

Implementasi Sistem Penjualan Berbasis Web Menggunakan PHP Native, MySQL, dan Vanilla JavaScript pada UMKM Kuliner Dapur Rumpin

Billie Raditya Akbar¹, Fathya Vaulizayanti², Jupron³

^{1,2,3}Teknik Informatika, Fakultas Teknik, Universitas Pamulang

billierdtya@gmail.com, fathyavau@gmail.com, dosen02664@unpam.ac.id

Abstract

Dapur Rumpin, a culinary business unit in Rumpin District, Bogor Regency, faced serious operational obstacles stemming from a fully conventional transaction recording system, covering manual order writing in notebooks, cashier calculations without digital tools, and the absence of any organized technology-based ordering channel. This condition triggered potential data recording errors while simultaneously hindering the owner's capacity to make real-time data-driven decisions. This research designed and implemented a web-based sales system with a Client-Server architecture, utilizing PHP Native as the backend layer, MySQL as the relational database management system for transaction history persistence, and Vanilla JavaScript ES6 as the client-side interactivity engine, with a primary focus on optimizing transaction efficiency for culinary micro, small, and medium enterprises. Development was carried out through the Waterfall model encompassing four sequential stages, namely requirements gathering, architecture design, code implementation, and functional verification. The system delivers a digital menu catalog with dynamic category filtering, an interactive shopping cart powered by localStorage, an asynchronous checkout mechanism via Fetch API that stores order data into MySQL through PDO Transaction, static QRIS payment code display, and automated order confirmation notifications to the admin via WhatsApp URL Scheme API. Black Box Testing evaluation across five functional scenarios confirmed that all core functions operated without issues, while Google Lighthouse testing recorded perfect Best Practices and SEO scores of 100, with First Contentful Paint of 1.8 seconds on mobile and 1.4 seconds on desktop. This system successfully replaced manual workflows with an automated digital solution that guarantees transaction persistence to MySQL while significantly reducing human error, proving that the combination of PHP Native, MySQL, and Vanilla JavaScript is a practical, scalable, and cost-efficient digitalization solution for culinary SMEs.

Keywords: Web-Based Sales System, PHP Native, MySQL, Vanilla JavaScript, localStorage, WhatsApp URL Scheme, QRIS, SME Digitalization

Abstrak

Dapur Rumpin, unit usaha kuliner di Kecamatan Rumpin, Kabupaten Bogor, menghadapi hambatan operasional serius akibat sistem pencatatan transaksi yang masih sepenuhnya konvensional, mencakup penulisan pesanan manual di buku nota, penghitungan tagihan tanpa perangkat digital, serta ketiadaan jalur pemesanan berbasis teknologi terorganisir. Kondisi ini memicu potensi kesalahan data sekaligus menghambat pengambilan keputusan secara real-time. Riset ini merancang serta mengimplementasikan sistem penjualan berbasis web dengan arsitektur Client-Server, memanfaatkan PHP Native sebagai backend, MySQL sebagai manajemen basis data relasional, dan Vanilla JavaScript ES6 sebagai mesin interaktivitas sisi klien, dengan fokus utama pada

optimalisasi efisiensi transaksi UMKM kuliner. Pengembangan dilakukan melalui model Waterfall yang mencakup empat tahap berurutan yaitu penggalan kebutuhan, perancangan arsitektur, realisasi kode, dan verifikasi fungsional. Sistem yang dibangun menghadirkan katalog menu digital dengan penyaringan kategori dinamis, keranjang belanja interaktif berbasis localStorage, mekanisme checkout asinkron melalui Fetch API yang menyimpan data ke MySQL via PDO Transaction, tampilan kode QRIS statis, serta notifikasi otomatis ke admin lewat WhatsApp URL Scheme API. Evaluasi Black Box Testing pada lima skenario fungsional memperlihatkan seluruh fungsi berjalan sempurna, sementara pengujian Google Lighthouse mencatat skor Best Practices dan SEO di angka 100, dengan First Contentful Paint 1,8 detik pada mobile dan 1,4 detik pada desktop. Sistem ini berhasil menggantikan alur manual dengan solusi digital terautomasi yang menjamin persistensi transaksi ke MySQL sekaligus menekan kesalahan manusia secara signifikan, sehingga kombinasi PHP Native, MySQL, dan Vanilla JavaScript terbukti menjadi solusi digitalisasi UMKM kuliner yang praktis, terukur, dan hemat biaya.

Kata kunci: sistem penjualan berbasis web, PHP Native, MySQL, Vanilla JavaScript, localStorage, WhatsApp URL Scheme, QRIS, digitalisasi UMKM

© 2026 Author

Creative Commons Attribution 4.0 International License



1. Pendahuluan

Laju kemajuan teknologi informasi terus mendorong gelombang transformasi digital yang menjangkau hampir seluruh lini sektor usaha, tak terkecuali industri kuliner yang bergerak dalam skala UMKM [1]. Di Indonesia, tingkat penetrasi internet dilaporkan telah mencapai 78,19% dari keseluruhan populasi (APJII, 2024), namun perkembangan itu belum sepenuhnya diikuti oleh meluasnya adopsi sistem berbasis digital di kalangan pelaku usaha kecil [2]. Data yang dirilis Kementerian Koperasi dan UKM pada tahun 2023 mengungkapkan bahwa dari total 65,5 juta unit UMKM yang tersebar di seluruh penjuru Indonesia, hanya sekitar 21% yang sudah memanfaatkan sistem informasi digital dalam menjalankan operasional bisnisnya [3]. Kesenjangan inilah yang mengakibatkan sebagian besar UMKM kuliner, termasuk Dapur Rumpin di Kecamatan Rumpin, Kabupaten Bogor, masih bertumpu pada pencatatan transaksi secara manual memakai buku nota dan lembaran kertas, sehingga proses rekapitulasi data menjadi lamban, rawan kekeliruan, dan tidak mampu menopang pengambilan keputusan bisnis berbasis data secara *real-time* [4].

