

Integrasi REST API Laravel-MySQL untuk Sinkronisasi Stok Toko Sembako Mbah

Wiro

Daniar Galang Setya Putra*, Indrawan Ady Saputro
Program Studi S1 Informatika, STMIK Amikom Surakarta, Sukoharjo, Indonesia

*Corresponding author: galangdaniarr@gmail.com
(Diterima: 11 Mei 2026; Disetujui: 21 Juni 2026)

ABSTRACT

Micro, Small, and Medium Enterprises (MSMEs) urgently require an accurate information system for stock management and sales transactions. The main problem faced by the Toko Sembako Mbah Wiro grocery store, as the case study in this research, is stock discrepancy caused by conventional ledger-based manual recording. This study aims to implement a web service-based system that integrates the Laravel framework and MySQL database utilizing a REST API architecture with stateless authentication for automated data synchronization. The system development employs the linear sequential model (Waterfall), which is tailored to the internal kinship-based governance of the store. The testing results indicate that the system is capable of synchronizing stock updates in real-time through a Database Transaction mechanism, thereby eliminating recording errors due to transaction failures. Furthermore, the REST API generates structured JSON data output that is ready for direct integration into the thermal receipt printer module. The implementation of this system has proven effective in improving the efficiency of stock management and sales transactions at Toko Sembako Mbah Wiro.

Keywords: laravel, mysql, rest api, stock synchronization, sales transaction

ABSTRAK

Usaha Mikro, Kecil, dan Menengah (UMKM) membutuhkan sistem informasi yang akurat dalam pengelolaan stok dan transaksi penjualan. Masalah utama yang dihadapi oleh Toko Sembako Mbah Wiro sebagai objek studi kasus dalam penelitian ini adalah ketidakkonsistenan data stok (stock discrepancy) akibat pencatatan manual berbasis buku besar konvensional. Penelitian ini bertujuan mengimplementasikan sistem berbasis web service yang mengintegrasikan framework Laravel dan database MySQL berbasis arsitektur REST API dengan stateless authentication untuk sinkronisasi data otomatis. Pengembangan menggunakan model sekuensial linier (Waterfall) yang disesuaikan dengan tata kelola internal objek berbasis kekerabatan. Hasil pengujian menunjukkan bahwa sistem mampu melakukan sinkronisasi pembaruan stok secara real-time melalui mekanisme Database Transaction, mengeliminasi kesalahan pencatatan akibat kegagalan transaksi, serta membantu meningkatkan efisiensi pengelolaan stok dan transaksi penjualan pada Toko Sembako Mbah Wiro. Hasil data transaksi dari REST API ini menghasilkan luaran data JSON terstruktur yang siap diintegrasikan langsung ke modul pencetakan struk belanja (thermal receipt printer).

Kata Kunci: laravel, mysql, rest api, sinkronisasi stok, transaksi penjualan

PENDAHULUAN

Perkembangan teknologi informasi yang eksponensial telah mentransformasi lanskap operasional berbagai sektor industri dan bisnis. Era digitalisasi menuntut pergeseran paradigma dari manajemen konvensional menuju sistem yang terintegrasi guna mengoptimalkan efisiensi waktu serta akurasi pengolahan data. Kendati demikian, implementasi teknologi pada sektor Usaha Mikro, Kecil, dan Menengah (UMKM) di Indonesia masih menghadapi tantangan yang kompleks. Sebagian besar UMKM masih mengandalkan tata kelola internal tradisional yang berbasis kekerabatan, yang berimplikasi pada rendahnya adopsi standardisasi sistem informasi modern serta rentan terhadap kesalahan pengelolaan operasional harian [1-3].

Guna memitigasi problematika tersebut, pemanfaatan arsitektur perangkat lunak berbasis web service menjadi salah satu solusi taktis yang menawarkan fleksibilitas dan aksesibilitas tinggi [4-6]. Dalam pengembangan aplikasi web modern, penggunaan framework menjadi krusial untuk menjamin keamanan, skalabilitas, dan efisiensi waktu pengerjaan. Framework Laravel, yang mengadopsi pola arsitektur *Model-View-Controller* (MVC), menyediakan ekosistem yang andal untuk memisahkan logika bisnis dengan antarmuka pengguna [7][8]. Integritas data pada arsitektur ini diperkuat melalui adopsi *Relational Database Management System* (RDBMS) berbasis MySQL, yang mampu menangani transaksi data bervolume tinggi secara konsisten.

Toko Sembako Mbah Wiro merupakan representasi dari sektor UMKM perdagangan retail yang menghadapi kendala signifikan dalam manajemen inventaris. Aktivitas pencatatan barang

masuk, barang keluar, dan transaksi penjualan pada toko ini masih dikelola secara manual menggunakan buku besar. Mekanisme konvensional tersebut memicu terjadinya disparitas data stok (*stock discrepancy*) akibat faktor kelalaian manusia (*human error*), dokumen fisik yang hilang, serta keterlambatan pembaruan data inventaris saat volume transaksi padat. Ketidakakuratan informasi stok ini pada akhirnya menghambat proses pengambilan keputusan dan menurunkan kualitas pelayanan terhadap pelanggan.

