International Journal of Innovative Research in Engineering and Management (IJIREM)
ISSN (Online): 2350-0557, Volume-11, Issue-6, December 2024
https://doi.org/10.55524/ijirem.2024.11.6.15
Article ID IJIR3052, Pages 138-150
www.ijirem.org

Point of Sale Application for MSMEs in Indonesia with Payment Gateway Integration and NoSQL-Based

Indra Briliansyah[1], Donny Avianto[2]

[1, 2] Department of Informatics, Faculty of Science and Technology, University of Technology Yogyakarta, Yogyakarta, Indonesia

Correspondence should be addressed to Indra Briliansyah; indrabriliansyah@gmail.com

Received: 24 November 2024 Revised: 8 December 2024 Accepted: 22 December 2024

ABSTRACT- Micro, Small, and Medium Enterprises contribute significantly to the Gross Domestic Product of the country and also happen to be the largest generator of employment in Indonesia. Nevertheless, most of them find it difficult to adopt digitalization because of their limitations in infrastructure and relevant information systems and also due to poor digital literacy. These challenges, therefore, extend to fundamental business operations such as financial management, inventory monitoring, and sales, which are usually done manually. In line with this, the development of an efficient and user-friendly Point of Sale application, available on both Android and web platforms, is a strategy in the right direction. The application is designed to bring efficiency in financial management, recording of inventories, and sales operations for MSMEs, since it offers them the instruments needed for migrating to digital workflows. In addressing these constraints, the POS application hopes to empower MSMEs so that they can participate in the thriving digital economy and contribute more firmly to Indonesia's development. The focus of the study is on designing and developing a POS system that will respond to the needs of MSMEs in the digital era.

KEYWORDS- MSME, Point of Sale, Android, Web

I. INTRODUCTION

MSMEs play a very strategic role in Indonesia's economy. In 2018, MSMEs contributed 57.24% to Indonesia's Gross Domestic Product (GDP), equivalent to Rp 5,721.14 trillion. MSMEs also play a significant role in job absorption, employing 97.05% of the workforce in Indonesia or equivalent to 116.97 million workers, and contributing 14.37% to Indonesia's non-oil and gas exports, equivalent to Rp 293.84 trillion [1]. Similarly, in India, MSMEs contribute around 30% of India's GDP based on annual reports and, according to conventional estimates, employ over 50% of industrial workers. They also contribute about 40% of overall exports. Over 97% of micro-firms are part of MSMEs, and 94% of firms are registered with the government [2].

Digitalization has influenced and shaped the business world for years, yet many have not maximized this opportunity [3]. Digitalization is believed to be a solution for MSMEs to rise, but its implementation still faces various obstacles, one of which is the lack of information system infrastructure that not all MSMEs possess. Additionally, many MSMEs are confronted with the problem of insufficient knowledge and skills in using digital technology [4]. This has resulted in many SMEs still facing obstacles in financial management, inventory recording, and sales processes, which are still done manually to this day.

According to the issues described in the previous paragraph, MSMEs play a strategic role in the Indonesian economy; however, MSMEs still face problems in the field of information systems, as not all MSMEs are integrated with a system. The lack of knowledge and skills in using digital technology is also one of the reasons why MSMEs face problems in financial management, inventory recording, and sales processes. SMEs themselves have great opportunities for growth and innovation, despite facing various obstacles. One of the keys to the growth of MSMEs is the increasingly broad access to technology [5]. Therefore, a Point of Sale (POS) application is needed to address the issues faced by MSME actors. Ramadi's research focuses on the development of an Android-based Point of Sale (POS) application using the Kotlin programming language, designed to handle inventory, employee, transaction, and report management [6]. Unlike Ramadi, Kusmawati & Utami took a different approach by developing a POS application that encompasses both web and mobile platforms simultaneously. The application was created using the Flutter framework, allowing web and mobile development to be done simultaneously, making the development process more efficient. With the presence of web and mobile-based applications, it provides high flexibility for users, allowing them to manage operations anywhere [7].

Another study conducted by Dharmalau focused on an Android-based POS application using the Flutter framework. The POS application created already includes product data and sales transactions as well as generating reports in real-time [8]. Not much different from Dharmalau, Naufal developed a web-based POS application using the Laravel framework. The application developed includes features such as inventory management, transactions, and reporting. However, from both studies mentioned above, there are still shortcomings in the integration of digital wallet payment methods and virtual accounts from certain banks [9].

From relevant research, it can be concluded that the POS application plays an important role in assisting MSME management with features such as inventory tracking, transaction recording, and report generation. Nevertheless,

in the relevant research, there are still some shortcomings in terms of integrating digital payment methods and multi-platforms. The integration of digital payments and multi-platforms will enhance user convenience and meet modern business needs. Therefore, in this research, a POS application that is already integrated with digital payments and multi-platforms will be developed.

II. RESEARCH METHODOLOGY

A. Method

This research method uses the waterfall method, as illustrated in Figure 1. The waterfall method is a software development life cycle that is sequential in nature, where progress flows steadily downward through various phases [10].

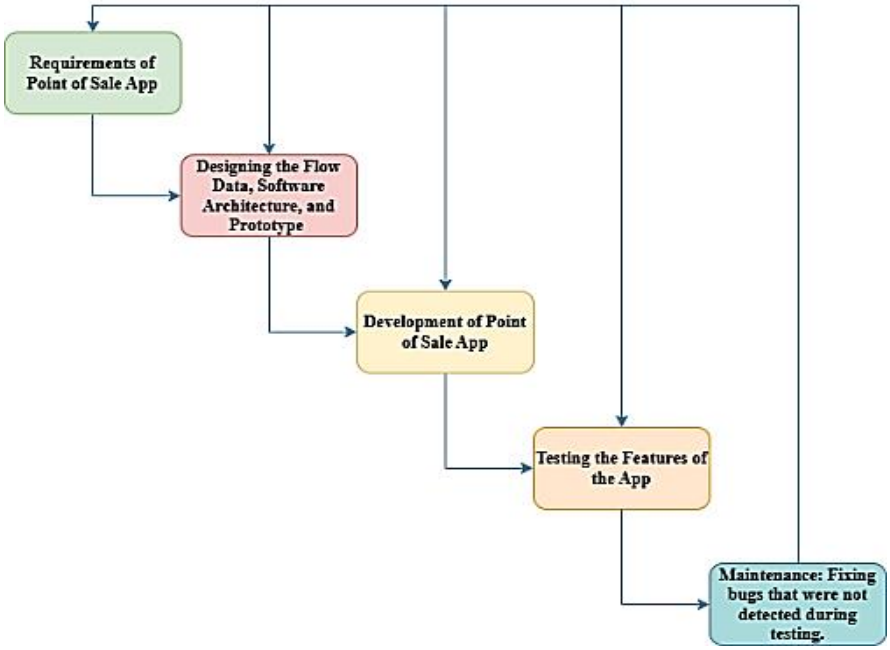

Figure 1: Waterfall Diagram

- **Requirements**

The earliest stage in the waterfall method is the analysis of the requirements. During this stage, the system or software requirements are collected to meet the needs of the users [11]. Information on the system requirements can be identified through various methods [12]. One of them is by observing MSME actors, after which the obtained data is further analyzed to determine the functional and non-functional needs of the users.

