



Implementasi SaaS Multi-Tenant Berbasis Database-per-Tenant dan Metode FEFO pada Sistem Inventori dan Point of Sale UMKM Bakery

Implementation of Database-per-Tenant Multi-Tenant SaaS and the FEFO Method in an Inventory and Point of Sale System for Bakery SMEs

Rizki Wahyu*, Jefry Sunupurwa Asri, Nizirwan Anwar, Adi Widianono

Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Esa Unggul, Jakarta, Indonesia
Jl. Arjuna Utara No. 9, Duri Kepa, Kec. Kb. Jeruk, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta, Indonesia
Email: ¹*rw43917@student.esaunggul.ac.id, ²jefry.sunupurwa@esaunggul.ac.id, ³nizirwan.anwar@esaunggul.ac.id, ⁴adi.widianono@esaunggul.ac.id

Email Penulis Korespondensi: rw43917@student.esaunggul.ac.id

Received: 19/08/2026, Revised: 17/09/2026, Accepted: 07/10/2026, Published: 18/10/2026

Abstrak—Usaha Mikro, Kecil, dan Menengah (UMKM) pada sektor bakery mengelola produk dengan masa simpan terbatas dengan masa simpan terbatas sehingga membutuhkan pengendalian persediaan yang akurat serta rotasi produk berdasarkan tanggal kedaluwarsa. Toko Kue Khaerul Akbar masih menghadapi berbagai kendala, seperti pencatatan stok secara manual, ketidaksesuaian hasil stock opname, belum optimalnya prioritas terhadap produk yang mendekati kedaluwarsa, serta keterlambatan penyajian laporan operasional. Penelitian ini mengembangkan sistem inventori dan Point of Sale berbasis Software as a Service dengan menerapkan arsitektur database-per-tenant dan metode First Expired First Out. Pengembangan sistem dilakukan menggunakan metode Extreme Programming dalam empat sprint yang meliputi pembangunan fondasi multi-tenant, pengelolaan inventori berbasis batch, integrasi FEFO dengan POS, serta pengembangan fitur Owner Central. Metode FEFO menentukan prioritas batch berdasarkan nilai expired at terdekat dan menggunakan received at sebagai kriteria kedua apabila terdapat kesamaan tanggal kedaluwarsa. Pengujian unit dilakukan terhadap 12 kasus yang dikelompokkan ke dalam empat skenario, dengan seluruh kasus dinyatakan berhasil. Sementara itu, pengujian black-box terhadap 31 skenario menunjukkan tingkat keterpenuhan sebesar 100%. Pengujian isolasi juga membuktikan bahwa akses lintas tenant berhasil ditolak. Evaluasi terhadap 10 responden menghasilkan skor UAT sebesar 4,4 dari 5,0 dan skor SUS sebesar 81,7 dari 100. Sistem yang dikembangkan meningkatkan keterlacakan persediaan, memprioritaskan produk berdasarkan kedekatan tanggal kedaluwarsa, menyediakan laporan secara real-time, serta mendukung pemantauan beberapa unit usaha secara terpusat tanpa mencampurkan data operasional antar-tenant. Kontribusi penelitian ini adalah integrasi arsitektur SaaS database-per-tenant, alokasi FEFO berbasis batch yang terhubung langsung dengan checkout POS, serta Owner Central untuk pemantauan lintas unit dengan tetap mempertahankan isolasi data operasional setiap tenant.

Kata Kunci: SaaS; Multi-Tenant; FEFO; Point of Sale; Manajemen Inventori

Abstract—Bakery micro, small, and medium-sized enterprises manage products with limited shelf lives, requiring accurate inventory control and product rotation based on expiration dates. Toko Kue Khaerul Akbar continues to face several operational challenges, including manual stock recording, discrepancies in stock-taking results, inadequate prioritization of products nearing expiration, and delays in generating operational reports. This study develops an inventory and Point of Sale system based on the Software as a Service model by implementing a database-per-tenant architecture and the First Expired, First Out method. The system was developed using the Extreme Programming method through four sprints, comprising the development of the multi-tenant foundation, batch-based inventory management, FEFO integration with the POS system, and the Owner Central feature. The FEFO method prioritizes batches according to the earliest expiration date, using the received date as a secondary criterion when expiration dates are identical. Unit testing was conducted on 12 test cases organized into four scenario groups, with all cases successfully passing. In addition, black-box testing involving 31 scenarios achieved a fulfillment rate of 100%. Isolation testing confirmed that cross-tenant access was successfully denied. An evaluation involving 10 respondents yielded a UAT score of 4.4 out of 5.0 and a SUS score of 81.7 out of 100. The developed system improves inventory traceability, prioritizes products according to expiration proximity, provides real-time operational reports, and enables centralized monitoring of multiple business units while preserving the isolation of operational data across tenants. The contribution of this study is the integration of a database-per-tenant SaaS architecture, batch-based FEFO allocation directly connected to POS checkout, and Owner Central for cross-unit monitoring while maintaining operational data isolation for each tenant.

Keywords: SaaS; Multi-Tenant; FEFO; Point of Sale; Inventory Management

1. PENDAHULUAN

Usaha Mikro, Kecil, dan Menengah memiliki peran strategis dalam perekonomian Indonesia melalui perluasan aktivitas ekonomi masyarakat serta kontribusinya terhadap penyerapan tenaga kerja [1], [2]. Dalam konteks usaha bakery dan toko kue, pengelolaan operasional memiliki karakteristik tersendiri karena produk jadi dan sebagian bahan bakunya bersifat mudah rusak (perishable) dengan masa simpan yang terbatas. Perencanaan persediaan yang kurang tepat pada usaha kue dan roti dapat menimbulkan kelebihan maupun kekurangan stok serta meningkatkan pemborosan produk, yang pada akhirnya berdampak pada biaya operasional [3], [4]. Oleh sebab itu, pencatatan



persediaan pada UMKM bakery tidak cukup hanya memuat jumlah stok, tetapi juga perlu mencakup identitas batch, tanggal penerimaan, tanggal kedaluwarsa, serta riwayat perubahan stok. Informasi tersebut diperlukan untuk memastikan kondisi fisik persediaan dapat direkonsiliasi secara akurat dengan data pada sistem.

Permasalahan tersebut juga dijumpai pada Toko Kue Khaerul Akbar. Berdasarkan hasil observasi dan wawancara, pencatatan persediaan sebelumnya masih dilakukan secara manual dan belum terintegrasi secara langsung dengan transaksi penjualan. Kondisi tersebut menyulitkan proses rekonsiliasi hasil stock opname, terutama ketika pergerakan barang berlangsung cepat, karena perubahan stok belum terdokumentasi secara terstruktur hingga tingkat batch. Sejumlah penelitian telah menerapkan digitalisasi inventori untuk meningkatkan keteraturan dan keterbaruan informasi persediaan [5], [6], termasuk pengembangan e-inventory berbasis perangkat bergerak yang menitikberatkan pada pembaruan data secara real-time [7] serta sistem inventori bahan baku berbasis web yang mendukung penyajian laporan persediaan [8]. Namun, kebutuhan pada objek penelitian ini tidak terbatas pada digitalisasi pencatatan stok, tetapi juga mencakup penentuan batch yang perlu diprioritaskan ketika produk dijual.

