

# Sistem *E-Commerce* Perlengkapan *Aquatic* Berbasis Web dengan *Payment Gateway* dan *Firebase Cloud Messaging*

DOI: <http://dx.doi.org/10.35889/jutisi.v14i3.3371>

Creative Commons License 4.0 (CC BY – NC)

Ade Prima Krifitsky<sup>1\*</sup>, Tri Widodo<sup>2</sup>

Informatika, Universitas Teknologi Yogyakarta, Yogyakarta, Indonesia

\*e-mail Corresponding Author: ade.5220411413@student.uty.ac.id

## Abstract

*Sky Aquatic Store in Dukun, Magelang faces several operational challenges, including limited customer reach, error-prone manual stock management, and inefficient transaction communication. This study develops a web-based e-commerce system integrated with the Midtrans payment gateway and Firebase Cloud Messaging (FCM) to support the digitalization of micro, small, and medium enterprises (MSMEs). The system was built using React.js for the frontend, Node.js with Express.js for the backend, and MySQL as the database, employing the Agile Scrum methodology over an eight-week development cycle. Black-box testing conducted on ten test cases demonstrated a 100% success rate under both normal and abnormal conditions. The system successfully integrates multi-method payments (Virtual Account, E-wallet, and Credit Card) via Midtrans, along with FCM infrastructure for real-time notifications. The results confirm that all core features operate according to specifications with robust error handling, providing a comprehensive e-commerce solution that supports the digital transformation of aquatic equipment MSMEs.*

**Keywords:** *E-commerce; Payment gateway; Firebase Cloud Messaging; REST API; Aquatic equipment*

## Abstrak

Toko Sky Aquatic di Dukun, Magelang menghadapi kendala jangkauan pelanggan terbatas, pengelolaan stok manual yang rawan kesalahan, dan komunikasi transaksi tidak efisien. Penelitian ini mengembangkan sistem *e-commerce* berbasis web terintegrasi dengan *payment gateway* Midtrans dan *Firebase Cloud Messaging* (FCM) untuk mendukung digitalisasi UMKM. Sistem dibangun menggunakan React.js (*frontend*), Node.js dengan Express.js (*backend*), dan MySQL (*database*), dengan metodologi *Agile Scrum* selama 8 minggu. Pengujian *black-box testing* terhadap 10 *test case* menunjukkan tingkat keberhasilan 100% pada kondisi normal dan tidak normal. Sistem berhasil mengintegrasikan *payment gateway* multi-metode (*Virtual Account*, *E-wallet*, Kartu Kredit) dan infrastruktur FCM untuk notifikasi *real-time*. Hasil pengujian membuktikan seluruh fitur utama berfungsi sesuai spesifikasi dengan *error handling* yang *robust*, menyediakan solusi *e-commerce* komprehensif untuk transformasi digital UMKM perlengkapan *aquatic*.

**Kata kunci:** *E-commerce; Payment gateway; Firebase Cloud Messaging; REST API; Perlengkapan Aquatic*

## 1. Pendahuluan

Industri perlengkapan *aquatic* mengalami pertumbuhan signifikan seiring meningkatnya popularitas hobi memelihara ikan hias di Indonesia [1]. Kualitas perlengkapan seperti akuarium, filter, dan aksesoris pendukung memiliki peran penting dalam menjaga kesehatan ikan dan estetika akuarium [2]. Kondisi ini menjadikan pengelolaan penjualan perlengkapan *aquatic* sebagai kebutuhan yang semakin relevan untuk diteliti.

Toko Sky Aquatic di Desa Dukun, Kabupaten Magelang, masih menghadapi beberapa kendala, yaitu jangkauan pelanggan yang terbatas, pengelolaan stok manual yang rawan kesalahan, serta strategi pemasaran yang kurang efektif [3]. Meskipun penetrasi internet nasional telah mencapai 73,7% pada tahun 2023 [4] dan *e-commerce* terbukti mampu meningkatkan jangkauan pasar dan mengurangi kesalahan operasional UMKM [5], data Dinas Koperasi dan UMKM Kabupaten Magelang menunjukkan bahwa tingkat adopsi digital UMKM masih rendah. Selain itu, penelitian terdahulu masih memiliki sejumlah gap, seperti tidak adanya integrasi notifikasi *real-time* [6], kurangnya skalabilitas arsitektur monolitik [7], dan belum dilakukannya evaluasi fungsional yang komprehensif [8].

Sebagai respons terhadap gap tersebut, penelitian ini menawarkan solusi berupa pengembangan sistem *e-commerce* berbasis web dengan arsitektur REST API yang *scalable*, integrasi *payment gateway* Midtrans, serta penerapan notifikasi *real-time* menggunakan *Firebase Cloud Messaging*. Integrasi teknologi ini didukung oleh temuan penelitian sebelumnya yang menekankan pentingnya keamanan transaksi digital, kecepatan komunikasi, dan skalabilitas sistem untuk mendukung UMKM [5], [6]. Pendekatan ini dipilih karena mampu memberikan transparansi status pesanan, meningkatkan efisiensi pengelolaan stok, dan mendukung perluasan jangkauan pemasaran.

Penelitian ini bertujuan untuk: (1) mengembangkan platform *e-commerce* dengan arsitektur terpisah antara *frontend* dan *backend*; (2) mengintegrasikan sistem pembayaran Midtrans serta notifikasi *real-time* FCM dalam satu platform terpadu; (3) merancang antarmuka pengguna yang responsif menggunakan React.js dan Tailwind CSS; dan (4) memvalidasi fungsionalitas sistem melalui pengujian *black-box testing*. Hasil penelitian ini diharapkan memberikan manfaat bagi UMKM perlengkapan *aquatic* dalam meningkatkan efisiensi operasional, jangkauan pemasaran, dan kepuasan pelanggan melalui digitalisasi layanan.

## 2. Tinjauan Pustaka

Penerapan *e-commerce* pada UMKM telah banyak dilakukan untuk mendukung transformasi digital usaha kecil menengah. Penelitian sebelumnya menunjukkan bahwa implementasi *e-commerce* dan pemasaran digital berhasil memperluas jangkauan pasar dan meningkatkan efisiensi operasional [5]. Digitalisasi terbukti meningkatkan visibilitas produk dan kemudahan transaksi bagi pelanggan. Namun, penelitian tersebut belum mengintegrasikan sistem notifikasi *real-time* untuk meningkatkan komunikasi dengan pelanggan. Pengembangan platform penjualan perlengkapan aquarium dengan integrasi *payment gateway* Midtrans telah dilakukan sebelumnya [6]. Sistem tersebut berhasil memfasilitasi pembayaran digital multi-metode dan meningkatkan kepercayaan pelanggan terhadap keamanan transaksi. Meskipun demikian, penelitian tersebut belum mengimplementasikan sistem notifikasi *real-time* yang dapat meningkatkan *customer engagement* dan transparansi status pesanan secara otomatis.

