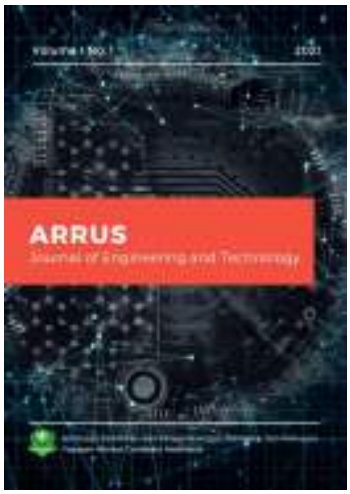
https://doi.org/10.35877/jetech4924

OPEN ACCESS

ISSN : 2776-7914 (Print) / 2807-3045 (Online)




RESEARCH ARTICLE

# WarungSync: Development of an Android-Based Integrated Warung Management Application Using Kotlin and Firebase

**Muh. Akbar* & Ahmad Dinejad Halik**

Department of Informatics and Computer Engineering Education, Faculty of Engineering, Universitas Negeri Makassar, Makassar, South Sulawesi, Indonesia.

*Corresponding author: Muh. Akbar, Department of Informatics and Computer Engineering Education, Faculty of Engineering, Universitas Negeri Makassar, Makassar, South Sulawesi, Indonesia.

E-mail: muh.akbarjaya@unm.ac.id

**Abstract:** This study aims to design and develop WarungSync, an Android-based integrated warung management application intended to assist micro, small, and medium enterprise (MSME) owners in digitalizing their daily business operations. The majority of traditional warung businesses in Indonesia are still managed manually, resulting in inaccurate stock data, unrecorded transactions, and the absence of systematic financial reporting. This study employs the Waterfall software development methodology, consisting of five sequential phases: requirements analysis, system design, implementation, testing, and deployment. The application was built using the Kotlin programming language with Model-View-ViewModel (MVVM) architecture and Firebase as the cloud backend infrastructure, encompassing Firebase Authentication for user security and Firebase Firestore for real-time data synchronization. The results demonstrate that WarungSync successfully integrates Point of Sale (POS) and inventory management functions within a single mobile platform, featuring a real-time dashboard, smart inventory management with low-stock notifications, a digital cashier system, sales reporting with weekly chart visualization, and automated cloud synchronization. The findings of this study reveal that all 13 functional features of WarungSync passed Black Box Testing, confirming that the application operates in accordance with the specified functional requirements and is capable of replacing conventional manual management practices in warung businesses.

**Keywords**: Android Application, Firebase, Inventory Management, Point of Sale, Warung Management

## 1. Introduction

The warung kelontong (small grocery store) represents one of the most prevalent micro-enterprise sectors in Indonesia. According to data from the Ministry of Cooperatives and Small and Medium Enterprises (Kemenkop UKM), millions of warung units are distributed throughout the country, contributing significantly to the national Gross Domestic Product (GDP) and serving as the economic backbone of low-income communities (Kemenkop UKM RI, 2023). Despite their ubiquity, the majority of warung businesses continue to be managed through conventional, manual methods, including handwritten stock records, unrecorded transactions, and the absence of systematic financial reports (Anatan & Nur, 2023).

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

These conditions frequently give rise to critical operational problems, including stock data inaccuracies, difficulty in monitoring sales turnover precisely, and delayed business decision-making due to limited data availability. In the era of rapid digitalization, the penetration rate and availability of Android smartphones among warung business owners is already considerably high; however, adoption for integrated business operations remains extremely limited, with usage largely confined to basic communication (Rahmawati et al., 2023).

Empirical evidence demonstrates that mobile-based management information systems are capable of enhancing operational efficiency, reducing human error, and improving data accuracy in the MSME sector (Putra & Sancoko, 2024). However, existing applications are either too complex for non-technical users, require paid subscriptions, or fail to integrate both POS and inventory management within a single affordable platform. This gap motivates the development of WarungSync as a practical digital solution tailored specifically for warung owners.

This study therefore addresses two primary research questions: (1) How can an Android-based warung management application be designed and developed to assist warung owners in efficiently managing their stock? (2) How can a real-time sales transaction recording system using Firebase Firestore be implemented on the Android platform? The objective is to develop a Kotlin-based Android application integrating POS and inventory management with cloud-based synchronization through Firebase Authentication and Firebase Firestore.