Beberapa penelitian sebelumnya telah mengembangkan sistem informasi persediaan berbasis web untuk meningkatkan akurasi pengelolaan stok barang [9][10]. Harahap dan Herdinata mengembangkan sistem inventory berbasis web menggunakan metode Waterfall yang mampu membantu proses pencatatan stok secara lebih terstruktur [3]. Penelitian lain oleh Ramadhan dkk. mengimplementasikan framework Laravel pada sistem manajemen inventori dan aset berbasis website sehingga meningkatkan kemudahan pengelolaan data inventaris [11]. Selain itu, Herdiyatomoko mengembangkan backend berbasis REST API, menggunakan Laravel untuk mendukung pertukaran data yang lebih fleksibel antar sistem [4]. Namun, penelitian tersebut belum secara khusus mengintegrasikan sinkronisasi stok secara otomatis menggunakan REST API dan Database Transaction pada lingkungan UMKM retail [4-11]. Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem informasi web service dengan mengintegrasikan arsitektur *REpresentational State Transfer Application Programming Interface* (REST API) berbasis Laravel dan MySQL. Sistem

ini dirancang untuk melakukan sinkronisasi pembaruan stok secara otomatis melalui mekanisme Database Transaction untuk mengeliminasi risiko kegagalan parsial pada basis data. Pengembangan sistem dilakukan secara terstruktur menggunakan model sekuensial linier (*Waterfall*) yang mencakup tahapan analisis, desain, pengodean, hingga pengujian. Luaran data transaksi dari REST API ini menghasilkan format JSON terstruktur yang siap diintegrasikan langsung dengan modul pencetakan struk belanja (*thermal receipt printer*), sehingga diharapkan mampu meningkatkan efisiensi operasional dan akurasi data inventaris secara signifikan.

Penelitian ini memiliki kebaruan dibandingkan penelitian sebelumnya melalui integrasi framework Laravel, arsitektur REST API, dan mekanisme Database Transaction untuk mendukung sinkronisasi data stok secara real-time pada lingkungan UMKM. Selain meningkatkan konsistensi data transaksi, sistem yang dikembangkan juga menghasilkan keluaran data berformat JSON yang siap diintegrasikan dengan modul *thermal receipt printer*. Dengan demikian, penelitian ini tidak hanya berfokus pada pengembangan sistem informasi persediaan, tetapi juga menawarkan mekanisme sinkronisasi data yang lebih andal dan mudah diintegrasikan dengan perangkat pendukung transaksi.

METODE PENELITIAN

Penelitian ini menggunakan metode *Research and Development* (R&D) dengan pendekatan model pengembangan sistem *Waterfall* [12][13]. Pemilihan metode *Waterfall* didasarkan pada karakteristiknya yang sistematis, terstruktur, dan berurutan (*sequential*). Hal ini memastikan setiap tahapan pengembangan sistem pada Toko

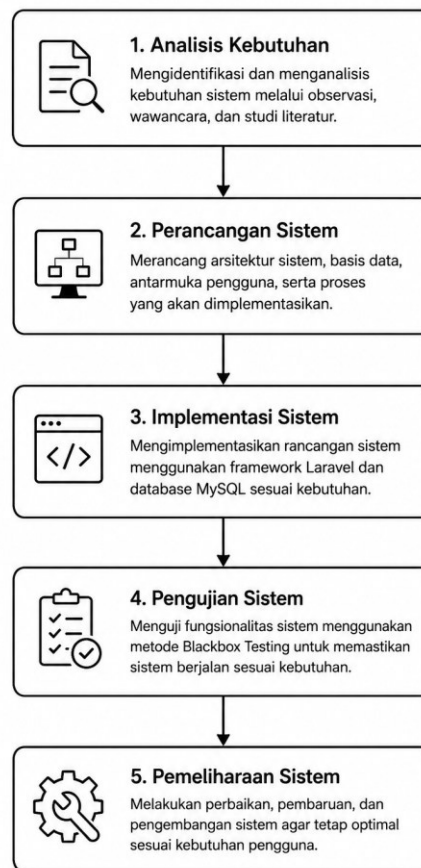
Sembako Mbah Wiro dapat terpantau dan terdokumentasi dengan baik guna meminimalisir kesalahan logika pada modul sinkronisasi data stok inventaris.

