JUSIBI (JURNAL SISTEM INFORMASI DAN E-BISNIS)
***p-ISSN* : 2655-7541** *Volume 8, Nomor , June 2026*
e-ISSN : 2745-5823 *Hal : 55~63*

Optimasi Infrastruktur *Cloud Free Tier* Berbasis *Serverless* pada Platform *E-Commerce* UMKM di Indonesia

Fadel Muhamad Rifai[1*], Vina Ayumi[2]
Universitas Dian Nusantara, Jakarta, Indonesia
411221027@mahasiswa.undira.ac.id[1], vina.ayumi@dosen.undira.ac.id[2]
*Corresponding author: 411221027@mahasiswa.undira.ac.id[2]

Abstrak—*Marketplace* membantu UMKM menjangkau pembeli, tetapi biaya komisi 2,5% sampai 10% per transaksi dapat mengurangi margin usaha kecil. Penelitian ini membangun TumbuhBersama, platform *e-commerce* mandiri berbasis *serverless* yang berjalan pada layanan cloud *free-tier*. Metode penelitian menggunakan *prototyping*, mulai dari identifikasi kebutuhan UMKM, perancangan arsitektur, implementasi bertahap, sampai pengujian fungsional. Sistem dikembangkan menggunakan SvelteKit, Supabase, Vercel, Drizzle ORM, Komerce Payment API, RajaOngkir API (via Komerce), Gmail API, dan Fonnte API. Penelitian diarahkan pada penyediaan fitur toko *online* sekaligus pengendalian katalog, *checkout*, pembayaran, pengiriman, produk digital, lelang, *flash sale*, laporan bisnis, dan *admin dashboard* agar tetap berjalan di bawah batas layanan gratis. Optimasi dilakukan melalui *server-side rendering*, *caching* tarif pengiriman, cache in-memory berbasis *LRU*, *cron job* operasional, pelacakan kuota API, *auto-expire* pembayaran, *retry* email, *keepalive* database, *cleanup log*, *Row Level Security*, *role-based access*, *payment webhook idempotent*, dan *update* stok atomik saat checkout. Hasil implementasi memperlihatkan bahwa platform dapat menggantikan fungsi dasar marketplace untuk satu *merchant* UMKM tanpa biaya *hosting* awal dan tanpa potongan komisi marketplace. Biaya transaksi dari *payment gateway* dan layanan logistik tetap ada sesuai tarif penyedia, tetapi dicatat terpisah sebagai *service fee* dan ongkos kirim.

Abstract—*Marketplaces help MSMEs reach buyers, but commission fees of 2.5% to 10% per transaction can reduce small business margins. This study develops TumbuhBersama, a standalone e-commerce platform based on a serverless architecture and cloud free-tier services. The research uses prototyping, starting from MSME requirement identification, architecture design, staged implementation, and functional testing. The system uses SvelteKit, Supabase, Vercel, Drizzle ORM, Komerce Payment API, RajaOngkir API (via Komerce), Gmail API, and Fonnte API. The research provides online store features while controlling catalog browsing, checkout, payment, shipping, digital products, auctions, flash sales, business reporting, and an admin dashboard so they remain usable within free-tier limits. The optimization uses server-side rendering, shipping rate caching, in-memory LRU caching, operational cron jobs, API quota tracking, payment auto-expiry, email retry, database keepalive, log cleanup, Row Level Security, role-based access, idempotent payment webhooks, and atomic stock updates during checkout. The implementation indicates that the platform can replace basic marketplace functions for a single MSME merchant without initial hosting cost and without marketplace commission deductions. Transaction costs from the payment gateway and logistics provider still apply according to each provider's tariff, but the system records them separately as service fees and shipping costs.*

Keywords—*cloud free-tier, standalone e-commerce, serverless, Supabase, MSMEs*

1. Pendahuluan

Teknologi telah mendukung pegembangan platform bisnis di Indonesia [1], [2], [3], [4], [5], [6]. Aplikasi Tokopedia dan Shopee memberi akses pasar yang luas bagi UMKM, tetapi model ini memiliki biaya komisi. Beberapa marketplace mengenakan potongan sekitar 2,5% sampai 10% per transaksi [7]. Bagi UMKM dengan margin tipis, potongan tersebut langsung mengurangi ruang laba [8]. Di luar marketplace, membangun kanal penjualan sendiri juga tidak sederhana [9]. Pelaku usaha tetap membutuhkan katalog produk, *checkout*, pembayaran, ongkos kirim,

***p-ISSN* : 2655-7541**
e-ISSN : 2745-5823

pelacakan pesanan, laporan penjualan, dan panel admin yang dapat dipakai tanpa biaya operasional yang besar [10], [11], [12], [13], [14], [15], [16].

Beberapa penelitian sudah membahas *marketplace*, *e-commerce*, dan *serverless* dari sisi yang berbeda. Penelitian oleh Saputri et al. [17] meneliti peran marketplace dalam meningkatkan daya saing UMKM melalui studi literatur sistematis. Manurung [18] meneliti akses pemasaran UMKM melalui marketplace dengan fokus pada peningkatan jangkauan pasar. Ardiansyah [19] membangun model *e-commerce* untuk UMKM tanpa membahas optimasi infrastruktur. Rizky dan Zulfikar [20] mengembangkan sistem *e-commerce* dengan optimasi database tetapi tidak menggunakan pendekatan *serverless*. Haryadi [21] meneliti optimalisasi *e-commerce* untuk peningkatan pendapatan UMKM tanpa membahas batasan layanan cloud. Fujiyanti et al. [22] membandingkan performa *server-based* dan *serverless* di Google Cloud Platform (GCP). Namun, penelitian-penelitian tersebut belum membahas pengendalian kuota API logistik dan mekanisme *caching* untuk platform *e-commerce* UMKM yang berjalan pada layanan *free-tier*.

