NUANSA INFORMATIKA
Volume 19 Nomor 1, Januari 2025

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

Design and Implementation of a RESTful API-Based Point of Sale System

Fulandi Hudza Grahitama[1], Waskitho Cito Adiwiguno[2], Syafrial Fachri Pane[3]
[1,2,3] Applied Bachelor Program of Informatics Engineering, Vocational School, Universitas Logistik dan Bisnis Internasional, Jl. Sariasih No.54, Sarijadi, Kec. Sukasari, Bandung, West Java 40151, Indonesia

E-mail: syafrial.fachri@ulbi.ac.id

Abstract

Point of Sale (POS) systems are essential for modern businesses, streamlining transactions, inventory management, and customer interactions. However, traditional POS systems face challenges such as limited real-time data processing, scalability issues, and restricted integration capabilities. This study proposes a RESTful API-based POS system using Supabase and Express.js to overcome these limitations.The system is developed using a hybrid waterfall methodology, combining structured phases with iterative refinement, and employs a relational database normalized to the third normal form (3NF) for data integrity and scalability. Supabase, as a backend-as-a-service platform, simplifies backend operations with its robust features for database management, authentication, and real-time APIs. Meanwhile, Express.js provides a lightweight and efficient framework for developing RESTful APIs, ensuring seamless integration and efficient data handling. Comprehensive testing, including black box testing, confirms the system's reliability, ensuring its readiness for real-world implementation. The results highlight the system's ability to enhance operational efficiency and adapt to dynamic business requirements. This study demonstrates how integrating RESTful APIs, Supabase, and Express.js can modernize POS systems, providing scalable, secure, and efficient solutions tailored to the demands of a data-driven marketplace.

Keywords* : *Point of Sale (POS), RESTful API, Supabase, Express.js

Submission: 24 January 2025 *Accepted: 30 January 2025* *Published: 31 January 2025*

1. INTRODUCTION

In the era of globalization, the exponential growth in population has resulted in an unprecedented surge in the volume of data generated on a daily basis. This data has evolved into a critical asset for businesses, serving not only as a foundation for strategic decision-making but also as a driver of operational efficiency and innovation [1], [2]. As organizations strive to remain competitive and responsive in an ever-changing market landscape, the adoption of advanced technological solutions has become indispensable. This trend is particularly evident in areas such as transaction management and customer engagement, where streamlined processes and data-driven insights are essential for enhancing productivity, improving customer satisfaction, and fostering sustainable growth [3], [4].

Point of Sale (POS) systems have become pivotal in modern business operations, functioning as central hubs for streamlining transaction processing, optimizing inventory management, and enhancing customer interaction [4]. These systems play an important role in enabling businesses to maintain efficiency and accuracy in their everyday activities [5]. However, traditional POS systems are not without challenges. They frequently encounter limitations, including inefficiencies in handling real-time data, restricted flexibility when integrating with other systems, and diminished performance under heavy data loads [6]. These shortcomings can hinder the ability of businesses to scale operations, adapt to dynamic market demands, and leverage data-driven insights effectively, thereby underscoring the need for innovative, more robust POS solutions tailored to contemporary business needs [3], [7], [8].

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

To address these challenges, modern technological solutions such as RESTful (Representational State Transfer) APIs are increasingly being adopted. As a standard, The RESTFUL web service is a web service that is built on the REST architecture, enabling seamless integration and efficient real-time data management [9]. RESTful APIs offer a robust mechanism for integrating diverse system components, including databases, hardware, and software platforms [10]. This integration facilitates efficient real-time data management and allows for seamless operation across various platforms [11]. By leveraging RESTful APIs, businesses can develop more adaptable, scalable, and secure POS systems, which align with the dynamic needs of the digital age [12].