Permasalahan selanjutnya berkaitan dengan rotasi produk berdasarkan masa kedaluwarsa. Dalam proses sebelumnya, pengambilan produk oleh kasir atau pelayan cenderung didasarkan pada posisi fisik yang paling mudah dijangkau. Kondisi ini memungkinkan batch yang lebih baru terjual terlebih dahulu, sementara batch dengan masa simpan lebih pendek tertinggal. Metode First Expired First Out relevan diterapkan karena memprioritaskan persediaan dengan tanggal kedaluwarsa paling dekat [9]. Dalam pengelolaan pangan, prinsip First Expired, First Out juga digunakan untuk mengurangi potensi kehilangan produk akibat keterbatasan umur simpan dan gangguan pada rantai dingin [10]. Oleh karena itu, penerapan FEFO dalam penelitian ini tidak hanya berfungsi untuk mengurutkan data inventori, tetapi juga harus terintegrasi dengan transaksi penjualan agar batch yang direkomendasikan sistem sesuai dengan produk fisik yang diambil oleh kasir.

Digitalisasi transaksi melalui Point of Sale telah banyak diterapkan pada sektor UMKM. Aplikasi POS berbasis web dapat mendukung pencatatan penjualan sekaligus menyajikan informasi transaksi secara real-time [11]. Sementara itu, pendekatan berbasis pengguna menekankan aspek kemudahan interaksi agar sistem dapat diterima dan dimanfaatkan secara optimal dalam kegiatan operasional [12]. Penerapan POS pada perangkat bergerak juga dilaporkan berkontribusi terhadap peningkatan kinerja usaha [13]. Selain itu, metode Extreme Programming telah digunakan dalam pengembangan POS secara iteratif untuk memenuhi kebutuhan lingkungan usaha yang memerlukan adaptasi cepat [14]. Meskipun demikian, POS konvensional pada umumnya hanya memperbarui persediaan pada tingkat produk dan belum menjadikan tanggal kedaluwarsa sebagai dasar utama dalam menentukan batch yang harus dikurangi saat proses checkout.

Pada aspek arsitektur, kompleksitas penelitian meningkat karena pemilik dapat mengelola lebih dari satu unit usaha. Penelitian [15] mengembangkan aplikasi penjualan dengan metode Extreme Programming dan menerapkan model multi-tenancy. Kajian tentang SaaS menunjukkan bahwa layanan terpusat dapat mendukung operasional bisnis sekaligus meningkatkan skalabilitas penggunaan aplikasi [16], [17]. Dalam arsitektur multi-tenant, isolasi data merupakan aspek krusial untuk mencegah suatu organisasi mengakses data milik organisasi lainnya [18]. Pengelolaan penyimpanan data dapat dilakukan melalui basis data bersama, skema bersama, atau database-per-tenant. Setiap strategi memiliki konsekuensi yang berbeda dalam hal tingkat isolasi, efisiensi sumber daya, dan kompleksitas pengelolaan [19], [20]. Penerapan SaaS juga relevan pada distribusi produk segar karena mampu menyediakan layanan secara berkelanjutan bagi pelaku usaha dengan kebutuhan operasional yang dinamis [21].

Berdasarkan kajian penelitian terdahulu, dapat diidentifikasi tiga kesenjangan utama. Pertama, sebagian besar pengembangan sistem POS dan inventori untuk UMKM masih menggunakan pendekatan single-tenant, sehingga belum secara eksplisit mendukung pengelolaan beberapa unit usaha yang berdiri secara independen dengan basis data operasional yang terpisah. Kedua, penelitian mengenai FEFO pada umumnya berfokus pada penerapan aturan kedaluwarsa dalam proses pengeluaran persediaan. Sementara itu, integrasi hasil alokasi FEFO dengan proses checkout pada POS, pengurangan stok berdasarkan batch, pencatatan detail penjualan, dan pencatatan stock movement dalam satu alur transaksi yang konsisten masih belum banyak dikaji. Ketiga, kebutuhan pemilik untuk memantau beberapa unit usaha melalui satu panel terpusat berpotensi bertentangan dengan prinsip isolasi data. Oleh karena itu, diperlukan rancangan sistem yang mampu menyediakan pemantauan lintas unit tanpa menggabungkan data transaksi dan operasional dari masing-masing tenant.

Penelitian ini mengembangkan sistem manajemen inventori dan Point of Sale berbasis Software as a Service dengan menerapkan pola database-per-tenant. Setiap unit toko direpresentasikan sebagai tenant independen yang memiliki domain dan basis data operasional tersendiri. Central Database hanya menyimpan metadata tenant, domain, status, akun pengguna pusat, serta hubungan kepemilikan melalui `owner_admin_id`. Sementara itu, data produk, batch, stok, transaksi, dan pergerakan stok disimpan pada Tenant Database. Pada modul inventori, metode First Expired First Out digunakan untuk mengurutkan batch aktif berdasarkan tanggal kedaluwarsa terdekat melalui atribut `expired_at`. Apabila terdapat batch dengan tanggal kedaluwarsa yang sama, atribut `received_at` yang paling awal digunakan sebagai kriteria prioritas berikutnya. Hasil alokasi tersebut ditampilkan kepada kasir dalam bentuk rekomendasi batch, sehingga pengurangan stok pada sistem dapat disesuaikan dengan produk fisik yang diambil.

Penelitian ini bertujuan mengimplementasikan SaaS multi-tenant berbasis database-per-tenant yang mampu menjaga isolasi data antarunit usaha, menerapkan metode FEFO pada inventori berbasis batch, mengintegrasikan alokasi FEFO dengan proses POS, serta menyediakan fasilitas pemantauan beberapa tenant melalui Owner Central. Sistem yang dikembangkan diharapkan dapat meningkatkan keterlacakan persediaan, memprioritaskan produk dengan tanggal kedaluwarsa yang lebih dekat, menggantikan proses rekapitulasi manual dengan penyajian laporan secara real-time, dan menyediakan akses pemantauan terpusat bagi pemilik tanpa mencampurkan data operasional setiap tenant. Kontribusi utama penelitian ini terletak pada penggabungan tiga aspek dalam satu alur operasional, yaitu isolasi data melalui arsitektur database-per-tenant, alokasi dan konfirmasi FEFO hingga tingkat batch yang terintegrasi langsung dengan checkout POS, serta pemantauan multi-unit melalui Owner Central tanpa mereplikasi data transaksi antar-tenant ke basis data pusat.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini dilaksanakan di Toko Kue Khaerul Akbar, Kabupaten Tangerang, Banten, dengan fokus pada pengelolaan inventori dan transaksi penjualan. Data diperoleh melalui observasi terhadap proses penerimaan barang, penyimpanan produk, transaksi penjualan, perubahan stok, dan penyusunan laporan. Selain itu, wawancara semi-terstruktur dilakukan dengan satu orang pemilik dan dua orang kasir. Data yang terkumpul digunakan sebagai dasar dalam merumuskan kebutuhan fungsional dan nonfungsional sistem. Tenant utama merepresentasikan Toko Kue Khaerul Akbar, sedangkan tenant tambahan dimanfaatkan sebagai lingkungan pengujian untuk memvalidasi proses tenant provisioning, relasi kepemilikan, dan isolasi data.

Tahapan penelitian meliputi identifikasi masalah, pengumpulan data, analisis kebutuhan, pengembangan sistem menggunakan metode Extreme Programming, penerapan arsitektur multi-tenant dan metode FEFO, serta pengujian dan evaluasi sistem. Rangkaian tahapan tersebut dirancang agar solusi yang dikembangkan tidak hanya dijelaskan secara konseptual, tetapi juga diimplementasikan pada modul inventori dan POS serta dievaluasi berdasarkan kebutuhan pengguna. Ringkasan tahapan penelitian disajikan pada Tabel 1.

