IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5237~5250
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5237-5250

AI chatbot integration in enterprise resource planning systems for intelligent transaction and supply chain management

Mesran[1], Surya Darma Nasution[2], Guidio Leonarde Ginting[2]
[1]Department of Management, Sukma College of Management Sciences, Medan, Indonesia
[2]Department of Digital Business, Sukma College of Management Sciences, Medan, Indonesia

Article Info

Article history:

Received Nov 13, 2025
Revised Aug 12, 2026
Accepted Sep 20, 2026

Keywords:

ChatGPT API
Conversational enterprise resource planning
Enterprise resource planning
Flowise platform
Large language model integration
Real-time data interaction
Text-to-structured query language

ABSTRACT

The present study develops and evaluates an artificial intelligence (AI)-integrated enterprise resource planning (ERP) architecture that enables natural language interaction for real-time, data-driven decision support. The research aims to design a ChatGPT application programming interface (API)–based ERP integration using the Flowise platform to achieve enterprise-grade performance in accuracy, responsiveness, and scalability. The system was implemented using Laravel–Vue.js for the interface layer, Flowise for orchestration and workflow management, and MySQL for backend data handling. Performance evaluation involved sixty test trials across transactional, analytical, and predictive queries, supported by integration and error analyses to assess stability, security, and consistency. Experimental results showed an overall accuracy of 95.8%, an average response time of 3.2 seconds, and a consistency rate of 95.8%. The Flowise orchestration effectively optimized prompt routing, query validation, and secure communication, maintaining excellent performance under 100 concurrent users. Minor deviations were limited to rounding and predictive variations, with no syntax or execution errors detected. The proposed Flowise–ChatGPT–ERP integration demonstrates a technically viable and efficient framework for real-time enterprise data interaction, offering a scalable foundation for intelligent business analytics.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Surya Darma Nasution
Department of Digital Business, Sukma College of Management Sciences
Sakti Lubis Rd., Sitirejo II, Medan City, Medan, North Sumatera, Indonesia
Email: surya.darma.nasution1@gmail.com

1. INTRODUCTION

Enterprise resource planning (ERP) systems store massive volumes of operational data, ranging from financial transactions to inventory movements, procurement records, and supply chain activities. Despite this data richness, accessing actionable information remains challenging for end users. ERP users seeking simple information such as "Which products have the highest sales this month?" or "Display inventory items below the reorder point" are often required to navigate multiple interface layers, execute complex queries, or depend on information technology (IT) support. This accessibility gap leads to operational delays, suboptimal decision-making, and reduced organizational responsiveness. As ERP platforms continue to integrate cross-departmental business processes from accounting and finance to inventory and supply chain management which their data and functional scope expand significantly [1]. However, prior studies consistently highlight that increased integration simultaneously introduces navigational and cognitive complexity, limiting users' ability to access information in real time [2].

Consequently, while ERP systems excel at data consolidation, their interaction paradigms have not evolved to support efficient, user-centric data exploration.

Conventional strategies to improve ERP usability, including interface redesign, user training, and dashboard customization, have yielded incremental improvements but do not fundamentally alter how users interact with ERP systems. These systems remain dependent on hierarchical menus, predefined reporting templates, and structured query languages (SQLs) that require technical expertise. In parallel, artificial intelligence (AI) has been increasingly embedded into ERP back-end processes, enabling predictive analytics and resource optimization, as well as enhancing supply chain efficiency and adaptive decision making [3]–[5]. Nevertheless, these AI-driven advancements primarily target computational optimization rather than improving end-user accessibility to transactional data. At the same time, conversational AI has matured rapidly. Business chatbots now demonstrate advanced capabilities across multiple enterprise domains [6], [7], while large language models (LLMs) exhibit strong contextual understanding and reasoning abilities [8], [9]. Generative pre-trained transformers such as GPT-3 have shown notable few-shot learning performance, and pre-trained language models are effective in extracting structured information [10], [11]. Moreover, ChatGPT has demonstrated promising text-to-SQL capabilities, and chain-of-thought prompting has been shown to substantially improve reasoning accuracy [12], [13]. These developments suggest the potential for natural language interfaces to transform how users interact with complex information systems.

Despite these advances, the convergence of conversational AI and ERP systems remains limited. Existing ERP-focused AI research predominantly concentrates on backend optimization, while conversational AI studies largely address consumer-oriented or knowledge-based applications, without accounting for the transactional and relational complexity of enterprise databases. This limitation is particularly evident in ERP environments. Nascimento *et al.* [14] report that LLM performance degrades significantly when applied to real-world databases characterized by numerous tables, complex schemas, and dense foreign-key relationships. These characteristics are intrinsic to ERP systems, where transactional data spans multiple tightly coupled modules. As a result, strong benchmark performance in text-to-SQL tasks does not necessarily translate to authentic ERP contexts, revealing a critical empirical gap.

More importantly, prior studies have not explicitly addressed three fundamental limitations that hinder the practical adoption of conversational AI in ERP systems. First, there is no documented end-to-end architecture that integrates ChatGPT with ERP databases through orchestration platforms such as Flowise, leaving practitioners without reusable and deployable implementation blueprints. Second, although text-to-SQL approaches report promising results on general benchmarks [15], ERP-specific performance baselines for complex, multi-table transactional queries governed by business logic constraints remain largely unexplored. Third, existing work rarely employs systematic and reproducible evaluation protocols, instead relying on conceptual discussions or isolated proof-of-concept demonstrations. As a consequence, the feasibility, reliability, and performance of LLM-based conversational ERP interfaces remain uncertain. Unlike existing conversational ERP studies that primarily emphasize direct LLM integration or proof-of-concept implementations, this research positions Flowise as an orchestration layer responsible for coordinating prompt management, SQL generation, query validation, and secure interaction with ERP databases. In addition, rather than relying on generic text-to-SQL benchmarks, the proposed evaluation introduces ERP-specific benchmarking across transactional, analytical, and predictive query categories, thereby providing a more representative assessment of conversational AI performance within enterprise information systems. Collectively, these architectural and evaluation contributions establish a reusable implementation framework and distinguish the proposed approach from existing LLM-based ERP integration studies. Building upon these identified research gaps, the present study operationalizes its novelty through three specific contributions to AI-enabled enterprise systems: i) it proposes and implements a deployable end-to-end architecture that integrates the ChatGPT application programming interface (API) with ERP databases via the Flowise orchestration platform; ii) it establishes ERP-specific text-to-SQL performance benchmarks using complex, multi-table transactional queries reflecting realistic business logic; and iii) it introduces a systematic and reproducible evaluation protocol to assess accuracy, response time, and consistency of conversational ERP interactions. This research pursues two interrelated research questions to address these gaps.

i) RQ1: how ChatGPT API technically integrated with ERP systems through the Flowise platform for real-time transaction queries, and what architectural patterns enable this integration?
ii) RQ2: what levels of accuracy, response time, and consistency can the prototype achieve in interpreting various query types from simple transactional queries (STQ) to predictive analysis through a testing protocol with 10 trials per query type?

