

## Pengembangan Sistem Pelacakan Air Galon Isi Ulang dengan Pemanfaatan Google Maps API

### Development of Refillable Gallon Water Tracking System Using Google Maps API

Nur Azizah, Mychael Maoeretz Engel\*

Program Studi Informatika, Universitas Ciputra Surabaya, Surabaya 60219, Indonesia

(\*Email Korespondensi: mychael.engel@ciputra.ac.id)

---

**Abstrak:** Dalam menerima pesanan, penerapan jasa pengantaran di perusahaan air galon isi ulang mengharuskan karyawannya untuk mengunjungi titik lokasi konsumen sebanyak dua kali. Proses pemesanan yang dilakukan masih manual yaitu dengan cara telepon atau pesan singkat. Karyawan juga perlu mengetahui lokasi tujuan pengantaran yang telah diberikan oleh konsumen secara akurat. Aplikasi dengan sistem pemesanan saja masih belum cukup karena jasa pengantaran membutuhkan informasi geografis. Dibutuhkan solusi untuk dapat memudahkan proses pemesanan dan pencarian rute pengantaran produk. Penelitian ini menerapkan *Google Maps API* sebagai media pendukung untuk mengelola informasi geografis pengguna. Metode pengembangan aplikasi yang digunakan yaitu dengan *Waterfall* dan menggunakan *Model View Controller (MVC)* sebagai *software design pattern*. Sistem aplikasi dikembangkan dengan *framework* Laravel yang menggunakan *Livewire* agar bisa memberikan tampilan web yang *real-time*. Aplikasi yang akan dihasilkan dalam penelitian ini menggunakan MySQL sebagai tempat penyimpanan data. Pengumpulan data penelitian dilakukan dengan cara wawancara sehingga jenis data penelitian ini berupa kualitatif. Jumlah informan yang berhasil didapatkan yaitu berasal dari tiga *water refill store* dan tiga ibu rumah tangga yang sudah terbiasa dalam memesan jasa antar air galon isi ulang. Sumber informan yang dipakai dalam *water refill store* yaitu *store manager* dan karyawan. Hasil temuan dari penelitian ini yaitu adanya aplikasi berbasis web yang dijadikan sebagai media pemesanan produk di *water refill store* yang juga bisa memudahkan proses pengantaran bagi karyawan di *store*. Sistem yang dihasilkan bisa membantu proses pemesanan dan pengantaran produk di *water refill store*. Saran pengembangan berikutnya yaitu menambahkan fitur pembayaran secara daring dan membuat pelacakan lokasi tetap dapat berfungsi pada perangkat dengan mode *lock screen*.

**Kata Kunci:** Rute Pengantaran, Air Galon Isi Ulang, Google Maps API, Laravel

**Abstract:** In receiving orders, the implementation of delivery services at a refillable gallon water company requires its employees to visit the consumer location point twice. The ordering process is still manual, namely by telephone or short message. Employees also need to know the location of the delivery destination that has been given by consumers accurately. Applications with ordering systems alone are still not enough because delivery services require geographic information. A solution is needed to be able to facilitate the ordering process and find efficient delivery routes. This research applies the Google Maps API as a supporting medium for managing user geographic information. The application development method used is Waterfall and uses Model View Controller (MVC) as a software design pattern. The application system is developed with the Laravel framework that uses Livewire in order to provide a real-time web display. The application that will be produced in this study uses MySQL as a data storage. The research data collection was carried out by means of interviews so that this type of research data is qualitative. The number of informants who were successfully obtained came from three water refill stores and three housewives who were accustomed to ordering refillable gallon water delivery services. The source of informants used in the water refill store is the store manager and employees. The findings of this study are the existence of a web-based application that is used as a product ordering media in a water refill store which can also facilitate the delivery

process for employees in the store. The resulting system makes the product ordering and delivery process at the water refill store easier and more structured. The next development suggestion is to add an online payment feature and make location tracking still work on devices with lock screen mode.

**Keywords:** Delivery Route, Refillable Gallon Water, Google Maps API, Laravel

Naskah diterima 29 Mei 2024; direvisi 23 September 2024; dipublikasi 30 November 2024.  
JUI SI is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



## 1. Pendahuluan

Sistem penjualan di perusahaan air galon isi ulang (*water refill store*) menawarkan layanan pengantaran untuk mencapai lebih banyak konsumen. Namun, dalam penerapannya diperlukan sistem yang terstruktur untuk memfasilitasi proses pemesanan produk. Perencanaan rute pengantaran juga menjadi penting untuk diperhatikan agar meningkatkan efisiensi waktu dan jarak tempuh pengantaran produk yang telah dipesan oleh konsumen.

Metode pemesanan tradisional, seperti melalui telepon dan pesan singkat telah menjadi pilihan umum. Namun, kelemahan dari cara ini adalah sulitnya pelacakan status pesanan dan lokasi konsumen secara akurat. Dalam menerima pesanan air galon isi ulang, karyawan membutuhkan dua kali kunjungan di setiap titik lokasi konsumen. Pada kunjungan pertama, karyawan mengambil air galon yang kosong dan kemudian air galon tersebut diisi dan dikembalikan ke konsumen di lokasi yang sama. Proses ini tidak bisa diselesaikan dengan cara sistem pemesanan yang hanya berlaku untuk satu kali kunjungan di setiap lokasi konsumen. Oleh karena itu, dibutuhkan sistem pendukung untuk dapat memudahkan proses pemesanan dan pengantaran produk.

Berdasarkan permasalahan yang telah dijelaskan, tujuan dari penelitian ini adalah untuk merancang dan membuat sebuah sistem penjualan air galon isi ulang berbasis web dengan penerapan *Google Maps API* pada sistem pengantaran. Dengan adanya sistem penjualan baru berbasis teknologi ini diharapkan dapat memudahkan proses pemesanan produk, pelacakan pengiriman produk dan penentuan rute pengantaran.

## 2. Kajian Pustaka

### 2.1 Studi Terdahulu

Pada penelitian sebelumnya, Zenjaya & Engel (2022) telah berhasil mengembangkan aplikasi berbasis iOS yang dijadikan sebagai sistem pemesanan makanan dalam lingkup kampus. Penelitian kedua, yang dilakukan oleh Anggraini et al. (2020) berhasil merancang aplikasi penjualan sepeda dengan menggunakan *framework* Codeigniter. Ahmadar et al. (2021) juga telah membuat sistem informasi penjualan berbasis *website* khusus untuk toko alat tulis kantor. Sistem yang telah dibuat dari ketiga penelitian ini berhasil meningkatkan penjualan dan mempermudah proses pemesanan produk. Ketiga penelitian ini memiliki kesamaan dengan penelitian yang dilakukan, yaitu membangun sebuah sistem untuk penjualan produk. Meskipun demikian, penelitian ini lebih fokus pada sistem penjualan di perusahaan air galon isi ulang.