Tahapan pengembangan sistem

Tahapan rekayasa perangkat lunak dalam penelitian ini mengacu pada prosedur terstandarisasi yang meliputi empat fase utama. Pertama, analisis kebutuhan (*requirements analysis*) dilakukan untuk mengidentifikasi proses bisnis konvensional pada Toko Sembako Mbah Wiro, khususnya alur masuk barang dari supplier dan alur keluar barang retail melalui kasir. Kebutuhan fungsional yang dirumuskan mencakup kemampuan sistem dalam mengeksekusi pemotongan stok otomatis saat data transaksi baru disimpan, serta fungsi pengembalian kuantitas stok (*restock*) secara otomatis jika transaksi dibatalkan atau dihapus. Kedua, desain sistem (*system design*) mencakup perancangan arsitektur menggunakan pola *Enterprise Application Architecture* untuk memisahkan logika bisnis utama dengan lapisan basis data. Tahap ini meliputi pemodelan proses bisnis lewat Use Case Diagram, pemetaan struktur tabel relasional melalui *Entity Relationship Diagram* (ERD) berbasis MySQL, serta perancangan antarmuka pengguna (*User Interface*) halaman dashboard dan kasir. Ketiga, implementasi (*implementation*) merupakan tahap transformasi desain ke dalam bentuk kode program yang dilakukan menggunakan bahasa pemrograman PHP dengan framework Laravel. Integrasi antar-modul operasional kasir dibangun menggunakan pendekatan REST API [14],[15], menghasilkan luaran data JSON terstruktur guna memastikan skalabilitas sistem di masa depan. Keempat, pengujian (*testing*) dilakukan menggunakan metode *Blackbox Testing* [16],[17]. untuk

memvalidasi fungsionalitas seluruh fitur pada antarmuka kasir. Pengujian difokuskan pada validasi logika

sinkronisasi data stok dan memastikan tidak adanya celah tumpang-tindih data (*data race*) pada basis data saat transaksi padat.

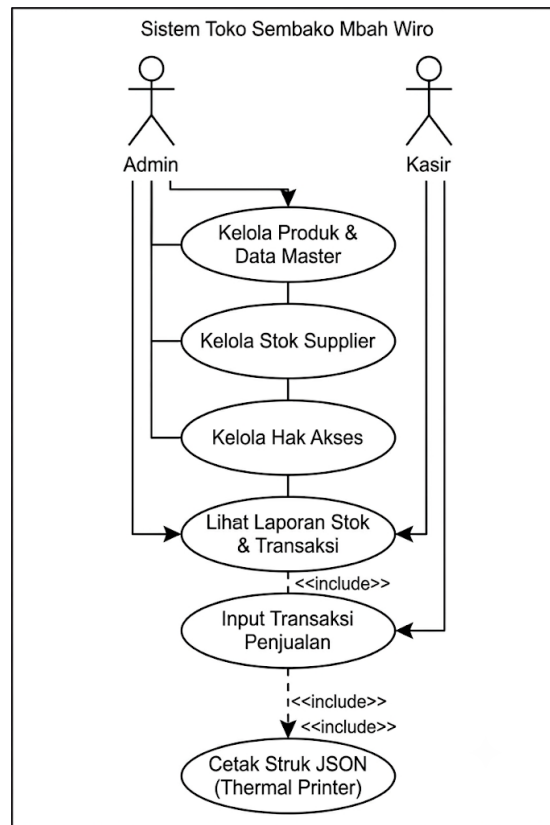


Gambar 1. Tahapan Pengembangan Sistem Menggunakan Metode Waterfall

Gambar 1 menunjukkan tahapan pengembangan sistem menggunakan metode Waterfall yang terdiri dari analisis kebutuhan, perancangan sistem, implementasi, pengujian, dan pemeliharaan. Setiap tahapan dilakukan secara berurutan untuk memastikan sistem yang dikembangkan sesuai dengan kebutuhan pengguna dan mampu mendukung sinkronisasi data stok secara optimal.

Perancangan fungsional (*use case*)

Untuk menggambarkan interaksi antara aktor (Admin dan Kasir) dengan sistem informasi yang dibangun, dirancang sebuah *Use Case Diagram*. Diagram ini memetakan batasan hak akses dan fungsionalitas utama dalam proses sinkronisasi stok dan transaksi operasional toko.

