Edu Komputika 11 (1) (2024)

Edu Komputika Journal

https://journal.unnes.ac.id/journals/edukom/

Development of Android Application for Ordering Vegetables and Food Condiments

Adam Hendra Brata✉, Komang Candra Brata, Arya Yudha Mahendra
Department of Informatics Engineering, Faculty of Computer Science, Universitas Brawijaya

Article Info

Article History:
Received: 28 July 2024
Revised: 27 August 2024
Accepted: 29 August 2024

Keywords:
Android, Application, Buy and Sell, Vegetables

Abstract

The growth of internet users continues to increase rapidly each year, supported by the growth of accompanying devices. Mobile applications are the main drivers of the increasing number of internet users. In Indonesia, internet users have reached 47% of the total population. Based on this data, around 46% use it to visit online trading sites. The phenomenon of the rise of online buying and selling is certainly very beneficial, especially for the residents of Malang City, where 34% of its population are traders. Supporting applications for conducting trade are essential for both merchants and consumers to facilitate the business process of buying and selling. This research will study the development of a buying and selling application between sellers and consumers, emphasizing ease of use. The development of this application uses the Mobile-D development model as a reference. The Mobile-D method was adopted because offers a development process well-suited for users who are not directly alongside the developers. Additionally, Mobile-D's iterative nature allows it to accommodate changing requirements, making it ideal for gathering needs directly from the application's end users. Based on the results of the study, user evaluations of the application showed a score more than 70, leading to the conclusion that the developed application is user-friendly and acceptable to users.

✉ Correspondence Address:
Jl. Veteran, Ketawanggede, Lowokwaru, Kota Malang,
Jawa Timur, Indonesia - 65145
E-mail: adam@ub.ac.id

ISSN 2252-6811
E-ISSN 2599-297X

INTRODUCTION

The rapid advancement of technology today is particularly evident in the field of information technology. This is demonstrated using computers, mobile devices, various support applications, and the internet as a medium for information exchange. On average, Indonesians spend 3-4 hours using these technologies, aged 16-64 years. About 46% of them visit online shopping sites, and 33% make online purchases through mobile internet (WeareSocial, 2024). Most of these users, 76.67%, use the Android operating system, 3.38% use iOS and 7.73% are unidentified (Statista, 2024). The data also indicates that 65% of the users are located on Java Island.

A survey was also conducted among mobile device users in one area of Malang City, East Java. The survey involved 40 people of various ages and genders, focusing on their daily grocery shopping habits. The author also surveyed suppliers, specifically five mobile vegetable sellers. Approximately 129,000, or 34% of Malang City's population, work as traders, including mobile vegetable sellers (BPS, 2024). The survey included questions about the factors influencing consumers' shopping and their satisfaction with the current conditions. For the sellers, the survey contained questions about the challenges they face in their business. The results showed that only 27.5% of respondents were satisfied with the current buying and selling conditions. Furthermore, 85% of respondents indicated the need for an application to support the buying and selling process. Among the seller, 100% reported experiencing losses in their business, ranging from 1-5%. Additionally, 100% of these sellers expressed interest in an application that could assist them in their sales process.

Based on this data, the author attempted to develop an Android-based application to meet the community's high mobility needs (Sufandi & Trihapningsari, 2022). The application, named Mlijo, is designed to be easily used by mobile vegetable sellers and consumers, facilitating the buying and selling process wherever they go. According to research conducted by Ramani et al, developing Android applications can significantly benefit public services, particularly in activities that involve many users in the service process (Ramani et al., 2024). Focused on user ease in understanding the application, the author employs the Mobile-D method, a derivative of the Agile methodology, to produce a high-quality product (Castilla et al., 2023).

RESEARCH METHODS

This section outlines the systematic steps involved in developing the Mlijo software. It serves as a workflow guide for the research, ensuring that the process proceeds systematically according to this workflow. The development of this application uses the Mobile-D development model as a reference (Castilla et al., 2023). The flowchart of the research steps is shown in Figure 1.