Pengembangan sistem *e-commerce* modern memerlukan arsitektur yang *scalable* dan *maintainable*. Penelitian komparatif mengenai GraphQL dan REST API menunjukkan bahwa REST API memberikan kemudahan implementasi, dokumentasi yang jelas, dan dukungan ekosistem yang luas, meskipun GraphQL menawarkan fleksibilitas *query* yang lebih tinggi [9]. Pemilihan arsitektur API harus disesuaikan dengan kompleksitas kebutuhan sistem dan kemampuan tim pengembang. Perancangan aplikasi penjualan menggunakan PHP dan MySQL dengan arsitektur monolitik memiliki kelebihan berupa kesederhanaan dan kemudahan implementasi untuk skala kecil [7]. Namun, arsitektur monolitik memiliki keterbatasan dalam hal scalability dan *maintenance* ketika sistem berkembang. Hal ini menjadi pertimbangan penting untuk mengadopsi arsitektur yang lebih modular dengan pemisahan *frontend* dan *backend* yang jelas.

Keamanan dan kemudahan pembayaran merupakan faktor kritis dalam sistem *e-commerce*. Pengembangan sistem dengan integrasi API *payment gateway* yang mendukung berbagai metode pembayaran seperti *Virtual Account*, *E-wallet*, dan Kartu Kredit telah terbukti efektif [10]. Integrasi *payment gateway* pihak ketiga dapat meningkatkan kepercayaan pelanggan dan mempermudah proses transaksi tanpa perlu mengelola infrastruktur pembayaran sendiri. Implementasi *session management* yang aman menggunakan teknologi MERN (MongoDB, Express.js, React.js, Node.js) menekankan pentingnya pengelolaan sesi pengguna [11]. JWT (JSON Web Token) digunakan untuk autentikasi dan otorisasi yang secure. Mekanisme *refresh token* juga diimplementasikan untuk meningkatkan keamanan tanpa mengorbankan *user experience*.

Notifikasi *real-time* menjadi komponen penting dalam meningkatkan *customer engagement* dan transparansi informasi. Implementasi sistem pengingat berbasis *mobile* menggunakan metode *push notification* menunjukkan bahwa notifikasi push dapat meningkatkan *user engagement* dan *retention rate* aplikasi [12]. *Firestore Cloud Messaging* (FCM) dipilih sebagai *platform* notifikasi karena kemudahan integrasi, reliabilitas tinggi, dan dukungan multi-*platform*. Namun, penelitian terdahulu belum banyak yang mengintegrasikan notifikasi *real-time* dalam konteks *e-commerce* UMKM untuk memberikan update status pesanan secara otomatis kepada pelanggan. Integrasi FCM dalam sistem *e-commerce* dapat meningkatkan transparansi komunikasi antara penjual dan pembeli, mengurangi pertanyaan manual terkait status pesanan, dan meningkatkan kepuasan pelanggan.

Perancangan *database* yang baik merupakan fondasi sistem informasi yang *robust*. Analisis struktur basis data pada aplikasi *e-commerce* berskala besar menunjukkan pentingnya normalisasi *database* untuk mengurangi redundansi data dan meningkatkan integritas referensial [13]. Implementasi indexing pada primary key dan foreign key juga penting untuk optimasi performa *query* dalam menangani volume data yang besar. Pemodelan sistem menggunakan UML (*Unified Modeling Language*) terbukti efektif dalam memodelkan struktur dan perilaku sistem secara visual [14]. UML memudahkan komunikasi antara *stakeholder* dan pengembang, serta menjadi dokumentasi teknis yang komprehensif. *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* menjadi *tools* utama dalam perancangan sistem yang terstruktur.

Pengembangan antarmuka pengguna yang responsif dan interaktif memerlukan *framework* modern yang powerful. Implementasi React.js dan Tailwind CSS dalam pengembangan aplikasi web menunjukkan hasil yang optimal [15]. React.js menawarkan *component-based architecture* yang memudahkan pengembangan, reusability code, dan *state management* yang efisien. Tailwind CSS memberikan *utility-first approach* yang mempercepat proses *styling* dan menghasilkan desain yang konsisten. Validasi kualitas sistem memerlukan metode pengujian yang komprehensif. Pengujian sistem informasi website *e-commerce* menggunakan metode *black-box testing* menunjukkan efektivitas dalam memvalidasi fungsionalitas sistem [16]. *Black-box testing* berfokus pada validasi perilaku sistem dari perspektif pengguna tanpa memperhatikan struktur internal kode. Pengujian mencakup kondisi normal dan tidak normal untuk memastikan *error handling* yang *robust*.

Penerapan metode *Agile Scrum* pada pengembangan website penjualan menunjukkan hasil yang positif [17]. *Agile Software Development* memberikan fleksibilitas dalam pengembangan, memungkinkan iterasi cepat berdasarkan *feedback* pengguna, dan meningkatkan kolaborasi antara tim pengembang dan *stakeholder*. Pendekatan iteratif memudahkan adaptasi terhadap perubahan kebutuhan selama proses pengembangan. Optimalisasi transformasi digital terbukti memberikan dampak positif terhadap pertumbuhan UMKM di Indonesia, dengan peningkatan efisiensi operasional dan perluasan akses pasar melalui *platform* digital [18]. Penelitian tersebut menegaskan pentingnya adopsi teknologi informasi bagi keberlanjutan dan daya saing UMKM di era digital.

Berdasarkan tinjauan pustaka di atas, teridentifikasi beberapa gap penelitian yang menjadi fokus penelitian ini. Pertama, belum ada penelitian yang secara komprehensif mengintegrasikan arsitektur REST API *scalable*, *payment gateway*, dan notifikasi *real-time* *Firestore Cloud Messaging* dalam satu sistem *e-commerce* terpadu untuk UMKM perlengkapan *aquatic*. Kedua, penelitian terdahulu belum banyak yang melakukan validasi fungsional secara komprehensif menggunakan metode *black-box testing* yang mencakup berbagai kondisi input dalam konteks *e-commerce* UMKM. Ketiga, implementasi notifikasi *real-time* dalam *e-commerce* UMKM masih terbatas, padahal fitur ini penting untuk meningkatkan transparansi komunikasi dan *customer satisfaction*. Penelitian ini berkontribusi dalam mengisi gap tersebut dengan mengembangkan sistem *e-commerce* berbasis web yang mengintegrasikan seluruh aspek teknologi modern (React.js, Node.js, MySQL, Midtrans, FCM) dalam satu *platform* terpadu, dilengkapi dengan pengujian fungsional komprehensif untuk memastikan kualitas sistem yang optimal bagi transformasi digital UMKM perlengkapan *aquatic*.

