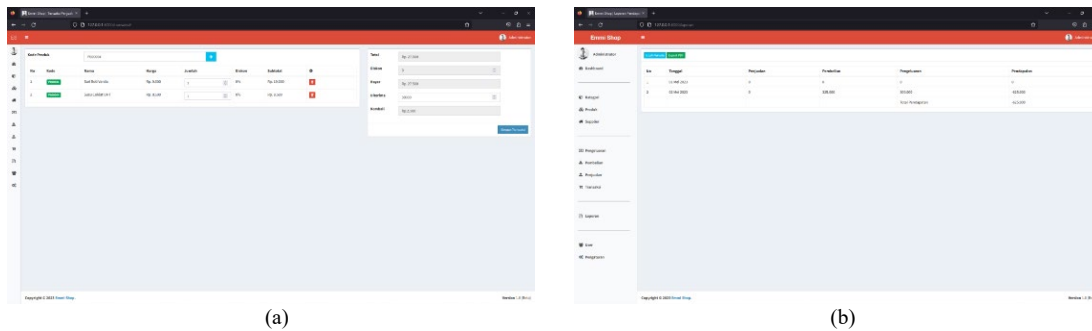


Gambar 13. Hasil Halaman: (a) Pembelian, (b) Penjualan

Pada halaman penjualan berfungsi untuk melihat dan menghapus data penjualan.

5) View Transaksi dan Laporan

Pada halaman transaksi berfungsi untuk menghitung harga barang yang dibeli oleh *customer*.

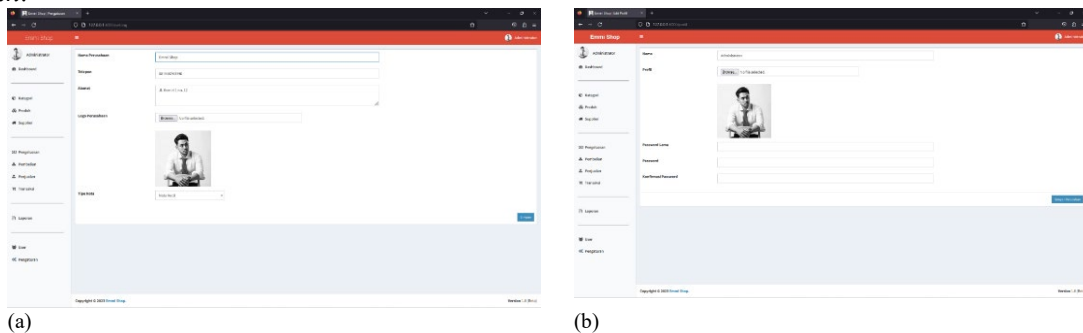


Gambar 14. Hasil Halaman: (a) Transaksi, (b) Laporan

Pada halaman laporan berfungsi untuk mendata semua transaksi yang ada mulai dari penjualan, pembelian serta pengeluaran.

6) View User dan Pengaturan

Pada halaman *user* berfungsi untuk menambah, mengubah dan menghapus data *user* atau kasir oleh *admin*.



Gambar 15. Hasil Halaman: (A) Pengaturan, (B) Edit Profil

Pada halaman pengaturan digunakan untuk membuat profil dari toko Emmi Shop. Berikut merupakan halaman edit profil yang digunakan untuk mengubah profil dari toko Emmi Shop.

C. Implementasi Arsitektur Controller Laravel : Menangani User Request

Controller bertindak sebagai penghubung antara *Model* dan *View*. *Controller* menerima dan memproses permintaan pengguna, berinteraksi dengan *Model* untuk mengambil atau mengubah, dan meneruskan data diproses ke *View* yang sesuai untuk dirender. Pada Laravel, *Controller* bertanggung jawab untuk mendefinisikan metode, atau *action*. *Action* merangkum logika yang diperlukan untuk menangani permintaan spesifik dan mengelola aliran data. *Controller* juga memfasilitasi validasi masukan pengguna dan melakukan pemeriksaan otorisasi yang diperlukan sebelum berinteraksi dengan *Model*.