In addition to RESTful APIs, Supabase provides a robust solution to enhance modern POS systems. As an open-source backend-as-a-service (BaaS), Supabase offers seamless tools for managing and scaling relational databases, authentication, and real-time APIs. Its intuitive interface and strong integration capabilities enable developers to build secure and efficient systems, making it ideal for data-driven applications in POS environments. By simplifying backend operations, Supabase caters to the growing demand for scalable and reliable POS solutions, allowing businesses to focus on delivering value to their customers [13].

Meanwhile, Express.js, a simple and adaptable Node.js web application framework, is widely utilized for building RESTful APIs in POS systems. Its lightweight design and extensive middleware ecosystem make it particularly effective for managing server-side logic and processing HTTP requests, and ensuring efficient data exchange. Express.js is an excellent choice for businesses seeking cost-effective and high-performing API solutions. By leveraging Express.js, developers can create tailored POS systems that address specific operational requirements and enhance overall efficiency [14], [15].

Additionally, the integration of RESTful APIs in POS systems enhances business agility, allowing businesses to react quickly to shifts in the market and client needs. Effective implementation strategies, including robust security measures and structured API lifecycle management, are critical for ensuring the operational success and longevity of such systems. Moreover, the adoption of cloud technology in POS systems further amplifies their potential, providing scalability and reducing the operational complexities faced by businesses [16], [17].

In conclusion, implementing RESTful APIs in modern POS systems is a strategic step toward overcoming traditional system limitations. This innovation not only enhances operational efficiency but also positions businesses to thrive in a highly competitive and data-driven marketplace.

2. METHODOLOGY

The hybrid waterfall methodology blends the structured, sequential nature of the traditional waterfall model with the adaptability of iterative refinement. The traditional waterfall model is a system development method in which each stage cannot proceed until the previous step is completed, ensuring a linear progression and thorough focus on foundational elements [18]. Based on figure 1, the hybrid approach maintains this structured phase progression, such as data collection, system analysis, system implementation and testing, while incorporating feedback loops during testing [19]. These feedback mechanisms allow for adjustments in the design and planning stages to address issues and improve outcomes [20]. This combination of structured phases and iterative flexibility makes hybrid approaches well-suited for managing diverse and dynamic project requirements [21].

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

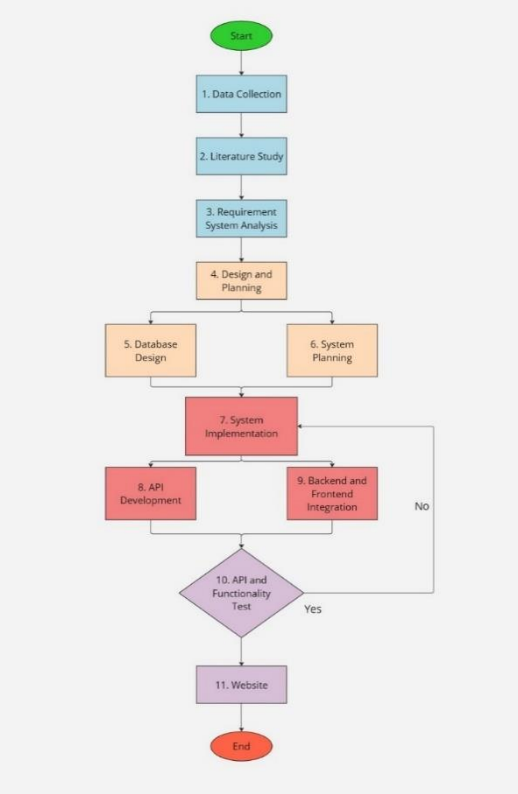

Figure 1 Application Creation Procedure

This iterative refinement allows for continuous improvement, ensuring the final product meets requirements and functions as intended. Blending the structure of the waterfall model with the adaptability of iterative processes offers a balanced approach, suitable for dynamic and complex project environments [22]. Research highlights that hybrid models, which integrate Agile and Waterfall methodologies, optimize project delivery by combining structured planning with iterative development cycles, making them ideal for projects requiring a balance of control and flexibility [23], [24]. Moreover, iterative methodologies such as Agile are known for their ability to handle unforeseen challenges, especially in adaptive software environments.