Tabel 1. Tahapan Penelitian

Tahap	Aktivitas Utama	Keluaran
Identifikasi masalah	Observasi proses stok, rotasi kedaluwarsa, transaksi, dan laporan	Masalah dan kebutuhan awal
Pengumpulan data	Observasi dan wawancara pemilik serta dua kasir	Kebutuhan fungsional/non-fungsional
Pengembangan	Empat sprint Extreme Programming	SaaS multi-tenant, inventori batch, POS FEFO, Owner Central
Penerapan metode	Pengurutan expired_at dan received_at serta alokasi batch	Rekomendasi batch dan pengurangan stok FEFO
Pengujian	Unit, black-box, isolasi tenant, UAT, dan SUS	Hasil keberhasilan fungsi dan evaluasi pengguna

Berdasarkan Tabel 1, penelitian dilaksanakan secara berurutan mulai dari identifikasi masalah dan pengumpulan data sebagai dasar penyusunan kebutuhan sistem. Hasil analisis tersebut selanjutnya digunakan dalam proses pengembangan melalui empat sprint Extreme Programming, penerapan arsitektur SaaS multi-tenant dan metode FEFO, serta pengujian untuk memastikan fungsi sistem, isolasi data antar-tenant, dan kesesuaian sistem dengan kebutuhan pengguna.

2.2 Metode Pengembangan Extreme Programming

Pengembangan perangkat lunak dalam penelitian ini menggunakan metode Extreme Programming karena mendukung siklus iterasi yang singkat, adaptasi terhadap perubahan kebutuhan, serta pelaksanaan pengembangan oleh tim berukuran kecil. Metode ini telah diterapkan pada pengembangan aplikasi penjualan dan Point of Sale [14], termasuk sistem yang mengadopsi model multi-tenancy [15]. Tahapan Extreme Programming yang digunakan meliputi planning, designing, coding, dan testing, yang dilaksanakan secara berulang dalam empat sprint. Sprint pertama berfokus pada pembangunan Central Panel, autentikasi pengguna pusat, pengelolaan tenant, konfigurasi domain, Central Database, dan proses provisioning. Sprint kedua mencakup pengembangan fitur kategori, produk, unit penjualan, penerimaan stok, serta pengelolaan product batches. Sprint ketiga diarahkan pada integrasi metode FEFO dengan POS, penyajian rekomendasi batch, pemrosesan transaksi penjualan, pencatatan detail transaksi, dan pencatatan stock_movements. Sprint keempat mencakup pembangunan dashboard Owner Central, penyusunan laporan, serta evaluasi sistem.

Implementasi aplikasi web dilakukan menggunakan framework Laravel karena menyediakan struktur pengembangan berbasis PHP yang mendukung pembangunan aplikasi web [22]. Meskipun demikian, penelitian ini tidak berfokus pada perbandingan framework, melainkan pada perancangan arsitektur tenant, layanan alokasi

FEFO, dan proses transaksi POS untuk memastikan konsistensi perubahan stok. Penerapan prinsip pemisahan tanggung jawab dan konsistensi data juga relevan dengan karakteristik sistem intensif data yang menuntut reliabilitas dalam proses pembacaan serta pembaruan data oleh berbagai komponen [23].

2.3 Arsitektur Sistem SaaS Multi-Tenant

Sistem menerapkan pola Software as a Service dengan arsitektur database-per-tenant. Central Database berfungsi menyimpan metadata tenant, domain atau subdomain, status tenant, akun pengguna pusat, serta owner_admin_id. Basis data pusat tersebut tidak menyimpan data produk, batch, stok, transaksi POS, maupun stock_movements. Setiap tenant memiliki Tenant Database tersendiri yang digunakan untuk menyimpan seluruh data operasionalnya. Ketika domain tenant diakses, aplikasi mengidentifikasi konteks tenant dan mengaktifkan koneksi ke basis data yang sesuai. Pemilihan desain ini didasarkan pada pertimbangan bahwa strategi pengelolaan basis data berpengaruh langsung terhadap tingkat isolasi data dan skalabilitas arsitektur multi-tenant [18], [19], [20].

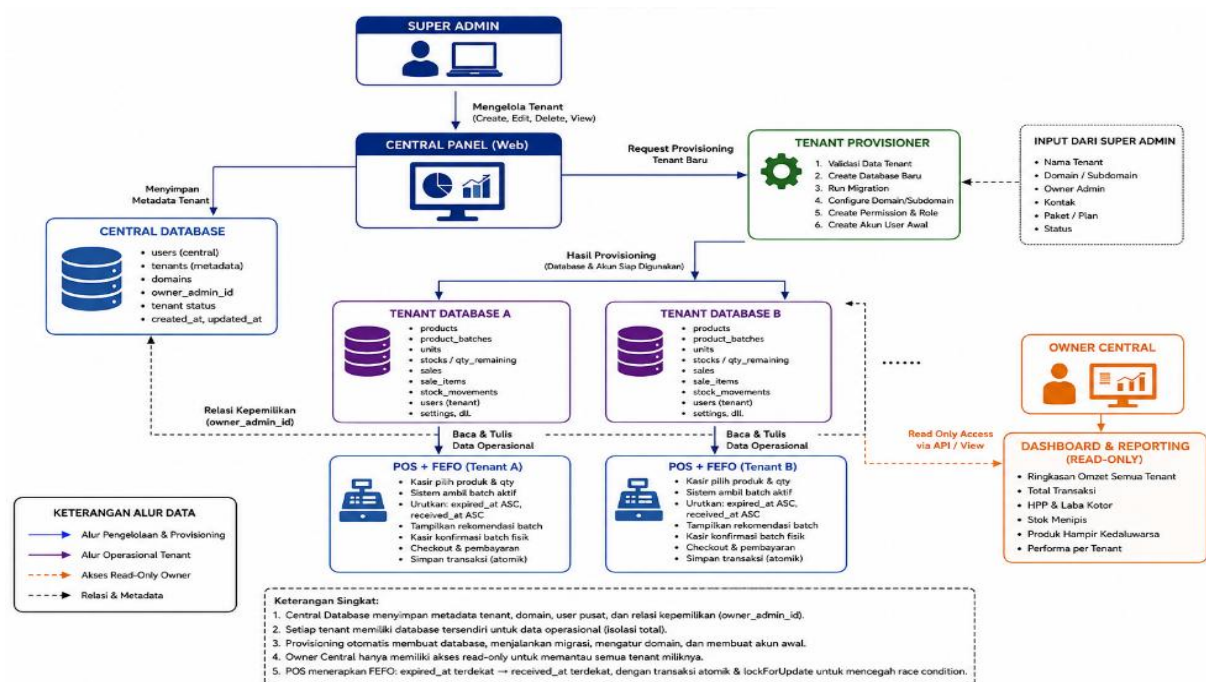
Proses tenant provisioning dilakukan ketika Super Admin membuat tenant baru. Tahap ini meliputi validasi input, penyimpanan metadata, pembuatan basis data tenant, pelaksanaan migrasi, konfigurasi domain atau subdomain, penetapan hak akses, serta pembuatan akun pengguna awal. Owner Central dihubungkan melalui atribut owner_admin_id dan hanya diberikan hak akses baca (read-only) terhadap informasi tenant yang berada dalam kepemilikannya. Pemisahan layanan yang tetap dapat dipantau secara terpusat sejalan dengan konsep penyediaan aplikasi sebagai layanan [16], [17]. Selain itu, pemisahan komponen layanan turut mendukung kejelasan batas tanggung jawab dalam pengembangan dan pemeliharaan sistem [24]. Arsitektur keseluruhan sistem ditunjukkan pada Gambar 1.