Kode Program 1. Kode program *controller*

```

1. class ProdukController extends Controller
2. {
3.     public function data()
4.     {
5.         $produk = Produk::leftJoin('kategori', 'kategori.id_kategori',
        'produk.id_kategori')
6.         ->select('produk.*', 'nama kategori')
7.         ->orderBy('kode_produk', 'asc')
8.         ->get();
9.         return datatables()
10.        ->of($produk)
11.        ->addIndexColumn()
12.        ->addColumn('select_all', function ($produk) {
13.
20.     }
21. }
    
```

Dalam Laravel, ada *routing* yang menjembatani permintaan user dan pengontrol. Dengan demikian, pengontrol tidak langsung menerima permintaan dari *user* [15]. Kode Program 3 merupakan bagian dari *controller ProductController* yang berperan untuk mengambil informasi produk dari database serta menampilkannya pada suatu tabel yang menggunakan *plugin DataTables* pada tampilan *web*. Data produk yang diambil telah di-join dengan tabel kategori untuk memperoleh data nama kategori. Selain itu, pada kolom-kolom tertentu di dalam tabel ditambahkan fitur khusus seperti *checkbox* untuk hapus data, format uang, serta tombol aksi untuk edit dan hapus data. Data yang ditampilkan di dalam tabel bersifat dinamis serta akan otomatis terupdate ketika terdapat perubahan pada data produk di dalam database.

D. Pengujian Sistem

1. Functionality Test

Dalam pengembangan perangkat lunak, pengujian *Alpha* adalah strategi pengujian perangkat lunak yang khusus digunakan untuk mengembangkan aplikasi agar sistem yang dibuat tidak mengalami masalah atau kegagalan saat digunakan. *Alpha test* dilakukan secara internal melalui dokumentasi metode pengembangan *prototype* [16].

TABEL 1
 UJI BLACK BOX

No.	Item Kontrol	Expected	Revisi 1	Revisi 2	Displayed	Kesimpulan
1.	Pengguna	Pengaturan create dan update	error	Webserver dipulihkan	Aktivitas <i>create</i> dan <i>update</i>	Valid
2.	Barang	Manajemen barang di sistem	Manajemen barang tidak ditemukan	Informasi untuk mengaktifkan get data barang di database	Crud barang di sistem	valid
3.	Kategori	Validasi kategori barang di sistem	-	-	Crud kategori produk di sistem	valid
4.	Laporan	Manajemen laporan transaksi	-	-	List data laporan	valid
5.	Stok	Daftar stok barang di <i>inventory</i>	-	-	List data stok barang di sitem	valid
6.	Transaksi	Alur pengendalian transaksi oleh admin	-	-	Menampilkan data transaksi dari berbagai parameter	valid

Pada pengembangan penelitian ini beberapa kali ditemukan cacat temuan kemudian penulis melakukan revisi produk hingga dua kali.

2. Pengujian Pengguna

Pengujian selanjutnya adalah pengujian Beta, yang menggunakan kuesioner yang berisikan empat aspek kualitas perangkat lunak menurut ISO 9126: fungsionalitas, reliabilitas, usability, efficiency, maintainability, dan portability. Pengujian beta diberikan kepada 30 responden untuk menilai kelayakan perangkat lunak berdasarkan aspek kualitas perangkat lunak yang telah ditentukan. Fungsionalitas, *reliabilitas*, *usability*, *efficiency*, *maintability*, dan *portability* adalah semua elemen yang termasuk dalam hasil pengujian. Skala Likert digunakan untuk menganalisis data ini. [16].

TABEL 2
 HASIL UJI BETA

No.	Quality	Jumlah Item	$\sum \bar{x}$	Std. Dev	Mode
1.	Fungsionalitas	5	121	0,345	Sesuai
2.	Reliabilitas	3	116	0,378	Sesuai
3.	Kegunaan	4	120,25	0,297	Sesuai
4.	Efisiensi	2	115	0,461	Sesuai
5.	Pemeliharaan	4	119,75	0,325	Sesuai
6.	Portabilitas	4	115,75	0,540	Sesuai

Pada tabel 2, masing-masing aspek kualitas aplikasi memiliki jumlah item indikator yang berbeda. Aspek fungsionalitas menerima nilai tertinggi sebesar 121, dan aspek efisiensi menerima nilai terendah sebesar 115. Secara keseluruhan, aspek kualitas ISO/IEC 9.126 dihasilkan pada nilai persepsi 4, yang sesuai dengan skala likert 1–5.

3. Performance Test

Tujuan dari pengujian ini adalah untuk mengevaluasi kemampuan sistem informasi dalam hal respons waktu, kecepatan akses, dan ketahanan terhadap beban pengguna yang besar.

Load Test

Pengujian untuk mengevaluasi kemampuan untuk mengisi situs web untuk mengetahui jumlah beban yang dikirimkan ke *database* atau *server*.

