Journal of Media, Sciences and Education
https://ojs.institutemandalika.com/index.php/jomet

Vol. 4, No. 4, 2025
e-ISSN: 2986-2728
hh . 168-181

Implementation *Extreme Programming* in Development Application Management Shop Based *Android*

Deni Sanjaya [1] , Sri Wahyuni [2] , Nadia Rista [3]
[1.2] Informatics Engineering Study Program , Faculty Science and Technology ,
[3] Economic Education Study Program , Faculty Teaching and Education Science
University Five Powers of Bekasi
*Corresponding Author e-mail: denisanjaya720@gmail.com ,
Email: sriyuni82.sw@gmail.com

Abstract
Ferry Electronic Store still conducts its operational activities using conventional methods, particularly in recording product data and sales transactions, which leads to several issues such as slow data retrieval, risk of data loss, and low efficiency and accuracy in inventory management. This study aims to design and develop an Android-based store management application to improve the effectiveness of store operations. The system was developed using the Extreme Programming (XP) methodology, which consists of planning, design, coding, and testing phases. Data collection techniques included observation, literature study, and interviews. The application was developed using Kotlin in Android Studio with Firebase Realtime Database as the database system and is equipped with a QR Code scanning feature to accelerate product identification and transaction processes. The results indicate that the developed application is user-friendly, provides an informative interface, and effectively supports inventory management, sales transactions, and real-time reporting in an accurate and structured manner. The implementation of Extreme Programming enables an adaptive development process that meets user requirements and significantly enhances operational efficiency at Feri Electronic Store.
Keywords *: Extreme Programming, QR Code, Android .*

How to Cite: Deni Sanjaya, Sri Wahyun [2] , Nadia Rista . (2025). Application of Extreme Programming in Development Application Management Shop Based Android . Journal of Media, Sciences, and Education, 4(2), 88–91. https://doi.org/10.36312/jomet.v4i2.131

https://doi.org/10.36312/jomet.v3i4.131

Introduction

In the modern digital era, mobile applications have become an integral component of both business operations and consumer interactions. The proliferation of smartphones and mobile platforms has created a dynamic environment where businesses, including retail and shop-based enterprises, are increasingly reliant on mobile applications to manage operational processes and enhance customer engagement (Bhimanapati, Jain, & Aggarwal, 2023). Specifically, Android-based applications dominate the global mobile ecosystem due to their affordability, flexibility, and extensive developer community support (Mounir, 2023). In this context, efficient and adaptive software development methodologies are crucial for ensuring high-quality, timely, and cost-effective software delivery.

Extreme Programming (XP), a subset of Agile methodologies, has emerged as a practical and effective approach for developing software in dynamic and rapidly changing environments (Abrahamsson & Koskela, 2004; Capiluppi et al., 2007). XP emphasizes iterative development, continuous feedback, and collaboration between developers and stakeholders, enabling software teams to respond rapidly to changing requirements while maintaining high standards of code quality (Serrador & Pinto, 2015). Unlike traditional waterfall models, which follow a linear progression of requirements gathering, design, implementation, and testing, XP promotes flexible cycles of planning, coding, testing, and customer evaluation. This

adaptability is particularly beneficial for small- and medium-sized enterprises (SMEs) and shop-based applications where market conditions and user preferences can change rapidly (Conforto & Amaral, 2020).

The retail sector, especially shop-based enterprises, faces unique challenges in managing operational and transactional processes. These businesses often require real-time inventory management, customer tracking, sales reporting, and integration with payment systems. Mobile applications that support these functionalities need to be highly responsive, scalable, and secure. Traditional software development methodologies may not be agile enough to accommodate these requirements efficiently (Chow & Cao, 2008). Consequently, adopting Extreme Programming offers significant advantages for shop-based application development by enhancing responsiveness, ensuring frequent delivery of functional components, and facilitating stakeholder involvement throughout the development process (Awad, 2017).

Several studies have highlighted the effectiveness of XP in software development projects across various domains. Abrahamsson and Koskela (2004) conducted a controlled case study demonstrating that XP practices, including pair programming, continuous integration, and test-driven development, significantly enhance software quality and team productivity. Similarly, Capiluppi et al. (2007) examined the evolution of agile-developed systems and found that iterative feedback loops and customer collaboration contribute substantially to project success. Moreover, Chow and Cao (2008) identified critical success factors in agile projects, emphasizing that stakeholder involvement, team communication, and adaptive planning are essential for achieving desired outcomes. These findings underscore the relevance of XP practices in environments characterized by frequent changes and uncertainty, such as shop-based Android applications.