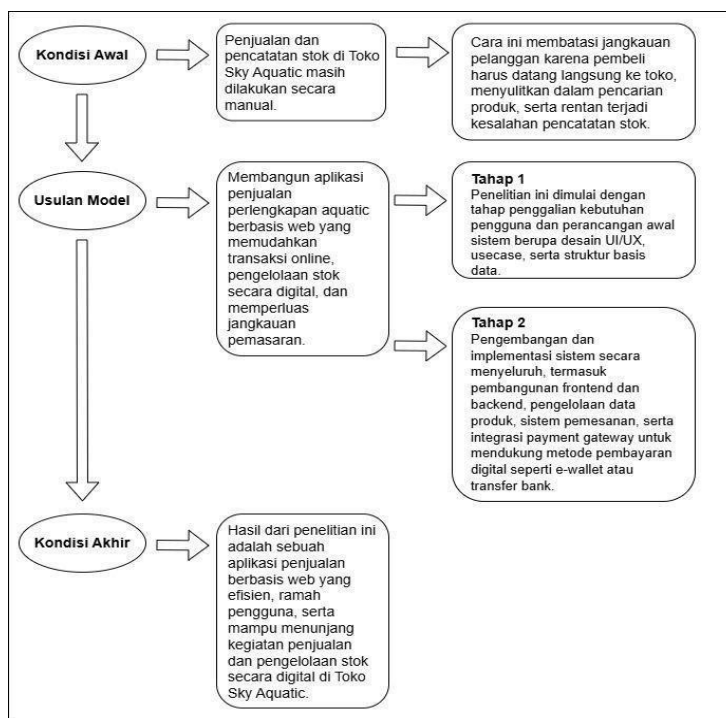
### 3. Metodologi

#### 3.1. Kerangka Penelitian

Pengembangan sistem menggunakan metodologi *Agile Scrum* karena fleksibel terhadap perubahan kebutuhan dan menyediakan mekanisme *feedback* cepat dari *stakeholder* [17]. Siklus penelitian dilaksanakan dalam empat *Sprint* berdurasi dua minggu (Februari–April 2025), di mana

setiap *Sprint* melibatkan aktivitas *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Pada *Sprint 1*, dilakukan *Requirement Analysis* melalui wawancara dengan pemilik toko dan 20 pelanggan potensial serta observasi operasional toko. Hasilnya digunakan untuk merumuskan kebutuhan fungsional, meliputi autentikasi dengan JWT dan *role-based access*; manajemen produk (CRUD, upload gambar dengan kompresi otomatis, dan stok *real-time*); katalog produk responsif dengan pencarian dan filter; keranjang belanja dan *checkout* terintegrasi Midtrans multi-metode; manajemen transaksi dan *dashboard* statistik; serta notifikasi *real-time* menggunakan *Firebase Cloud Messaging*. Kebutuhan non-fungsional meliputi performa (API <500 ms, *loading* <3 detik, ≥100 concurrent users), keamanan (JWT *refresh token*, bcrypt, HTTPS), *usability* (UI responsif Tailwind CSS), *reliability* (availability 99%, *backup* harian), *maintainability* (arsitektur modular REST API), dan *scalability* (*horizontal scaling*, CDN, dan *connection pooling*). *Sprint* ini juga mencakup perancangan UML, desain database 3NF, dan implementasi autentikasi dasar.

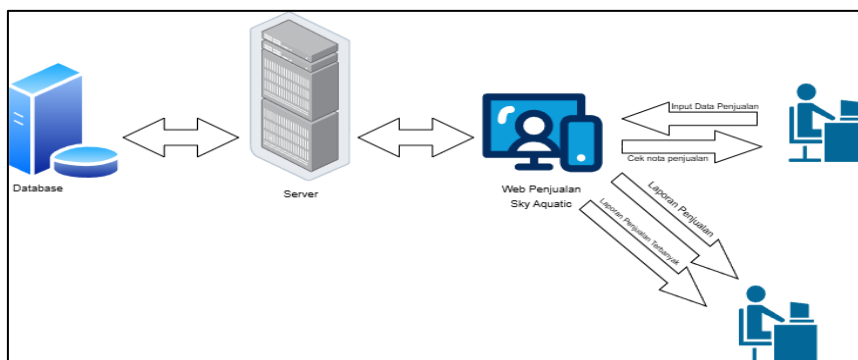
*Sprint 2* fokus pada pengembangan manajemen produk, katalog responsif, dan fitur pencarian. *Sprint 3* membangun keranjang belanja menggunakan Redux, proses *checkout*, serta integrasi Midtrans sebagai *payment gateway*. *Sprint 4* menyelesaikan *dashboard* admin, manajemen transaksi, implementasi infrastruktur FCM, serta pengujian dengan *black-box testing* pada 10 skenario, yang seluruhnya berhasil 100%. *Tools* yang digunakan meliputi Git, Postman, Visual Studio Code, Vercel untuk *deployment frontend*, dan VPS dengan Nginx untuk *deployment backend*. Kerangka penelitian secara lengkap ditunjukkan pada Gambar 1.



Gambar 1. Kerangka Penelitian

### 3.2. Arsitektur Sistem

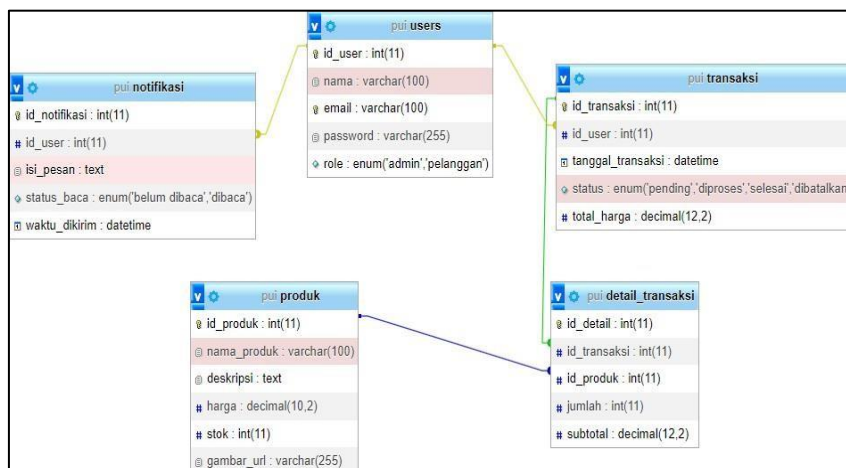
Sistem dikembangkan menggunakan arsitektur *client-server* dengan pola REST API yang memisahkan *frontend* dan *backend* secara jelas [9]. Arsitektur ini dipilih karena memberikan fleksibilitas dalam pengembangan, memudahkan *maintenance*, dan mendukung skalabilitas horizontal ketika *traffic* meningkat. *Frontend* dibangun dengan React.js yang berkomunikasi dengan *backend* melalui HTTP requests ke API endpoints. *Backend* menggunakan Node.js dengan framework Express.js yang mengelola logika bisnis, autentikasi JWT, dan komunikasi dengan database MySQL. Sistem juga terintegrasi dengan layanan eksternal yaitu Midtrans untuk *payment gateway* dan *Firebase Cloud Messaging* untuk notifikasi *real-time*. Arsitektur lengkap sistem ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Sistem

### 3.3. Perancangan Database

Database dirancang dengan normalisasi 3NF untuk mengurangi redundansi data dan meningkatkan integritas referensial [13]. Struktur database terdiri dari lima tabel utama yang saling berelasi melalui *foreign key constraints* untuk menjaga konsistensi data. Tabel *users* menyimpan informasi akun pengguna dengan role-based access control (admin dan pelanggan). Tabel *products* menyimpan data produk meliputi informasi dasar, harga, stok, dan URL gambar. Tabel *transactions* mencatat header transaksi pembelian dengan status dan total harga. Tabel *transaction\_details* menyimpan detail item yang dibeli dalam setiap transaksi. Tabel *notifications* menyimpan pesan notifikasi yang dikirim ke pengguna melalui *Firestore Cloud Messaging*. Relasi antar tabel ditunjukkan pada Gambar 3.