2.4 Penerapan Metode First Expired First Out

FEFO diterapkan setelah kasir memilih produk dan memasukkan jumlah yang diperlukan. Sistem kemudian mengambil batch yang berstatus aktif, memiliki nilai qty_remaining lebih dari nol, dan belum mencapai tanggal kedaluwarsa (expired_at). Batch tersebut diurutkan secara menaik berdasarkan expired_at. Apabila terdapat batch dengan tanggal kedaluwarsa yang sama, nilai received_at paling awal digunakan sebagai kriteria prioritas berikutnya. Aturan ini mempertahankan prioritas berdasarkan kedekatan tanggal kedaluwarsa [9], [10] sekaligus menggunakan waktu penerimaan sebagai pembeda ketika tanggal kedaluwarsa pada beberapa batch identik.

Apabila persediaan pada batch pertama tidak mencukupi, sistem melanjutkan alokasi ke batch berikutnya hingga jumlah kebutuhan terpenuhi. Namun, apabila total stok aktif tidak mencukupi, transaksi ditolak. Pada tahap checkout, proses penyimpanan data penjualan dan detail item, pengurangan nilai qty_remaining, serta pencatatan stock_movements dijalankan dalam satu transaksi basis data yang bersifat atomik. Mekanisme ini diterapkan untuk mencegah perubahan parsial, seperti transaksi yang berhasil tersimpan tanpa disertai pengurangan stok atau stok yang telah berkurang ketika transaksi gagal disimpan. Dengan demikian, konsistensi data menjadi bagian integral dari implementasi FEFO, bukan sekadar konsekuensi dari proses pengurutan batch.

Struktur pemisahan Central Database dan Tenant Database serta hubungan antaraktor dalam sistem secara visual disajikan pada Gambar 1.



Gambar 1. Arsitektur SaaS Multi-Tenant dengan Database-per-Tenant

Gambar 1 menunjukkan pemisahan antara Central Database dan dua Tenant Database. Central Panel berfungsi sebagai pusat administrasi serta pemantauan oleh Owner Central, sedangkan aktivitas POS dan inventori dijalankan dalam konteks masing-masing tenant. Relasi kepemilikan memungkinkan satu Owner Central memantau lebih dari satu tenant tanpa memindahkan tabel transaksi operasional tenant ke dalam basis data pusat.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi Arsitektur Multi-Tenant dan Tenant Provisioning

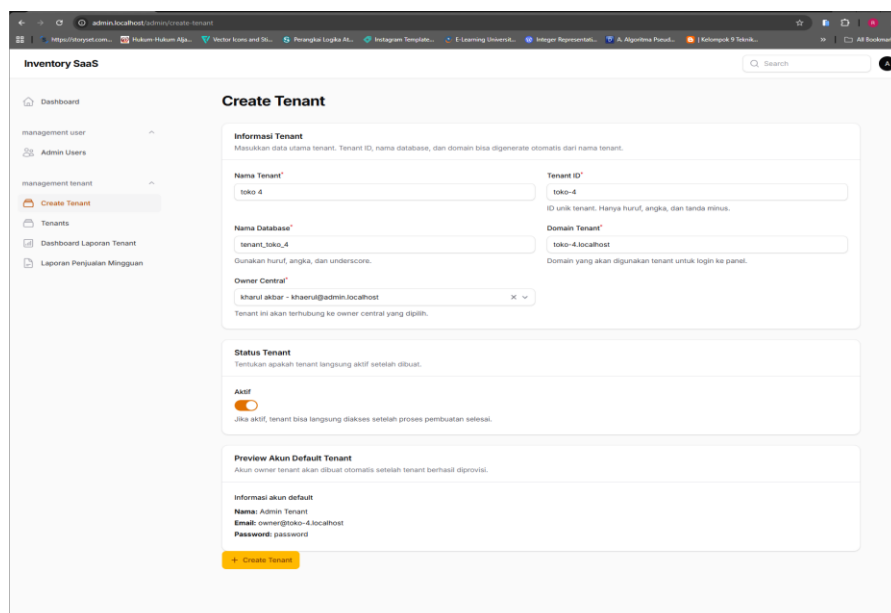
Implementasi menghasilkan dua lingkungan tenant untuk pengujian, masing-masing dengan domain dan basis data operasional yang berbeda. Tenant pertama merepresentasikan Toko Kue Khaerul Akbar, sedangkan tenant kedua digunakan untuk menguji proses provisioning, pemisahan koneksi, relasi kepemilikan, dan mekanisme pencegahan akses silang. Super Admin membuat tenant melalui Central Panel dengan mengisi nama tenant, Tenant ID, nama basis data, domain atau subdomain, status tenant, serta Owner Central. Formulir tersebut ditampilkan pada Gambar 2. Setelah seluruh data dinyatakan valid, layanan provisioning menyimpan metadata tenant pada Central Database, membuat basis data baru, menjalankan migrasi struktur, mengonfigurasi hak akses, dan membuat akun pengguna awal.

Hasil provisioning menunjukkan bahwa setiap tenant memiliki domain dan basis data operasional yang terpisah. Ketika Tenant A diakses, aplikasi mengarahkan koneksi ke Tenant Database A, sedangkan akses melalui domain Tenant B mengaktifkan koneksi ke Tenant Database B. Dengan demikian, data produk, batch, transaksi, pengguna, dan pergerakan stok tidak disimpan dalam tabel operasional yang sama. Batas isolasi antar-tenant tidak hanya diterapkan melalui antarmuka atau penyaringan berdasarkan `tenant_id`, tetapi juga melalui pemilihan koneksi basis data yang sesuai. Pendekatan ini sejalan dengan literatur yang menempatkan strategi penyimpanan sebagai salah satu faktor penting dalam menjamin isolasi pada arsitektur SaaS multi-tenant [18], [19], [20].

Gambar 2 memperlihatkan bahwa Owner Central dapat dipilih pada saat pembuatan tenant, sehingga relasi kepemilikan terbentuk sejak proses provisioning. Satu akun Owner Central dapat dikaitkan dengan beberapa tenant, sementara setiap tenant tetap memiliki independensi pada tingkat domain dan basis data. Pola ini berbeda dari pendekatan multi-unit yang hanya menambahkan atribut cabang pada tabel transaksi yang digunakan secara bersama-sama.

Pada pengujian otorisasi, pengguna Tenant A yang berupaya mengakses sumber daya milik Tenant B tidak memperoleh data dari tenant tersebut. Pengujian dengan menggunakan token Tenant A untuk mengakses API Tenant B menghasilkan respons 403 Forbidden. Hasil ini menegaskan bahwa mekanisme isolasi tidak semata-mata bergantung pada penyembunyian menu atau pembatasan pada antarmuka pengguna. Meskipun autentikasi dan otorisasi tetap menjadi komponen yang diperlukan, penerapan arsitektur database-per-tenant memberikan lapisan pembatas tambahan karena koneksi operasional yang aktif tidak menyediakan akses terhadap tabel transaksi milik tenant lain. Temuan ini memperkuat dasar pemilihan pola database-per-tenant dalam penelitian ini, meskipun penerapannya menimbulkan overhead pengelolaan yang lebih besar dibandingkan dengan penggunaan basis data bersama [19], [20].