## 2. Literature Review

### 2.1. *Mobile Android Applications*

Mobile applications are software programs specifically designed to run on mobile devices such as smartphones and tablets. Guntara (2022) notes that Android-based mobile applications have become one of the most widely adopted technological solutions in various service sectors due to their accessibility and ease of distribution. The Android platform enables developers to create responsive applications usable by diverse user demographics. Application development generally employs frameworks such as Flutter or native Android using Kotlin or Java. Kotlin, as the officially recommended language for Android development by Google, offers concise syntax and enhanced null-safety features compared to Java, making it the preferred choice for modern Android application development (Sommerville, 2016).

### 2.2. *Warung Management Systems and Point of Sale (POS)*

A warung management system is a digital system designed to assist small business operators in managing their daily operations more efficiently and systematically. Pratama and Somya (2021) demonstrate that an Android-based business management system as an application integrating various operational functions such as stock management, transaction recording, and financial reporting within a single platform accessible through a mobile device.

A core component of warung management systems is the Point of Sale (POS), a system that digitally processes sales transactions. Moroney (2017) explains that modern POS systems not only function as cashier tools but are also directly integrated with inventory management, so that each transaction automatically updates stock data in real-time. This integration is key to reducing manual recording errors that commonly occur in conventional warung operations. Sadali et al. (2023) further assert that the implementation of automatic threshold-based stock notification on mobile applications significantly enhances inventory management efficiency and minimizes the risk of stock-outs.

### 2.3. *Waterfall Development Methodology*

The Waterfall method is a sequential and structured software development model first introduced by Winston W. Royce in 1970. Pressman and Maxim (2019) explain that the Waterfall method is most suitable when system requirements can be fully defined at the outset

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

and are unlikely to change substantially during development. The five main phases are: Requirements Analysis, System Design, Implementation (Coding), Testing, and Deployment & Maintenance. Widianti and Darussalam (2023) validated this approach, demonstrating that the Waterfall method produces well-structured, well-documented systems aligned with end-user requirements.

### 2.4. *Black Box Testing*

Black Box Testing is a software testing method that focuses on the functionality of a system without examining its internal code structure. Testing is conducted by providing specific inputs and verifying whether the resulting outputs match expectations. This method is highly effective for testing user interfaces, system functions, and overall application workflows (Myers et al., 2011). Black Box Testing is particularly appropriate for validating functional requirements in MSME-oriented mobile applications where user-facing behavior is the primary concern.

### 2.5. *Firebase as Cloud Backend*

Firebase is a Backend-as-a-Service (BaaS) platform developed by Google that provides a comprehensive suite of cloud services for mobile and web application development. Firebase Authentication provides secure user identity management through email-password and third-party provider mechanisms, while Firebase Firestore is a NoSQL cloud database that supports real-time data synchronization via snapshot listeners. Moroney (2017) demonstrates that Firebase integration in mobile applications enables rapid data synchronization, flexible access, and adequate data security without the need for a dedicated server infrastructure, making it a cost-effective solution for small business applications.

## 3. Research Method and Materials

This study employs the Waterfall software development methodology to design and develop the WarungSync application. The Waterfall model was selected because the system requirements were clearly definable from the outset, allowing each phase to be executed sequentially and in a structured manner. The development was conducted on the Android platform using Kotlin programming language with MVVM architecture and Firebase as the cloud backend. The following subsections describe each development phase in detail.

### 3.1. *Requirements Analysis*

The requirements analysis phase involved the systematic identification of both functional and non-functional requirements through structured problem analysis of warung management challenges. Functional requirements identified include: (1) user authentication system with registration and login; (2) product data management including add, edit, delete, and restock operations; (3) sales transaction recording with automatic change calculation; (4) sales report display with seven-day graphical visualization; (5) automatic low-stock notification when inventory reaches or falls below five units; and (6) a dashboard displaying warung statistical summaries.

Non-functional requirements include limited offline availability, user data security through Firebase Authentication, and responsive application performance across a range of mid-range Android devices.

### 3.2. *System Design*

The system design phase encompassed the design of application architecture, user interface (UI), and database structure. The application architecture follows the MVVM (Model-View-ViewModel) pattern with Firebase as the backend. The database design uses Firebase Firestore with the following collection structure: (1) users collection storing uid, namaLengkap, email, and namaWarung fields; (2) barang collection storing product data with uid field for user data isolation; and (3) transaksi and notifikasi collections for recording transactions and system-generated notifications respectively.

