

Android-based Nahdlatut Turots Manuscript Application

Mochammad Irfan Efendi^{1*}, Dwi Cahyono²

^{1*}Teknik Informatika, Fakultas Teknik, Universitas Dr. Soetomo, Indonesia
Jl. Semolowaru No.84, Menur Pumpungan, Kec. Sukolilo, Surabaya, Jawa Timur
^{1*}m.irfan.efendi19@gmail.com, ²dwik@unitomo.ac.id

Article history:

Received 20 August 2025
Revised 10 October 2025
Accepted 28 November 2025
Available online 30 November 2025

Keywords:

Turots ,
Digital Preservation,
Android Application,
Manuscripts,
Pegon,
Nusantara Scholars.

Abstract

The digitization of Turots manuscripts by Nahdlatut Turots is an important step in preserving Islamic intellectual heritage, which is vulnerable to deterioration. Turots refers to the works of scholars in fields such as fiqh, tauhid, tasawuf, and Arabic language, often written in Arabic script or Pegon. This research aims to develop an Android-based application to facilitate the digital preservation and management of kitab Turots. The application is designed to improve access to classical books and create opportunities for further research. Through this application, it is expected that the preservation of kitab Turots will be enhanced, accessibility will be increased, and the digital conservation of Nusantara scholars' intellectual heritage will be supported. The system is developed using the System Development Life Cycle (SDLC) with the Rapid Application Development (RAD) model to accelerate the development process while ensuring system quality. The Technology Acceptance Model (TAM) is used to evaluate user acceptance, measuring perceived usefulness, ease of use, and behavioral intention to use. Additionally, Google Cloud computing services are integrated to enhance storage, security, and scalability, ensuring seamless access to digital manuscripts. The system is tested using the black-box testing method to verify functionality.



This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. ©2025 by author.

I. INTRODUCTION

The development of digital technology has opened new opportunities in various fields, including the preservation of turots manuscripts. Turots represents the intellectual heritage of Islam in the form of classical scholars' writings, covering various disciplines such as fiqh, tauhid, tasawuf, and the Arabic language. In Indonesia, turots books play a crucial role in the development of Islamic knowledge, particularly in pesantren (Islamic boarding schools) [1]. However, many turots manuscripts suffer from physical deterioration or even disappear due to age, environmental factors, and the lack of conservation efforts. Therefore, digitization is a crucial solution to ensure their accessibility and preservation for future generations.

In this effort, Nahdlatut Turots emerges as a community dedicated to collecting, preserving, and studying turots manuscripts in Indonesia [2]. However, the primary challenge in turots preservation lies in the limited access and the risk of physical damage to the original manuscripts. Hence, a digital system is needed to facilitate the search, storage, and accessibility of turots books.

^{1*} Corresponding author

The Android-based Nahdlatut Turots application is developed to digitize turots books and provide broader access to the public. This application includes key features such as a digital turots book collection, author and manuscript origin information, category-based search, and turots collection statistics. With this application, users can easily find and study turots books without needing direct access to physical manuscripts.

The development of this application utilizes the System Development Life Cycle (SDLC) method with the Rapid Application Development (RAD) model, allowing for faster and more flexible system development [3]. To measure user acceptance, this study employs the Technology Acceptance Model (TAM) evaluation framework, which focuses on three main aspects: Perceived Usefulness (PU), Perceived Ease of Use (PEU), and Behavioral Intention to Use (BI) [4]. Additionally, black-box testing is applied to ensure that the application's features function as expected [5].

The Nahdlatut Turots application is expected to contribute to the preservation and dissemination of the intellectual heritage of Nusantara's scholars. This digitization effort not only protects turots books from damage but also provides wider access for academics, students, and the general public who wish to study classical texts more easily and practically.

