

IMPLEMENTASI SISTEM INFORMASI PENJUALAN BERBASIS WEB DENGAN INTEGRASI API UNTUK OPTIMALISASI PROSES TRANSAKSI PADA MICROSTADIO

IMPLEMENTATION OF WEB-BASED SALES INFORMATION SYSTEM WITH API INTEGRATION FOR TRANSACTION PROCESS OPTIMIZATION AT MICROSTADIO

Carissa Puteri¹, Jap Tji Beng^{1*}, Novario Jaya Perdana¹, Nathasya Kristianti Ferdiana¹

Fakultas Teknologi Informasi, Program Studi Sarjana Sistem Informasi, Universitas

Tarumanagara, Jakarta¹

t.jap@untar.id^{1*}

ABSTRACT

MicroStadio is a small enterprise specializing in the sale of miniature football stadiums. Previously, its transaction process was conducted manually through social media, often resulting in data duplication, input errors, and reporting delays. This study aims to implement a web-based sales information system integrated with an Application Programming Interface (API) to optimize transaction processes at MicroStadio. The system was developed using the Laravel framework and MySQL database, integrating product, cart, order, and payment modules within a Model-View-Controller (MVC) architecture. The API acts as a bridge between the interface and server, enabling faster data exchange and consistent transactions. Implementation results show improved efficiency, fewer errors, and faster report generation. A fallback service mechanism ensures stability during API downtime. Overall, the system enhances automation and supports the digital transformation of small-scale enterprises like MicroStadio.

Keywords: Sales Information System, Web Application, MVC, API Integration, Small Enterprise

ABSTRAK

MicroStadio merupakan usaha kecil yang bergerak di bidang penjualan miniatur stadion sepak bola. Sebelumnya, proses transaksi dilakukan secara manual melalui media sosial, sehingga sering terjadi duplikasi data, kesalahan input, dan keterlambatan laporan. Penelitian ini bertujuan mengimplementasikan sistem informasi penjualan berbasis web dengan integrasi *Application Programming Interface* (API) untuk mengoptimalkan proses transaksi pada MicroStadio. Sistem dikembangkan menggunakan framework Laravel dan basis data MySQL, serta mengintegrasikan modul produk, keranjang, pesanan, dan pembayaran dalam arsitektur *Model-View-Controller* (MVC). API berfungsi sebagai jembatan komunikasi antara antarmuka pengguna dan server untuk mempercepat pengolahan data, menjaga konsistensi transaksi, serta meningkatkan skalabilitas sistem. Hasil implementasi menunjukkan peningkatan efisiensi, penurunan kesalahan input, dan percepatan pembuatan laporan. Mekanisme *fallback service* diterapkan untuk menjaga stabilitas sistem saat API eksternal tidak tersedia. Secara keseluruhan, sistem ini meningkatkan otomatisasi dan mendukung transformasi digital pada usaha kecil seperti MicroStadio.

Kata Kunci: Sistem Informasi Penjualan, Aplikasi Web, MVC, Integrasi API, Usaha Kecil

PENDAHULUAN

Perkembangan teknologi informasi telah membawa perubahan besar dalam berbagai sektor, termasuk perdagangan dan manajemen bisnis. Digitalisasi sistem informasi menjadi salah satu faktor utama dalam meningkatkan efisiensi operasional dan daya saing organisasi di era industri 4.0. Penerapan sistem informasi berbasis web memungkinkan integrasi data yang lebih cepat, akurat, dan terpusat sehingga mampu meminimalkan kesalahan manusia serta mempercepat proses pengambilan

keputusan (Gurianova et al., 2023). Selain itu, optimalisasi sistem informasi berperan penting dalam menekan biaya transaksi dan meningkatkan performa manajerial melalui pemanfaatan basis data yang terstruktur dengan baik (Hartono & Masyhur, 2023). Penelitian lain juga menunjukkan bahwa integrasi modul dan otomatisasi proses melalui sistem berbasis web dapat meningkatkan efisiensi kerja serta akurasi pengelolaan data (Perdana et al., 2021). Dengan demikian, transformasi digital tidak lagi sekadar tren, tetapi merupakan

kebutuhan strategis bagi organisasi modern, termasuk pelaku usaha skala kecil dan menengah.