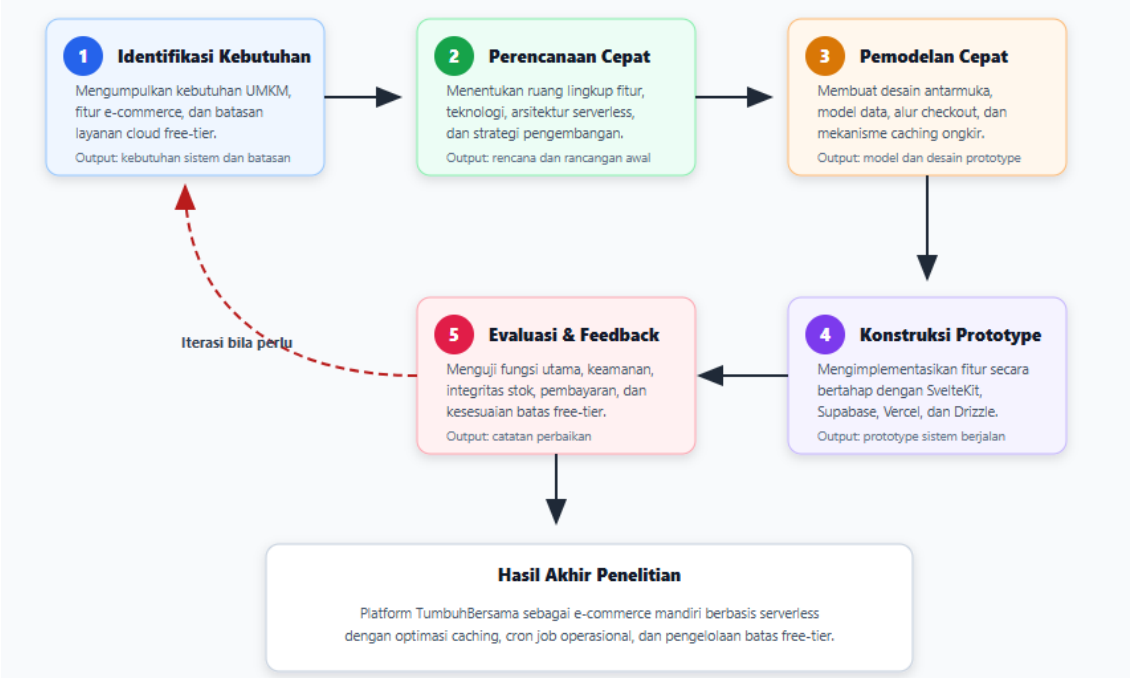
Penelitian ini mengusulkan TumbuhBersama sebagai platform *e-commerce* mandiri untuk satu *merchant* UMKM. Sistem dirancang agar UMKM dapat menjual produk fisik, produk digital, *flash sale*, dan lelang tanpa bergantung pada potongan komisi *marketplace*. Infrastruktur awal memakai layanan cloud *free-tier*, terutama Vercel untuk *hosting serverless* dan Supabase untuk database, autentikasi, storage, dan *realtime*. Selama penggunaan masih berada di batas *free-tier*, biaya *hosting* awal dapat ditekan sampai nol rupiah.

Masalah utamanya adalah batas sumber daya. RajaOngkir API (via Komerce) sebagai layanan ongkos kirim memiliki batas permintaan harian. Supabase *free-tier* dapat berhenti sementara jika tidak ada aktivitas dalam beberapa hari. Vercel *serverless* memiliki batas waktu eksekusi fungsi. Gmail API dan Fonnte API juga memiliki batas pengiriman notifikasi. Karena itu, platform tidak cukup hanya di-deploy ke layanan gratis. Sistem harus mengatur kapan API dipanggil, kapan data disimpan ulang, bagaimana stok divalidasi secara atomik, bagaimana pembayaran kedaluwarsa, dan bagaimana admin mengetahui kondisi layanan. Kontribusi penelitian ini adalah rancangan dan implementasi platform *e-commerce* mandiri berbasis *serverless* yang disesuaikan dengan batas layanan *free-tier*, mencakup *caching*, *cron job* operasional, pelacakan kuota API, keamanan akses, integritas stok, *payment callback idempotent*, dan fitur operasional UMKM.

2. Metodologi Penelitian

Penelitian ini menggunakan metode *prototyping*. Tahap penelitian meliputi identifikasi kebutuhan UMKM dan batasan layanan *free-tier*, perancangan arsitektur *serverless*, perancangan mekanisme sinkronisasi dan *caching*, implementasi iteratif, uji fungsional, serta evaluasi kemampuan sistem dalam menjalankan fitur *e-commerce* mandiri tanpa server dedicated. Metode *prototyping* dipilih karena kebutuhan UMKM dan batasan *free-tier* perlu divalidasi secara bertahap sebelum sistem final dibangun. Diagram alir metodologi penelitian ditunjukkan pada **Gambar 1**.