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

Use Case Diagrams and Activity Diagrams were developed to model user interactions and system flows across all major features, including account registration, login, product management, transaction processing, report viewing, and notification handling. A document-oriented data schema was established to define the collections and fields within the NoSQL architecture prior to database implementation.

**Tabel 1.** Firebase Firestore Database Schema Mapping for WarungSync

| Collection Name | Document Fields | Data Type | Purpose |
|---|---|---|---|
| **users** | id, namaLengkap, email, namaWarung | String | Stores merchant profile and authentication mapping |
| **barang** | id, name, price, stock, unit, category | String, Number | Stores inventory items with user isolation |
| **transaksi** | id, total, cash, change, paymentMethod, timestamp | String, Number | Records digital cashier sales events |
| **notifikasi** | id, type, message, status, timestamp | String | Handles system-generated inventory alerts |

### 3.3. *Implementation*

The implementation phase translated the design specifications into a functioning Android application. WarungSync was developed using Kotlin with Android Studio as the Integrated Development Environment (IDE). The MVVM architecture separates the application into three layers: the Model layer managing data and Firebase interactions, the ViewModel layer handling business logic and state management, and the View layer managing UI rendering and user interaction. Firebase SDK integration was performed for both Firebase Authentication (user management) and Firebase Firestore (real-time database operations), with snapshot listeners enabling automatic UI updates upon data changes without manual refresh.

### 3.4. *Testing*

Application testing was conducted using the Black Box Testing method, evaluating all 13 primary functional features of WarungSync. Each test case defined a specific input scenario, an expected output, and the actual result. Test coverage included authentication flows, all CRUD operations for product management, the complete transaction processing workflow, financial reporting, dashboard statistics, and the automatic notification system.

### 3.5. *Development Tools and Technologies*

The development of WarungSync employed a combination of modern tools and technologies selected to support efficient Android application development with cloud-based backend integration. Kotlin was chosen as the primary programming language due to its official support for Android development and its conciseness compared to Java (Sommerville, 2016). Firebase was adopted as the cloud backend platform, providing authentication, real-time database, and serverless infrastructure without the need for a dedicated server (Moroney, 2017). Table 2 presents a complete summary of the tools and technologies utilized throughout the development process.

**Tabel 2.** Development Tools and Technologies Used in WarungSync

| Component | Technology / Tool | Purpose |
|---|---|---|
| Programming Language | Kotlin | Android application development |
| Architecture Pattern | MVVM | Separation of concerns and maintainability |
| IDE | Android Studio | Development environment |
| Authentication | Firebase Authentication | Secure user identity management |
| Database | Firebase Firestore (NoSQL) | Real-time cloud data storage and synchronization |

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

| Component | Technology / Tool | Purpose |
|---|---|---|
| Minimum SDK | Android 7.0 (API 24) | Broad device compatibility |
| Testing Method | Black Box Testing | Functional feature validation |

## 4. Results and Discussion

### 4.1. *Development Tools and Technologies*

WarungSync is a cloud-integrated warung management system specifically designed to assist MSME operators in digitalizing their daily operations. The application consolidates POS and inventory management functionalities within a single mobile platform, eliminating the need for separate applications to manage sales and stock. Built on Kotlin with MVVM architecture and Firebase as the cloud backbone, all operational data is securely stored in the cloud and accessible in real-time through any Android smartphone.

The application features five core functional modules: (1) Real-time Dashboard displaying monthly turnover statistics and transaction summaries updated automatically via Firebase Snapshot Listener; (2) Intelligent Inventory Management with automatic early-warning notifications when stock reaches a critical threshold (≤ 5 units); (3) Digital Cashier System processing sales transactions with automatic change calculation and atomic stock reduction via Firebase WriteBatch; (4) Business Reports and Analytics providing weekly sales chart visualization and best-selling product analysis; and (5) Cloud Sync and Security utilizing Firebase Authentication for data security and Firestore for cross-device data synchronization.

### 4.2. *Key Feature Implementation*

The user authentication system is implemented using Firebase Authentication with email and password credentials. Upon registration, the application creates a Firebase Auth account and subsequently stores the user profile data (full name and warung name) to the users collection in Firestore. The login process utilizes the signInWithEmailAndPassword method provided by the Firebase Authentication SDK, with password reset functionality supported via email link delivery. The graphical user interface (GUI) of the developed WarungSync application is comprehensively designed to support intuitive user interactions, as illustrated in Figure 1.


