ISSN 2355-0082

Enhancing Operational Efficiency in FMCG Customer Care: Design and Implementation of a High-Performance Web-Based Knowledge Management System Using Next.js with Hybrid Rendering Architecture

Alghifari Rasyid Zola[1], Meredita Susanty[2]
[1,2] Computer Science, Faculty of Sains and Computer Science, Universitas Pertamina, Jakarta, Indonesia
[1]105222006@student.universitaspertamina.ac.id, [2]meredita.susanty@universitaspertamina.ac.id

Accepted 22 May 2026
Approved 17 July 2026

***Abstract*— In the highly competitive Fast-Moving Consumer Goods (FMCG) sector, the speed and accuracy of customer support are critical differentiators. This study addresses a significant operational bottleneck at PT Paragon Technology and Innovation, where reliance on fragmented, static documentation led to high information retrieval latency and service inconsistencies. To mitigate these issues, we designed and developed an integrated Knowledge Management System (KMS) serving as a Single Source of Truth (SSOT). Engineered using an Agile methodology, the system utilizes a modern Full-stack architecture based on the Next.js framework. It employs a hybrid rendering strategy, leveraging Server-Side Rendering (SSR) to optimize load performance and Client-Side Rendering (CSR) for interactive interfaces. The backend integrates a PostgreSQL database managed via Prisma ORM to ensure data integrity. Key features include a unified global search engine and a transparent audit trail mechanism for governance. Post-implementation evaluation was conducted through Lighthouse-based page-level performance testing and pilot usage observation. The proposed system achieved an average First Contentful Paint (FCP) of 0.35 seconds, Largest Contentful Paint (LCP) of 1.52 seconds, Total Blocking Time (TBT) of 20.00 ms, Speed Index of 0.82 seconds, and Cumulative Layout Shift (CLS) of 0.00 across repeated measurements. These results indicate that the proposed KMS provides fast initial rendering, responsive page behavior, and stable visual layout while supporting the transition from fragmented document-based workflows to a centralized customer-care knowledge platform.**

***Index Terms*— Agile Software Development; Customer Service; Knowledge Management System; Next.js; Server-Side Rendering; Client-Side Rendering; Single Source of Truth**

I. INTRODUCTION

The landscape of the Fast-Moving Consumer Goods (FMCG) industry has undergone a fundamental shift. In an era defined by hyper-connectivity, competitive advantage is no longer solely derived from product differentiation but increasingly from the quality of the customer experience [1]. As consumer touchpoints expand across social media, instant messaging, and e-commerce platforms, organizations are compelled to adopt "Omnichannel" strategies to maintain seamless engagement [2]. Consequently, Customer Care divisions have evolved from traditional support centers into critical hubs for brand loyalty and retention. In this high-velocity environment, the speed and accuracy of information retrieval by support agents are paramount metrics defining operational success.

The management of organizational knowledge to support these operations has been extensively studied under the domain of Knowledge Management (KM). Foundational theories by Nonaka and Takeuchi establish the critical need to convert tacit knowledge (internal expertise) into explicit knowledge (documented procedures) to scale organizational capability [6]. Further studies by Davenport and Prusak emphasize that effective KM systems (KMS) are essential for preventing knowledge loss and ensuring consistency in decision-making [8]. In customer care contexts, operational indicators such as Average Handle Time (AHT) and First Contact Resolution (FCR) are commonly used to assess service performance. However, measuring these indicators requires longitudinal customer interaction records, ticket resolution outcomes, and service quality data. Therefore, this study positions the proposed KMS as an initial technical implementation that focuses on page-level performance, knowledge centralization, and update governance as foundational elements for future operational performance evaluation [15], [16]. Traditionally, organizations have relied on static repositories, such as intranets, shared file servers, or basic document management systems to store Standard Operating Procedures (SOPs) and product details.

While these methods solve the basic storage problem, they often function as passive archives rather than dynamic tools optimized for real-time retrieval.