Babako & Sitokdana (2022) telah merancang aplikasi pemesanan di perusahaan air galon isi ulang dengan perangkat *mobile*. Aplikasi yang dihasilkan terbukti dapat mempermudah proses penjualan dan pemasaran produk di *store*. Pada penelitian ini aplikasi yang dikembangkan hanya terbatas pada aplikasi *mobile*. Masih ada keterbatasan terhadap fitur yang disediakan pada aplikasi yang dihasilkan dalam penelitian ini. Bagi konsumen atau pembeli, belum tersedia fitur penentuan lokasi pengantaran produk berdasarkan titik koordinat *latitude* dan *longitude* dan pemantauan lokasi karyawan yang sedang mengantarkan pesanan. Selain itu, bagi karyawan belum terdapat fitur untuk pengurutan pengantaran pesanan dan penentuan rute pengantaran pesanan.

Beberapa peneliti terdahulu pun telah memanfaatkan *Google Maps API* sebagai teknologi sistem informasi geografis. Salah satunya yaitu penelitian yang dilakukan oleh Budiawan & Afrianto (2020) yang berhasil merancang aplikasi berbasis android untuk memudahkan pembimbing jemaah mengetahui lokasi jemaah haji dan umrah yang

dibimbing. Aplikasi yang dikembangkan memanfaatkan *Google Maps API* dan aplikasi berhasil berjalan dengan baik. Namun, pemanfaatan *Google Maps API* sebagai alat pemantauan pada aplikasi ini hanya diperuntukkan jemaah haji dan umrah.

## 2.2 Teknologi

### 2.2.1 Website

*Website* adalah tempat penyimpanan data yang masif dan dinamis untuk mengirimkan sejumlah besar informasi. Halaman *website* bersifat semi-terstruktur yang menyajikan berbagai informasi kepada pengguna dengan cara yang mudah dibaca. *Website* disatukan oleh halaman web yang dapat diakses dengan memasukkan alamat URL (*Uniform Resource Locator*) dan dapat berisi teks, grafik, dan *hyperlink* ke halaman *website* lain (Umadevi et al., 2020).

*Website* dinamis termasuk salah satu jenis *website* yang dapat menyajikan kolaborasi antar pengguna baik dalam mengembangkan, berbagi, dan beroperasi secara bersamaan di *website* sehingga menciptakan pengalaman *online* bagi pengguna dan memanfaatkan teknologi baru untuk memperluas merek dan produk (Mbunge et al., 2022). Penelitian ini menggunakan media *website* sebagai media aplikasi penelitian.

### 2.2.2 HTML (Hyper Text Markup Language)

HTML (*Hypertext Markup Language*) telah menjadi standar *hypertext* resmi sejak awal sebagai bahasa pemrograman *website* untuk membangun aplikasi berbasis *website* dan menghubungkan halaman *website* dengan halaman lainnya (Tabarés, 2021). Pada penelitian ini HTML digunakan sebagai bahasa markah untuk membangun struktur dasar dari sebuah halaman *website* dengan menentukan elemen-elemen seperti teks, gambar, tautan, tabel, formulir, dan lainnya. Pada penelitian ini HTML diterapkan dalam implementasi di setiap tampilan pengguna.

### 2.2.3 PHP (Hypertext Preprocessor)

PHP atau *hypertext preprocessor* menjadi salah satu bahasa terkemuka untuk pengembangan *script* pada *server website* (Li et al., 2019). PHP digunakan dalam pengembangan backend pada *website* dengan tujuan untuk menentukan aturan dan alur pada pengoperasian sisi *server* yaitu membangun database, menyiapkan *server website*, dan membuat logika *server* (Szujo et al., 2023).

### 2.2.4 CSS (Cascading Style Sheet)

CSS atau *cascading style sheet* merupakan bahasa pemrograman yang digunakan untuk melakukan pengolahan visual dan animasi pada *website* (Smith et al., 2022). CSS mendeskripsikan representasi dari halaman *website*, termasuk warna, tata letak, dan ukuran (Kotsyuba et al., 2022).

### 2.2.5 JavaScript

Sebagian besar konten dinamis pada situs *website* saat ini dibuat menggunakan JavaScript karena penggunaannya yang sederhana dan aksesibilitasnya yang baik (Phung & Mimura, 2021). JavaScript digunakan untuk membuat konten yang diperbarui secara dinamis, menganimasikan gambar, dan mengontrol multimedia pada halaman *website* (Sherin et al., 2023).

### 2.2.6 MySQL (My Structured Query Language)

Bahasa pemrograman MySQL digunakan sebagai sistem manajemen basis data dalam mengatur pengelolaan *relational database*. MySQL mengatur data ke dalam satu atau lebih tabel data dan dapat berkaitan satu sama lain untuk keperluan *website* (Karydas et al., 2023).

### 2.2.7 Laravel

Laravel merupakan kerangka kerja dari *website* PHP. Laravel memiliki kecepatan dan kestabilan dalam pengembangan aplikasi *website* (Medici et al., 2021). Kerangka kerja ini berbasis bahasa pemrograman PHP yang bisa digunakan untuk membantu proses pengembangan sebuah *website* agar lebih maksimal dan dinamis. Laravel

dijadikan sebagai *framework* PHP dalam pengembangan aplikasi penelitian.

### 2.2.8 Google Maps API

*Google Maps API* telah menyediakan fitur *Maps Javascript API*, *Geocoding API*, *Places API*, *Geolocation*, *Direction*, dan *Distance Matrix* (Google Maps Platform Documentation, n.d.).

**Maps JavaScript API** memungkinkan pengembang aplikasi untuk menampilkan peta interaktif di situs atau aplikasi *website*. Kegunaan dari API (*application programming interface*) ini yaitu sebagai berikut.

- Menampilkan peta interaktif dengan berbagai kontrol, seperti *zoom*, *panning*, dan tampilan peta yang berbeda.
- Menambahkan *marker*, *info window*, *polygon*, *polyline*, dan elemen-elemen lain ke peta.
- Mendeteksi interaksi pengguna, seperti klik pada peta atau *marker*.
- Terintegrasi dengan berbagai layanan *google maps* lainnya, seperti *Directions API* dan *Places API*.

**Geocoding API** memungkinkan pengembang aplikasi untuk mengkonversi alamat fisik menjadi koordinat geografis (*latitude* dan *longitude*) dan sebaliknya. Kegunaan dari API (*application programming interface*) ini yaitu sebagai berikut.

- Menyediakan layanan pencarian alamat dan *geocode* untuk mendapatkan informasi lokasi berdasarkan alamat tertentu.
- Mendukung *geocode* untuk mendapatkan alamat dari koordinat geografis.

**Places API** memungkinkan pengembang untuk mengakses dan menampilkan informasi lokasi bisnis, tempat wisata, dan *landmark* lainnya di peta. Kegunaan dari API (*application programming interface*) ini yaitu sebagai berikut.