The application of XP in mobile application development has been increasingly documented in both national and international contexts. Bhimanapati, Jain, and Aggarwal (2023) highlighted the role of agile methodologies in mobile app development, noting that XP practices facilitate rapid prototyping, iterative testing, and continuous improvement. Mounir (2023) similarly emphasized that software engineering challenges in mobile application development, including device heterogeneity, user interface design, and performance optimization, can be effectively addressed through agile approaches like XP. Additionally, empirical research conducted by Awad (2017) and Dingsøyr, Prieto, and Karlsen (2018) demonstrates that XP contributes to higher team performance and project success in agile software projects, which is particularly relevant for small, shop-based enterprises where development teams are often compact and cross-functional.

The implementation of XP in shop-based application development involves a series of structured practices aimed at maximizing efficiency and software quality. Core XP practices such as pair programming, continuous integration, test-driven development, and frequent releases allow developers to identify and resolve issues early, maintain high code quality, and adapt quickly to evolving business requirements (Isnanto, Alkodri, & Supardi, 2020). For example, in the context of an online retail store application, XP enables developers to iteratively design modules for inventory management, order processing, and payment integration while incorporating feedback from store managers and customers. This iterative process ensures that the final product aligns closely with business objectives and user expectations, reducing the risk of costly post-deployment modifications (Maulidah et al., 2023; Permana & Utami, 2023).

In addition to methodological benefits, XP contributes to enhanced stakeholder collaboration and engagement. Continuous involvement of customers and end-users throughout the development cycle ensures that requirements are accurately captured and implemented. Rahman et al. (2022) demonstrated that integrating XP in mobile application development enhances communication between developers and stakeholders, resulting in applications that are not only functional but also user-centric. Similarly, Rokhim, Azhar, and Saputra (2024) found that XP practices in SME software projects led to improved stakeholder satisfaction and faster delivery of functional modules. In the context of shop-based Android applications, this approach ensures that operational features such as point-of-sale management, sales analytics, and customer loyalty programs are delivered according to business priorities and user needs.

The adoption of XP also addresses common challenges associated with mobile application development. Android applications, in particular, require frequent updates to maintain compatibility with new operating system versions, device models, and security standards. Traditional development approaches often struggle to accommodate such frequent changes without significant time and resource investment (Suhadi, Hariyanto, & Sastra, 2021). XP mitigates this challenge by promoting incremental development and continuous integration, allowing teams to implement small, manageable updates while preserving system stability (Hidayat, Sakethi, & Ardiansyah, 2022). Furthermore, XP practices such as refactoring and automated testing contribute to maintainable and reliable codebases, which are essential for sustaining long-term application performance in the competitive mobile market (Misra, Kumar, & Kumar, 2009).

In Indonesia, several studies have explored the integration of XP in mobile and web-based applications. Isnanto, Alkodri, and Supardi (2020) documented the successful application of XP in developing a student information system, demonstrating that agile practices can improve development speed and software quality. Similarly, Mahesa, Hayati, and Mardiani (2024) applied XP in the context of a laboratory information system, achieving improved stakeholder communication and iterative delivery of functional modules. National studies by Maulidah et al. (2023) and Permana and Utami (2023) provide empirical evidence of XP's effectiveness in mobile application projects, particularly those oriented toward retail and user-centric services. These studies collectively reinforce the suitability of XP for shop-based Android application development, where rapid response to business requirements and user expectations is critical.

Despite the demonstrated advantages, implementing XP is not without challenges. Teams must maintain discipline in practices such as pair programming, test-driven development, and continuous integration, which may require cultural adaptation and skill development (Tetteh, 2024). Additionally, effective communication among team members and stakeholders is essential to prevent misalignment and ensure that iterative deliverables reflect evolving business needs (Serrador & Pinto, 2015). Nevertheless, when properly adopted, XP provides a structured yet flexible framework that enhances software quality, development speed, and stakeholder satisfaction.