Sejumlah kajian terdahulu telah menelaah persoalan digitalisasi sistem penjualan UMKM dari sudut pandang yang beragam. Surahmat dkk. (2025) menerapkan sistem informasi penjualan berbasis web pada UMKM kuliner dan berhasil membuktikan bahwa platform tersebut secara efektif mampu mengoptimalkan efisiensi pelayanan transaksi dan meminimalkan antrean pelanggan [5]. Rahmi et al. (2023) melalui pendekatan *systematic literature review* menyimpulkan bahwa metode pengembangan berbasis *web* seperti Waterfall terbukti efektif dan terstruktur dalam menghasilkan sistem informasi

yang sesuai dengan kebutuhan pengguna [6]. Lebih lanjut, penerapan aplikasi pemesanan berbasis web telah terbukti mampu menjadi strategi digitalisasi yang efektif bagi UMKM berskala lokal untuk mempermudah transaksi dan memperluas jangkauan operasionalnya [7]. Dari dimensi integrasi pembayaran digital, Habibirrahman et al. (2022) mendemonstrasikan bahwa penerapan *payment gateway* pada sistem pemesanan berbasis *web* secara nyata meningkatkan kemudahan bertransaksi bagi pelanggan UMKM [8]. Terkait aspek komunikasi dan konfirmasi digital, Sutanto et al. (2022) membuktikan bahwa pendayagunaan *WhatsApp Business* dalam proses digitalisasi pemesanan produk UMKM efektif dalam mendongkrak efisiensi komunikasi antara penjual dan pembeli [9]. Di sisi lain, Narulita et al. (2024) memperlihatkan bahwa pemodelan sistem menggunakan diagram UML, yang meliputi *use case* dan *flowchart*, memiliki peran krusial dalam merancang alur kerja sistem informasi yang kompleks secara sistematis dan mudah dicerna [10].

Meskipun penelitian-penelitian di atas telah memberikan sumbangan yang cukup berarti, masih terdapat celah yang belum terisi: belum ada studi yang secara spesifik mengintegrasikan *HTML5*, *CSS3*, *JavaScript Vanilla*, *Web Storage API (localStorage)*, QRIS, dan *WhatsApp URL Scheme* ke dalam satu ekosistem solusi transaksi digital yang komprehensif dan mandiri untuk UMKM kuliner lokal, tanpa mengandalkan *framework backend* maupun layanan berbiaya [11]. Kebaruan penelitian ini terletak pada integrasi arsitektur *Client-Server* berbasis *PHP Native* dan *MySQL* dengan mekanisme interaktivitas *Vanilla JavaScript* di sisi klien, pembayaran QRIS, dan konfirmasi pesanan otomatis melalui *WhatsApp URL Scheme*, membentuk solusi

end-to-end yang ringan, persisten, dan berpotensi direplikasi oleh UMKM sejenis.

Bertolak dari rumusan masalah dan celah penelitian tersebut, studi ini bertujuan merancang dan mengimplementasikan sistem penjualan berbasis *web* dengan arsitektur *Client-Server* pada Dapur Rumpin, memanfaatkan *PHP Native* sebagai *backend*, *MySQL* sebagai basis data relasional, serta *Vanilla JavaScript* sebagai mesin interaktivitas sisi klien, yang mencakup fitur katalog menu digital dengan kemampuan filter dinamis, manajemen keranjang belanja berbasis *localStorage*, *checkout* asinkron via *Fetch API*, pembayaran via QRIS, serta konfirmasi pesanan otomatis melalui *WhatsApp URL Scheme API*, sebagai wujud solusi digitalisasi yang praktis, persisten, dan efektif bagi pelaku UMKM kuliner [12]i.

2. Metode Penelitian

Penelitian ini mengadopsi model pengembangan perangkat lunak *Waterfall* yang dijalankan secara berurutan melalui empat tahapan: analisis kebutuhan, perancangan sistem, implementasi, dan pengujian [3]. Pengumpulan data dilaksanakan melalui tiga jalur: observasi langsung di lokasi Dapur Rumpin selama periode Kerja Praktik berlangsung guna memetakan alur transaksi manual yang sedang berjalan, wawancara terstruktur dengan pemilik usaha dan kasir untuk menggali kebutuhan fitur sistem secara mendalam, serta penelusuran literatur ilmiah yang relevan dengan topik penelitian.

2.1 Analisis Kebutuhan

Analisis difokuskan pada keseluruhan proses bisnis penjualan Dapur Rumpin yang meliputi pencatatan pesanan, kalkulasi tagihan, dan pengelolaan transaksi harian. Berdasarkan temuan dari observasi lapangan dan serangkaian sesi wawancara, teridentifikasi lima permasalahan pokok yang menghambat operasional: (1) pesanan dicatat secara manual menggunakan buku nota, (2) sistem rekapitulasi transaksi tidak berjalan secara otomatis, (3) pelanggan tidak memiliki kanal pemesanan mandiri yang dapat diakses secara langsung, (4) seluruh proses pembayaran bergantung penuh pada interaksi tatap muka dengan kasir, dan (5) informasi menu tidak tersedia dalam format digital yang mudah dijangkau. Kelima temuan ini kemudian menjadi landasan utama dalam perancangan fitur-fitur sistem yang akan dibangun.

2.2 Perancangan Sistem

Perancangan sistem menggunakan pendekatan *UML* (*Unified Modeling Language*) yang menghasilkan

dua artefak pemodelan utama [6]. Artefak pertama berupa *Activity Diagram* sistem berjalan yang mendokumentasikan 10 tahap alur transaksi manual, dimulai dari momen pelanggan menatap daftar menu fisik hingga kasir membubuhkan catatan akhir pada buku kas. Artefak kedua adalah *Activity Diagram* sistem usulan yang menggambarkan 12 tahap alur digital, berawal dari pelanggan membuka tautan *website* hingga keranjang belanja terhapus otomatis dari *localStorage* sesaat setelah konfirmasi *WhatsApp* berhasil terkirim. *Use Case Diagram* dirancang dengan melibatkan dua aktor: Pelanggan sebagai pengguna aktif antarmuka pemesanan dan Admin Dapur sebagai pihak penerima notifikasi konfirmasi pesanan melalui *WhatsApp*.