- Menyediakan layanan pencarian tempat berdasarkan kategori, kata kunci, atau lokasi tertentu.
- Menampilkan detail tempat, termasuk nama, alamat, *rating*, ulasan, dan informasi lainnya.
- Terintegrasi dengan fitur pencarian otomatis dan rekomendasi tempat terdekat.

**Geolocation** adalah fitur yang memungkinkan aplikasi *website* untuk mendapatkan informasi lokasi pengguna berdasarkan perangkat pengguna. Kegunaan dari *geolocation* yaitu sebagai berikut.

- Mengizinkan aplikasi *website* untuk mengakses koordinat geografis pengguna, baik menggunakan GPS (*Global Positioning System*), Wi-Fi, atau sinyal seluler.
- Memungkinkan aplikasi untuk menyesuaikan konten atau layanan berdasarkan lokasi pengguna.
- Berguna untuk aplikasi seperti peta interaktif, layanan penyediaan lokasi, atau rekomendasi tempat terdekat.

**Directions API** memungkinkan pengembang aplikasi untuk mendapatkan rute perjalanan antar dua atau lebih lokasi berdasarkan moda transportasi yang dipilih. Kegunaan dari API (*application programming interface*) ini yaitu sebagai berikut.

- Menyediakan instruksi langkah demi langkah untuk mencapai tujuan, termasuk jarak, waktu tempuh, dan arah.
- Mendukung berbagai moda transportasi, seperti berkendara, berjalan kaki, bersepeda, atau menggunakan transportasi umum.
- Memperhitungkan kondisi lalu lintas saat ini untuk memberikan estimasi waktu yang akurat.

**Distance Matrix API** memungkinkan pengembang aplikasi untuk menghitung jarak dan waktu tempuh antar beberapa lokasi. Kegunaan dari API (*application programming interface*) ini yaitu sebagai berikut.

- Memberikan matriks jarak dan waktu tempuh antara beberapa titik, baik berdasarkan moda transportasi tertentu maupun mode pengguna yang berbeda.
- Mendukung perhitungan jarak dan waktu tempuh untuk perjalanan berkendara, berjalan kaki, atau bersepeda.
- Berguna untuk menentukan rute optimal, menghitung biaya perjalanan, atau menyesuaikan estimasi waktu pengiriman.

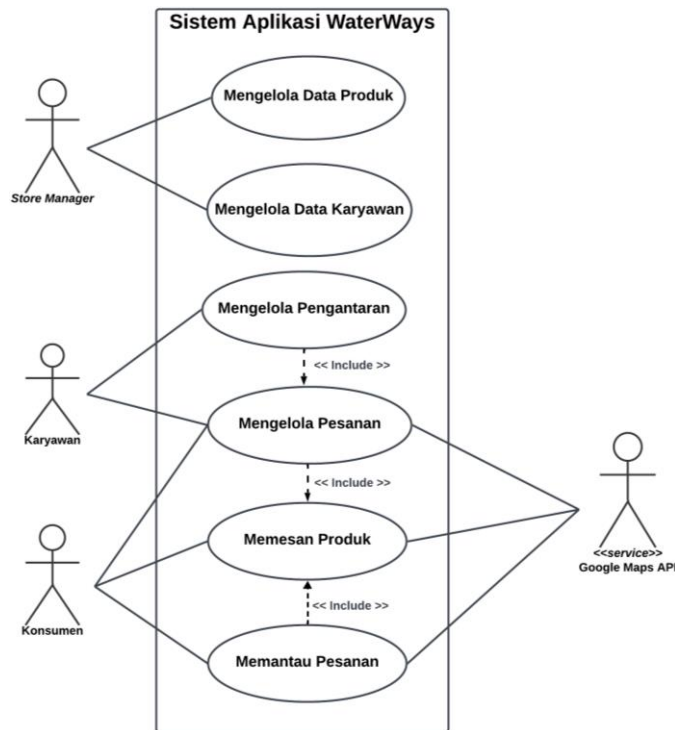
### 3. Metode Penelitian

*Software Development Life Cycle* (SDLC) merupakan proses pengembangan perangkat lunak untuk menghasilkan solusi dari tugas atau masalah yang dihadapi dengan tujuan untuk memaksimalkan kerja tim dalam pengembangan perangkat lunak secara efektif (Najihi et al., 2022). Pemilihan SDLC sangat penting dalam manajemen tim dan alur pengembangan perangkat lunak. Ketepatan penerapan SDLC dapat membantu untuk merancang, mengembangkan, dan menguji perangkat lunak agar dapat menghasilkan produk yang berkualitas tinggi dan memenuhi harapan pengguna (Nath et al., 2023).

Pada penelitian ini metode yang digunakan yaitu *waterfall*. Metode ini termasuk SDLC tradisional yang menggunakan fase pengembangan bertahap dan konsisten (Najihi et al., 2022). Tahapan-tahapan yang dilakukan dalam *waterfall* ada enam, diantaranya yaitu analisa kebutuhan, desain sistem, implementasi, pengujian serta analisa sistem, distribusi sistem ke seluruh pengguna, dan pemeliharaan sistem (Wang & Wang in Engel & Santoso, 2020). Pada tahap analisa kebutuhan dilakukan identifikasi kebutuhan sistem bagi pengguna. Selanjutnya, dilakukan desain sistem. Desain ini mencakup struktur data, antar muka pengguna, dan modul-modul sistem yang diperlukan. Setelah itu, masuk ke tahap implementasi yaitu proses pembuatan kode program berdasarkan hasil desain sistem. Setelah sistem diimplementasikan, dilakukan pengujian sistem untuk mengetahui kinerja sistem terhadap spesifikasi pengguna. Dalam pengujiannya sistem WaterWays menggunakan *black box testing*. Tujuan utama dari *black box testing* adalah untuk memastikan bahwa sistem berperilaku sesuai dengan spesifikasi fungsional yang telah ditentukan, tanpa harus memperhatikan bagaimana sistem tersebut diimplementasikan (Desikan et. al, 2006). Dalam *black box testing*, pengujian dilakukan dengan memasukkan *input* ke dalam sistem dan mengamati *output* yang dihasilkan, serta memeriksa apakah responnya sesuai dengan yang diharapkan. Tahapan pertama yang dilakukan dengan *black box testing* yaitu menentukan *input* yang akan diuji berdasarkan spesifikasi fungsional sistem. Kemudian, pengguna memasukkan *input* ke dalam sistem dan mengamati *output* yang dihasilkan. Setelah itu, *output* tersebut dibandingkan dengan *output* yang diharapkan berdasarkan spesifikasi fungsional sistem. Pada tahapan pengujian yang terakhir, hasil pengujian dicatat termasuk setiap penyimpangan dari harapan dan *bug* yang ditemukan. Sistem yang telah diuji kemudian didistribusikan kepada pengguna akhir. Pengguna diberikan pelatihan dan dokumentasi yang diperlukan untuk menggunakan sistem. Pada tahap ini dilakukan wawancara dengan beberapa pengguna untuk mendapatkan umpan balik. Wawancara dilakukan dengan teknik semi-terstruktur. Pada tahap terakhir dalam proses *waterfall*, dilakukan perbaikan *bug* dan penyesuaian sistem berdasarkan umpan balik dari pengguna.