The significance of applying XP in shop-based Android application development extends beyond technical efficiency. By enabling iterative and responsive development, XP empowers businesses to deliver applications that closely align with operational goals and customer expectations (Yusuf Bakhtiar, Suwela, & Ismaillah, 2024). For instance, online stores can leverage XP to deploy modules for inventory updates, order tracking, and customer

relationship management in short cycles, thereby enhancing competitiveness and market responsiveness (Akbar & Haryanti, 2023). Additionally, the focus on continuous testing and quality assurance reduces the likelihood of defects, minimizes downtime, and improves user satisfaction, which are critical success factors for digital retail operations (Chow & Cao, 2008; Awad, 2017).

The objectives of this study are therefore twofold: first, to investigate the practical implementation of Extreme Programming in the development of shop-based Android applications, and second, to evaluate its impact on software quality, development speed, and stakeholder satisfaction. By integrating empirical insights from both national and international research, this study aims to provide a comprehensive framework for developers and businesses seeking to adopt XP in mobile application projects (Capiluppi et al., 2007; Rahman et al., 2022). The scope of the research focuses specifically on Android-based applications for shop management, including modules for inventory, sales, and customer interactions. This focus is motivated by the widespread adoption of Android devices in Indonesia and other emerging markets, making it a practical platform for business-oriented mobile solutions (Mounir, 2023).

In conclusion, the implementation of Extreme Programming in the development of shop-based Android applications offers substantial benefits in terms of agility, code quality, stakeholder engagement, and responsiveness to business requirements. Existing literature, both national and international, supports the efficacy of XP practices in enhancing project success, improving team performance, and delivering software that meets user expectations (Abrahamsson & Koskela, 2004; Conforto & Amaral, 2020; Misra, Kumar, & Kumar, 2009). By applying these practices in the context of shop-based applications, developers can create robust, user-centric solutions that support operational efficiency, customer satisfaction, and business competitiveness. The current study builds upon these insights, aiming to contribute empirical evidence and practical guidelines for implementing XP in Android application development for retail and shop-based enterprises.

Method Study

Method design application is something stages carried out For development or development application , using method *Extreme Programming* (XP). Method This chosen writer Because the application that will built need change fast *requirements* and limited time as well as team in scale small . As for stages in methodology development XP system , namely *planning* , *design* , *coding* , *testing* .

1. *Planning*

Planning is phase first on the method *extreme programming* where in the phase This done *brainstorming* and analysis For determine planning beginning development device soft , start from determine feature What only one will entered to in application said , determines specification *minimum* , and determine method *testing* that will be used when device soft Ready For released .

2. *Design* (Planning)

Design or design is stage to two on the method *extreme programming* where at the stage This writer designing logic from application or the system to be built like UML creation consisting of from *use case diagrams, class diagrams, activity diagrams, sequence diagrams, as well as middle fidelity wireframes and high fidelity wireframes.*

3. *Coding*

At the stage This writer will start implement results design from stages previously to in code with use *Kotlin* on *Android Studio* and not only That just writer will make *database* from *firebase realtime database* which will be can be accessed by the *user* .

4. *Testing*

At the stage This will done testing to applications that have been built previously , where testing This will use method *black box testing* that focuses on requirements functional for users

Study This done in the shop Ferry Electronics , located at Jl. Monumen Pancasila Sakti RT 03/ RW 06 No. 6, Cipayung , East Jakarta. study This implemented during period from May 2025 to June 2025, covering stage data collection (observation , study) literature , and interviews), analysis needs , design interface , development application use method *Extreme Programming* (XP), testing system , up to implementation limited

Results and Discussion

Planning

Planning is stage first on the method *Extreme Programming* (XP) where at this stage This done *brainstorming* and analysis For determine planning beginning development device soft , start from determine feature What only one will entered to in application said , determines minimum specifications , and determine testing methods that will be used when device soft Ready For released .

