



Modern Lifestyle Transformation: The Role of an Agile Android Diary App for Time Management and Self-Reflection

Andi Abdiyah Kurnia Dela^{1*}, Nurhaliza², Winda³

^{1,2,3} Universitas Negeri Makassar, Jl. Malengkeri Raya, Parangtambung, Kec. Tamalate, Kota Makassar, Sulawesi Selatan 90224, Indonesia

Corresponding e-mail : andiabdyah16@gmail.com

ARTICLE INFO	ABSTRACT
Keywords: Android Java Local Storage SharedPreferences Short Notes	The rapid development of information technology drives the need for simple yet functional mobile applications to support daily productivity. One solution that is increasingly in demand is the short notes application, which allows users to record ideas, tasks, or important information quickly and practically. This study aims to design and develop an Android-based shortnotes application that prioritizes ease of use, efficiency, and local data storage without requiring an internet connection or integration with an external database. The system development method is carried out through several stages, namely planning, needs analysis, interface design, code implementation, and application testing. The Java programming language is used in the coding process with Android Studio as the main development environment. This application utilizes SharedPreferences and local file storage as data storage media, which are chosen to maintain the simplicity and efficiency of application performance on the user's device. The main features of the application include creating new notes, editing, deleting, and checking notes based on the results produced. The test results using the emulator show that the application runs smoothly and responsively in various usage scenarios. With a minimalist and intuitive interface, this application is expected to help users manage their daily notes more effectively.
Article History Received: Januari 02, 2024 Revised: Februari 20, 2024 Accepted: April 6, 2024	

This is an open access article under the [CC BY-SA](#) license



To cite this article : Author. (20xx). Title. Information Technology Education Journal, X(X), XX-XX. Doi. xxxx

INTRODUCTION

The development of information and communication technology in the digital era has had a major impact on people's lifestyles. Modern lifestyles characterized by high dynamics, time constraints, and productivity demands have created significant psychological and emotional stress. Individuals are required to stay connected, adapt quickly, and manage multiple responsibilities simultaneously. This condition makes time management and emotional balance a crucial aspect in maintaining quality of life [1]. In response to these challenges, technological innovations present various digital solutions designed to help individuals manage their daily lives more effectively [2]. One of them is a digital diary app that not only functions as a recording medium, but also as a tool for time management and self-reflection. This app is equipped with various features such as scheduling, habit monitoring, reflective journaling, and data grouping that make it easier for users to plan, evaluate, and organize their activities in a structured manner [3].

Although digital diary apps are widely used in everyday life, academic studies on their effectiveness in supporting time management and self-reflection amidst the complexity of modern lifestyles are limited [4]. Most studies have emphasized the technical aspects of app development or their use in the context of work productivity, but few have specifically explored

how these apps contribute to the balance between professional demands and users' psychological well-being [5]. Therefore, this study aims to examine the role of digital diary apps in helping individuals manage time and self-reflect as part of a strategy to deal with the demands of a modern lifestyle [6]. This research is expected to contribute to the development of literature related to the use of digital technology to support the quality of life and emotional balance of individuals in a fast-paced and stressful era [7].

METHOD

In this research, the application development approach applied is the Agile method. Agile is an iterative and adaptive approach to software development, which emphasizes cross-functional team collaboration, adaptability to change, and continuous development and responsiveness to user needs. This approach is executed through short cycles called sprints, where each sprint provides an opportunity for the team to evaluate progress and make adjustments to project needs [10]. These adjustments are then outlined in the sprint backlog. In the context of this research, three sprints have been designed that will be implemented during the development process [11]. This method was chosen because it is in line with the principles of the Software Development Life Cycle (SDLC), which includes the main stages such as planning, needs analysis, system design, implementation, and testing, Agile is the main choice because this approach is proven to be able to increase productivity, accelerate the product release process with more stable features, and provide high flexibility in the face of change, especially in the development of dynamic and competitive applications, such as mobile applications. The cycle of the Agile method can be seen in Figure 1

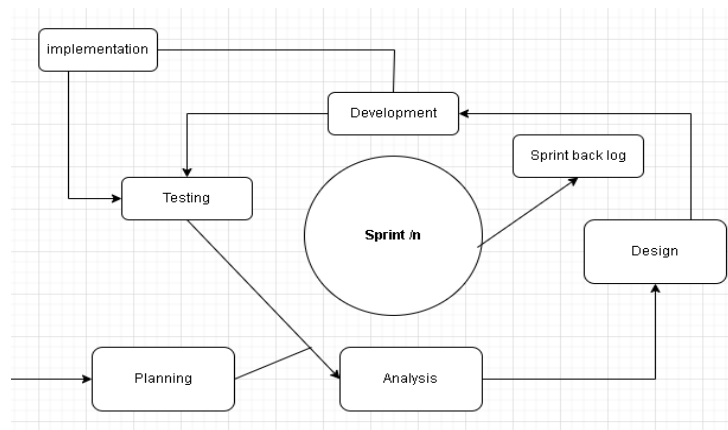


Figure 1. Agile methodology

System Planning

In the early stages of software development with the Agi method, the process begins with system planning. Two important components in this phase are the system request and the feasibility study. The system request includes identifying the needs of the user or client collected through observation, interviews, or questionnaires. The collected information is then analyzed through a feasibility study, which includes three main aspects: economic feasibility (whether the project is financially feasible), technical feasibility (whether the system can be built with available technology), and organizational feasibility (whether the system is aligned with organizational goals) [12].