This study adopts a design science research (DSR) approach as proposed by Hevner *et al.* [16]. A conversational ERP prototype is developed and evaluated using 263 synthetic transactions that represent realistic operational scenarios for small- to medium-sized enterprises. This controlled experimental setting

enables precise measurement against known ground truth, ensures reproducibility, and avoids data privacy constraints. The main contributions of this study are as follows:

i) Proposing and implementing a Flowise-orchestrated architecture for integrating the ChatGPT API with an ERP system to enable natural language access to transactional enterprise data.
ii) Empirically demonstrating that the proposed architecture supports accurate, fast, and consistent natural language querying across simple, multi-table, and predictive ERP queries.
iii) Validating the enterprise readiness of conversational ERP systems through systematic evaluation of scalability, security, and operational reliability under realistic usage conditions.

2. METHOD

2.1. Research design

This research uses the DSR framework from Hevner *et al.* [16], through several stages: problem identification, solution objectives definition, prototype design and development, demonstration with synthetic data, and systematic evaluation. This study chose a prototype-based approach with a controlled experimental environment because: it enables precise accuracy measurement, it eliminates data privacy issues, and it guarantees internal validity. Although limiting generalization to the real world, this approach is appropriate for exploratory research aiming to establish technical feasibility and initial performance benchmarks.

2.2. System architecture and data flow

The overall architecture of the proposed AI chatbot–integrated ERP system is illustrated in Figure 1. The system adopts a modular and service-oriented design in which the conversational interface embedded within the ERP dashboard serves as the primary interaction layer for users. As shown in Figure 1, user queries are first transmitted to the Flowise orchestration layer, which enriches the input with contextual information, including database schema and query constraints, before forwarding the request to the ChatGPT API. ChatGPT generates structured SQL queries and corresponding natural language explanations, which are then validated and executed against the ERP database through the Laravel backend. The retrieved results are returned to the user via the ERP interface in conversational form, completing a bidirectional data flow between the user, AI services, ERP modules, and the database. This architecture ensures secure, interpretable, and scalable interaction between LLMs and enterprise data sources.

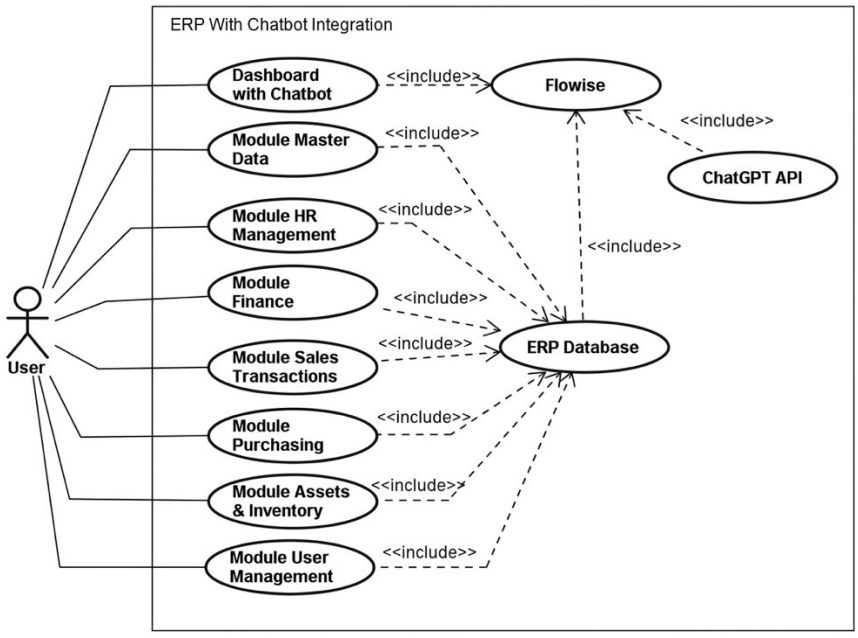

Figure 1. Architecture and data flow of the AI chatbot–integrated ERP system

2.3. Dataset creation

To evaluate the proposed AI chatbot-integrated ERP system, a synthetic dataset containing 263 transactions from January 2025 was developed to closely mirror realistic business operations commonly

found in small and medium enterprises (SMEs). The dataset was systematically constructed through four stages to ensure authenticity and internal consistency, including: i) master data was prepared, including customer, supplier, and product records. Customer and supplier names were generated using the Faker library with the Indonesian locale to ensure contextual relevance, while product data were manually curated to represent electronic retail inventories typical of Indonesian micro, small, and medium enterprises (MSMEs); ii) transaction data were generated, comprising approximately 130 sales orders and 80 purchase orders representing routine business activities. Sales quantities followed a Poisson distribution ($\lambda = 2$) to model typical consumer purchasing behavior, and transaction dates were uniformly distributed across working days in January 2025; iii) financial records were created by appending payment information to each transaction, with payment statuses assigned using a controlled distribution of 65% fully paid, 25% partially paid, and 10% pending, reflecting common cash-flow conditions in retail MSMEs; and iv) inventory movement data were simulated by linking stock inflows and outflows to sales and restocking processes. Each sales transaction generated a corresponding stock-out record, whereas stock-in records were triggered based on predefined reorder points, resulting in approximately 184 inventory movement records representing near real-time stock dynamics.

Comprehensive validation procedures were applied to ensure data accuracy, reliability, and business logic consistency. Referential integrity checks confirmed valid relationships among tables, while logical validation ensured that payment values remained within order limits and inventory levels never became negative. Additional validation included statistical distribution analysis, query coverage testing, and manual inspection of 20 randomly selected transactions. Following validation, the dataset was imported into a MySQL database to serve as the ground truth for evaluating SQL generation accuracy and overall chatbot performance. The one-month dataset was deliberately designed to provide a controlled yet sufficiently representative environment for evaluating conversational ERP queries while avoiding the privacy risks associated with operational enterprise data. Its scope was intentionally limited to sales and inventory management, enabling systematic assessment of conversational query generation within a representative retail SME scenario. Consequently, ERP domains such as human resources (HRs), financial accounting, procurement, and other cross-functional workflows were outside the scope of the present evaluation. Nevertheless, several limitations should be acknowledged. Synthetic data cannot fully reproduce real-world ERP characteristics, including missing values, duplicate records, inconsistent formats, legacy database structures, and undocumented business rules. In addition, the evaluated query set may not fully represent the diversity of real user interactions involving ambiguous language, domain-specific terminology, multi-intent requests, or long-term historical analyses. Although the present study employed a synthetic dataset to ensure experimental control and data privacy, the proposed architecture was designed to be directly deployable on anonymized ERP databases without requiring modifications to the orchestration workflow. The SQL generation pipeline, Flowise orchestration, and Laravel integration are independent of specific database instances, enabling straightforward adaptation to real organizational ERP systems. However, the current evaluation was limited to a controlled synthetic dataset representing retail SME operations and did not include enterprise-scale databases, high-concurrency workloads, or cross-module ERP scenarios.

