

Application Development for Fasting and Worship Management Ramadan Using the Incremental Method: Kareem

Daffa Fakhir^{1*}, Nabila Al Buqari Jufri², Muh Akhlatul Ihsan³, Putri Nirmala⁴, Ummul Khaeri Masna⁵

^{1,2,3} Universitas Negeri Makassar, Indonesia

¹daffafakhir@gmail.com, ²nabiljupri0@gmail.com, ³muhakhlatulihسان@gmail.com, ⁴putranirmala2711@gmail.com, ⁵ummul.khaerimasna05@gmail.com

ITEJ Journal

Article History:

Submitted: 03-May-2025

Accepted: 01-June-2025

Published: 30-June-2025

Keywords:

Android App; Ramadan Worship; Incremental; Time management; Digital spirituality

*Corresponding Author:

Daffa Fakhir

ABSTRACT

The month of Ramadan is an important moment for Muslims to improve the quality of their worship and spirituality. However, in today's fast-paced modern era, managing worship time has become a challenge in itself. This study aims to develop a mobile application called KAREEM that supports Muslims in managing their worship activities during Ramadan, such as prayer reminders, recording the completion of the Qur'an, and community interaction. The development method used is an incremental model with a structured System Development Life Cycle (SDLC) approach. Development was carried out using Android Studio and Firebase for real-time data storage. System implementation includes key features such as login, profile, worship reminders, community, and Quran reading progress. Testing was conducted using black box and white box methods, showing that all features worked as expected and were free from logical errors. The results show that the KAREEM application can help users schedule and record worship activities effectively. Continuous development is recommended, such as improvements to the interface, two-factor security, and voice features for user inclusion. This application has great potential to become a useful digital worship companion solution during Ramadan and beyond.

INTRODUCTION

The month of Ramadan holds a special place for Muslims around the world as it is believed to be the time when the Quran was first revealed and a time to increase worship. Various religious activities, ranging from fasting, prayer, and zakat to religious social activities, are carried out with the aim of increasing faith and closeness to God [1], [2]. According to spiritual studies cited in Islamic literature, the worship practices performed during this month are designed to shape the spirit so that it always remembers the Creator and maintains closeness to Him [3]. Therefore, Ramadan is seen not only as a month of rituals, but also as a momentum for improving the spiritual and social quality of Muslims. However, amid the busyness of modern society, this worship practice often faces new challenges that require adaptive solutions.

In the modern era, people's increasingly busy lifestyles pose new challenges in managing time for worship. Fadillah emphasizes that the digital age has brought about consequences in the form of difficulties in maintaining religious values, but at the same time opens up opportunities to utilize technology to strengthen Islamic practices [4]. In line with this, Zahraini explained the importance of integrating spiritual values with a technological approach in contemporary Islamic curricula [5]. Furthermore, Mar highlighted the need for a technology integration strategy in Islamic education, both to counter the negative impact of social media and to utilize digital literacy to strengthen Islamic values [6]. This shows that digital technology has great potential to support the continuity of worship, especially during Ramadan. This potential is becoming increasingly apparent with the development of digital applications that are beginning to be used to support religious activities.

Digital applications are now not only present in the professional or social sphere, but are also developing to support various specific activities. For example, research on futsal reservation applications shows that Android-based systems can facilitate the booking and management of sports activities in a practical manner [7]. These findings show that mobile technology can meet the needs of society in various contexts, both in sports and religion. In line with this, other studies show that Android-based daily worship management applications help Muslims organize and monitor their worship activities more effectively [8]. In addition, another study found that the use of Android-based learning media can increase student motivation in learning about Islam and morality [9]. Research conducted by Cahyani developed an Android-

based application to guide the observance of Ramadan fasting, providing guidance on fasting, sunnah prayers, zakat, Eid al-Fitr, and prayers [10]. However, the study used the Agile method, which is a flexible and adaptive software development model [11]. Another study by Kurniawan also applied the Agile Unified Process method to develop an Android-based Ramadan application with features such as worship guidance, religious lectures, fasting penalty calculations, adhan, and backup and restore [2]. Thus, previous research shows that there are opportunities for developing mobile-based religious applications, but these are still limited to the Agile methodology. It is this limitation that opens up opportunities for developing applications using a different, more structured methodological approach.

This research focuses on developing a Ramadan application for mobile platforms using the Incremental methodology. Unlike the Agile method, the Incremental methodology allows for the gradual development of a system with results that can be used immediately each time a module is completed. The main focus of this application is to help users manage their worship time, set daily goals, record spiritual achievements, and interact in a positive community. The implementation was carried out using Android Studio with the Java programming language because Android Studio as an IDE provides important features such as a debugger and emulator that facilitate the Android application development process [12], [13]. With this approach, the study is expected to make a real contribution in supporting the Ramadan worship activities of Muslims through the use of digital technology.

METHODOLOGY

The application development method used in this study is the incremental method. In the incremental model, system development begins with determining the requirements that the system must meet. Then, these requirements are divided into small components that can be developed separately. To ensure that the enhancements function properly and do not interfere with existing components, they are thoroughly tested before being accepted and ready for use [14]. This approach was also applied by Gunawan using this method to develop an Academic Progress Information System; they demonstrated that the incremental model enables flexible and structured system development by conducting testing at each stage to ensure the overall quality and functionality of the system [15]. In addition, Rafidah shows that the application of the incremental delivery model in the development of the Cipta Jakarta Hotel information system application allows the system to be used gradually without waiting for all modules to be completed, and facilitates the testing and evaluation process for each increment [16].

System design aims to build a system that is in accordance with user needs. To obtain optimal research results, the design process is carried out using a structured method with a System Development Life Cycle (SDLC) approach, which is implemented through an iterative and incremental model. This process is known as the incremental approach. In general, over time, each feature addition becomes smaller, and all fixed requirements are met incrementally. Each new release of the system includes the addition of previously designed functions, and this process continues until all planned features are successfully implemented. This approach was also applied by Badri & Sari in the development of the Drug Sales Management Information System in a web-based pharmacy, which showed that the incremental model allows flexible and structured system development, with testing at each stage to ensure the quality and functionality of the system as a whole [17]. Similarly, a study by Khan showed that the application of the incremental model to the development of a community satisfaction survey application in West Muna Regency allows the system to be used in stages without waiting for all modules to be completed, and facilitates the testing and evaluation process at each increment [18]. In addition, a study by Melinda showed that the use of the incremental model in the development of an IoT-based locker lending application allows software development to be carried out in a series of iterative cycles, so that each iteration can produce a product that is better than the previous version [19], [20]. The stages of these processes can be seen in Figure 1 below.