System Analysis

After the planning process, proceed to the analysis phase which aims to explore system requirements in detail. In this phase, identification of functional requirements, namely services

or features that the system must provide, and non-functional requirements, such as security, reliability, and user comfort [13]. This analysis is usually reinforced by modeling using the Unified Modeling Language (UML), such as use case diagrams to describe the user's role in the system and activity diagrams to show the system's workflow. Other diagrams will be prepared according to the needs of the system that develops in the next design stage.

System Design

The design stage is an advanced step that aims to translate the results of the analysis into a form of technical system design. This process begins with designing the user interface to provide a visual description of the system display. After that, the system structure is designed through class diagrams and sequence diagrams that describe the relationship between objects and the sequence of interactions that occur in the system. This design becomes an important guideline in the implementation process [14].

System Implementation

The implementation stage is the process of realizing the entire system planning and design into the form of real applications. This implementation consists of two main parts: 4. Coding In this phase, the design that has been made will be developed into software through the programming process. Coding includes the development of the backend side, which handles business logic and data processing, as well as the frontend side, which plays a role in presenting visual displays to users. The choice of programming language and framework is tailored to the needs of the project.

System Testing

After the coding process is complete, testing is carried out to check whether the application actually performs all the expected tasks as designed in the analysis and design document. Testing is carried out using the Black Box and White Box Testing methods, which are methods that assess the functionality of the system from the end user's point of view without checking the internal structure of the program code and knowing the internal structure and logic of the program code. At this stage, testing is focused on user interaction with the system interface, such as the system accepts input properly, displays appropriate results, and functions in normal and extreme situations (e.g. empty input, very long data, lost connection). A total of five test subjects were tested as representative end-users, who tried to use the application like regular users. Each person was asked to run a number of test scenarios to assess the app's ease of use, system accuracy, and stability of functionality.

RESULTS AND DISCUSSION

The result of the development is a web-based mobile application, which will be used by users consisting of respondents and researchers. The development of this short notes application uses the Agile development method which has 4 stages, namely: system planning, needs analysis, design and finally implementation which consists of coding and testing. The presentation of the results of the Agile development stages is as follows in Table 1.

Table 1 Feasibility study

Feasibility Study of Digital Diary Application System "ShortNotes"
A feasibility study assessment was conducted by Nurhaliza for the development of an Incentive Survey (Re-Quest) system.
Technical feasibility study

The "ShortNotes" application is considered technically adequate for development, with some challenges remaining within manageable limits:

Risks Related to Application Mastery: Medium Risk

- Development Team: Still learning in the experience of developing Flutter-based mobile apps, but not yet specialized in cloud-based daily note taking systems with high privacy security features.
- Content and Testing Team: Need to learn more about the user journey in the context of a diary app in order to conduct optimal testing.

Risks Related to Technology Used: Low Risk

- Support and Operations Team: Requires basic technical training to understand the maintenance process and periodic updates on the mobile application connected to the server.

Risks Based on Project Scale: Low Risk With a team of 3 core developers and 1 UI/UX designer, this application is projected to be developed within 3-4 months. Integration with External Systems: Medium Risk Need integration with cloud storage and external notification systems. Challenges lie in cross-device compatibility (Android and iOS) and offline-to-online data management.

Economic feasibility study

Although the digital diary app "ShortNotes" is still in the development stage and has not yet been publicly released, a preliminary economic feasibility analysis has been conducted to evaluate the financial potential and sustainability of the app in the medium to long term. Based on simulations of revenue and cost projections, the development of this app shows promising economic potential, especially with the freemium business model approach and ad integration.

- Revenue Model:
The app is designed to combine revenue from in-app advertising and premium subscriptions that offer additional features such as cloud storage, note encryption, exclusive themes and multi-device synchronization.
- Estimated Initial Investment:
The initial development cost (including UI/UX design, backend and frontend development, and testing) is estimated to be around IDR 45,000,000.
- Return on Investment (ROI):
Assuming gradual user growth and a premium conversion rate of around 5-10%, the ROI is estimated to reach 120% within 3 years. This means that the net profit can exceed the initial capital of Rp 54,000,000 within that timeframe.
- Break Even Point (BEP):
The break-even point is projected to be reached in about 15-18 months after the official launch, depending on the speed of user acquisition and the effectiveness of the digital marketing campaign.
Three-Year Profit Projection:
If the monetization strategy goes according to plan, the app has the potential to generate a net profit of Rp 130,000,000 - Rp 150,000,000 in three years, assuming stable user retention and engagement rates.