- **Design**

Once the analysis phase is complete and the system requirements have been captured, it is time for the phase known as Design. During this stage, a DFD will be created, which depicts data logic applied to process the data, the originator of the data, and the purpose of the data [13]. On top of that, based on the data flow needs within the system, both the software architecture design and the application prototype design shall be developed.

- **Development**

During this stage of the waterfall methodology, the developed designs and system architecture from the design phase are now transformed into workable and functional program codes. All the previously designed layouts regarding user interface and system workflow must be translated into a working application or software. The implementation is done through a programming language and frameworks that match the technical requirements and specifications of the system. The developers ensure that the code is written according to the planned design, keeping in view modularity, scalability, and readability for future maintenance.

- **Testing**

Once the development is complete, the software enters the testing phase, where its functionality and reliability are put to test. This stage is meant to ensure that the software meets the user's requirements and functions as expected without bugs or errors [14]. Testing is done to ensure that all features work well, and the system is generally very smooth. The test also covers performance to see that the software is responsive under various conditions and can handle expected workloads efficiently. Additionally, compliance testing is carried out using blackbox testing, a method focused on validating the input, process, and output of each feature without delving into the internal code structure. This ensures that the software meets the defined functional requirements, identifies potential issues, and resolves them before deployment, guaranteeing a high-quality final product for end users.

- **Maintenance**

The final stage of the waterfall method is the maintenance phase, which plays a role in ensuring long-term reliability and functionality. During this stage, continuous monitoring is carried out to identify and address issues that may arise after the software has been deployed. These could be bugs or errors that were not caught during the testing phase, or new challenges arising due to changes in user requirements, technological changes, or changes in operating environments. The process of maintenance involves diagnosing and correcting problems, applying updates for

performance enhancement, and ensuring compatibility with the evolution of systems. This is the most crucial stage in terms of retaining users' satisfaction with the software, keeping the software fit for purpose over time.

B. System Analysis

The existing system, which is currently operating, still performs buying and selling transactions with customers manually, as illustrated in Figure 2. This has some limitations due to the nature of the manual process, affecting the operational efficiency of the existing one. Potential errors might also occur due to this and can lead to reduced revenue. Hence, a design proposal for an automated system can be made in order to enhance the efficiency and increase sales coverage of the existing range of products.

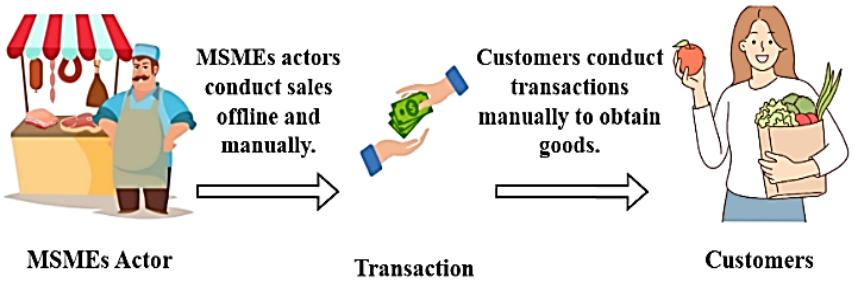

Figure 1: Current System

The web and mobile-based Point of Sale application, as illustrated in Figure 3, will be developed to process transactions conducted by the MSME actors. Later, MSME actors will be able to input goods to be sold into the application through the available mobile and web applications. The web application also provides an online selling space for MSME actors that can be purchased by customers. Meanwhile, customers can make purchases online or offline by visiting the store directly.

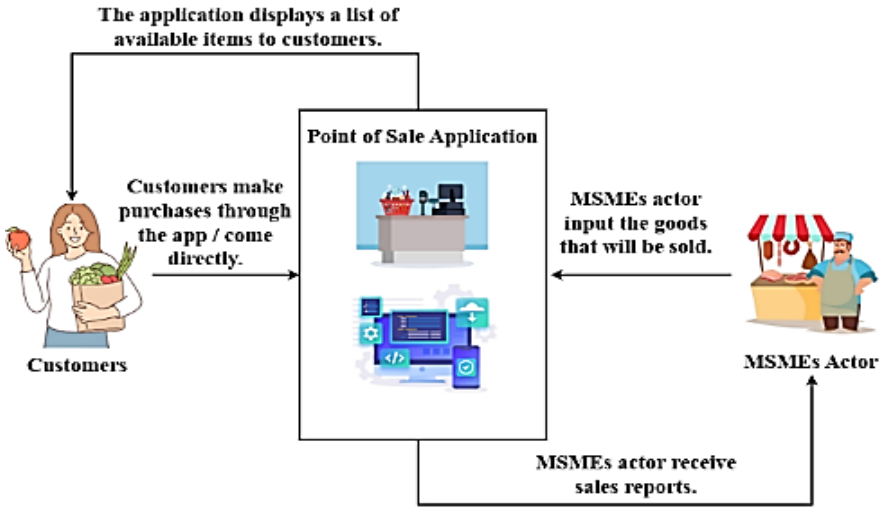

Figure 2: Proposed System

III. RESULT AND DISCUSSION

Based on the system design, this stage aims to explain how the application is developed. Additionally, it aims to realize the specifications and system flow into an application that can function according to needs. This includes coding, integrating system components, and ensuring that all features work as intended. The development process also emphasizes adherence to design standards and system requirements to maintain functionality and usability.

A. Result

- **System Requirements**

In determining problems and analyzing needs from data obtained through observations of MSME actors, it can be concluded that functional and non-functional needs are generated as follows:

Table 1: Functional Requirements

No	Functional Requirements	Syst. Feature Name
1	Input Requirements	Users / MSME actors can input the goods to be sold, and the application can make requests such as account creation.
2	Process Requirements	The application allows for the addition and removal of products, including cash and non-cash sales with digital payment integration, and ongoing transaction recording.
3	Output Requirements	System providing reports to MSME actors generated from ongoing transactions.

According to Table 1, users/UMKM actors need to enter the detailed data of the goods to be sold. In addition, the

application also provides input for creating customer accounts who will make purchases online. The application must be able to process the addition and deletion of products, payments in cash as well as non-cash. The application will also generate output in the form of reports obtained from ongoing transaction data.

Table 2: Non-Functional Requirements

No	Non-Functional Requirements	Description
1	Software Requirements	Operating System: Windows 10 64-bit IDE: Visual Studio Code and Android Studio Database: MongoDB
2	Hardware Requirements	Processor: Ryzen 5 5500 RAM: 8GB Storage: SSD 500GB

In application development, non-functional requirements can be explained in two categories as shown in Table 2. The application requires Windows 10 64-bit for its development, and also uses the Visual Studio Code and Android Studio IDEs to implement the application design into code. The MongoDB database is used to store the entire application's data in NoSQL format. The use of supporting hardware is needed in the development process to avoid any unwanted issues.