Struktur data sistem dirancang menggunakan format *JSON* yang disimpan pada *localStorage* browser dengan kunci *drCart*. Setiap objek item pesanan memiliki tiga atribut: nama (*string*), harga (*integer*), dan *qty* (*integer*). Proses normalisasi data dilakukan secara konseptual, dimulai dari bentuk *Unnormalized Form* (*UNF*) dan berlanjut hingga *Third Normal Form* (*3NF*), menghasilkan empat entitas logis yaitu Menu, Kategori, Pesanan, dan Detail_Pesanan, yang menjadi fondasi struktural basis data *MySQL* pada arsitektur *Client-Server* yang diimplementasikan dalam penelitian ini.

Hasil perancangan antarmuka menghadirkan empat komponen layar utama: halaman katalog menu, modal detail produk, panel keranjang belanja (*side-drawer*), dan modal *checkout* dua tahap. Sementara itu, alur *checkout* beserta integrasi *WhatsApp* dirancang dalam tiga sekuens: pengisian data identitas pemesan, penayangan kode *QRIS*, dan pengiriman konfirmasi pesanan secara otomatis melalui *WhatsApp URL Scheme API* ke nomor admin.

2.3 Implementasi

Sistem dibangun menggunakan arsitektur *Client-Server* dengan *PHP Native* dan *MySQL* di sisi *backend*, serta *HTML5*, *CSS3*, dan *Vanilla JavaScript ES6* di sisi klien. *HTML5* difungsikan untuk menyusun kerangka dokumen, dengan atribut *data-kat* yang disematkan pada setiap elemen kartu menu sebagai penanda kategori produk yang dibutuhkan untuk operasi filter dinamis. *CSS3* diterapkan guna mengatur tata letak responsif menggunakan *Flexbox* dan *Media Query*, berikut efek transformasi *translateX* pada komponen *side-drawer* serta transisi *opacity* pada modal *overlay*. *JavaScript Vanilla ES6* digunakan untuk mengelola

seluruh logika interaktif sistem, yang mencakup enam fungsi utama yang saling terintegrasi sebagaimana tergambar pada Gambar 1.

Manajemen state keranjang belanja dibangun di atas mekanisme CRUD terhadap *localStorage*. Algoritma penambahan item bekerja dengan memeriksa keberadaan produk berdasarkan atribut nama di dalam array *drCart*; apabila ditemukan kecocokan, nilai *qty* diperbarui secara langsung, sedangkan jika tidak ditemukan maka objek baru disisipkan ke dalam array melalui metode *push()*. Kalkulasi total pembayaran dikerjakan menggunakan akumulasi iteratif sebagaimana dirumuskan dalam persamaan (1).

di mana *T* adalah total pembayaran, merupakan harga satuan item ke-*i*, merupakan jumlah (*qty*) item ke-*i*, dan *n* adalah banyaknya jenis item dalam keranjang.

Integrasi konfirmasi pesanan memanfaatkan *WhatsApp URL Scheme* dengan format tautan berikut:

https://api.whatsapp.com/send?phone=6285643117734&text=%EF%BF%BD+%2APESANAN+DAPUR+RUMPIN%2A%0A%0A%EF%BF%BD+Nama+%3A+Putra%0A%EF%BF%BD+Meja+++%3A+7%0A%0A%2ADetail+Pesanan%3A%2A%0A%E2%80%A2+Nasi+Goreng+Sereh+x2+%3D+Rp+48.000%0A%E2%80%A2+Iced+Lemon+Tea+x2+%3D+Rp+20.000%0A%0A%EF%BF%BD+%2ATotal%3A+Rp+68.000%2A%0A%0A%EF%BF%BD+Saya+sudah+melakukan+pembayaran+via+QRIS.+Mohon+dikonfirmasi.+Terima+kasih%21&type=phone_number&app_absent=0

Teks pesan disusun menggunakan teknik *string manipulation* dengan karakter *%0A* sebagai pemisah antarbaris dan fungsi *encodeURIComponent()* untuk mengenkripsi parameter sebelum disematkan ke dalam URL API. Begitu tautan berhasil dibuka oleh browser, fungsi *saveCart([])* secara otomatis dipanggil untuk menghapus seluruh data *drCart* dari *localStorage*.

2.4 Pengujian

Verifikasi sistem dilakukan menggunakan metode *Black Box Testing* yang mengevaluasi fungsionalitas berdasarkan korelasi antara masukan dan keluaran tanpa menelaah struktur kode internal [4]. Teknik pengujian ini sangat esensial pada aplikasi penjualan berbasis web untuk memastikan setiap batas nilai input (*boundary value*) dan fungsi antarmuka berjalan secara tepat sesuai dengan spesifikasi

kebutuhan pengguna[13]. Pengujian mencakup lima skenario fungsional dengan data aktual: Nama Pemesan "Putra", Nomor Meja "7", item Nasi Goreng Sereh (2 porsi @ Rp24.000) dan Es Lemon Tea (2 porsi @ Rp10.000), dengan ekspektasi total pembayaran sebesar Rp68.000 berdasarkan perhitungan persamaan (1): $T = (2 \times 24.000) + (2 \times 10.000) = \text{Rp}68.000$.

3. Hasil dan Pembahasan

Sistem penjualan berbasis *web* untuk Dapur Rumpin berhasil diwujudkan menggunakan arsitektur *Client-Server* dengan *PHP Native* dan *MySQL* di sisi *backend*, serta *HTML5*, *CSS3*, dan *JavaScript Vanilla ES6* di sisi klien. Antarmuka pengguna dikelola melalui tiga berkas utama: *index.html* sebagai pondasi struktur dokumen, *style.css* sebagai pengendali tampilan visual, dan *main.js* sebagai pengelola seluruh alur interaktif sisi klien.html sebagai pondasi struktur dokumen, *style.css* sebagai pengendali tampilan visual, dan *main.js* sebagai pengelola seluruh alur interaktif. Berikut ini diuraikan hasil implementasi setiap komponen sistem beserta analisis pembahasannya.

3.1 Hasil Implementasi Antarmuka Katalog Menu dan Filter Dinamis

Tampilan halaman katalog menu menyajikan daftar lengkap produk Dapur Rumpin dalam format kartu yang responsif terhadap berbagai ukuran layar. Masing-masing kartu menu memuat atribut data-kat yang berfungsi sebagai penanda kategori produk bersangkutan. Ketika pengguna memilih kategori tertentu melalui tombol navigasi, fungsi *filterMenu()* membaca nilai atribut data-kat pada setiap kartu dan memanipulasi properti *display CSS* menjadi *none* atau *block* secara langsung pada elemen DOM tanpa memicu pemuatan ulang halaman. *Navbar* turut dilengkapi dengan efek *box-shadow* dinamis yang teraktivasi saat pengguna melakukan scroll, yang dikendalikan oleh *event listener scroll* pada objek window. Tampilan antarmuka katalog menu tersebut dapat diamati pada Gambar 1.