Antarmuka yang digunakan Super Admin dalam proses pembuatan tenant dan penentuan Owner Central disajikan pada Gambar 2.



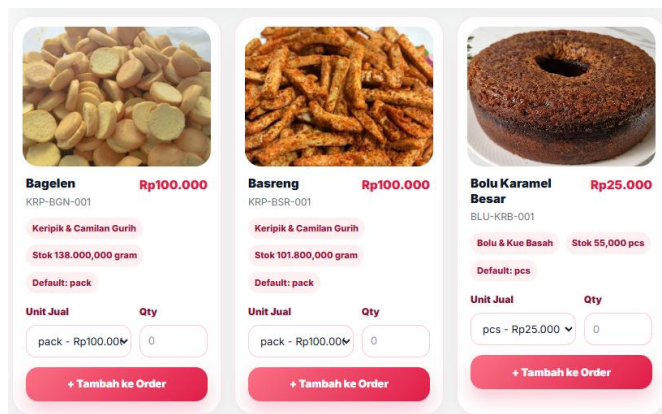
Gambar 2. Formulir Tenant Provisioning dan Pemilihan Owner Central

3.2 Implementasi FEFO pada Point of Sale

Layanan FefoStockAllocator mengambil data dari tabel product batches dan menyusun urutan alokasi berdasarkan expired_at sebagai kriteria utama serta received_at sebagai kriteria sekunder. Sistem hanya memproses batch yang berstatus aktif, memiliki qty_remaining lebih dari nol, dan belum melewati tanggal kedaluwarsa. Sebagai ilustrasi, apabila BATCH-001 memiliki tiga unit dengan tanggal kedaluwarsa 15 Juli 2026, sedangkan BATCH-002 memiliki sepuluh unit dengan tanggal kedaluwarsa 20 Juli 2026, pembelian dua unit akan dialokasikan seluruhnya dari BATCH-001. Apabila kebutuhan mencapai lima unit, sistem mengalokasikan tiga unit dari BATCH-001 dan dua unit dari BATCH-002. Mekanisme tersebut menerapkan prinsip FEFO dengan memprioritaskan batch yang memiliki tanggal kedaluwarsa paling dekat [9], [10] .

Perbedaan utama penelitian ini terletak pada pemanfaatan hasil alokasi yang tidak hanya berfungsi untuk memperbarui persediaan dalam basis data, tetapi juga sebagai rekomendasi pengambilan produk secara fisik pada halaman POS. Kasir memilih produk, unit penjualan, dan kuantitas, kemudian sistem menampilkan batch yang direkomendasikan berdasarkan hasil alokasi. Sebelum proses checkout, kasir mengonfirmasi batch yang digunakan sehingga identitas batch yang dikurangi dalam sistem sesuai dengan produk yang benar-benar diserahkan kepada pelanggan. Mekanisme ini meminimalkan risiko ketidaksesuaian antara stok fisik dan stok digital akibat perbedaan antara batch yang dikurangi oleh sistem dan batch yang diambil oleh kasir.

Integrasi tersebut memperluas peran POS yang dalam penelitian terdahulu umumnya berfokus pada digitalisasi transaksi dan peningkatan kemudahan penggunaan [11], [12], [13]. Dalam sistem yang dikembangkan, transaksi penjualan secara langsung memicu proses alokasi batch dan pembaruan data inventori. Setelah kasir mengonfirmasi item, keranjang menyimpan informasi produk, unit penjualan, kuantitas, harga, subtotal, serta batch yang dialokasikan. Namun, pengurangan stok tidak dilakukan pada tahap penyusunan keranjang. Perubahan persediaan baru diterapkan setelah proses checkout berhasil diselesaikan, sehingga pembatalan transaksi sebelum pembayaran tidak memengaruhi nilai qty_remaining. Implementasi rekomendasi batch FEFO pada antarmuka transaksi kasir dapat dilihat pada Gambar 3.



Gambar 3. Tampilan Produk pada Modul Point of Sale

Gambar 3 memperlihatkan tampilan produk pada modul Point of Sale yang digunakan kasir dalam memilih produk dan jumlah transaksi. Informasi pada modul tersebut terhubung dengan proses alokasi FEFO sehingga sistem dapat memberikan rekomendasi batch sesuai urutan tanggal kedaluwarsa sebelum transaksi dikonfirmasi.

Proses checkout dikelola oleh SaleService, yang bertanggung jawab menyimpan data transaksi dan detail penjualan, mengurangi nilai qty_remaining pada batch yang telah dikonfirmasi, serta mencatat pergerakan stok dalam stock_movements. Seluruh operasi tersebut dijalankan dalam satu transaksi basis data yang bersifat atomik. Dengan demikian, apabila salah satu operasi mengalami kegagalan, seluruh perubahan akan dibatalkan sehingga tidak terjadi kondisi ketika data penjualan tersimpan tanpa disertai pembaruan stok, atau sebaliknya. Pendekatan ini penting dalam sistem inventori karena keandalan informasi sangat bergantung pada konsistensi antarentitas yang diperbarui dalam satu rangkaian proses [23]. Dalam konteks manajemen pergudangan, keterkaitan antara transaksi dan inventori juga diperlukan untuk memastikan setiap pergerakan stok dapat ditelusuri [25].

3.3 Implementasi Owner Central

Owner Central diperuntukkan bagi pemilik yang mengelola satu atau lebih tenant. Pengguna mengakses sistem melalui Central Panel dan hanya dapat melihat tenant yang memiliki kecocokan pada atribut owner_admin_id. Data transaksi tidak direplikasi secara permanen ke Central Database. Sebaliknya, layanan pelaporan pusat mengambil data secara langsung dari Tenant Database yang telah memperoleh izin akses, kemudian menyajikan informasi tersebut dalam bentuk ringkasan. Akses Owner Central bersifat hanya-baca (read-only), sehingga pengguna tidak memiliki kewenangan untuk membuat atau menghapus tenant, mengubah data produk, memodifikasi batch, maupun melakukan transaksi pada Tenant Database.

Dashboard Owner Central menyajikan informasi mengenai jumlah tenant, omzet, jumlah transaksi, estimasi harga pokok penjualan, estimasi laba kotor, produk dengan stok menipis, batch yang mendekati kedaluwarsa, serta riwayat transaksi. Perancangan ini memungkinkan pemusatan informasi tanpa menggabungkan data operasional antar-tenant. Dengan demikian, pemilik dapat memperoleh gambaran aktivitas lintas unit, sementara pengelolaan stok dan transaksi tetap dilakukan melalui Tenant Panel masing-masing. Konsep tersebut sejalan dengan karakteristik layanan SaaS yang menyediakan akses terpusat untuk mendukung pengelolaan berbagai konteks pengguna [16], [17], [21].

3.4 Pengujian Sistem

3.4.1 Unit Testing FEFO

Pengujian unit dilakukan pada FefoStockAllocator untuk memverifikasi bahwa proses alokasi stok telah menerapkan prinsip FEFO secara konsisten. Pengujian tersebut mencakup empat kelompok skenario, yaitu kondisi ketika satu batch mampu memenuhi seluruh kebutuhan, kebutuhan yang harus dipenuhi dari beberapa batch, keberadaan batch yang telah kedaluwarsa, serta kondisi ketika total stok yang tersedia tidak mencukupi. Ringkasan hasil pengujian disajikan pada Tabel 2.