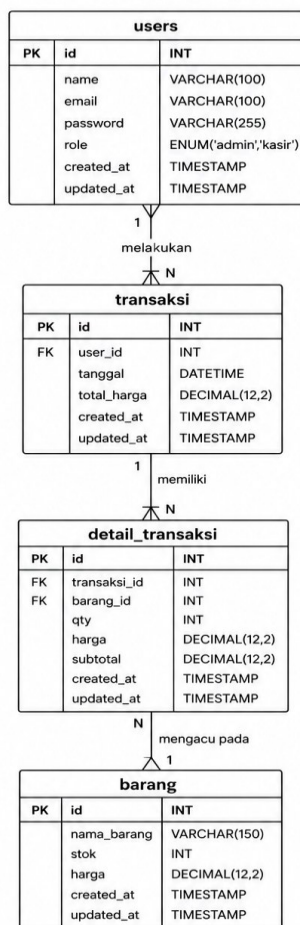


Gambar 2. *Use Case* Diagram integrasi stok dan penjualan Toko Sembako Mbah Wiro

Gambar 2 menunjukkan interaksi antara aktor admin dan kasir dengan sistem yang dikembangkan. Admin memiliki hak akses untuk mengelola data barang, pengguna, dan laporan transaksi, sedangkan kasir bertugas melakukan transaksi penjualan serta pembaruan data transaksi. Diagram ini menggambarkan batasan fungsi yang dapat diakses oleh masing-masing aktor sehingga proses pengelolaan stok dan transaksi dapat berjalan sesuai dengan hak akses yang telah ditentukan.

Selain *Use Case* Diagram, perancangan basis data dilakukan menggunakan Entity Relationship Diagram (ERD) untuk menggambarkan hubungan antar entitas yang digunakan dalam sistem. ERD digunakan sebagai acuan dalam proses implementasi database MySQL sehingga setiap data transaksi, stok barang, dan pengguna dapat terhubung secara konsisten. Struktur relasi basis data yang dirancang ditunjukkan pada Gambar 3.

**ERD Sistem Kasir dan Sinkronisasi Stok
Toko Sembako Mbah Wiro**



Gambar 3. Entity Relationship Diagram (ERD) Sistem Informasi Persediaan dan Penjualan

Gambar 3 menunjukkan hubungan antarentitas pada sistem informasi yang dikembangkan. Entitas *users* digunakan untuk mengelola data pengguna yang terdiri atas admin dan kasir. Entitas *transaksi* menyimpan data transaksi penjualan, sedangkan rincian barang yang terlibat dalam setiap transaksi disimpan pada entitas *detail_transaksi*. Selain itu, entitas *barang* digunakan untuk menyimpan informasi produk dan jumlah stok yang tersedia. Relasi antarentitas tersebut

mendukung proses sinkronisasi data stok dan transaksi secara terintegrasi pada sistem yang dibangun.

Instrumen penelitian

Instrumen yang digunakan dalam pengembangan sistem ini terdiri dari perangkat keras berupa laptop berspesifikasi standar pengembang dan instrumen perangkat lunak penunjang. Perangkat lunak utama yang mendukung jalannya penelitian ini secara rinci dipetakan pada Tabel 1.

Tabel 1. Spesifikasi lingkungan perangkat lunak (*software environment*)

No	Komponen Perangkat Lunak	Spesifikasi / Batasan Sistem
1	Sistem Operasi	Windows 11 / Linux
2	Bahasa Pemrograman	PHP v8.1 atau versi lebih tinggi
3	Web Server	Apache (XAMPP / Laravel Sail)
4	Basis Data (Database)	MySQL v8.0
5	Framework Utama	Laravel v10.x
6	Code Editor	Visual Studio Code

Alat dan bahan

Selain instrumen utama, penelitian ini menggunakan beberapa perangkat lunak spesifik untuk menjamin efisiensi dan validitas hasil pengembangan. Pengelolaan *library* dan *package* pihak ketiga pada Laravel ditangani menggunakan *dependency manager* Composer, sedangkan manajemen aset *frontend* dikelola via Node *Package Manager* (NPM) untuk memantau integritas data pada tabel MySQL secara visual dan langsung (*real-time*), digunakan *database client* seperti TablePlus, HeidiSQL, atau phpMyAdmin. Pengujian keandalan fungsi sinkronisasi stok yang dibangun melalui REST API dilakukan secara terpisah menggunakan *API testing tool* berupa Postman atau Insomnia sebelum diintegrasikan ke antarmuka kasir. Keamanan dokumentasi skrip dan pelacakan riwayat perubahan kode program dikelola melalui *Version Control System* (VCS) berbasis Git dengan repositori GitHub. Proses pelacakan kesalahan (*debugging data*) selama pengerjaan didukung penuh oleh pemanfaatan ekstensi *developer tools* pada web browser Google Chrome atau Mozilla Firefox serta memanfaatkan fitur Laravel Telescope.

HASIL DAN PEMBAHASAN

Pengembangan sistem informasi pada Toko Sembako Mbah Wiro telah berhasil diimplementasikan sesuai dengan analisis kebutuhan yang dirumuskan.

Penyimpanan data fungsional dikelola melalui basis data relasional MySQL v8.0 dengan arsitektur tabel yang saling berelasi guna menjamin konsistensi informasi secara *real-time*. Struktur basis data yang dibangun terdiri dari entitas utama meliputi tabel *user* untuk manajemen autentikasi hak akses, tabel *barang* untuk merekam kuantitas stok produk, tabel *transaksi* untuk mencatat data utama nota penjualan, serta tabel *detail transaksi* sebagai tabel perantara (*junction table*) yang mencatat rincian kuantitas produk yang terjual dalam setiap transaksi.