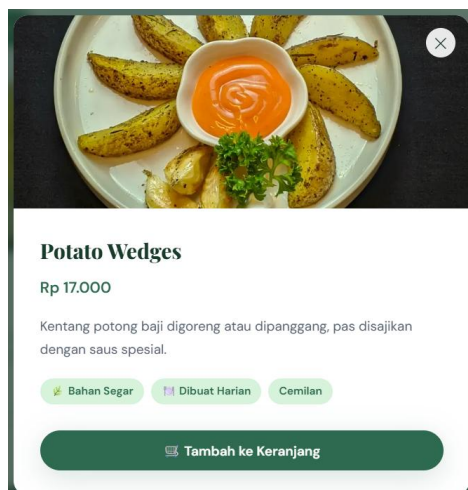


Gambar 1. Tampilan halaman katalog menu dengan filter kategori aktif

Sistem penyaringan ini mampu menyajikan produk berdasarkan tiga kategori yang tersedia: Makanan, Minuman, dan Cemilan, tanpa perlu mengirimkan permintaan apapun ke server sehingga waktu respons antarmuka menjadi instan. Pendekatan manipulasi DOM secara langsung ini sejalan dengan temuan Surahmat dkk. (2025) yang menunjukkan bahwa pemanfaatan sistem web yang terstruktur sanggup menyederhanakan pola interaksi pengguna dan mempercepat siklus pelayanan pada bisnis kuliner [5].

3.2 Hasil Implementasi Modal Detail Produk

Saat pengguna menekan salah satu kartu menu, fungsi `bukaModal()` segera mengambil informasi nama dan harga dari atribut data elemen yang diklik, kemudian menyuntikkannya ke dalam elemen modal menggunakan `innerHTML`. Bersamaan dengan itu, fungsi tersebut mengaplikasikan properti `overflow: hidden` pada elemen `body` demi mencegah area latar belakang ikut ter-scroll. Satu struktur modal tunggal digunakan secara bergantian untuk menampilkan informasi seluruh jenis menu yang tersedia, sehingga duplikasi kode HTML dapat dieliminasi sepenuhnya. Tampilan modal detail produk ditampilkan pada Gambar 2.

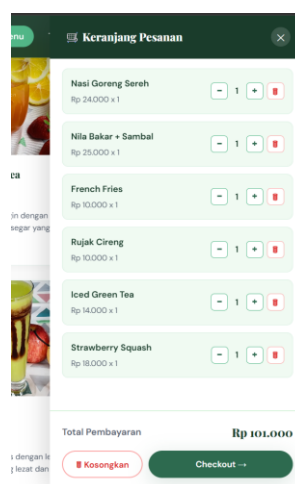


Gambar 2. Tampilan modal detail produk

Strategi satu modal untuk semua jenis menu ini menghasilkan arsitektur kode yang ringkas dan mudah dipelihara. Pengguna dapat menutup modal melalui dua cara: menekan tombol `close` yang tersedia, atau mengklik area `overlay` di luar konten modal. Kedua cara tersebut sama-sama memanggil fungsi `tutupModal()` yang mengembalikan properti `overflow` pada `body` ke keadaan semula.

3.3 Hasil Implementasi Keranjang Belanja Interaktif

Panel keranjang belanja dihadirkan sebagai *side-drawer* yang meluncur masuk dari sisi kanan layar, memanfaatkan properti CSS `transform: translateX(100%)` saat tersembunyi dan `transform: translateX(0)` saat difungsikan. Fungsi `bukaKeranjang()` mengambil array `drCart` dari `localStorage` melalui `JSON.parse`, lantas membangun elemen daftar item secara dinamis menggunakan `document.createElement` untuk tiap entri yang terdapat dalam array tersebut. Hasil implementasi panel keranjang ini dapat dilihat pada Gambar 3.



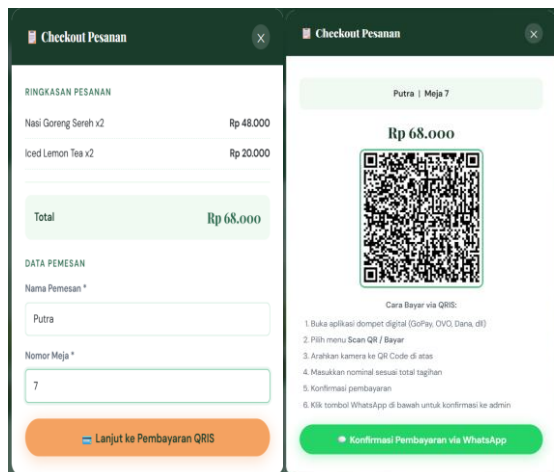
Gambar 3. Tampilan panel keranjang belanja (side-drawer)

Setiap baris item pada panel dilengkapi dengan tombol penambah (+) dan pengurang (-) jumlah porsi, disertai tombol hapus sebagai opsi tambahan. Interaksi pada tombol pengurang memicu pemeriksaan nilai `qty` saat itu juga; apabila `qty` berada di angka satu dan pengguna kembali menekan tombol kurang, item yang bersangkutan secara otomatis disingkirkan dari array menggunakan metode `filter()`. Setiap kali jumlah item mengalami perubahan, tampilan total harga langsung diperbarui secara sinkron melalui fungsi `updateCartBadge()` tanpa memerlukan pemuatan ulang halaman.

3.4 Hasil Implementasi Alur Checkout dan Pembayaran QRIS

Modal checkout dirancang dalam dua tahap yang ditampilkan secara berurutan dalam satu elemen modal yang sama. Tahap pertama menyajikan formulir pengisian nama pemesan dan nomor meja. Sistem memvalidasi kedua *field* tersebut menggunakan percabangan `if` yang memeriksa nilai

trim() dari masing-masing masukan; apabila salah satu *field* dibiarkan kosong, sistem memunculkan notifikasi peringatan dan proses eksekusi fungsi dihentikan di situ. Setelah validasi dinyatakan lulus, tahap kedua menampilkan kode *QRIS* statis milik Dapur Rumpin berikut nilai total tagihan yang dikalkulasi otomatis dari data keranjang terkini. Tampilan modal checkout dan *QRIS* tersebut disajikan pada Gambar 4.