Dalam konteks nasional, digitalisasi sistem informasi juga telah diadopsi secara luas oleh pelaku Usaha Mikro, Kecil, dan Menengah (UMKM) untuk meningkatkan efisiensi dan transparansi proses bisnis. Santoso dan Dewi (2022) menegaskan bahwa penerapan sistem informasi berbasis web dapat membantu UMKM mengelola transaksi penjualan, pembelian, dan stok secara lebih teratur, menggantikan proses manual yang rentan terhadap kesalahan pencatatan. Hal serupa juga disampaikan oleh Suwita et al. (2021), bahwa sistem penjualan berbasis web mampu mempermudah pelanggan dalam melakukan transaksi sekaligus memperluas jangkauan pemasaran tanpa batas geografis. Namun, sebagian besar penelitian sebelumnya masih berfokus pada digitalisasi usaha kuliner dan perdagangan umum, sedangkan penerapan sistem informasi pada sektor usaha kreatif seperti penjualan miniatur stadion belum banyak dikaji.

MicroStadio merupakan salah satu UMKM yang bergerak di bidang penjualan miniatur stadion klub sepak bola. Sebelum penerapan sistem digital, proses transaksi pada MicroStadio dilakukan secara manual melalui media sosial dan pencatatan terpisah. Kondisi ini menimbulkan berbagai kendala seperti duplikasi data, kesalahan input pesanan, serta keterlambatan dalam pembuatan laporan penjualan. Situasi tersebut menunjukkan perlunya penerapan sistem informasi penjualan berbasis web yang mampu mengintegrasikan seluruh proses bisnis agar lebih efisien dan terstruktur. Selain itu, integrasi *Application Programming Interface (API)* menjadi komponen penting dalam arsitektur sistem modern karena berfungsi sebagai jembatan komunikasi antara antarmuka pengguna dengan server secara efisien dan terstandarisasi. Menurut Jayaprakash (2025), penggunaan **RESTful API** memungkinkan sistem beroperasi

secara *stateless* dan ringan, sehingga mempercepat pertukaran data antar modul. Dalam penelitian ini, integrasi **API diterapkan pada modul transaksi, khususnya untuk perhitungan ongkos kirim (*shipping cost*)** berdasarkan lokasi pelanggan. Penerapan ini memungkinkan sistem menghitung total pembayaran secara otomatis dan akurat pada tahap *checkout*, sehingga mengurangi potensi kesalahan input serta mempercepat proses transaksi. Dengan integrasi API ongkir tersebut, sistem informasi penjualan tidak hanya menjadi lebih efisien dan konsisten dalam pengolahan data, tetapi juga mampu meningkatkan pengalaman pengguna melalui proses pemesanan yang lebih praktis dan terotomatisasi.

Berdasarkan permasalahan tersebut, penelitian ini dilakukan untuk **mengimplementasikan sistem informasi penjualan berbasis web pada MicroStadio** dengan fokus pada empat aspek utama: (1) perbandingan proses transaksi antara sistem manual dan sistem otomatis, (2) integrasi modul produk, keranjang, pesanan, dan pembayaran, (3) penerapan arsitektur sistem menggunakan pendekatan *Model-View-Controller (MVC)*, serta (4) analisis dampak sistem terhadap efisiensi dan pengambilan keputusan manajerial. Berbeda dengan penelitian sebelumnya yang berfokus pada sektor UMKM umum atau restoran (A. B. Santoso & Dewi, 2022; Suwita et al., 2021), penelitian ini menyoroti penerapan sistem informasi digital pada sektor usaha kreatif berbasis koleksi, yaitu penjualan miniatur stadion.