Mekanisme sinkronisasi data stok barang dilakukan pada lapisan *Controller* di dalam framework Laravel dengan mengimplementasikan fungsi *Database Transactions (DB::transaction)*. Alur kerja dari fitur ini dimulai ketika kasir mengeksekusi finalisasi pembayaran pada antarmuka. Sistem secara otomatis membuka sesi transaksi basis data, menyimpan record ke tabel *Database Transactions (DB::transaction)* mengiterasi setiap item ke tabel *detail_transaksi*, dan secara simultan melakukan operasi pengurangan (*decrement*) kuantitas stok pada tabel *barang*. Penggunaan *Database Transactions* ini bertujuan untuk menjamin sifat *Atomicity* (prinsip *ACID* dalam basis data), di mana seluruh rangkaian perintah manipulasi data harus berhasil sepenuhnya atau gagal sama sekali (*all or nothing*). Fenomena ilmiah yang sering terjadi pada

aplikasi kasir konvensional tanpa enkapsulasi transaksi adalah munculnya tumpang-tindih data (*data race*) atau inkonsistensi stok ketika koneksi server terputus di tengah proses pemrosesan. Melalui pendekatan ini, apabila terjadi kegagalan pemrosesan data di tengah jalan, sistem secara otomatis memicu perintah

rollback untuk membatalkan seluruh perubahan. Hal tersebut efektif mencegah terjadinya transaksi hantu (*phantom read*) maupun selisih pencatatan stok fisik di gudang. Skrip pemrograman PHP pada *TransactionController* yang menangani fungsi pengurangan stok otomatis secara ringkas disajikan pada Gambar 4.

```
TransactionController.php <- X
TransactionController.php > TransactionController > store
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use Illuminate\Support\Facades\DB;
7 use App\Models\Barang;
8 use App\Models\Transaksi;
9 use App\Models\DetailTransaksi;
10
11 class TransactionController extends Controller
12 {
13     // Windows Refactor | Explain | Generate Function Comment | X
14     public function store(Request $request)
15     {
16         // Menggunakan Database Transactions untuk menjamin prinsip ACID
17         DB::beginTransaction();
18
19         try {
20             // 1. Simpan data utama ke tabel transaksi
21             $transaksi = Transaksi::create([
22                 'nomor_nota' => 'NOTA-' . time(),
23                 'total_harga' => $request->total_harga,
24                 'kasir_id' => auth()->id(),
25             ]);
26
27             // 2. Iterasi item belanja dan sinkronisasi stok ke tabel barang
28             foreach ($request->items as $item) {
29                 // Simpan rincian ke tabel detail_transaksi
30                 $detail_transaksi = DetailTransaksi::create([
31                     'transaksi_id' => $transaksi->id,
32                     'barang_id' => $item['barang_id'],
33                     'qty' => $item['qty'],
34                     'subtotal' => $item['subtotal'],
35                 ]);
36
37                 // Proses sinkronisasi potong stok otomatis (Decrement)
38                 $barang = Barang::findOrFail($item['barang_id']);
39
40                 if ($barang->stok < $item['qty']) {
41                     throw new \Exception("Stok barang tidak mencukupi.");
42                 }
43
44                 $barang->decrement('stok', $item['qty']);
45             }
46
47             // Jika semua proses berhasil tanpa error, commit data ke database
48             DB::commit();
49
50             return response()->json([
51                 'success' => true,
52                 'message' => 'Transaksi berhasil dan stok telah disinkronisasi.',
53             ], 200);
54         } catch (\Exception $e) {
55             // Jika terjadi kegagalan/koneksi terputus, batalkan semua perubahan (Rollback)
56             DB::rollback();
57
58             return response()->json([
59                 'success' => false,
60                 'message' => 'Transaksi gagal, database di-rollback: ' . $e->getMessage(),
61             ], 500);
62         }
63     }
64 }
```

Gambar 4. Kode program sinkronisasi stok otomatis menggunakan Database Transactions di Laravel