II. METHODS

The research method used in this research is System Development Life Cycle (SDLC) with Rapid Application Development (RAD) model. SDLC is a systematic approach in building and developing software systems that includes planning, analysis, design, implementation, and maintenance stages [3]. Meanwhile, the RAD model emphasizes rapid and iterative development with intensive user involvement and continuous feedback. The stages of SDLC with the RAD model used in this research are illustrated in figure 2.

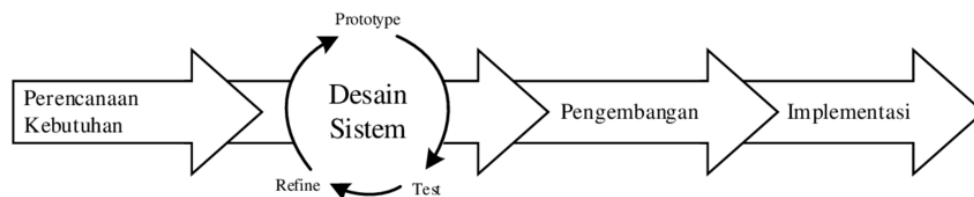


Figure 1 Rapid Application Development (RAD)

The use of SDLC with the RAD model enables the development of high-quality applications in a shorter time and more cost-efficient manner [3]. This approach is particularly suitable for projects that require rapid adaptation to changing user needs or business environments. Thus, the Android-based Nahdlatut Turots Manuscript application can be developed more effectively to support the preservation and development of Islamic intellectual values.

A. Preliminary Study

The initial stage of this research is a preliminary study aimed at gathering information about the Nahdlatut Turots manuscripts, understanding the digital archive system, and identifying the needs and expectations of Nahdlatut Turots. This stage involves a literature review and observations of existing manuscript digitization systems to gain deeper insights into relevant technologies and methods.

B. Requirements Analysis

The requirements analysis stage involves identifying problems and system specifications for the Nahdlatut Turots manuscript application based on Android. Data collection is conducted through direct interviews with Nahdlatut Turots administrators and members to understand the needs of the digital archive repository and the required system features. The results of this analysis serve as the foundation for system design.

C. System Design

At this stage, system design is carried out based on the results of the needs analysis that has been collected. This design stage includes several main aspects as follows:

1. System Architecture

The architecture of the android-based Nahdlatut Turots manuscript application system is designed to support the development, preservation, and accessibility of manuscripts through android applications. The system consists of several main components that work in an integrated manner to provide the functions of digitization, storage, and access to the collection of turots books by Nusantara scholars. This system integrates Google Cloud computing technology to ensure secure, scalable, and efficient management of digital manuscripts [6]. Details regarding the components and workflow of this system can be seen in Figure 3.

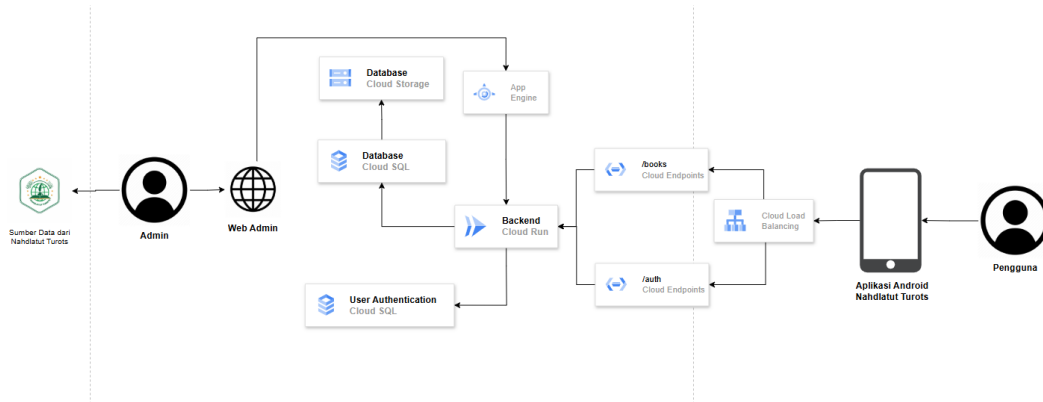


Figure 2 System Architecture

The system architecture of the Nahdlatut Turots application utilizes Google Cloud Platform (GCP) as the backend and the Android application as the client. The Android application serves as the user interface, handles authentication via JWT (JSON Web Token), and accesses manuscript data via API [7]. The backend is developed using Node.js with Express.js, ensuring efficient handling of API requests and seamless communication between the client and the server[8].