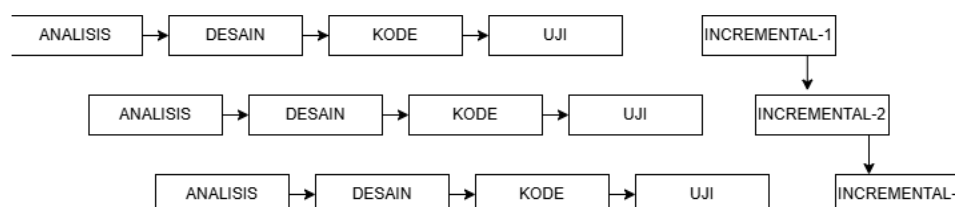


Figure 1. Model Incremental

Figure 1 shows the flowchart of the incremental method in the application development process. This method divides the development process into several increments, each of which consists of analysis, design, coding, and testing. Each incremental stage produces a version of the application that can be used or tested further before proceeding to the next increment.

RESULT & DISCUSSION

The results of system development in the form of an application for fasting management and Ramadan worship, Kareem which will be used by its users. The development of this Kareem application uses the Incremental development method which has 4 stages, namely: analysis, design, code and finally system testing. The explanation of the results of the Incremental development stages is as follows:.

Analysis Results

The first stage is analysis, the main focus is on analyzing functional and non-functional requirements. Functional requirements describe the specific functions or key features that the system must perform, while non-functional requirements relate to operational criteria such as security, performance, and usability. Information regarding functional and non-functional requirements can be seen in Table 1 and Table 2.

Table 1
Functional Requirements

ID	Fitur	Explanation
FR-001	Splash Screen	The initial display when the application is opened before entering the main page.
FR-002	Halaman Home Page	Displays the main information of the application after the user logs in.
FR-003	Login	Users can log in using their username and password
FR-004	Profil Pengguna	Users can manage their profile information, such as name, photo, and bio.
FR-005	Community	Users can interact with other users through forums or discussion groups.
FR-006	Logout	Users can log out of the application

The demographic analysis results show that the majority of respondents are female (73.49%), while only 26.51% are male. In terms of age, most respondents are in the age range of 18-20 years (79.52%), which is a productive age category and actively uses social media. Based on semester, most respondents came from semester III (63.10%), followed by semester I (15.48%), which shows that the majority of respondents are still in the early phase of college. When viewed from the class, the most respondents came from the class of 2023 (63.86%), followed by the class of 2024 (13.25%). This finding illustrates that the research participants are dominated by new and beginning students, who usually use social media more intensely in academic and social activities. Overall, the characteristics of these respondents show the profile of a digitally active young generation, which is in accordance with the research focus on the influence of social environment and intensity of internet use on cyberbullying.

Table 2
Non-Functional Needs

ID	Fitur	Explanation
NFR-001	Availability	The initial display when the application is opened before entering the main page.
NFR-002	Reliability	Displays the main information of the application after the user logs in.
NFR-003	Ergonomy	Users can log in using their username and password
NFR-004	Portability	Users can manage their profile information, such as name, photo, and bio.
NFR-005	Memory	Users can interact with other users through forums or discussion groups.
NFR-006	Response time	Application response time is no more than 3 seconds for each user request.
NFR-007	Security	N/A
NFR-008	Others : Communication languages	All questions and answers in the application must be presented in Indonesian.

In addition to defining the functional and non-functional requirements, this analysis process is also usually described through UML (Unified Modeling Language) diagrams, a modeling language that is widely used in software system design and offers various diagrams to model various aspects of the system [21], One of which is the use case diagram

which is one of the tools that can be used to model user interactions with the system [22]. The following is an illustration of the use case in Figure 2.

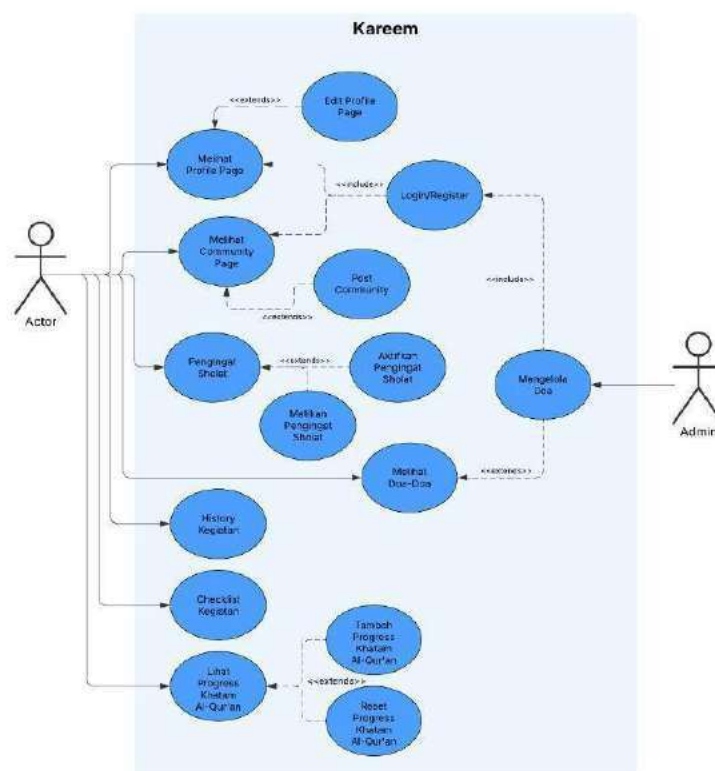


Figure 2. Use Case Diagram

The following is a brief explanation of each use case in the "Kareem" system diagram:

1. Login/Register, Users can login to their account or register to create a new account in order to access all features of the application.
2. View Profile Page: Users can view their profile page which displays their personal information and activities.
3. Edit Profile Page: Users can update personal data such as name, email, or profile photo.
4. View Community Page: Users can access the community page to view posts from other users.
5. Post Community: Users can create new posts on the community page to share information or experiences.
6. Prayer Reminder: Users can use the prayer time reminder feature which can be turned on or off as needed.
7. Enable Prayer Reminder: Enable the prayer time reminder feature so that users are notified.
8. Turn off Prayer Reminder: Disables the prayer time reminder feature when not required.
9. View Prayers: Users can view a list of daily or thematic prayers to recite and practice.
10. Manage Prayers: Admin can add, change, or delete prayers available in the system.
11. Activity History: Users can view the history of activities that have been performed, such as worship or interaction in the application.
12. Activity Checklist: Users can record and mark daily activities that have been carried out.
13. View Khatam Al-Qur'an Progress: Users can see how far they have progressed in reading or memorizing the Qur'an.
14. Add Khatam Al-Qur'an Progress: Users can add their Al-Qur'an recitation achievement to their progress.
15. Reset Khatam Al-Qur'an Progress: Users can reset their Qur'an recitation progress from the beginning if needed.

The Kareem app facilitates users in practicing worship through community features, prayer reminders, prayers, as well as daily activity tracking and Qur'an khatam progress. With a structured and interactive system, this application helps users maintain consistency in worship and build involvement in religious communities.

Furthermore, the use of activity diagrams is a tool that can be used in various ways to describe system behavior and the internal logic of complex operations [23]. A clearer picture of the activity diagram of this system can be seen in

The figure contains four UML Use Case Diagrams, each with a swimlane for 'User' and 'System'.