2.4. Testing protocol, evaluation metrics, and data analysis

The dataset in this study was designed to evaluate the performance of an AI chatbot–integrated ERP system across varying levels of query complexity relevant to enterprise operations. Three query categories were defined: STQ for basic data retrieval, multi-table analytical queries (MTAQ) for aggregation and multi-join analysis, and predictive analytical queries (PAQ) for trend analysis and statistical reasoning. Six representative queries were systematically selected based on three criteria: coverage of increasing query complexity, representation of core ERP functions, and practical relevance documented in ERP implementation literature and practitioner surveys. These queries span single-table lookups, multi-table aggregations, and reasoning-based predictions, ensuring a balanced evaluation across complexity levels while covering essential ERP operations such as transaction tracking, demand forecasting, inventory monitoring and replenishment, sales performance analysis, and product optimization. Two queries were assigned to each complexity category, providing a structured yet representative benchmark while acknowledging that the selected queries constitute a focused subset of potential ERP interactions. Specifically, the current query set does not include HR management functions (e.g., payroll, attendance, or employee records), financial accounting operations (e.g., general ledger or accounts payable/receivable), or multi-step transactional workflows that span multiple ERP modules. These domains were intentionally excluded to maintain experimental control and to isolate the system's core conversational reasoning and text-to-SQL capabilities. Nevertheless, this limitation applies only to the experimental evaluation rather than to the proposed system architecture. Because the Flowise–ChatGPT framework operates by leveraging database schema information and structured SQL generation instead of module-specific business logic, the proposed architecture is

designed to support extension to HR, finance, procurement, and other ERP modules through appropriate schema configuration and prompt adaptation without requiring fundamental architectural modifications. Accordingly, future work will extend the experimental evaluation to HR, finance, procurement, and cross-module workflows to further assess the generalizability and scalability of the proposed conversational ERP framework under broader operational scenarios.

Six representative queries as in Table 1 covering key ERP functions such as stock monitoring, sales reporting, and forecasting were executed ten times under controlled conditions with a stable network, fixed database state, consistent testing time, cleared cache, and thirty-second intervals, resulting in a total of sixty executions to ensure statistical reliability. System performance was evaluated using accuracy rate, response time, and consistency score. Accuracy was determined based on data correctness, logic, completeness, and explanation quality. Response time was measured from query submission to display, and consistency was assessed across repeated trials. Ground truth data were prepared by an expert SQL developer, validated by a second researcher, and used to benchmark chatbot outputs for accuracy and reliability. Data analysis was conducted using a quantitative approach through descriptive statistical methods, focusing on response time, accuracy rate, and consistency across multiple query executions to evaluate the system's overall performance and reliability. The evaluation was intentionally conducted under controlled low-concurrency conditions to assess functional correctness, SQL generation accuracy, response consistency, and session stability. Consequently, system-level performance metrics such as latency percentiles, throughput, and error rates under high-concurrency workloads were outside the scope of the present experimental protocol.

Table 1. Six representative ERP queries used for system evaluation

No	Category	Query	Complexity
1	STQ	"What is the total transactions on January 15, 2025?"	Low
2	PAQ	"Predict 'Xiaomi 14 Series' product sales for February 2025 from January data"	High
3	STQ	"Display 10 products with lowest stock"	Low
4	STQ	"Display products below minimum stock"	Low
5	MTAQ	"Display 10 products with highest sales January 2025"	Medium
6	MTAQ	"Display 10 products with lowest sales January 2025"	Medium

3. RESULTS AND DISCUSSION

3.1. Result

3.1.1. Technical integration architecture of ChatGPT API with ERP through Flowise platform

The integration of the ChatGPT API with a Laravel–Vue. js-based ERP system through the Flowise platform represents a pivotal advancement in the evolution of intelligent enterprise systems. These findings reveal that LLM-based architectures can function as intelligent mediators between users and complex ERP databases, transforming traditional static dashboards into interactive, conversational interfaces. This approach aligns with recent research emphasizing that LLMs enhance data interaction and accessibility by allowing users to retrieve structured business insights through natural language queries [17]. Such capability not only democratizes data access across organizational levels but also supports a shift toward adaptive, user-centered ERP environments. The Flowise-driven orchestration layer further strengthens this interaction by managing data flow and prompt contextualization, thereby maintaining accuracy and transparency in AI responses (Figure 2).

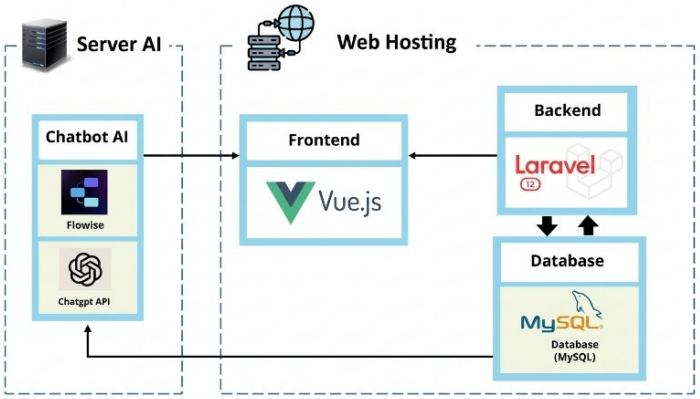

Figure 2. Internal architecture of the flowise-based AI chatbot for ERP integration

The ERP module structure developed in this study comprises seven main components such as master data, HR management, finance, sales transactions, purchasing, assets and inventory, and user management that each interconnected through a relational database designed to reflect realistic SME operations with proper referential integrity and embedded business rules. This database configuration ensures consistency and operational accuracy across all modules while enabling the AI chatbot to dynamically query, retrieve, and analyze information through natural language interactions. The integration of these modules within an AI-driven ERP architecture further enhances automation, reliability, and scalability in enterprise workflows. Flowise acts as the intermediary layer that facilitates modular interoperability between the ChatGPT engine, the Laravel backend, and the MySQL database, ensuring that user inputs are seamlessly translated into validated SQL queries. This design supports the automation of routine data operations, aligning with Ekatpure's findings [18] that LLMs can revolutionize ERP functionalities through intelligent error handling and process automation. Moreover, the modular structure of the ERP system mirrors the microservices-oriented frameworks identified in prior research, which promote incremental AI adoption and operational flexibility in small and medium-sized enterprises [19]. Overall, the integration demonstrates a balanced approach between performance efficiency and system scalability, allowing AI components to be deployed strategically without disrupting existing ERP processes.

The findings reveal that the Flowise workflow architecture plays a central role in governing how user inputs are processed and translated into meaningful database operations within the AI chatbot-integrated ERP system. As illustrated in Figure 3, this workflow follows a structured chain of thought pattern beginning from the chat model node (ChatGPT 4.1), which processes enriched user prompts using predefined prompt templates that include database schema, contextual examples, and query guidance. The workflow then proceeds through the MySQL connector, enabling query execution, and integrates an SQL parser and security validator to ensure compliance with data access policies before generating coherent responses through the response formatter and conversation memory. This layered orchestration aligns with Cadet *et al.* [20] who emphasize that compliance-oriented automation frameworks are vital to maintaining governance boundaries in AI-driven systems. Moreover, the security mechanisms embedded in this architecture, including read only database access, query timeouts, row limits, and activity logging, reflect the implementation of controlled automation principles designed to safeguard data integrity and confidentiality. From an architectural standpoint, the system's API based and modular design resonates with Chamari *et al*. [21] who highlight the importance of service oriented models for facilitating seamless communication between AI agents and enterprise systems. Similarly, the synergistic interaction between ChatGPT and Flowise functions as an intelligent middleware that unifies the modular ERP components and enhances interoperability across fragmented enterprise infrastructures [22]. Collectively, these results underscore that effective AI ERP integration requires not only technical interoperability and process automation but also a well-structured orchestration layer that maintains transparency, security, and control across enterprise workflows.