The API Gateway manages requests, forwards them to Google Cloud App Engine via Google Cloud Load Balancer, and uses Google Cloud SQL and Google Cloud Storage to manage metadata and manuscript files. Content management is done through a web admin built with Node.js and hosted on Google Cloud App Engine, allowing administrators to manage the book collection with JWT authentication.

The system also utilizes Cloud Run to run microservices and Google Cloud Load Balancer for optimal traffic distribution. This integration of Node.js with GCP services ensures security, scalability, and efficiency in the management and access of digital kitab turots.

2. Use Case Diagram

Use case diagrams are visual diagrams in the Unified Modelling Language (UML) that describe the relationships and interactions between users and the system [9]. To properly describe and ensure that the system runs according to predetermined procedures, a Use Case Diagram is made that explains the workflow of the user actor (user) and administrator (admin) who will interact with the android-based Nahdlatut Turots manuscript application system. Figure 4, which is a Use Case Diagram, provides an overview of the course of the system being built.

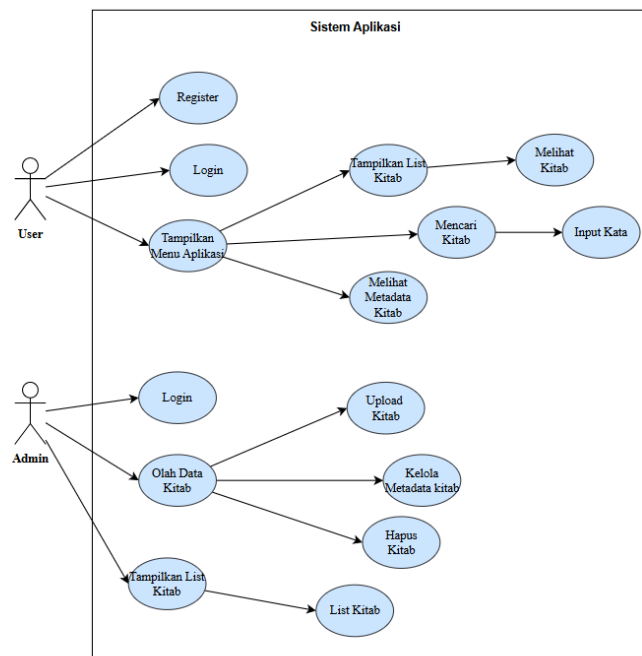


Figure 3 Use Case Diagram

3. Class Diagram

Class diagram is a diagram that describes the static structure of the classes in a system, including the attributes, operations, and relationships between these classes [10]. The Class Diagram in this research visualizes the structure of the classes involved in the application being developed. This diagram is very important because it helps in modelling the relationships and interactions between various components in the application system, which will ensure that the application can operate in accordance with the needs and objectives that have been set in this study. Figure 5 Class diagram is also used as a database design that will be used in the application later.

