International Journal Software Engineering and Computer Science (IJSECS)
6 (3), 2026, 1039-1045
Published Online December 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i3.8325.

RESEARCH ARTICLE Open Access

Implementation of a Website-Based QR Menu and Self-Ordering System at Warkop Abas

Romi Naufal Salam [1*], **Nova Dahliyanti** [2], **Mesra Betty Yel** [3]

[1*,2,3] Department of Informatics Engineering, Faculty of Computer Technology, Sekolah Tinggi Ilmu Komputer Cipta Karya Informatika, East Jakarta City, Special Capital Region of Jakarta, Indonesia.

Article History:

Received: August 7, 2026.
Revised: August 11, 2026.
Accepted: August 27, 2026.
Available online: September 20, 2026.
Published: December 1, 2026.

***Corresponding Author:**
naufalromi83@gmail.com

Abstract: Warkop Abas is a culinary enterprise operating with an indoor capacity of 25 dining tables and a daily volume reaching approximately 100 visitors. However, order recording currently relies entirely on manual staff operations managed by only four workers, causing acute service bottlenecks during peak rush hours. Consequently, food fulfillment intervals frequently extend up to 40 minutes, elevating operational stress and clerical order inaccuracies. This study designs, develops, and implements a website-based QR Menu and self-ordering platform using WordPress and WooCommerce to resolve these operational constraints. The browser-based architecture enables patrons to browse five distinct menu categories and order autonomously by scanning table-specific QR codes with guest checkout access, requiring no account registration or native application downloads. To accommodate diverse consumer behaviors, the system incorporates flexible settlement options via digital QRIS and direct cashier payment, while floor fulfillment workflows are tracked synchronously through an active, real-time Cashier Dashboard. Black Box testing across 14 scenarios confirmed complete functional validity across all transaction workflows. Furthermore, White Box structural analysis of critical execution paths—cashier authentication, checkout processing, order configuration, and fulfillment status updates—yielded low-to-moderate cyclomatic complexity values ($V(G) \leq 5$), demonstrating robust procedural stability and codebase maintainability. In addition, centralized database logging eliminates manual paper ticket reconciliation and provides instant sales auditing for business owners. The implemented system successfully decompresses front-line staff workloads, shortens customer ordering latency upon seating, and delivers an accessible, cost-effective digital ordering solution tailored for micro-scale food and beverage establishments.

Keywords: Qr Menu; Self-Ordering; Website; WordPress; WooCommerce; Warkop Abas.

1. Introduction

The continuous expansion of digital technology has created significant momentum for culinary businesses, particularly micro, small, and medium enterprises (MSMEs), to modernize their operational infrastructure. Traditional food stalls and coffee establishments, which have long relied on conventional face-to-face and paper-based order management, are increasingly encouraged to adopt website-based ordering platforms. These digital systems offer substantial operational advantages, including accelerated service processing, automated data recording, and a marked reduction in human-induced transcription errors (Rusito & Putri, 2024). In high-volume dining environments, transitioning to self-service digital platforms represents not merely a technical upgrade, but a strategic necessity to maintain service quality and operational agility amid growing consumer demands for rapid transactions. Warkop Abas represents a typical small-scale culinary business facing acute operational challenges due to reliance on manual mechanisms. The establishment operates with an indoor capacity of approximately 25 dining tables, serving a daily customer volume of roughly 100 visitors. During peak hours, customer arrivals frequently surge up to 50 visitors within compressed timeframes, generating an aggregate demand of over 300 menu items that must be prepared and served daily. All floor tasks are handled by only four staff members who must concurrently record customer orders, provide menu descriptions, coordinate item delivery with the kitchen staff, and handle payment transactions at the cashier. This centralized operational burden severely bottlenecks service flow during rush periods. While manual interactions remain manageable during quiet hours, staff multi-tasking during peak demand inevitably creates operational strain and clerical inaccuracies. Direct field observations indicate that meal fulfillment requires 10–20 minutes under normal conditions, but frequently extends to 20–40 minutes when customer queues escalate. Similarly, beverage preparation and delivery stretch from standard intervals of 5–10 minutes to protracted durations of 10–30 minutes, directly degrading customer satisfaction and constraining table turnover rates.