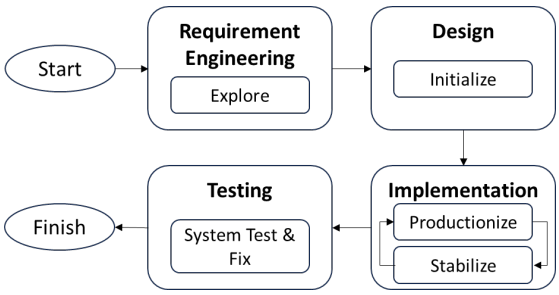


Figure 1. Research methodology

A. Requirement Engineering

Requirements engineering involves the process of identifying and establishing the conditions that must be met in the development or modification of a product, considering the various interrelated needs of stakeholders. The requirements engineering phase in this study adopts the explore phase from the Mobile-D method. In this phase, the adoption of the Mobile-D method includes customer establishment, initial requirements collection, environment selection, and architecture line definition When conducting research focused on software development, a crucial factor is determining user requirements. Models are used to gain an analytical perspective in obtaining these requirements. At this stage, the author analyzes the system requirements to provide an overview of the system development to be carried out. In this process, information from stakeholders regarding the desired system is gathered, and specifications for the software to be implemented are obtained. This analysis involves understanding the system process flow and identifying the needs and features that the application will include (Buede, 2024).

B. Design

After all the software system requirements are gathered and verified through the requirements analysis stage, the software design phase is carried out. The design phase adopts the initialize phase from the Mobile-D method. The initialize phase consists of high-level design. The objective of this stage is to design the system by

taking into account the information obtained in the previous stage. The system design phase generally aims to provide an overall explanation of the system's workflow (Syahputra & Musril, 2022). In this design phase, the process is divided into four parts, data design, architecture design, interface design and algorithm design. In data design, the database design is created to store all the information needed by the system. Architecture design involves designing the framework and workflow of the system to be developed. Interface design is created based on the results of the requirements analysis and the solutions to the problems. Algorithm design is based on the defined functional requirements. This algorithm is represented in the form of pseudocode, which will then be implemented into the program.

C. Implementation

During the implementation phase, the system is executed based on the design described in the system design phase. The implementation stage adopts the production and stabilizes phases from the Mobile-D method. In the production phase, the planning days, working days, and release day stages are adopted. The results from the production phase are then used for requirements analysis in the stabilization phase. The stabilizing phase is generally similar to the production phase, with the addition of a documentation wrap-up stage to complete the documentation in detail. The production phase in Mobile-D uses iteration methods together with the stabilizing phase to build better program implementation. This implementation phase details the comprehensive execution of the system as a solution to the identified problems. It is also a systematic process undertaken to complete the approved and validated system design. The system implementation is carried out using the Java programming language, which is used to build the Android application. The system implementation begins by identifying the hardware specifications, software specifications, and mobile device specifications. Once the specifications for building the Mlijo application are determined, the next step is to define the scope of the implementation. This is followed by the implementation of the database, class implementation, source code implementation, and finally, the interface implementation.

D. Testing

At this stage, a detailed explanation is provided regarding the testing process conducted to ensure that the developed system functions properly, meets quality standards, and satisfies the established requirements. System testing in Mobile-D is the stage where the system is thoroughly tested. Any errors found are documented for the purpose of iterative improvement, after which the developer fixes the code by adopting task refactoring. The testing in this research is divided into two types: functional testing and non-functional testing. Functional testing uses validation testing, while non-functional testing consists of usability testing and compatibility testing. Usability testing assesses whether the developed system is easy to learn and use. The usability testing involves 15 respondents, including 10 consumers and 5 sellers. Finally, compatibility testing is performed to determine whether the developed application functions well across different smartphones.

RESULT AND DISCUSSION

A. Requirement Engineering

Mlijo application operates through two main mechanisms: the consumer application and the seller application. The general workflow of the Mlijo application is illustrated in Figure 2. This application's operation requires an internet connection to access data stored in the Firebase database.