***p-ISSN* : 2655-7541**
e-ISSN : 2745-5823



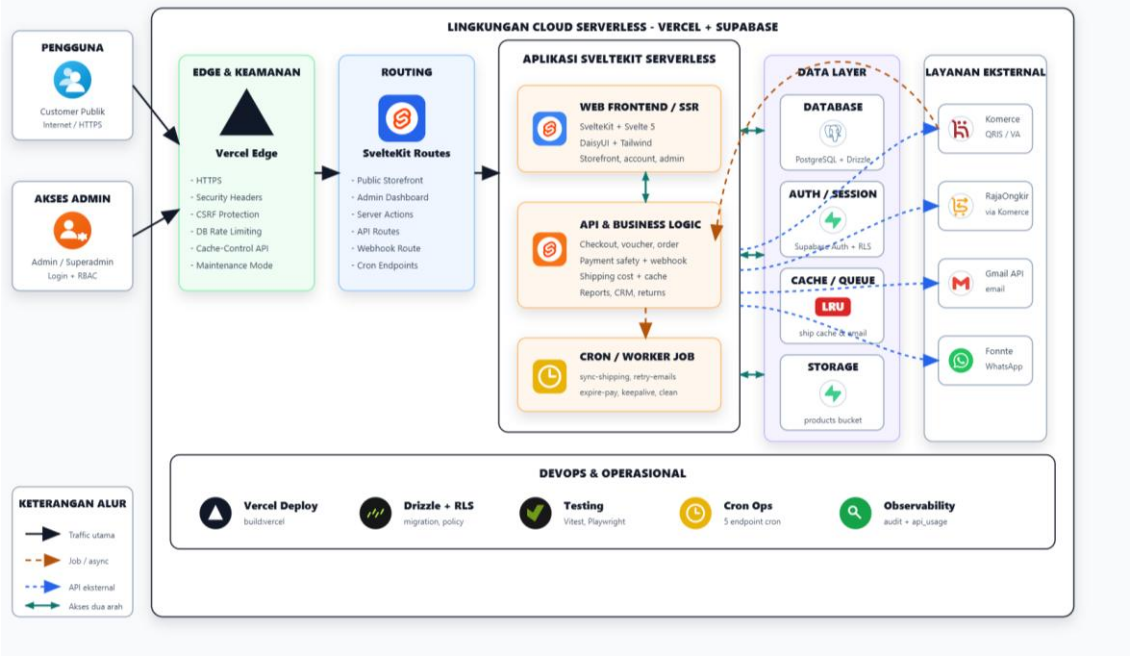
Gambar 1. Diagram Alir Metodologi Penelitian
Sumber: Dokumentasi penelitian, dibuat dan diolah sendiri oleh peneliti

Tahap pertama adalah identifikasi kebutuhan UMKM dan batasan layanan *free-tier* untuk menentukan fitur yang harus tersedia, seperti katalog produk, *checkout*, pembayaran, pengiriman, dan pengelolaan pesanan. Tahap kedua adalah perencanaan cepat, yaitu penentuan ruang lingkup fitur, teknologi, dan rancangan awal arsitektur *serverless*. Tahap ketiga adalah pemodelan cepat, yang mencakup desain antarmuka, model data, alur *checkout*, dan mekanisme *caching* ongkos kirim. Tahap keempat adalah konstruksi *prototype* menggunakan SvelteKit, Supabase, Vercel, dan Drizzle ORM. Tahap kelima adalah evaluasi dan *feedback* melalui pengujian fungsi utama, keamanan akses, integritas stok, pembayaran, dan kesesuaian sistem terhadap batasan *free-tier*. Jika hasil evaluasi belum sesuai, proses kembali ke tahap sebelumnya sampai *prototype* memenuhi kebutuhan penelitian.

Sistem dibangun dengan SvelteKit 2 dan Svelte 5 (Runes) sebagai framework, lalu di-deploy di Vercel menggunakan pola *serverless*. Database PostgreSQL di-host di Supabase *free-tier* dan diakses melalui Drizzle ORM. Autentikasi memakai Supabase *Auth* dengan dukungan Google OAuth dan *login* email. Pembayaran terintegrasi ke Komerce *Payment* API untuk QRIS dan *Virtual Account*. Pengiriman menggunakan RajaOngkir API (via Komerce). Sistem mengirimkan notifikasi melalui dua kanal: email menggunakan Gmail API dengan OAuth2 *refresh token*, dan WhatsApp melalui Fonnte API.