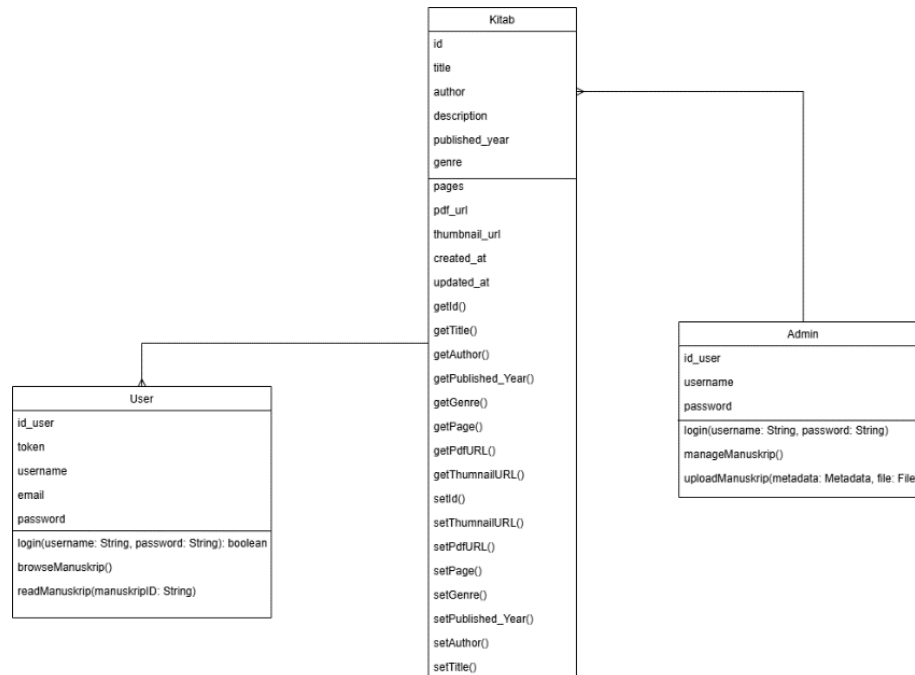


Figure 4 Class Diagram

D. System Development

The system development stage begins with implementing the previously designed system, which is developed progressively from the beta version to the final version. The development process is carried out iteratively, with each iteration producing a more refined application version based on user feedback. This stage involves the implementation of the main features identified during the requirements analysis stage.

E. Implementation

The implementation stage involves deploying all software components that have been designed and developed. This process includes:

1. Code Development

At this stage, the system's source code is written based on the established design specifications. The frontend of the system is developed using Vanilla JavaScript for the web-based administration panel, ensuring lightweight performance without relying on heavy frameworks [11]. Meanwhile, the Android application is built using Kotlin, a modern programming language optimized for Android development, providing better performance and security compared to Java [12]. The backend system is developed using Node.js with Express.js, which facilitates API-based communication between the Android application and the cloud database. The system uses MySQL as its database management system, ensuring efficient data storage, retrieval, and management for manuscript digitization [13]. Additionally, integration with Google Cloud computing services enhances data security and optimizes performance in handling large-scale manuscript archives.

2. Module Integration

Once individual components are developed, they must be integrated to ensure seamless interoperability. The integration process includes:

- a. Connecting the Kotlin-based Android application with the backend API using Retrofit for efficient data fetching [14].

- b. Ensuring real-time synchronization between the mobile application and Google Cloud Storage for manuscript access.
 - c. Establishing secure authentication using Firebase Authentication, allowing users to log in using email, Google accounts, or other authentication methods [15].
 - d. Implementing a web-based admin panel (built with Vanilla JavaScript) that allows administrators to manage manuscript data, user access, and metadata.
 3. Initial Testing

Before proceeding to formal testing phases, an initial testing process is conducted to ensure that each module works correctly. The testing phase includes:

 - a. Unit Testing: Verifying individual components, such as API endpoints, UI elements, and database queries, to confirm they function properly.
 - b. Integration Testing: Ensuring smooth communication between the Kotlin Android app, Vanilla JavaScript admin panel, and the Node.js backend.
 - c. Performance Testing: Evaluating the system's response time when handling manuscript searches, page navigation, and cloud storage interactions.
 - d. Security Testing: Assessing authentication mechanisms and data encryption to prevent unauthorized access.