Despite the availability of general KM theories, a significant gap exists in their technical implementation within high-volume FMCG environments. A case study at PT Paragon Technology and Innovation (ParagonCorp), Indonesia's leading cosmetic manufacturer, reveals that rapid growth often outpaces infrastructure development [3], [14]. The Customer Care division currently faces a "knowledge fragmentation" crisis. Vital information regarding thousands of Stock Keeping Units (SKUs) and evolving protocols is dispersed across hundreds of static PDF documents and spreadsheets [4]. This decentralized, document-centric approach presents two critical failures. First, the retrieval latency gap: agents must manually search through multiple files to answer a single query, a process incompatible with the real-time demands of live chat support. Second, the integrity gap: without a centralized control mechanism, there is a high risk of agents accessing obsolete document versions, leading to service inconsistencies [5]. Existing "off-the-shelf" KM solutions often lack the specific customization required for the complex product hierarchies of FMCG or suffer from performance issues associated with client-side rendering latency in web applications.

To bridge this gap, this research proposes the design and development of a bespoke, high-performance Knowledge Management System tailored for the FMCG sector. Unlike traditional static repositories, this work introduces a Single Source of Truth (SSOT) architecture built on modern web technologies. We utilize the Next.js framework with a hybrid rendering architecture, leveraging Server-Side Rendering (SSR) for public-facing pages to ensure near-instantaneous data retrieval and superior performance compared to conventional Single Page Applications, while employing Client-Side Rendering (CSR) for interactive agent interfaces that require dynamic user interactions and real-time updates. [7]. The system integrates a relational database (PostgreSQL) with a unified global search engine that queries across multiple data entities, unifying disparate data points into a cohesive, searchable platform. By moving from a file-based to a data-based architecture, this study demonstrates how modern software engineering practices can resolve the trade-off between information volume and retrieval speed in customer care operations.

The novelty of this study lies in the integration of performance-oriented web architecture and operational knowledge governance within the FMCG customer care context. Prior knowledge management studies have emphasized the importance of converting tacit knowledge into explicit organizational knowledge and maintaining knowledge consistency through structured knowledge management systems [6], [8]. In customer support contexts, previous studies have also shown that knowledge-centric help desk systems can support problem resolution by integrating multiple knowledge sources into daily help desk operations [15]. Meanwhile, call center information system studies commonly evaluate operational success using broader service indicators such as system quality, user satisfaction, and net benefits [16].

Unlike prior KMS studies that primarily focus on knowledge storage, organizational learning, or help desk performance evaluation, this study proposes a real-time operational KMS tailored to FMCG customer care activities. The proposed system combines four key elements: a Single Source of Truth architecture to reduce knowledge fragmentation, hybrid rendering using SSR and CSR to support fast initial content delivery and interactive agent workflows, unified multi-entity keyword search across product and SOP data, and an audit trail mechanism to support transparency and accountability in knowledge updates.

The main contributions of this study are as follows. First, this study designs a web-based KMS architecture tailored to FMCG customer care operations. Second, it implements a hybrid-rendered Next.js application to support fast and interactive knowledge access. Third, it transforms fragmented PDF- and spreadsheet-based knowledge assets into a structured relational database with unified search capability. Fourth, it evaluates the system using Lighthouse-based technical performance metrics and pilot usage indicators.

II. METHODS

The development of the Knowledge Management System (KMS) adhered to a rigorous software engineering paradigm, prioritizing high scalability, performance optimization, and user-centric design. The project lifecycle followed an adapted Agile Scrum framework [9], characterized by weekly sprints and iterative refinement. This approach allowed for continuous validation of features against the evolving requirements of the operational team at PT Paragon Technology and Innovation.

A. Requirement Engineering and Gap Analysis

The study commenced with a comprehensive requirement engineering phase aimed at deconstructing the operational inefficiencies of the legacy workflow. Through semi-structured interviews with Team Leaders and direct observation of agents' daily interactions, specific functional gaps were identified. The primary imperative was the transition from a decentralized "document-based" retrieval system to a centralized "data-driven" search engine. As summarized in Table 1, the legacy system's reliance on static PDF files resulted in high retrieval latency and a lack of version control, whereas the proposed system necessitates granular indexing and transparency.

TABLE I. COMPARISON OF FUNCTIONAL CAPABILITIES BETWEEN THE LEGACY DOCUMENT-BASED WORKFLOW AND THE PROPOSED KNOWLEDGE MANAGEMENT SYSTEM.