#### 3.1 Use Case Diagram

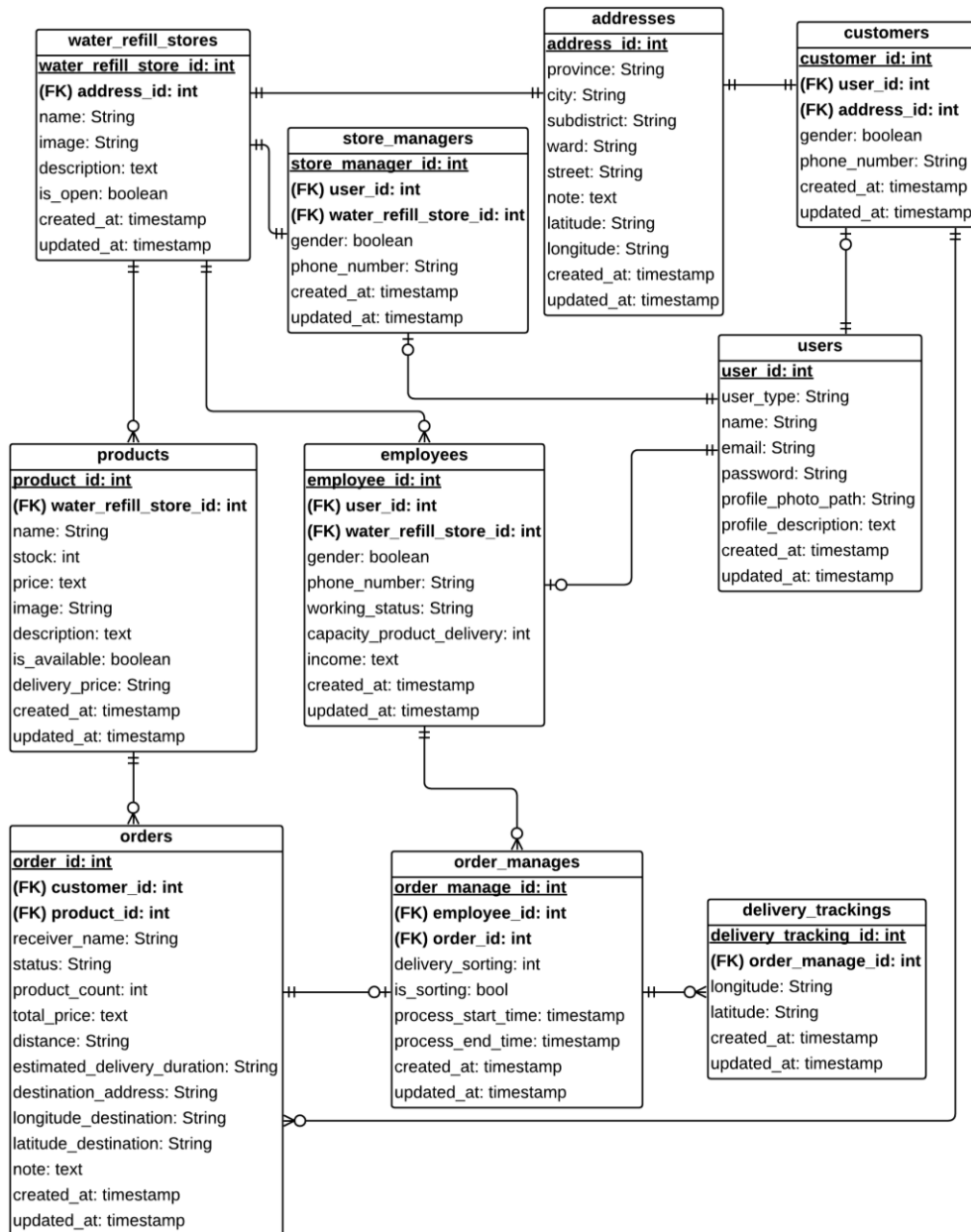
Desain *use case* sistem dapat dilihat pada Gambar 1. Terdapat tiga aktor primer yaitu *store manager*, karyawan, dan konsumen. *Store manager* bisa mengelola data produk dan karyawan. Karyawan bisa mengelola pesanan dan pengantaran. Sedangkan konsumen bisa mengelola pesanan, memesan produk, dan memantau pesanan. Ada satu aktor sekunder, yaitu *Google Maps API* yang digunakan dalam *use case* mengelola pengantaran, memesan produk, dan memantau pesanan.



Gambar 1. Use case diagram sistem

### 3.2 Desain Database

Pada Gambar 2 merupakan desain ERD (*Entity Relationship Diagram*) sebagai pemodelan relasi antar data yang digunakan di dalam *database*. Data pemantauan lokasi pengiriman produk terhubung dengan pesanan, sehingga memungkinkan bagi konsumen untuk bisa melakukan proses pemantauan lokasi pengantar secara langsung. ERD ini memodelkan hubungan antar entitas dalam sistem pengisian ulang air. Entitas *water\_refill\_stores* menyimpan informasi tentang toko air isi ulang termasuk alamat yang terhubung melalui relasi *one to one* dengan entitas *addresses*. Setiap toko dikelola oleh *store\_managers* yang merepresentasikan pengelola toko dan berhubungan *one to one* dengan entitas *users* untuk informasi pengguna. Entitas *employees* sebagai penyimpanan data karyawan dan berhubungan *many to one* dengan *water\_refill\_stores*. Entitas *employees* dan *customers* juga berhubungan *one to one* dengan *users* sebagai penyimpanan data pengguna. Entitas *products* merepresentasikan produk yang tersedia di setiap toko, seperti stok air yang dapat dibeli dengan relasi *one to many* ke *orders* yang berisi pesanan pelanggan. Entitas *customers* terhubung dengan *addresses* dan memiliki relasi *one to many* ke *orders* untuk melakukan pemesanan produk. Setiap pesanan diatur oleh entitas *order\_manages*, yang berisi informasi tentang pemrosesan dan pengiriman pesanan, serta terhubung *many to one* dengan *employees* yang melakukan pengantaran. Entitas *delivery\_trackings* mencatat lokasi real-time pengiriman, yang terhubung *many to one* dengan *order\_manages*, sehingga pelanggan dapat memantau proses pengiriman pesanan.