F. System Testing

System testing is conducted to ensure that the Nahdlatut Turots manuscript application functions as expected. This testing employs the black-box testing method, which focuses on application functionality without inspecting its internal structure [5]. The testing process includes several aspects, such as:

1. Functionality Testing – Ensuring that the application's key features operate as designed.
2. Integration Testing – Verifying that each system module works correctly when integrated.
3. Security Testing – Assessing the system's security against potential threats or risks.
4. Performance Testing – Measuring the speed and efficiency of the system in handling user requests.
5. Usability Testing – Evaluating the ease of use based on user experience.

G. Deployment and Evaluation

The deployment and evaluation stage are conducted to gather user feedback on the application's performance and usability. The evaluation follows the Technology Acceptance Model (TAM) to understand the extent to which users accept and adopt the application [4]. This evaluation focuses on three main factors:

1. Perceived Usefulness – The extent to which users find the application beneficial in their activities.
2. Perceived Ease of Use – The level of ease experienced by users when operating the application.
3. Behavioral Intention – The likelihood of users continuing to use the application in the future.

H. System Maintenance

The maintenance stage is carried out periodically to ensure the application continues to function properly and adapts to environmental and user needs. System maintenance includes:

1. Bug Fixing – Identifying and correcting errors or malfunctions found after deployment.
2. Feature Enhancements – Adjusting and developing application features to remain relevant to user needs.
3. Performance Monitoring – Observing system performance and optimizing it when necessary.

III. RESULTS AND DISCUSSIONS

A. Program Design

1. Web Admin

Web Admin is a web-based platform used by administrators to manage the Nahdlatut Turots system. It allows admins to manage kitab data, upload manuscripts, as well as edit metadata related to kitab.

a. Admin Login

This feature allows administrators to log into the system using their registered credentials. The login process is done through JWT authentication, which ensures that only admins with authorised access can enter the management dashboard. The login page interface can be seen in Figure 5.

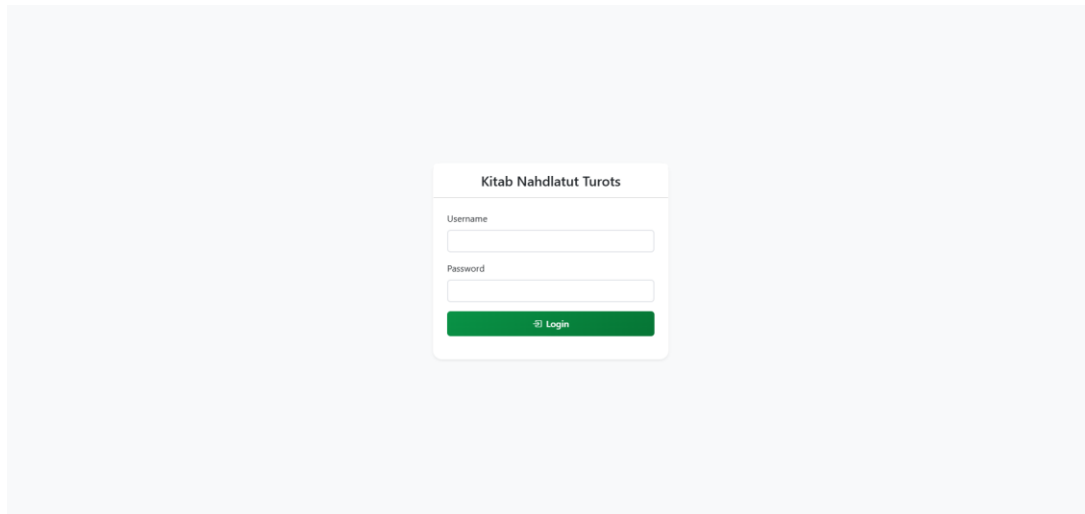


Figure 5 Admin Login

b. Book Management

This module provides features to add, edit, and delete kitab available in the system. Admins can upload kitab manuscripts in PDF format and update metadata, such as title, author, year of publication, and kitab category. The display of the book management feature can be seen in Figure 7.