Functional Aspect	Legacy System (PDF-Based)	Proposed System (Next.js SSR)
Data Storage Format	Unstructured Static Files (PDF, Excel) dispersed across local drives.	**Structured Relational Database** (PostgreSQL) centralized in the cloud.
Search Mechanism	**File-Level:** Agents search by file name; must open files to read contents.	**Content-Level:** Global Search queries across all text fields (titles, descriptions, content) for retrieval across multiple data entities.
Update Workflow	**Manual:** Manager edits file → converts to PDF → emails to team.	**Real-Time:** Manager edits via Dashboard → changes become available through the centralized system.
Version Control	**None:** High risk of agents utilizing outdated PDF versions.	**Automated:** System tracks all data modifications for accountability and transparency.
Accountability	**Low:** No record of who accessed or modified the documents.	**High:** AuditLog records every user action (Insert, Update, Delete).

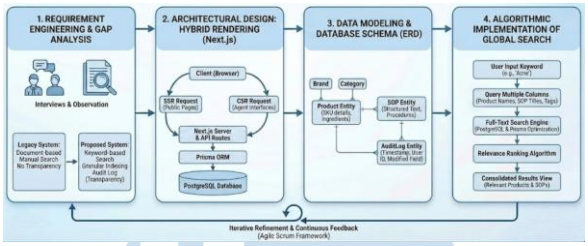

Fig. 1. Agile Software Engineering Approach for KMS Development

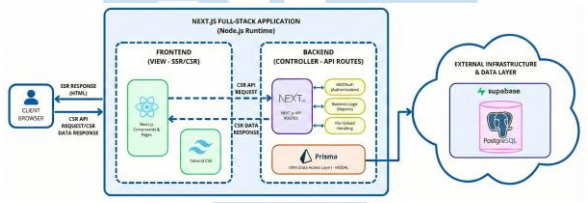

Fig. 2. High-Level System Architecture utilizing Next.js for Server-Side Rendering and Prisma ORM for database interaction

Agents required a system capable of indexing the specific contents of Standard Operating Procedures (SOPs) and product details, such as ingredients and usage instructions rather than merely retrieving file names. Furthermore, a critical non-functional requirement identified was "operational transparency." To ensure accountability, the system was required to log every modification to the knowledge base, a feature entirely absent in the previous workflow [10].

B. Architectural Design: The Full-Stack Monorepo Approach

To address the high-performance requirements of a fast-paced Customer Care environment, the system was built as a modern full-stack application. Figure 2 illustrates the high-level system architecture, highlighting the separation of concerns between the client-side presentation and server-side logic.

The core framework selected was Next.js, a production-grade framework for React [7]. The strategic decision to utilize Next.js was driven by its support for hybrid rendering architecture, combining Server-Side Rendering (SSR) and Client-Side Rendering (CSR) based on application requirements. For public-facing pages, the system leverages SSR, where the server pre-renders the complete HTML page with necessary data before transmission to the client, as depicted in the request flow in Figure 2. This approach significantly reduces the First Contentful Paint (FCP) time and improves initial load performance compared to traditional Single Page Applications (SPA) that download a minimal HTML shell and render content via JavaScript, often causing "hydration" delays when parsing large text datasets [13]. For interactive agent interfaces requiring dynamic user interactions, real-time updates, and state management, the system employs CSR to provide responsive user experiences. This hybrid approach ensures that agents have immediate access to information during live customer interactions while maintaining optimal performance for both public and authenticated user experiences.

For the backend infrastructure, the system leverages PostgreSQL, an advanced open-source relational database management system selected for its ACID compliance and robust data integrity [11]. To bridge the application layer and the database, Prisma ORM (Object-Relational Mapping) was employed. As shown in the architecture diagram, Prisma serves as the data access layer. It was chosen over raw SQL queries to enforce strictly typed database access, ensuring "type safety." This mechanism ensures that potential data

structure errors are caught during the compilation phase rather than causing runtime crashes, thereby enhancing overall system stability [12].

C. Data Modeling and Database Schema

The data modeling phase focused on constructing a relational structure capable of handling the complex, hierarchical nature of FMCG product portfolios. An Entity-Relationship Diagram (ERD) was constructed to map these complexities, as shown in Figure 3.