These persistent delays stem directly from the structural dependency of the ordering process on staff physical availability, as customers possess no autonomous means to log choices without waiting for an attendant. Comparable empirical settings corroborate that paperless, self-directed mechanisms resolve similar workflow bottlenecks. Integrating QR codes into restaurant menu ordering frameworks has been demonstrated to improve data accuracy, eliminate order miscommunication, and accelerate customer payment workflows (Rusito & Putri, 2024). Furthermore, rapid application development approaches have proven highly capable of delivering responsive, user-aligned QR-based menu solutions efficiently (Kaban & Nasution, 2020). Implementing self-ordering paradigms in dining venues noticeably alleviates front-desk congestion and enhances operational turnaround (Salsabiela *et al.*, 2020), while embedding QR codes into self-order systems markedly minimizes customer dependence on floor attendants (Sitepu *et al.*, 2025). Despite these documented advancements, prevailing studies often evaluate QR menus or self-ordering interfaces in fragmented configurations, frequently requiring proprietary mobile app installations or complex proprietary software architectures that are costly to maintain. The novelty of this study lies in engineering a unified, browser-accessible QR Menu and Self-Ordering system deployed on lightweight WordPress and WooCommerce infrastructure, specifically customized for micro-scale dining enterprises such as Warkop Abas. The proposed system integrates table-specific QR recognition, dual payment collection channels (QRIS and direct counter payment), and an active, real-time Cashier Dashboard for immediate order fulfillment monitoring. Accordingly, this research aims to systematically examine the deployment of this integrated website-based platform to mitigate service turnaround delays, evaluate its capability in supporting business operational capacity and sales potential, and conduct functional and structural validation of the system within the live operational environment of Warkop Abas.

2. Related Work

2.1 Theoretical Framework: Information Systems, QR Codes, and Self-Ordering

An information system is fundamentally defined as an integrated set of human resources, hardware, software, data, and standardized operating procedures functioning collaboratively to produce accurate operational intelligence and facilitate institutional decision-making. In the context of this study, this theoretical foundation underpins the development of the QR Menu and Self-Ordering architecture, ensuring that digital ordering, data storage, and transaction status monitoring proceed systematically, securely, and efficiently. A website serves as an interconnected repository of digital web pages accessible via internet protocols, acting as the primary communication bridge that enables customers to browse food selections, assemble orders, and execute payment without requiring attendant mediation. A QR (Quick Response) code, constructed as a two-dimensional matrix barcode decodable instantly through mobile camera optics, functions as an instantaneous gateway linking physical dining tables to specific digital menu routes, thereby eliminating the physical distribution of printed menu catalogs (Rusito & Putri, 2024). In parallel, a self-ordering system grants

customers the autonomy to configure, customize, and transmit their orders directly to the processing database, with the explicit objectives of smoothing queue distributions, shortening service intervals, and mitigating manual transcription inaccuracies (Salsabiela *et al.*, 2020). From an implementation perspective, WordPress was deployed as the core application engine owing to its architectural maintainability, extensibility, and broad plugin ecosystem that accelerates software prototyping cycles. WooCommerce functions as the integrated e-commerce processing layer managing menu categorizations, shopping cart logic, transaction settlement, and sales database storage, thereby anchoring end-to-end customer workflows and administrative data capture. To ensure operational integrity, Black Box Testing is utilized as the primary functional evaluation framework, focusing on the behavioral verification of system inputs and outputs without examining inner source-code logic, verifying that QR table routing, digital browsing, shopping cart modifications, checkout steps, payment selection, and administrative order handling perform according to functional specifications.

2.2 Review of Previous Studies

An examination of prior empirical literature was conducted to benchmark existing technical paradigms and methodologies applied to QR-code ordering and self-service infrastructures across diverse culinary environments. A comparative summary of relevant previous research and the positioning of this study is presented in Table 1.