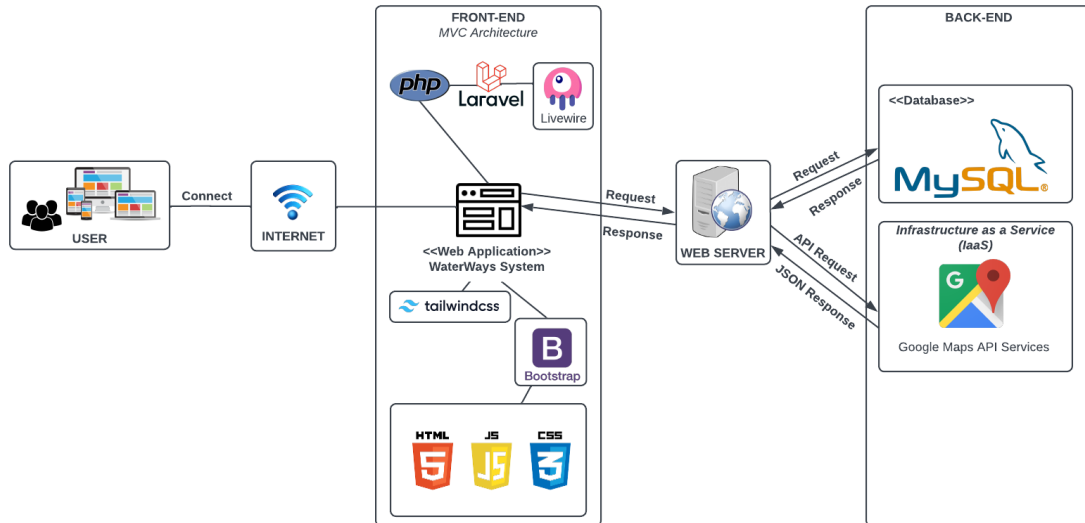


Gambar 2. Desain database sistem

### 3.3 Arsitektur Sistem

Sistem WaterWays merupakan sistem *website* yang *responsive*. Sistem ini diakses pengguna melalui jaringan internet. Dalam arsitektur sistemnya, dibagi menjadi *front-end* dan *back-end*. Dimana pada *front-end* menggunakan Laravel, Livewire, Tailwind CSS, dan Bootstrap untuk membangun tampilan aplikasi. Pada pengembangan tampilan sistem menggunakan *Model View Controller* (MVC) sebagai *software design pattern*. Sedangkan pada *back-end*

terdapat MySQL sebagai tempat penyimpanan *database* dan Google Maps API sebagai teknologi pendukung untuk mengelola informasi geografis pengguna. Desain arsitektur sistem penelitian dapat dilihat pada Gambar 3.



**Gambar 3.** Arsitektur sistem

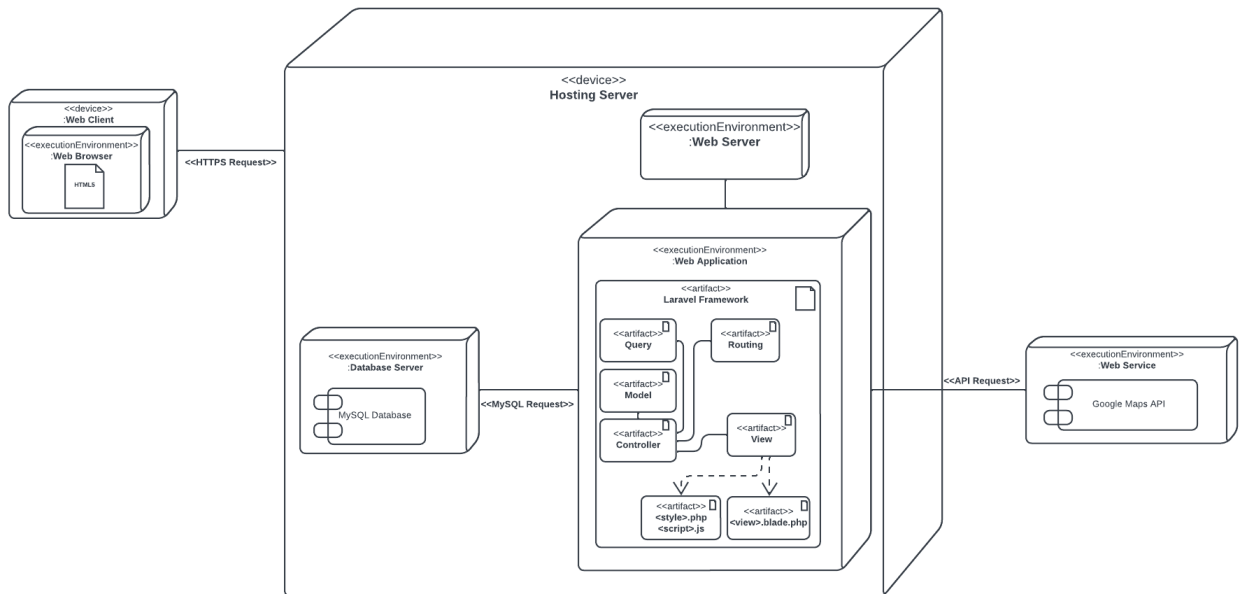
### 3.4 Deployment Diagram

*Deployment diagram* sistem yang dikembangkan terdiri dari tiga node utama yaitu *web client*, *hosting server*, dan *web service*. *Web client* merupakan node perangkat yang digunakan pengguna untuk mengakses aplikasi, *hosting server node* merupakan perangkat yang dijadikan sebagai media penyimpanan dan pengaksesan aplikasi web, dan *web service node* sebagai aplikasi pihak ketiga yaitu dengan *Google Maps API*. Pada *web client node* terdapat *web browser node* dimana di dalamnya terdapat file HTML5. *Web client node* berhubungan dengan *hosting server node* dalam koneksi internet.

Pada *hosting server node* terdapat tiga *node* yaitu *web server node*, *web application node*, dan *database server node*. *Web server node* berhubungan dengan *web application node*, dan *web application node* berhubungan dengan *database server node*. Disisi lain, *web application node* juga berhubungan dengan *web service node*.

*Web application node* memiliki artefak berupa *file* kode program yang diimplementasikan dengan menggunakan *Laravel framework*. Sedangkan *database server node* merupakan tempat penyimpanan data dengan *MySQL*. *Deployment diagram* sistem dapat dilihat pada Gambar 4.