- **Data Flow Diagram**

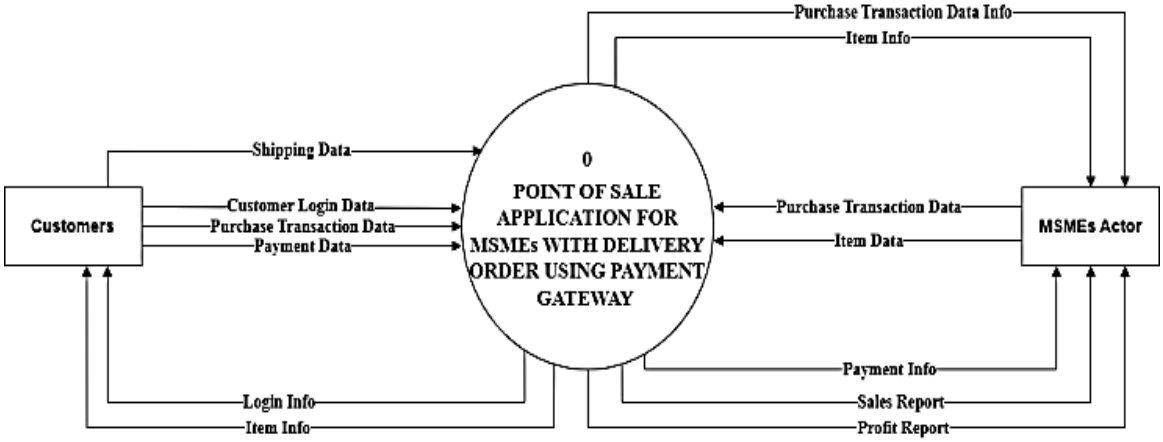

Figure 4: Context Diagram

As in the design stage, the data model design uses DFD (Data Flow Diagram) to illustrate the flow of data or the system. In Figure 4, the context diagram or DFD Level 0 is used to depict the system boundaries and its interactions with entities [15].

Customer entities have access rights to the system, such as logging in and registering an account. Customers also have the right to make purchases of available goods in the system. After making a purchase transaction, customers can make payments in the system using payment gateway integration.

Owner entities have access rights to the system, such as adding product data. In addition to adding product data, owners can also delete products within the system. The owner can view the payment status information made by customers during the purchase transaction. The owner can also see reports generated by the system based on existing transactions.

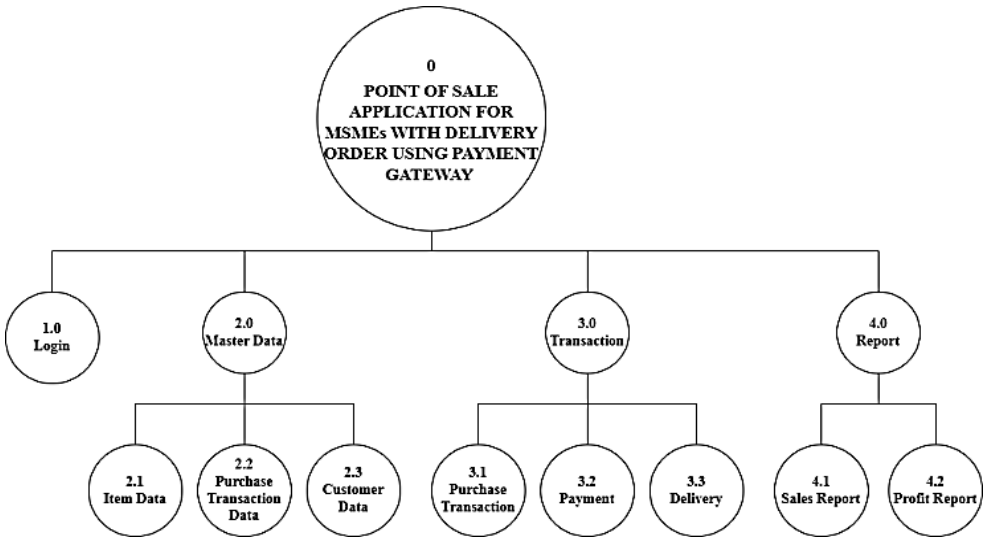

Figure 5: Level Diagram

The hierarchical or level diagram in the development of a point of sale system for MSMEs, as shown in Figure 5, is a graphical representation of the stages and processes in the development of that system. In this diagram, it includes login, master data, transactions, and reports. The system provides Login for customers at the earliest stage to store customer sessions in the system. Customers need to enter credentials such as a password to log into their accounts. Master data in the hierarchical diagram serves as a place to store data related to the system, such as product data, purchase transaction data, and customer data. This data will become the main part and will be used for transaction processes within the system. The Transaction Stage is the main activity carried out within the system, such as purchasing goods, payment processing, and shipping. The final stage in the hierarchical diagram, namely the Report, provides data on sales reports and the profits generated during the buying and selling process.

- **Database Model**

Unlike relational databases or databases that use SQL, NoSQL databases are non-relational data management systems that do not require a fixed schema or complex queries. The main difference between the two databases lies in the schema, where the schema of relational databases rigidly determines how data must be entered into the database, whereas NoSQL databases do the opposite [16].

Table 3: Products Database Model

Key	Data Type	Function
id	Number	As a unique identifier for each product
name	String	Storing the product name
image	String	Storing the product image URL
category	String	Storing product categories
price	Number	Storing the cost of production
discount_price	Number	Storing the selling price
	Number	Storing product stock
availability	Boolean	Indicating whether the product is available
description	String	Storing the product description
date	Date	Saving the product addition date.

Table 3 illustrates the database model designed to store information about products in a NoSQL-based data system. In the NoSQL context, data is stored in the form of documents that have key-value pairs, where the key represents the attribute name, and the value is the stored data value. This document is used to store information about the products added to the system, such as product name, product price, product stock, and product description, which will be used in managing product data in the system.

Table 4: Customers Database Model

Key	Data Type	Function
name	String	Storing customer names
email	String	Storing customer email addresses
password	String	Storing customer passwords
cart_data	Object	Storing customer cart data
date	Date	Storing the account creation date

In the users model Table 4, it is a database model for storing information about customers. With the existence of the Users document, customers can create accounts and log in to the system. In addition, customer cart data will be stored in the database in the form of an object that can hold up to 299 products, with each index representing the product ID, so that products already in the cart will remain stored even if the customer logs out.