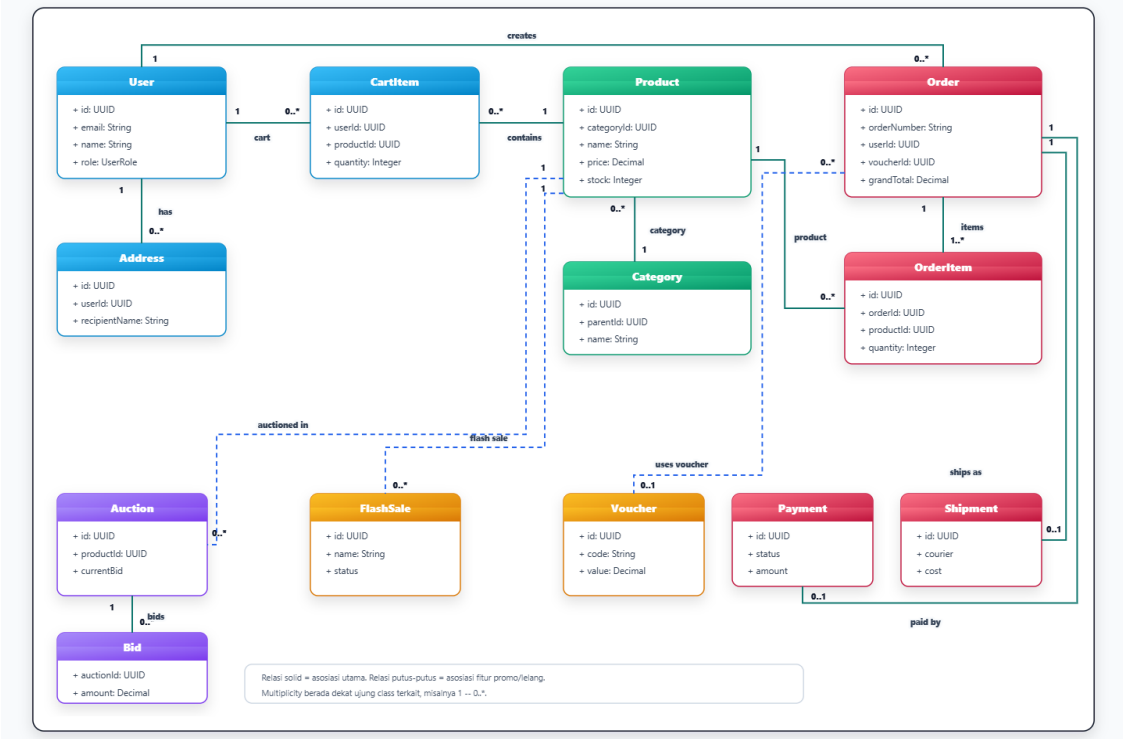
3. Hasil dan Pembahasan

Platform TumbuhBersama terdiri atas empat lapisan. Lapisan pertama adalah SvelteKit untuk routing dan rendering. Lapisan kedua adalah domain *services* yang memproses logika bisnis seperti *checkout*, pembayaran, pengiriman, promosi, notifikasi, dan laporan. Lapisan ketiga adalah *data access layer* dengan Drizzle ORM. Lapisan terakhir adalah database PostgreSQL di Supabase. Susunan ini disesuaikan dengan lingkungan *serverless* dan layanan *free-tier*. Pada implementasi ini, sistem dibuat sebagai *monolith modular* yang berjalan di satu platform *serverless*. Skema arsitektur dapat dilihat pada **Gambar 2**.



Gambar 2. Diagram Arsitektur Sistem TumbuhBersama
Sumber: Dokumentasi penelitian

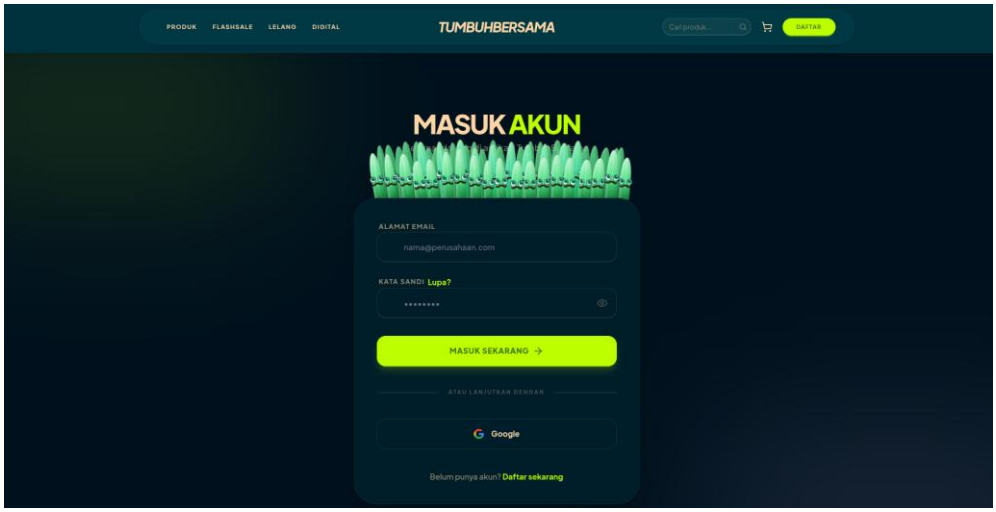
Use case diagram menggambarkan interaksi antara aktor (pengguna) dengan sistem TumbuhBersama. Terdapat tiga aktor utama: *Customer* (pelanggan), Admin (pengelola toko), dan System (proses otomatis). *Class diagram* menunjukkan struktur data utama dalam sistem beserta relasinya. Kelas utama meliputi *Profile* (pengguna), *Product*, *Order*, *Payment*, dan entitas seperti pada **Gambar 3**.