Melalui implementasi sistem informasi ini, diharapkan MicroStadio dapat meningkatkan efisiensi operasional, mengurangi kesalahan pencatatan transaksi, serta mempercepat proses pelaporan penjualan. Secara lebih luas, penelitian ini juga berkontribusi dalam memperkuat literatur mengenai digitalisasi UMKM dan menunjukkan bagaimana sistem informasi berbasis web dapat

diterapkan secara efektif pada usaha kecil yang bergerak di bidang kreatif dan koleksi.

METODE

Metode penelitian ini meliputi dua aspek utama, yaitu metode pengumpulan data dan metode pengembangan perangkat lunak. Pengembangan sistem dilakukan menggunakan metode Waterfall, yang terdiri atas tahapan analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan secara berurutan (Senarath, 2021). Arsitektur sistem dirancang dengan pola *Model-View-Controller* (MVC), yang memisahkan antara logika data (Model), antarmuka pengguna (View), dan pengendali alur aplikasi (Controller), sehingga sistem menjadi lebih modular, efisien, dan mudah dipelihara (Rahman et al., 2024). Selain itu, sistem ini juga mengadopsi pendekatan API-led integration untuk mendukung komunikasi antara frontend dan backend melalui RESTful API, sehingga pertukaran data antar modul menjadi lebih cepat, terstandarisasi, dan aman (Basavapura Jayaprakash, 2021).

PENGUMPULAN DATA

Pengumpulan data dilakukan untuk memperoleh informasi yang dibutuhkan dalam menganalisis kebutuhan sistem serta memastikan bahwa sistem yang dikembangkan sesuai dengan kondisi operasional MicroStadio. Metode yang digunakan meliputi observasi dan wawancara.

Metode observasi dilakukan dengan mengamati secara langsung proses transaksi penjualan yang masih dilakukan secara manual melalui media sosial. Pengamatan difokuskan pada alur pemesanan, pencatatan pesanan, pengelolaan stok, dan pembuatan laporan transaksi. Dari hasil observasi ditemukan beberapa permasalahan, seperti duplikasi data, kesalahan input pesanan, serta keterlambatan dalam pembuatan laporan penjualan.

Selain itu, dilakukan wawancara dengan pemilik MicroStadio untuk memperoleh informasi rinci mengenai kebutuhan sistem. Wawancara ini mencakup pembahasan tentang kendala dalam proses bisnis manual, fitur yang dibutuhkan dalam sistem baru, serta harapan terhadap peningkatan efisiensi dan kemudahan dalam pengelolaan transaksi. Berdasarkan hasil wawancara, pihak MicroStadio membutuhkan sistem informasi penjualan berbasis web yang mampu mengintegrasikan data produk, pesanan, dan pembayaran secara otomatis serta menghasilkan laporan transaksi secara real-time. Informasi yang diperoleh dari observasi dan wawancara ini menjadi dasar dalam tahap analisis kebutuhan dan perancangan sistem.

PENGEMBANGAN SOFTWARE

Pengembangan perangkat lunak pada penelitian ini menggunakan metode **Waterfall**, yang bersifat terstruktur dan sistematis dengan tahapan yang dilakukan secara berurutan. Model Waterfall dipilih karena sesuai untuk pengembangan sistem dengan kebutuhan yang telah terdefinisi dengan jelas berdasarkan hasil observasi dan wawancara. Menurut (C. Santoso et al., 2024), metode Waterfall terdiri atas lima tahapan utama, yaitu **analisis kebutuhan, perancangan, implementasi, pengujian, dan pemeliharaan**, di mana setiap tahap harus diselesaikan sebelum berlanjut ke tahap berikutnya.

1. Analisis Kebutuhan

Tahap ini dilakukan untuk mengidentifikasi kebutuhan sistem berdasarkan data hasil observasi dan wawancara dengan pihak MicroStadio. Diperoleh kebutuhan utama berupa sistem yang mampu mengelola data produk, keranjang, pesanan, dan pembayaran secara terintegrasi, serta menghasilkan laporan transaksi otomatis.