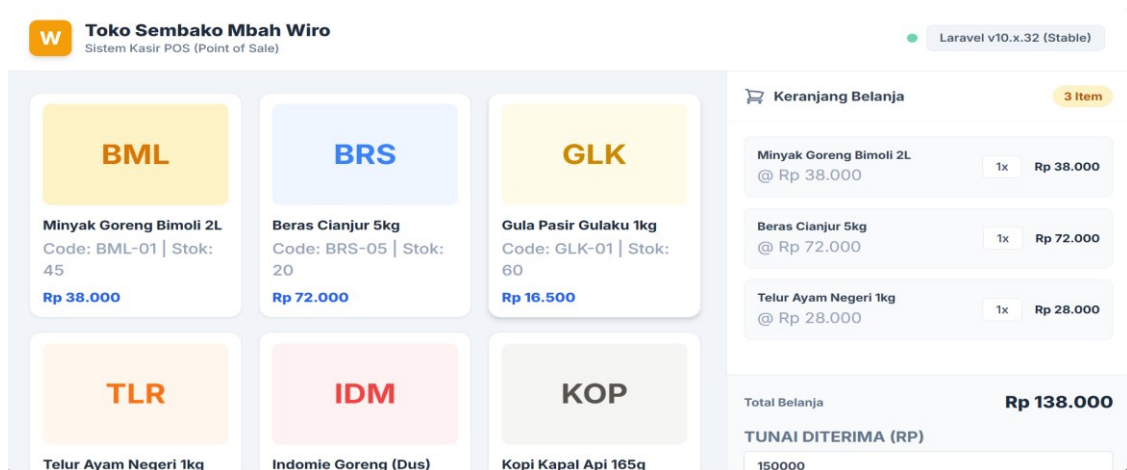
Gambar 4. menunjukkan implementasi fungsi Database Transaction pada framework Laravel yang digunakan untuk mengelola proses penyimpanan transaksi dan pembaruan stok barang secara bersamaan. Melalui mekanisme tersebut, seluruh proses akan dieksekusi sebagai satu kesatuan transaksi sehingga apabila terjadi kegagalan pada salah satu proses, sistem

secara otomatis melakukan rollback untuk menjaga konsistensi data.

Halaman antarmuka penjualan (*Point of Sale*) dirancang minimalis guna mendukung efisiensi operasional kasir Toko Sembako Mbah Wiro. Ketika pengguna memasukkan item belanja, komponen frontend menangkap data tersebut dan mengirimkannya ke server melalui arsitektur REST API dengan luaran

format JSON (*JavaScript Object Notation*). Penggunaan JSON dipilih karena strukturnya yang berbasis teks dan ringan, sehingga mampu mempercepat waktu respon (*response time*) server dalam

memproses pemotongan stok dibandingkan dengan format XML tradisional. Tampilan visual dari halaman kasir yang telah berhasil diimplementasikan ditunjukkan pada Gambar 5.



Gambar 5. Antarmuka halaman transaksi kasir (Point of Sale)

Gambar 5. menunjukkan tampilan antarmuka transaksi penjualan yang digunakan oleh kasir. Halaman ini menyediakan fitur pemilihan barang, penginputan jumlah pembelian, perhitungan total transaksi, serta proses penyimpanan data penjualan. Melalui integrasi REST API, setiap transaksi yang berhasil disimpan akan secara otomatis memperbarui data stok pada basis data sehingga informasi persediaan dapat terjaga secara real-time.

Guna memastikan integrasi antara aplikasi Laravel dan basis data MySQL berjalan dengan baik, dilakukan pengujian terhadap endpoint REST API yang digunakan dalam proses autentikasi, pengelolaan data barang, dan transaksi penjualan. Pengujian dilakukan menggunakan aplikasi Postman dengan memverifikasi keberhasilan respon sistem serta kesesuaian format data JSON yang dikirimkan maupun diterima oleh server. Hasil pengujian endpoint tersebut disajikan pada Tabel 2.

Tabel 2. Hasil Pengujian *Endpoint REST API*

No	Endpoint	Method	Fungsi	Status
1	/login	POST	Autentikasi pengguna	Berhasil
2	/barang	GET	Menampilkan data barang	Berhasil
3	/transaksi	POST	Menyimpan transaksi penjualan	Berhasil
4	/transaksi/{id}	DELETE	Membatalkan transaksi dan mengembalikan stok	Berhasil

Berdasarkan hasil pengujian yang dilakukan, seluruh endpoint REST API berhasil dijalankan sesuai fungsi yang dirancang. Setiap permintaan (request) yang dikirimkan oleh aplikasi memperoleh respon (response) yang valid dalam format JSON dan mampu berinteraksi dengan basis data MySQL tanpa mengalami kegagalan transaksi. Hasil tersebut menunjukkan bahwa mekanisme integrasi REST API yang diterapkan telah berjalan dengan baik

dan mendukung sinkronisasi data stok secara real-time.

Validasi fungsional terhadap fitur-fitur krusial di dalam sistem informasi dilakukan menggunakan metode *Blackbox Testing*. Pengujian difokuskan pada keselarasan antara aksi pengguna pada antarmuka dengan respon pembaruan data stok di dalam basis data MySQL. Hasil pengujian fungsionalitas disajikan secara rinci pada Tabel 3.