- **System Architecture**

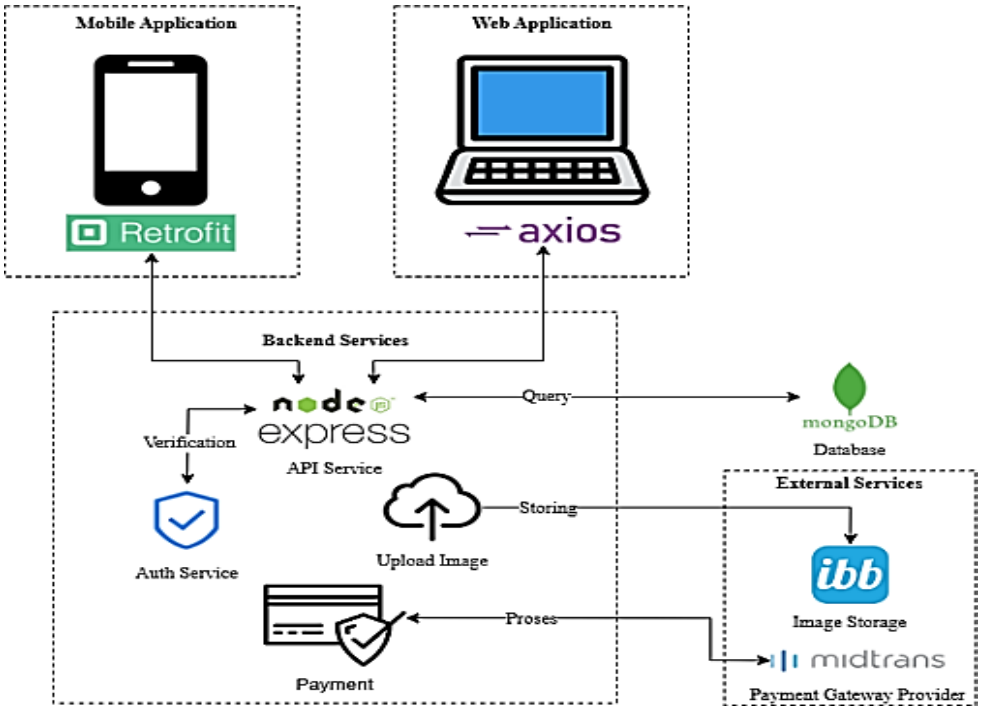

Figure 6: System Architecture

In the system architecture in Figure 6, it explains the workflow of the point of sale application developed to facilitate MSME actors. The architecture consists of several components, namely Mobile and Web for users and customers to interact, backend services that manage all activities in the system, external services like ImgBB for image storage and midtrans for the payment gateway, and the MongoDB database.

The Mobile Application is developed using the Kotlin programming language and utilizes the Retrofit library, which is used for communication between the Android application and the backend via the HTTP protocol. Retrofit will make API requests using several methods such as GET, POST, PUT, and DELETE [17]. By using Retrofit, API integration becomes simpler and more structured. Meanwhile, the Web Application is developed using the React framework. The framework was chosen as an alternative to native development because it simplifies and accelerates the application creation process. Specifically, React has become a popular choice due to its support from a large community and its development being backed by Meta [18]. Not much different from mobile applications, React uses Axios to communicate with the API in the backend via the HTTP protocol. Axios is responsible for sending requests such as GET, POST, PUT, and DELETE to the API endpoints provided by the backend. The backend will process these requests according to the implemented logic, then return a response in the form of data. This process ensures data synchronization between the frontend and backend.

On the backend services or server side, Express.js is used as the framework to handle various requests from mobile and web applications, MongoDB as the database, and Midtrans for payment integration. The backend services provide API endpoints that include features such as user registration and login with JWT tokens, adding and deleting products, and managing the shopping cart. Integration with ImgBB also allows for image uploads. Midtrans Snap API is used to process payments, with stock validation before the transaction and automatic payment status updates. Data will be stored in MongoDB in the form of documents for product data, transaction data, and user data. Auth Service is used to provide additional security in the user authentication process using JWT.

External Service provides additional features that are not implemented directly in the backend, but rather by utilizing third-party APIs, namely ImgBB and Midtrans. ImgBB itself is used to manage product image uploads, providing URL-based image storage. The use of ImgBB was chosen to reduce local storage burden and provide quick access to display product images in the application. Meanwhile, Midtrans is used to process customer payments in the marketplace system. Midtrans is used to handle payment transactions with various methods such as bank transfers, e-wallets, and credit cards [19].

MongoDB is used as the main database to store the information required by the application, such as user data, products, shopping carts, and transactions. This database is NoSQL, making it flexible for storing data in JSON format like documents. MongoDB was chosen for its scalability in handling large amounts of data and its support for fast data storage and retrieval.

B. Testing and Discussion

- **Blackbox Testing**

During the application testing phase, the blackbox testing method is used to ensure that each and every feature in the application works as expected. In this testing approach, the application output is analyzed for a set of input scenarios given without actually going into the code or its implementation details. Testing of this application covers key functionalities like customer registration, customer login, adding new products, and payment processing. By systematically testing these features, the testing validates the features on their reliability and functionality. The results of the blackbox testing in detail are summarized in the following table, which shows some performances of the application and areas it needs to improve.

Table 5: Blackbox Testing

Feature	Testing Scenario	Expected Outcome	Actual Results	Status
Customer Signup	Customers sign up on the website for the first time.	Customer data will be stored in the database.	Customer data has been successfully saved in the database.	Success
Customer Signup	The customer signed up with the same email.	The system will reject and send the message "Email already in use."	The system successfully rejected and sent the message "Email already in use."	Success
Adding Product	The user added a new product.	The product will be stored in the database and the system will send a message "Product successfully added!"	The product has been successfully saved and the system sends a message "Product successfully added!"	Success
Deleting product	The user deleted the product.	The product will be deleted from the database.	The product has been successfully deleted from the database.	Success
Adding products to the cart	The customer adds products to the cart.	The product will be added to the cart.	The product has successfully been added to the cart.	Success
Payment using Payment Gateway	The customer makes an order.	The system will display payments using a payment gateway.	The system successfully processed the payment using the payment	Success

			gateway.	
Notification of item purchase	Notification received on the Android application	Users will receive notifications for incoming purchases.	The user successfully received a purchase notification.	Success

The black box testing method, as outlined in Table 5, tests the system mainly on functional requirements. Testing of each feature is based on expected input, processing, and output to show that the feature behaves as intended. For example, the registration of customers is tested for proper storage of customer details in the database when the registration is effected. It also tests the system on special scenarios, like blocking the use of duplicate emails with the help of proper error messages. Similarly, add and remove products are also tested for smooth database integration along with confirmation messages as a success validation.
Other than the basic functionalities, the blackbox testing was conducted on functionalities such as adding products to the cart, providing payment through a payment gateway, and sending notifications related to purchase. These mechanisms for testing ensure that the system is working with the expected user flow. This becomes very important, for instance, when a customer goes to make a payment, the system has to be able to redirect him to the payment gateway successfully and process the transaction without any errors. This level of testing is very critical to ensure that all steps in the transaction process are executed to perfection.
Blackbox testing is found to be very effective in verifying functionalities that are in direct interaction with the users. This testing, by simulating real-life scenarios and user interactions, ensures that the system meets not only the specified requirements but also provides a seamless and reliable experience for users. Its focus on output verification and user-centric functionalities makes it indispensable in finding and fixing issues before deployment.

- **User Interface Results**

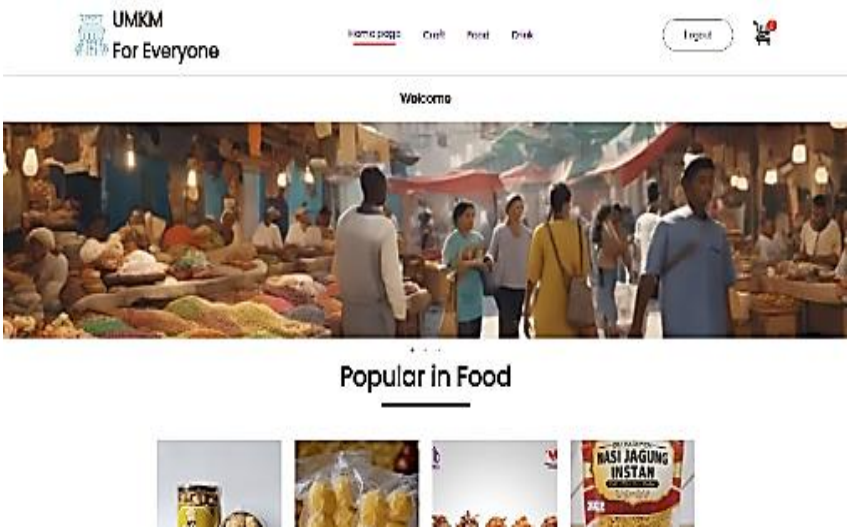

Figure 7: Homepage

The homepage, as shown in Figure 7, is used to display the goods sold by MSME actors. This page showcases popular products and the latest additions. At the top of the homepage, there is also a website menu that can be accessed by customers. The menus include crafts, food, and beverages, which are used to sort according to the existing product categories. In addition to the menus, there are buttons used for login and a cart to store data on products that have been added to the cart.