- Masyarakat: Profile**
 - Use Case 1: Menampilkan Informasi Profil (System)
 - Use Case 2: Mengedit Informasi Profil (System)
 - Flow: User starts at 'View Profile' (System), goes to 'Edit Profile' (System), then to 'Update Profile' (System), and finally to 'View Profile' (System).
- Masyarakat: Merguesee Community Page**
 - Use Case 1: Menampilkan halaman Home (System)
 - Use Case 2: Posting Komentar (User)
 - Use Case 3: Tampilkan Informasi Login (System)
 - Flow: User starts at 'Post Comment' (User), goes to 'Display Login Information' (System), then to 'Login' (User), and finally to 'Register' (User).
- Merguesee Pungingat Sholat**
 - Use Case 1: Menampilkan Informasi Sholat (System)
 - Use Case 2: Pilih Menu Pengingat Sholat (User)
 - Use Case 3: Tampilkan Informasi Pengingat (System)
 - Use Case 4: Pengaturan Dibatalkan (System)
 - Use Case 5: Pengaturan Berhasil (System)
 - Use Case 6: Simpan Pengingat (System)
 - Flow: User starts at 'Select Prayer Reminder Menu' (User), goes to 'Display Prayer Reminder Information' (System), then to 'Prayer Reminder Settings' (System), then to 'Save Prayer Reminder' (System), and finally to 'Prayer Reminder Settings' (System).
- Khataman Al-Quran**
 - Use Case 1: Menampilkan Informasi Khataman (System)
 - Use Case 2: Pilih Menu Khataman Al-Quran (User)
 - Use Case 3: Tampilkan Informasi Khataman (System)
 - Use Case 4: Progress Dibatalkan (System)
 - Use Case 5: Progress Berhasil (System)
 - Use Case 6: Simpan Progress (System)
 - Flow: User starts at 'Select Quran Completion Menu' (User), goes to 'Display Quran Completion Information' (System), then to 'Quran Completion Progress' (System), then to 'Save Progress' (System), and finally to 'Quran Completion Progress' (System).

The activity diagram in Figure 3 illustrates the flow of main activities in the KAREEM application that supports Muslim daily worship. This diagram consists of four main activities, namely:

1. **Profile Management:** Users can view and edit profile information after login or registration. The system will display the page according to the selected action and save the data changes if any.
2. **Community Page Access:** This feature allows logged-in users to access the community page and post messages that will be displayed after being confirmed by the system.
3. **Prayer Reminder:** Users can schedule or disable prayer time reminders. The system then saves the settings according to the user's input.
4. **Khatam Al-Qur'an Tracking:** The user can add or delete reading progress. The system records and saves the changes.

Design Results

The design results of this research are in the form of Class Diagram design and also Sequence Diagram and UI that has been designed in accordance with the features of the fasting management application and Ramadan worship "Kareem".

Class diagram is a diagram describing the structure of the system in terms of defining the classes that will be created to build the system. Class is a collection of objects with and that have a common structure, common behavior, common relationships, and common semantic / words [24]. Class diagrams are used to statically describe the structure of the KAREEM application system, including the main classes involved along with the attributes and methods owned by each class. This diagram also shows the relationship between classes, either in the form of association, inheritance, or dependency. By using class diagrams, developers can understand how data and functions are organized in the system, and help in the coding process to be more structured and modular. Figure 4 below presents the class diagram of the KAREEM application which includes community features, login, register, prayer reminders, khatam Qur'an, and user profile management.

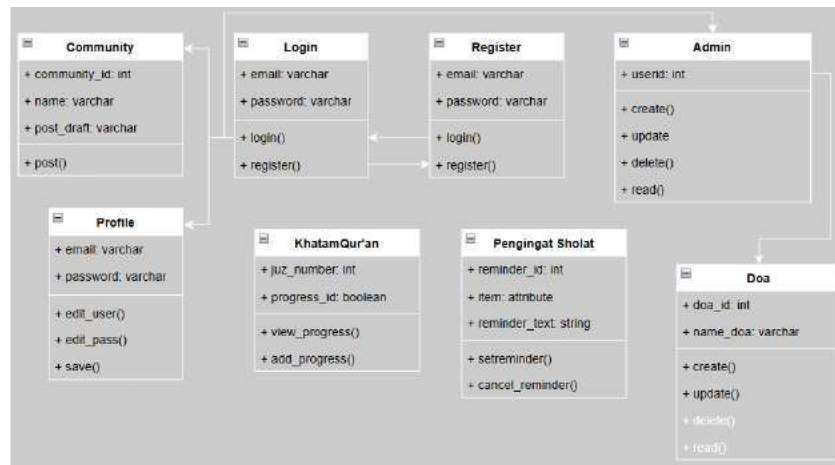


Figure 4. Activity Diagram

The class diagram of the Kareem application consists of eight main classes that represent core features, such as authentication (Login and Register), account management (Profile), community (Community), Qur'an reading progress (KhatamQur'an), prayer reminder, prayer list, and Admin control. Each class has specific attributes and functions that support the functionality of the application in a modular manner. This structure is designed to simplify feature development, data management, and user interface integration.

Sequence Diagram

Sequence diagram is a description of the interaction between objects, which is used to show the communication or messages that exist between these objects [25]. The implementation of interactions between components in the KAREEM application can be visualized through a sequence diagram. This diagram illustrates the flow of communication between objects in the system, ranging from the user interface, controller, to the database, to ensure each process runs in a structured manner and in accordance with their respective functions. The illustration of the sequence diagram in Figure 5 below shows how each main feature, such as login, register, worship reminders, community, to recording the Qur'an's khatam, is executed based on predetermined interaction scenarios.

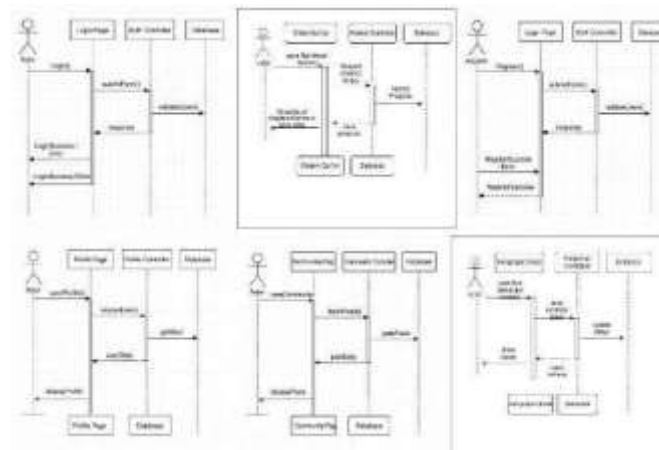


Figure 5. Sequence Diagram

The Kareem app sequence diagram describes the flow of user interaction with the system in six main scenarios: login, register, profile management, Khatam Qur'an progress, community, and prayer reminder. Each process involves data exchange between the interface page, controller, and database. This diagram shows how the system validates, retrieves and updates data according to user requests, ensuring each feature runs in sync and supporting an efficient and structured user experience.