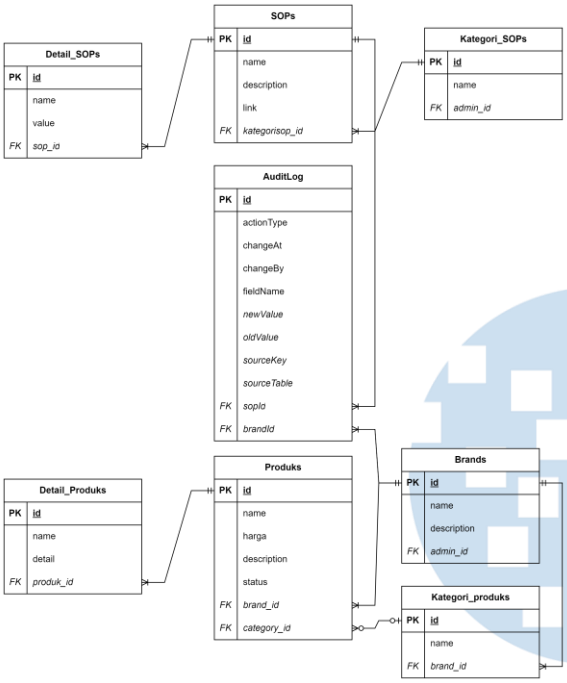

Fig. 3. Entity-Relationship Diagram (ERD) illustrating the relation structure between Products, SOPs, and the Audit Log mechanism.

The schema, visualized in Figure 3, comprises three core clusters:

1. Product Entity: Designed to store granular details of each Stock Keeping Unit (SKU). This entity maintains many-to-one relationships with *Brand* and *Category* entities, enabling the faceted navigation required by agents.
2. SOP Entity: Stores procedural knowledge. Unlike the legacy system, SOPs are stored as structured text blocks rather than binary files, enabling keyword-based search capabilities described in the subsequent section.
3. AuditLog Entity: A crucial component for system integrity. This entity is designed to automatically capture metadata for every INSERT, UPDATE, or DELETE operation. As illustrated in the schema, the log records the timestamp, the actor (User ID), and the specific field modified. This data model directly supports the organization's "One Voice Policy" by allowing supervisors to trace exactly when and by whom a piece of information was altered.

D. Algorithmic Implementation of Global Search

A pivotal feature of the methodology was the implementation of a unified Global Search algorithm. The global search mechanism was implemented using case-insensitive keyword matching across multiple table columns, including product names, descriptions, meta-tags, and SOP titles.

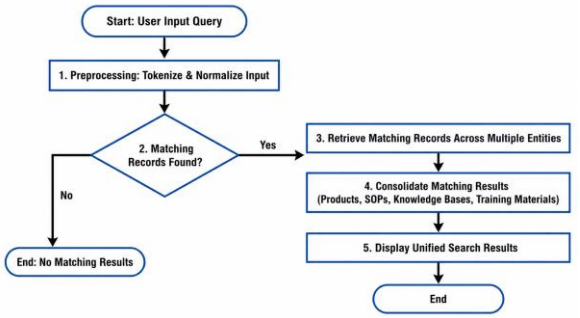

Fig. 4. Flowchart of the Global Search Algorithm using case-insensitive multi-entity keyword matching.

The logic flow, detailed in Figure 4, utilizes Prisma ORM's pattern matching capabilities with case-insensitive search across multiple database entities. The system performs parallel queries across different data types (products, SOPs, knowledge bases, and training materials) using the 'contains' operator, which enables flexible text matching. When an agent queries a keyword like "Acne," the system retrieves matching results from all relevant entities and consolidates them into a single, unified view. While the current implementation does not employ relevance-based ranking, the comprehensive search across multiple data sources ensures that agents can quickly access related products (e.g., acne serums) and corresponding handling procedures (SOPs) in one consolidated interface, thereby minimizing the agent's cognitive load by eliminating the need to search multiple separate systems.

E. Experimental Setup and Evaluation Procedure

The evaluation was designed to assess the front-end performance and page-level responsiveness of the proposed Knowledge Management System. The benchmarking process was conducted using Google Chrome Lighthouse, which provides standardized measurements for web performance indicators such as First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), Speed Index, Cumulative Layout Shift (CLS), and overall performance score.

The evaluation was conducted on the deployed KMS connected to a PostgreSQL database containing more than 1,000 SKU records and structured SOP entries. The tested pages represented common agent activities, including the main dashboard, product

knowledge page, SOP detail page, and update tracking page. These pages were selected because they represent the most frequently accessed areas of the system during daily customer care operations.

To reduce measurement bias, each tested page was evaluated using 30 repeated Lighthouse runs under the same testing environment. The repeated results were summarized using mean and standard deviation values. This approach was used to determine whether the reported performance values were consistent across multiple observations rather than being based on a single isolated measurement.