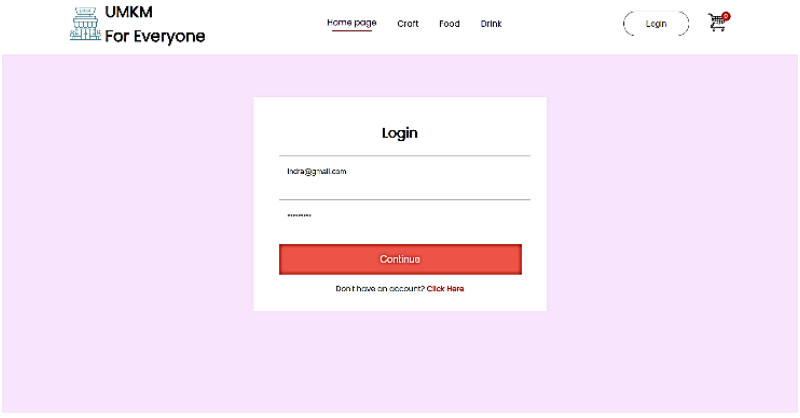

Figure 3: Login Menu

The login menu, as illustrated in Figure 8, is used for customers to enter user data, such as email and password, to access their accounts. By logging in, the data in the cart that has been added can be saved to the customer's account, making the shopping process easier in the future. In addition to providing a login menu, there is a button that directs to the account creation page if you do not yet have an account.

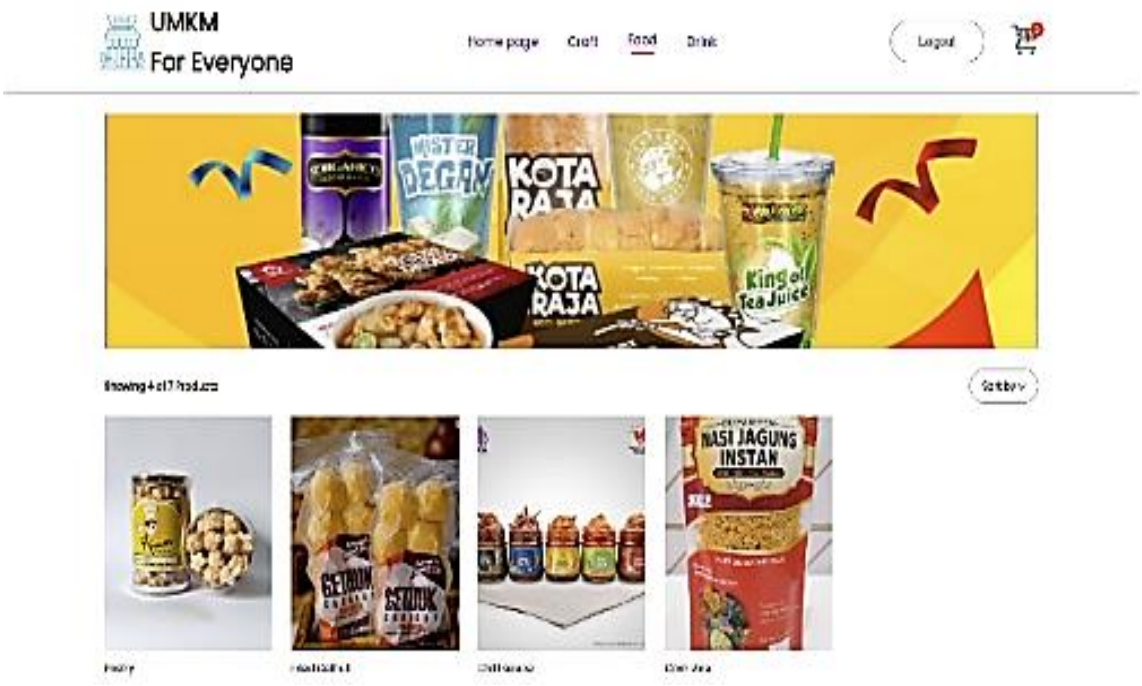

Figure 9: Product Category Menu

On this product category page, as shown in Figure 9, it is used to help customers explore products based on specific categories. There are three product categories, namely food, beverages, and crafts, which can be accessed on the navigation tab. With the category menu, customers do not need to browse through unwanted product sections, making it more efficient and organized. In addition, sellers are also helped to display their products in a more structured manner, thereby increasing the chances of customers finding the desired products.

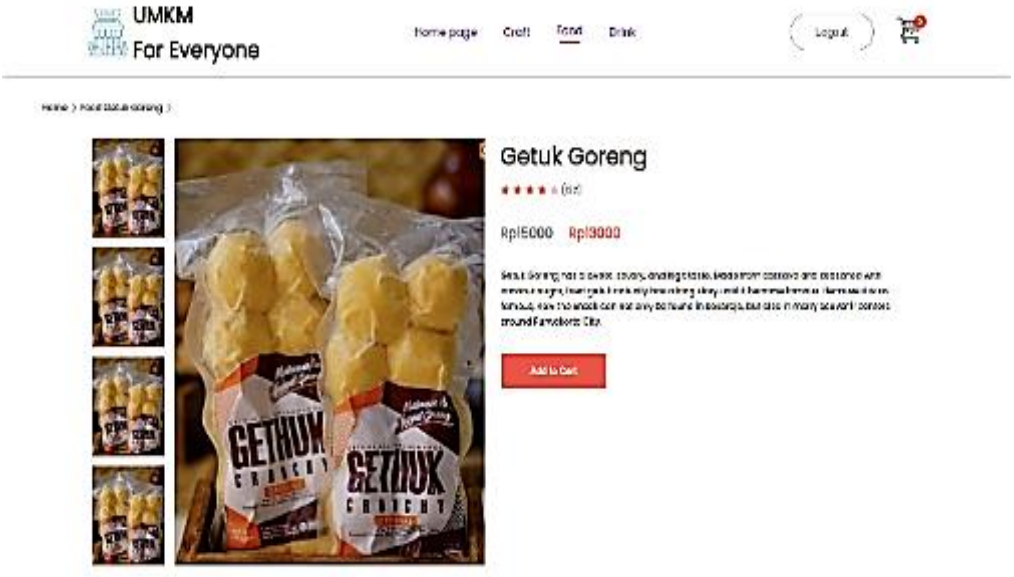

Figure 10: Menu Product Details

The product description menu, as illustrated in Figure 10, will appear when you click on the available product. This menu provides more detailed information about the product to help customers understand the related product. Additionally, there is a button to add the product to the cart. With the presence of a complete product description, customer trust in the platform increases, while also reducing the risk of product returns due to mismatched expectations.