Overall, from an economic standpoint, the "ShortNotes" app has the potential to grow financially and become a sustainable digital productivity solution. Economic risks remain, mainly related to user acquisition and initial promotion costs, but are still within manageable levels with proper planning.

Organizational feasibility study

Organizationally, the development of the "ShortNotes" application is considered feasible, with relatively low risk. The development team consists of individuals with appropriate backgrounds and the internal organizational support is strong enough to support the continuity of the project.

The main objective of the app development is to provide a secure, private, digital note-taking tool that is easy to use and understand by general users.

Comparison of ShortNotes App Contributions with Related Research

The development of the ShortNotes app as a secure and private digital diary platform makes its own contribution to the friendly technology of personal productivity tools. To strengthen the justification of the feasibility of its development, the following is a comparison of the contribution of this application with six similar research or projects that have been carried out previously:

- 1) Ramadhani (2021) - Design and Development of Android-based Notes Application The main focus is on developing a simple application for taking notes on general and

personal matters on Android devices. There is no cloud integration or data encryption. Comparison: ShortNotes is much more sophisticated as it not only provides basic note-taking features, but also offers user privacy through encryption as well as monetization through a freemium model.

- 2) Artika (2021) - Android-based Daily Notes Application Design This application emphasizes the ease of users in making daily notes and managing important information. There are note input and user profile features, but no security aspects or business model are described. Comparison: ShortNotes is superior with security integration and the potential for a freemium-based economy that allows for app sustainability.
- 3) Damayanti & Yulianingsih (2024) - Android-Based Daily Note Application Design with Encryption This research focuses on note security through cryptography. Although this application has an encryption feature, the development and usage model is still limited to individual access with a decryption key. Comparison: ShortNotes offers a similar concept in premium features, but adds value through multi-device synchronization, monetization, and technically and economically complete risk analysis.
- 4) Tri Wahyudi (2022) - Development of Web-Based and Android Applications as Work Support in Indonesia: Systematic Literature Review This study reviews the trend of Android application development in Indonesia, especially in the education and public service sectors. However, it does not discuss in detail personal record applications. Comparison: ShortNotes targets the unexplored personal productivity segment in this study, revealing a new market opportunity that has yet to be maximized.
- 5) Fahri Rivaldi (2016) - Mobile Application Design "KAMUSKU" Although not a note application, this application provides ease of learning languages with an offline approach and simple interface. There is no cloud feature or security system. Comparison: ShortNotes has a more complex technology and user experience approach, with cross-platform support and user data protection as key values.
- 6) Rangga Desta (2020) - Learning to Create Simple Web Services: Daily Journal Application with PHP This article demonstrates the creation of a web-based daily journal application using PHP and MySQL with RESTful API. The main focus is on backend processing (CRUD: create, read, update, delete), as well as connecting to Android apps using web services. While it was able to demonstrate the basic functionality of logging and API integration, security aspects such as user authentication and data encryption were not covered in depth. Comparison: ShortNotes goes a step further with its encryption-based security approach for user data, cloud integration, and freemium business model. In addition, ShortNotes is developed with Flutter that supports cross-platform (Android and iOS), whereas Rangga Desta's study is limited to a simple web service architecture for Android.

System Planning Results

The planning stage begins with observation and in-depth interviews with clients to collect system requirements in the form of system request documentation. The results of this system request show the need for a platform that is able to manage short or daily notes efficiently, with easy recording, editing, deleting features. Furthermore, the results of the identification of needs in the form of a system request will be analyzed using a feasibility study with the aim of knowing whether the system to be developed can be realized in accordance with the available resources. The feasibility study also helps in determining the potential success of the project, estimating

risks, and preparing the project team with a better understanding of the challenges that may be faced.

Analysis Results

At the analysis stage, the functional and non-functional requirements of the application were defined in detail. Functional requirements include key features such as survey creation, incentive distribution, and user management, while non-functional requirements relate to data security, application performance, and usability, details of which can be seen in table 2 and table 3.

Table 2 Non-functional Requirement

Non-Functional Requirement		
ID	Parameter	Explanation
NFR-01	Availability	The application can be used at any time because it does not use the internet
NFR-02	Reliability	The system must be able to run stably without frequent crashes or errors.
NFR-03	Ergonomy	The user interface should be simple and easy to use for all.
NFR-04	Portability	The app must be compatible with various devices, including mobile phones
NFR-05	Memory	Memory usage must remain efficient so that the application can run on low-specification devices.
NFR-06	Response time	Navbar menu switching is less than 2 seconds
NFR-07	Safety	N/A
NFR-008	Security	User data should be protected with encryption and only accessible by its owner

Table 3 Functional Requirements

Functional Requirement		
ID	Fitur	Explanation
FR-01	Note-taking	Users can create new notes with titles and contents that can be stored in the system.
FR-02	Record Deletion	Users can edit previously created notes.
FR-03	Dark Mode	Users can manually enable or disable dark mode through the app.
FR-04	Check	Users can mark or check off saved notes and read notes.
FR-05	Sign UP	Users can create an account first before entering the notes application.
FR-06	Log In	Users can enter 1 note app with an email that has been created in advance.