Desain User Interface (UI)

User Interface (UI) and User Experience (UX) are important elements in software development that determine the quality of user interaction with digital products [26]. Furthermore, there is a UI design design that is made based on the previously defined functionality using the Figma design application. There are several incremental stages carried out as follows:

1. Incremental -1

In the first incremental development of the Kareem app, the team successfully completed four main interfaces Splashscreen, Homepage, Profile, and Community. The Splashscreen, Homepage presents a daily prayer schedule and a list of worship services that users can bookmark. The Profile page displays user information and a logout button. While the Community feature allows users to interact with each other through short messages. Navigation between pages is functional and the initial UI design has been designed consistently. This sprint focused on the foundation of the interface and application structure as a basis for the development of dynamic features in subsequent sprints such as authentication and data integration. The development of the KAREEM application was done in stages using an incremental approach. The first incremental focused on completing the initial appearance of the application which included the splash screen, homepage, user profile page, and community features. At this stage, testing was conducted to ensure that all basic functions of the application could run properly, including navigation between screens, retrieval and storage of user data, and community management. Figure 6 below displays the user interface results from the first incremental implementation of the KAREEM application.



Figure 6. Result Incremental 1

2. Incremental -2

In the second incremental, the team developed the Khatam feature, which allows users to mark the progress of reciting the Qur'an per juz from 1 to 30. A visual progress display and reset button were added for easier tracking. This feature supports a more structured and personalized target of khatam during Ramadan. Figure 7 shows the results of the second incremental development, where the team added the Khatam Qur'an feature to facilitate users in recording and monitoring the progress of Qur'an recitation. Each juz from 1 to 30 can be marked according to the user's achievements. The progress display is clearly visualized, complete with a reset button that allows users to repeat the recording if needed. This feature makes it easy to manage Al-Qur'an recitation targets during Ramadan in a more systematic and personalized way.



Figure 7. Result Incremental 2

3. Incremental -3

In the third incremental, the team added the Register and Login features to manage user authentication. Users can now create a new account with a username and password, and login to access personalized app features. Navigation between the login and register forms is intuitive, supporting a more secure and structured user experience. The interface remains consistent with Kareem's design. Figure 8 shows the third incremental development, which focused on implementing the Register and Login features. This feature allows users to create a new account by filling in username and password data, and login again to access personalized application features. Navigation between forms was designed intuitively to support user convenience. In addition, the interface design is kept consistent with the visual identity of the KAREEM application to provide a cohesive and professional experience.

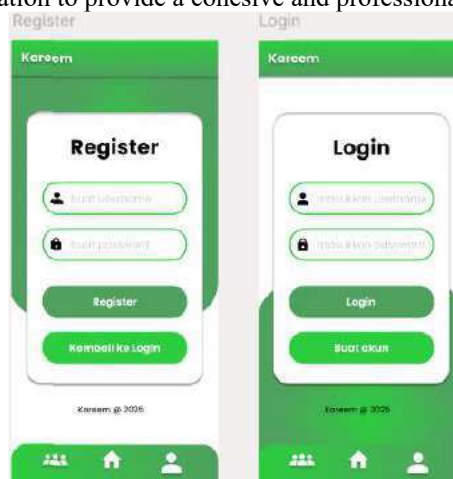


Figure 8. Result Incremental 3

4. Incremental -4

In the fourth incremental, the team added three important features: Prayer Reminder, Worship History, and Daily Prayer. The reminder feature allows users to enable notifications for each prayer time. Worship history presents daily achievements and worship progress, including khatam status. While the prayer feature provides text and translation of important prayers such as meal prayers, iftar, and fasting intentions. Figure 9 shows the result of the fourth incremental which includes the addition of three main features, namely Prayer Reminder, Worship History, and Daily Prayer. The Prayer Reminder feature allows users to set notifications at each prayer time, so that worship is more scheduled. The Worship History feature provides a recap of daily achievements and worship progress, including progress in reading the Qur'an. Meanwhile, the Daily Prayer feature provides a collection of important prayers complete with Arabic text, translations, and brief explanations, such as prayers for eating, breaking the fast, and fasting intentions. These three features strengthen the role of the KAREEM application as a personal worship assistant during Ramadan.

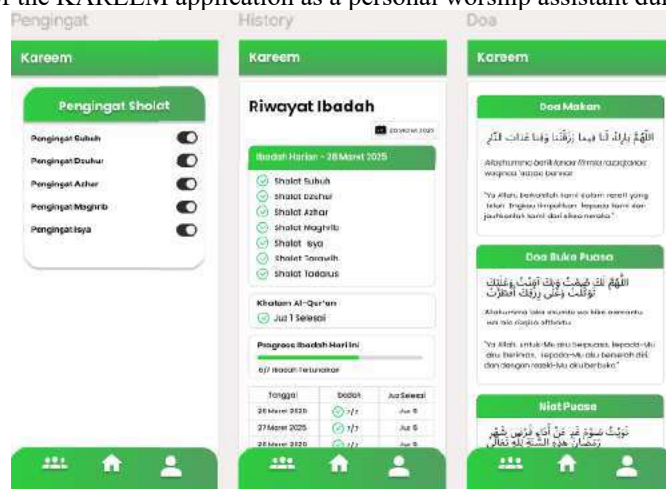


Figure 9. Result Incremental 4

Implementation Results

Based on the results of the analysis and design that has been carried out, the development of the KAREEM

application is carried out using the incremental method in 7 stages of development. Each stage represents the addition of key features, which are then implemented in the form of coding, integration with the database, and functionality testing.