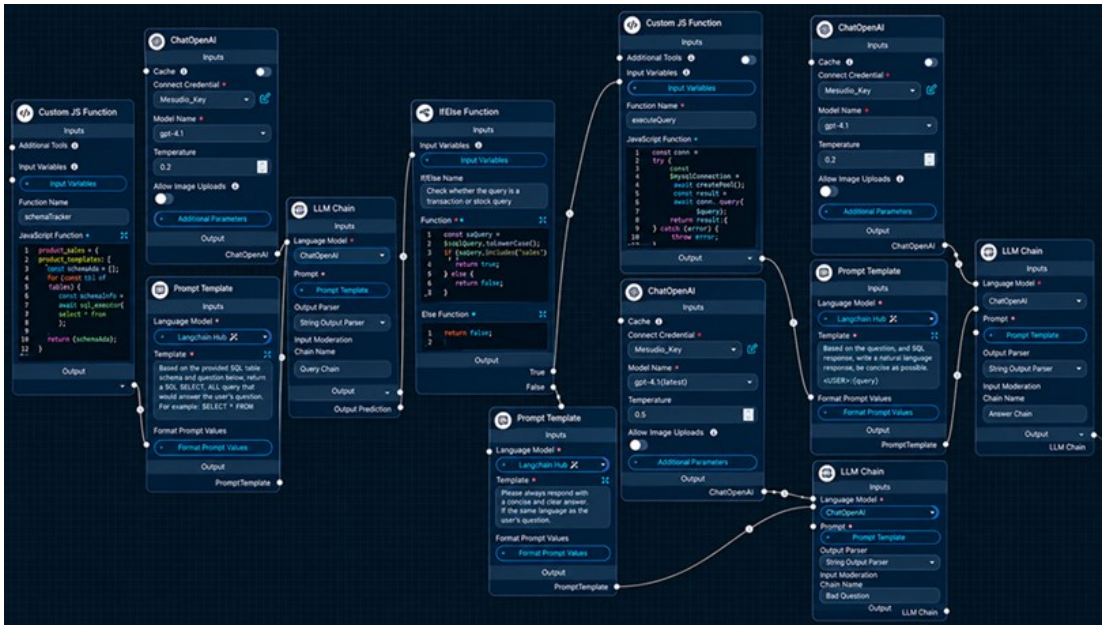

Figure 3. Internal architecture of the Flowise-based AI chatbot for ERP integration

The integration process begins when a user submits a query through the chatbot interface. The Flowise API processes the request by enriching it with relevant database schema information before sending it to the ChatGPT model, which generates an SQL query and a natural language explanation. The system validates and secures the query before execution on the MySQL database, and the results are presented to the user in conversational form within two to five seconds. This workflow demonstrates seamless collaboration between AI and ERP components, ensuring secure, accurate, and efficient enterprise interactions. As shown in Figure 4, the chatbot is integrated into the ERP dashboard, allowing users to access real-time data through simple natural language commands. For example, when asked for the top ten products with the highest sales in January 2025, the chatbot instantly provides details such as product names, sales totals, prices, and stock levels, supporting fast and informed decisions. The theoretical foundation of this integration aligns with socio-technical systems theory, which emphasizes the balance between human interaction and automated processes in improving usability and decision-making. Likewise, system integration theory explains how modular, interoperable architectures such as Flowise enable scalable and adaptive enterprise operations without disrupting existing logic. From a human–computer interaction (HCI) perspective, the conversational interface illustrates a shift toward cognitive usability and intuitive engagement, consistent with studies showing that chat-based systems enhance adoption and satisfaction [23]. Overall, this cloud-based architecture establishes a foundation for next-generation ERP systems that are adaptive, transparent, and capable of supporting intelligent, real-time decision ecosystems [24].

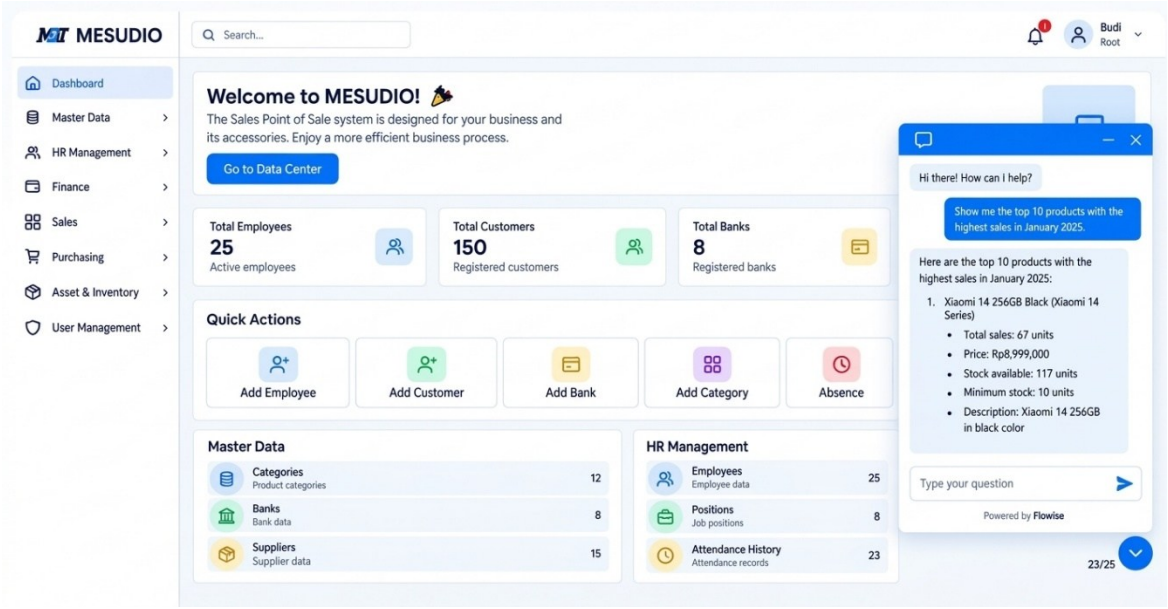

Figure 4. Conversational interface in AI chatbot-integrated ERP system

3.1.2. System performance evaluation

The performance evaluation of the AI chatbot integrated ERP system was conducted using six representative queries executed ten times each, resulting in sixty total trials. These tests were designed to assess the system's reliability and robustness across different query complexities, including STQ, MTAQ, and PAQ. As summarized in Table 2, the system achieved strong overall performance across all categories, maintaining high levels of accuracy, low response time, and stable consistency. These findings confirm that the system performs effectively within enterprise-grade standards, ensuring reliable and efficient data-driven interactions.