In addition to the identification of functional and non-functional requirements, the analysis phase generally also involves visual representations using UML (Unified Modeling Language) diagrams. These representations aim to provide a more structured picture of the relationship between the system and its users. One diagram that is often used in this stage is the use case

diagram, which describes various scenarios of interaction between actors (users) and the system based on the main functions provided. The following Use case diagram in Figure 2.

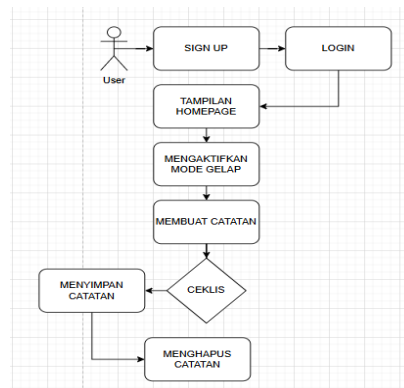


Figure 2. Use Case Diagram

- Use case Sign Up: Allows users to create an account first or register their username if they do not have an account, before logging in.
- Login Use case: Allows the user to log in and enter the username and password that has been created previously.
- Dark mode use case: Allows users to change the appearance of the application according to comfort.
- Use case Take notes: Allows the user to make notes according to what will be made.
- Use case Checklist: Allows users to mark and check notes.
- Use case Save: Allows the user to save the note after creating the note.
- Use case Delete: Allows the user to delete a note if the previous note is wrong or does not match the desired one.

Activity diagram is a type of diagram in the Unified Modeling Language (UML) that is used to model the workflow or activities in a system process. This diagram illustrates the sequence of activities, decisions, and data flow from one step to the next, both in business processes and software system flow. The following Activity diagram in Figure 3.

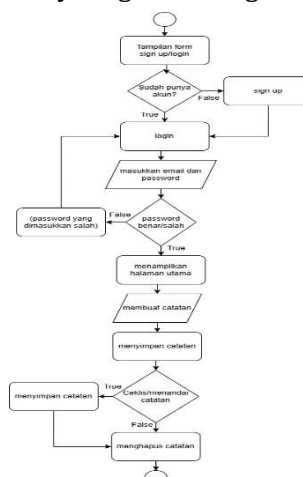


Figure 3. Activity diagram

The process starts with the login or signup form. The user will first be asked whether they already have an account or not. If not, the user is directed to the signup process. If so, the user enters the login process by entering an email and password. After that, the system will verify the

password. If the password is incorrect, the user will be asked to repeat the login process. If it is correct, the user will be directed to the main page of the system. On the main page, the user can create a note, as well as save or delete the note. The process is terminated once the saving or deleting action is completed.

Design

The design stage in this research produces visual designs in the form of Class Diagrams, and Sequence Diagrams that have been adjusted based on relevant references and feature needs of an incentive-based mobile survey application called "Re-Quest". This design was developed to provide a clear picture of the system structure and interactions between components in the application.

Class Diagram Design

In the system design process, the Unified Modeling Language (UML) approach is used as a tool to describe the system with an object-oriented approach. UML makes it easy to visualize complex system components into models that are more structured and easy to analyze. One important part of this modeling is the class diagram, which describes the blueprint of the classes in the system, including attributes, methods, and relationships between interconnected classes. The design of the class diagram is shown in Figure 4.

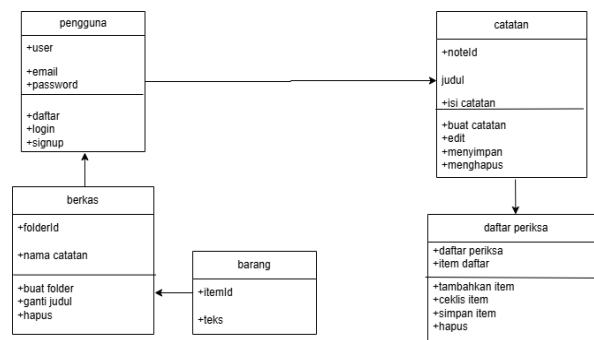


Figure 4. Class Diagram

Sequence diagrams are used to describe the interaction between objects over time for a particular process, providing a clear guide to the flow of execution of operations in the application based on each use case which in detail can be seen in Figure 5.