***Design* (Planning)**

Use Case Diagram

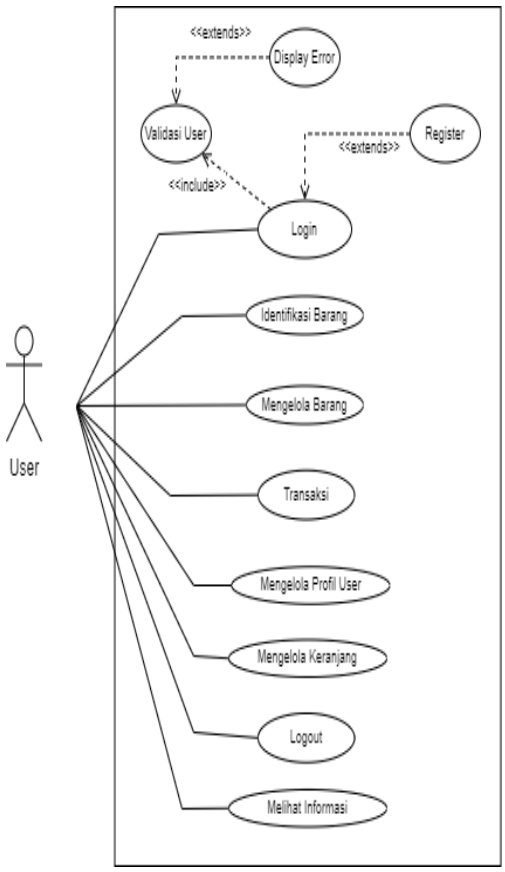

Figure 1 *Use Case* Diagram

Class Diagram

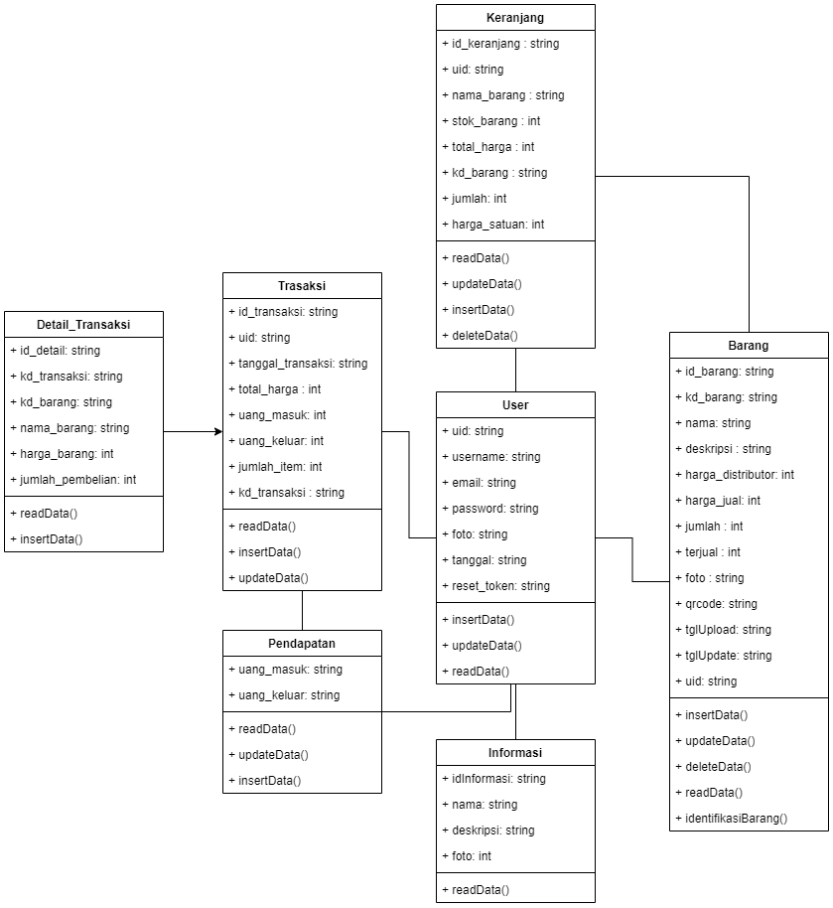

Figure 2 *Class* Diagram

Application This own seven entity Main : User, Item , Cart , Transaction , Transaction Detail , Income , and Information .

1. *User* store user data and connect with basket as well as transactions .
2. Goods includes product details , price , stock , sales , and QR *code* .
3. Basket save a list of items to be purchased by the user.
4. Transaction record summary purchase , while Detail Transaction take notes details each goods in transactions .
5. Income take notes daily cash flow .
6. Information provide content guide for users and only Can read .

Every class generally equipped with CRUD method according to its function , except class Information that is of a nature *read only* and class transaction_details which only can add as well as reading data. Structure *class diagram* This made For ensure the process of managing user data , products , transactions and information shop can walk in a way efficient , systematic , and appropriate with need operational shop .