Tabel 2. Hasil Unit Testing FEFO

Skenario	Jumlah Uji	Berhasil	Gagal	Akurasi
Satu batch mencukupi	5	5	0	100%
Multiple batch	3	3	0	100%
Batch kedaluwarsa	2	2	0	100%
Stok tidak mencukupi	2	2	0	100%
Total	12	12	0	100%

Berdasarkan Tabel 2, seluruh 12 kasus uji menghasilkan keluaran yang sesuai dengan ekspektasi. Batch dengan tanggal kedaluwarsa paling dekat secara konsisten diprioritaskan. Apabila stok pada batch pertama tidak mencukupi, alokasi dilanjutkan ke batch berikutnya. Batch yang telah kedaluwarsa tidak dimasukkan sebagai kandidat alokasi, sedangkan transaksi ditolak apabila total stok aktif tidak mencukupi jumlah yang diminta. Hasil tersebut menunjukkan bahwa aturan berdasarkan atribut `expired_at` dan `received_at` telah diterapkan secara konsisten dalam seluruh skenario yang diuji. Meskipun demikian, nilai akurasi sebesar 100% merepresentasikan kesesuaian keluaran terhadap skenario pengujian, bukan performa sistem dalam menghadapi transaksi konkuren berskala besar.

3.4.2 Black-Box Testing dan Isolasi Data

Pengujian black-box dilakukan untuk mengevaluasi fungsionalitas sistem berdasarkan masukan dan keluaran yang dihasilkan, tanpa meninjau struktur internal kode, sebagaimana penerapannya dalam pengujian sistem informasi stok dan penjualan berbasis web [26]. Dalam penelitian ini, pengujian mencakup 31 skenario yang meliputi tenant provisioning, autentikasi, pengelolaan produk dan batch, rekomendasi FEFO, konfirmasi batch, transaksi POS, pembayaran, pengurangan stok, pelaporan, isolasi tenant, Owner Central, serta pembatasan hak akses. Ringkasan hasil pengujian disajikan pada Tabel 3.

Tabel 3. Ringkasan Black-Box Testing

Kelompok Pengujian	Hasil
Tenant provisioning dan autentikasi	Terpenuhi
Produk, batch, dan FEFO	Terpenuhi
POS, pembayaran, dan stock movement	Terpenuhi
Laporan dan Owner Central	Terpenuhi
Isolasi data dan otorisasi	Terpenuhi
Total	31/31 (100%)

Berdasarkan Tabel 3, seluruh 31 skenario pengujian memperoleh status Terpenuhi dengan tingkat keterpenuhan sebesar 100%. Pengujian isolasi merupakan aspek penting karena tujuan arsitektur tidak hanya menyediakan beberapa domain, tetapi juga mencegah terjadinya akses silang antar-tenant. Ketika sesi Tenant A berupaya mengakses sumber daya milik Tenant B, sistem tidak menampilkan data Tenant B. Selain itu, penggunaan token Tenant A untuk mengakses API Tenant B ditolak dengan respons 403 Forbidden. Hasil tersebut menunjukkan bahwa mekanisme pembatasan akses berfungsi pada jalur akses langsung dan tidak hanya bergantung pada antarmuka pengguna.

Kombinasi mekanisme otorisasi dan pemilihan koneksi menyediakan dua lapisan pengendalian. Dalam pendekatan shared-table, kesalahan query yang tidak menyertakan filter tenant berpotensi menyebabkan terbaca-nya baris milik tenant lain. Sementara itu, pada pendekatan database-per-tenant, konteks koneksi aktif hanya diarahkan ke basis data tenant yang sedang digunakan sehingga data operasional tenant lain tidak tersimpan pada tabel yang sama. Meskipun demikian, pendekatan tersebut tidak menghilangkan kebutuhan terhadap

autentikasi, validasi domain, dan otorisasi karena keamanan sistem tetap bergantung pada penerapan pengendalian akses secara menyeluruh [18], [19], [20].

3.4.3 Evaluasi Pengguna

User Acceptance Testing dan System Usability Scale dilakukan terhadap 10 responden. UAT menggunakan skala Likert 1–5 untuk menilai aspek kemudahan akses, penginputan produk, penginputan batch, kecepatan transaksi, akurasi FEFO, pelaporan, notifikasi kedaluwarsa, tampilan, dan keandalan sistem. Hasil evaluasi menunjukkan bahwa UAT memperoleh skor rata-rata 4,4 dari 5,0. Skor tertinggi, yaitu 5,0, diperoleh pada aspek kecepatan transaksi POS, notifikasi kedaluwarsa, dan keandalan sistem. Sementara itu, aspek penginputan batch memperoleh skor terendah, yaitu 3,7, karena pengguna perlu memasukkan kode batch, tanggal penerimaan, tanggal kedaluwarsa, jumlah, serta harga modal.

Evaluasi SUS terhadap responden yang sama menghasilkan skor rata-rata 81,7 dari 100, yang termasuk dalam kategori Baik dengan tingkat penerimaan Acceptable. Hasil tersebut menunjukkan bahwa kompleksitas proses internal FEFO tidak sepenuhnya dibebankan kepada kasir. Sistem secara otomatis melakukan pencarian, pengurutan, dan alokasi batch, sedangkan pengguna hanya perlu menerima rekomendasi serta melakukan konfirmasi. Temuan ini sejalan dengan penelitian mengenai sistem POS yang menempatkan kemudahan interaksi sebagai salah satu faktor penting dalam penerimaan pengguna [12]. Ringkasan hasil evaluasi disajikan pada Tabel 4.

Tabel 4. Ringkasan Evaluasi Pengguna

Parameter	Hasil
Jumlah responden	10
Rata-rata UAT	4,4 dari 5,0
Skor SUS rata-rata	81,7 dari 100
Kategori SUS	Baik
Tingkat penerimaan	Acceptable

Berdasarkan Tabel 4, evaluasi terhadap 10 responden menghasilkan rata-rata UAT sebesar 4,4 dari 5,0 dan skor SUS sebesar 81,7 dari 100. Hasil tersebut menempatkan usability sistem pada kategori Baik dengan tingkat penerimaan Acceptable, sehingga sistem secara umum dapat digunakan oleh pengguna dalam aktivitas operasional yang diuji.

3.5 Pembahasan

Dibandingkan dengan penelitian digitalisasi inventori pada [6], [7], [8], [25], penelitian ini tidak berhenti pada pembaruan data stok dan penyajian laporan, tetapi memperluas keterlacakan hingga tingkat batch serta menghubungkan perubahan persediaan langsung dengan transaksi POS. Penelitian [6] dan [7] menekankan digitalisasi serta pembaruan inventori, sedangkan [8] berfokus pada laporan inventori bahan baku dan [25] pada pengelolaan pergudangan. Pada penelitian ini, identitas batch, detail penjualan, dan stock movement ditempatkan dalam satu alur sehingga riwayat pengurangan stok dapat ditelusuri kembali ke transaksi yang memicunya.