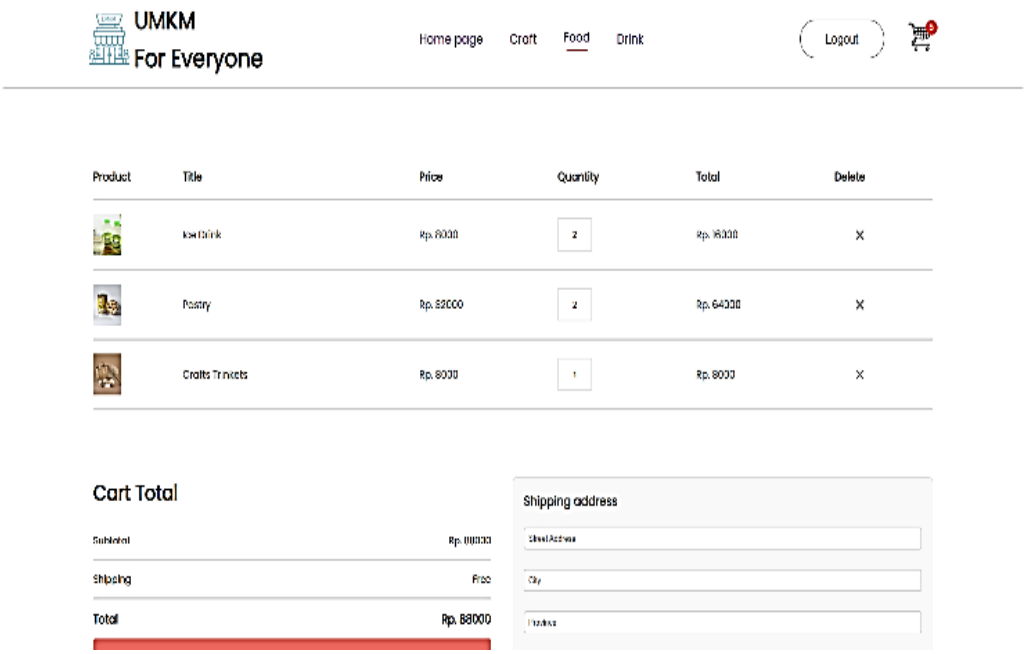

Figure 11: Shopping Cart Menu

The cart menu, as shown in Figure 11, serves as a temporary storage place for products selected by customers before making a payment. In this menu, customers can view the products that have been added, complete with information such as name, quantity, price, and total shopping cost. This feature also allows customers to change unwanted products or review their orders before proceeding to the payment process.

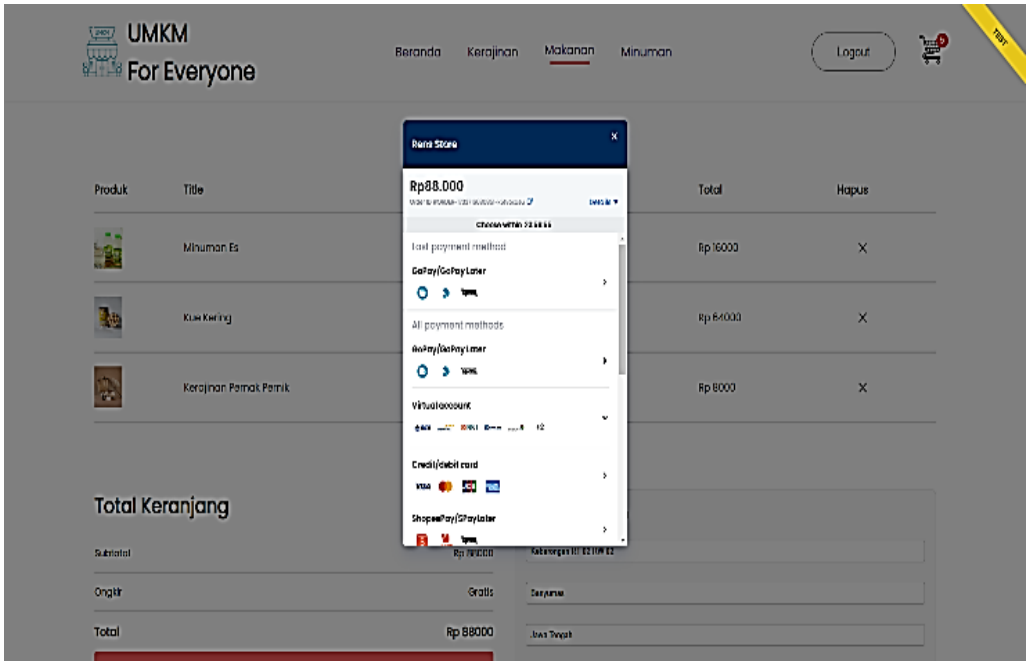

Figure 12: Payment Menu

In payments, as shown in Figure 12, we utilize the Midtrans payment gateway to make it easier for customers to complete transactions with multiple payment methods. With the integration of Snap, customers can choose from various payment methods, such as bank transfers, credit cards, e-wallets, or payments at retail outlets, all within a user-friendly interface. The payment process is conducted in real-time, with automatic confirmation that speeds up transaction validation.

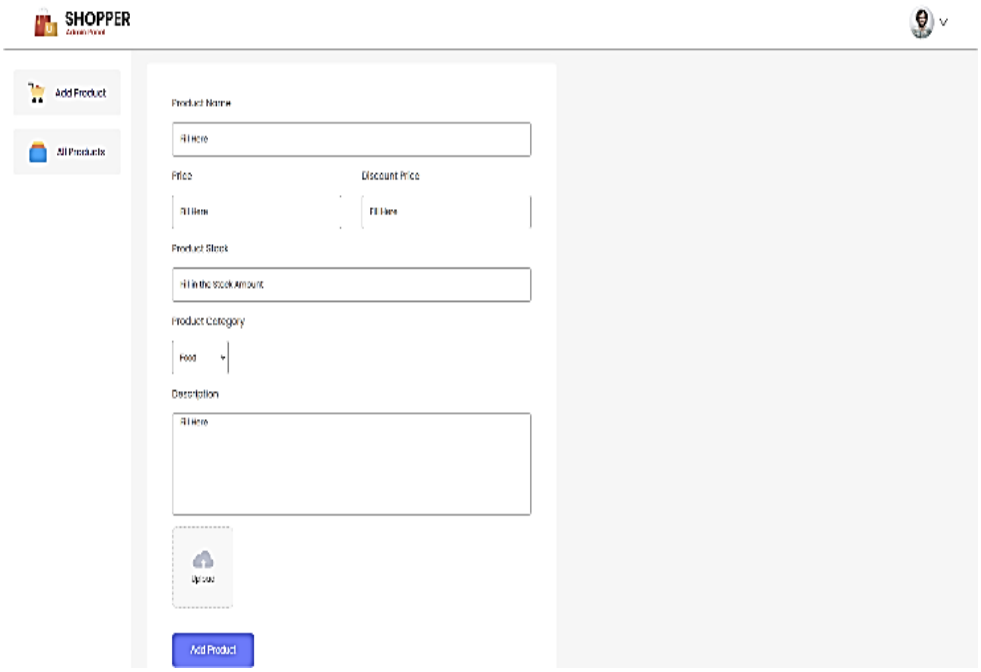

Figure 13: Admin Panel

There are two main menus in the admin menu, as shown in Figure 13: the menu to add items and the menu to view all items. Both are designed to make it easier to manage data and operate the system for MSME actors. Through the Add Item menu, MSME actors can add new product data, such as the name of the product, category, price, stock, description, and image, so that the product can immediately appear on the platform. Meanwhile, the All Products menu allows MSME actors to access the complete list of available products.