Gambar 3. Class Diagram Platform TumbuhBersama
Sumber: Dokumentasi penelitian

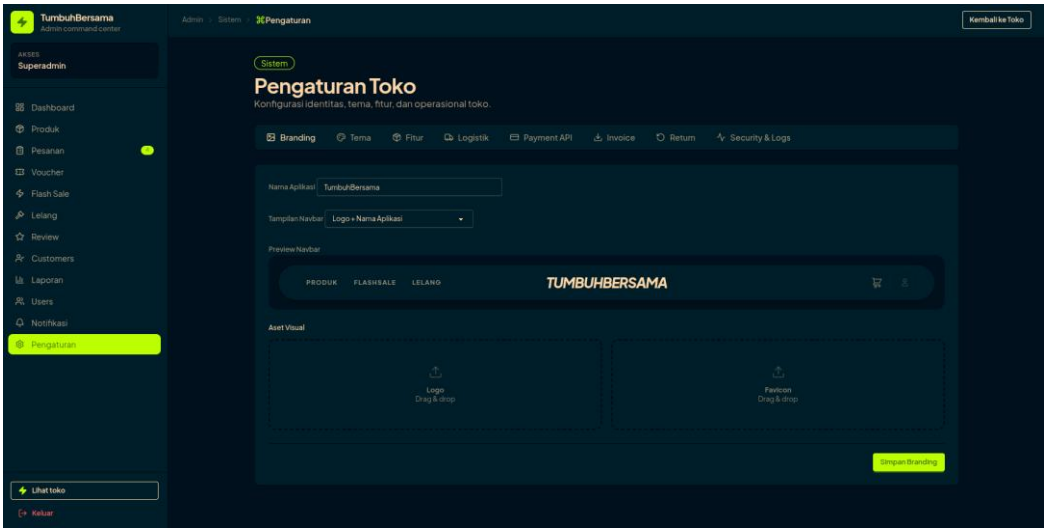
***p-ISSN* : 2655-7541**
e-ISSN : 2745-5823

Customer dan Admin mengakses aplikasi SvelteKit yang berjalan di Vercel. Proses bisnis dijalankan melalui *Server Actions* dan *API Routes*, lalu data disimpan melalui Drizzle ORM ke Supabase PostgreSQL. Layanan eksternal seperti Komerce, RajaOngkir, Gmail, dan Fonnte hanya dipanggil saat diperlukan. Proses berulang seperti sinkronisasi ongkir, *retry* email, *keepalive*, dan *cleanup* dijalankan melalui Vercel *Cron*. Halaman login dapat dilihat pada **Gambar 4**.



Gambar 4. Tampilan Halaman Login
Sumber: Dokumentasi penelitian

Sistem dikembangkan menggunakan SvelteKit 2 dan Svelte 5 sebagai framework utama dengan dukungan Tailwind CSS v4 dan DaisyUI untuk antarmuka pengguna yang responsif. Pengelolaan data dilakukan menggunakan PostgreSQL melalui Supabase dan Drizzle ORM sebagai lapisan akses data, sedangkan proses hosting dan penjadwalan tugas (*cron job*) memanfaatkan layanan Vercel. Fitur pendukung sistem meliputi integrasi Komerce Payment API untuk pembayaran QRIS dan Virtual Account, RajaOngkir API untuk perhitungan ongkos kirim, Fonnte API untuk notifikasi WhatsApp, serta Gmail API untuk notifikasi email. Halaman administrator dapat dilihat pada **Gambar 5**.



Gambar 5. Tampilan Halaman Admin
Sumber: Dokumentasi penelitian

Katalog produk, promosi, laporan bisnis, dan data pelanggan dikelola secara mandiri melalui sistem yang dikembangkan, sementara proses checkout, pembayaran, dan perhitungan ongkos kirim diintegrasikan menggunakan SvelteKit, Komerce Payment API, dan RajaOngkir API. Berbeda dengan marketplace konvensional yang membatasi akses data dan mengenakan komisi platform, TumbuhBersama mendukung UMKM memiliki kontrol penuh terhadap data pelanggan dan strategi bisnis, dengan biaya transaksi yang dapat disesuaikan melalui layanan pembayaran yang digunakan.

Optimasi *free-tier* dicatat dalam tiga mekanisme teknis: data yang sering dibaca disimpan sementara, proses yang dapat dijadwalkan dipindahkan ke *cron job*, dan proses sensitif divalidasi di sisi server. Pada TumbuhBersama, mekanisme tersebut diterapkan melalui *SSR* SvelteKit, *cache* ongkir, *cache in-memory* berbasis *LRU*, pelacakan kuota API, *cron job*, dan *cleanup data* berkala. *SSR* SvelteKit digunakan agar halaman seperti katalog, produk, *checkout*, dan *admin dashboard* tetap direnderi dari sisi server. Data yang sering dibaca, seperti pengaturan toko, produk, kategori, dan ongkos kirim, tidak selalu diambil langsung dari database atau API eksternal. Sistem memakai *cache* dengan TTL pendek agar data tetap segar tetapi tidak membebani *free-tier*.

Sistem menjalankan lima *cron job* yang dikonfigurasi di vercel.json dan berjalan otomatis tanpa server dedicated. Jadwal cron job operasional menggambarkan berbagai proses otomatis yang dijalankan secara terjadwal untuk mendukung stabilitas, efisiensi, dan keandalan sistem TumbuhBersama. *Cron job* sync-shipping dijalankan setiap hari pukul 23.59 WIB untuk melakukan *pre-fetch* dan penyimpanan data ongkos kirim ke dalam cache sehingga proses perhitungan ongkir pada saat pengguna melakukan checkout dapat berlangsung lebih cepat dan mengurangi jumlah permintaan ke API RajaOngkir.