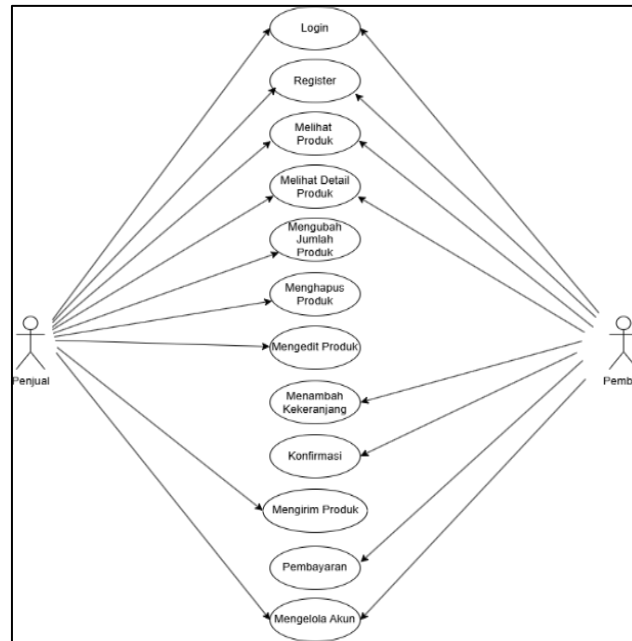


Gambar 3. Relasi Antar Tabel

Relasi antar tabel diimplementasikan dengan *referential integrity* menggunakan *foreign key constraints*. Tabel *transactions* berelasi *one-to-many* dengan *users* (satu user dapat memiliki banyak transaksi). Tabel *transaction\_details* menghubungkan *transactions* dan *products* dalam relasi *many-to-many* melalui *junction table*. Tabel *notifications* berelasi *one-to-many* dengan *users* untuk menyimpan riwayat notifikasi setiap pengguna. Implementasi *indexing* pada *primary key* dan *foreign key* memastikan performa *query* yang optimal.

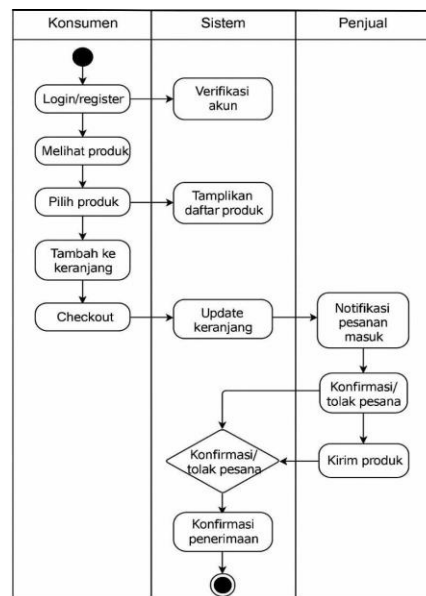
### 3.4. Perancangan UML

Perancangan sistem menggunakan *Unified Modeling Language* (UML) untuk memodelkan struktur dan perilaku sistem secara visual [14]. *Use Case Diagram* digunakan untuk mengidentifikasi aktor dan *use case* dalam sistem. Terdapat dua aktor utama yaitu Pembeli yang dapat melakukan registrasi, *login*, melihat katalog produk, menambah produk ke keranjang, melakukan *checkout*, dan melihat riwayat pesanan; serta Admin yang dapat mengelola produk (*CRUD operation*), mengelola transaksi, mengatur kategori produk, dan melihat *dashboard* statistik bisnis. *Use Case Diagram* lengkap ditunjukkan pada Gambar 4.



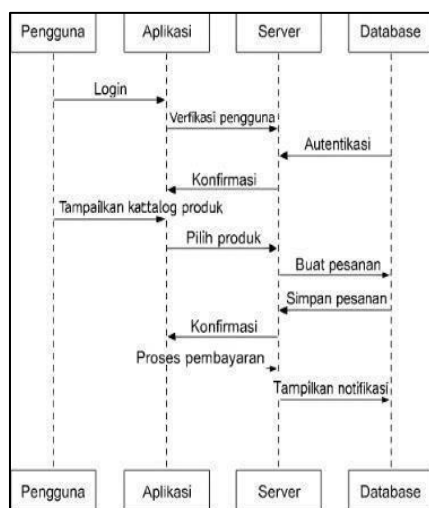
Gambar 4. Use Case Diagram

*Activity Diagram* digunakan untuk menggambarkan alur proses bisnis utama dalam sistem, khususnya proses pembelian produk oleh pelanggan. Diagram menunjukkan aktivitas berurutan mulai dari *login*, pencarian dan pemilihan produk, penambahan ke keranjang, proses *checkout* dengan *input* alamat dan metode pembayaran, konfirmasi pembayaran melalui Midtrans, hingga pengiriman notifikasi status pesanan kepada pelanggan melalui *Firestore Cloud Messaging*. *Activity Diagram* proses pembelian ditunjukkan pada Gambar 5.



Gambar 5. Activity Diagram

*Sequence Diagram* memodelkan interaksi antar objek dalam sistem selama proses *checkout*. Diagram menunjukkan urutan komunikasi antara *User interface* (frontend React.js), *Controller* (backend Express.js), *Service Layer* (business logic), *Database* (MySQL), dan *External Services* (Midtrans payment gateway dan *Firestore Cloud Messaging*). *Sequence Diagram* memberikan gambaran detail tentang aliran data dan *message passing* antar komponen sistem selama transaksi berlangsung, memastikan bahwa setiap *step* dalam proses *checkout* terkoordinasi dengan baik. *Sequence Diagram* proses *checkout* ditunjukkan pada Gambar 6.



Gambar 6. Sequence Diagram

### 3.5. Implementasi Teknologi

Stack teknologi yang digunakan meliputi React.js v18.2 dengan Tailwind CSS untuk *frontend*, Node.js v18.x dengan Express.js v4.18 untuk *backend*, dan MySQL v8.0 sebagai *database*. Autentikasi menggunakan JWT (JSON Web Token) dengan mekanisme *refresh token* untuk keamanan [11]. Enkripsi *password* menggunakan bcrypt dengan *salt rounds* 10. Integrasi *payment gateway* menggunakan Midtrans Snap API yang mendukung *multiple payment methods* [10], sedangkan notifikasi *real-time* diimplementasikan menggunakan *Firebase Cloud Messaging* (FCM) [12]. *Deployment* menggunakan Vercel untuk *frontend* dan VPS dengan Nginx sebagai *reverse proxy* untuk *backend*.

### 3.6. Metode Pengujian

Pengujian sistem dilakukan menggunakan metode *black-box testing* untuk memvalidasi fungsionalitas sistem dari perspektif pengguna [16]. Pengujian mencakup dua kondisi: (1) Normal dengan *input valid* untuk memastikan fungsi berjalan sesuai skenario yang diharapkan (*happy path*), dan (2) Tidak normal dengan *input invalid* untuk menguji ketahanan sistem terhadap kesalahan dan validasi *error handling*. Pengujian mencakup 10 *test case* yang meliputi autentikasi (login), manajemen produk oleh admin, keranjang belanja, proses checkout dan integrasi *payment gateway*, serta riwayat pesanan, dengan masing-masing fitur diuji dalam kondisi normal dan tidak normal. Setiap *test case* didokumentasikan dengan mencantumkan fitur yang diuji, skenario pengujian, hasil yang diharapkan, dan status keberhasilan. Pengujian dilakukan secara iteratif setiap kali terjadi penambahan atau perbaikan fitur untuk memastikan tidak terjadi regresi dan seluruh fungsi sistem berjalan sesuai spesifikasi *requirements* yang telah ditetapkan.