Activity Diagram

1. *Activity Register*

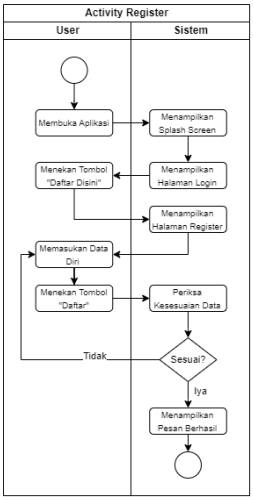

Figure 3 *Activity Register*

Based on picture *activity* diagram above can seen that system started moment user open application , then display *splash screen* for 3 seconds , continue with page *login* . *User* pressing " Register Here " button for going to page *register* , then fill in *username, email, password* , and verification *password* . After pressing knob save , system inspect completeness and suitability of data. If valid, the data is saved to *database* and system display return page *login* .

2. *Activity Login*

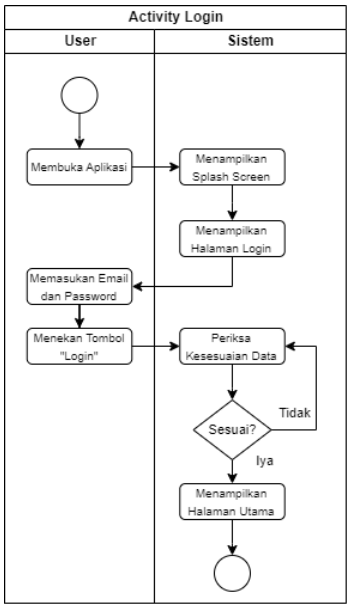

Figure 4 *Activity Login*

Based on picture *activity* diagram above can seen that system started moment *user* open application and display *splash screen* for 3 seconds , then to be continued with page *login* . *User* enter *email* and *password* , then pressing knob *login* . System will do validation against the data in *the database* . If valid, the system display page homepage ; if no , *user* requested enter repeat *email* and *password* .

3. *Activity* Forgot the password

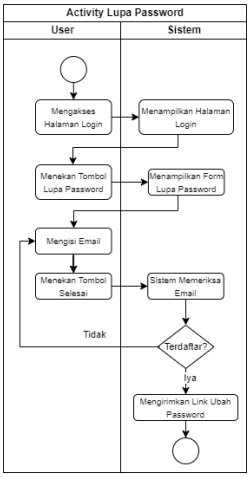

Figure 5 *Activity* Forget *Password*

System started when the user opens application , then display *splash screen* for 3 seconds and continue to page *login* . *User* pressing " Forget " button *Password* ", then system display page forget *password* . There , *user* enter the registered email and press knob finished . System will do validation . If valid *email* , system send *password update link* to *gmail user* and display page *login* . If no , system display message *error* , and *user* must enter *e-mail* return .

4. *Activity* Plus Goods

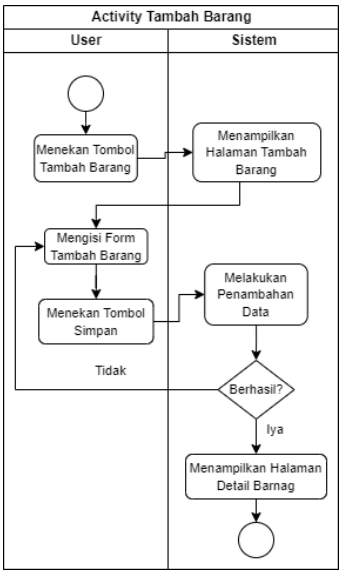

Figure 6 *Activity* Plus Goods

System started moment *user* pressing knob plus items on *the dashboard* , then system display page plus goods . *User* fill in the item data and press knob save . If storage success , system display item detail page . If failed , system request *user* For fill in repeat item data .

5. *Activity Update* Goods

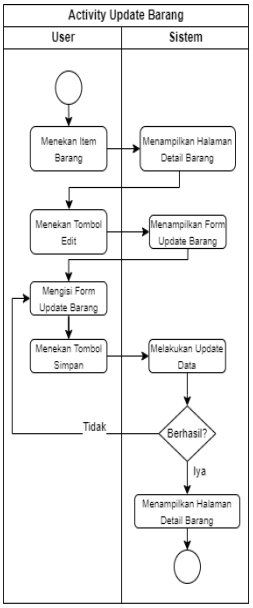

Figure 7 *Activity Update* Goods

System started when *user* press the item , then system display item details page . Next , *the user* pressing knob plus items on *the dashboard* application , then system display page plus goods . *User* must enter the item data , then pressing knob save . If success , system display item details . If failed , system ask the user to fill in repeat item data .