The chatbot achieved an overall accuracy rate of 95.8% across sixty test runs, correctly responding to 57.5 out of 60 trials, as shown in Table 2. The STQ category reached 95% accuracy, with minor deviations in transaction value calculations, while MTAQ queries achieved a perfect 100% accuracy, effectively handling complex multi-table joins and aggregations. The PAQ category obtained 90% accuracy, exhibiting small variations that remained within acceptable forecasting ranges. Overall, these results demonstrate that integrating the ChatGPT API into an ERP system through the Flowise platform delivers enterprise-level performance in terms of accuracy, response speed, and reliability, with an average response time of

3.2 seconds and a consistency rate of 95.8%. To contextualize this performance, baseline comparisons were conducted against two conventional ERP query approaches: manual SQL querying by an experienced developer and dashboard-based filtering and aggregation using standard ERP reporting interfaces. For each of the six representative queries, equivalent SQL scripts were manually written and executed on the MySQL database, while dashboard-based queries relied on predefined filters, grouping options, and summary views. Although manual SQL execution achieved high accuracy, it required substantially longer task completion times due to query formulation and verification overhead, particularly for multi-table analytical and predictive queries. Dashboard-based querying enabled faster execution for simple transactional lookups but showed limited flexibility and increased interaction steps when handling complex aggregations and forecasting-related tasks.

Table 2. Performance results across all ERP query categories

No	Category	Question	Accuracy (%)	Response time	Consistency (%)
1	STQ	Total transactions Jan 15, 2025	85	2.9 seconds	85
2	PAQ	Predict Xiaomi 14 sales Feb 2025	90	4.3 seconds	90
3	STQ	10 products lowest stock	100	2.7 seconds	100
4	STQ	Products below minimum stock	100	2.8 seconds	100
5	MTAQ	10 products highest sales January	100	3.2 seconds	100
6	MTAQ	10 products lowest sales January	100	3.1 seconds	100

In contrast, the Flowise–ChatGPT approach consistently achieved comparable or higher accuracy with significantly reduced user effort and interaction time, especially for MTAQ and PAQ categories, highlighting its effectiveness in managing complex ERP queries through natural language interaction. These findings are consistent with recent studies reporting that advanced frameworks such as retrieval-augmented generation (RAG), LangChain, and Flowise enhance contextual accuracy by combining data retrieval, workflow management, and modular orchestration [25], [26]. Compared to alternative orchestration frameworks such as LangChain and RAG, the Flowise-based architecture emphasizes low-code visual workflow orchestration, integrated SQL validation, and fine-grained control over query execution pipelines. While prior studies indicate that LangChain and RAG can improve contextual accuracy through embedding-based retrieval and tool chaining, they often introduce additional latency and configuration complexity due to vector indexing, retriever tuning, and multi-stage reasoning pipelines.

In contrast, Flowise provides a more streamlined orchestration path optimized for structured ERP databases, resulting in lower orchestration overhead and more predictable response times in transactional enterprise environments. Such architectural characteristics reduce computational delays, enhance scalability, and support stable real-time interactions across complex enterprise systems. The Flowise–ChatGPT setup exemplifies these principles through secure authentication, efficient caching, and effective context control. Overall, the combined empirical evidence and theoretical insights indicate that Flowise-integrated LLMs offer a reliable and scalable foundation for modern ERP environments, improving real-time analytics, decision-making, and operational efficiency [27], [28].

Response time analysis, summarized in Table 3, demonstrates the system's operational efficiency with an average end-to-end latency of 3.2 seconds across all query types, ranging from 2.7 to 4.5 seconds—well within the "excellent" threshold of under five seconds. STQ averaged 2.8 seconds, while more complex analytical and predictive queries required 3.1 to 4.3 seconds. To strengthen the statistical validity of the performance evaluation, additional descriptive and inferential analyses were conducted. Mean response times were complemented with standard deviations and 95% confidence intervals across repeated executions. STQ exhibited the lowest variance, whereas PAQ showed slightly higher dispersion due to increased AI inference complexity. A one-way analysis of variance (ANOVA) confirmed statistically significant differences in response time across query categories (STQ, MTAQ, and PAQ) ($p < 0.05$). Accuracy results similarly demonstrated low variance across trials, with narrow confidence intervals indicating stable and reproducible performance across all query types. These findings confirm that the observed performance differences are systematic and related to query complexity rather than random variation.

Table 3. Average response time by query category

Category	Average	Range
STQ	2.8 seconds	2.7 - 2.9
MTAQ	3.15 seconds	3.1 - 3.2
PAQ	4.3 seconds	4.1 - 4.5
Overall	3.2 seconds	2.7 - 4.5

The decomposition of processing time shows that AI inference contributed largest share (60 – 70%), followed by database execution (15 – 20%), network latency (10 – 15%), and result formatting (5 – 10%). This distribution reflects a balanced and optimized orchestration pipeline, in which Flowise effectively manages computational tasks between AI processing, database execution, and network communication. Such efficiency aligns with recent advancements in AI-driven ERP architectures emphasizing multi-layered and cloud-based orchestration to reduce latency and computational load [29]. Furthermore, Flowise's modular design supports node caching and prompt reuse, resembling similarity-based retrieval mechanisms that enhance throughput and minimize delay [30].

The integration testing was conducted to evaluate the reliability, security, and scalability of the AI chatbot integrated ERP system under realistic operational conditions. Four critical aspects were examined including mobile device responsiveness, multi-user performance, security mechanisms, and data export functionalities. These all aspects achieved excellent results, as summarized in Table 4. From a performance standpoint, the system demonstrated seamless responsiveness across smartphones and tablets, with all front-end components rendering correctly and maintaining full interactivity, confirming its adaptability to diverse screen sizes and operating systems. Scalability tests involving over 100 concurrent users showed that the system sustained stable performance with no latency or session conflicts, supported by a role-based access control module that effectively managed user permissions. These outcomes highlight the robustness of the system's microservice-based architecture, which aligns with hybrid service-oriented architecture (SOA) and RESTful API integration frameworks proposed by Palanichamy [27]. In terms of security and data integrity, token-based authentication, parameterized SQL queries, and structured logging operated effectively, ensuring the prevention of SQL injection attempts and unauthorized access. Authentication tokens expired as intended, while chatbot interactions were limited to read-only database operations, preserving data consistency. The use of asynchronous API calls and optimized query caching further contributed to the absence of latency spikes during high-load scenarios. Additionally, data export functions performed with precision and completeness across Excel, PDF, and CSV formats, supporting business intelligence and analytical workflows. Collectively, these findings confirm that the proposed system achieves enterprise-grade standards of scalability, security, and reliability.

Table 4. Integration testing results for scalability, security, and functionality

Aspect	Evaluation criteria	Performance status
Mobile device responsive	Element and component alignment across devices	Excellent
Multi-user access	System stability on handling 100+ concurrent users	Excellent
Security features	Token authentication, expiration handling, SQL injection prevention, logging integrity	Excellent
Data export functionality	Accuracy and completeness of exported data (Excel, PDF, CSV)	Excellent

For the error analysis, the AI chatbot–integrated ERP system demonstrated a high level of reliability and precision. As shown in Table 5, only three minor errors were observed across 60 total trials, resulting in an overall error rate of 5%. Numerical calculation deviations accounted for 3.3% of cases and were primarily associated with aggregation-based queries, where rounding operations and implicit type conversions occurred during AI-generated SQL formulation. These deviations did not indicate logical or semantic errors, but rather minor numerical discrepancies inherent to floating-point arithmetic and result formatting. Prediction-related variations (1.7%) were observed exclusively in the predictive analytical query, reflecting the probabilistic nature of demand forecasting and its sensitivity to historical data patterns.