2. Perancangan (*Design*)

Pada tahap ini dilakukan perancangan arsitektur sistem menggunakan pola MVC

yang diterapkan dalam framework Laravel. Selain itu, dirancang pula struktur RESTful API untuk menghubungkan antar modul secara stateless menggunakan format data JSON (Basavapura Jayaprakash, 2021). Pendekatan ini mempermudah pertukaran data antara frontend dan backend serta memungkinkan sistem lebih fleksibel dan mudah dikembangkan.

3. Implementasi (*Implementation*)

Implementasi dilakukan dengan mengembangkan modul produk, keranjang, pesanan, dan pembayaran yang saling terhubung melalui API endpoint. Bahasa pemrograman PHP digunakan dengan framework Laravel, serta integrasi basis data MySQL. Setiap *endpoint* diuji untuk memastikan bahwa data yang dikirim dan diterima sesuai dengan format JSON dan berjalan secara *real-time*.

4. Pengujian (*Testing*)

Pengujian sistem dilakukan menggunakan metode blackbox testing untuk memastikan setiap fungsi berjalan sesuai kebutuhan. Selain itu, dilakukan pengujian API menggunakan Postman guna memverifikasi bahwa seluruh *endpoint* berfungsi dengan benar dan memiliki waktu respon yang efisien.

5. Pemeliharaan (*Maintenance*)

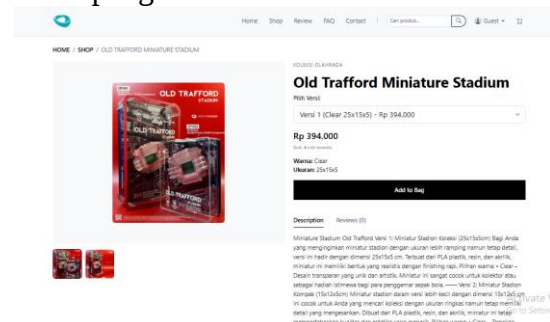
Tahap akhir adalah pemeliharaan sistem, yang meliputi perbaikan kesalahan minor, pembaruan fungsi API jika terjadi perubahan kebutuhan pengguna, serta peningkatan keamanan sistem melalui pembaruan token otentikasi dan pengaturan hak akses.

Melalui tahapan tersebut, sistem informasi penjualan berbasis web dengan integrasi API berhasil dikembangkan secara terstruktur dan memenuhi kebutuhan pengguna sesuai hasil analisis awal.

HASIL DAN PEMBAHASAN

Sistem informasi penjualan yang dikembangkan untuk MicroStadio berbasis web menggunakan framework Laravel dan

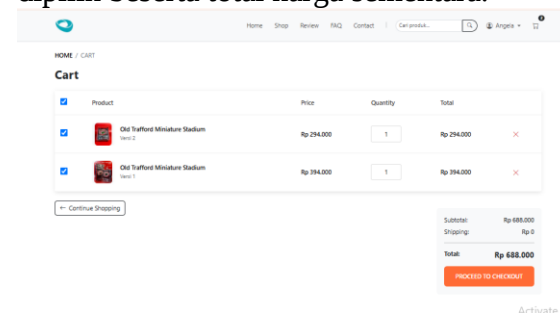
basis data MySQL. Sistem ini dirancang dengan arsitektur *Model-View-Controller* (MVC) sehingga seluruh alur data dari antarmuka pengguna ke basis data dapat berjalan secara efisien dan terstruktur. Antarmuka pengguna dirancang sederhana dan responsif untuk memudahkan pelanggan dalam melakukan pemesanan produk miniatur stadion secara daring. Modul utama yang dibangun dalam sistem meliputi produk, keranjang (*cart*), pesanan (*order*), pembayaran (*payment*), serta API perhitungan ongkir yang terintegrasi pada tahap *checkout*. Melalui modul tersebut, pengguna dapat menelusuri daftar produk, menambahkan item ke keranjang, melakukan pemesanan, dan menghitung total pembayaran otomatis berdasarkan lokasi pengiriman.