Since this study focuses on pilot implementation and front-end performance evaluation, concurrent-user load testing was not included in the current evaluation scope. Therefore, the reported results should be interpreted as page-level technical performance indicators rather than full server-side scalability measurements.

III. RESULTS

A. *System Implementation and User Interface Performance*

The culmination of the development phase resulted in the successful deployment of the "Customer Care Paragon Web" (CCPW), a centralized platform designed to consolidate the division's fragmented knowledge assets. The user interface (UI), constructed using the Tailwind CSS utility-first framework, provides a responsive and uniform experience across devices, as illustrated in Figure 5.

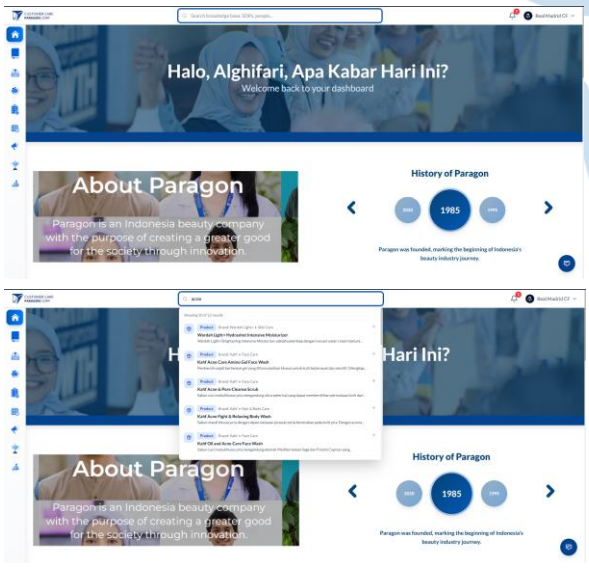

Fig. 5. The main dashboard of the Customer Care Paragon Web, featuring the centralized search bar and modular content categorization.

The implemented system transforms the previous document-based workflow into a centralized web-based knowledge management platform. In the legacy workflow, agents were required to manually search across multiple PDF and spreadsheet files, which increased retrieval steps and created a risk of accessing outdated information. In contrast, the proposed KMS provides centralized access to product information, SOPs, update tracking, and audit logs through a Single Source of Truth architecture. The operational and performance comparison between the legacy workflow and the proposed KMS is summarized in Table II.

TABLE II. PERFORMANCE AND OPERATIONAL COMPARISON: LEGACY WORKFLOW VS. PROPOSED KMS

Metric	Legacy System (PDF/Static Files)	Proposed System (Next.js SSR)	Improvement
Initial Page Load / FCP	4.2 seconds	0.35 seconds	approximately 92% faster
Information Retrieval Flow	Manual search across multiple PDF/spread sheet files	Centralized keyword-based search interface	Reduced retrieval steps
Update Propagation	Hours to days through manual redistribution	Near real-time update visibility through centralized database	Faster synchronization
Version Error Risk	High due to multiple document copies	Low due to Single Source of Truth architecture	Reduced outdated information risk
Update Accountability	No structured modification history	AuditLog records data changes	Improved governance

To validate the front-end performance of the proposed system, repeated Google Chrome Lighthouse measurements were conducted on the tested system interface. The evaluation focused on page-level performance indicators, including First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), Speed Index, and Cumulative Layout Shift (CLS). The results were summarized using mean and standard deviation values, as shown in Table III.

TABLE III. REPEATED LIGHTHOUSE PERFORMANCE EVALUATION RESULTS

Metric	N	Mean	Std. Dev.	Interpretation
First Contentful Paint (FCP)	30	0.35 s	0.28 s	Fast initial rendering
Largest Contentful Paint (LCP)	30	1.52 s	0.48 s	Good main content loading
Total Blocking Time (TBT)	30	20.00 ms	15.49 ms	Responsive interface
Speed Index	30	0.82 s	0.66 s	Fast visual loading, but varied
Cumulative Layout Shift (CLS)	30	0.00	0.00	Stable layout

As shown in Table III, the proposed KMS achieved strong page-level performance across 30 repeated Lighthouse measurements. The system recorded an average First Contentful Paint (FCP) of 0.35 seconds, Largest Contentful Paint (LCP) of 1.52 seconds, Total Blocking Time (TBT) of 20.00 ms, Speed Index of 0.82 seconds, and Cumulative Layout Shift (CLS) of 0.00. These results indicate that the system provides fast initial rendering, responsive page behavior, and stable visual layout for the tested interface. However, since the evaluation was limited to Lighthouse-based page-level testing, the results should be interpreted as front-end performance indicators rather than full server-side scalability or concurrent-user load testing results.