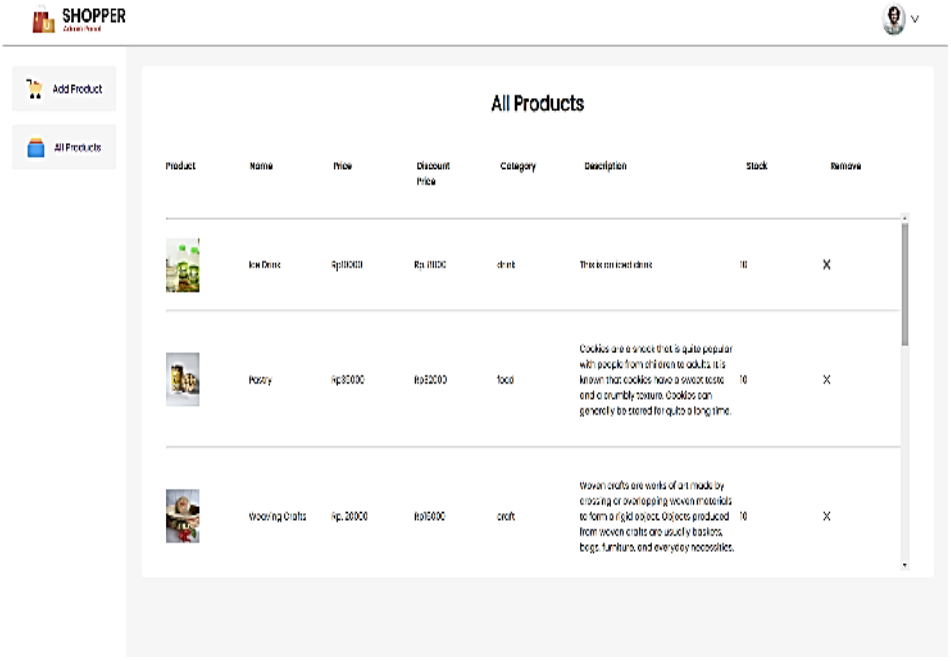

Figure 14: All Product Menu

The All Products page, as shown in Figure 14 is designed to help MSME actors monitor what products exist within the application. The items to be sold by MSME actors will appear fully, including information about the name of the product, price, stock, and category on this page. By using this feature, MSME actors can find out which products are currently available, keep stocks at levels that are always sufficient for sales, and change data when needed. This page serves as the control center for MSME actors to manage their inventory effectively in such a way that the latter can sustain consistency of services to customers and enhance their efficiency.

Figure 15: Android Home Menu

On the main page of the Android application, as shown in Figure 15, MSME actors are presented with several menus that will make it easier for them to navigate and access a variety of important features. Among these menus are Manage Goods, POS, Web Orders, Reports, and Settings. Through the Manage Goods menu, product inventory is managed, while the POS menu opens into the cash register system for recording sales. Notifications of incoming orders when it goes through the website are displayed on the web menu; the report menu generates the reports which help in analysis of business.

Figure 16: Manage Products

Manage Items is a feature included in the system to support MSME actors in managing their products effectively, as shown in Figure 16. This menu will show all the available products in a clear and organized manner, including all the essential details like the name of the product, the price of the product, and the stock availability of the product. The menu will be able to provide ease in organizing and tracking inventory with its user-friendly interface. The menu also has features for adding new products, where MSME actors can fill in the product details such as the name, price, category, stock, and description, and upload product images. It also has a delete button for deleting items that are no longer available or relevant. This dual functionality ensures that the inventory remains accurate and up-to-date. With an intuitive design and practical features, the Manage Items menu streamlines product management, thus enabling MSME actors to have a better overview of their business and act in a timely manner regarding changes in inventory.

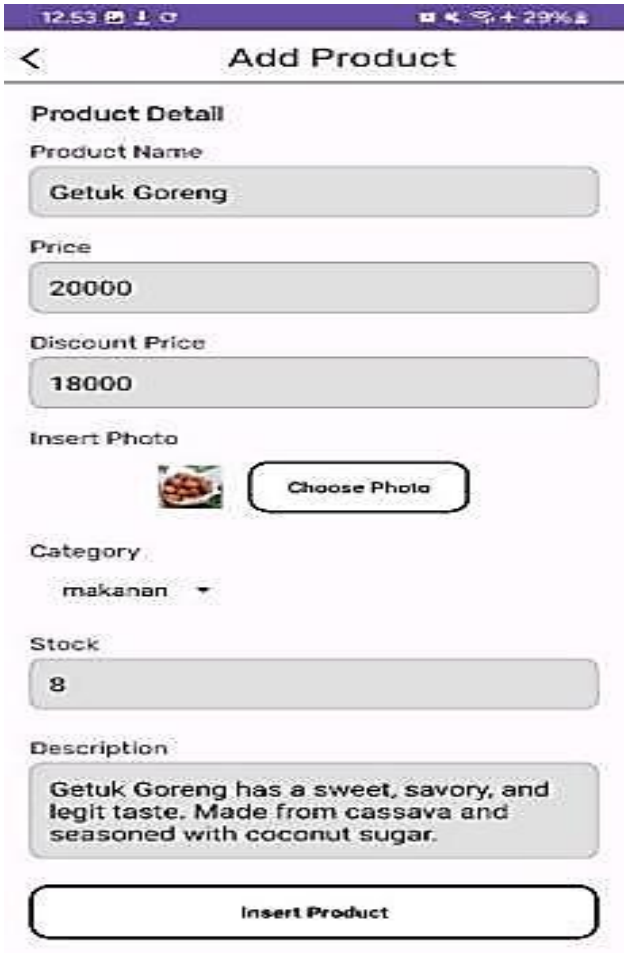

Figure 17: Insert New Product

The product insertion menu in the Android application, as shown in Figure 17, serves as a tool for adding new products into the system efficiently. Similar to its web-based counterpart, this menu allows MSME (Micro, Small, and Medium Enterprise) operators to input comprehensive product details, including the product name, category, price, stock, description, and even upload images directly from the device's gallery. Once the information is successfully submitted, the product data is seamlessly saved in the database, ensuring it is readily accessible and manageable through the Manage Goods menu.

Figure 18: Point of Sale Menu

The POS menu in the application, as shown in Figure 18, is built to ease the transaction process for direct sales. This menu displays all available products, complete with details like name, price, and stock. Users can select a desired product by clicking on it, and each selection automatically updates the total price and quantity in the order summary. With this feature, users can manage transactions in real-time, which makes the sales process faster and more accurate.

IV. CONCLUSION

The results of the study show that the development of a POS application has the potential to ease the challenges facing the MSME stakeholders. In this regard, the application will have the following features: transaction recording process, inventory management, preparation of financial reports, and online sales of the products. In addition, the integrated payment gateway feature within the application will facilitate customers with regard to convenience when transacting to either shop online or by visiting the physical store. The POS application becomes a fresh and related solution for empowering MSMEs against increasingly fierce market competition.