Gambar 1. Halaman Detail Produk

Sumber: Dokumentasi pribadi

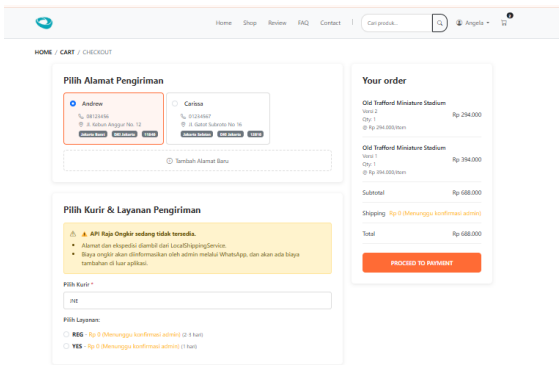
Gambar 1 menunjukkan tampilan halaman utama sistem yang menampilkan daftar produk miniatur stadion. **Gambar 2** memperlihatkan halaman keranjang belanja yang menampilkan daftar produk yang dipilih beserta total harga sementara.



Gambar 2. Halaman Keranjang

Sumber: Dokumentasi pribadi

Sedangkan **Gambar 3** merupakan tampilan form *checkout* di mana sistem melakukan perhitungan otomatis terhadap ongkos kirim berdasarkan lokasi pelanggan menggunakan API.



Gambar 3. Halaman Checkout

Sumber: Dokumentasi pribadi

Integrasi *Application Programming Interface* (API) pada sistem dilakukan menggunakan layanan RajaOngkirService, sebuah kelas khusus di framework Laravel yang bertanggung jawab terhadap komunikasi antara sistem MicroStadio dan API RajaOngkir. Layanan ini berfungsi untuk mengambil data provinsi, kota, kecamatan, serta menghitung ongkos kirim secara otomatis berdasarkan lokasi tujuan pelanggan. Proses komunikasi dilakukan melalui metode HTTP GET dan POST menggunakan format JSON, dengan pengiriman parameter seperti *origin*, *destination*, *weight*, dan *courier*.

Fungsi utama yang diimplementasikan dalam layanan integrasi ongkir dijabarkan pada **Tabel 1**. Fungsi-fungsi tersebut (mis. *get Provinces()*, *getCities()*, *getSubdistricts()*, *getCost()*, *getLocalFallbackCost()*, *findCityByName()*, *getPostalCodes()*) bertugas mulai dari penyediaan daftar lokasi hingga perhitungan biaya pengiriman dan mekanisme fallback saat API eksternal tidak tersedia. Penjelasan singkat per fungsi beserta urgensinya disajikan untuk memberikan gambaran fungsional sistem dan alasan mengapa tiap fungsi diperlukan dalam alur *checkout* berbasis API. Bukti implementasi yang mendukung (cuplikan kode representatif, hasil uji *endpoint* dengan Postman, screenshot halaman *checkout* yang menampilkan hasil ongkir, log error dan fallback, serta query bukti penyimpanan data pesanan beserta ongkir) disertakan setelah tabel ini sebagai Gambar/Tabel pendukung dan disimpan di

lampiran apabila diperlukan untuk verifikasi teknis.