Tabel 3. Hasil pengujian fungsionalitas sistem (*blackbox testing*)

No	Fitur Fungsi yang Diuji	Hasil yang Diharapkan	Status
1	Autentikasi Login Akun	Admin dan Kasir dapat masuk ke dashboard sesuai hak akses masing-masing.	Sukses
2	Input Transaksi Kasir	Sistem berhasil menyimpan data nota transaksi baru ke tabel MySQL.	Sukses
3	Sinkronisasi Potong Stok	Kuantitas stok produk otomatis berkurang setelah transaksi penjualan berhasil disimpan.	Sukses
4	Pembatalan Transaksi (<i>Restock</i>)	Jumlah stok produk otomatis bertambah kembali jika transaksi dibatalkan atau dihapus.	Sukses

Berdasarkan hasil pengujian Black Box yang telah dilakukan, seluruh fitur utama sistem berhasil berjalan sesuai dengan kebutuhan pengguna. Fungsi autentikasi, transaksi penjualan, sinkronisasi stok, dan pembatalan transaksi dapat dieksekusi tanpa ditemukan kesalahan fungsional. Hasil penelitian ini sejalan dengan penelitian Harahap dan Herdinata yang menunjukkan bahwa sistem inventory berbasis web mampu meningkatkan akurasi pencatatan stok barang [3]. Selain itu, implementasi Laravel pada penelitian ini juga mendukung temuan

Ramadhan dkk [11]. bahwa framework Laravel mampu mempercepat proses pengembangan sistem inventaris berbasis website. Perbedaan utama penelitian ini terletak pada penerapan REST API dan Database Transaction yang digunakan untuk menjamin sinkronisasi data stok secara real-time.

Penerapan arsitektur web berbasis Laravel dan MySQL terbukti mampu menyelesaikan hambatan operasional pencatatan stok manual pada Toko Sembako Mbah Wiro. Mekanisme sinkronisasi otomatis via REST API

berhasil memangkas waktu pembaruan data inventaris secara signifikan dari yang sebelumnya membutuhkan waktu harian menjadi instan (hitungan milidetik) sesaat setelah tombol simpan ditekan. Temuan penelitian ini sejalan dengan hasil penelitian terdahulu yang menyatakan bahwa integrasi REST API pada sistem inventaris retail mampu meminimalisir kesalahan pencatatan stok akibat proses pencatatan manual [14],[5]. Selain itu, penggunaan fitur *Database Transactions* untuk mengamankan stok otomatis ini juga membuktikan bahwa enkapsulasi data transaksi menggunakan *framework* modern mampu mencegah kerusakan integritas data (*data corruption*) pada basis data UMKM saat menghadapi lonjakan transaksi yang padat secara bersamaan. Dengan demikian, solusi digitalisasi yang diterapkan pada penelitian ini memberikan kontribusi nyata dalam mempercepat perputaran informasi stok dan meningkatkan akurasi data finansial secara efektif.

KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada sistem informasi kasir berbasis web (Point of Sale) di Toko Sembako Mbah Wiro, dapat disimpulkan bahwa penerapan arsitektur REST API berformat JSON (JavaScript Object Notation) pada framework Laravel yang dikombinasikan dengan fitur Database Transactions (DB::transaction) berhasil menyelesaikan permasalahan ketidaksinkronan pencatatan stok barang secara instan dan aman. Mekanisme enkapsulasi ini menjamin prinsip Atomicity (prinsip ACID) pada database MySQL berjalan sempurna, sehingga mencegah terjadinya tumpang-tindih data (data race), selisih pencatatan inventaris fisik, maupun

munculnya transaksi hantu (phantom read) saat server mengalami gangguan koneksi di tengah proses. Keandalan solusi digital ini diperkuat oleh hasil pengujian fungsionalitas menggunakan metode Blackbox Testing yang menunjukkan tingkat keberhasilan sebesar 100% pada seluruh skenario utama, membuktikan bahwa sistem ini valid dan siap diimplementasikan untuk mendukung efisiensi serta akurasi operasional transaksi pada sektor retail UMKM.

DAFTAR PUSTAKA

- [1] H. Elmi, A. Ambiyar, N. Jalinus, R. Abdullah, dan F. Fadhillah, "Hubungan Ketersediaan Layanan Informasi Dan Kepuasan Pengguna Terhadap Kualitas Sistem Informasi," *J. SAINTIKOM J. Sains Manaj. Inform. Dan Komput.*, vol. 22, no. 1, hlm. 81, Feb 2023, doi: 10.53513/jis.v22i1.7570.
- [2] Muhammad Afif Dzaky Khairullah, E. Rohaini, dan Yovi Pratama, "Perancangan Sistem Informasi Persediaan Pada Reklame Afia Berbasis Web," *J. Inform. Dan Rekayasa Komputer JAKAKOM*, vol. 3, no. 2, hlm. 581–590, Sep 2023, doi: 10.33998/jakakom.2023.3.2.851.
- [3] R. M. A. Harahap dan R. S. Herdinata, "Sistem Informasi Inventory Barang Berbasis Web Dengan Menggunakan metode Waterfall Studi Kasus (Butik Lubis Collection)," *J-SISKO TECH J. Teknol. Sist. Inf. Dan Sist. Komput. TGD*, vol. 6, no. 2, hlm. 500, Jul 2023, doi: 10.53513/jsk.v6i2.8226.
- [4] H. F. Herdiyatomoko, "Desain Sistem Backend Berbasis Rest Api Menggunakan Framework Laravel 7," *SKANIKA*, vol. 5, no. 2, hlm. 136–144,