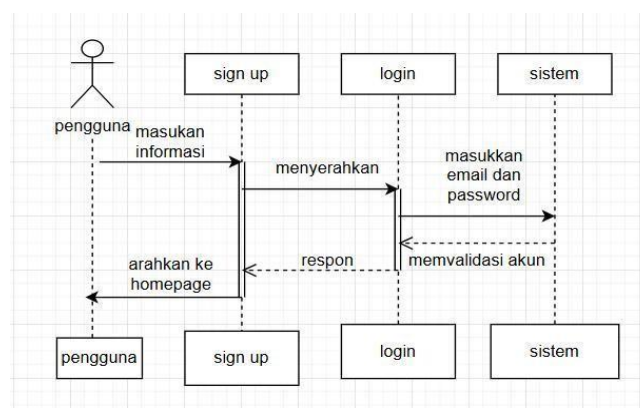


Figure 5. Sequence Diagram

User Interface Design (UI)

Furthermore, there is a UI design that is made based on the functionality that has been defined above using the Figma design application based on a style guide to maintain design consistency and make it easier for developers to create the application. The UI design can be seen in Figure 6 of the following design:

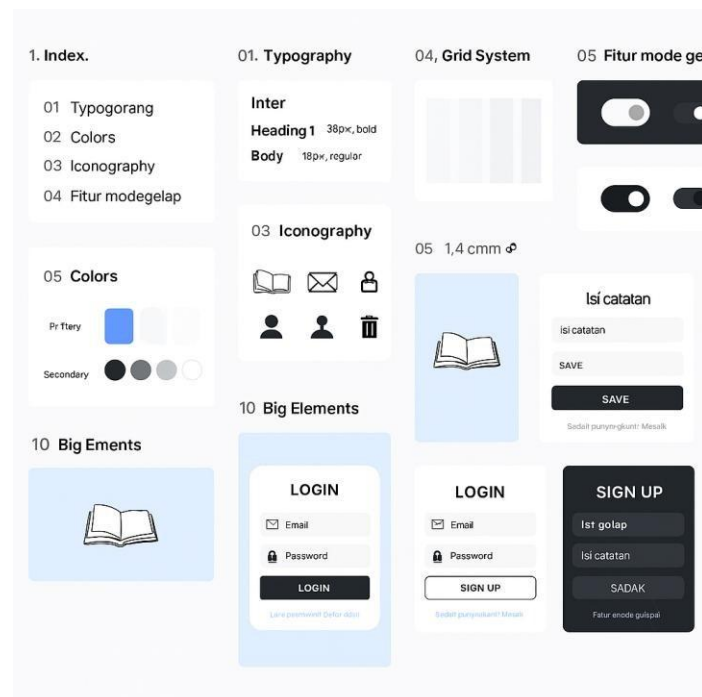


Figure 6. Style guide

First sprint

In the first sprint, authentication features were developed in the form of login and sign up pages. The login page includes email and password input, while the sign up page provides a new account registration form. After evaluating the initial design, a forgot password feature was added to help users access their account if they forgot their password. In addition, a note page was also developed as the main post-login page, which includes a note writing area, save button, dark mode option, as well as navigation icons at the bottom for ease of use Figure 7.

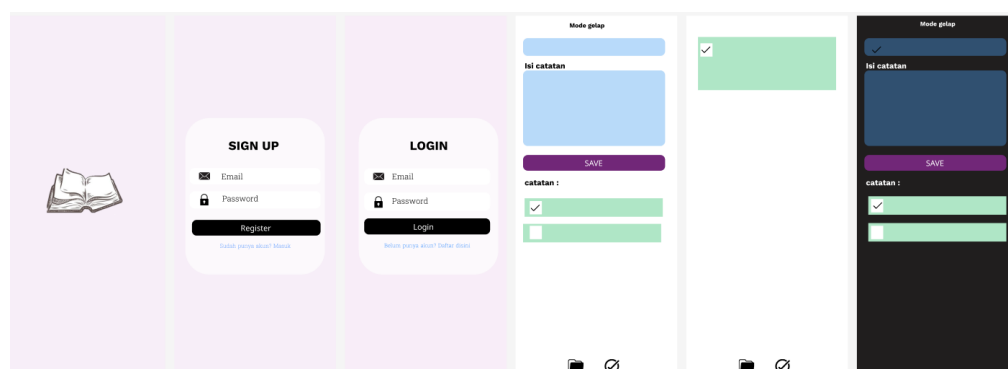


Figure 7: Design of Login, Signup and home page features

Second sprint

In the second sprint, the team focused on implementing user authentication and daily logging features. Sign Up and Login pages were developed for the process of registering and logging into the application. After logging in, users were redirected to the main note page, where they could write the note title and content, save it, as well as view the list of saved notes. Additional features include a dark mode for visual comfort, a tick function to mark completed notes, as well as a delete note feature with confirmation. Navigation is made easy through icons such as "Folder" and "Notes", with a simple and streamlined UI design in Figure 8.