Tabel 1. Fungsi Layanan API Ongkir

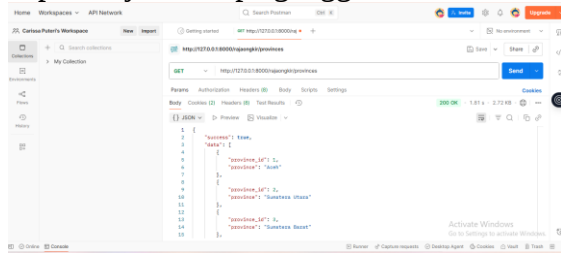
Fungsi (Metode)	Penjelasan singkat	Mengapa penting
<i>getProvinces()</i>	Mengambil daftar provinsi dari API RajaOngkir untuk dropdown alamat.	Memudahkan input alamat, mencegah kesalahan penulisan lokasi.
<i>getCities(\$provinceId)</i>	Mengambil daftar kota berdasarkan provinsi terpilih.	Menjamin konsistensi data lokasi dan validasi pengiriman.
<i>getSubdistricts(\$cityId)</i>	Mengambil kecamatan untuk detail alamat.	Membantu akurasi perhitungan ongkir & pengiriman terakhir.
<i>getCost(\$origin,\$destination,\$weight,\$courier)</i>	Mengirim request cost ke API dan mengembalikan opsi tarif & estimasi pengiriman.	Fungsi inti perhitungan ongkir; menentukan total bayar pada <i>checkout</i> .
<i>getLocalFallbackCost(...)</i>	Menghasilkan data biaya alternatif (set biaya = 0 + pesan informasi) jika API utama gagal.	Menjamin kelangsungan proses <i>checkout</i> (fault tolerance) dan komunikasi ke admin.
<i>findCityByName(\$name)</i>	Pencarian kota fleksibel (normalisasi & partial match).	Mempermudah input pengguna (toleransi variasi nama kota).
<i>getPostalCodes(\$cityName)</i>	Mendapatkan kode pos dari nama kota atau fallback lokal.	Berguna untuk cetak label kirim dan verifikasi data alamat.

Sumber: Dokumentasi pribadi

Pengujian integrasi API dilakukan untuk memastikan sistem mampu berkomunikasi dengan layanan pihak ketiga, dalam hal ini RajaOngkir API, yang digunakan untuk memperoleh data alamat dan perhitungan biaya pengiriman. Pengujian dilakukan menggunakan Postman dengan mengakses *endpoint* <http://127.0.0.1:8000/rajaongkir/provinces> menggunakan metode GET. Hasil pengujian menunjukkan bahwa permintaan berhasil diproses oleh server dan

menghasilkan status 200 OK, yang menandakan bahwa koneksi antara aplikasi dan API berjalan dengan baik. Respons yang diterima berupa data provinsi dari layanan RajaOngkir, yang kemudian digunakan sistem untuk menampilkan pilihan alamat pengguna pada halaman *checkout*.

Keberhasilan pengujian ini membuktikan bahwa sistem telah menerapkan integrasi *Application Programming Interface* (API) secara efektif, di mana proses pertukaran data antara klien dan server berlangsung real-time dan sesuai standar RESTful. Selain itu, hasil pengujian menunjukkan bahwa sistem juga mampu melakukan *fallback* otomatis menggunakan *LocalShippingService* jika layanan RajaOngkir tidak dapat diakses, sehingga memastikan proses transaksi tetap dapat berjalan tanpa gangguan.



Gambar 3. Hasil Pengujian Endpoint API RajaOngkir

Sumber: Dokumentasi pribadi

Meskipun pengujian *endpoint* /rajaongkir/provinces berhasil dengan status 200 OK, pada pengujian *endpoint* lain seperti /rajaongkir/cost ditemukan kendala berupa gagalnya respons dari server API RajaOngkir. Hal ini terjadi karena versi layanan API yang digunakan merupakan versi gratis yang memiliki keterbatasan jumlah permintaan (*rate limit*) serta stabilitas server yang tidak selalu tersedia. Kondisi tersebut menyebabkan sistem tidak menerima data biaya ongkir secara langsung dari server eksternal.

Untuk mengatasi permasalahan ini, sistem dilengkapi dengan mekanisme *fallback* otomatis melalui kelas *LocalShippingService* yang diintegrasikan ke dalam *service layer* (RajaOngkirService). Apabila server API utama tidak merespons, sistem akan

memanggil metode *getLocalFallbackCost()* yang secara otomatis menetapkan nilai ongkir sementara sebesar Rp 0 dan menampilkan pesan kepada pengguna bahwa biaya pengiriman akan dikonfirmasi oleh admin melalui WhatsApp seperti yang ditunjukkan pada **Gambar 4**. Mekanisme ini menunjukkan bahwa sistem tetap dapat beroperasi meskipun API eksternal tidak tersedia, tanpa menghentikan proses *checkout* pengguna. Implementasi *error handling* ini memperlihatkan prinsip resiliensi sistem dan mendukung konsep *fault tolerance*, di mana sistem mampu menanggulangi kegagalan koneksi dengan tetap menjaga kontinuitas layanan.