6. *Activity Delete* Goods

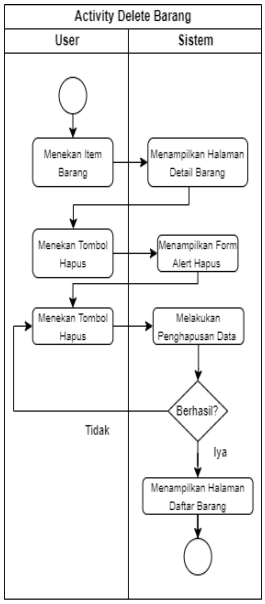

Figure 8 *Activity Delete* Goods

Based on picture activity *diagram* above , system started moment *user* press the item , then system display item detail page . *User* pressing knob delete , and system will delete data in *the database* . If success , system display page plus goods , where the user must enter item data Then pressing knob save . If storage success , system display item details . If failed , system display item list page with the updated list . However , if no , the user must try click knob wipe again on the item details page .

7. *Activity Update* Profile

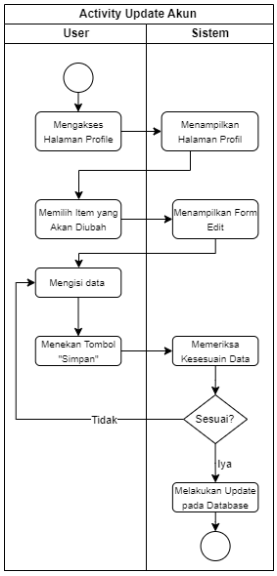

Figure 9 *Activity Update Profile*

Based on picture activity *diagram* above , system started with *user* access page profiles and systems display page *profile user* . *User* select the item to be changed , then system display the edit form according to the item . After *the form* appear , *user* fill and click knob Save . System Then validate data; if Already according to the system change data in *the database* , if no , system display message *error* and *user* must fill in *form* with Correct .

8. *Activity* Transaction

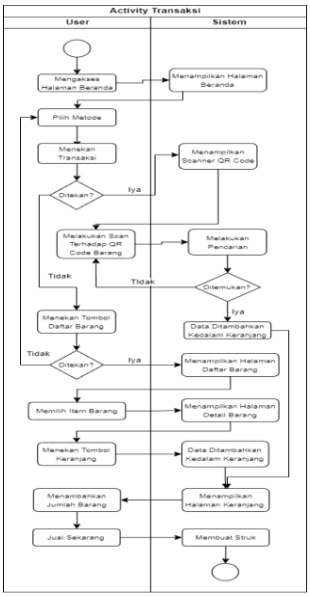

Figure 10 *Activity* Transaction

Based on picture *activity* diagram, system started when the user accesses page homepage . *User* choose method transaction :

1. *Scan QR code* – *User* pressing knob Transactions , systems display *scanner* , *user* do *scan* , item data added to basket , *user* enter number and clicks Sell Now , the system make receipt .
2. Conventional – *User* press the item list , select the item, system display item details , *user* click Basket , data entry to basket , *user* add number and clicks Sell Now , the system make receipt .

9. Viewing Activity Information

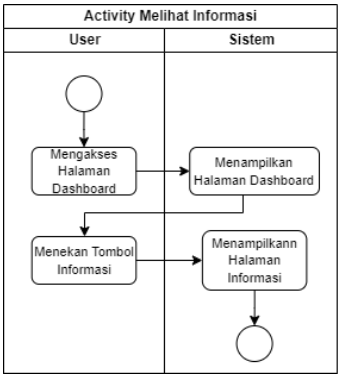

Figure 11 *Activity* See Information

Coding

At the stage this , implemented program coding . Stage results This is a form that becomes between advance system among others:

1. Page *Splash Screen.* Page This is appearance first at the time application executed
2. Page *Login.* The page containing the user login and password used . For user data verification .
3. Forget *Password.* Page This is facility If there is a user who forgot *password* then on the page This Can perform a password reset.
4. Page *Register.* Page This allows users new For make account that will used to log in to in application .
5. Page Home . Page home page is page the initial display that appears when the user successfully logs in.
6. Page Profile . Page This containing about profile of the currently logged in user .
7. Page Cart . Page This accessible For enter items to be issued from warehouse .
8. Page History . Page This is facility For see history transactions that happened . On the page This For see history transaction can filtered based on date .
9. Page History Details . Page This is page For open details each transactions that occur .
10. Page Information . Page This is feature complement in application .
11. Page List of Items . Page This contains a list of items that have been input to in table .
12. Page Plus Items . Page This accessed by admin to can add master data for goods .
13. Page Item Details . Page This is page that displays item data in detail per item .
14. Page *Update* Goods . Halaam This is page containing facility For edit existing item data Once saved .