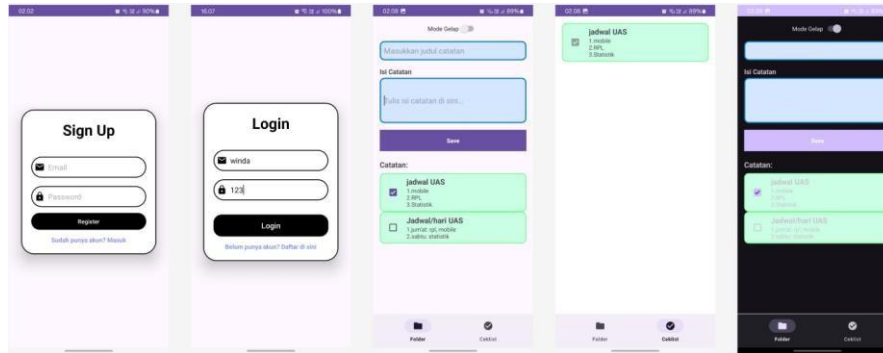


Figure 8. Design of signup, login, main page, and how to add notes

Implementation Results

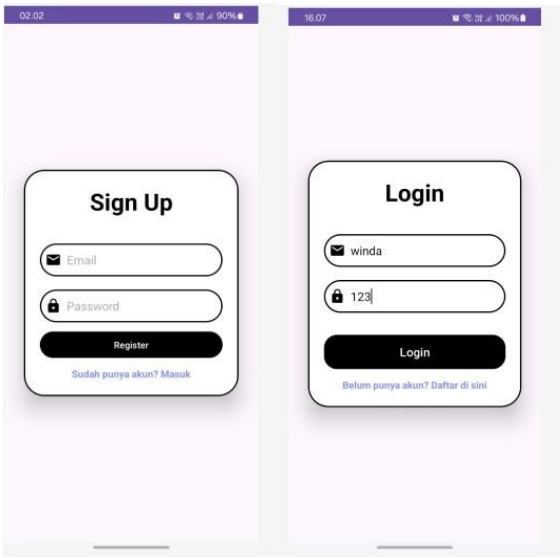
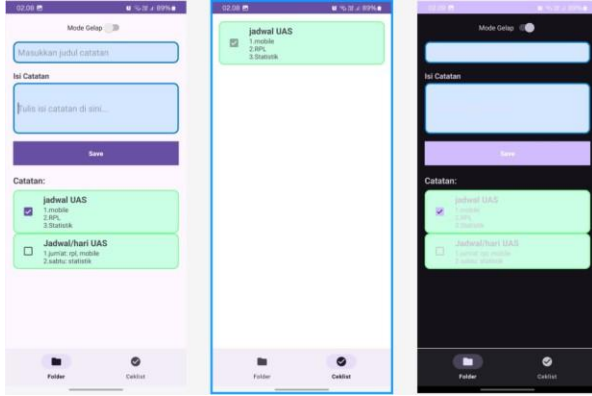
Based on the results of the analysis and design above, several changes have been obtained which have become 2 sprints and have been implemented in coding and tested so as to produce a ShortNotes application which more details can be seen in the following 2 sections:

a. Coding Result

The result of the coding stage in this research includes the complete code that implements the ShortNotes mobile application. During the coding process, the researcher used Android Studio as the primary development environment due to its extensive support for the Java programming language, which was used as the primary language in the development of this application. Android Studio provides a streamlined interface as well as various tools that ease the coding and debugging process. This application is designed to store records online by using Firebase as an external database system. Firebase Realtime Database or Cloud Firestore is used to centrally and synchronously store, edit, and delete user notes. This approach allows data to be stored securely in the cloud, accessible from various devices, and supports real-time data updates. The use of Firebase also provides flexibility in user authentication and easy integration with various other Google services.

In order to test the functionality of the code, an emulator from Android Studio was used to simulate the behavior of the application on a mobile device. The use of the emulator allowed the researcher to test various usage scenarios, including online note-taking, editing, and deleting, to ensure that all features were working properly and in sync before the app was officially released. all aspects of the app worked as expected before launching to the market, the results of which can be seen in the figure in Table 4 below:

Table 4 Coding results

Coding	Result Display
First Sprint Sing Up, Login <pre> 1 <?xml version="1.0" encoding="utf-8"?> 2 <LinearLayout 3 xmlns:android="http://schemas.android.com/apk/res/android" 4 xmlns:app="http://schemas.android.com/apk/res-auto" 5 xmlns:tools="http://schemas.android.com/tools" 6 android:layout_width="match_parent" 7 android:layout_height="match_parent" 8 android:orientation="vertical" 9 android:gravity="center" 10 tools:context=".MainActivity"> 11 12 <androidx.cardview.widget.CardView 13 android:layout_width="match_parent" 14 android:layout_height="wrap_content" 15 android:layout_margin="38dp" 16 app:cardCornerRadius="38dp" 17 app:cardElevation="20dp"> 18 19 <LinearLayout 20 android:layout_width="match_parent" 21 android:layout_height="wrap_content" 22 android:orientation="vertical" 23 android:gravity="center_horizontal" 24 android:padding="24dp" 25 android:background="@drawable/custom_edittext"> </pre>	
Second Sprint Homepage <pre> 1 <?xml version="1.0" encoding="utf-8"?> 2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" 3 xmlns:app="http://schemas.android.com/apk/res-auto" 4 xmlns:tools="http://schemas.android.com/tools" 5 android:layout_width="match_parent" 6 android:layout_height="match_parent" 7 tools:context=".MainActivity"> 8 9 <!-- Framelayout untuk menampilkan fragment --> 10 <FrameLayout 11 android:id="@+id/frameLayout" 12 android:layout_width="match_parent" 13 android:layout_height="match_parent" 14 android:layout_above="@id/bottomBarContainer"> 15 16 <ScrollView 17 android:layout_width="match_parent" 18 android:layout_height="wrap_content" 19 android:fillViewport="true"> 20 21 <LinearLayout 22 android:layout_width="match_parent" 23 android:layout_height="wrap_content" 24 android:orientation="vertical" 25 android:padding="16dp"> </pre>	

b. Testing Results

Application testing is done through Black Box and White Box testing which involves end users testing the functionality of the application without knowing the code behind it. This helps in identifying problems from the user's perspective with test cases that can be seen in Table 5 and Table 6 below:

Table 5 Black Box test results

Pengujian Black Box				
Kode Uji	Kasus Uji	Hasil yang Diharapkan	Hasil yang Diperoleh	Status
UB-1	Login dengan username dan password kosong	Pesan error "Username dan password tidak boleh kosong" ditampilkan	Pesan error "Username dan password tidak boleh kosong" tampil	Berhasil
UB-2	Login dengan username dan password yang salah	Pesan error "Login Gagal" ditampilkan	Pesan error "Login Gagal" tampil	Berhasil
UB-3	Login dengan username dan password yang benar	Berhasil login dan pengalihan ke homepage	Berhasil login dan mengalihkan ke homepage	Berhasil
UB-4	Halaman utama	Berhasil masuk ke homepage	Berhasil mengalihkan ke homepage	Berhasil
UB-5	Menyimpan catatan	Berhasil disimpan	Berhasil menyimpan teks yang sudah dicatat	Berhasil
UB-6	Menghapus catatan	Berhasil dihapus	Berhasil menghapus catatan yang tidak dibutuhkan	Berhasil
UB-7	Ceklis catatan	Berhasil diceklis	Berhasil menceklis catatan yang telah dibuat	Berhasil

Table 6 White Box test results

White Box Testing					
Test Code	Test Case	Testing Steps (White Box)	Expected Results	Results Obtained	Status
UW-1	Log in with your username and password	1. Check the input validation function 2. Check the backend validation function call	Validation and authentication functions work logically	Function works according to logic, login successful/failed	Successful
UW-2	Login with empty data	1. Check the if condition for empty input 2. Make sure the login function is not called	Blank input rejected, no further login process	Validation prevents login when input is empty	Successful
UW-3	Keep a record	1. trace the flow of the save function from the "save" button to the save function in the database/file	The record is saved to storage and appears in the list	Note successfully saved and displayed	Successful

Based on the development results, this application has gone through two sprint stages, each of which resulted in key features that have been tested using the blackbox testing method. In the first sprint, the team focused on developing user authentication features in the form of login and sign up pages [15]. The login page uses email and password as the main parameters, and was tested with several scenarios such as blank input, incorrect input, and correct input, all of which gave the expected results-where an error message is displayed if the data is invalid, and the login

is successful when the data is correct. Meanwhile, the sign up page allows new users to create an account by filling in a username and password.

Still in the same sprint, the main page of the note was also developed, which became the center of activity after the user successfully logged in. On this page, users can write the title and content of the note through the two input forms available, then save it by pressing the "Save" button. The saved note will immediately appear in the list of notes below it. In addition, users can enable dark mode for visual comfort, especially at night, which turns the background dark and adjusts other UI elements. Also, users can mark by ticking a previously created note to ensure it has been done or completed.

In the second sprint, key functional features such as delete note, which brings up a confirmation dialog to prevent accidental deletion of notes, were further developed. In-app navigation was also strengthened by the presence of icons at the bottom of the screen such as "Folder" and "Notes", which make it easier for users to move between features. Overall, the UI design is focused on simplicity and efficiency, making the app convenient to use in the user's daily life. On the dark-colored homepage, there is a button to change the homepage display to black, this can be set at any time according to user needs and convenience.