Studies comparing iterative development with traditional Waterfall processes emphasize that the former excels in environments requiring regular refinement, particularly in software design and quality evaluation [25]. Furthermore, the inclusion of feedback loops has been demonstrated to improve system effectiveness and alignment with project objectives [26]. This methodology ultimately ensures rigorous planning while remaining responsive to emerging challenges, aligning with best practices for iterative software development and hybrid methodologies [27].

3. RESULTS

3.1 Analysis

The analysis step defines the system requirements analysis procedure. This research's system requirements study includes identifying the planned systems, functional and non-functional requirements, and analyzing user needs to ensure the system meets both technical specifications and user expectations. It also examines how the system will interact with other systems, the performance criteria it must meet, and the necessary resources for successful implementation.

3.2 Planning

This design is made into two parts, namely database design and system design that uses UML diagrams and creates a system interface design.

3.2.1 Database Design

Database design is an important part of Point of Sale (POS) system development, as it ensures that data can be managed effectively and efficiently. This database is designed to support a Point of Sale (POS) system with five main tables. Table 1 Users stores user data, such as id, name, email, role, phone number, and activation status. Table 2 Products records product information, including product id, name, price, stock, image, and category id linked to Table Categories in Table 3, which organizes products by category. Table 4 Transactions records transactions, including transaction id, user id, total amount, payment method, date, customer name, and table number. Details of each item in a transaction are stored in Table 5 Transaction Items, covering item id, transaction id, product id, quantity, unit price, and total cost. The relationships between tables enable efficient management of products, users, and transactions within the POS system. In this chapter, the database structure designed to record transactions, manage products, and generate reports is explained.

user_id (PK)	name	email	password	role	phone_number	is_active

Table 1 Users

id_product(PK)	name	description	price	stock	id_categories(FK)	picture

Table 2 Products

Id_categories (PK)	name	description

Table 3 Categories

id_ transactions (PK)	id_ users (FK)	total_ amount	payment_ method	transaction_ date	customer_ name	customer_ phone	table_ number

Table 4 Transactions

id_transaction_items (PK)	id_transactions (FK)	id_products (FK)	quantity	unit_price	total_amount

Table 5 Transaction Items

Based on figure 2 the database was designed using a relational model, with interconnected tables to record transaction data, products, users, and sales reports. The database is designed to 3rd normal form (3NF) to reduce data duplication. For example, product and category data are separated in separate tables to facilitate data management and updates.

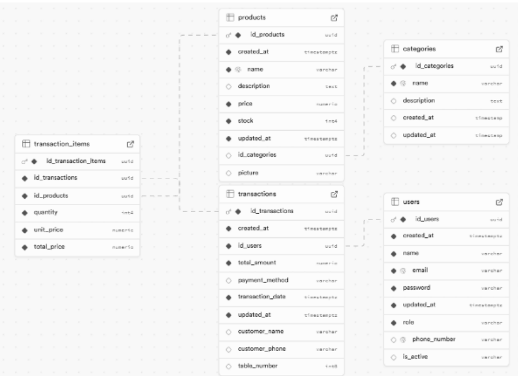

Figure 2 Relationships Between Tables

3.2.2 System Planning

UML diagrams, which include use case, activity, sequence, and class diagrams, are used in system design. The Use Case Diagram is a description of the activities carried out by actors or actors in the POS system.