Cron job expire-pending-payments yang berjalan setiap pukul 07.00 WIB bertugas memeriksa transaksi yang masih berstatus menunggu pembayaran dan telah melewati batas waktu yang ditentukan, kemudian membatalkan transaksi tersebut secara otomatis serta mengembalikan stok produk ke inventori agar dapat kembali dijual. Selanjutnya, *cron job* retry-emails dijalankan setiap hari pukul 13.00 WIB untuk mengirim ulang email notifikasi yang sebelumnya gagal terkirim akibat gangguan jaringan atau keterbatasan kuota layanan email. Untuk menjaga database Supabase tetap aktif pada layanan *free-tier*, *cron job* keepalive dijalankan setiap empat hari sekali pada pukul 15.00 WIB dengan mengirimkan permintaan ringan ke database sehingga risiko *idle shutdown* dapat diminimalkan. Selain itu, *cron job* cleanup-logs yang dijalankan setiap hari Minggu pukul 10.00 WIB berfungsi menghapus log lama, data sementara, dan file *orphan* yang tidak lagi digunakan guna menghemat ruang penyimpanan serta menjaga performa sistem tetap optimal.

Cron job sync-shipping menangani *pre-fetch* ongkir untuk rute populer dan menyimpannya di *cache*. Algoritmanya: (1) cek sisa kuota API hari ini dari tabel *api_usage_log*, (2) bentuk kombinasi asal, tujuan, berat 1 kg, dan kurir aktif, (3) cari destinasi yang belum memiliki *cache* fresh dalam TTL 7 hari, (4) ambil sebanyak min(sisa_kuota, jumlah_belum_*cached*, 40) destinasi, (5) untuk tiap destinasi, ambil ongkir melalui helper *cache*, dan (6) berhenti jika API *error* atau kuota habis. Batas 40 destinasi per eksekusi digunakan pada fungsi *serverless*. Saat pembeli meminta ongkir, sistem memeriksa *cache* lebih dulu. Jika data masih berada dalam TTL 7 hari, API RajaOngkir tidak dipanggil lagi.

Helper *cache* yang sama dipakai pada empat titik: *endpoint* pilihan ongkir, validasi *checkout*, sinkronisasi manual admin, dan *cron* sync-shipping. *Cron job expire-pending-payments* berjalan tengah malam untuk membatalkan pembayaran yang kedaluwarsa dan mengembalikan stok. *Cron job retry-emails* mencari email berstatus *pending* yang jumlah percobaannya belum sampai 3. *Cron job keepalive* berjalan setiap 4 hari untuk menjaga database tetap aktif pada lingkungan *free-tier*. *Cron job cleanup-logs* menghapus *audit_logs* dan *api_usage_log* lebih tua dari 90 hari serta file orphan di *storage*. *Caching* diterapkan dalam dua lapisan. Lapisan pertama adalah *cache* database melalui tabel *shipping_rates_cache* yang menyimpan hasil perhitungan ongkir dari

RajaOngkir API dalam format *jsonb*. Data dianggap fresh selama 7 hari sejak *created_at*. Jika *cache* kedaluwarsa, sistem mencoba mengambil data baru; jika API *error* atau kuota habis, sistem tetap mengembalikan data *cache* lama sebagai *fallback*.

Lapisan kedua adalah *cache in-memory* menggunakan struktur *LRU* (Least Recently Used). Karena layanan *free-tier* tidak menyediakan Redis, *cache* ini dibuat sebagai *cache* lokal yang hidup selama *instance serverless* masih aktif. Saat *cold start* terjadi, *cache* akan kosong lagi dan database tetap menjadi sumber data utama. Tabel 9 menunjukkan konfigurasi masing-masing *cache instance*. TTL yang pendek (30 detik untuk sebagian besar *cache*) disengaja karena di lingkungan *serverless* Vercel, *instance* bisa hilang kapan saja saat *cold start*. Tabel *api_usage_log* mencatat berapa kali API dipanggil per hari per layanan. Sebelum memanggil RajaOngkir, sistem membaca pemakaian hari itu dan menghitung sisa *request*. Jika kuota habis, sistem tidak memaksa *request* baru dan memakai *cache* yang masih tersedia sebagai *fallback*.

Implementasi yang diuji mencakup fungsi toko *online* dan mekanisme operasional di lingkungan *serverless free-tier*. Pada validasi implementasi, rancangan awal berupa tabel reservasi stok tidak digunakan lagi. Mekanisme tersebut disederhanakan menjadi *update* stok atomik di dalam transaksi *checkout* dengan syarat *stock >= quantity* saat stok dikurangi. Pengujian dilakukan melalui *unit test* dan pengujian alur aplikasi di *browser*. Vitest dipakai untuk modul *utility* dan *service*, sedangkan pengujian *browser* digunakan untuk memeriksa alur utama seperti autentikasi, *checkout*, dan halaman admin. Pada pengujian lokal, svelte-check menghasilkan 0 *error* dan 0 *warning*. *Unit test* dengan Vitest menjalankan 36 file *test* dengan 230 *test case*, seluruhnya lulus. Skenario pengujian dan metrik evaluasi implementasi dapat dilihat pada **Tabel 1**.