Table 1. Summary of Previous Studies and Current Research Positioning

No	Study Title / Research Focus	Authors (Year)	Summary of Key Findings	Current Research Positioning
1	Implementation of QR Code for Web-Based Menu Ordering Information System at Restoran Omah Gedhe Pandean Kaliwungu	Rusito & Putri (2024)	QR codes coupled with website interfaces enhance order turnaround efficiency, improve transaction recording, and facilitate digital settlement.	Introduces a complete, autonomous self-ordering flow coupled with dual settlement channels (QRIS and cash at counter).
2	Web-Based Mobile Coffee Ordering System	Kevin & Dewayani (2024)	Web platforms simplify ordering procedures within small-scale mobile retail coffee setups.	Contextualized specifically for dine-in operations with a 25-table capacity and unique table-specific QR routing.
3	Web-Based Self-Ordering Food Ordering System for Dining Restaurants	Salsabiela *et al.* (2020)	Self-ordering structures reduce front-counter queues and noticeably accelerate customer service delivery.	Embeds complete self-ordering coupled with automated table identification parsed directly from table QR parameters.
4	Application of Rapid Application Development (RAD) in Menu Ordering System Design Using QR Codes	Kaban & Nasution (2020)	RAD frameworks effectively compress prototyping cycles for user-centered QR ordering designs.	Emphasizes direct end-to-end deployment using WordPress and WooCommerce rather than conceptual design prototypes alone.
5	Design of Self-Order Information Systems with QR Technology at Healing Cafe	Sitepu *et al.* (2025)	QR-code deployment noticeably diminishes customer operational reliance on floor attendants.	Adds an integrated real-time Cashier Dashboard alongside dual payment collection routes.
6	Analysis of the Impact of Menu Ordering Applications at AFA Warkop Using Android-Based QR Codes	Rejeki *et al.* (2025)	Mobile QR-based applications improve customer satisfaction in traditional coffee shop environments.	Uses a browser-accessible web approach requiring zero app downloads to eliminate consumer device barriers.
7	Implementation of Menu Ordering System at Sunny Bangkinang Cafe & Resto Using Web-Based QR Code on CodeIgniter	Milenia *et al.* (2023)	Web systems utilizing QR technology substantially shorten customer order processing latency.	Implements WordPress and WooCommerce as an agile, lightweight operational stack without custom MVC framework overhead.

As outlined in Table 1, previous studies have generally implemented QR codes or self-ordering mechanics in isolated service segments, with limited literature addressing the cohesive combination of table-specific QR menus, self-directed checkout, dual payment flexibility (QRIS and counter payment), and an active real-time Cashier Dashboard deployed on WordPress and WooCommerce for medium-capacity dine-in coffee establishments. This research addresses this practical and methodological gap by designing and evaluating a website-based QR Menu and self-ordering architecture specifically engineered around the day-to-day operational constraints of Warkop Abas.

3. Methodology

3.1 Research Approach and Context

This study adopted an empirical descriptive method integrated with an applied software engineering approach. The descriptive aspect captured the actual operational conditions at the study site using collected field data, while the applied implementation addressed operational bottlenecks through the deployment of a website-based QR Menu and self-ordering architecture. The research progressed through structured stages, including problem identification, field data collection, requirement modeling, software prototyping, implementation, and functional testing. The operational investigation was conducted at Warkop Abas, involving the business owner, service attendants, and dining customers as primary end-users.

3.2 Data Collection Techniques

Empirical data were gathered on-site at Warkop Abas through three complementary techniques. Direct observation was conducted on daily service operations alongside the owner, floor staff, and patrons to analyze order dispatching flows, recording procedures, and service turnaround latency. Semi-structured interviews were carried out with the business owner and service personnel to gather detailed operational insights into the business profile, transaction flows, technical requirements, and transaction administration bottlenecks. A literature study was performed across scientific journals, technical books, and relevant empirical publications covering information systems, web engineering, QR code technology, and self-ordering paradigms to establish theoretical and technical foundations.

3.3 System Development Life Cycle

The system development framework comprised five structured phases: 1) requirements analysis, identifying customer and administrative operational needs from observation and interview data; 2) prototype design, establishing ordering workflows and administrative order-processing interfaces configured for guest access without mandatory customer login; 3) prototype evaluation, presenting the preliminary interface and transaction workflows to the business owner to verify practical alignment with daily operations; 4) prototype refinement, modifying interface elements and transaction handling based on owner feedback; and 5) system implementation, constructing the complete operational features comprising QR menus, self-directed checkout, QRIS and counter payment routes, and the real-time Cashier Dashboard using WordPress and WooCommerce as the technical core.