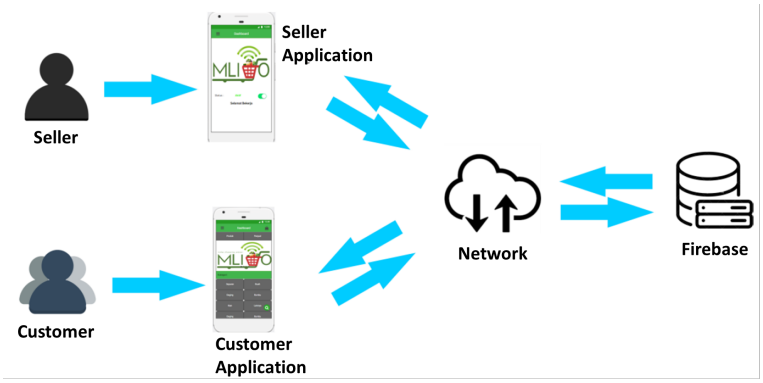


Figure 2. Workflow mechanism

The process of gathering software requirements was conducted through interviews with several respondents, divided into two groups: respondents for the consumer application, who place orders, and respondents for the seller application, who are the owners or providers of products. The results of these interviews form the basis for defining the requirements. Table 1 shows the identified actors in this research. After identifying the actors, the requirements definition was clarified and specified in more detail in the requirements specifications. During the requirements engineering phase, 21 functional requirements were identified for the consumer side and 20 functional requirements for the seller side, as shown in Tables 2 and Table 3.

Table 1. Actor Identification

No	Actor	Description
1	Consumer	These are the users of the kitchen and vegetable ordering system in Malang City, who can purchase products through the system.
2	Seller	These are the users of the kitchen and vegetable ordering system in Malang City who can sell products through the system.

Table 2. Functional Requirements Specification for Consumer Actor

No	Function	Description
1	Authentication	The system must provide a facility for authentication where users can enter their phone number to log into the system.
2	Search for Products	The system must provide a product search facility based on category or using a search field.
3	Apply Search Filters	The system must provide a filtering facility for searches based on price, category, and location.
4	View Product List	The system must provide a facility to view a list of all available products.
5	View Product Details	The system must provide a facility to view detailed information about the product selected by the consumer.
6	Make Orders	The system must provide a facility to place orders for products from available sellers.
7	Make Special Orders	The system must provide a facility to place special orders for products not listed in the product catalog with specific seller.
8	View Transaction Details	The system must provide a facility to view detailed transaction information for products.
9	View Transaction List	The system must provide a facility to view a list of transactions.
10	Check Order Status	The system must provide a facility to display information related to order status.
11	Confirm Receipt	The system must provide a facility to confirm that the product has been received by the consumer.
12	Write Reviews	The system must provide a facility to leave reviews for sellers.
13	View Transaction History	The system must provide a facility to display the consumer's transaction history.
14	View Current Pricing	The system must provide a facility to display current pricing information.
15	View Active Seller Locations	The system must provide a facility to display the locations of active sellers around the consumer.
16	View Seller Profiles	The system must provide a facility to view seller profiles.
17	Send Chats	The system must provide a facility to send chats to sellers.
18	Manage Profile	The system must provide a facility to manage the consumer’s profile.
19	Cancel Orders	The system must provide a facility to cancel orders before the seller confirms receipt of the order.
20	Display Notifications	The system must provide a facility to display notification messages under certain conditions.
21	Logout	The system must provide a facility to log out of the system.

Table 3. Functional Requirements Specification for Seller Actor

No	Function	Description
1	Authentication	The system must provide a facility for authentication where users can enter their phone number to log into the system.
2	Manage Product Data	The system must provide a facility to manage product data for items being sold.
3	Create Product Data	The system must provide a facility to create new product data.
4	Edit Product Data	The system must provide a facility to edit existing product data.
5	Delete Product Data	The system must provide a facility to delete existing product data.
6	View Product List	The system must provide a facility to view data on products being sold.
7	Manage Sales	The system must provide a facility to manage sales.
8	View New Orders	The system must provide a facility to view a list of the latest product orders from consumers.
9	View Order Details	The system must provide a facility to view detailed information about each consumer's order.
10	Confirm Orders	The system must provide a facility to confirm orders from consumers on the seller side.
11	View Map to Consumer Location	The system must provide a facility to view a map/route from the seller's location to the consumer's location.
12	View Sales Status	The system must provide a facility to view a list of sales statuses.
13	Update Sales Status	The system must provide a facility to update sales statuses.
14	View Transaction History	The system must provide a facility to view the transaction history.
15	Manage Account Activation	The system must provide a facility to manage account activation status.
16	Manage Profile Data	The system must provide a facility to manage seller profile data.
17	View Current Pricing Information	The system must provide a facility to display current pricing information.
18	Send Chats	The system must provide a facility to send chats to consumers.
19	Manage Locations	The system must provide a facility to manage the seller's location settings.
20	Logout	The system must provide a facility to log out of the system.