Table 5. Error types observed during system evaluation

Error type	Frequency	Impact
Numerical calculation	2/60 (3.3%)	Small (<20% deviation)
Prediction variation	1/60 (1.7%)	Small (reasonable range)
SQL syntax error	0/60 (0%)	None
Semantic misunderstanding	0/60 (0%)	None
Execution failure	0/60 (0%)	None

The predictive query exhibited greater variability than transactional and analytical queries because the ChatGPT model performs probabilistic reasoning rather than deterministic database retrieval. Unlike SQL-based queries that return exact records, predictive responses depend on historical trends and inference patterns. Consequently, small deviations are expected even when the generated predictions remain within acceptable analytical ranges. Importantly, no errors were detected in SQL syntax, semantic interpretation, or

query execution, confirming that the integration between Flowise, ChatGPT, and the MySQL database operated smoothly and consistently produced valid outputs. This behavior aligns with Hettiarachchi's framework on relation-aware embeddings and RAG, which enhances semantic understanding and minimizes misinterpretations in LLM-driven systems [28]. Compared to deterministic transactional and analytical queries, predictive queries inherently introduce higher uncertainty; however, the observed deviations remained within acceptable analytical tolerances and did not compromise decision-level validity, reaffirming the semantic stability and computational soundness of the proposed architecture.

Based on the observed error patterns, several targeted mitigation strategies are proposed to further strengthen system robustness. Numerical calculation deviations can be reduced through prompt optimization that explicitly enforces rounding rules, numeric precision constraints, and deterministic aggregation instructions within SQL generation templates. For predictive variations, hybrid forecasting approaches that combine LLM reasoning with traditional statistical or time-series methods (e.g., moving averages or regression-based forecasting) can improve stability while preserving explainability. Additionally, confidence-aware response formatting, in which predictive outputs are accompanied by uncertainty ranges, can enhance decision reliability. These strategies directly address the identified sources of deviation and are grounded in the empirical findings of this evaluation, providing a practical pathway for real-world ERP deployment.

Overall, the integration of ChatGPT, Flowise, and ERP technologies demonstrates a technically robust and theoretically coherent model for real-time, AI-assisted enterprise data interaction. The system's balance of interpretability, speed, and reliability is supported by empirical performance results and aligns with emerging research on adaptive orchestration and hybrid predictive modeling [31]. Building on the error-aware mitigation strategies identified in this study, adaptive prompt optimization, precision-aware SQL generation, and hybrid forecasting emerge as evidence-driven refinements rather than speculative future enhancements. Collectively, this architecture establishes a scalable, secure, and intelligent foundation for operationalizing generative AI within ERP ecosystems, effectively bridging conversational intelligence and structured business analytics.

3.2. Discussion

This study demonstrates that the integration of LLMs into ERP systems through a structured orchestration layer can significantly enhance enterprise data interaction, particularly for SMEs. From a socio-technical systems perspective, the findings support the principle that system performance and organizational value emerge from the alignment between technical infrastructure and human interaction rather than from technological advancement alone [32]. The consistently high accuracy, low response time, and stable performance across multiple query categories indicate that the proposed Flowise–ChatGPT–ERP architecture successfully optimizes both technical execution and user engagement. By reducing dependence on rigid dashboards and specialized technical skills, the system enables broader participation in data-driven decision-making, reinforcing the social subsystem of ERP usage. From a HCI standpoint, the conversational interface evaluated in this study addresses longstanding usability challenges associated with traditional ERP systems. Conventional ERP dashboards often impose high cognitive load due to complex menu hierarchies, static reporting structures, and limited flexibility in query formulation. The empirical results show that natural language interaction allows users to retrieve both simple and complex analytical information efficiently and consistently, even for multi-table and predictive queries. This observation aligns with previous findings that mentioned natural language interfaces improve learnability, reduce mental workload, and enhance perceived control in complex information systems [33]–[35]. The low variance in accuracy and response time across repeated trials further suggests that conversational querying supports not only intuitive interaction but also reliable and predictable system behavior, which is critical for enterprise environments.

The findings also contribute to the growing body of research on orchestration frameworks for enterprise AI integration. Compared to manual SQL querying and traditional ERP dashboards, the Flowise-based orchestration layer provides a structured yet flexible mechanism for translating user intent into validated database operations. The absence of SQL syntax errors, semantic misinterpretations, and execution failures across all trials indicates that low-code orchestration can effectively mitigate risks commonly associated with direct LLM–database interaction. This supports recent studies emphasizing the importance of workflow orchestration and process-aware AI in ensuring reliability, governance, and scalability of intelligent enterprise systems [35]. The predictable response time observed across query categories further reinforces the suitability of orchestration-based architectures for real-time operational decision support. Beyond theoretical contributions, the results yield several concrete practical implications for SMEs, including: i) the findings indicate that user training should prioritize the formulation of clear, goal-oriented natural language queries rather than technical SQL proficiency. This reduces training costs and lowers adoption barriers, which are critical constraints for SMEs; ii) governance mechanisms play a central role in

responsible deployment. The effective use of read-only access controls, role-based permissions, and interaction logging in this study demonstrates that conversational ERP systems can be adopted without compromising data integrity or auditability. These findings align with emerging views on trustworthy and accountable AI, where explainability is essential for building user trust, supporting managerial oversight, and enabling responsible decision-making [36]–[39]; iii) the results support a phased deployment strategy for AI-assisted ERP adoption. While transactional and analytical queries exhibited near-perfect accuracy, predictive queries showed slightly higher variability due to the probabilistic nature of forecasting. This suggests that SMEs should initially deploy conversational interfaces for reporting, inventory monitoring, and sales analysis before extending usage to predictive decision support. Incremental adoption allows organizations to build trust in AI systems while minimizing operational risk.

Furthermore, the modular design of the Flowise-based architecture enables scalable integration of AI capabilities without requiring major restructuring of existing ERP workflows. This finding is consistent with digital transformation literature emphasizing modularity and incremental innovation as key success factors for SMEs [39]. Although the present evaluation was limited to sales, inventory, and forecasting scenarios, the modular and schema-driven design of the proposed architecture enables straightforward extension to HR, finance, procurement, and other ERP modules, thereby supporting broader enterprise adoption without substantial architectural.

Beyond technical performance, successful deployment of conversational AI within enterprise environments also requires careful consideration of ethical and governance issues. From an ethical and governance perspective, the proposed conversational ERP framework emphasizes responsible AI deployment by preserving data privacy and organizational accountability. The use of synthetic data during system evaluation eliminates risks associated with exposing sensitive enterprise information, while future deployment on anonymized ERP datasets is intended to maintain confidentiality. In addition, the framework employs read-only database access, SQL validation, authenticated API communication, and activity logging to support traceability, prevent unauthorized data modification, and ensure that AI-generated responses remain auditable and subject to human oversight. These mechanisms contribute to transparent and accountable AI-assisted decision support within enterprise environments.