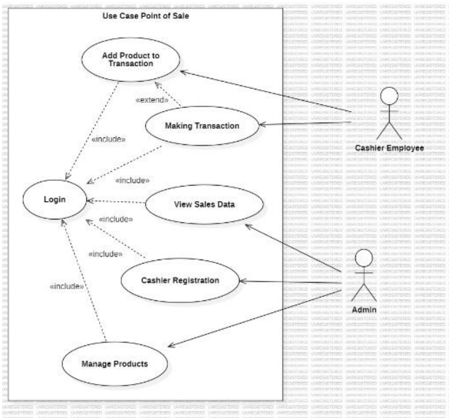

Figure 3 Use Case Diagram

Based on figure 3, the use case diagram that represents a Point of Sale (POS) system with two primary actors: the cashier and the admin. The cashier handles transaction-related tasks, including logging in, adding products to transactions, and processing checkout. Adding products is an extension of the checkout process, as it occurs before finalizing the sale. The admin manages system functions, including viewing sales data, managing cashier accounts, and overseeing product information, such as updating prices or stock. The login use case is included in all activities to ensure secure access and role-based functionality. The diagram highlights the relationships between use cases, with mandatory steps represented by <<include>> relationships, while optional

p-ISSN :1858-3911 , e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

extensions are shown with <<extend>>, ensuring flexibility and modularity in the system's design

4. DISCUSSIONS

The Web-Based POS Application Program was built using Node.js, and Express Framework to create the RESTful API in the backend, while frontend using JavaScript to create the interface.

4.1.1 Backend

The backend handles essential features such as user authentication, product management, sales transactions, and sales reports. Each functionality is accessible through RESTful API endpoints. Table 6 is an example of a request body and table 7 show response for adding a new transactions.


```
{
  "total_amount": "80000",
  "payment_method": "cash",
  "customer_name": "Uca",
  "customer_phone": "6287781574213",
  "table_number": "7",
}
```


Table 6 Request Body


```
{
  "id_transactions": "2e10cde6-001f-4987-9323-4f3cdd38566a",
  "created_at": "2025-01-13T02:51:33.689Z",
  "id_users": "a07638cf-af8c-42db-a9ed-8b1a8da06049",
  "total_amount": "80000",
  "payment_method": "cash",
  "transaction_date": "2025-01-13T02:51:31.716Z",
  "updated_at": "2025-01-13T02:51:33.689Z",
  "customer_name": "Uca",
  "customer_phone": "6287781574213",
  "table_number": "7",
  "user_name": "Hudza"
}
```


Table 7 Response

4.1.2 Frontend

The system page interface display is divided into two, namely the admin and the cashier, the admin has 8 system page interface displays, namely login page, dashboard page, master data page (product), master data page (category), master data page (cashier), transactional data page (transactions), and transactional data page (transaction items).

The admin's dashboard page displays products data, categories data, transactions data, cashiers data and transaction items data. Figure 4 illustrates the display of the admin's dashboard page.

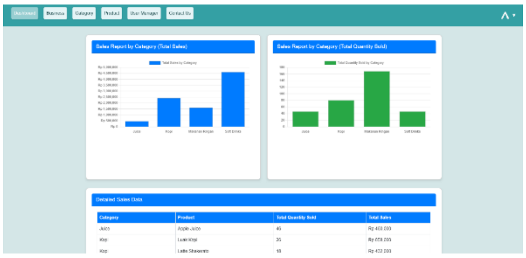

Figure 4 Admin Dashboard Page

The Transaction items data page displays transaction items data on the system. This page can be accessed to view details of transaction items. Figure 5 illustrates the display of the cashier page, for cashiers to make transactions.

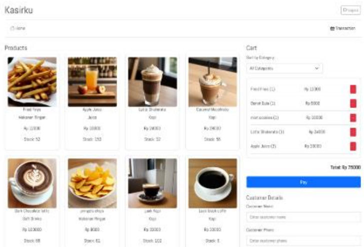

Figure 5 Cashier Page

4.1.3 Testing

The system underwent thorough testing, including black box testing to ensure functional compliance. Black box testing confirmed that all modules, such as transaction recording, and product management functioned as expected. These results highlight the system's reliability, robustness, and readiness for real-world implementation. Table 8 illustrates results of the test.

Objective	Expected Results	Actual Results
Users login with valid credentials	Users are directed to the admin/cashier page	As expected

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

	according to their role	
Adding new products	The product is added and appears in the list.	As expected
Making transactions	Transactions are recorded. Product stock is reduced. Notes are displayed or printed.	As expected

Table 8 Black Box Testing

5. CONCLUSIONS

The black box testing approach is now being used to test the system. In order to determine the functions of system features that are not operational, this technique is used to evaluate the operation of each module or feature in the system being developed. Every feature of the Point of Sale Application functions successfully, according to the test class results with the anticipated outcome.