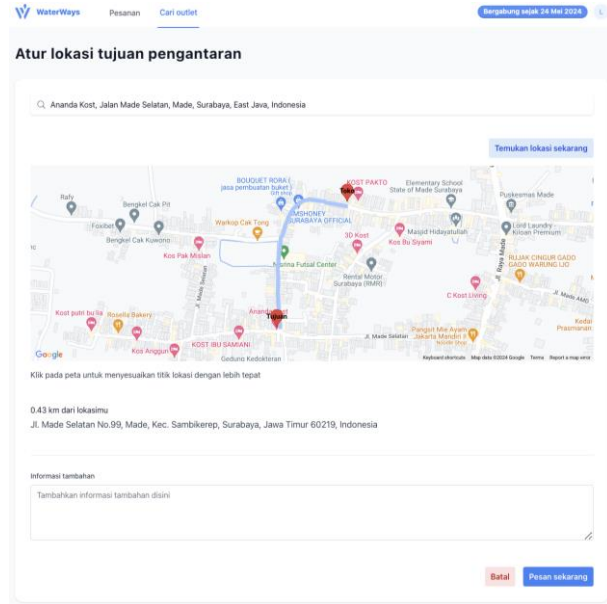


Gambar 4. Deployment diagram sistem

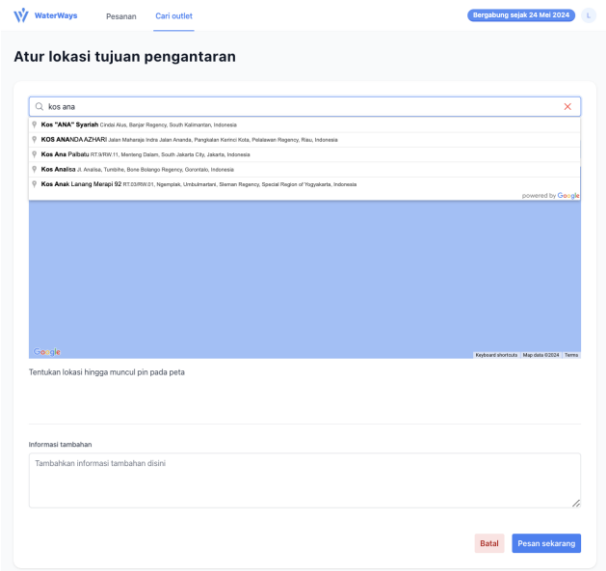
## 4. Hasil dan Pembahasan

### 4.1 Hasil Implementasi

Gambar 5(a) merupakan tampilan halaman pengaturan lokasi tujuan pengantaran pesanan dengan menggunakan bantuan *map*. Pengguna dapat mengklik tombol “Temukan lokasi sekarang” untuk mendapatkan lokasi saat ini dengan menggunakan fitur *geolocation*. Pengguna juga bisa berinteraksi secara langsung pada *map* untuk menentukan lokasi dengan cara mengklik area pada *map*. Pengguna bisa memasukkan alamat pada kolom pencarian kemudian sistem akan menampilkan rekomendasi alamat yang diperoleh dari fitur *Google Maps API*. Hasil rekomendasi alamat dapat dilihat pada Gambar 5(b).



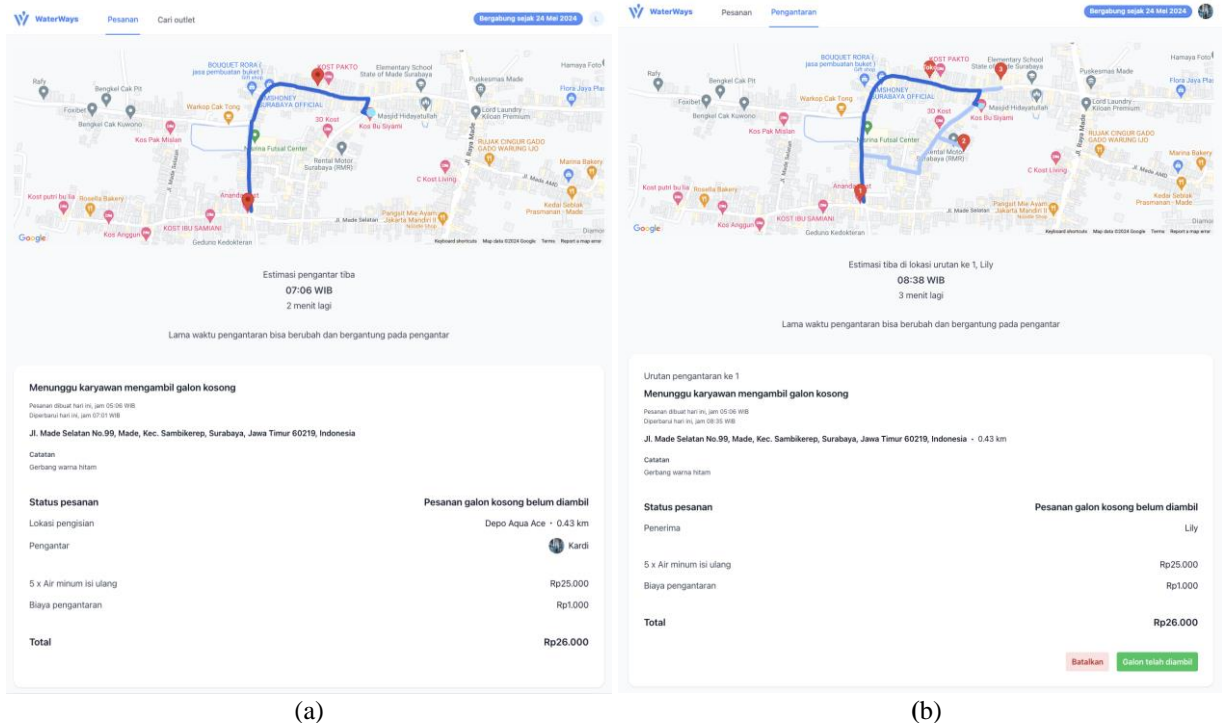
(a)



(b)

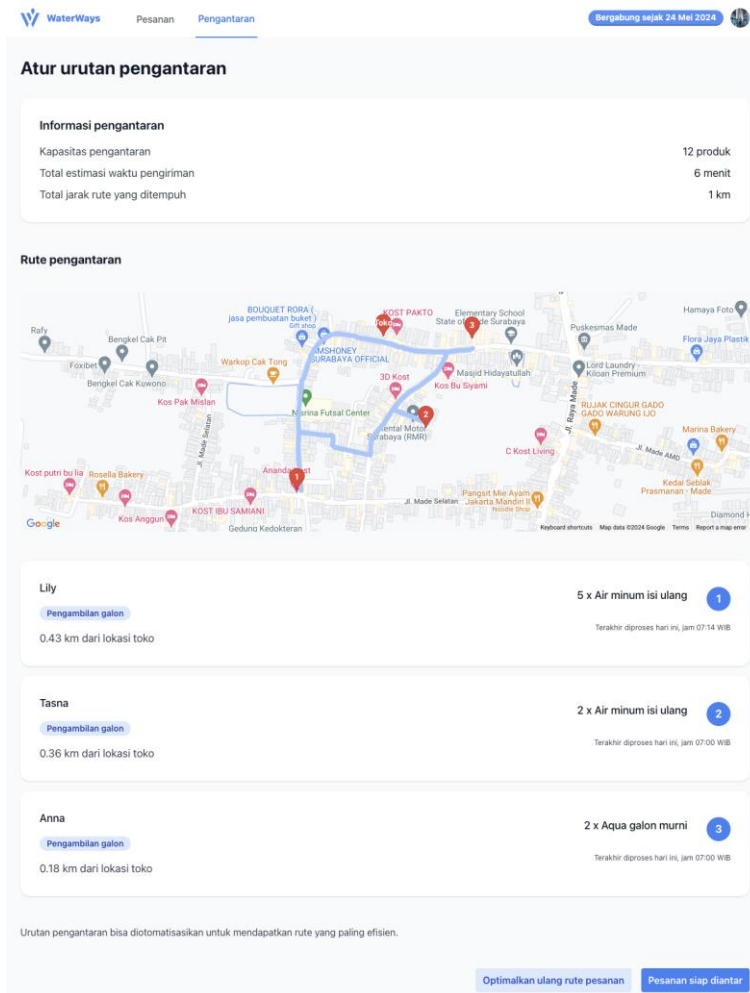
**Gambar 5.** (a) Hasil implementasi menentukan lokasi tujuan pengantaran dan (b) Hasil implementasi menampilkan rekomendasi alamat