Although the proposed framework demonstrated stable multi-user operation during controlled testing, the evaluation was intentionally conducted under low-concurrency conditions. The reported average response time of 3.2 seconds therefore represents end-to-end application latency under controlled experimental settings rather than latency under enterprise-scale workloads. Accordingly, the reported performance primarily reflects functional correctness, session isolation, and conversational stability. Latency percentiles, throughput, and error rates under higher concurrency loads were not quantified, because each conversational query depends on external LLM API calls whose response times may vary independently of the internal Flowise orchestration pipeline. Future work should therefore conduct systematic load testing across multiple concurrency levels while separately reporting end-to-end latency, orchestration latency, throughput, and error rates to provide a more comprehensive evaluation of enterprise-scale performance.

Several limitations of this study must be acknowledged. The evaluation relied on a synthetic ERP dataset, which ensures reproducibility and data privacy but may not fully capture the complexity and variability of real-world enterprise data. Additionally, the set of test queries, while representative of core ERP functions, reflects a limited range of possible user interactions and does not fully address ambiguous or multi-intent queries commonly encountered in practice. Moreover, the controlled experimental environment does not account for real-world variability such as fluctuating network conditions, large-scale concurrent workloads, or evolving data distributions. In addition, the predictive evaluation was limited to short-term forecasting using a controlled historical dataset and therefore does not fully reflect long-term forecasting performance under seasonal demand fluctuations, evolving business trends, or changing operational conditions. These limitations suggest important directions for future research. Empirical validation using anonymized real ERP datasets is necessary to assess system robustness under authentic operational conditions. Furthermore, pilot deployment and validation on large-scale enterprise ERP systems with substantially larger transaction volumes, concurrent users, and complex cross-module operations are required to further evaluate the scalability, enterprise readiness, and generalizability of the proposed architecture. Extending the system to support multilingual natural language queries would enhance applicability in diverse SME contexts, particularly in non-English-speaking regions. Furthermore, integrating explainable artificial intelligence (XAI) mechanisms into conversational ERP interfaces could strengthen user trust and governance by clarifying how queries are interpreted and how results are generated.

4. CONCLUSION

This study confirms that integrating the ChatGPT API with an ERP system through the Flowise platform provides a reliable and efficient framework for real-time enterprise data interaction. The system,

built using Laravel–Vue.js, Flowise, and MySQL, achieved an overall accuracy of 95.8%, an average response time of 3.2 seconds, and a consistency rate of 95.8% across diverse query types. These results demonstrate that the architecture can handle both simple and complex transactional queries with high precision and stability. Performance evaluation showed that Flowise effectively managed prompt orchestration, query validation, and secure data routing without significant latency. Integration tests confirmed excellent scalability, mobile responsiveness, and security, with stable operation under more than 100 concurrent users. Error analysis revealed only minor deviations related to rounding and predictive variance, with no syntax or execution errors. Overall, the system demonstrates strong accuracy, speed, and robustness, validating the Flowise–ChatGPT–ERP integration as a practical solution for secure, real-time decision support in enterprise environments.

ACKNOWLEDGMENTS

This research was supported by the Directorate of Research, Technology, and Community Service (DRTPM) through the 2025 fundamental research grant, conducted under the Management Study Program, Sukma College of Management Sciences. The authors express their sincere gratitude to DRTPM and Sukma College of Management Sciences for their continuous support and facilitation throughout the research process.

FUNDING INFORMATION

This study was funded by the Directorate of Research, Technology, and Community Service (DRTPM), Indonesia, in 2025 through the fundamental research grant awarded to the Management Study Program at Sukma College of Management Sciences under grant agreement no. 65/SPK/LL1/AL.04.03PL/2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mesran	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Surya Darma Nasution	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Guidio Leonarde Ginting	✓	✓	✓	✓		✓	✓			✓	✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

This study did not involve human participants. All experimental data were generated synthetically to simulate enterprise transactions and supply chain scenarios.

ETHICAL APPROVAL

The study used a controlled synthetic dataset with no human involvement and was approved by the authors' institutional review board at Sukma College of Management Sciences.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [SDN], upon reasonable request.

REFERENCES

[1] S. Katuu, "Enterprise resource planning: past, present, and future," *New Review of Information Networking*, vol. 25, no. 1, pp. 37–46, 2020, doi: 10.1080/13614576.2020.1742770.

[2] R. Dubey, A. Gunasekaran, S. J. Childe, C. Blome, and T. Papadopoulos, "Big data and predictive analytics and manufacturing performance: integrating institutional theory, resource-based view and big data culture," *British Journal of Management*, vol. 30, no. 2, pp. 341–361, 2019, doi: 10.1111/1467-8551.12355.

[3] Z. N. Jawad and V. Balázs, "Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review," *Beni-Suef University Journal of Basic and Applied Sciences*, vol. 13, no. 1, pp. 1–13, 2024, doi: 10.1186/s43088-023-00460-y.

[4] G. Culot, M. Podrecca, and G. Nassimbeni, "Artificial intelligence in supply chain management: a systematic literature review of empirical studies and research directions," *Computers in Industry*, vol. 162, 2024, doi: 10.1016/j.compind.2024.104132.

[5] R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, and M. Fischl, "Artificial intelligence in supply chain management: a systematic literature review," *Journal of Business Research*, vol. 122, pp. 502–517, 2021, doi: 10.1016/j.jbusres.2020.09.009.

[6] E. Adamopoulou and L. Moussiades, "Chatbots: history, technology, and applications," *Machine Learning with Applications*, vol. 2, 2020, doi: 10.1016/j.mlwa.2020.100006.

[7] Y. Zhang, R. Y. K. Lau, J. D. Xu, Y. Rao, and Y. Li, "Business chatbots with deep learning technologies: state-of-the-art, taxonomies, and future research directions," *Artificial Intelligence Review*, vol. 57, no. 5, pp. 1–63, 2024, doi: 10.1007/s10462-024-10744-z.

[8] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, "Generative AI," *Business and Information Systems Engineering*, vol. 66, no. 1, pp. 111–126, 2024, doi: 10.1007/s12599-023-00834-7.

[9] J. Yang *et al.*, "Harnessing the power of LLMs in practice: a survey on ChatGPT and beyond," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 6, 2024, doi: 10.1145/3649506.

[10] T. B. Brown *et al.*, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, no. 159, pp. 1877–1901, 2020, doi: 10.5555/3495724.3495883.

[11] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, "Pre-trained language models and their applications," *Engineering*, vol. 25, pp. 51–65, 2023, doi: 10.1016/j.eng.2022.04.024.

[12] Z. Hong *et al.*, "Next-generation database interfaces: a survey of LLM-based text-to-SQL," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 12, pp. 7328–7345, 2025, doi: 10.1109/TKDE.2025.3609486.

[13] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *36th International Conference on Neural Information Processing Systems*, 2022, vol. 35, pp. 24824-24837, doi: 10.52202/068431-1800.

[14] E. R. Nascimento *et al.*, "LLM-based text-to-SQL for real-world databases," *SN Computer Science*, vol. 6, no. 2, 2025, doi: 10.1007/s42979-025-03662-6.

[15] M. Pourreza and D. Rafiei, "DIN-SQL: decomposed in-context learning of text-to-SQL with self-correction," *arXiv:2304.11015*, 2023.

[16] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly: Management Information Systems*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

[17] K. Gangapatnam, "Revolutionizing enterprise resource planning through AI integration: a technical deep dive," *European Journal of Computer Science and Information Technology*, vol. 13, no. 15, pp. 55–69, 2025, doi: 10.37745/ejcsit.2013/vol13n155569.