6. SUGGESTIONS

Future development of the Point of Sale (POS) system could include the creation of a mobile application to improve accessibility for users who need to manage transactions and monitor business operations on the go. This enhancement would provide greater flexibility and convenience, especially for businesses with mobile sales or remote operations.

Additionally, integrating the system with third-party services such as payment gateways, accounting software, and customer loyalty programs could significantly enhance its functionality. These integrations would streamline operations, simplify financial reporting, and improve customer engagement, offering a more comprehensive solution for modern businesses.

REFERENCE

[1] D. Milenković, J. Petković, and S. Marinković, “Globalization and Its Impact on Technological Development,” *41 st International Conference on Organizational Science Development*, 2022, [Online]. Available: https://api.semanticscholar.org/CorpusID:249974620

[2] S. Burlacu, M. D. O. Negescu, S. R. Pătărlăgeanu, and R. Vasilescu, “Digital globalization and its impact on economic and social life,” *SHS Web of Conferences*, p., 2021, doi: 10.1051/shsconf/202112906003.

[3] A. Ambore, “Design and Development of a Smart POS App with Cloud,” *Int J Res Appl Sci Eng Technol*, p., 2023, doi: 10.22214/ijraset.2023.51484.

[4] H. Asrani, S. Vishwakarma, D. Asrani, and D. Asrani, “Point of Sale Systems,” *International Journal of Innovative Research in Computer Science and Technology (IJIRCST)*, p., 2024, doi: 10.55524/csistw.2024.12.1.63.

[5] Y. Ramos and Á. Castro, “Point of Sales Systems in Food and Beverage Industry: Efficient Technology and Its User Acceptance,” vol. 6, pp. 582–591, 2017, [Online]. Available: https://consensus.app/papers/pointofsales-systems-in-food-and-beverage-industry-ramos-castro/73c65697abcd5c89a982a09bfce50bd9/

[6] D. Rode and C. Stammen-Hegener, “Digital Technologies Within the DIY Store: A Systematic Literature Review,” *Business Ethics and Leadership*, p., 2022, doi: 10.21272/bel.6(2).116-126.2022.

[7] Y. Bar-Zeev *et al.*, “IQOS marketing strategies at point-of-sales: a cross-sectional survey with retailers,” *Tob Control*, vol. 32, p., 2022, doi: 10.1136/tobaccocontrol-2021-057083.

[8] N. F. Waruwu and H. T. Amijaya, “Web-Based Sales Information System in One of The Cash Shops Using the Business to Customer Method,” *Jurnal Computech & Bisnis*, p., 2024, doi: 10.56447/jcb.v17i2.244.

[9] S. F. Pane, A. Yanuar, A. F. Kurniawan, and M. Az, “Implementasi Middleware Pada Evomo Dengan Metode Web Service Restfull Dan Pengujian CI/CD, Coverage Serta Simulasi Protokol

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

Grafana," *Jurnal Tekno Insentif*, vol. 15, no. 2, pp. 110–121, 2021.

[10] A. Golmohammadi, M. Zhang, and A. Arcuri, "Testing RESTful APIs: A Survey," *ACM Transactions on Software Engineering and Methodology*, vol. 33, pp. 1–41, 2022, doi: 10.1145/3617175.

[11] A. Ehsan, M. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Applied Sciences*, p., 2022, doi: 10.3390/app12094369.

[12] I. Engelbrecht and H. Steyn, "Does TDWG Need an API Design Guideline?," *Biodiversity Information Science and Standards*, p., 2021, doi: 10.3897/biss.5.75372.

[13] S. Hyrynsalmi, "The State-of-the-Art of the Integration Platforms as a Service research," *2022 IEEE/ACM International Workshop on Software-Intensive Business (IWSiB)*, pp. 17–22, 2022, doi: 10.1145/3524614.3528634.