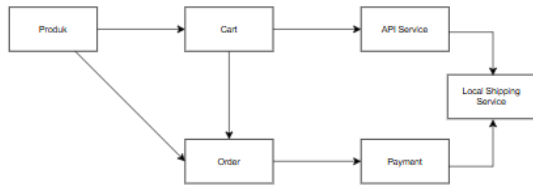


Gambar 4. Tampilan Info Fallback

Sumber: Dokumentasi pribadi

Sistem informasi penjualan berbasis web pada MicroStudio terdiri atas empat modul utama, yaitu **produk**, **keranjang (cart)**, **pesanan (order)**, dan **pembayaran (payment)** yang saling terintegrasi dalam satu alur transaksi. Pengguna memilih produk dari modul *produk*, menambahkannya ke *cart*, lalu sistem meneruskan data tersebut ke *order* saat proses *checkout*. Pada tahap ini, sistem memanfaatkan **API RajaOngkir** untuk mengambil data alamat dan menghitung biaya pengiriman secara otomatis. Jika API tidak merespons, sistem menjalankan mekanisme **fallback lokal (LocalShippingService)** agar proses transaksi tetap berjalan. Setelah pesanan dibuat, modul *payment* mencatat bukti dan status pembayaran yang dapat diverifikasi melalui *dashboard admin*. Hubungan antar modul serta keterkaitannya dengan API RajaOngkir ditunjukkan pada **Gambar 6**, yang menggambarkan alur integrasi data dari proses pemilihan produk hingga penyelesaian transaksi pembayaran. Integrasi antar modul ini memungkinkan

sinkronisasi data secara *real-time*, mempercepat proses transaksi, serta meningkatkan efisiensi operasional.



Gambar 5. Diagram Arsitektur Sistem

Sumber: Dokumentasi pribadi

SIMPULAN

Penelitian ini berhasil mengimplementasikan sistem informasi penjualan berbasis web pada MicroStudio untuk mengoptimalkan proses transaksi yang sebelumnya dilakukan secara manual. Sistem dikembangkan menggunakan framework Laravel dengan arsitektur *Model-View-Controller* (MVC), sehingga alur data antara antarmuka pengguna, logika bisnis, dan basis data berjalan secara efisien dan terstruktur. Integrasi *Application Programming Interface* (API) pada sistem ini, khususnya dalam layanan ongkos kirim berbasis lokasi, berperan penting dalam mendukung otomatisasi proses dan sinkronisasi data antar modul.

Hasil implementasi menunjukkan bahwa pemanfaatan API mampu meningkatkan efisiensi komunikasi antar sistem dan menjaga konsistensi data transaksi, meskipun masih ditemukan keterbatasan pada kestabilan layanan API eksternal seperti RajaOngkir. Untuk mengatasi hal tersebut, mekanisme error handling dan *fallback service* dikembangkan agar sistem tetap berfungsi optimal. Temuan ini sejalan dengan (Bandaru, 2025) yang menegaskan bahwa API merupakan inti dari integrasi perangkat lunak modern karena mempercepat pertukaran data dan mendukung fleksibilitas pengembangan sistem. Lebih lanjut, penelitian oleh (Efuntade & Efuntade, 2023) juga menunjukkan bahwa penerapan API dalam sistem informasi berbasis web memungkinkan interaksi antar layanan secara aman dan terstandarisasi, serta mendukung

pengelolaan data transaksi dan pelaporan secara *real-time* dalam konteks sistem informasi akuntansi (AIS).