Tabel 1. Metrik Evaluasi Implementasi

Metrik	Hasil yang Diamati
Validasi TypeScript/Svelte	0 error dan 0 warning pada svelte-check
Unit test	36 file *test* dan 230 *test case* lulus pada *Vitest*
Cron job operasional	5 *cron job* berjalan tanpa server dedicated
TTL cache ongkir	7 hari untuk data *shipping_rates_cache*
Batas *pre-fetch* ongkir	Maksimal 40 destinasi per eksekusi *cron sync-shipping*
Rate limit cek ongkir	Maksimal 30 *request* per pengguna per menit
Batas berat pengiriman	Maksimal 50.000 gram per permintaan ongkir dan *checkout*
Retry email	Email *pending* dicoba ulang sampai 3 kali sebelum ditandai *failed*
Pembatasan akses	*RLS* dan *RBAC* memisahkan akses *customer*, *admin*, dan *superadmin*
Biaya marketplace	Tidak ada komisi *platform marketplace* pada transaksi di kanal mandiri
Biaya eksternal	*Service fee payment gateway* dan ongkos kirim tetap dicatat terpisah

Sumber: Hasil pengujian dan dokumentasi implementasi

4. Kesimpulan

Penelitian ini menjawab kebutuhan kanal penjualan mandiri bagi satu *merchant* UMKM dengan menunjukkan bahwa fungsi dasar *marketplace* dapat dijalankan di atas Vercel dan Supabase *free-tier* tanpa biaya *hosting* awal dan tanpa komisi *marketplace*. Kelayakan tersebut bergantung pada mekanisme pengendalian sumber daya, yaitu *cache* ongkir, *cache in-memory*, *cron job* untuk *pre-fetch* ongkir, *auto-expire* pembayaran, *retry* email, *keepalive* database, *cleanup log*, serta validasi *server-side* untuk stok, pembayaran, dan akses admin. Temuan ini juga membatasi ruang klaim penelitian: platform belum diuji dengan load testing, belum memiliki *failover* saat kuota *free-tier* habis, dan beberapa fitur pendukung seperti tag produk, varian produk transaksional, serta ledger refund belum menjadi alur utama. Penelitian lanjutan perlu mengukur kapasitas pengguna simultan, menghitung biaya ketika sistem naik dari *free-tier*, dan menentukan prioritas fitur saat platform dipakai dalam operasi UMKM yang lebih luas.

5. Daftar Pustaka

[1] H. Noprisson, "Exploring e-Tourism : Technology and Human Factors," *Int. J. Sci. Res. Sci. Eng. Technol.*, pp. 169–177, Sep. 2021, doi: 10.32628/IJSRSET207540.

[2] G. Gata and A. Ratnasari, "Penerapan E-Commerce Content Management System Menggunakan Metode Business Model Canvas Studi Kasus Qorina Garden," *Bit (Fakultas Teknol. Inf. Univ. Budi Luhur)*, vol. 18, no. 2, pp. 55–62, 2021.

[3] D. A. Pangrepti and V. Ayumi, "Sistem Pemesanan Berbasis CodeIgniter Untuk Layanan Moju Massage Berdasarkan Analisis Software Requirements Specification," *JUKOMIKA (Jurnal Ilmu Komput. dan Inform.*, vol. 8, no. 2, pp. 148–154, 2025.

[4] M. Purba, J. Junaidi, L. Iryani, N. Umilizah, H. Noprisson, and N. Ani, "An Ensemble Boosting Approach with Boruta Feature Selection for Predicting E-Payment Adoption," *Int. J. Adv. Comput. Sci. Appl.*, vol. 17, no. 4, p. 403, 2026.

[5] J. Junianto and H. Noprisson, "Pengembangan Aplikasi Inventory Permintaan Alat Tulis Kantor Pemerintahan Berbasis Web Framework Codeigniter," *JUKOMIKA (Jurnal Ilmu Komput. dan Inform.*, vol. 8, no. 2, pp. 142–147, 2025.

[6] M. M. Kananga and V. Ayumi, "Perancangan Sistem Untuk Monitoring Operasional Kontainer dengan Implementasi Metode Agile dan Decision Tree," *JUSIBI (Jurnal Sist. Inf. dan E-Bisnis)*, vol. 8, no. 1, pp. 31–37, 2026.

[7] N. Alifia, E. Permana, and H. Harnovinsah, "Analisis penggunaan QRIS terhadap peningkatan pendapatan UMKM," *J. Ris. Pendidik. Ekon.*, vol. 9, no. 1, pp. 102–115, 2024.

[8] R. N. Husna, L. Syafina, and Y. S. J. Nasution, "Analisis BEP dan MOS sebagai alat perencanaan laba pada UMKM di Toko Berlian Pancing," *J. Manaj. Terap. Dan Keuang.*, vol. 13, no. 03, pp. 772–782, 2024.

[9] R. Kusumastuti, D. Oktafiani, T. D. Putra, and H. Turmudi, "Optimalisasi Digital Marketplace: Sosialisasi Jual Beli Online Untuk Mendukung Peningkatan Ekonomi di Desa Tanjungsari Boyolali," *JGEN J. Pengabdi. Kpd. Masy.*, vol. 3, no. 6, pp. 819–827, 2025.