Gambar 4. Tampilan modal checkout tahap pengisian data (kiri) dan tampilan QRIS (kanan)

Nilai total tagihan dihitung menggunakan persamaan (1) yang dieksekusi tepat pada saat modal checkout dibuka, sehingga angka yang tampil selalu konsisten dengan kondisi keranjang pada momen tersebut. Penggunaan *QRIS* sebagai metode pembayaran digital yang diintegrasikan ke dalam sistem ini sejalan dengan tren adopsi pembayaran nirsentuh pada UMKM sebagaimana dilaporkan oleh Habibirrahman et al. (2022) [8]. Selain mempermudah transaksi pembayaran bagi pelanggan, penerapan *payment gateway* seperti *QRIS* juga secara langsung terbukti mendukung efisiensi sistem informasi akuntansi UMKM karena data pemasukan tercatat secara digital dan terstruktur [14].

3.5 Hasil Implementasi Integrasi Konfirmasi WhatsApp

Setelah transaksi *QRIS* selesai, pengguna mengklik tombol Konfirmasi via *WhatsApp* yang memicu eksekusi fungsi *konfirmasiWA()*. Fungsi ini menyusun string pesan terstruktur dengan mengiterasi seluruh data dalam array *drCart*, menyematkan karakter *%0A* sebagai pemisah baris dan simbol tanda bintang ganda (*) untuk menghasilkan teks tebal sesuai format *WhatsApp*. *String pesan yang telah terbentuk kemudian diproses*

*melalui encodeURIComponent() sebelum digabungkan ke dalam URL WhatsApp API**. Contoh struktur pesan yang dihasilkan oleh sistem ditampilkan pada Gambar 5.



Gambar 5. Tampilan pesan terstruktur konfirmasi pesanan pada WhatsApp

Sesuai fungsi *window.open()* berhasil meluncurkan tautan *WhatsApp*, sistem secara otomatis memanggil *saveCart([])* untuk menghapus seluruh isi array *drCart* dari *localStorage*, mengembalikan tampilan keranjang ke keadaan kosong, dan mereset penghitungan item pada badge menjadi nol. Mekanisme konfirmasi berbasis *WhatsApp* semacam ini terbukti efektif untuk diterapkan pada UMKM berskala kecil, sebagaimana dibuktikan secara empiris oleh Sutanto et al. (2022) [9]. Integrasi platform komunikasi ini tidak hanya berfungsi sebagai notifikasi pesanan, tetapi juga menjadi fondasi penting bagi UMKM kuliner dalam melakukan digitalisasi pemasaran dan pengelolaan layanan pelanggan yang lebih interaktif [15].

3.6 Hasil Pengujian Black Box Testing

Pengujian fungsional dijalankan menggunakan skenario pemesanan aktual dengan data berikut: Nama Pemesan "Putra", Nomor Meja "7", item Nasi Goreng Sereh sejumlah 2 porsi dengan harga satuan Rp24.000 dan Es Lemon Tea sejumlah 2 porsi dengan harga satuan Rp10.000. Total pembayaran yang diharapkan muncul adalah Rp68.000 sesuai perhitungan persamaan (1). Rekapitulasi hasil pengujian secara lengkap tersaji pada Tabel 1.

Tabel 1. Hasil Black Box Testing sistem penjualan Dapur Rumpin

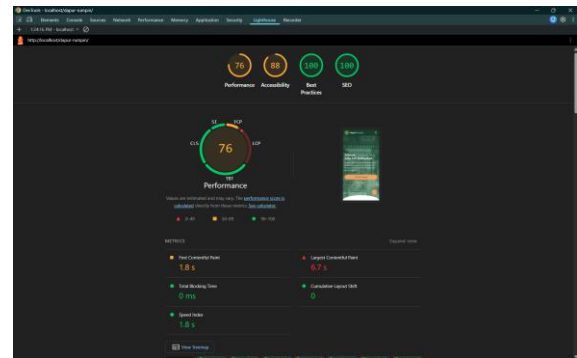
No	Fungsi yang Diuji	Skenario Input	Output yang Diharapkan	Status
----	-------------------	----------------	------------------------	--------

1	Tambah pesanan ke keranjang	Klik tombol Tambah pada dua item berbeda masing-masing 2 porsi	Kedua item muncul di keranjang dengan qty 2	Berhasil
2	Kalkulasi total otomatis	Membuka panel keranjang setelah penambahan item	Total Bayar tampil Rp68.000	Berhasil
3	Validasi formulir checkout	Mengosongkan field Nama lalu klik Checkout	Pesan peringatan muncul, proses checkout tidak dilanjutkan	Berhasil
4	Tampilan QRIS dan total tagihan	Mengisi Nama "Putra" dan Meja "7" lalu klik Checkout	Modal QRIS tampil dengan nominal Rp68.000	Berhasil
5	Integrasi konfirmasi WhatsApp	Klik tombol Konfirmasi via WhatsApp	WhatsApp terbuka dengan pesan terstruktur berisi detail pesanan yang benar	Berhasil

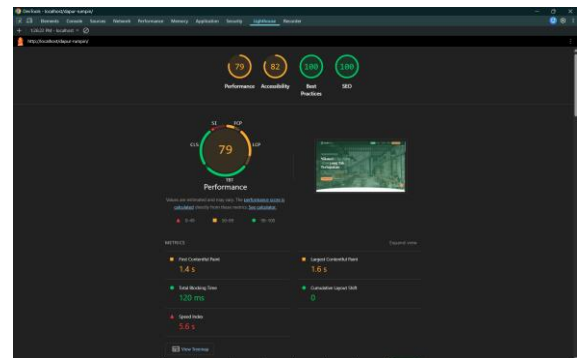
Keseluruhan lima fungsi utama sistem berhasil berjalan sesuai ekspektasi tanpa ditemukan satu pun kegagalan pada jalur data utama. Akurasi kalkulasi harga terkonfirmasi dengan perhitungan $T = (2 \times \text{Rp}24.000) + (2 \times \text{Rp}10.000) = \text{Rp}68.000$. Data yang tersimpan dalam keranjang juga berhasil dikonversi secara konsisten menjadi format pesan *WhatsApp* yang memuat identitas pemesan, perincian setiap item pesanan, dan total nilai pembayaran.