Dengan demikian, penerapan sistem informasi penjualan berbasis web dengan integrasi API pada MicroStudio juga memberikan implikasi bisnis yang signifikan bagi sektor UMKM kreatif lainnya. Sistem ini menunjukkan bahwa digitalisasi proses transaksi tidak hanya meningkatkan efisiensi operasional, tetapi juga memperluas potensi pasar melalui kemudahan akses pelanggan secara daring. Bagi pelaku usaha kecil yang bergerak di bidang koleksi atau produk kreatif, penerapan sistem serupa dapat mengurangi ketergantungan pada proses manual yang rawan kesalahan dan memperkuat kepercayaan pelanggan melalui transparansi data pesanan serta pelacakan pembayaran yang *real-time*. Sehingga, penelitian ini tidak hanya memberikan manfaat teknis bagi MicroStudio, tetapi juga menawarkan model transformasi digital yang dapat direplikasi oleh UMKM kreatif lainnya untuk meningkatkan daya saing di era ekonomi digital.

DAFTAR PUSTAKA

- Bandaru, S. P. (2025). The Role of APIs in Modern Web Development: Enhancing System Integrations. *International Journal of Computer Science and Mobile Computing*, 14(3), 11–19. <https://doi.org/10.47760/ijcsmc.2025.v14i03.002>
- Basavapura Jayaprakash, D. (2021). *Sarcouncil Journal of Multidisciplinary under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 (CC BY-NC-ND 4.0) International License Understanding API Integration in Modern Enterprise Applications*. <https://doi.org/10.5281/zenodo.16149567>
- Efuntade, O. O., & Efuntade, A. O. (2023). *Application Programming Interface*

- (API) And Management of Web-Based Accounting Information System (AIS): Security of Transaction Processing System, General Ledger and Financial Reporting System. *Journal of Accounting and Financial Management*, 9(6), 1–18. <https://doi.org/10.56201/jafm.v9.no6.2023.pg1.18>
- Gurianova, E., Gurianov, A., Mechtcheriakova, S., & Khairullina, A. (2023). Reduction of transaction costs in the process of business informatization. *E3S Web of Conferences*, 458. <https://doi.org/10.1051/e3sconf/202345805002>
- Hartono, N., & Masyhur, Z. (2023). Optimizing transaction data performance in database management systems. *MATRIX: Jurnal Manajemen Teknologi Dan Informatika*, 13(2), 106–114. <https://doi.org/10.31940/matrix.v13i2.106-114>
- Perdana, N. J., Chandra, D., & Amanto, A. F. (2021). *Seminar Nasional Hasil Penelitian dan Pengabdian Kepada Masyarakat 2021 Pengembangan Ekonomi Bangsa Melalui Inovasi Digital Hasil Penelitian dan Pengabdian Kepada Masyarakat Jakarta*.
- Rahman, M. H., Naderuzzaman, M., Kashem, M. A., Salahuddin, B. M., & Mahmud, M. Z. (2024). Comparative Study: Performance of MVC Frameworks on RDBMS. *International Journal of Information Technology and Computer Science*, 16(1), 26–34. <https://doi.org/10.5815/ijitcs.2024.01.03>
- Santoso, A. B., & Dewi, M. U. (2022). Digitalisasi UMKM untuk Optimalisasi Sistem Informasi dan Integrasi Layanan Aplikasi Website Transaksi Online di Masa Pandemi. *Jurnal Abdidas*, 3(1), 198–205. <https://doi.org/10.31004/abdidas.v3i1.560>
- Santoso, C., Arisandi, D., & Jap, T. B. (2024). Perancangan sistem informasi dalam penjualan pada Toko Furniture Bahagia berbasis web. *Jurnal Ilmu Komputer dan Sistem Informasi (JIKSI)*, 12(1). <https://doi.org/10.24912/jiksi.v12i1.28290>
- Senarath, U. S. (2021). *Waterfall Methodology, Prototyping and Agile Development*. <https://doi.org/10.13140/RG.2.2.17918.72001>
- Suwita, F. S., Sholihat, S. A., & Dewi, N. P. (2021). Web-based information system sales. *Journal of Physics: Conference Series*, 1764(1). <https://doi.org/10.1088/1742-6596/1764/1/012189>