(a) (b) (c) (d) (e)

**Figure 1:** WarungSync Application User Interface: (a) User Authentication Login Interface, (b) Real-time Dashboard and Store Overview, (c) Digital Cashier and Product Management System, (d) Sales Analytics and Weekly Chart Reporting, and (e) Automated Low-Stock Notification Alerts.

Product management enables users to add, edit, delete, and restock items. Each product stores attributes including name, price, stock quantity, unit, and category. The application employs a Firestore snapshot listener (addSnapshotListener) enabling data changes to be reflected in the UI in real-time without manual refresh. When a new product is added or restocked, the system automatically generates a notification entry in the notifikasi collection.

The transaction processing feature supports two payment methods: cash (with automatic change calculation) and e-wallet (non-cash). Upon transaction confirmation, the system

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

employs Firebase WriteBatch to atomically write the transaction record and reduce stock quantities in a single operation, ensuring data consistency. Following a successful batch operation, the system inspects the remaining stock of each sold item and generates a low-stock warning notification if the quantity is at or below five units.

The notification system operates automatically without requiring manual user input. Three notification types are generated: TAMBAH_BARANG (when a new product is added), STOK_MENIPIS (when stock reaches or falls below five units after a transaction), and RESTOK (when stock is replenished). Notifications are stored in Firestore and can be marked as read in bulk or deleted individually.

### 4.3. *Black Box Testing Results*

Application testing was conducted using the Black Box Testing method to verify that all functional features operate as specified. Thirteen test cases were evaluated, covering the complete user journey from authentication through reporting. The results are presented in Table 3.

**Tabel 3.** Development Tools and Technologies Used in WarungSync

| No | Feature Tested | Expected Result | Result |
|---|---|---|---|
| 1 | New Account Registration | Account created and user directed to Home page | Pass |
| 2 | User Login | User successfully enters Home page | Pass |
| 3 | Password Reset | Reset link sent to user email | Pass |
| 4 | Add Product | New product saved and appears in inventory list | Pass |
| 5 | Edit Product | Product data changes saved and displayed correctly | Pass |
| 6 | Delete Product | Product removed from inventory list | Pass |
| 7 | Restock Product | Product stock quantity increased per input | Pass |
| 8 | Sales Transaction Processing | Transaction saved; stock reduced automatically | Pass |
| 9 | Cash Change Calculation | Correct change amount calculated and displayed | Pass |
| 10 | Low-Stock Notification | Notification triggered automatically when stock ≤ 5 units | Pass |
| 11 | Real-time Dashboard | Monthly turnover and transaction statistics displayed and auto-updated | Pass |
| 12 | Sales Report | Weekly chart and best-selling products displayed correctly | Pass |
| 13 | Logout | User successfully logged out and returned to Login page | Pass |

As presented in Table 2, all 13 functional features of the WarungSync application passed Black Box Testing without any failures. This outcome confirms that the application has been successfully designed and implemented in accordance with its initial development objectives and fulfills all functional requirements established during the requirements analysis phase.

### 4.4. *Discussion*

The successful development and functional validation of WarungSync demonstrates that Android-based mobile applications with cloud integration represent a viable and practical solution for the digitalization of warung management in Indonesia. The use of the Kotlin programming language combined with the MVVM architectural pattern contributed to a well-structured codebase with clear separation of concerns, facilitating maintainability and future feature extension.

The integration of Firebase Firestore with snapshot listeners proved particularly effective in ensuring real-time data synchronization across the application modules, which is critical for operational scenarios where stock levels and financial data must reflect the current state of the business at all times. The atomic WriteBatch mechanism for simultaneous transaction

https://doi.org/10.35877/jetech4924

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

recording and stock reduction effectively eliminates the data inconsistency risks associated with multi-step manual operations or non-atomic sequential writes.

Compared to existing warung management solutions, WarungSync offers a distinctive combination of POS and inventory management integration within a free, cloud-synchronized, and user-friendly platform. Widianti and Darussalam (2023) similarly demonstrated that the Waterfall methodology produces systems closely aligned with end-user needs; the sequential, phase-by-phase approach adopted in this study yielded a well-documented system with clear traceability between requirements, design decisions, and implemented features.

The automatic three-tier notification system (TAMBAH_BARANG, STOK_MENIPIS, RESTOK) addresses a key operational gap identified in the literature, enabling proactive inventory management without requiring manual monitoring. Sadali et al. (2023) established that such threshold-based notification systems significantly reduce stockout risk; WarungSync operationalizes this principle with a five-unit critical threshold configurable per business context.