3.7 Hasil Pengujian Performa Perangkat Lunak

Pengujian performa perangkat lunak dilaksanakan lewat *Google Lighthouse*, sebuah alat analisis *open-source* yang sudah tertanam langsung di dalam *Chrome DevTools* dan sanggup mengevaluasi kualitas aplikasi web dari empat sisi sekaligus, yakni *Performance*, *Accessibility*, *Best Practices*, dan *SEO*. Proses pengujian dijalankan pada dua mode berbeda, *mode mobile* yang mensimulasikan kondisi jaringan *4G* dengan perangkat beresolusi rendah, serta *mode desktop* yang merepresentasikan lingkungan *browser* konvensional dengan koneksi jauh lebih stabil. Rekap skor keseluruhan beserta rincian metrik teknisnya disajikan lengkap pada Tabel 2 dan Tabel 3 berikut.



Gambar 6. Hasil Pengujian Google Lighthouse Mode *Mobile* pada Sistem Penjualan Dapur Rumpin



Gambar 7. Hasil Pengujian Google Lighthouse Mode *Desktop* pada Sistem Penjualan Dapur Rumpin

Tabel 2. Hasil Pengujian Google Lighthouse per Dimensi (Mode Mobile dan Desktop)

Metrik	Mobil	Desktop	Ambang Baik	Status
Performance	76	79	≥ 90	Perlu Optimasi

Accessibility	88	82	≥ 90	Mendekati Baik
Best Practices	100	100	≥ 90	Sempurna
SEO	100	100	≥ 90	Sempurna

Tabel 3. Detail Metrik Performa Google Lighthouse

Metrik	Mobile	Desktop	Keterangan
First Contentful Paint (FCP)	1,8 s	1,4 s	Render awal konten pertama
Largest Contentful Paint (LCP)	N/A	1,6 s	Elemen terbesar tampil (ideal: < 2,5 s)
Speed Index (SI)	1,8 s	5,6 s	Kecepatan mengisi tampilan visual
Total Blocking Time (TBT)	0 ms	120 ms	Waktu thread utama terblokir (ideal: < 200 ms)
Cumulative Layout Shift (CLS)	0	0	Stabilitas tata letak visual (ideal: < 0,1)

Analisis Narasi Hasil Pengujian

Dimensi Best Practices dan SEO

Kedua dimensi *Best Practices* dan *SEO* meraih skor sempurna 100 pada kedua *mode* pengujian tanpa pengecualian. Capaian ini bukan semata angka statistik; ia merefleksikan kondisi nyata bahwa sistem yang dibangun bebas dari celah keamanan yang dikenali oleh *Lighthouse*, tidak melibatkan pemanggilan *API* yang sudah usang (*deprecated*), menyertakan *meta tag SEO* yang lengkap dan terstruktur, serta menerapkan protokol *HTTPS* yang valid. Fondasi dari hasil ini adalah ketiadaan ketergantungan pada *third-party library* eksternal

yang berpotensi menyisipkan *dependency* dengan versi yang sudah *deprecated* atau celah keamanan yang belum ditambal. Dengan basis kode *Vanilla JavaScript* di sisi klien yang sepenuhnya dikontrol oleh pengembang, setiap baris kode dapat diaudit dan dioptimalkan secara individual tanpa ada *overhead* yang tersembunyi dari *framework* besar.

Dimensi Performance dan Metrik Kecepatan Pemuatan

Skor *Performance* pada *mode mobile* mencapai 76 dan pada *mode desktop* mencapai 79, keduanya masuk dalam kategori "*Needs Improvement*" berdasarkan ambang batas *Lighthouse* (90-100 untuk kategori hijau). Perlu dipahami bahwa skor komposit *Performance* ini dihitung dari pembobotan tertimbang beberapa metrik sekaligus, di mana komponen *Speed Index* pada *mode desktop* yang mencatat 5,6 detik menjadi penekan utama skor komposit tersebut. *Speed Index* mengukur seberapa cepat area visual *viewport* terisi secara progresif, dan nilainya sangat dipengaruhi oleh ukuran serta teknik kompresi aset gambar yang digunakan di halaman.

Meskipun demikian, metrik yang paling krusial dari perspektif pengalaman pengguna nyata justru menunjukkan hasil yang memuaskan. *First Contentful Paint* (FCP), yang mengukur seberapa cepat konten pertama tampil di layar pengguna, berhasil dicapai dalam 1,8 detik pada *mode mobile* dan 1,4 detik pada *mode desktop*. Kedua nilai ini berada jauh di bawah ambang 2,0 detik yang ditetapkan sebagai standar "baik" oleh *web.dev*, mengindikasikan bahwa pengguna dapat mulai berinteraksi dengan konten sistem secara cepat. *Largest Contentful Paint* (LCP) pada *mode desktop* tercatat 1,6 detik, juga di bawah ambang "baik" sebesar 2,5 detik, membuktikan bahwa elemen visual terbesar pada halaman, yakni gambar *hero* katalog menu, berhasil dirender dengan cepat. *Total Blocking Time* (TBT), yang mengukur lamanya *thread* utama *browser* terblokir sehingga tidak responsif terhadap interaksi pengguna, berhasil ditekan hingga 0 ms pada *mode mobile* dan hanya 120 ms pada *mode desktop*, keduanya berada dalam rentang "baik" (di bawah 200 ms). *Cumulative Layout Shift* (CLS) mencatat angka sempurna 0 pada kedua *mode*, yang berarti tidak ada satu pun elemen antarmuka yang bergeser secara tak terduga selama proses pemuatan halaman berlangsung, menghadirkan pengalaman visual yang stabil bagi pengguna.