In further research, the integration feature with courier services can enhance the POS application for MSME actors. This will allow the automatic linking of incoming orders by MSMEs with several available delivery services. The feature thus has the benefits of making the delivery management process more effective, starting from ordering couriers to tracking and real-time notification of delivery status to customers.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Danny Hermawan, "TANTANGAN DIGITALISASI UMKM BERBASIS BUDAYA," BI Institute. Accessed: Oct. 15, 2024. Available from: https://www.bi.go.id/id/bi-institute/BI-Epsilon/Pages/Tantangan-Digitalisasi-UMKM-Berbasis-Budaya.aspx

[2] Dr. A. Singh, "MSMES SECTOR IN INDIA: CURRENT STATUS AND PROSPECT," International Journal of Innovative Research in Engineering & Management (IJIREM), vol. 8, no. 4, Jul. 2021. Available from: https://doi.org/10.21276/ijirem.2021.8.4.7

[3] M. Kellermayr-Scheucher, L. Hörandner, and P. Brandtner, "Digitalization at the Point-of-Sale in Grocery Retail - State of the Art of Smart Shelf Technology and Application Scenarios," in Procedia Computer Science, Elsevier B.V., 2021, pp. 77–84. Available from: https://doi.org/10.1016/j.procs.2021.11.075

[4] R. Prakasa Hasibuan and A. Badruszaman, "Training on Improving Production Efficiency and Sales Turnover of UMKM through the Utilization of Information Technology and Lean Manufacturing," 2024. Available from: http://dx.doi.org/10.1504/IJTM.2012.043955

[5] I. G. G. H. Marwanto, Penganggaran Bisnis UMKM, Cetakan I., no. I. Medan: PT Media Penerbit Indonesia, 2024. Available from: http://dx.doi.org/10.56371/ijess.v5i2.317

[6] S. Ramadi, J. Raya Tengah No, K. Gedong, P. Rebo, and J. Timur, "PERANCANGAN APLIKASI POINT OF SALES (POS) DI CAFÉ REHAT DAN SINGGAH BERBASIS ANDROID," Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI), vol. 04, 2023. Available from: https://doi.org/10.30998/jrami.v4i01.5017

[7] A. Kusmawati and W. S. Utami, "KLIK: Kajian Ilmiah Informatika dan Komputer Pengembangan Aplikasi Point of Sale untuk Meningkatkan Kinerja Agen Softdrink Menggunakan Metode Prototype," Media Online), vol. 4, no. 2, pp. 1121–1133, 2023, Available from: https://doi.org/10.30865/klik.v4i2.1300

[8] A. Dharmalau, N. Sucahyo, and I. Mukti, "PERANCANGAN APLIKASI POINT OF SALES (POS) BERBASIS ANDROID MENGGUNAKAN FRAMEWORK FLUTTER DI KAFE ELANGSTA," JRIS: JURNAL REKAYASA INFORMASI SWADHARMA, vol. 3, no. 2, pp. 6–13, Jul. 2023, Available from: http://dx.doi.org/10.46576/djtechno.v5i3.5108

[9] A. R. Naufal, D. A. Nawangnugraeni, and A. T. Suseno, "RANCANG BANGUN SISTEM INFORMASI POINT OF SALE MULTI OUTLET DENGAN MENGGUNAKAN FRAMEWORK LARAVEL DI KOPERASI ITSNU PEKALONGAN," Jurnal Teknik Informasi dan Komputer (Tekinkom), vol. 5, no. 2, p. 280, Dec. 2022, Available from: http://dx.doi.org/10.37600/tekinkom.v5i2.591

[10] Shamsulhuda Khan and Shubhangi Mahadik, "A Comparative Study of Agile and Waterfall Software Development Methodologies," International Journal of Advanced Research in Science, Communication and Technology, pp. 399–402, Jul. 2022, doi: 10.48175/ijarsct-5696. Available from: http://dx.doi.org/10.48175/IJARSCT-5696

[11] N. Hidayati and S. Sismadi, "Application of Waterfall Model In Development of Work Training Acceptance System," INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi, vol. 4, no. 1, pp. 75–89, Feb. 2020, Available from: http://dx.doi.org/10.29407/intensif.v4i1.13575

[12] F. F. Almubarok, F. Y. Muthasina, M. Darwis, and D. G. Purnama, "UI UX Design for Mobile Based Foster Parent Information Application (IOTA) Using the Waterfall Method," Jurnal Teknologi Dan Sistem Informasi Bisnis, vol. 6, no. 3, pp. 550–555, Jul. 2024, Available from: http://dx.doi.org/10.47233/jteksis.v6i3.1418
[13] B. Haqqi and V. Vivianti, "Pengembangan Sistem Informasi Penjualan dan Stok Barang Toko Penjualan Plafon Berbasis Web," Jurnal Edukasi Elektro, vol. 6, no. 2, pp. 116–127, 2022, Available from: http://dx.doi.org/10.21831/jee.v6i2.52170
[14] H. K. Aroral, "Waterfall process operations in the fast-paced world: project management exploratory analysis," International Journal of Applied Business and Management Studies, vol. 6, no. 1, pp. 91–99, 2021. Available from: http://dx.doi.org/10.4018/jitpm.2010040103
[15] V. Widiyanti and R. Tisnawati, "Perancangan Sistem Informasi Point of Sale di PT. Abercode Software Berbasis Web," Jurnal Sosial Teknologi, vol. 4, no. 10, pp. 838–850, Oct. 2024, Available from: http://dx.doi.org/10.59188/jurnalsostech.v4i10.23179
[16] T. L. Sinaga, N. Charibaldi, and N. H. Cahyana, "Perbandingan Waktu Respon Aplikasi Database NoSQL Elasticsearch dan MongoDB pada Pengujian Operasi CRUD," JISKA (Jurnal Informatika Sunan Kalijaga), vol. 8, no. 1, pp. 22–35, Jan. 2023, Available from: http://dx.doi.org/10.14421/jiska.2023.8.1.22-35
[17] "Retrofit - A type-safe HTTP client for Android and Java." Accessed: Nov. 15, 2024. Available from: https://square.github.io/retrofit/
[18] "React - The library for web and native user interfaces." Accessed: Nov. 15, 2024. Available from: https://react.dev/
[19] "Midtrans - Payment Gateway Indonesia." Accessed: Nov. 17, 2024. Available from: https://midtrans.com/id

ABOUT THE AUTHORS

Indra Briliansyah is a student currently pursuing a degree in Computer Science at Yogyakarta Technology University, a prominent private university located in Sleman, Special Region of Yogyakarta, Indonesia. With a deep interest in technology and programming, Indra continues to develop knowledge and skills in various aspects of informatics, such as software development, databases, and game development. In addition to focusing on academic studies, Indra is also actively involved in various extracurricular activities that support the enhancement of practical skills and team collaboration. With a commitment to learning and innovation, Indra aspires to contribute to the world of technology, particularly in the field of developing digital solutions that benefit society.

Donny Avianto was born in Yogyakarta, in 1990. He got his Bachelor Degree in 2012 from Universitas Atma Jaya Yogyakarta majoring in soft computing especially pattern recognition, and Master Degree in 2015 from the same university majoring in Soft Computing as well. He got his Doctoral Degree in 2023 form Universitas Gadjah Mada, Yogyakarta, Indonesia. He has been a lecturer at Universitas Teknologi Yogyakarta, Indonesia since 2015. His research interests are in Pattern Recognition, Computer Vision, Image Processing and Deep Learning.