Coding Result

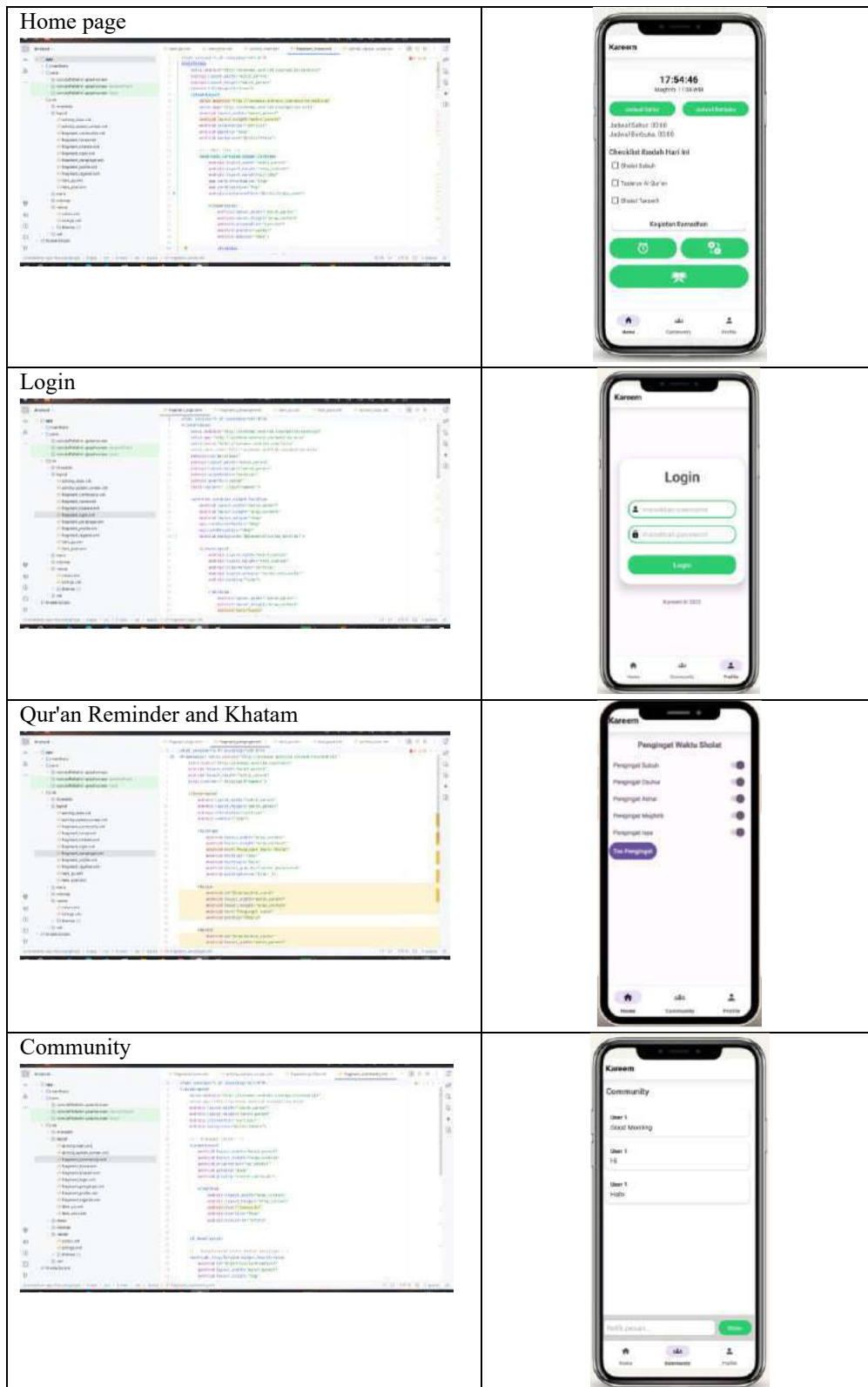
The result of this coding process includes all the main features contained in the KAREEM application. Development was done using Android Studio as the main environment due to its ability to support Java-based Android application development, which is also the main programming language in this project. Android Studio provides a structured interface, complete debugging tools, and strong integration with emulators for application testing purposes.

In addition, the KAREEM application also uses Firebase as a backend database system. Firebase was chosen because of its ease of integration with Android applications, support for real-time storage, and security features such as authentication and database rules that ensure user data remains safe and reliable. As explained by Meiah, Firebase Realtime Database enables realtime data synchronization with low latency, making it suitable for mobile applications that require instant data updates [27]. The KAREEM application also uses the Aladhan API to retrieve and display data about the 5 daily prayer times based on the user's location.

Each stage of feature development is tested using a live Android device. Testing on real devices allows developers to evaluate application performance in real usage conditions, ensuring each function runs well, is responsive, and free of bugs before being released to users. The implementation process of each main feature in the KAREEM application is documented in the form of visual coding and display results on user devices. This implementation shows how the system design has been realized into the form of real interfaces and functions that can be accessed by users. Each implementation result reflects the achievement of the function as planned in the design stage. Table 4 below presents the coding results for each feature along with the final appearance produced in the application.

Table 3
Result Implementasi

Coding	Result Display
Splash Screen 	



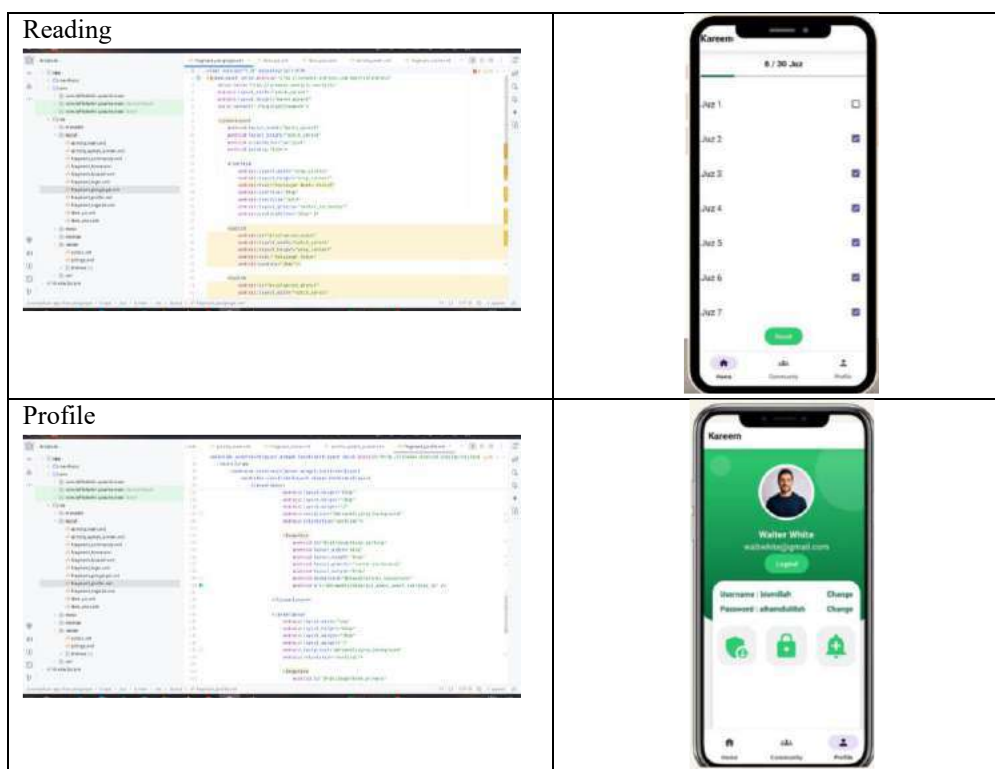


Table 3 shows the coding results that represent the seven incremental stages in the development of the KAREEM application. The first stage begins with the creation of a Splash Screen that functions to display the logo and identity of the application when it is first run, as an initial transition before the user is directed to the main display. The second stage is the Home page, which provides main information such as worship times, daily worship progress, and navigation to various other features in the application. The third stage is the development of the Login feature, which allows user authentication via email and password to ensure secure and personalized access.

Furthermore, the fourth stage includes the Edit Profile feature, where users can update personal information such as name and profile photo stored in real-time in Firebase. The fifth phase added the Worship Reminder feature, which allows users to set notifications for prayer, dhikr, and other worship services to support consistency in worship. In the sixth stage, the Khatam Qur'an feature was developed, which provides a record of the progress of reading the Qur'an and helps users achieve khatam targets in a structured manner. Finally, the seventh stage presents the Community feature, a community space that allows users to share worship achievements, provide motivation, and build positive communication between users.