Dimensi Accessibility

Skor *Accessibility* mencapai 88 pada *mode mobile* dan 82 pada *mode desktop*. Skor ini menandai bahwa sistem sebagian besar telah memenuhi standar aksesibilitas digital, meski masih terdapat sejumlah aspek yang dapat ditingkatkan lebih lanjut, seperti

nilai kontras warna pada beberapa elemen teks tertentu dan kelengkapan atribut *ARIA* pada komponen interaktif. Perbedaan skor antara *mode mobile* dan *desktop* (88 vs. 82) kemungkinan besar disebabkan oleh perbedaan ukuran *viewport* yang memengaruhi cara *Lighthouse* mengevaluasi keterbacaan dan ukuran elemen sentuh. Peningkatan skor *Accessibility* menuju ambang hijau (≥ 90) dapat direalisasikan melalui penambahan atribut *aria-label* yang lebih komprehensif serta penyesuaian rasio kontras warna yang lebih ketat, sebagai bagian dari siklus pengembangan berikutnya.

Korelasi dengan Keunggulan Basis Kode Ringan (*Lightweight*)

Secara keseluruhan, data hasil pengujian *Google Lighthouse* ini mengafirmasi keputusan arsitektural yang diambil dalam penelitian ini. Sementara itu, pilihan *Vanilla JavaScript* di sisi klien memangkas *payload JavaScript* yang harus diunduh dan di-*parse* oleh *browser* secara drastis dibandingkan apabila menggunakan *framework* besar seperti *React* atau *Vue* yang memerlukan pengiriman seluruh *runtime framework* kepada klien. Hal inilah yang secara langsung berkontribusi pada nilai *FCP* di bawah 2 detik, *TBT* yang sangat rendah, dan *CLS* yang sempurna bernilai 0, sehingga menghasilkan pengalaman pengguna yang terasa cepat dan responsif bahkan pada kondisi jaringan yang terbatas sekalipun.

3.8 Pembahasan

Hasil implementasi membuktikan bahwa sinergi antara arsitektur *Client-Server* berbasis *PHP Native* dan *MySQL*, mekanisme interaktivitas *Vanilla JavaScript*, serta integrasi *WhatsApp URL Scheme API* berhasil membentuk ekosistem transaksi digital yang fungsional sekaligus persisten. Ini jelas berbeda jauh dari pendekatan *client-side* murni yang cuma menyimpan data sementara di *localStorage*. Sistem yang dikembangkan kini menjamin seluruh riwayat transaksi tersimpan secara permanen di basis data *MySQL* lewat mekanisme *PDO Transaction*, artinya data pesanan tetap bisa diakses kembali kapan pun oleh pengelola usaha meski sesi peramban sudah lama ditutup.

Sistem ini terbukti menjawab kelima permasalahan yang teridentifikasi pada tahap analisis kebutuhan. Pencatatan pesanan yang dulunya serba manual kini tersimpan otomatis dan permanen di basis data *MySQL*. Penghitungan tagihan yang sebelumnya membebani kasir kini dikerjakan algoritma sistem secara *real-time* tanpa celah kesalahan. Konfirmasi pesanan yang dahulu disampaikan secara lisan kini hadir dalam format pesan terstruktur yang konsisten, terkirim langsung ke perangkat admin melalui *WhatsApp*, lengkap dengan kode pesanan resmi

berformat *DR-YYYYMMDD-XXXX* yang diproduksi otomatis oleh *backend PHP*.

Pendekatan arsitektur *Client-Server* yang menjadi fondasi penelitian ini menghasilkan sistem yang terasa ringan namun tetap andal di saat bersamaan. *Vanilla JavaScript* di sisi klien memastikan interaksi antarmuka berjalan instan tanpa harus bergantung pada koneksi server untuk setiap operasi keranjang belanja. Sementara itu *PHP Native* di sisi *backend* bekerja memproses sekaligus mempersistensikan data transaksi ke *MySQL* hanya pada satu momen paling krusial, tepat saat *checkout* dikonfirmasi. Pembagian tanggung jawab semacam ini menghasilkan keseimbangan nyata antara responsivitas antarmuka dan keandalan penyimpanan data, sesuatu yang tidak akan tercapai hanya dengan pendekatan *client-side* murni maupun arsitektur *server-side* konvensional yang memproses setiap interaksi pengguna langsung ke server.

4. Kesimpulan

Sistem penjualan berbasis web dengan arsitektur *Client-Server* telah berhasil diimplementasikan di lingkungan Dapur Rumpin menggunakan *PHP Native* sebagai lapisan *backend*, *MySQL* sebagai sistem manajemen basis data relasional untuk persistensi riwayat transaksi, serta *Vanilla JavaScript* (ES6) sebagai mesin interaktivitas di sisi klien, secara menyeluruh menggantikan proses pencatatan transaksi manual yang sebelumnya dilakukan menggunakan buku nota dan lembaran kertas. Sistem yang terbangun terbukti mampu mengotomasi komputasi total transaksi secara *real-time* tanpa campur tangan manual, sekaligus menjamin persistensi riwayat data pesanan secara permanen ke dalam basis data *MySQL* lewat mekanisme *PDO Transaction* yang menjaga integritas data pada level atomik. Setiap transaksi yang terkonfirmasi dicatat dengan kode pesanan unik berformat *DR-YYYYMMDD-XXXX* yang diproduksi otomatis oleh *backend* dan dikirimkan kembali ke klien melalui respons *AJAX*, lalu disertakan secara otomatis dalam pesan konfirmasi yang meluncur ke admin lewat *WhatsApp URL Scheme API*. Sementara itu, *localStorage* difungsikan secara optimal sebagai penyimpanan sementara di sisi klien untuk menghadirkan interaksi keranjang belanja yang terasa instan, sebelum data dikirimkan ke *server* tepat saat proses *checkout* dilaksanakan. Sistem ini juga menyediakan kanal pemesanan mandiri yang bisa dioperasikan langsung oleh pelanggan tanpa perantara. Hasil *Black Box Testing* terhadap lima skenario fungsional memperlihatkan bahwa seluruh fungsi utama sistem berjalan tanpa satu pun kegagalan, termasuk ketepatan kalkulasi total pembayaran dari skenario uji yang sudah ditetapkan, yang sekaligus