Testing

At the stage This testing process is carried out to the system that has been built For ensure that all over functionality walk in accordance with needs and designs that have been designed . Testing done with use two approach , namely *white box* testing and *black box testing* . *White box testing* used For test internal logic and program flow based on structure code , whereas *black box testing* used For test function system from side users without see structure internally . Through combination second method this , it is hoped system can running optimally, freely from error logic , and provide appropriate experience for users end

Conclusion

Based on the results of the research and the implementation of the system, it can be concluded that the development of a shop-based Android application for Store Ferry Electronic using the Extreme Programming (XP) method effectively addresses various operational

challenges. Traditional manual recording processes, slow identification of goods, and limitations in stock reporting were successfully overcome through the implementation of this application (Isnanto, Alkodri, & Supardi, 2020; Maulidah et al., 2023). The application was designed with core features including goods recording, QR Code scanning, transaction management, electronic receipt generation, shopping baskets, and sales history reporting (Rahman et al., 2022; Permana & Utami, 2023).

The adoption of XP methods supported the development process by enabling iterative cycles, frequent feedback, and adaptive responses to the shop's specific operational needs (Abrahamsson & Koskela, 2004; Awad, 2017). Pair programming, continuous integration, and regular testing facilitated the rapid delivery of functional modules while maintaining high software quality (Bhimanapati, Jain, & Aggarwal, 2023; Serrador & Pinto, 2015).

As a result, the application not only increases staff work efficiency but also provides shop owners with real-time, accurate, and structured monitoring of inventory and sales data (Hidayat, Sakethi, & Ardiansyah, 2022; Mahesa, Hayati, & Mardiani, 2024). Stakeholder engagement throughout the development cycle ensured that the system aligns with user expectations and operational requirements (Chow & Cao, 2008; Conforto & Amaral, 2020).

In conclusion, the implementation of XP in developing this Android-based shop management application demonstrates that agile methodologies can enhance software performance, improve operational efficiency, and support business decision-making effectively. The system represents a practical solution for modern retail management in dynamic environments.

Recommendation

In order for the system can develop more Good to front , it is recommended For add feature notification minimum stock as warning for owner shop For quick do *restock* . Besides that , development application version *mobile* based *multiplatform* also possible become step strategic so that users *non-Android* can participate access application . Party shops are also recommended For in a way periodically do evaluation and maintenance system use guard performance application remain optimal. Training use system for staff shops are also important done so that every feature in application can utilized in a way maximum in accordance objective beginning development .

Bibliography

1. Abrahamsson, P., & Koskela, J. (2004). *Extreme programming: A survey of empirical data from a controlled case study*. Proceedings of ISESE 2004. https://doi.org/10.1109/ISESE.2004.1334895
2. Abrahamsson, P., et al. (2009). *Challenges of agile software development in global projects*. IEEE.
3. Armiady, D. (2020). Pengembangan aplikasi responsive website menggunakan Extreme Programming. *Jurnal Tika*, 7(2). https://doi.org/10.51179/tika.v7i2.1266
4. Akbar, IS, & Haryanti , T. (2023). Development of Entity Relationship Diagram Database for Ira Surabaya Online Store . *Computing Insight: Journal of Computer Science* , *3* (2), 28–35. https://doi.org/10.30651/comp_insight.v3i2.12002 .
5. Awad, M. A. (2017). Agile practices in software engineering: A systematic review of the literature. *International Journal of Computer Applications*, 165(8), 20–23. https://doi.org/10.5120/ijca2017914186
6. Dingsøyr, T., Prieto, E., & Karlsen, J. R. (2018). Team performance in agile software development: A systematic literature review. *Information and Software Technology*, 106, 49–65. https://doi.org/10.1016/j.infsof.2018.04.006