[10] A. R. T. Ramadhana, H. Hermansyah, and S. Wahyuni, "Perancangan UI/UX Website Sakiverse Dengan Metode User-Centered Design Dalam Meningkatkan Kemudahan Penjualan Produk Pakaian," *J. Minfo Polgan*, vol. 14, no. 2, pp. 1519–1533, 2025.

[11] R. Yanuar and F. Fersellia, "Implementasi E-commerce Penjualan Kaos Distro Berbasis Website pada Toko Art n Soul Purbalingga," *INSOLOGI J. Sains dan Teknol.*, vol. 4, no. 3, pp. 379–394, 2025.

[12] A. Wahab, F. Z. Rifa'i, F. M. Serimole, and M. B. Yel, "Perancangan E-Katalog Berbasis Web untuk Meningkatkan Promosi dan Branding Usaha UMKM Air Ikan Otista Jakarta Timur," *J. Innov. Comput. Sci.*, vol. 5, no. 1, pp. 1–18, 2026.

[13] J. S. Zai and H. Noprisson, "Pengembangan Sistem Monitoring dan Pelaporan Distribusi Barang Menggunakan Metodologi Waterfall dan Blackbox (Studi Kasus: PT. Mitra Mega Sinergi)," *JSAI (Journal Sci. Appl. Informatics)*, vol. 8, no. 2, pp. 514–519, 2025.

[14] N. Luhyana and V. Ayumi, "Model Sistem Manajemen Proyek Sosial Media di Perusahaan Periklanan Menggunakan Metode Waterfall," *JUKOMIKA (Jurnal Ilmu Komput. dan Inform.*, vol. 8, no. 2, pp. 136–141, 2025.

[15] A. Suhendi and V. Ayumi, "Development of Web-Based Cash Management System (CMS) Based on Cash Processing Center (CPC) at PT. XYZ," *JSAI (Journal Sci. Appl. Informatics)*, vol. 8, no. 2, pp. 503–508, 2025.

[16] S. Mailana and V. Ayumi, "Aplikasi ConTrack Untuk Manjemen dan Monitoring Data Stok Barang di Perusahaan Jasa Konstruksi," *JSAI (Journal Sci. Appl. Informatics)*, vol. 8, no. 2, pp. 520–525, 2025.

[17] S. A. Saputri, I. Berliana, and M. F. Nasrida, "Peran marketplace dalam meningkatkan daya saing UMKM di Indonesia," *Knowl. J. Inov. Has. Penelit. dan Pengemb.*, vol. 3, no. 1, pp. 69–75, 2023.

[18] M. Manurung, "Peran marketplace dalam meningkatkan akses pemasaran UMKM di Indonesia," *AB-JOIEC Al-Bahjah J. Islam. Econ.*, vol. 2, no. 02, pp. 74–81, 2024.

[19] T. Ardiansyah, "Model platform e-commerce dalam mendukung kesuksesan UMKM di Indonesia," *J. Usaha*, vol. 1, no. 1, pp. 1–12, 2020.

[20] D. Rizky and F. Zulfikar, "Pengembangan Sistem E-Commerce Berbasis Web dengan Optimalisasi Database untuk Penjualan Produk UMKM," *J. Inf. Syst. Bus. Technol.*, vol. 1, no. 4, pp. 50–53, 2025.

[21] W. Haryadi, "Optimasi Pemanfaatan E-Commerce Pada Pemasaran Produk Umkm Dalam Membantu Peningkatan Pendapatan Masyarakat," *J. Pengemb. Masy. Lokal*, vol. 4, no. 2, pp. 243–248, 2021.

[22] V. Fujiyanti, G. M. Suranegara, and I. N. Ichsan, "Comparative Analysis of Server-Based and Serverless Service Performance on Google Cloud Platform (GCP)(Case Study: Machine Learning Model Deployment)," *J. Inf. Syst. Informatics*, vol. 6, no. 2, pp. 1172–1194, 2024.

6. Penulis

Fadel Muhamad Rifai merupakan mahasiswa Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Dian Nusantara, Jakarta, Indonesia. Minat akademiknya meliputi pengembangan perangkat lunak, kecerdasan buatan, analisis data, dan teknologi informasi. Saat ini, ia aktif mengikuti berbagai kegiatan akademik dan penelitian di bidang informatika untuk mengembangkan kompetensi dan pengalaman profesional. Penulis dapat dihubungi melalui email: faaadelmr@gmail.com.

Vina Ayumi meraih gelar Sarjana Komputer (S.Kom.) dari Universitas Islam Negeri Syarif Hidayatullah Jakarta pada tahun 2012, gelar Magister Komputer (M.Kom.) dari Universitas Indonesia pada tahun 2015, dan gelar Doktor (Dr.) bidang Informatika dari Universitas Sriwijaya pada tahun 2023. Saat ini, beliau merupakan dosen senior pada Program Studi Informatika, Fakultas Teknik dan Informatika, Universitas Dian Nusantara, Jakarta. Bidang penelitian yang ditekuni meliputi kecerdasan buatan, *deep learning*, *computer vision*, pengolahan citra, pengenalan pola, pemrosesan bahasa alami, serta sistem pendukung keputusan cerdas. Beliau telah menghasilkan lebih dari 100 publikasi ilmiah pada jurnal dan konferensi nasional maupun internasional, dengan lebih dari 700 sitasi di Google Scholar.