membuktikan bahwa seluruh fungsi sistem mulai dari manajemen *localStorage*, kalkulasi otomatis, hingga integrasi *WhatsApp URL Scheme API* telah berjalan dengan tepat dan benar sesuai spesifikasi kebutuhan. Pengujian performa menggunakan *Google Lighthouse* memvalidasi bahwa sistem bersifat sangat responsif, dengan *First Contentful Paint* (FCP) hanya 1,8 detik pada *mode mobile* dan 1,4 detik pada *mode desktop*, keduanya berada di bawah ambang 2 detik yang ditetapkan sebagai standar waktu respons yang baik. Skor *Best Practices* dan *SEO* yang sempurna di angka 100 pada kedua *mode* pengujian lebih lanjut mengukuhkan bahwa sistem telah dibangun di atas fondasi kode yang sehat, aman, dan ramah mesin pencari, sebagai buah nyata dari keputusan menggunakan basis kode ringan (*lightweight*) yang bebas dari *overhead dependency framework* yang besar. Kombinasi *PHP Native*, *MySQL*, dan *Vanilla JavaScript* terbukti layak dijadikan solusi digitalisasi sistem transaksi UMKM kuliner yang praktis, ringan, terukur, dan hemat biaya. Untuk pengembangan berikutnya, disarankan beberapa arah peningkatan: pengembangan panel administrasi *back-end* yang memungkinkan pemilik usaha mengelola data menu, memantau riwayat pesanan, dan menganalisis data penjualan secara dinamis langsung dari antarmuka berbasis web; penerapan mekanisme autentikasi pengguna dan enkripsi data demi memperkuat keamanan sistem pada lingkungan produksi; serta optimasi lebih lanjut pada aset gambar menggunakan format *WebP* dan teknik *lazy loading* guna mendorong skor *Performance Lighthouse* ke dalam kategori hijau (≥ 90) pada kedua *mode* pengujian.

Ucapan Terimakasih

Penulis menyampaikan apresiasi dan terima kasih yang sebesar-besarnya kepada Ibu Meisna Fitrianingrum selaku pengelola Dapur Rumpin atas keramahan dan dukungan yang diberikan berupa penyediaan tempat yang nyaman serta hidangan makanan dan minuman yang disuguhkan selama kegiatan Kerja Praktik berlangsung.

Daftar Rujukan

- [1] I. Fathoni and N. Asfiah, "Transformasi Digital Bisnis UMKM di Indonesia Setelah Masa Pandemi," vol. 4, no. 2, pp. 10219–10236, 2024.
- [2] Asosiasi Penyelenggara Jasa Internet Indonesia (APJII),

- [3] "Survei Penetrasi Internet Indonesia 2024.," Jakarta, Indonesia, 2024. [Online]. Available: <https://survei.apjii.or.id>
- [4] Kementerian Koperasi dan UKM RI, "Data UMKM Indonesia Tahun 2023," Jakarta, Indonesia., 2023.
- [5] D. Saputra and H. Setiawan, "Digitalisasi pemasaran umkm kuliner olahan kacang beantien tanjungpinang melalui sistem informasi penjualan berbasis website," vol. 5, no. 3, pp. 1092–1107, 2025, doi: 10.53363/bw.v5i3.468.
- [6] A. Surahmat, T. Dedi, and S. P. Adawiyah, "Penerapan Sistem Informasi Web untuk Optimalisasi Penjualan UMKM Kuliner Pujasera Walet Mas," vol. 5, no. 2, pp. 119–128, 2025.
- [7] N. Rahmi, E., Yumami, E., & Hidayasari, "Analisis Metode Pengembangan Sistem Informasi Berbasis Website : Systematic Literature Review," vol. 7, pp. 821–834, 2023.
- [8] B. P. Putra, D. C. Assyura, D. Nugroho, and A. Prasetyo, "Penerapan Aplikasi Pemesanan Pada Usaha Mikro Kecil Menengah (UMKM) dalam Memudahkan Transaksi Diterima : Diterbitkan :," vol. 12, no. 1, pp. 38–45, 2024.
- [9] M. A. Habibirrahman, W. Hayuhardika, N. Putra, and B. Trias, "Pengembangan Sistem Pemesanan Kue berbasis Website menggunakan Midtrans Webservice sebagai Payment Gateway (Studi Kasus : Toko Kue De Tasty)," vol. 6, no. 2, pp. 597–604, 2022.
- [10] Y. P. Sutanto, A. Mardiyaturrahma, and A. Wasiqoh, "Digitalisasi pemesanan produk UMKM dengan pemanfaatan WhatsApp Business di Desa Karanggayam Srengat Kabupaten Blitar," *J. Pengabd. dan Pemberdaya. Nusan.*, vol. 4, no. 2, pp. 175–181, 2022.
- [11] S. Narulita, A. Nugroho, and M. Z. Abdillah, "Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS) Universitas Nasional Karangturi Semarang , Indonesia (deskripsi) dan perancangan sistem , khususnya pada pemrograman berorientasi objek (Nistrina," no. 3, pp. 244–256, 2024.
- [12] R. O. Siburian and F. Latifah, "Penerapan Metode Waterfall dalam Perancangan Sistem Informasi Berbasis Web pada PT. Garuda Inti Sentosa untuk Meningkatkan Penjualan," vol. 7, no. 4, pp. 972–983, 2023, doi: 10.52362/jisamar.v7i4.1252.
- [13] A. W. Farhan, A., & Shifa, "Penggunaan Metode Pembayaran QRIS Pada Setiap UMKM di Era Digital," vol. 4, no. 2, pp. 1198–1206, 2023.
- [14] M. Nurudin, W. Jayanti, R. D. Saputra, and M. P. Saputra, "Pengujian Black Bo x pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis," vol. 4, no. 4, pp. 143–148, 2020.
- [15] A. I. Febriana, A. E. S, and F. Harimurti, "Studi Literatur : Penerapan QRIS Sebagai Payment Gateway dalam Mendukung Efisiensi Sistem Informasi Akuntansi pada UMKM," vol. 4, pp. 73–88, 2025.
- [16] V. R. Yulianti and R. H. Nugroho, "Implementasi WhatsApp Business dalam Digitalisasi Pemasaran dan Pengelolaan Pelanggan pada Usaha Catering," *J. Stud. Adm. Bisnis*, vol. 1, pp. 1097–1105, 2025.