[14] H. Bhardwaj, "Challenges with Implementation of Node.Js," *Int J Res Appl Sci Eng Technol*, p., 2021, doi: 10.22214/ijraset.2021.37464.

[15] H. Srivastava, S. Gupta, K. Anand, and A. Sharma, "PDF-Chat SaaS Platform Using MERN Stack," *International Journal of Innovative Research in Engineering*, p., 2024, doi: 10.59256/ijire.20240501007.

[16] C. Lee, H. F. Kim, and B. G. Lee, "A Systematic Literature Review on the Strategic Shift to Cloud ERP: Leveraging Microservice Architecture and MSPs for Resilience and Agility," *Electronics (Basel)*, p., 2024, doi: 10.3390/electronics13142885.

[17] O. C. Oyeniran, O. T. Modupe, A. A. Otitoola, O. O. Abiona, A. O. Adewusi, and O. J. Oladapo, "A comprehensive review of leveraging cloud-native technologies for scalability and resilience in software development," *International Journal of Science and Research Archive*, p., 2024, doi: 10.30574/ijsra.2024.11.2.0432.

[18] S. F. Pane, R. M. Awangga, and B. R. Azhari, "Qualitative evaluation of RFID implementation on warehouse management system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1303–1308, 2018.

[19] S. Gawande and S. Jiwani, "Constructive Effect of Hybrid Project Management Methodologies on Success Rates of Projects," *International Journal of Science and Research (IJSR)*, p., 2024, doi: 10.21275/sr24929184750.

[20] O. Hillaire, "Best Practices for Implementing a Hybrid Project Management Methodology," p., 2018, [Online]. Available: https://consensus.app/papers/best-practices-for-implementing-a-hybrid-project-hillaire/a050dcb6c62b50fa93742c1ed335b57a/

[21] N. Prenner, C. Unger-Windeler, and K. Schneider, "How are Hybrid Development Approaches Organized? - A Systematic Literature Review," *2020 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pp. 145–154, 2020, doi: 10.1145/3379177.3388907.

[22] T. Chen, K. J. Leong, M. Yee, O. Baitsegi, L. Palanisamy, and R. Ramasamy, "Hybrid Project Management between Traditional Software Development Lifecycle and Agile Based Product Development for Future Sustainability," *Sustainability*, p., 2023, doi: 10.3390/su15021121.

[23] P. Vijaya and V. Reddy, "New Hybrid Model with Fine Blend of Agile and Waterfall," *Indian Journal of Computer Science and Engineering*, p., 2024, doi: 10.21817/indjcse/2024/v15i1/241501028.

[24] A. Rahman, "IT Project Management Frameworks: Evaluating Best Practices and Methodologies for Successful IT Project Management," *ACADEMIC JOURNAL ON SCIENCE, TECHNOLOGY, ENGINEERING & MATHEMATICS EDUCATION*, p., 2024, doi: 10.69593/ajaimldsmis.v1i01.128.

p-ISSN :1858-3911* , *e-ISSN : 2614-5405
https://journal.fkom.uniku.ac.id/ilkom

[25] P. Sharma, “Impact of Agile Methodologies in Project Success,” *Int J Res Appl Sci Eng Technol*, p., 2022, doi: 10.22214/ijraset.2022.43204.

[26] P. Amajuoyi, L. B. Benjamin, and K. B. Adeusi, “Optimizing agile project management methodologies in high-tech ssoftware development,” *GSC Advanced Research and Reviews*, p., 2024, doi: 10.30574/gscarr.2024.19.2.0182.

[27] S. Chathuranga, S. Jayasinghe, J. Antuchevičienė, R. Wickramarachchi, N. Udayanga, and W. Weerakkody, “Practices Driving the Adoption of Agile Project Management Methodologies in the Design Stage of Building Construction Projects,” *Buildings*, p., 2023, doi: 10.3390/buildings13041079.