B. *Page-Level Performance Interpretation*

The Lighthouse evaluation indicates that the proposed KMS achieved fast page-level performance across repeated measurements. The average FCP of 0.35 seconds suggests that the system can render initial content quickly, while the LCP of 1.52 seconds indicates that the main content remains within an acceptable loading range. The TBT value of 20.00 ms shows that the interface remains responsive during loading, and the CLS value of 0.00 indicates stable visual layout. These results support the use of a hybrid rendering strategy for improving the front-end responsiveness of the proposed KMS. However, this evaluation does not measure API-level query latency, server-side throughput, or concurrent-user scalability.

C. *User Adoption and System Usage*

Post-deployment logs indicated a high adoption rate. Within the first month of implementation, the system recorded an average of 450 unique search queries per day. Additionally, the "Update Tracking" module logged 15 major SOP revisions during the pilot period, all of which were acknowledged by agents within 24 hours of publication. This provides quantitative evidence of the system's active role in daily information dissemination.

D. *Technical Interpretation: The Efficacy of SSR in Corporate Intranets*

The empirical results support the architectural decision to utilize a hybrid rendering approach that strategically combines Server-Side Rendering (SSR) and Client-Side Rendering (CSR) based on application requirements. While Banks and Porcello [13] argue that CSR offers smoother page transitions, this study argues that in a high-pressure Customer Care environment, the initial load time is the critical metric. The average FCP of 0.35 seconds supports the premise that optimized page delivery can reduce initial waiting time for agents. Simultaneously, the system leverages CSR for interactive agent interfaces where smooth transitions and real-time updates are essential, thereby combining the performance benefits of SSR with the interactivity advantages of CSR. This finding contributes to the growing body of literature advocating for context-aware, performance-first web architectures in enterprise tools that balance initial load performance with dynamic user experience requirements.

E. *The Impact of the Single Source of Truth (SSOT)*

Beyond technical metrics, this system facilitates a fundamental shift in organizational behavior, aligning with Nonaka and Takeuchi's Knowledge Creation Theory [6]. By transitioning from scattered PDF files to a centralized database—a Single Source of Truth (SSOT)—the organization successfully converts tacit knowledge (often held by senior managers who edit documents) into explicit, accessible knowledge available instantly to all agents. The transition to a Single Source of Truth (SSOT) architecture has fundamentally restructured the division's information workflow, effectively mitigating the risks associated with data fragmentation [4]. Figure 6 contrasts the legacy dissemination process with the new centralized model.

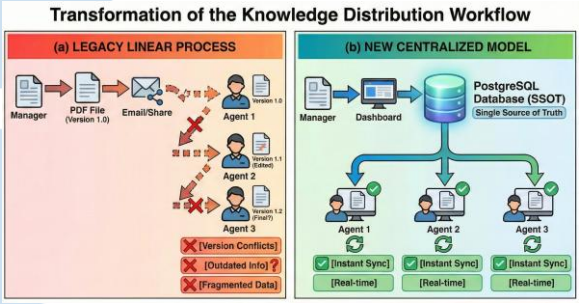

Fig. 6. Transformation of the Knowledge Distribution Workflow. (a) The legacy linear process prone to versioning errors. (b) The new centralized model supporting near real-time synchronization.

In the legacy model (Fig. 6a), updating a product specification was a multi-step manual process prone to human error and version control discrepancies. A manager was required to edit a master document, convert it to PDF, and redistribute it via disjointed communication channels such as email or chat groups. Conversely, the CCPW system (Fig. 6b) utilizes a centralized database architecture where updates committed by managers are immediately available to all users through a Single Source of Truth (SSOT). The "Update Tracking" feature, visualized on the agent dashboard, provides real-time notifications of recent changes through a polling mechanism that checks for updates every 30 seconds, ensuring agents are informed of modifications shortly after they occur. Upon login, agents are immediately presented with the latest dataset, and subsequent updates are detected through the notification system. This ensures that all agents, regardless of their physical location or shift, access an identical and synchronized dataset from the centralized database, thereby strictly adhering to the "One Voice Policy" and minimizing the risk of customer misinformation [5].