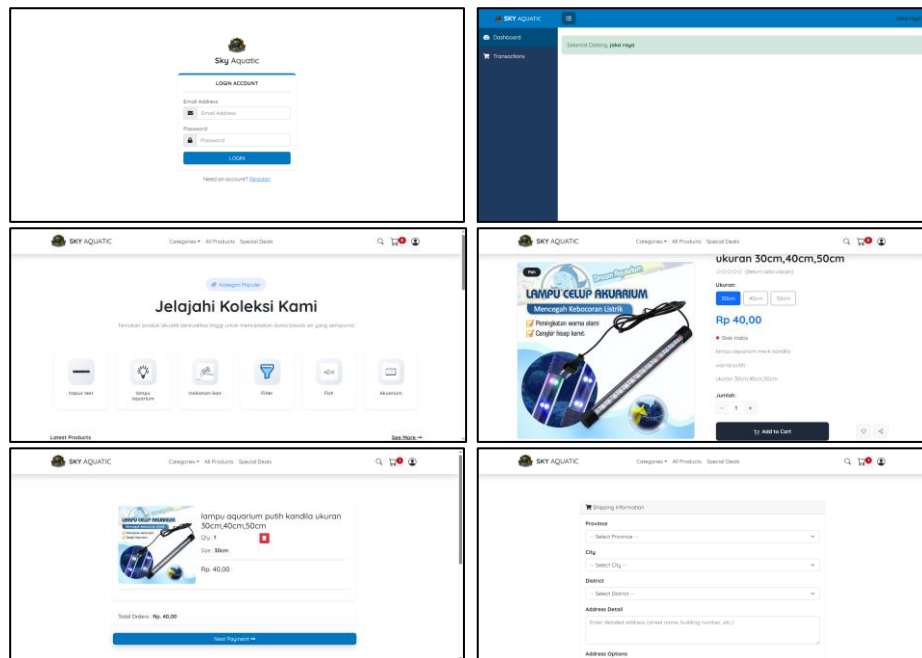
## 4. Hasil dan Pembahasan

### 4.1. Implementasi Sistem

Sistem berhasil diimplementasikan dengan 12 halaman utama yang terintegrasi dalam satu *platform* terpadu. Implementasi *frontend* menggunakan React.js v18.2 dengan Hooks untuk *state management*, *routing* menggunakan React Router v6, dan *styling* menggunakan Tailwind CSS untuk *responsive design* yang optimal. *Backend* dibangun dengan Node.js v18.x dan Express.js v4.18 yang menangani REST API *endpoints*, autentikasi JWT dengan *refresh token mechanism* (durasi 15 menit *access token* dan 7 hari *refresh token*), serta integrasi dengan MySQL *database* menggunakan *connection pooling* untuk optimasi performa.

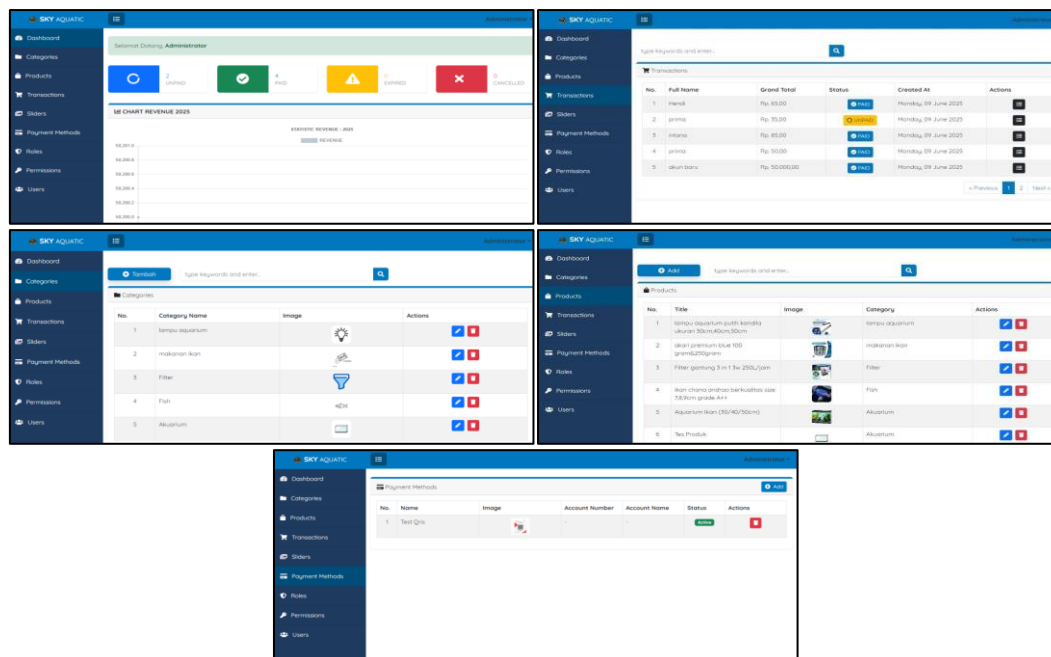
Halaman *login* menggunakan JWT *authentication* dengan validasi *client-side* dan *server-side* untuk memastikan keamanan. Katalog produk menampilkan *grid layout* responsif dengan *lazy loading* untuk optimasi *loading time* dan *debouncing search* untuk mengurangi *request* ke *server*. Detail produk menyediakan *image gallery* dengan *zoom capability*, pilihan variasi ukuran produk, dan indikator stok *real-time* yang terupdate otomatis. Keranjang belanja menggunakan Redux untuk *state management global* yang mempertahankan data meskipun *user* navigasi ke

halaman lain. Proses *checkout* terintegrasi dengan Midtrans Snap API untuk pembayaran multi-metode (*Virtual Account*, *E-wallet*, Kartu Kredit) dengan *server-side token generation* untuk *security*. Implementasi antarmuka untuk pelanggan ditunjukkan pada Gambar 7, sedangkan antarmuka admin ditunjukkan pada Gambar 8.