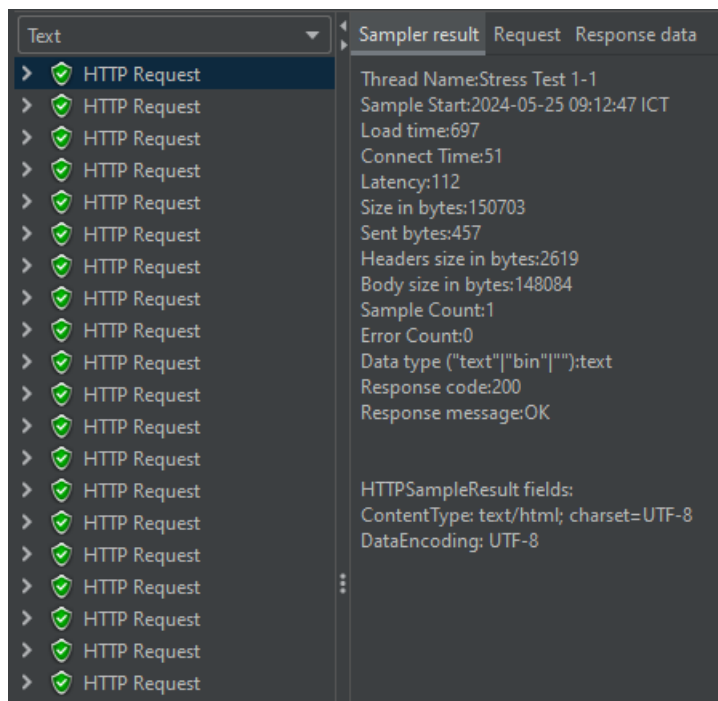
TABEL 3
 THREAD LOAD TEST

Sample	Sample Time (ms)	Bytes	Sent Bytes	Latency	Connect Time (ms)
1	2419	225796	361	105	41
2	2757	225783	361	228	152
3	2156	225783	361	125	47
4	2368	225783	361	105	40
5	1966	225779	361	142	59

Jumlah respon dari *website* diuji coba sebanyak 5 user. *Sample time* adalah waktu yang dibutuhkan oleh sebuah data dari komputer ke *webiste*, kemudian dari *website* merespon dan mengirim data ke *user*. *Latency* merupakan saat dari respon user diterima oleh *web server*, diolah kemudian dikirim kembali

Stress Test

Pengujian tekanan digunakan untuk memeriksa kemampuan aplikasi untuk menerima trafik dari luar sehingga *website* tidak *down* saat banyak pengguna mengunjunginya.



Gambar 16. Hasil *Tree Stress Test*

Pengujian dilakukan dengan jumlah pengunjung dalam satu periode sebanyak 20 user. Ramp-up period (jangka waktu) setiap periode yang digunakan 20 seconds. *Loop count* (jumlah pengulangan) sebanyak satu kali. Telah diperiksa *particular element* pada *website* dengan menambahkan *response assertion* dengan *pattern to test* dengan nilai 200. Status telah berwarna hijau yang menunjukkan bahwa *website* telah berjalan dengan dengan baik.

Spike Test

Target *concurrency* yang digunakan sebanyak 50 user dan *Hold Target Rate Time* sebesar 5 menit.

TABEL 4
 AGGREGATE REPORT

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Throughput	Received KB/sec	Sent KB/sec
HTTP Request	523083	483	226	948	1319	2725	1	24125	780.5/sec	1684.49	12.52

Pengujian Spike menggunakan peningkatan tiba-tiba dalam jumlah pengguna untuk mengukur kinerja sistem. Tujuan utama dari pengujian ini adalah untuk mengetahui apakah sistem akan dapat mempertahankan kinerjanya saat ini.

4. Compatibility Test

Pengujian ini mengevaluasi kemampuan sistem informasi untuk berfungsi dengan baik di berbagai *platform* dan perangkat yang berbeda, seperti berbagai *browser web* dan perangkat seluler.

This tab shows pages that exhibit browser-specific behavior, or trigger browser bugs.

Browser	Edge	Firefox	Safari	Opera	Chrome	iOS	Android		
Version	124	124	17	109	124	≤ 16	17	120	
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	● Missing content or functionality
Major Issues	✓	✓	⚠	✓	✓	⚠	⚠	✓	⚠ Major layout or performance problems
Minor Issues	✓	✓		✓	✓			✓	⚠ Minor layout or performance problems

Priority	Description and URL	Guideline and Line#	Count
Priority 2	1 issues on 1 pages		
▶	The CSS <code>backdrop-filter</code> property is not supported by some browsers.	Safari iPhone/iPad	1 pages
▶	Expand all 1 issues		

Gambar 17. Compatibility Report

Uji kompatibilitas diperiksa untuk mengetahui kerusakan tautan, kesalahan ejaan, kompatibilitas browser, dan standar *website*. Hasil uji kompatibilitas tidak ditemukan konten dan fungsionalitas yang hilang. Kekurangan sistem yaitu properti *css backdrop-filter* belum didukung untuk beberapa *browser* pada versi iOS.