F. *Operational Transparency and Governance*

Beyond operational efficiency, the implementation of the AuditLog module has introduced a new layer of governance and transparency. Figure 7 presents the historical data view available to management, which details the chronological evolution of Standard Operating Procedures (SOPs).

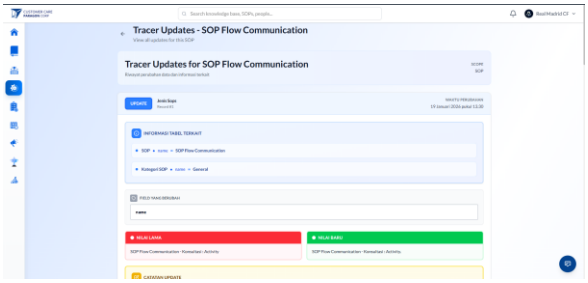

Fig. 7. The Audit Log interface displaying a chronological history of modifications to an SOP, ensuring data accountability.

This capability allows the quality assurance team to audit previous versions of service protocols and trace specific modifications to individual user accounts. By capturing metadata for every `INSERT`, `UPDATE`, and `DELETE` operation, the system fosters a culture of accountability [6]. This feature not only satisfies internal audit requirements but also facilitates continuous improvement by allowing management to analyze the frequency and nature of knowledge updates required to support dynamic market trends [3].

G. *Limitations and Future Work*

Despite its success, the current iteration has limitations. The system currently relies on manual data entry for stock levels, which introduces a potential latency between actual warehouse inventory and the information displayed to agents. Future development should focus on integrating an API connection to the company's ERP system to automate inventory updates. Additionally, the current search algorithm relies on exact keyword matching. Future work could explore Natural Language Processing (NLP) to handle semantic queries (e.g., understanding that "breakout" is related to "acne") to further enhance retrieval accuracy.

Another limitation of this study is that it has not yet measured direct customer-care performance indicators such as Average Handle Time (AHT), First Contact Resolution (FCR), agent productivity, and user satisfaction. The current evaluation focused on system design, implementation, and Lighthouse-based page-level performance testing. Measuring AHT and FCR requires longitudinal customer interaction logs, ticket resolution outcomes, and service quality records before and after implementation, which were outside the accessible scope of this pilot study. Therefore, the Lighthouse metrics and pilot usage observations in this study should be interpreted as initial technical feasibility indicators rather than direct evidence of customer service performance improvement. Future research should conduct a pre-post operational evaluation to measure whether the proposed KMS can reduce AHT, improve FCR, increase agent productivity, and enhance user satisfaction.

IV. CONCLUSION

This study sets out to address the critical operational challenge of knowledge fragmentation within the high-velocity Customer Care environment of the FMCG sector. The diagnostic phase revealed that the prevalent reliance on decentralized, static document repositories (PDFs and spreadsheets) created significant bottlenecks in information retrieval latency and jeopardized service consistency. To mitigate these issues, this research successfully engineered and deployed a dynamic, web-based Knowledge Management System (KMS) designed as a Single Source of Truth (SSOT).

The technical implementation demonstrates that modern full-stack web technologies can support enterprise-grade internal tools when aligned with operational requirements. By leveraging the hybrid rendering architecture of the Next.js framework, which combines Server-Side Rendering (SSR) for fast initial content delivery and Client-Side Rendering (CSR) for interactive interfaces, the system achieved an average First Contentful Paint (FCP) of 0.35 seconds in repeated Lighthouse measurements. These findings indicate strong page-level front-end performance compared to the legacy document-based workflow. Furthermore, the structured relational data model implemented via PostgreSQL and Prisma ORM successfully transformed disconnected product details and SOPs into a cohesive, searchable knowledge repository accessible through a unified keyword-based search mechanism.

Beyond technical metrics, the substantive impact of this work lies in the fundamental transformation of the organizational workflow. The shift from a manual, linear information dissemination process to a centralized database architecture with a Single Source of Truth (SSOT) has significantly reduced the risk of agents accessing obsolete data, as all users now access an identical and synchronized dataset. Updates committed to the database are immediately available to all agents upon their next data access, with periodic polling mechanisms ensuring agents are informed of changes within 30 seconds. The integration of granular Audit Logs has further introduced a necessary layer of digital governance, shifting the culture from reliance on tacit, individual knowledge to explicit, accountable organizational knowledge.