Testing Results

Application testing is carried out using the Black Box method, where end users test the functions of the application without the need to know the details of the program code used. This approach allows the identification of problems from the user's side, thus ensuring that the application features run as expected. As explained by (Agustian et al., 2021), the Black Box method is effective in identifying errors from the user's perspective without examining the source code [28]. In addition, Wulandari emphasizes the importance of the Equivalence Partitioning technique in Black Box Testing to efficiently validate application inputs and outputs [29]. The series of test scenarios used can be seen in Table 5 below.

Table 4
Black Box Testing Results

Black Box Testing				
Test Code	Test Case	Expected Results	Results Obtained	Status
BB-1	Register with email and	"Register successfully" message is displayed	"Register successfully" message	Successful

	password filled		displayed	
BB-2	Register with email and password blank	"Enter Email/Password" error message is displayed	The error message "Enter Email" error message is displayed	Successful
BB-3	Register with email filled in and empty password	The error message "Enter Password" is displayed	The error message "Enter Password" error message is displayed	Successful
BB-4	Login with email and password blank	"Enter Email/Password" error message shown	The error message "Enter Email" error message is displayed	Successful
BB-5	Login with an email and password that not already registered	The error message "Login Failed" is displayed	The error message "Login Failed!" appears	Successful
BB-6	Login with your email and password correct	A successful "Login Successful" message is displayed and redirected to the profile page	The message "Login Successful" is displayed and redirected to the profile page	Successful
BB-7	Click the checkbox next to the Qur'an	Displays a check and the progress bar increments	Check appear and progress bar added	Successful
BB-8	Click the reset button on khatam Al-Qur'an	Progress bar and ticks are cleared	Progress bar and ticks that appear erased	Successful
BB-9	Update schedule prayer schedule without turning on the location	The schedule is not displayed and displays the error message "Failed to update schedule"	The schedule is not displayed and the error message "Failed to update schedule" shown	Successful
BB-10	Update the schedule prayer schedule by turning on the location	The schedule is updated and Display the prayer schedule	Prayer schedule displayed and updated	Successful
BB-11	Click the Logout button	The account exits and returns to the login page.	Account successfully logged out and returned to login page	Successful
BB-12	Activate Reminder Fajr	At Fajr time, displays the Fajr prayer notification	Prayer notification displayed Fajr when entering Fajr prayer time	Successful
BB-13	Disable Dawn Reminder	At dawn, does not display Fajr prayer notification	Fajr prayer notification does not appear when entry of prayer time Fajr	Successful

BB-14	Activate Reminder Dzuhur	At Dhuhr time, displays the Dhuhr prayer notification	Prayer notification displayed Dzuhur when entering Dzuhur prayer time	Successful
BB-15	Disable Dhuhr Reminder	At Dhuhr time, does not display Dzuhur prayer notification	Prayer notification Dhuhr prayer notification does not appear when the time for Dzuhur prayer time	Successful
BB-16	Enable Asr Reminder	At Asr time, displays the Asr prayer notification	Prayer notification displayed Asr prayer notification when entering Asr prayer time	Successful
BB-17	Disable Reminders Asr Prayer	At Asr time, Does not display Asr prayer notification	Prayer notification Asr prayer notification does not appear when entering Asr prayer time	Successful
BB-18	Activate Reminder Maghrib	At Asr time, display Maghrib prayer notification	Prayer notification displayed Asr prayer notification when entering Asr prayer time	Successful
BB-19	Disable Reminder Maghrib	At Asr time, Does not display Maghrib prayer notification	Prayer notification Asr prayer notification does not appear when entering Maghrib prayer time	Successful
BB-20	Activate Reminder Isha	At Asr time, display Isha prayer notification	Prayer notification displayed Asr prayer notification when entering Asr prayer time	Successful
BB-21	Disable Reminders Isha	At Asr time, Does not display Isha prayer notification	Prayer notification Asr prayer notification does not appear when entering Isha prayer time	Successful
BB-22	Type a message and send it on the Community	The message will displayed on Community page	Message appears on Community page	Successful
BB-23	Order blanks and click send on page Community	The message will not displayed on Community page	Message not displayed on Community Page	Successful

The test results in Table 4 show that all the main functions of the KAREEM application have run according to the designed scenario. Each test, starting from the registration process, login, worship reminders, to interaction in the community feature, provides results that are consistent with predetermined expectations. All 23 test cases were successfully executed without any bugs or errors that disrupt functionality. This proves that the application system is able to respond appropriately to user input, both for normal conditions and boundary conditions. This test also confirms that the integration of Firebase and Aladhan API as the backend database works optimally, especially in storing user data, updating worship schedules, and displaying real-time worship notifications.

Next comes white-box testing. White-box testing is a software testing method that focuses on evaluating the internal structure and logic of program code [30]. In this approach, testers have full access to the source code and design

Daffa Fakhir; 72

test scenarios to ensure that each execution path in the program functions according to specifications. This technique is very effective for detecting logic errors, unreachable paths, and potential security vulnerabilities. According to research by Rafli white-box testing enables the identification of logic errors and inappropriate code structures, thereby improving overall software quality [31]. However, whitebox testing requires a deep understanding of the internal structure of the software and can be a complex and time-consuming process, especially for large-scale systems. Nonetheless, this approach remains important in ensuring the overall quality and security of the software.

To ensure that each internal component of the application functions as expected, testing is performed using the white-box method. This includes evaluating the code structure, program logic, and execution path to identify potential errors or discrepancies.

Table 5
White Box Testing Results

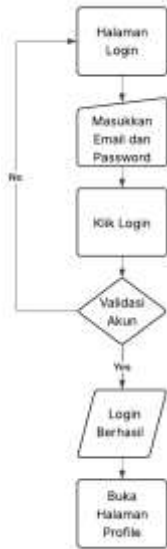
Coding	Flowchart Login	Flowgraph Login
Splash Screen 		
$V(G) = (E - N) + 2$ $V(G)$ = Number of Regions E = Number of edges symbolized as arrows N = Number of graph nodes or nodes with circle images $V(G) = (6 - 6) + 2 = 2$ Path 1 = 1-2-3-4-5-6 Path 2 = 1-2-3-4-1-2-3-4-5-6 The following is an illustration of the number of regions that have been determined in Table 6.		

Table 6
Login Feature Path Testing Results

Path	1	Testing Result
Track	1-2-3-4-5	Successful
	1 Open Khatam Page 2 Show Juz List 3 Check Juz 4 Update Progress 5 Show Progress	
Path	1	
Track	1-2-3-4-1-2-3-4-5	
Skenario	1 Open Khatam Page 2 Show Juz List 3 Check Juz (No) 4 Show Progress	

	1 Open Khatam Page2 Show Juz List 3 Check Juz (Yes) 4 Update Progress 5 Show Progress Bar	Successful
--	--	------------

Once the number of independent paths is known, a comparison is made using the relationship table between procedure type and risk level as shown in Table 7.