B. Design and Implementation

System design is carried out after the requirements analysis process is completed. The system developed in this research uses an n-tier architecture (Smys, 2020). This architecture separates the user interface processing from the data processing within the back-end architecture. Figure 3 illustrates the system architecture developed in this research.

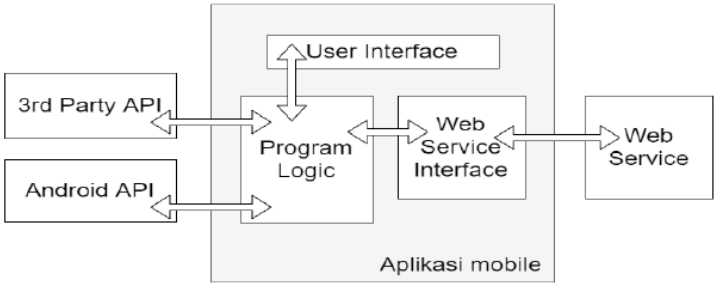
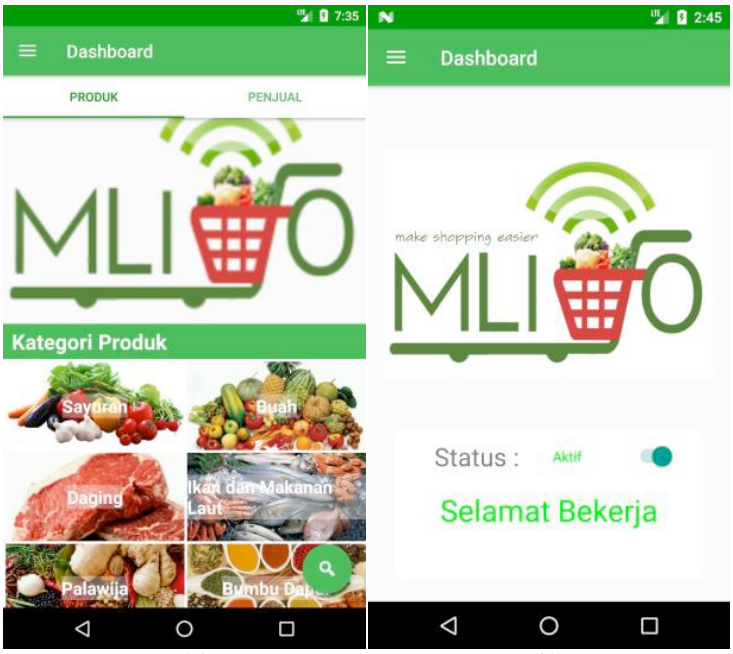