- Jul 2022, doi: 10.36080/skanika.v5i2.2947.
- [5] K. Gowell dan Supriyadi, “Perancangan Web Service REST API Menggunakan PHP dan Framework Laravel di Tenta Tour Salatiga,” *J. JTIK J. Teknol. Inf. Dan Komun.*, vol. 8, no. 1, hlm. 49–57, Jan 2024, doi: 10.35870/jtik.v8i1.1269.
- [6] B. Baharuddin, H. Wakkang, dan B. Irianto, “Implementasi Web Service Dengan Metode Rest Api Untuk Integrasi Data Covid 19 Di Sulawesi Selatan,” *J. Sintaks Log.*, vol. 2, no. 1, hlm. 236–241, Feb 2022, doi: 10.31850/jsilog.v2i1.1035.
- [7] D. Purnama Sari dan R. Wijanarko, “Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus di Rumah Kamera Semarang),” *J. Inform. Dan Rekayasa Perangkat Lunak*, vol. 2, no. 1, hlm. 32, Mar 2020, doi: 10.36499/jinrpl.v2i1.3190.
- [8] U. Ma’sum, D. Rahmawati, dan S. Muryanah, “Rancang Bangun Sistem Informasi E-Gudang Pada PT.XYZ Berbasis Website Dengan Framework Laravel,” *J. Ilm. Fak. Tek.*, vol. 3, no. 1, hlm. 51–61, Okt 2023, doi: 10.33592/jimtek.v3i1.3793.
- [9] “Sistem Inventori Persediaan Produk Berbasis Web di Toko Nanik Tri Menggunakan Framework Laravel,” *J. Ilm. Komputasi*, vol. 24, no. 3, Sep 2025, doi: 10.32409/jikstik.24.3.3834.
- [10] A. B. Saputra dan J. Kasih, “Rancang Bangun Sistem Informasi Persediaan Barang Menggunakan Framework Laravel dengan Metode Waterfall: Studi Kasus Toko Amora Computer,” *DBESTI J. Digit. Bus. Technol. Innov.*, vol. 3, no. 1, hlm. 108–115, Mei 2026, doi: 10.54914/dbesti.v3i1.2088.
- [11] A. I. Ramadhan dan P. A. R. Devi, “Implementasi Sistem Manajemen Inventori dan Aset REA TEHNIK Berbasis Web Menggunakan Framework Laravel,” *J. Ilm. Sist. Inf.*, vol. 5, no. 2, hlm. 416–428, Apr 2026, doi: 10.51903/mpwjt966.
- [12] Y. E. Achyani dan S. Saumi, “Penerapan Metode Waterfall Pada Sistem Informasi Manajemen Buku Perpustakaan Berbasis Web,” *J. SAINTEKOM*, vol. 9, no. 1, hlm. 83–94, Mar 2019, doi: 10.33020/saintekom.v9i1.84.
- [13] S. N. Fatonatur Rohma dan O. A. Opoku, “Design Get Up System Information Inventory Goods Based Web Using the Waterfall Method at CV. Tekno Maju Jaya,” *J. Bus. Soc. Technol.*, vol. 6, no. 2, hlm. 93–101, Feb 2026, doi: 10.59261/jbt.v6i2.567.
- [14] H. F. Herdiytmoko dan Y. D. Pratama, “Rest Api Pada Toko Kelontong Untuk Transaksi Penjualan Menggunakan Framework Laravel,” *IDEALIS Indones. J. Inf. Syst.*, vol. 7, no. 1, hlm. 118–127, Jan 2024, doi: 10.36080/idealis.v7i1.3113.
- [15] L. S. Saputra dan H. Fery Herdiytmoko, “Implementasi Rest Api Pada Sistem Pendataan Kapal Menggunakan Laravel 9 Dan Nuxtjs,” *J. Comput. Technol.*, vol. 2, no. 1, hlm. 10–21, Jul 2024, doi: 10.69916/comtechno.v2i1.138.
- [16] S. Nidhra, “Black Box and White Box Testing Techniques - A Literature Review,” *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, hlm. 29–50, Jun 2012, doi: 10.5121/ijesa.2012.2204.
- [17] S. Saifudin, “Penerapan Model Waterfall Dan Blackbox Testing Dalam

Pembuatan Sistem Informasi E-Aset
Desa Berbasis Web,” *Inform. Comput.*

Eng. J., vol. 1, no. 2, hlm. 14–18, Jul
2021, doi: 10.31294/icej.v1i2.445.