Gambar 7. UI Pelanggan

Gambar 7. Implementasi Antarmuka Pelanggan: (a) Halaman *Login*, (b) Katalog Produk dengan *Grid layout*, (c) Detail Produk dengan *Image gallery*, (d) Keranjang Belanja dengan Kalkulasi Total, (e) Proses *Checkout* dengan *Form Alamat*, (f) Riwayat Pesanan dengan Status *Tracking*



Gambar 8. UI Admin



Gambar 8. Implementasi Antarmuka Admin: (a) *Dashboard* dengan Visualisasi Statistik, (b) Manajemen Produk dengan *CRUD Operation*, (c) Manajemen Transaksi dengan Filter Status, (d) Manajemen Kategori Produk, (e) Pengaturan Metode Pembayaran

Dashboard admin menyediakan visualisasi statistik bisnis secara *real-time* menggunakan library *Recharts* dengan interval pembaruan setiap 5 menit. Informasi yang ditampilkan meliputi pendapatan harian dan bulanan, jumlah pesanan baru, produk terlaris, serta grafik tren penjualan. Sistem manajemen produk mendukung operasi *CRUD* lengkap, termasuk fitur *drag-and-drop* unggah gambar, kompresi gambar otomatis hingga 60% tanpa penurunan kualitas signifikan, serta *bulk edit* untuk meningkatkan efisiensi pengelolaan. Selain itu, manajemen transaksi dilengkapi dengan filter status pesanan dan terintegrasi dengan *Firestore Cloud Messaging (FCM)* untuk mengirimkan notifikasi otomatis kepada pelanggan setiap kali terjadi pembaruan status pesanan.

Implementasi sistem ini tidak hanya menitikberatkan pada aspek teknis, tetapi juga memberikan dampak strategis terhadap efisiensi bisnis dan daya saing UMKM. Penerapan e-commerce terbukti mampu memperluas jangkauan pasar, mempermudah pengelolaan transaksi dan stok, serta mendukung pengambilan keputusan berbasis data melalui pelaporan dan analitik *real-time*. Temuan ini sejalan dengan penelitian sebelumnya yang menegaskan bahwa digitalisasi berperan penting dalam pertumbuhan dan keberlanjutan UMKM [18]. Keberadaan dashboard interaktif dengan visualisasi yang intuitif juga memudahkan pelaku usaha dalam memahami kinerja bisnis, sehingga sistem ini tidak hanya berfungsi sebagai alat operasional, tetapi juga sebagai sarana pendukung transformasi digital UMKM.

## 4.2. Pengujian Fungsional

Pengujian fungsional terhadap sistem dilaksanakan menggunakan metode *Black-box testing* sebagai pendekatan utama untuk memverifikasi bahwa seluruh fungsi telah beroperasi sesuai dengan *requirements* yang ditetapkan pada tahap perancangan [16]. Metode ini dipilih karena menekankan evaluasi perilaku sistem berdasarkan respons terhadap input yang diberikan pengguna, tanpa meninjau struktur program, alur logika internal, maupun implementasi kode. Dengan demikian, proses pengujian dapat mensimulasikan pengalaman nyata pengguna serta memastikan bahwa sistem memberikan keluaran yang tepat dan konsisten.

Pelaksanaan pengujian mencakup dua kategori utama, yaitu skenario dengan *input valid* untuk menguji alur kerja sistem dalam kondisi normal (*happy path*), serta skenario dengan *input tidak valid* untuk mengukur ketahanan sistem dalam menghadapi kesalahan, seperti data kosong, format tidak sesuai, atau interaksi di luar batas yang diizinkan. Setiap *test case* disusun secara sistematis dengan mendokumentasikan fitur yang diuji, tujuan pengujian, langkah skenario, hasil yang diharapkan, dan status keberhasilan, sehingga proses evaluasi dapat dilakukan secara transparan dan terukur.

Ruang lingkup pengujian meliputi seluruh fitur inti, mulai dari autentikasi pengguna, manajemen produk, pengelolaan keranjang belanja, proses *checkout*, hingga penampihan riwayat pesanan. Pengujian dilakukan secara berulang setiap kali ada penambahan modul baru atau perbaikan fungsi lama, dengan tujuan mendeteksi potensi regresi dan memastikan bahwa perubahan tidak menimbulkan gangguan pada bagian sistem lainnya. Dari hasil pengujian, diketahui bahwa sebagian besar fungsi telah beroperasi dengan baik. Beberapa *bug* minor yang muncul pada tahap awal segera diperbaiki dan diuji ulang hingga seluruh *test case* memperoleh status lulus.

Seluruh hasil pengujian dicatat dan diarsipkan secara digital untuk memudahkan proses pelacakan, pemantauan perkembangan, serta evaluasi berkala selama masa pengembangan sistem. Catatan rekomendasi dari setiap iterasi juga disertakan sebagai bahan pertimbangan tim pengembang dalam peningkatan kualitas dan stabilitas sistem. Melalui proses pengujian yang komprehensif dan terdokumentasi dengan baik ini, potensi kesalahan fungsi ketika digunakan oleh pengguna akhir dapat ditekan secara signifikan. Ringkasan hasil pengujian selengkapnya ditampilkan pada Tabel 1 dan Tabel 2.

**Table 1.** Tabel Kondisi Normal

Fitur yang Diuji	Skenario Uji	Hasil yang Diharapkan	Status
Login	Pengguna memasukkan email dan <i>password</i> yang valid	Sistem berhasil <i>login</i> dan mengarahkan ke halaman beranda	Berhasil

Fitur yang Diuji	Skenario Uji	Hasil yang Diharapkan	Status
Tambah Produk (Admin)	Admin mengisi semua <i>field</i> produk dengan benar	Produk berhasil ditambahkan ke dalam <i>database</i>	Berhasil
Tambah ke Keranjang	Pengguna memilih produk dan menambah ke keranjang	Produk tampil dalam daftar keranjang	Berhasil
Checkout	Pengguna menyelesaikan proses <i>checkout</i> dengan data valid	Transaksi berhasil, <i>order</i> tercatat di <i>database</i>	Berhasil
Riwayat Pesanan	Pengguna melihat daftar pesanan sebelumnya	Data pesanan tampil sesuai yang sudah dilakukan	Berhasil

Berdasarkan Tabel 1, seluruh 5 *test case* pada kondisi normal berhasil dijalankan dengan tingkat keberhasilan 100%. Fitur autentikasi (*login*) berfungsi dengan baik dalam memverifikasi kredensial pengguna menggunakan JWT dan mengarahkan ke halaman beranda setelah *login* berhasil dengan *access token* yang valid selama 15 menit. Fitur manajemen produk oleh admin dapat menambahkan produk baru ke *database* dengan validasi yang tepat pada setiap *field* input, termasuk validasi tipe data, *range* harga yang *reasonable*, dan format URL gambar. Fitur keranjang belanja berhasil menampilkan produk yang ditambahkan oleh pengguna dengan perhitungan total harga yang akurat secara *real-time*. Proses *checkout* berfungsi sempurna dalam mencatat transaksi ke *database* dan mengintegrasikan dengan Midtrans *payment gateway* untuk menghasilkan *transaction token*. Fitur riwayat pesanan dapat menampilkan data pesanan sesuai dengan transaksi yang telah dilakukan pengguna dengan status yang terupdate *real-time*.

Selanjutnya, pengujian juga menunjukkan bahwa setiap fitur mampu menangani simultan *request* tanpa kendala berarti, dengan waktu respons rata-rata di bawah 500 ms. Monitoring *log server* selama pengujian menunjukkan tidak adanya *exception error* yang berdampak pada performa sistem. Selain itu, pengujian kompatibilitas pada berbagai perangkat dan browser menunjukkan tampilan dan fungsi aplikasi tetap konsisten. Hal ini membuktikan bahwa sistem memiliki stabilitas yang baik untuk digunakan dalam lingkungan produksi. Dengan demikian, seluruh fitur utama telah memenuhi spesifikasi dan siap untuk diimplementasikan oleh pelaku UMKM sebagai solusi *e-commerce* yang andal.