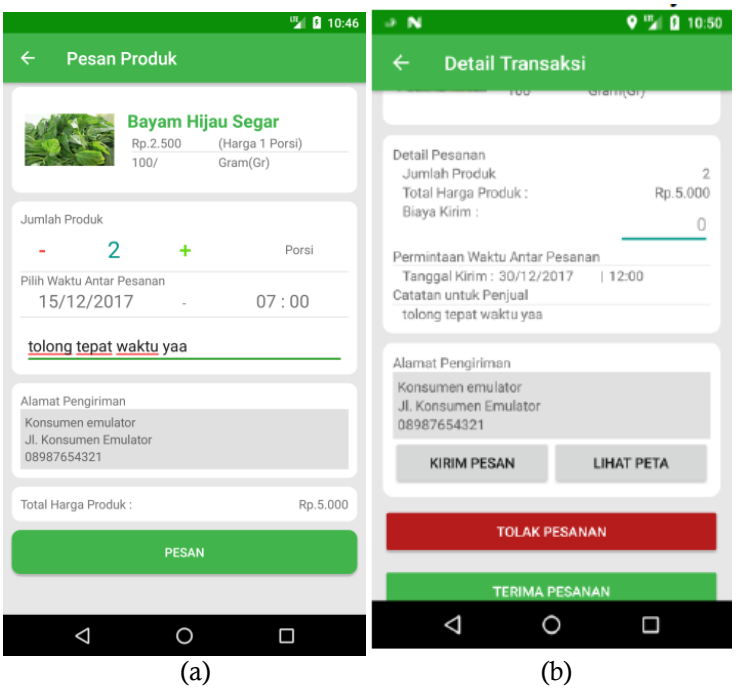
Figure 3. Application architecture

The implementation results of the Mlijo application are divided into two parts: the consumer application and the seller application. Figure 4 (a) shows the main page of the consumer application, which is used to display the product categories available for order. Figure 4 (b) shows the dashboard view of the seller application, which can be used to activate the seller's status.



(a) (b)

Figure 4. Main page view (a) consumer application (b) seller application dashboard



(a) (b)

Figure 5. Product order page view (a) consumer application (b) seller application

Figure 5 displays the product order detail page within the application. In part (a), the consumer application provides detailed information about the orders placed. In part (b), the seller application displays the same

information and includes additional functionality to confirm the receipt of consumer orders.

C. Testing

The testing conducted during this system's development includes validity, usability, and compatibility testing. Validation testing is used to determine whether the system that has been built is correct and meets the requirements list. The items outlined in the requirements list, which result from the needs analysis, will serve as the reference for conducting validation testing (Zafrullahx et al., 2023). The validation testing results are analyzed by comparing the test outcomes with the previously documented requirements list. If the test results align with the system scenarios, then the system is considered valid because it has implemented the functional requirements or needs. The testing result of the Mlijo application is shown in Table 4.

Table 4. Validation Testing Result for Mlijo Application

No	Application	Number of Functions	Validity Rate
1	Consumer	21	100%
2	Seller	20	100%

The validation testing result shows that all of the functionalities of the Mlijo Application have met the predefined functional requirements because most of the tested functional requirements produced valid test results.

Usability testing aims to assess user feedback or satisfaction with the developed application. User satisfaction is measured using the System Usability Scale (SUS) questionnaire, a usability testing method (Mohtar, 2023). The SUS questionnaire used is shown in Table 5 (Usman & Gustalika, 2022).

Table 5. SUS Questionaire

No	Function
1	I will use the Mlijo System to place orders or sell kitchen products.
2	I find that this system provides a complex workflow.
3	I find that this system is easy to use.
4	I need assistance from others to use this system.
5	I find that this system provides functions that integrate well with each other.
6	I find that this system contains many inconsistencies.
7	I believe many people will easily use this system to place orders or sell kitchen products.
8	I find that this system is very complicated to use.
9	I feel that I can use this system effectively.
10	I need to learn more to use this system effectively.

The SUS scores obtained from the usability testing are categorized into several groups. The first category, with a score range of 0 – 50, falls into the "Not Acceptable" category. The second category, with a score range of 51 – 70, is classified as "Marginal," and the third category, with a score range of 71 – 100, is considered "Acceptable" (Schrepp, Kollmorgen, & Thomaschewski, 2023).

The usability testing involves 15 respondents, including 10 consumers and 5 sellers. The usability testing results for the Mlijo application showed an average SUS score of 87.5 for the consumer side and 72.5 for the seller side, indicating that the application is easy to use and generally acceptable to users. The results of the usability testing for the Mlijo application are shown in Figure 6.