Tampilan halaman pemantauan lokasi pengantar ditunjukkan pada Gambar 6(a). Pada halaman ini, konsumen dapat memantau lokasi pengantar pada *map* secara langsung. Sedangkan pada Gambar 6(b) merupakan tampilan pada sisi karyawan untuk memantau proses pengantaran. Pada halaman ini, karyawan dapat mengelola pesanan atau memberikan respon terhadap pesanan yang diterima.



**Gambar 6.** (a) Hasil implementasi tampilan pantau posisi pengantar dan (b) Hasil implementasi halaman pemantauan proses pengantaran

Gambar 7 merupakan tampilan halaman pengaturan urutan pengantaran produk. Pada bagian ini karyawan mendapatkan informasi rute perjalanan berdasarkan urutan pengantaran pesanan yang telah ditentukan. Pada halaman ini, pengguna bisa mengurutkan pengantaran secara manual dan optimalisasi urutan rute dengan memanfaatkan *WayPoints* pada fitur *Google Maps API*. Ketika pengurutan pesanan telah dilakukan, karyawan dapat mengklik tombol “Pesanan siap diantar” untuk menuju ke halaman Pengantaran.



**Gambar 7.** Hasil implementasi pengurutan pengantaran

## 4.2 Hasil Pengujian

Dalam pelaksanaan pengujiannya, sistem telah berhasil diuji oleh 5 orang calon pengguna diantaranya 3 orang konsumen, 1 orang *store manager*, dan 1 orang karyawan. Pelaksanaan uji coba dilakukan secara langsung di lokasi pengujian yang berada di kota Tarakan. Hasil pengujian sistem dengan *black box testing* dapat dilihat pada Tabel 1.

**Table 1.** Hasil pengujian *black box testing*

Skenario Pengujian	Hasil yang Diharapkan	Hasil yang Sebenarnya	Status
Masukkan alamat pada kolom pencarian	Sistem dapat menampilkan rekomendasi alamat	Sistem dapat menampilkan rekomendasi alamat	Berhasil
Mengklik tombol “Temukan lokasi sekarang”	Sistem dapat menampilkan <i>marker</i> sesuai dengan titik lokasi pengguna saat ini	Sistem dapat menampilkan <i>marker</i> sesuai dengan titik lokasi pengguna saat ini	Berhasil
Mengklik tampilan <i>map</i>	Sistem dapat menampilkan <i>marker</i> pada <i>map</i> sesuai dengan titik lokasi <i>map</i> yang diklik	Sistem dapat menampilkan <i>marker</i> pada <i>map</i> sesuai dengan titik lokasi <i>map</i> yang diklik	Berhasil
Mengakses halaman pemantauan lokasi pengantar	Sistem dapat menampilkan informasi pesanan dan menampilkan <i>map</i> serta <i>marker</i> posisi pengantar yang selalu diubah setiap 10 detik	Sistem dapat menampilkan informasi pesanan dan menampilkan <i>map</i> serta <i>marker</i> posisi pengantar yang selalu diubah setiap 10 detik	Berhasil
Mengakses halaman pengaturan urutan pengantaran pesanan	Sistem dapat menampilkan halaman pengaturan urutan pengantaran pesanan	Sistem dapat menampilkan halaman pengaturan urutan pengantaran pesanan	Berhasil
Mengatur urutan pengantaran pesanan	Sistem dapat menampilkan informasi pengantaran dan rute berdasarkan urutan yang ditentukan	Sistem dapat menampilkan informasi pengantaran dan rute berdasarkan urutan yang ditentukan	Berhasil
Mengakses halaman pemantauan proses pengantaran	Sistem dapat menampilkan halaman pemantauan proses pengantaran	Sistem dapat menampilkan halaman pemantauan proses pengantaran	Berhasil
Mengklik tombol untuk mengubah status pesanan	Sistem dapat mengubah status pesanan	Sistem dapat mengubah status pesanan	Berhasil

Berdasarkan hasil pengujian *black box testing* yang telah dilakukan tidak ditemukan adanya kesalahan pada fungsionalitas sistem. Sistem dapat memberikan *output* sesuai dengan hasil yang diharapkan pada skenario pengujian. Setelah sistem diuji fungsionalitasnya, dilakukan pengujian beta untuk mendapatkan umpan balik dari calon pengguna. Pengujian beta dilakukan dengan menggunakan wawancara. Berdasarkan hasil wawancara yang dilakukan diperoleh respon yang sangat baik. Informan yang merupakan *store manager* menganggap aplikasi dapat memperluas pemasaran produk. Selain itu, lokasi *store* bisa diketahui oleh konsumen dengan mudah.

Bagi karyawan *store*, fitur-fitur yang ada juga sudah sangat jelas dan bisa membantu dalam mengelola pesanan. Proses pengecekan pesanan juga menjadi lebih mudah karena tidak perlu lagi mengecek pesanan satu persatu melalui kontak pribadi seperti melalui Whatsapp. Distraksi terhadap konsumen yang menanyakan status pesanan ketika karyawan bekerja juga menjadi lebih berkurang. Perencanaan rute pengantaran menjadi lebih jelas dan lebih mudah

yang bisa mempercepat dalam mengantarkan pesanan.

Bagi konsumen, fitur-fitur aplikasi sudah lengkap dan sangat mudah dipahami. Dengan adanya aplikasi, pemesanan produk dan pelacakan pesanan menjadi lebih praktis dan mudah. Selama mencoba aplikasi, konsumen tidak merasa kesulitan dalam menggunakan aplikasi. Fitur pelacakan pesanan sudah sangat jelas dan lengkap serta mampu membantu dalam memantau proses pesanan. Penentuan lokasi tujuan pemesanan produk juga bisa menjadi lebih fleksibel dikarenakan bisa menentukan lokasi tujuan dengan tidak menggunakan lokasi saat ini. Konsumen memberikan umpan balik yang positif terhadap keseluruhan fitur aplikasi. Aplikasi ini juga bisa dijadikan sebagai aplikasi pendukung dalam melakukan proses pemesanan produk. Dengan adanya pelacakan pesanan, membuat konsumen menjadi lebih aman dalam memesan produk karena bisa melihat lokasi pengantar. Informasi pesanan menjadi lebih terbuka dan fleksibel.