While the current iteration successfully centralizes static product data and procedural protocols, it is not without limitations. The system currently relies on manual updates for dynamic information, such as warehouse stock levels, creating a potential latency between data availability and reality. Consequently, future development work should prioritize the integration of external APIs to connect the KMS directly with the company's Enterprise Resource

Planning (ERP) or Inventory Management Systems for automated, real-time stock updates. Additionally, further research could explore enhancing the current keyword-based search algorithm with semantic analysis capabilities through Natural Language Processing (NLP) to better handle ambiguous agent queries.

However, the current evaluation was limited to Lighthouse-based page-level performance testing and pilot usage observation. This study did not measure direct customer-care indicators such as AHT, FCR, agent productivity, or user satisfaction. Therefore, the findings should be interpreted as evidence of technical feasibility and workflow improvement potential rather than a complete measurement of customer service performance improvement. Future work should incorporate longitudinal operational data to evaluate the direct impact of the KMS on AHT, FCR, and agent productivity.

ACKNOWLEDGMENT

We express our gratitude to PT Paragon Technology and Innovation for facilitating this case study, particularly Mrs. Nurul Rizka Hadiningsih for her industrial mentorship. We also appreciate the valuable input provided by the Customer Care team leaders and agents during the system's development and testing.

REFERENCES

[1] A. Rifazka, "Kemajuan Sektor FMCG di Indonesia Melalui Digitalisasi," Digital Transformation Indonesia, 2024. [Online]. Available: https://digitaltransformation.co.id/kemajuan-sektor-fmcg-di-indonesia-melalui-digitalisasi/. [Accessed: Jan. 15, 2026].

[2] P. C. Verhoef, P. K. Kannan, and J. J. Inman, "From multi-channel retailing to omni-channel retailing: Introduction to the special issue on multi-channel retailing," J. Retail., vol. 91, no. 2, pp. 174–181, Jun. 2015.

[3] D. Suryadi, "ParagonCorp, Lakukan Transformasi Customer Care Besar-besaran," SWA, 2024. [Online]. Available: https://swa.co.id/read/443343/paragoncorp-lakukan-transformasi-customer-care-besar-besaran. [Accessed: Jan. 15, 2026].

[4] B. Sandria, "11 Masalah Customer Service yang Sering Terjadi, Waspada!," Qiscus Insight, 2025. [Online]. Available: https://www.qiscus.com/id/blog/masalah-customer-service/.

[5] Khoros, "6 ways knowledge management improves customer service," Khoros Blog, 2025. [Online]. Available: https://khoros.com/blog/knowledge-management-for-customer-service.

[6] I. Nonaka and H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. New York, NY, USA: Oxford Univ. Press, 1995.

[7] Vercel, "Next.js Documentation: Introduction," Next.js, [n.d.]. [Online]. Available: https://nextjs.org/docs. [Accessed: Jan. 15, 2026].

[8] T. H. Davenport and L. Prusak, Working Knowledge: How Organizations Manage What They Know. Boston, MA, USA: Harvard Business School Press, 1998.

[9] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's Approach, 9th ed. New York, NY, USA: McGraw-Hill Education, 2019.

[10] Voxtron, "How to Improve Customer Service with Knowledge Management?," Voxtron Middle East, 2025. [Online]. Available: https://www.voxtronme.com/.

[11] T. Connolly and C. Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 6th ed. Boston, MA, USA: Pearson, 2014.

[12] Prisma, "Prisma ORM Documentation," Prisma.io, [n.d.]. [Online]. Available: https://www.prisma.io/docs. [Accessed: Jan. 15, 2026].

[13] A. Banks and E. Porcello, Learning React: Modern Patterns for Developing React Apps, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2020.

[14] A. Nugroho, "Industri Kosmetik Tumbuh Pesat, Dosen UGM Ingatkan Bahaya Pemakaian Kosmetika Palsu," Universitas Gadjah Mada News, 2025. [Online]. Available: https://ugm.ac.id/id/berita/industri-kosmetik-tumbuh-pesat/.

[15] L. M. González, R. E. Giachetti, and G. Ramirez, "Knowledge management-centric help desk: Specification and performance evaluation," Decision Support Systems, vol. 40, no. 2, pp. 389–405, 2005.

[16] H. A. Baraka, H. A. Baraka, and I. H. El-Gamily, "Assessing call centers' success: A validation of the DeLone and McLean model for information system," Egyptian Informatics Journal, vol. 14, no. 2, pp. 99–108, 2013.