E. Pembahasan Hasil Penelitian

Menurut penelitian yang telah dilakukan, Emmi Shop, sebuah convenience store, memiliki aplikasi sistem informasi penjualan. Situs web ini menggunakan framework Laravel, yang sangat populer dan disukai oleh programmer di seluruh dunia. Ini ringan dan mudah diakses dan mengadopsi konsep MVC. [17]. Hasil penelitian menunjukkan hubungan dengan sistem [18] yang dibangun menggunakan metode waterfall, framework PHP dengan laravel, dan database MySQL untuk mengolah basis datanya. Penelitian sebelumnya menggunakan objek sistem informasi Zakat Infak Sedekah (ZIS), sedangkan pada penelitian ini berfokus pada sistem informasi penjualan. Hasil penelitian menunjukkan bahwa framework Laravel memungkinkan pembuatan sistem informasi penjualan yang mudah karena menyediakan plugin yang membantu membangun sistem yang kompleks. Selain itu, karena Laravel memisahkan desain database dari desain tampilan, perancangan database tidak menjadi masalah. Pada sistem informasi penjualan penelitian yang dilakukan sebelumnya [19] juga diketahui memiliki tiga elemen penting: halaman pembayaran, dashboard admin, dan menu cart dan login. Setiap fungsi sistem diuji langsung oleh pengguna. Ada sekitar sepuluh pengguna yang mengikuti metode pengujian black box. studi sebelumnya [20] dirancang situs web dengan mempertimbangkan peran etalase dan manajemen stok barang. Agar website katalog online aksesoris handphone berfungsi sebagai etalase dan stok ketersediaan barang dagangan, maka fitur etalase meliputi nama produk, harga produk, stok produk, chat kontak, gambar produk, rating produk, dan tag produk. Fitur stok administrasi meliputi menambah produk ke list produk, katalog produk, dan tag produk. Pada penelitian ini pengujian dilakukan dengan aspek fungsionalitas (*black box*), pengujian pengguna (uji beta), pengujian kinerja (*load test*, *stress test* dan *spike test*) dan pengujian kompatibilitas.

IV. KESIMPULAN DAN SARAN

A. Kesimpulan

Laravel memberikan kebebasan kepada developer untuk berkreasi dalam menyelesaikan persoalan berkaitan dengan pengembangan sistem informasi. Dalam proses pengembangannya, kode laravel disediakan dengan sederhana dan dapat menghadirkan fungsionalitas di setiap fitur. Pengembang dapat membangun aplikasi *full-stack* yang tanggung dalam kerangka PHP dan JavaScript. *Starter kit* Laravel dapat membuat pengembang menjadi lebih produktif dalam membangun sistem informasi. Hal tersebut membuat teknologi informasi yang dirancang menjadi lebih cepat dan akurat dan dapat meningkatkan efisiensi bisnis. Dengan adanya sistem informasi penjualan, maka pekerjaan dapat lebih teratur, terarah dan terintegrasi dengan data sehingga dapat mendukung operasional harian di dalam bisnis.

B. Saran

Dalam menyelesaikan Tugas Akhir beberapa gagasan muncul sebagai suatu saran yang dapat menjadikan implementasi *framework* laravel pada sistem informasi penjualan menggunakan metode *waterfall* berbasis *web* menjadi lebih baik, diantaranya penambahan fitur *membership* serta pengembangan UI/UX yang lebih baik sehingga memudahkan pengguna dalam mengoperasikannya.