CONCLUSIONS

This research successfully implemented an Agile method-based shortnotes application, which allows users to take notes conveniently and efficiently according to their needs and preferences. With an intuitive interface and personalization features, the app is expected to increase users' productivity and engagement in taking notes of important information in daily life as well as in academic contexts. The app not only speeds up the note-taking process but also enriches the user experience with features that support flexible and organized note management. Thus, the use of this application can provide significant benefits, especially for students in recording lecture material or important ideas in a practical way. As a suggestion, it is recommended to conduct a longitudinal study to monitor the long-term impact of using the app on user satisfaction, note-taking quality, and the influence of personalization features on users' note-taking habits. Further research should also include the development of additional features such as note collaboration, cloud storage integration, and cross-device synchronization support, which can improve users' convenience and efficiency in accessing and sharing their notes.

REFERENCES

- [1] M. S. Ummah, "Sustain., vol. 11, no. 1, pp. 1-14, 2019, [Online]. Available: http://scioteca.caf.com/bitstream/handle/123456789/1091/RED2017-Eng-8ene.pdf?sequence=12&isAllowed=y%0Ahttp://dx.doi.org/10.1016/j.regsciurbeco.2008.06.005%0Ahttps://www.researchgate.net/publication/305320484_SISTEM_PEMBETUNGAN_TERPUSAT_STRATEGI_MELESTARI
- [2] N. R. Damayanti and E. Yulianingsih, "Design Aplikasi Catatan Daily Berbasis Android Menggunakan Metode Waterfall," vol. 5, no. 1, pp. 52-60, 2024, doi: 10.51519/journalita.v5i1.469.
- [3] R. A. Setyawan, "Penerapan Firebase Realtime Database Pada Aplikasi Catatan Harian Diabetes Melitus," *J. Inform. Komputer, Bisnis dan Manaj.*, vol. 22, no. 1, pp. 1-9, 2024, doi: 10.61805/fahma.v22i1.102.

- [4] A. A. Ramadhani and U. P. Raya, "Perancangan Dan Pembuatan Aplikasi Notes Berbasis Android," no. July, pp. 0–7, 2021.
- [5] S. D. Artika, "RANCANG BANGUN APLIKASI DAILY NOTES BERBASIS ANDROID DI ERA RANCANG BANGUN APLIKASI DAILY NOTES BERBASIS," no. July, 2021.
- [6] O. O. Almando, "Rancang Bangun Aplikasi Catatan Keuangan Sederhana Berbasis Android," *Researchgate.Net*, no. July, pp. 1–6, 2021.
- [7] B. Prasetyo, E. N. Kusumaningrum, and I. Saraswaty, "CATATAN SINGKAT : POTENSI KOPI ROBUSTA (Coffea robusta Linden) SEBAGAI ANTIOKSIDAN DAN ANTIBAKTERIAL SHORT NOTES : THE POTENTIAL OF ROBUSTA COFFEE (Coffea robusta Linden) AS ANTIOXID ...," *J. Sains dan Teknol. Univ. Terbuka*, vol. 1, no. 1, pp. 1–25, 2023, [Online]. Available: <https://www.researchgate.net/publication/375596917%0ACATATAN>
- [8] M. Hamka and I. Purnama, "Rancang Bangun Aplikasi Catatan Ku Berbasis Android," *J. Gemilang Inform.*, vol. 2, no. 1, pp. 22–27, 2024, doi: 10.58369/git.v2i1.167.
- [9] Z. Trinova, W. Izati, and A. Suci, "Penerapan Strategi Catatan Terbimbing (Guided Note Taking) dalam Pembelajaran Fiqih," *J. Tarb. Al-Awlad*, vol. 7, no. 1, pp. 45–58, 2017.
- [10] Rofi'atul Mukaromah, "Pengembangan Materi Teks Catatan Harian Dalam Bentuk Buku Suplemen Berbasis Pendekatan Pengalaman," *J. Educ. Learn. Sci.*, vol. 2, no. 1, pp. 108–135, 2022, doi: 10.56404/jels.v2i1.20.
- [11] A. Sudarsyah, "KERANGKA ANALISIS DATA FENOMENOLOGI (Contoh Analisis Teks Sebuah Catatan Harian)," *J. Penelit. Pendidik.*, vol. 13, no. 1, 2016, doi: 10.17509/jpp.v13i1.3475.
- [12] G. J. B. Dwi, R. T. A. K. Putri, and A. Apriliani, "Pengelolaan keuangan pribadi berbasis aplikasi teknologi digital," 2021.
- [13] B. A. B. Iii and A. D. A. N. Implementasi, "Bab iii analisa dan implementasi 3.1," pp. 18–54, 2007.
- [14] L. N. Hakim, *Pembuatan Aplikasi Catatan Harian Berbasis Web Untuk Monitoring Siswa SMK Pelita Nusantara 2 Semarang Menggunakan PHP Dan MySQL*. 2020.
- [15] Mukrodin, "Implementation of the Waterfall Method in Building School Information Systems in Vocational Schools. S Al Habibata'in Bumiayu," *ZONasi: Journal of Information Systems*, vol. 2, no. 1, pp. 34–47, 2020, doi: 10.31849/zn.v2i1.4266.