The current study is, however, limited to functional validation through Black Box Testing. The absence of field deployment data means that usability outcomes, user acceptance, and actual operational efficiency improvements among warung owners remain to be empirically measured. Future research should incorporate user acceptance testing (UAT) with actual MSME operators and longitudinal field studies to quantify the operational benefits of the application in real warung environments.

## 5. Conclusion

This study successfully designed and developed WarungSync, an Android-based integrated warung management application using Kotlin with MVVM architecture and Firebase cloud services. The application integrates Point of Sale (POS) and inventory management functionalities within a single mobile platform, offering real-time dashboard statistics, intelligent low-stock notifications, digital cashier processing, weekly sales analytics, and automated cloud synchronization.

Implementation of Firebase Authentication and Firebase Firestore provided a reliable and secure cloud backend with real-time data synchronization via snapshot listeners, ensuring operational data is consistently current without requiring manual refresh. The Waterfall development methodology enabled structured, well-documented development with clear phase-by-phase progression from requirements through deployment.

Black Box Testing results confirmed that all 13 functional features of WarungSync operate in accordance with specified requirements, validating the application as a functionally complete solution for warung management digitalization. Future development directions include: (1) integration of net profit calculation with purchase price tracking; (2) financial report export to PDF or Excel format; (3) barcode scanning for inventory input; and (4) Firestore Offline Persistence for uninterrupted operation without internet connectivity.

## References

Anatan, L., & Nur. (2023). Micro, small, and medium enterprises' readiness for digital transformation in Indonesia. *Economies*, *11*(6), 156. https://doi.org/10.3390/economies11060156

Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, *13*(3), 319–340. https://doi.org/10.2307/249008

Giotopoulos, I., Kontolaimou, A., Korra, E., & Tsakanikas, A. (2017). What drives ICT adoption by SMEs? Evidence from a large-scale survey in Greece. *Journal of Business Research*, *81*, 60–69. https://doi.org/10.1016/j.jbusres.2017.08.007

ISSN : 2776-7914 (Print) / 2807-3045 (Online)

Guntara, R. G. (2022). Firebase Realtime Database untuk aplikasi point of sales UMKM berbasis cloud computing pada smartphone Android. *Impression: Jurnal Teknologi dan Informasi*, *1*(2), 50–57. https://doi.org/10.59086/jti.v1i2.75

Kemenkop UKM RI. (2023). *Data perkembangan UMKM di Indonesia tahun 2023*. Kementerian Koperasi dan Usaha Kecil dan Menengah Republik Indonesia. https://www.kemenkopukm.go.id

Moroney, L. (2017). *The definitive guide to Firebase: Build Android apps on Google's mobile platform*. Apress. https://doi.org/10.1007/978-1-4842-2943-9

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). John Wiley & Sons. ISBN: 978-1-118-03196-4

Pratama, R. Y., & Somya, R. (2021). Perancangan aplikasi point of sales (POS) berbasis Android (studi kasus: Warkop Vape Salatiga). *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, *8*(4), 1923–1938. https://doi.org/10.35957/jatisi.v8i4.1218

Pressman, R. S., & Maxim, B. R. (2019). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education. ISBN: 978-1-259-87297-6

Putra, H. B. P., & Sancoko, S. D. (2024). Penerapan sistem point of sale berbasis Android untuk peningkatan kinerja usaha. *Infotek: Jurnal Informatika dan Teknologi*, *7*(1), 195–204. https://doi.org/10.29408/jit.v7i1.23934

Sadali, M., Putra, Y. K., & Anugrah, A. W. (2023). Rancang bangun sistem point of sale berbasis mobile untuk meningkatkan produktivitas usaha. *Infotek: Jurnal Informatika dan Teknologi*, *6*(2), 371–380. https://doi.org/10.29408/jit.v6i2.16811

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education. ISBN: 978-0-13-394303-0

Widianti, L. W., & Darussalam, M. I. (2023). Penerapan metode Waterfall dalam digitalisasi sistem pelayanan publik pemerintah kantor kecamatan Pamulang. *Jurnal Ilmiah Komputasi*, *22*(1), 1–10. https://ejournal.jak-stik.ac.id/index.php/komputasi/article/view/3329

Zaki, N., & Sejati, R. H. P. (2025). Implementasi aplikasi Android dalam sistem restock UMKM Maju Jaya Accessories. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, *6*(1), 318–333. https://doi.org/10.35870/jimik.v6i1.1167