DAFTAR PUSTAKA

- [1] E. Y. Anggraeni, Pengantar sistem informasi, Yogyakarta: Penerbit Andi, 2017.
- [2] H. H. Solihin dan N. A. Fuja, "Rancang Bangun Sistem Informasi Penjualan, Pembelian Dan Persediaan Suku Cadang Pada Bengkel Tiga Putra Motor Garut," *Infotronik : Jurnal Teknologi Informasi Dan Elektronika*, vol. 2, no. 2, p. 107, 2017.
- [3] R. Syabania dan N. Rosmawani, "Perancangan Aplikasi Customer Relationship Management (CRM) pada Penjualan Barang Pre-Order Berbasis Website," *Rekayasa Informasi*, vol. 10, no. 1, pp. 44-49, 2021.
- [4] Y. Siyanto, J. Triyanto dan M. R. Alwatoni, "Implementasi Framework Laravel Dalam Perancangan Sistem Informasi Penjualan UKM Kota Surakarta," dalam *Seminar Nasional Ilmu Sosial dan Teknologi (SNISTEK)*, Surakarta, 2023.
- [5] Y. K. D. Prasetyo dan E. Sedyono, "Perancangan dan Implementasi Sistem Informasi Penjualan Berbasis Web Menggunakan Framework Laravel," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 10, no. 3, pp. 191-202, 2023.
- [6] A. Ratino, R. Astri dan P. Anggraini, "Implementasi Framework Laravel Dalam Pengembangan Aplikasi E-Commerce Untuk Toko Jago Software," *Journal of Informatics and Business*, vol. 01, no. 02, pp. 33-43, 2023.
- [7] M. A. Firmansyah, N. Ramsari dan A. D. Rachmanto, "Rancang Bangun Sistem Informasi Penjualan pada Toko Buku Kita Tasikmalaya Berbasis Web Menggunakan Framework Laravel 8," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 12, no. 1, pp. 26-38, 2022.
- [8] A. Fauziah, S. Noer, J. Junaedi dan M. Riyandi, "Sistem Penjualan Sayur Menggunakan Framework Laravel," *Jurnal Restikom : Riset Teknik Informatika dan Komputer*, vol. 3, no. 1, pp. 42-50, 2021.
- [9] K. Nugroho, Sumardi, Murdowo dan Muljono, "Mobile Cloud Learning System Using Laravel Framework and Android Studio Web View," dalam *International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, 2019.
- [10] D. P. Sari dan R. Wijanarko, "Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus Di Rumah Kamera Semarang)," *Informatika dan RPL*, vol. 2, no. 1, pp. 32-36, 2019.
- [11] R. S. Pressman, *Rekayasa Perangkat Lunak : Pendekatan Praktisi*, Yogyakarta: Andi Offset, 2015.
- [12] Lasimin, A. B. Haq dan Verry, "Sistem Informasi Penggajian PT. Kalisha Utama Ghani Cilacap Menggunakan Framework Laravel," *Information Management for Educators and Professionals*, vol. 4, no. 2, pp. 153 - 162, 2020.
- [13] H. B. Fansury dan M. A. Rahman, *Developing Mobile English Application As Teaching Media*, Yogyakarta: Deeppublish, 2021.
- [14] C. Sujana dan Darmansyah, "Analisa Dan Perancangan Sistem Penjualan Barang Berbasis Web Pada PT. Asia Tiara," *Jurnal Iterkom*, vol. 12, no. 4, pp. 33-39, 2018.
- [15] A. L. Yudanto, H. Tolle dan A. H. Brata, "Rancang Bangun Aplikasi Sistem Informasi Manajemen Laboratorium Biomedik Fakultas Kedokteran Universitas Brawijaya," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 1, no. 8, p. 628-634, 2017.
- [16] M. Shi, "Software Functional Testing from the Perspective of Business Practice," *Computer and Information Science*, vol. 3, no. 4, pp. 49-52, 2010.
- [17] R. M. Noer dan Y. Anistyasari, "Pembuatan Siroline (Sistem Informasi Rapor Online) Berbasis Web Menggunakan Framework Laravel," *Jurnal IT-Edu*, vol. 03, no. 02, pp. 35-43, 2019.
- [18] Asroni, A. Setiawan dan B. D. Alfian, "Implementasi Framework Laravel Pada Sistem Informasi ZIS," dalam *PROSIDING SEMNAS PPM 2020: Inovasi Teknologi dan Pengembangan Teknologi Informasi dalam Pemberdayaan Masyarakat Pasca Covid-19*, Yogyakarta, 2020.
- [19] T. C. Zulkhaidi, Yulianto dan Suswanto, "Implementasi Sistem Informasi Penjualan Produk Elektronik Berbasis Web Dengan Menggunakan Laravel Framework," *Buletin Poltanesa*, vol. 20, no. 2, pp. 51-56, 2019.
- [20] R. Priambodo, N. A. Prasetyo dan I. Susanto, "Perancangan Website Katalog Online Aksesoris Handphone Berbasis Framework Laravel Menggunakan Metode Rapid Application Development," dalam *Seminar Nasional Inovasi dan Pengembangan Teknologi Terapan (SENOVTEK)*, Cilacap, 2022.