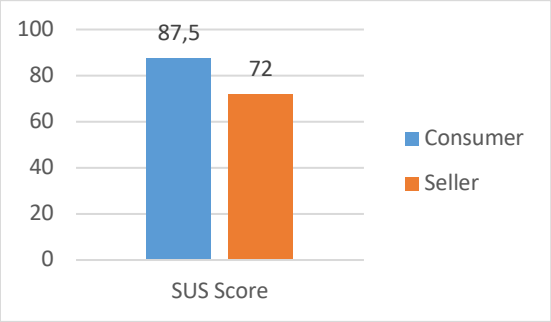


Figure 6. Average SUS score

Compatibility testing aims to validate the dependency between the tested mobile application and its various operating environments. The results of this testing serve as a benchmark to determine whether the application can run on different Android operating system versions. In this study, the Firebase Test Lab was used to conduct this testing. The testing result is shown in Table 6.

Table 6. Compatibility Testing Result

No	Testing Device	Result
1	Nexus 5, Virtual, API Level 19	Success
2	Nexus 5, Virtual, API Level 21	Success
3	Nexus 5, Virtual, API Level 22	Success
4	Nexus 5, Virtual, API Level 23	Success
5	Nexus 5, Virtual, API Level 24	Success
6	Nexus 5, Virtual, API Level 25	Success
7	Nexus 5, Virtual, API Level 26	Success

Based on the compatibility testing conducted on the application with Android test devices ranging from API 19 (KitKat) to API 26 (Oreo), the results showed a status of success. This indicates that the application can run on multiple devices with different Android API versions (Mahmud, Che, & Yang, 2023).

CONCLUSION

The system development process successfully produced an application that functions well. The testing results indicate that the application effectively meets user needs. Validation testing shows that all of the user requirements were implemented successfully. Compatibility testing shows that the Mlijo application can run well on almost all of the Android versions used in smartphones. User evaluation shows a high level of satisfaction, with an average SUS score of 87.5 from consumers and 72.5 from sellers. This demonstrates that the application is easy to use, well-received by its users, and operates across various versions of Android.

REFERENCES

BPS. (2024). *Proporsi Lapangan Kerja Informal Sektor Non Pertanian*.

Buede, D. a. (2024). *he engineering design of systems: models and methods.* John Wiley & Sons.

Castilla, R., Pacheco, A., & Franco, J. (2023). Digital government: Mobile applications and their impact on access to public information. *SoftwareX*.

Mahmud, T., Che, M., & Yang, G. (2023). Detecting android api compatibility issues with api differences. *IEEE Transactions on Software Engineering*, 3857-3871.

Mohtar, S., Jomhari, N., Omar, N., Mustafa, M., & Yusoff, Z. (2023). The usability evaluation on mobile learning apps with gamification for middle-aged women. *Education and Information Technologies*, 1189-1210.

Ramani, K., Jyothi, A., Kinjarapu, S. R., Preethi, K., & Reddy, K. U. (2024). An Android Application for Temporary Driver Booking System. *International Conference on Pervasive Computing and Social Networking (ICPCSN)*, (pp. 736-741).

Schrepp, M., Kollmorgen, J., & Thomaschewski, J. (2023). A Comparison of SUS, UMUX-LITE, and UEQ-S. *Journal of User Experience*.

Smys, S. (2020). A novel multi-tier architecture based mobile cloud computing for enhanced energy utilization. *Journal of ISMAC*.

Statista. (2024). *Smartphone penetration rate in Indonesia from 2020 to 2024 with forecasts until 2029*.

Sufandi, U., & Trihapningsari, D. (2022). Pengembangan Aplikasi Mobile SITTA Universitas Terbuka Berbasis Android. *INSERT: Information System and Emerging Technology Journal*.

Syahputra, Y., & Musril, H. (2022). Perancangan Aplikasi Bursa Kerja Khusus di SMKN 1 Bukittinggi. *Edu Komputika Journal*.

Usman, M., & Gustalika, M. (2022). Pengujian Validitas dan Reliabilitas System Usability Scale (SUS) Untuk Perangkat Smartphone. *Jurnal Ecotipe (Electronic, Control, Telecommunication, Information, and Power Engineering)*.

WeareSocial. (2024). *Digital in 2024: Indonesia Overview*.

Zafrullah, Z., Suyanto, S., Wahyuni, A., Ayuni, R., & Novilanti, F. (2023). Development of android-based learning to improve computational thinking skills in junior high school. *Jurnal Cendekia: Jurnal Pendidikan Matematika*.