3.4 System Requirements Analysis

The requirement analysis yielded distinct functional specifications for dining customers and administrative personnel. Customers require frictionless access to digital menus by scanning table-specific QR codes, category browsing, item visual descriptions, shopping cart persistence, guest checkout without authentication, flexible payment selection between QRIS and the cashier counter, and detailed order summaries. Administrative personnel require authenticated access, menu item and category administration, real-time order arrival monitoring, order status modification (pending, processing, completed), payment verification tracking, and access to transaction histories and sales summaries. The identified functional requirements are detailed in Table 2.

Table 2. System Functional Requirements

Code	System Functional Requirement
F-01	The system shall display the full catalog and detailed product descriptions.
F-02	The system shall recognize and parse table numbers automatically via table QR codes.
F-03	The system shall maintain and display customer shopping cart items.
F-04	The system shall process customer checkouts without requiring account registration or login.
F-05	The system shall facilitate digital payments via QRIS and cash payments at the counter.
F-06	The system shall deliver real-time incoming orders to the Cashier Dashboard.

F-07	The system shall enable administrators to manage product data, categories, promotions, and bundled offerings.
F-08	The system shall enable administrators and cashiers to update order fulfillment statuses and view transaction reports.

Non-functional requirements include hardware provisions comprising a desktop computer or laptop for administrative and cashier terminals; client-side software compatibility with modern web browsers including Google Chrome, Apple Safari, and Microsoft Edge; stable internet network connectivity; administrative security enforced via encrypted credential authentication; an intuitive, low-friction user interface; and broad responsive compatibility across consumer smartphones.

3.5 Proposed System Design and Workflow

The proposed architecture assigns a unique, cryptographically distinct QR code to each dining table, enabling the web application to resolve and bind the table identifier automatically upon scanning. Following table recognition, customers browse the digital menu catalog, select items, adjust order quantities, and proceed to guest checkout by reviewing the pre-assigned table number and adding optional order notes. Customers then choose between QRIS digital settlement and direct cash payment at the cashier counter. Upon transaction confirmation, order records are stored within the WooCommerce database and routed to the Cashier Dashboard, enabling staff to acknowledge, process, update progress statuses, and deliver items to the designated table. System interactions and procedural pathways were modeled using use case diagrams defining interactions between customer and cashier actors, alongside activity diagrams and structured flowcharts detailing order submission and administrative handling routines.

3.6 Testing Methodology

System validation was conducted following full architectural deployment, utilizing Black Box Testing and White Box Testing. Black Box Testing evaluated fourteen functional scenarios based on input-output behavioral verification across cashier authentication, table QR routing, catalog categorization, menu querying, cart modifications, checkout execution, QRIS and counter payment processing, real-time dashboard notifications, status modifications, catalog maintenance, order summaries, and aggregate sales reports, without inspecting internal source code. White Box Testing analyzed logical control flows across four mission-critical routines: cashier authentication, checkout execution, order placement, and status updates. Logical complexity was measured using McCabe's Cyclomatic Complexity metric:

$$V(G) = E - N + 2 \tag{1}$$

Where $V(G)$ represents the cyclomatic complexity, E is the number of edges, and N is the number of nodes in the corresponding control flow graph. This metric established the minimum number of linearly independent paths required to verify structural test coverage.

4. Results and Discussion

4.1 Results

The website-based QR Menu and Self-Ordering system was successfully constructed and trialed at Warkop Abas utilizing WordPress and WooCommerce as the technical foundation. From the customer interface, ordering workflows that previously depended entirely on floor attendants can now be executed independently by scanning table-specific QR codes. Customers browse offerings across five organized categories—Rice Dishes, Noodles, Beverages, Coffee, and Snacks—add items to their digital shopping basket, and complete checkout by confirming the pre-assigned table number and selecting either QRIS settlement or cash payment at the counter, without requiring user account registration or login. Submitted orders are recorded automatically in the centralized database alongside their respective table identifiers, mitigating transcription discrepancies. On the administrative side, the Cashier Dashboard displays incoming orders in real time, detailing ordered items, selected payment options, and total transaction values. Cashiers can modify fulfillment statuses across three stages—pending, processing, and completed—while reviewing transaction logs and daily sales summaries without manual reconciliation. The administrative panel enables straightforward management of catalog items, prices, and stock availability, immediately reflecting changes on the customer-facing interface. Functional validation was performed via Black Box Testing across fourteen comprehensive scenarios, with all test cases achieving valid outcomes, as summarized in Table 3.