Table 7
Procedure Type and Risks

Value	Tippe Prosedur	Risk Level
1-4	Short Procedures	Low
5-10	Well-structured Procedure	Low
11-20	Complex Procedures	Medium
21-50	More complex and critical procedures	High
>50	Procedures are difficult to test and error prone	Very High

Based on the table above, the function of this login feature has a short procedure and a low level of risk because it has 2 independent paths. After the path is known, the next step is a test case, such as table 8.

Table 8
Test Case Fitur Login

No	Scenario Name	Activities	Expected Outcome	Result	Description
1	Feature Test Khatam 1	User selects Ticks juz	Progres bar bertambah sesuai jumlah juz yang dicentang	Display the progress bar incremented by the amount checked	Valid
2	Feature Test Khatam 2	User chooses not to check juz, then User chooses To tick the juz and the progress bar is displayed.	Progress bar is not displayed, then Display progress bar after user ticks juz	No Displays the Progress bar before it is checked, then displays the progress bar after it is checked	Valid

Based on Table 8, testing of the Khatam feature shows that the designed functionality has worked as expected. In the first scenario, the user ticks one of the available juz. The system responds by displaying a progress bar that increases according to the number of juz checked. The results of this test show that the feature runs well and is declared valid.

Meanwhile, in the second scenario, the user initially does not check any juz, so the progress bar is not displayed. After the user ticks the juz, the system starts displaying the progress bar according to the progress achieved. This shows that the system can distinguish the initial condition without progress and the condition after interaction from the user. Thus, this test also showed appropriate results and was declared valid.

CONCLUSION

Based on the results of implementation and testing, the KAREEM application was successfully developed in stages using the incremental method which includes seven main stages, namely Splash Screen, Home, Login, Edit Profile, Worship Reminder, Khatam Qur'an, and Community. This application was built using Android Studio and utilizes Firebase as a real-time data storage medium. Testing was conducted using a combination of black box and white box methods. In general, black box testing shows that all features can run as expected without any system errors. Meanwhile, the Login and Khatam Qur'an features were further tested using the white box method to ensure that all program logic paths, branching, and execution flow had run correctly. The test results show that the login feature is able to handle valid and invalid inputs appropriately, while the khatam feature can display progress accurately according to user interaction. All tests show that the core functions of the application have run well and in accordance with the design.

Although the app has functioned well, further development is still needed so that the app can provide a more optimal experience for users. The app interface can be improved by adding visual transitions and animations to make interactions feel more responsive and engaging. In terms of security, the addition of two-factor authentication to the login feature can improve user account protection. In addition, voice-based reminders can be a more inclusive alternative, especially for users who require audio access. Personalization of the display theme can also be an added attraction to make users feel more comfortable when using the app. Continued testing on various devices and network conditions is highly recommended to ensure stable and optimal app performance in various situations. With thorough testing and continuous development plans, the KAREEM app has great potential to become an effective and useful digital worship companion for Muslims, especially during the month of Ramadan.

REFERENCES

- [1] H. Lisa, Mardiah, and M. Napratilora, "Program Pesantren Kilat Ramadhan untuk Meningkatkan Motivasi Ibadah Siswa SMPN 3 Tembilahan Hulu," *ABDIMASY: Jurnal Pengabdian dan Pemberdayaan Masyarakat*, vol. 1, no. 2, pp. 63–74, 2020, doi: 10.46963/ams.v1i2.268.
- [2] A. A. Kurniawan, M. A. Izzuddin, and F. M. Amin, "Penerapan Agile Unified Process Pada Pengembangan Aplikasi Puasa Ramadhan Berbasis Mobile Android," *Systematics: Information Systems and Informatics Journal*, vol. 5, no. 2, pp. 28–36, 2019, doi: 10.29080/systemic.v5i2.740.
- [3] F. N. Monalisa, R. Akmalia, A. S. S. Harahap, and P. F. Aulia, "Upaya dalam Menumbuhkan Karakter Agamis Siswa pada Bulan Suci Ramadhan di Madrasah Tsanawiyah Negeri Binjai," *Fondatia: Jurnal Pendidikan Dasar*, vol. 6, no. 2, pp. 206–222, 2022, doi: 10.36088/fondatia.v6i2.1791.
- [4] R. R. Fadillah, D. B. Alfarizy, and A. Rafaendra, "Agama dan Era Digital: Tantangan dan Peluang dalam Mempertahankan Nilai-Nilai Keagamaan," *Jurnal Agama dan Lintas Budaya*, vol. 5, no. 2, pp. 151–165, 2024, [Online]. Available: <https://maryamsejahtera.com/index.php/Education/index>
- [5] Zahraini, Akib, Rosidin, and O. Sulaeman, "Islamic Education Reform in the Digital Age: Challenges and Opportunities for a Modern Curriculum," *Jurnal Noesantara Islamic Studies*, vol. 2, no. 1, pp. 1–11, 2025, doi: 10.70177/jnis.v2i1.1841.
- [6] N. A. Mar, "Integration of Technology and Islamic Education in the Digital Era: Challenges, Opportunities and Strategies," *Journal of Scientific Insight*, vol. 4, no. 4, pp. 69–84, 2019, doi: 10.69930/jsi.v1i1.74.
- [7] M. Gilang Ramadhan, M. A. Alif, M. I. Hidajat, P. Nirmala, N. Agusnaya, and W. Hidayat, "PERANCANGAN APLIKASI RESERVASI FUTSAL (FUTSAL-IN) GUNA MEMPERMUDAH PEMESANAN APLIKASI SECARA ONLINE", [Online]. Available: <https://journal.diginus.id/index.php/decoding>
- [8] G. G. Ramadhan, A. H. Brata, and K. C. Brata, "Pengembangan Aplikasi Perangkat Bergerak Manajemen Ibadah Harian Umat Islam Berbasis Android," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 3, pp. 1190–1195, 2021, [Online]. Available: <http://jptiik.ub.ac.id/>
- [9] S. Rahmah and M. F. Hadi, "Pengaruh Penggunaan Aplikasi Pembelajaran Berbasis Android Terhadap Motivasi Belajar Siswa," *Teknologi Pendidikan*, vol. 2, no. 1, pp. 89–96, 2023, doi: 10.56854/tp.v2i1.223.
- [10] M. C. Cahyani, D. Nababan, and Risald, "Aplikasi Penuntun Pelaksanaan Ibadah Puasa Ramadhan Berbasis Android," *METHOMIKA: Jurnal Manajemen Informatika dan Komputerisasi Akuntansi*, vol. 6, no. 2, pp. 117–123, 2022, doi: 10.46880/jmika.vol6no2.pp117-123.
- [11] I. Larasati, A. N. Yusril, and P. A. Zukri, "Systematic Literature Review Analisis Metode Agile Dalam Pengembangan Aplikasi Mobile," in *SISTEMASI*, 2021, p. 369. doi: 10.32520/stmsi.v10i2.1237.
- [12] I. P. Dewi, L. Mursyida, and A. D. Samala, *Dasar-dasar Android Studio Dan Membuat Aplikasi Mobile Sederhana*, vol. 10. Widina Bhakti Persada Bandung, 2021. [Online]. Available: <https://repository.penerbitwidina.com/publications/345659/dasar-dasar-android-studio-dan-membuat-aplikasi-mobile-sederhana#cite>
- [13] F. R. Chan, H. Dusri, M. Ramdhani, Hanifah, and L. Efriyanti, "Perancangan Aplikasi Pengelolaan Gudang Berbasis Android Menggunakan Android Studio," *Jurnal Informatics and Advanced Computing*, vol. 3, no. 2, pp. 103–107, 2022, doi: 10.35814/jiac.v3i2.4522.
- [14] G. E. Peta, Z. Ilhan, and A. S. Muhammad, "Perancangan Sistem Informasi Masjid Nurut Taqwa Berbasis Web dengan Metode Incremental," *Scientia Sacra: Jurnal Sains, Teknologi dan Masyarakat*, vol. 3, no. 2, pp. 176–189, 2023, [Online]. Available: <https://www.pijarpemikiran.com/index.php/Scientia/article/view/509>
- [15] I. M. A. O. Gunawan, G. Indrawan, and S. Sariyasa, "Pengembangan Sistem Informasi Kemajuan Akademik Menggunakan Model Incremental Berbasis Evaluasi Usability Dan White Box Testing," *SINTECH (Science and Information Technology Journal)*, vol. 4, no. 1, pp. 67–78, 2021, doi: 10.31598/sintechjournal.v4i1.661.
- [16] V. Rafida, "Penerapan Model Incremental Delivery pada Pengembangan Aplikasi Sistem Informasi Hotel Cipta Jakarta," *URANUS: Jurnal Ilmiah Teknik Elektro, Sains dan Informatika*, vol. 2, no. 4, pp. 41–70, 2024, doi: 10.61132/uranus.v2i4.469.
- [17] F. Badri and S. U. R. Sari, "Sistem Informasi Manajemen Penjualan Obat Pada Apotek Berbasis Web Dengan