## 5. Kesimpulan dan Saran

Penelitian ini memberikan kontribusi signifikan terhadap bidang pengembangan sistem informasi geografis (GIS) dan *e-commerce*, khususnya dalam integrasi layanan lokasi dengan aplikasi penjualan untuk praktik bisnis di industri air minum. Berdasarkan analisis permasalahan hingga pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa penelitian ini berhasil merancang dan membangun sistem penjualan air galon isi ulang menggunakan kerangka kerja Laravel dengan pola Model View Controller (MVC) serta menggunakan bahasa pemrograman PHP dan HTML. Aplikasi ini memanfaatkan Livewire untuk pembaruan data otomatis dan MySQL sebagai tempat penyimpanan data. Integrasi dengan berbagai fitur *Google Maps API* seperti *Maps JavaScript API*, *Geocoding API*, *Places API*, *Geolocation*, *Directions API*, dan *Distance Matrix API* juga berhasil dilakukan, memungkinkan berbagai interaksi dan layanan terkait peta dan lokasi. Dengan sistem *WaterWays*, penentuan lokasi tujuan pengantaran menjadi lebih akurat dan mudah dengan fitur *geolocation*, rekomendasi alamat, dan *map* yang terdapat dalam *Google Maps API*. Fitur pelacakan pesanan memungkinkan konsumen mengetahui estimasi waktu tiba dan lokasi pengantar, yang tidak mungkin diperoleh dengan cara pemesanan manual. Status pesanan menjadi lebih mudah dilacak. Perencanaan rute pengantaran menjadi lebih jelas dan mempercepat proses pengantaran, memungkinkan karyawan menentukan urutan pengantaran secara manual atau otomatis untuk menemukan rute pengantaran produk dengan memanfaatkan *WayPoints* dalam fitur *Direction* pada *Google Maps API*. Saran untuk perbaikan mencakup peningkatan antarmuka pengguna, penambahan ikon dan fitur pengurutan jarak lokasi *store* mulai dari yang terdekat hingga yang paling jauh sehingga lebih mempercepat proses pemesanan produk, variasi desain *marker*, perbaikan desain rute, pelacakan yang tetap berjalan pada mode *lock screen* sehingga perangkat yang dimiliki pengantar tidak harus menyala pada saat mengantarkan produk, dan penambahan fitur *payment gateway* untuk dijadikan sebagai media pembayaran secara daring.

## Daftar Pustaka

- Ahmad, M., Perwito, P., & Taufik, C. (2021). Perancangan Sistem Informasi Penjualan Berbasis Web pada Rahayu Photo Copy dengan Database Mysql. *Dharmakarya: Jurnal Aplikasi Ipteks untuk Masyarakat*, 10(4), 284-289.
- Anggraini, Y., Pasha, D., Damayanti, D., & Setiawan, A. (2020). Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter. *Jurnal Teknologi dan Sistem Informasi (JTSI)*, 1(2), 64-70.
- Babako, W. S., & Sitokdana, M. N. (2022). Perancangan Aplikasi Pemesanan Air Galon Isi Ulang Berbasis Mobile. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 6(2), 974-987.
- Budiawan, M. I., & Afrianto, I. (2020). Development of Android Based Hajj and Umrah Pilgrims Monitoring Application in Dago Wisata International. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 14(3), 253-264.
- Desikan, S., Ramesh, G., & Kekre, H. B. (2006). *Software Testing: Principles and Practices*. Pearson Education India.
- Engel, M. M., & Santoso, N. G. (2020, August). Fuzzy Multi Attribute Decision Making – Simple Additive Weighting pada Rekomendasi Lowongan Pekerjaan Sampingan Barista. *JUI SI*, 06(02), 11-20.

- Google Maps Platform Documentation. (n.d.). Google for Developers. Retrieved May 18, 2024, from <https://developers.google.com/maps/documentation>.
- Karydas, C., Chatziantoniou, M., Stamkopoulos, K., Iatrou, M., Vassiliadis, V., & Mourelatos, S. (2023). Embedding a Precision Agriculture Service into a Farm Management Information System - ifarma/PreFer. *Smart Agricultural Technology*, 4, 100175.
- Kotsyuba, I., Shikov, A., Karpik, K., Galperin, M., Kudriashov, A., & Silko, J. (2022). Recommendation Web Service for Choosing an Individual Educational Path in the Field of Transportation Systems' Programming. *Transportation Research Procedia*, 63, 591–599.
- Li, Y., Huang, J., Ikusan, A., Mitchell, M., Zhang, J., & Dai, R. (2019). ShellBreaker : Automatically detecting PHP-based malicious web shells. *Computers & Security*, 87.
- Mbunge, E., Jiyane, S., & Muchemwa, B. (2022). Towards emotive sensory Web in virtual health care: Trends, technologies, challenges and ethical issues. *Sensors International*, 3.
- Medici, M., Pedersen, S. M., Canavari, M., Anken, T., Stamatiopoulos, P., Tsiropoulos, Z., Zotos, A., & Tohidloo, G. (2021). A Web-Tool for Calculating The Economic Performance of Precision Agriculture Technology. *Computers and Electronics in Agriculture*, 181.
- Najihi, S., Elhadi, S., Abdelouahid, R. A., & Marzak, A. (2022). Software Testing from an Agile and Traditional view. *Procedia Computer Science*, 203, 775–782. <https://doi.org/10.1016/j.procs.2022.07.116>
- Nath, P., Mushahary, J. R., Roy, U., Brahma, M., & Singh, P. K. (2023). AI and Blockchain-based source code vulnerability detection and prevention system for multiparty software development. *Computers and Electrical Engineering*, 106.
- Phung, N. M., & Mimura, M. (2021). Detection of Malicious Javascript on an Imbalanced Dataset. *Internet of Things (Netherlands)*, 13.
- Sherin, S., Muqet, A., Khan, M. U., & Iqbal, M. Z. (2023). QExplore: An exploration strategy for dynamic web applications using guided search. *Journal of Systems and Software*, 195.
- Smith, C. D., Atawala, N., Klatt, C. A., & Klatt, E. C. (2022). A Review of Web-Based Application of Online Learning In Pathology And Laboratory Medicine. *Journal of Pathology Informatics* (Vol. 13). Elsevier B.V.
- Szujó, G., Biber, Z., Gál, V., & Szabó, B. (2023). MaGISter-mine: A 2D and 3D web application in the service of mining industry. *International Journal of Applied Earth Observation and Geoinformation*, 116.
- Tabarés, R. (2021). HTML5 and the evolution of HTML; tracing the origins of digital platforms. *Technology in Society*, 65.
- Umadevi, K. S., Balakrishnan, P., & Amali, G. B. (2020). Influential Analysis of Web Structure Using Graph Based Approach. *Procedia Computer Science*, 172, 165–171.
- Zenjaya, K. T., & Engel, M. M. (2022). Rancang Bangun Aplikasi Pemesanan Makanan dalam Kampus Secara Online Berbasis Sistem Operasi iOS. *JUISI*, 8(2), 8-17.