Table 3.Summary of Black Box Testing Results

Test No	Functional Feature	Test Scenario	Validation Status
1–2	Cashier Authentication	Authentication using valid and invalid user credentials	Valid
3	QR Menu Routing	Table-specific QR code scanning and direct menu redirection	Valid
4–5	Catalog & Shopping Cart	Category filtering, item search, and cart quantity adjustments	Valid
6–7	Checkout Execution	Order submission with valid and empty table number fields	Valid
8–9	Settlement Selection	Transaction completion via QRIS and counter cash payment	Valid
10–11	Cashier Dashboard	Real-time order capture and order status modifications	Valid
12	Catalog Administration	Creation, modification, and deletion of menu items by admin	Valid
13–14	Summaries & Reporting	Detailed transaction receipt display and sales report aggregation	Valid

Structural validation via White Box Testing was conducted on the four mission-critical operational routines. Cashier authentication evaluated 5 nodes and 6 edges, yielding a cyclomatic complexity of $V(G) = 3$, corresponding to three independent control paths: successful login, authentication failure due to an invalid username, and failure due to an incorrect password. Checkout processing evaluated 6 nodes and 9 edges, producing $V(G) = 5$, covering successful checkout, submission failure due to a missing table number, failure due to an empty cart, and two conditional branches based on payment method selection. Order item processing evaluated 5 nodes and 7 edges, generating $V(G) = 4$, which accounted for adding new items, modifying quantities, removing items, and direct checkout without cart adjustments. Order status updates evaluated 5 nodes and 7 edges, producing $V(G) = 4$, representing state transitions to processing, completed, remaining pending, and cancellation of status changes. All linear independent paths across the four routines were successfully traversed and produced outputs aligned with expected program logic. The resulting low-to-moderate cyclomatic complexity values $V(G) \leq 5$ confirm that the internal procedural logic maintains structural maintainability and traceability for future code extensions.

4.2 Discussion

A comparative evaluation between pre-implementation and post-implementation workflows demonstrates a fundamental shift in operational coordination at Warkop Abas. Under the legacy manual framework, all transaction touchpoints were channeled through four floor attendants who simultaneously handled verbal ordering, kitchen coordination, order delivery, and bill processing. During peak hours, this concentration of tasks caused notable service bottlenecks, extending meal preparation intervals to 20–40 minutes and beverages to 10–30 minutes, while increasing order omissions. With the deployment of the digital platform, the front-end data logging burden shifts from service staff to the dining patrons themselves. Floor attendants are freed from routine recording tasks and can concentrate on order preparation and table service, while cashiers manage transactions through structured records on the Cashier Dashboard. Transaction logs also confirm that patrons actively utilize both QRIS and cash-at-counter settlement channels, proving that offering dual payment routes matches the payment habits of coffee shop consumers. The combined results of Black Box testing and White Box complexity analysis confirm that the platform achieves both external functional reliability and internal structural stability. Operationally, reducing order initiation latency allows customers to review the catalog and submit orders immediately upon seating, eliminating the initial waiting period for an attendant. For the business owner, automated database logging eliminates manual paper reconciliation and provides centralized transaction records that assist in tracking popular menu items and assessing daily revenue trends. Nevertheless, field evaluation revealed two technical limitations that represent opportunities for subsequent system enhancement. First, QRIS transaction verification remains partially manual, requiring the cashier to visually verify receipt uploads or account mutations because the system lacks direct integration with an automated payment gateway application programming interface (API). Second, the platform does not currently incorporate active push notification alerts, requiring cashiers to periodically observe the Cashier Dashboard to detect incoming tickets during rush periods. While these operational factors do not compromise the system's primary success in eliminating floor-level ordering bottlenecks, resolving them through automated webhook notifications and financial gateway integrations will further refine the end-to-end service cycle at Warkop Abas.