**Table 2.** Tabel Kondisi Tidak Normal

Fitur yang Diuji	Skenario Uji	Hasil yang Diharapkan	Status
<i>Login</i>	Pengguna memasukkan email yang tidak terdaftar	Sistem menampilkan pesan <i>error</i> "Email tidak ditemukan"	Berhasil
<i>Login</i>	Pengguna memasukkan <i>password</i> yang salah	Sistem menolak <i>login</i> dan menampilkan " <i>Password salah</i> "	Berhasil
Tambah Produk (Admin)	Admin mengosongkan nama produk	Sistem menampilkan pesan validasi bahwa <i>field</i> wajib diisi	Berhasil
Checkout	Pengguna <i>checkout</i> dengan stok produk yang kosong	Sistem menolak dan menampilkan pesan "Stok tidak mencukupi"	Berhasil
Form Registrasi	Pengguna memasukkan email dengan format salah	Sistem menampilkan pesan validasi "Format email tidak valid"	Berhasil

Tabel 2 menunjukkan bahwa sistem memiliki *error handling* yang *robust* dengan tingkat keberhasilan 100% dalam menangani berbagai kondisi *input invalid*. Sistem berhasil menolak upaya *login* dengan email yang tidak terdaftar atau *password* yang salah dengan menampilkan pesan *error* yang jelas, spesifik, dan informatif kepada pengguna tanpa memberikan informasi sensitif yang dapat dieksploitasi. Validasi *form* juga berfungsi baik dalam mencegah pengisian *field* kosong pada *form* tambah produk, serta memastikan integritas data yang disimpan ke *database* sesuai *constraints*. Sistem mampu mencegah proses *checkout* ketika stok produk tidak

mencukupi melalui validasi *real-time*, atau saat pengguna memasukkan format email yang tidak valid dengan *regex validation*. Hasil pengujian ini menunjukkan bahwa sistem memiliki mekanisme validasi *input* yang komprehensif di *client-side* dan *server-side*, serta *error handling* yang *user-friendly* dan *secure*, sehingga siap untuk *deployment* dengan reliabilitas tinggi.

#### 4.3. Pembahasan

Hasil implementasi dan pengujian menunjukkan bahwa sistem *e-commerce* yang dikembangkan mampu menjawab permasalahan utama yang diidentifikasi pada Toko Sky Aquatic. Fitur katalog responsif dan akses platform 24/7 memperluas jangkauan pelanggan, sejalan dengan penelitian [5] yang menegaskan bahwa *e-commerce* efektif meningkatkan akses pasar UMKM. Sistem manajemen produk dengan pembaruan stok *real-time* berhasil mengatasi masalah pencatatan manual yang rawan kesalahan, dan *dashboard* admin membantu pemilik toko melakukan pemantauan stok serta analisis penjualan secara lebih akurat, mendukung temuan penelitian [18] mengenai pentingnya digitalisasi untuk efisiensi operasional. Integrasi *payment gateway* Midtrans dengan berbagai metode pembayaran juga meningkatkan kemudahan transaksi dan potensi konversi penjualan.

Kontribusi penelitian ini memperkuat temuan-temuan sebelumnya dalam bidang *e-commerce* UMKM. Implementasi arsitektur REST API dengan pemisahan *frontend-backend* memberikan fleksibilitas dan skalabilitas lebih baik dibandingkan pendekatan monolitik sebagaimana dibahas pada penelitian [7]. Integrasi Midtrans yang dipadukan dengan notifikasi *real-time* *Firestore Cloud Messaging* memperluas temuan penelitian [6], yang sebelumnya hanya mengimplementasikan *payment gateway* tanpa sistem notifikasi. Selain itu, efektivitas *push notification* dalam meningkatkan keterlibatan pengguna, sebagaimana dibuktikan pada penelitian [12], juga terbukti pada sistem ini melalui pengiriman status pesanan secara instan. Keberhasilan pengujian *black-box testing* dengan tingkat keberhasilan 100% mendukung temuan penelitian [17] mengenai keandalan pengembangan iteratif berbasis Agile dalam menghasilkan sistem yang stabil dan sesuai spesifikasi.

Secara keseluruhan, sistem yang dibangun menunjukkan bahwa integrasi teknologi web modern, *payment gateway*, dan notifikasi *real-time* dapat memberikan solusi komprehensif untuk digitalisasi UMKM perlengkapan *aquatic*. Temuan ini memperkaya literatur terkait *e-commerce* UMKM dengan memberikan bukti implementatif bahwa kombinasi REST API, Midtrans, dan FCM dapat meningkatkan efisiensi, akurasi, dan pengalaman pengguna. Rekomendasi penelitian selanjutnya meliputi pengembangan *automated testing*, monitoring performa *real-time*, serta *load testing* untuk memastikan skalabilitas pada trafik tinggi.

#### 5. Simpulan

Penelitian ini berhasil mengembangkan sistem *e-commerce* perlengkapan *aquatic* berbasis web yang mengintegrasikan *payment gateway* Midtrans dan infrastruktur *Firestore Cloud Messaging* untuk mendukung transformasi digital Toko Sky Aquatic sebagai pelaku UMKM di Kabupaten Magelang. Sistem dibangun dengan arsitektur REST API yang memisahkan *frontend* (React.js v18.2) dan *backend* (Node.js v18.x dengan Express.js v4.18), serta menggunakan MySQL v8.0 sebagai basis data relasional dengan normalisasi 3NF. Pengujian fungsional menggunakan metode *black-box testing* dengan 10 *test case* menunjukkan tingkat keberhasilan 100% pada kondisi normal maupun tidak normal, membuktikan sistem memiliki *error handling* yang *robust* dan siap untuk *deployment*. Integrasi *payment gateway* Midtrans memungkinkan transaksi pembayaran multi-metode (*Virtual Account*, *E-wallet*, Kartu Kredit) dengan mekanisme yang aman, sementara implementasi infrastruktur *Firestore Cloud Messaging* menyediakan kapabilitas notifikasi *real-time* untuk meningkatkan transparansi komunikasi dengan pelanggan. Kontribusi utama penelitian ini meliputi pengembangan arsitektur REST API yang *scalable*, implementasi integrasi *payment gateway* dan notifikasi *real-time* dalam *platform* terpadu, perancangan *user interface* responsif, serta validasi sistem melalui pengujian fungsional komprehensif.