7. Bhimanapati, V. B. R., Jain, S., & Aggarwal, A. (2023). Agile methodologies in mobile app development. *Universal Research Reports*, 11(4). https://doi.org/10.36676/urr.v11.i4.1350
8. Capiluppi, A., Fernandez-Ramil, J., Higman, J. C., Sharp, H. C., & Smith, N. (2007). An empirical study of the evolution of agile-developed systems. *ICSE 2007*. https://doi.org/10.1109/ICSE.2007.14
9. Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961–971. https://doi.org/10.1016/j.jss.2007.08.020
10. Conforto, E., & Amaral, D. C. (2020). Agile and XP in SMEs. *Journal of Business Research*. https://doi.org/10.1016/j.jbusres.2020.05.016
11. Hidayat, M. A., Sakethi, D., & Ardiansyah, A. (2022). Pengembangan aplikasi Android One Day One Juz menggunakan XP. *Jurnal Pepadun*, 3(2). https://doi.org/10.23960/pepadun.v3i2.118
12. Isnanto, B., Alkodri, A. A., & Supardi, S. (2020). Penerapan Extreme Programming untuk merancang aplikasi kemahasiswaan berbasis Android. *Jurnal Sisfokom*, 9(2). https://doi.org/10.32736/sisfokom.v9i2.952
13. Mahesa, A. F., Hayati, N., & Mardiani, E. (2024). Implementasi Agile Extreme Programming pada sistem informasi laboratorium riset zoologi. *CICES*, 10(2), 155–167. https://doi.org/10.33050/cices.v10i2.3226
14. Maulidah, N., Diantika, S., Nalatissifa, H., Fauzi, A., & Supriyadi, R. (2023). Penerapan Extreme Programming dan Flutter dalam mobile application "Jago Masak". *JEKIN*, 4(2). https://doi.org/10.58794/jekin.v4i2.718
15. Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, 82(11), 1869–1890. https://doi.org/10.1016/j.jss.2009.05.052
16. Serrador, P., & Pinto, J. K. (2015). Does Agile work? A quantitative analysis of agile project success. *International Journal of Project Management*, 33(5), 1040–1051. https://doi.org/10.1016/j.ijproman.2015.01.006
17. Mounir, S. E. A. (2023). Software engineering for mobile applications: A survey on challenges and solutions. https://doi.org/10.48550/arXiv.2301.00602
18. Permana, F., & Utami, L. L. (2023). Pengembangan aplikasi e-arsip dengan metode Extreme Programming. *JITET*, 13(2). https://doi.org/10.23960/jitet.v13i2.6444
19. Rahman, M. F. F., Darussalam, K., Saphira, R. C., & Purwani, F. (2022). Implementasi Extreme Programming dalam pengembangan aplikasi mobile pengenalan organisasi. *Just IT*, 14(2), 128–132. https://doi.org/10.24853/justit.14.2.128-132
20. Rokhim, A., Azhar, F. I., & Saputra, W. E. (2024). Implementasi XP untuk meningkatkan pengembangan perangkat lunak pada UMKM. *Kohesi*, 6(1). https://doi.org/10.3785/kohesi.v6i1.9287
21. Septiani, N. A., & Julvian, A. (2022). Pengembangan aplikasi jadwal petugas tol dengan XP berbasis Android. *Jurnal Informatika*, 8(3). https://doi.org/10.32493/informatika.v8i3.34422
22. Suhadi, U., Hariyanto, D., & Sastra, R. (2021). Implementasi metode Agile Extreme Programming pada sistem informasi pendaftaran laboratorium. *IMTechno*, 4(2). https://doi.org/10.31294/imtechno.v4i2.2019
23. Supriyatna, A. (2018). Metode Extreme Programming pada pembangunan web aplikasi seleksi peserta pelatihan kerja. *Jurnal Teknik Informatika*, 11(1). https://doi.org/10.15408/jti.v11i1.6628
24. Tetteh, S. G. (2024). Empirical study of agile software development methodologies. *Asian Journal of Research in Computer Science*, 17(5), 30-42. https://doi.org/10.9734/ajrcos/2024/v17i5436

25. Yusuf Bakhtiar, M., Suwela, N., & Ismaillah. (2024). Pemanfaatan XP dalam pengembangan e-learning. *JSITP*, 5(1). https://doi.org/10.58436/jsitp.v5i1.1819.
26. VersionOne. (2020). *State of Agile Report*. https://doi.org/10.13140/RG.2.2.16819.12327