[18] S. R. Ekatpure, "Revolutionizing ERP systems: the integration of AI and large language models in manufacturing and retail," *Journal of Artificial Intelligence & Cloud Computing*, vol. 2, no. 1, 2023, doi: 10.47363/jaicc/2023(2)e187.

[19] R. T. Balla, "Enabling cognitive process automation using LLMs in ERP systems with generative AI," *World Journal of Advanced Engineering Technology and Sciences*, vol. 15, no. 3, pp. 1467–1474, 2025, doi: 10.30574/wjaets.2025.15.3.1045.

[20] E. Cadet *et al.*, "Developing scalable compliance architectures for cross-industry regulatory alignment," *Shodhshauryam International Scientific Refereed Research Journal*, vol. 7, no. 3, pp. 141–176, 2024, doi: 10.32628/shisrrj2472145.

[21] L. Chamari, P. Pauwels, and E. Petrova, J. W. Dubbeldam, N. D. Jong, and K. M. Gunderi, *Reference architecture for smart buildings*. Delft, Netherlands: Brains4Buildings, 2023.

[22] J. Chen, H. Lin, X. Han, and L. Sun, "Benchmarking large language models in retrieval-augmented generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024, vol. 38, no. 16, pp. 17754–17762, doi: 10.1609/aaai.v38i16.29728.

[23] C. Kooli, "Chatbots in education and research: a critical examination of ethical implications and solutions," *Sustainability*, vol. 15, no. 7, 2023, doi: 10.3390/su15075614.

[24] S. Sarferaz, "Implementing conversational AI into ERP software," *IEEE Access*, vol. 13, pp. 160238–160250, 2025, doi: 10.1109/ACCESS.2025.3608477.

[25] S. M. Kuo, S. K. Tai, H. Y. Lin, and R. C. Chen, "Automated clinical trial data analysis and report generation by integrating retrieval-augmented generation (RAG) and large language model (LLM) technologies," *AI*, vol. 6, no. 8, pp. 1–22, 2025, doi: 10.3390/ai6080188.

[26] Y. S. Low *et al.*, "Answering real-world clinical questions using large language model, retrieval-augmented generation, and agentic systems," *Digital Health*, vol. 11, pp. 1–11, 2025, doi: 10.1177/20552076251348850.

[27] R. T. Palanichamy, "Technical architecture for data synchronization between ERP, WMS, and TMS systems," *International Journal of Multidisciplinary Research and Growth Evaluation*, vol. 5, no. 5, pp. 1117–1121, 2024, doi: 10.54660/.ijmrge.2024.5.5.1117-1121.

[28] I. Hettiarachchi, "Orgverse: a temporal and multidimensional LLM framework for enterprise data integration and decision," *Journal of Artificial Intelligence General Science*, vol. 8, no. 02, pp. 245–261, 2025, doi: 10.60087/jaigs.v8i02.419.

[29] M. Priyadarshi, "Multi-layered scalable architecture diagram for AI-powered agentforce integration within Salesforce enterprise ecosystem," *European Modern Studies Journal*, vol. 9, no. 3, pp. 474–482, 2025, doi: 10.59573/emsj.9(3).2025.42.

[30] M. Zhang, B. Yuan, H. Li, and K. Xu, "LLM-cloud complete: leveraging cloud computing for efficient large language model-based code completion," *Journal of Artificial Intelligence General Science*, vol. 5, no. 1, pp. 295–326, 2024, doi: 10.60087/jaigs.v5i1.200.

[31] R. Li *et al.*, "Enhancing pulmonary disease prediction using large language models with feature summarization and hybrid retrieval-augmented generation: multicenter methodological study based on radiology report," *Journal of Medical Internet Research*, vol. 27, 2025, doi: 10.2196/72638.

[32] G. Baxter and I. Sommerville, "Socio-technical systems: from design methods to systems engineering," *Interacting with Computers*, vol. 23, no. 1, pp. 4–17, 2011, doi: 10.1016/j.intcom.2010.07.003.

[33] E. P. Nittala, "Leveraging large language models for natural language interface in ERP systems: a case study in user productivity and cognitive load," *International Journal of Emerging Trends in Computer Science and Information Technology Eureka Vision Publication*, vol. 5, no. 4, pp. 125–131, 2024, doi: 10.63282/3050-9246.IJETCSIT-V5I4P113.

[34] J. H. Nderitu, "Mental state adaptive interfaces as a remedy to the issue of long-term, continuous human machine interaction," *Journal of Robotics Spectrum*, vol. 1, pp. 78–89, 2023, doi: 10.53759/9852/jrs202301008.
[35] W. V. D. Aalst, "Process mining: overview and opportunities," *ACM Transactions on Management Information Systems*, vol. 3, no. 2, 2012, doi: 10.1145/2229156.2229157.
[36] A. Rai, "Explainable AI: from black box to glass box," *Journal of the Academy of Marketing Science*, vol. 48, no. 1, pp. 137–141, 2020, doi: 10.1007/s11747-019-00710-5.
[37] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, 2019, doi: 10.1145/3236009.
[38] National Institute of Standards and Technology, *Artificial intelligence risk management framework (AI RMF 1.0)*, Gaithersburg, United States: NIST, 2023, doi: 10.6028/NIST.AI.100-1.
[39] G. Vial, "Understanding digital transformation: a review and a research agenda," *Journal of Strategic Information Systems*, vol. 28, no. 2, pp. 118–144, 2019, doi: 10.1016/j.jsis.2019.01.003.

BIOGRAPHIES OF AUTHORS



Mesran earned his bachelor's degree in Computer Science in 2002 and master's degree in Computer Science in 2008 from Universitas Putra Indonesia "YPTK" Padang. He is currently a permanent lecturer at Sukma College of Management Sciences, focusing his teaching and research on information systems, decision support systems, and cryptography. As an active researcher, he has produced numerous scientific publications in reputable national and international journals. His research interests include the development of intelligent computing methods for decision-making, data security based on modern cryptography, and the application of information systems in various fields of management and business. He can be contacted at email: mesran.skom.mkom@gmail.com.



Surya Darma Nasution learned his bachelor's degree in Computer Science from STMIK Budi Darma and master's degree in Computer Science at Universitas Putra Indonesia "YPTK". Since 2011, he has been active as a lecturer and researcher in the field of Computer Science in the Department of Digital Business at Sukma College of Management Sciences. His research focuses on data compression, decision support systems, and cryptography. He is also involved in applied research on the use of AI in education and business. His current interests include informatics, AI, and the development of intelligent and secure information systems. He can be contacted at email: surya.darma.nasution1@gmail.com.



Guidio Leonarde Ginting earned his bachelor's degree in Computer Science (S.Kom) from Universitas Budi Darma, Medan, and later obtained his master's degree in Computer Science from the same institution. Since 2011, he has been active as a lecturer and researcher in the field of Computer Science. His research interests include data compression, decision support systems, and cryptography, as well as the application of AI and intelligent systems in education and business. He has published several scientific papers in national and international journals and is currently affiliated with Sukma College of Management Sciences. He can be contacted at email: guidio.leonard626@gmail.com.