5. Conclusion and Recommendations

The website-based QR Menu and Self-Ordering system deployed at Warkop Abas successfully addresses operational dependencies on floor attendants, replacing vulnerable manual order logging with an automated, structured digital transaction workflow. Dining patrons can autonomously place orders by scanning table-specific QR codes without requiring prior user authentication, supported by flexible settlement options through QRIS or counter cash payment, while cashiers actively monitor and update order fulfillment stages via the real-time Cashier Dashboard. Functional Black Box evaluation across fourteen operational scenarios confirmed full behavioral validity, while White Box structural analysis across cashier authentication, checkout, ordering, and status modification routines yielded low-to-moderate cyclomatic complexity values, demonstrating manageable program logic and code maintainability. Operationally, the system compresses customer ordering latency, balances workload distribution between dining patrons and service staff, and establishes centralized transaction logs that facilitate systematic operational decision-making. Future investigations and system enhancements should prioritize integrating automated payment gateway application programming interfaces (APIs) to enable direct QRIS transaction verification without manual cashier reconciliation. Furthermore, the platform would benefit from automated notification triggers, such as webhook-driven messaging or audio alerts on the Cashier Dashboard, to ensure instantaneous order awareness during peak operational hours. Subsequent iterations may also explore automated raw ingredient inventory tracking, the development of lightweight progressive web or mobile applications for expanded cross-device flexibility, and advanced sales analytics dashboards to provide deeper business intelligence for enterprise owners.

References

Kaban, R., & Nasution, R. J. (2020). Penerapan metode Rapid Application Development (RAD) dalam perancangan sistem pemesanan menu menggunakan Quick Response (QR) Code. *MEANS (Media Informasi Analisa dan Sistem)*, *5*(2), 144–152. https://doi.org/10.54367/means.v5i2.920

Kevin, & Dewayani, E. (2024). Sistem pemesanan kopi keliling berbasis website. *Jurnal Sistem Informasi Galuh*, *2*(2), 83–92. https://doi.org/10.25157/jsig.v2i2.3969

Milenia, A. D. P., Oktorina, F. K., & Nasari, F. (2023). Implementasi sistem pemesanan menu di kafe dan resto Sunny Bangkinang menggunakan QR code berbasis website menggunakan framework CodeIgniter. *Jurnal Elektronik dan Teknik Informatika Terapan (JENTIK)*, *1*(3), 229–240. https://doi.org/10.59061/jentik.v1i3.395

Pranata, D., & Suharyanto, C. E. K. O. (2020). Implementasi aplikasi menu pemesanan dengan QR code pada warung kopi menggunakan metode FCFS berbasis Android. *Comasie*, *5*(3), 97–106.

Purba, E. N. (2022). Sistem informasi manajemen inventori JJ Cafe. *Method: Jurnal Teknik Informatika dan Sistem Informasi*, *8*(2), 1–4. https://doi.org/10.46880/mtk.v8i2.945

Rejeki, C. S., Septryanti, A., & Magdalena, H. (2025). Analisis pengaruh aplikasi pemesanan menu pada AFA Warkop menggunakan QR-code berbasis Android. *Technologia: Jurnal Ilmiah*, *16*(3), 469–477. https://doi.org/10.31602/tji.v16i3.19139

Rusito, R., & Putri, D. I. M. (2024). Implementasi QR code untuk sistem informasi pemesanan menu pada Restoran Omah Gedhe Pandean Kaliwungu berbasis web. *LogicLink*, *1*(2), 91–105. https://doi.org/10.28918/logiclink.v1i2.9164

Salsabiela, D. H. (2020). *Sistem pemesanan makanan berkonsep self-ordering dan berbasis web pada rumah makan* [Skripsi sarjana, Universitas Muhammadiyah Surakarta]. Repositori Institusi UMS. http://eprints.ums.ac.id/84478/

Sitepu, K. P., Gultom, F., Syaifuddin, & Gultom, R. P. (2025). Perancangan sistem informasi self-order dengan teknologi QR pada Healing Cafe. *Jurnal Cendekia Ilmiah*, *5*(1), 3679–3686.

Yanto, H. (2022). Sistem informasi administrasi café & resto berbasis web. *Jurnal Teknologi dan Sistem Informasi Bisnis*, *4*(1), 116–121. https://doi.org/10.47233/jteksis.v4i1.383