- Menggunakan Metode Incremental Model,” *Jurnal Sistem Informasi dan Informatika*, vol. 1, no. 2, p. 70, 2023, [Online]. Available: <http://jurnal.unidha.ac.id/index.php/jteksis>
- [18] S. F. A. Khan, A. F. Novianti, J. N. Dini, and A. M. R., “Pengembangan Sistem Informasi Muna Barat Satisfaction Survey System (MUARA),” *Indonesian Technology Education Journal*, vol. 2, no. 2, pp. 157–171, 2024, doi: 10.61255/itej.v2i2.426.
- [19] M. Melinda, S. R. R. Na, Y. Nurdin, and Y. Yunidar, “Implementation of System Development Life Cycle (SDLC) on IoT-Based Lending Locker Application,” *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 4, pp. 982–987, 2023, doi: 10.29207/resti.v7i4.5047.
- [20] K. Rokoyah, Y. I. Chandra, and S. Lukman, “Penerapan Model Incremental dalam Merancang Aplikasi Pengenalan Bentuk dan Fungsi Gigi Pada Manusia,” *Jurnal Ilmiah SIKOMTEK*, vol. 12, no. 2, pp. 42–47, 2022, [Online]. Available: <https://sikomtek.jakstik.ac.id/index.php/jurnalsikomtek/article/view/18>
- [21] S. W. Ramdany, S. A. Kaidar, B. Aguchino, C. A. A. Putri, and R. Anggie, “Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web,” *Jurnal Industrial Engineering Systems*, vol. 5, no. 1, pp. 30–41, 2024, doi: 10.31599/2e9afp31.
- [22] L. Setiyani, “Desain Sistem : Use Case Diagram,” in *Seminar Nasional Inovasi dan Adopsi Teknologi*, 2021, pp. 246–260. [Online]. Available: <https://ejournal.rosma.ac.id/index.php/inotek/article/view/183>
- [23] S. Al-Fedaghi, “Validation: Conceptual versus Activity Diagram Approaches,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 6, pp. 287–297, 2021, doi: 10.14569/IJACSA.2021.0120632.
- [24] H. Malius, Apriyanto, and A. A. H. Dani, “Sistem Informasi Sekolah Berbasis Web Pada Sekolah Dasar Negeri (SDN) 109 Seriti,” *Indonesian Journal of Education and Humanities*, vol. 1, no. 3, pp. 156–168, 2021.
- [25] S. Nabila, A. R. Putri, A. Hafizhah, F. H. Rahmah, and R. Muslikhah, “Pemodelan Diagram UML Pada Perancangan Sistem Aplikasi Konsultasi Hewan Peliharaan Berbasis Android (Studi Kasus: Alopet),” *Jur.Ilm.Kom.Bis*, vol. 12, no. 2, pp. 130–139, 2021, doi: 10.47927/jikb.v12i2.150.
- [26] D. Hanafi, I. A. Prabadhi, and G. B. Hertantyo, “Perancangan Desain User Interface / User Experience: Systematic Literature Review,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 3, pp. 4665–4671, 2025.
- [27] K. N. M. Ngafidin, A. Arista, and R. N. S. Amriza, “Implementasi Firebase Realtime Database pada Aplikasi FeedbackMe sebagai Penghubung Guru dan Orang Tua,” *J. RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 2, pp. 327–334, 2021, doi: 10.29207/resti.v5i2.2909.
- [28] E. Agustian, A. W. Pamungkas, E. Rachman, S. Mulyati, and Yulianti, “Pengujian Aplikasi Sistem Informasi Laboratorium Kesehatan Daerah DKI Jakarta Secara Black Box,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 4, no. 3, pp. 129–137, 2021, doi: 10.32493/jtsi.v4i3.8520.
- [29] A. S. Wulandari, A. Saepudin, M. P. Kinanti, Z. Sudesi, A. Saifudin, and Y. Yulianti, “Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 2, pp. 102–109, 2022, doi: 10.32493/jtsi.v5i2.17561.
- [30] H. Gusdevi, S. Kuswayati, M. Iqbal, M. F. A. Bakar, N. Novianti, and R. Ramadan, “Pengujian White-Box Pada Aplikasi Debt Manager Berbasis Android,” *Naratif: Jurnal Nasional Riset, Aplikasi dan Teknologi Informasi*, vol. 4, no. 1, pp. 11–22, 2022, doi: 10.53580/naratif.v4i1.147.
- [31] H. Rafli, “Penerapan Whitebox Testing pada Pengujian Sistem Menggunakan Teknik Basis Path,” *Journal of Information Systems and Informatics Engineering*, vol. 8, no. 1, pp. 101–111, 2024, [Online]. Available: <https://doi.org/10.35145/joisie.v8i1.4229>