Pada aspek rotasi kedaluwarsa, penelitian [9] dan [10] menempatkan FEFO sebagai aturan prioritas pengeluaran persediaan berdasarkan tanggal kedaluwarsa. Penelitian ini menerapkan prinsip yang sama, tetapi mengintegrasikan hasil alokasi FEFO secara langsung ke proses checkout POS. Rekomendasi batch ditampilkan kepada kasir untuk dikonfirmasi sebelum transaksi diselesaikan, sehingga keputusan pengurangan stok pada sistem dapat diselaraskan dengan batch fisik yang benar-benar diambil. Perbedaan ini menjadikan FEFO tidak hanya sebagai mekanisme pengurutan persediaan, tetapi sebagai bagian dari alur transaksi penjualan.

Jika dibandingkan dengan penelitian POS pada [11], [12], [13] dan [14], penelitian terdahulu terutama menekankan digitalisasi penjualan, kemudahan interaksi, penggunaan perangkat bergerak, atau proses pengembangan secara iteratif. Penelitian ini menggabungkan fungsi POS tersebut dengan inventori berbasis batch dan aturan FEFO. Sementara itu, penelitian [15] telah menerapkan model multi-tenancy pada aplikasi penjualan, tetapi penelitian ini menambahkan pemisahan basis data operasional per tenant serta integrasi FEFO dengan transaksi POS dalam satu rancangan sistem.

Pada aspek SaaS dan multi-tenant, hasil penelitian ini sejalan dengan [16], [17] dan [21] yang menjelaskan pemanfaatan layanan terpusat untuk mendukung operasional bisnis. Namun, berbeda dari pendekatan yang hanya menempatkan beberapa pengguna atau unit pada layanan yang sama, penelitian ini menggunakan database-per-tenant agar data produk, batch, transaksi, dan stock movement setiap unit tetap terpisah. Pemilihan tersebut konsisten dengan pembahasan [18], [19] dan [20] mengenai pentingnya isolasi data serta konsekuensi pemilihan strategi penyimpanan pada arsitektur multi-tenant.

Dibandingkan dengan penelitian terdahulu tersebut, kebaruan penelitian ini terletak pada integrasi SaaS multi-tenant, FEFO berbasis batch, POS, dan pemantauan Owner Central dalam satu alur yang tetap mempertahankan isolasi data operasional antar-tenant. Dengan demikian, penelitian ini memperluas studi inventori dan POS yang sebelumnya cenderung membahas digitalisasi stok, FEFO, transaksi penjualan, atau multi-tenancy secara terpisah menjadi satu implementasi terpadu untuk konteks UMKM bakery.

4. KESIMPULAN

Penelitian ini menghasilkan sistem inventori dan Point of Sale berbasis Software as a Service dengan arsitektur database-per-tenant yang mampu memisahkan data operasional setiap tenant sekaligus menyediakan pemantauan beberapa unit melalui Owner Central. Penerapan First Expired First Out pada inventori berbasis batch memungkinkan sistem memprioritaskan produk berdasarkan tanggal kedaluwarsa terdekat dan menggunakan tanggal penerimaan sebagai kriteria berikutnya ketika tanggal kedaluwarsa sama, serta menghubungkan hasil alokasi tersebut dengan proses checkout dan pencatatan stock movement. Hasil pengujian menunjukkan seluruh 12 kasus unit testing FEFO dan 31 skenario black-box berhasil dijalankan, sedangkan pengujian isolasi membuktikan akses lintas tenant dapat ditolak. Evaluasi terhadap 10 responden menghasilkan skor UAT rata-rata 4,4 dari 5,0 dan SUS sebesar 81,7 dari 100, yang menunjukkan bahwa sistem dapat mendukung keterlacakan stok, rotasi produk berdasarkan kedaluwarsa, penyajian laporan operasional secara real-time, dan pemantauan multi-unit tanpa mencampurkan data antar-tenant. Kontribusi penelitian ini adalah implementasi terpadu antara isolasi database-per-tenant, alokasi FEFO hingga tingkat batch yang terhubung langsung dengan checkout POS, dan Owner Central untuk pemantauan beberapa unit usaha tanpa menggabungkan data operasional masing-masing tenant. Pengembangan selanjutnya dapat difokuskan pada pengujian beban dan konkurensi untuk skala tenant yang lebih besar serta integrasi metode pembayaran non-tunai.

5. PERNYATAAN PENULIS

5.1 Kontribusi Penulis

Konseptualisasi: Rizki Wahyu, Jefry Sunupurwa Asri, Nizirwan Anwar, dan Adi Widianono;

Metodologi: Rizki Wahyu, Jefry Sunupurwa Asri, Nizirwan Anwar, dan Adi Widianono;

Perangkat Lunak: Rizki Wahyu;

Validasi: Rizki Wahyu, Jefry Sunupurwa Asri, Nizirwan Anwar, dan Adi Widianono;

Analisis Formal: Rizki Wahyu, Jefry Sunupurwa Asri, Nizirwan Anwar, dan Adi Widianono;

Investigasi: Rizki Wahyu;

Sumber Daya: Rizki Wahyu;

Kurasi Data: Rizki Wahyu;

Penulisan Draf Awal: Rizki Wahyu;

Penulisan, Peninjauan, dan Penyuntingan: Rizki Wahyu, Jefry Sunupurwa Asri, Nizirwan Anwar, dan Adi Widianono;

Visualisasi: Rizki Wahyu;

Seluruh penulis telah membaca dan menyetujui versi manuskrip yang telah diterbitkan.

5.2 Ketersediaan Data

Data yang disajikan dalam penelitian ini tersedia berdasarkan permintaan kepada penulis korespondensi.

5.3 Pendanaan

Penelitian ini tidak menerima pendanaan eksternal.

5.4 Pernyataan Dewan Peninjau Institusional

Tidak Berlaku (Not applicable).

5.5 Pernyataan Persetujuan Berdasarkan Informasi

Persetujuan berdasarkan informasi diperoleh dari seluruh responden yang berpartisipasi dalam wawancara dan evaluasi pengguna.

5.6 Pernyataan Konflik Kepentingan

Para penulis menyatakan bahwa mereka tidak memiliki kepentingan keuangan yang bersaing atau hubungan pribadi yang diketahui yang dapat dianggap memengaruhi pekerjaan yang dilaporkan dalam artikel ini.

5.7 Penggunaan AI Generatif dan Teknologi Berbantuan AI dalam Proses Penulisan

Para penulis menggunakan AI Generatif untuk meningkatkan kejelasan penulisan dalam artikel ini. Para penulis telah meninjau dan menyunting konten yang dihasilkan dengan bantuan AI serta bertanggung jawab penuh atas versi akhir publikasi.

REFERENCES

- [1] R. F. J. S. A. Christianti, "Estimasi Jarak Berbasis Convolutional Neural Network dan Regresi Dalam Sistem Deteksi Kendaraan Udara Nirawak Berbasis Suara," *Jurnal SINTA: Sistem Informasi dan Teknologi Komputasi*, vol. 2, no. 3, pp. 153–162, Aug. 2025, doi: 10.61124/sinta.v2i3.92.