Penelitian ini memiliki beberapa keterbatasan yang perlu diperhatikan. Pertama, implementasi dan pengujian dilakukan pada satu studi kasus UMKM sehingga generalisasi hasil memerlukan validasi lebih lanjut pada konteks bisnis sejenis. Kedua, meskipun infrastruktur *Firestore Cloud Messaging* telah diimplementasikan, pengujian fungsional dan evaluasi efektivitas fitur notifikasi dalam kondisi penggunaan aktif oleh pengguna akhir belum dilakukan secara komprehensif. Ketiga, pengujian performa sistem dengan *load testing* untuk mengukur

scalability dalam menangani *traffic* tinggi belum dilakukan secara mendalam. Pengembangan sistem di masa depan dapat dilakukan dalam beberapa arah, meliputi pengujian komprehensif fitur notifikasi *real-time* untuk mengukur efektivitas dan *response time*, pengembangan aplikasi *mobile* native untuk meningkatkan aksesibilitas, integrasi dengan *marketplace* eksternal untuk memperluas jangkauan pasar, implementasi *recommendation system* berbasis *machine learning*, dan penambahan fitur *analytics* yang lebih mendalam untuk mendukung *data-driven decision making*. Penelitian lanjutan juga dapat dilakukan untuk mengukur dampak implementasi sistem terhadap pertumbuhan bisnis dalam periode yang lebih panjang.

#### Daftar Referensi

- [1] A. Ummung, R. Roswiyanti, M. A. Asgar, and A. N. A. Massiseng, "Analisis Pemasaran Ikan Hias Sebelum dan Selama Pandemi Covid-19 di Kelurahan Balang Baru Kota Makassar," *Akuatikisle: Jurnal Akuakultur, Pesisir dan Pulau-Pulau Kecil*, vol. 6, no. 1, pp. 47–50, 2022, doi: 10.29239/j.akuatikisle.6.1.47-50.
- [2] M. Anjasromo, M. Hadi, M. Maskan, I. L. Putra, and N. Maulana, "Pengaruh Harga dan Kualitas Produk terhadap Keputusan Pembelian pada Ikan Hias Predator di Aquatic Predator Malang," *Jurnal Media Akademik*, vol. 2, no. 10, pp. 1-12, 2024, doi: 10.62281/v2i10.793.
- [3] R. S. Handayani, G. A. Diva, E. N. Hermansyah, and A. Siswati, "Transformasi Digital E-commerce di Era Pasca-Pandemi: Analisis Tren dan Dampak pada Perilaku Konsumen Indonesia," *IKN: Jurnal Informatika dan Kesehatan*, vol. 2, no. 2, pp. 45-58, 2025, doi: 10.35473/ikn.v2i2.3704.
- [4] S. R. Pratamansyah, "Transformasi Digital dan Pertumbuhan UMKM: Analisis Dampak Teknologi pada Kinerja Usaha Kecil dan Menengah di Indonesia," *Jurnal Akuntansi Manajemen dan Perencanaan Kebijakan*, vol. 2, no. 2, pp. 1-17, 2024, doi: 10.47134/jampk.v2i2.475.
- [5] A. N. Solikhati, D. Febriyanti, D. Fibrianto, S. Amaleo, and S. D. Hawa, "Penerapan E-commerce dan Pemasaran Digital untuk Meningkatkan Daya Saing Usaha UMKM WINSHOP," *JAMU: Jurnal Abdi Masyarakat UMUS*, vol. 4, no. 1, pp. 61-66, 2023, doi: 10.46772/jamu.v4i01.1215.
- [6] M. A. Latief, Y. Irawan, and F. Nugraha, "Implementasi *Platform* Penjualan Berbasis Web dengan Fitur *Payment gateway* pada Shopfish Aquarium Kudus," *Jurnal Teknologi Informasi dan Komputer*, vol. 10, no. 1, pp. 222–233, 2025.
- [7] F. Sinlae, I. Maulana, F. Setiyansyah, and M. Ihsan, "Pengenalan Pemrograman Web: Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL," *Jurnal Siber Multi Disiplin*, vol. 2, no. 2, pp. 68–82, 2024, doi: 10.38035/jsmd.v2i2.156.
- [8] R. I. Borman, A. Harjoko, and Wahyono, "Improved ORB Algorithm Through Feature Point Optimization and Gaussian Pyramid," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 2, pp. 268-275, 2024, doi: 10.14569/IJACSA.2024.0150228.
- [9] A. Prasetyo and D. Kriestanto, "Analisis Perbandingan GraphQL dan REST API pada Aplikasi Menu Restoran dengan Node.js," *Jurnal Informatika dan Komputer*, vol. 9, no. 2, pp. 274-284, 2025.
- [10] G. K. Wardana, B. Rahayudi, and W. H. N. Putra, "Pengembangan E-commerce dengan Integrasi API *Payment gateway* Midtrans," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 11, pp. 4770–4774, 2021.
- [11] R. M. Ulah and S. Suartana, "Implementasi *Session management* pada Website Magang Menggunakan Teknologi MERN," *Jurnal Informatika dan Ilmu Komputer (JINACS)*, vol. 6, no. 3, pp. 765–777, 2025, doi: 10.26740/jinacs.v6n03.p765-777.
- [12] D. Syifasultana, D. Muriyatmoko, and E. P. Widhi, "Implementasi Sistem Pengingat pada Kumpulan Do'a Berbasis Mobile Menggunakan Metode Push Notifikasi," in *Seminar Nasional Penelitian dan Pengabdian Kepada Masyarakat CORISINDO*, 2024, pp. 430-435.
- [13] A. K. Wijaya, "Analisis Struktur Basis Data pada Aplikasi E-commerce Tokopedia: Studi Kualitatif terhadap Desain dan Optimalisasi Skema Relasional," *Jurnal Penelitian dan Pengkajian Sains dan Teknologi*, vol. 35, no. 2, pp. 50-57, 2025, doi: 10.37277/stch.v35i2.2346.
- [14] M. S. Harlina, B. Susilowati, S. Suharni, M. S. Herawati, and M. F. Atsilah, "Pemodelan Sistem Rancangan Website Toko Ummi Cookies Menggunakan UML (Unified Modeling

- Language)," *Jurnal Teknologi dan Sistem Informasi (JTEKSIS)*, vol. 7, no. 3, pp. 364-371, 2025, doi: 10.47233/jteksis.v7i3.1943.
- [15] G. D. Pamungkas, Y. Parwati, and B. D. Putranto, "Pengembangan Aplikasi Pendaftaran Siswa Baru Berbasis Web dengan React.Js dan Tailwind CSS," *Jurnal Algoritma*, vol. 22, no. 1, pp. 37-48, 2025, doi: 10.33364/algoritma/v.22-1.2135.
- [16] U. Saputra, N. Astrianda, B. R. Nasution, A. A. Anggara, R. S. Qaisa, and A. E. Jakfar, "Analisa Pengujian Sistem Informasi Website E-commerce Bali-Store Menggunakan Metode Black Box Testing," *Jurnal Teknologi Informasi*, vol. 2, no. 2, pp. 95-102, 2023.
- [17] M. Nurlies, A. Budiyantera, and I. Lewenusa, "Penerapan Metode Agile Scrum pada Pembuatan Website Penjualan Sembako Toko Erwin," *JUTISI: Jurnal Teknik Informatika dan Sistem Informasi*, vol. 13, no. 1, pp. 671-681, 2025, doi: 10.35889/jutisi.v13i1.1905.
- [18] R. Rauf, A. Syam, and M. F. Randy, "Optimalisasi Transformasi Digital dalam Mendorong Pertumbuhan Usaha Mikro, Kecil dan Menengah di Indonesia," *BJRM: Jurnal Bongaya Penelitian Manajemen*, vol. 7, no. 1, pp. 95-102, 2024, doi: 10.37888/bjrm.v7i1.594.