- [2] K. Ismail, M. Rohmah, and D. A. P. Putri, “Peranan UMKM dalam Penguatan Ekonomi Indonesia,” *Jurnal Neraca: Jurnal Pendidikan dan Ilmu Ekonomi Akuntansi*, vol. 7, no. 2, pp. 208–217, 2023, doi: 10.31851/neraca.v7i2.14344.
- [3] M. K. Nugraha, B. A. Sekti, N. Anwar, and A. M. Widodo, “Implementasi Sistem Enterprise Resource Planning pada Industri Makanan Opak C.V. Barokah Pandeglang Menggunakan ODOO,” *IKRA-ITH Informatika: Jurnal Komputer dan Informatika*, vol. 8, no. 1, pp. 221–233, 2024, doi: 10.37817/ikraith-informatika.v8i1.3216.
- [4] F. Rahman Lutfi and C. Sasongko, “Perencanaan Produksi dan Manajemen Persediaan pada Perusahaan Kue dan Roti,” *Studi Akuntansi dan Keuangan Indonesia*, vol. 5, no. 1, pp. 61–86, 2022, doi: 10.21632/saki.5.1.61-86.
- [5] G. H. Rajagukguk and A. Widianono, “Perancangan dan Implementasi Sistem Monitoring Mesin Produksi Berbasis Lampu IoT dan Web Dashoard di PT XYZ,” *Jurnal Minfo Polgan*, vol. 14, no. 2, pp. 2303–2317, Oct. 2025, doi: 10.33395/jmp.v14i2.15413.
- [6] D. A. K. Wati and Barnad, “Desain dan Implementasi Sistem Inventory Data Ikan CV Pesona Mandiri Berbasis Google Sheets dan Appsheets,” *Teknika*, vol. 11, no. 3, pp. 163–169, 2022, doi: 10.34148/teknika.v11i3.513.
- [7] S. A. Ghiffari and I. Aryanto, “Transformasi Digital Manajemen Stok UMKM melalui Implementasi Sistem E-Inventory Mobile Real-Time,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 6, no. 1, pp. 11–22, 2026, doi: 10.57152/malcom.v6i1.2334.
- [8] P. Fahrizal and N. C. Azhar, “Perancangan Sistem Informasi Laporan Inventori Bahan Baku Berbasis Web dengan Metode Waterfall pada Kedai Kopi Disiniaja,” *INFOMATEK*, vol. 27, no. 1, pp. 153–164, 2025, doi: 10.23969/infomatek.v27i1.24838.
- [9] Y. Siyamt, “Penggunaan Metode FIFO dan FEFO dalam Mengukur Efisiensi dan Efektifitas Persediaan Obat Paten 2020–2021,” *Jurnal Ilmiah Ekonomi Islam*, vol. 8, no. 2, p. 2221, 2022, doi: 10.29040/jiei.v8i2.6041.
- [10] C. B. Herron et al., “Building ‘First Expire, First Out’ Models to Predict Food Losses at Retail Due to Cold Chain Disruption in the Last Mile,” *Front. Sustain. Food Syst.*, vol. 6, p. 1018807, 2022, doi: 10.3389/fsufs.2022.1018807.
- [11] F. A. M. Andy and S. Widiono, “Inovasi Teknologi dalam Manajemen Penjualan: Aplikasi Point of Sales Berbasis Web untuk UMKM,” *INFOMATEK*, vol. 26, no. 2, pp. 161–174, 2024, doi: 10.23969/infomatek.v26i2.19007.
- [12] M. R. Raihan and D. Hidayatullah, “Pengembangan Sistem Point Of Sale Berbasis User Centered Design,” *Jurnal Media Informatika Budidarma*, vol. 6, no. 1, p. 74, 2022, doi: 10.30865/mib.v6i1.3412.
- [13] H. B. P. Putra and S. D. Sancoko, “Penerapan Sistem Point Of Sale Berbasis Android untuk Peningkatan Kinerja Usaha,” *Infotek: Jurnal Informatika dan Teknologi*, vol. 7, no. 1, pp. 195–204, 2024, doi: 10.29408/jit.v7i1.23934.
- [14] M. Dimiyati and A. J. T. Segara, “Rancang Bangun Aplikasi Point of Sales Menggunakan Metode Extreme Programming (Studi Kasus Hadi Sport Shop),” *Jurnal Sistem dan Teknologi Informasi (JustIN)*, vol. 12, no. 3, p. 485, 2024, doi: 10.26418/justin.v12i3.78761.
- [15] A. Ahmadi and A. Surachman, “Pengembangan Aplikasi Penjualan dengan Metode Extreme Programming dan Penerapan Model Multi-Tenancy,” *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 8, no. 3, p. 303, 2024, doi: 10.30998/string.v8i3.20276.
- [16] M. H. Ratsanjani, I. F. Risnandari, T. W. Sulaiman, and V. M. Hersianty, “Literatur Review: Peran Aplikasi SaaS dalam Kegiatan Bisnis E-Commerce,” *SINOMIKA Journal: Publikasi Ilmiah Bidang Ekonomi dan Akuntansi*, vol. 1, no. 4, pp. 1009–1020, 2022, doi: 10.54443/sinomika.v1i4.491.
- [17] A. Santhosh, D. Unnikrishnan, S. Shibu, M. H. Fathima, G. Lekshmi, and S. G. Joseph, “SaaS (Software as a Service) and Its Impact on Business Scalability,” *International Research Journal on Advanced Engineering and Management (IRJAEM)*, vol. 2, no. 12, pp. 3575–3584, 2024, doi: 10.47392/irjaem.2024.0527.
- [18] S. Pushpan, “Multi-Tenant Architecture: A Comprehensive Framework for Building Scalable SaaS Applications,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 10, no. 6, pp. 1117–1126, 2024, doi: 10.32628/cseit241061151.
- [19] S. K. Pippal, S. Kumar, and R. Rani, “Optimizing Multi-Tenant Database Architecture for Efficient Software as a Service Delivery,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 5, p. 1128, 2024, doi: 10.12928/telkomnika.v22i5.26385.
- [20] R. K. Sharma, “Multi-Tenant Architectures in Modern Cloud Computing: A Technical Deep Dive,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 1, pp. 307–317, 2025, doi: 10.32628/cseit25111236.
- [21] J. Chi, Q. Ou, and M. Wang, “A Study on SaaS ERP Continuous Use Intention of Fresh Food Distribution SME,” *iBusiness*, vol. 15, no. 2, pp. 99–111, 2023, doi: 10.4236/ib.2023.152007.
- [22] F. Sinlae, E. Irwanda, Z. Maulana, and V. E. Syahputra, “Penggunaan Framework Laravel dalam Membangun Aplikasi Website Berbasis PHP,” *Jurnal Siber Multi Disiplin*, vol. 2, no. 2, pp. 119–132, 2024, doi: 10.38035/jsmd.v2i2.186.
- [23] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems*. O’Reilly Media, 2017. [Online]. Available: <https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/>
- [24] S. Newman, *Building Microservices: Designing Fine-Grained Systems*. O’Reilly Media, 2015. [Online]. Available: <https://www.oreilly.com/library/view/building-microservices/9781491950340/>
- [25] F. T. Julfia, E. Satryawati, D. Setiadi, and A. C. Umam, “Perancangan Aplikasi Warehouse Management System Berbasis Web Menggunakan Laravel dan MySQL pada PT. Sinergi Dua Akal,” *JOINED Journal (Journal of Informatics Education)*, vol. 6, no. 2, p. 206, 2024, doi: 10.31331/joined.v6i2.3036.
- [26] A. Samdono, A. P. Sari, and F. P. Aditiawan, “Pengujian Black Box pada Sistem Informasi Stok dan Penjualan Berbasis Website Menggunakan Metode Equivalence Partitioning,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 1, pp. 880–885, 2024, doi: 10.36040/jati.v8i1.8893.