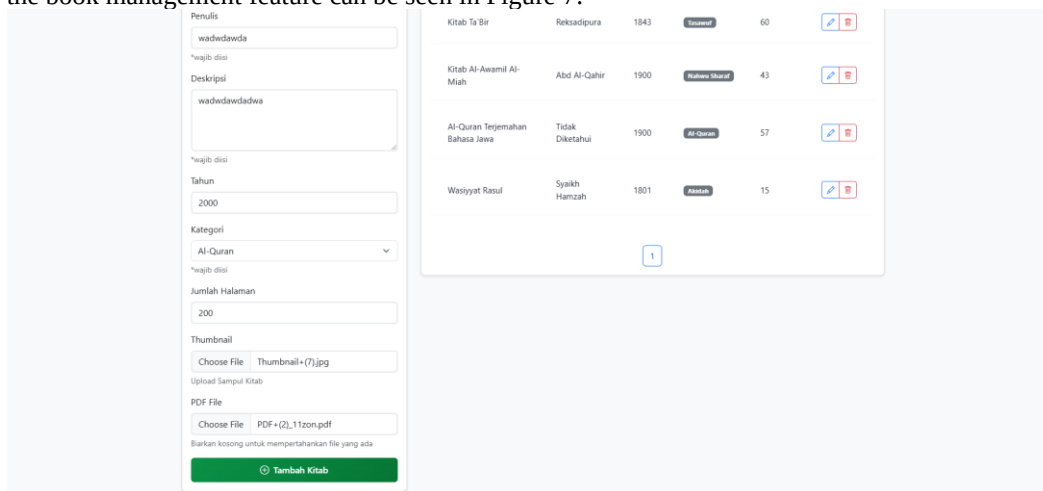


Figure 6 Book Management

2. User Mobile App

An Android-based mobile application was designed to facilitate users in accessing and reading the book of turots digitally.

a. User Sign Up

Users can register for a new account by entering basic information such as name, email, and password. After successful registration, the user will gain access to the features in the application. The interface for the user registration process can be seen in Figure 8.

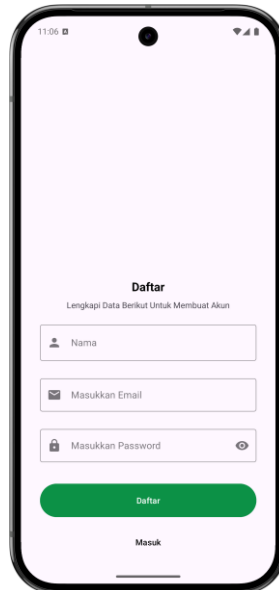


Figure 7 User Sign Up

b. User Sign In

Registered users can login to the application by entering their email and password. The system will validate using Firebase Authentication based on JWT (JSON Web Token) before allowing access to the book content. Firebase Authentication allows users to login with email and password securely and supports integration with other authentication methods, such as Google Sign-In [15]. Upon successful login, Firebase will generate an authentication token that is used to access backend services. The user login page interface can be seen in Figure 9.

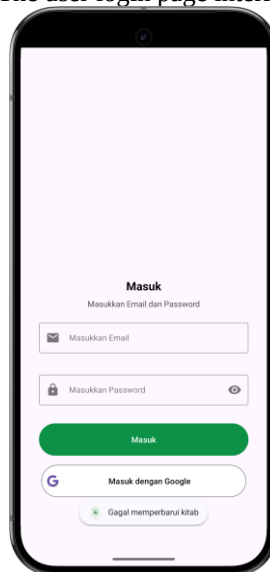


Figure 8 User Sign In

c. Read book

This feature allows users to read the kitab in digital format. The kitab are displayed in the form of PDF files stored in Google Cloud Storage and can be accessed directly through the app. When the user selects a kitab to read, the app will retrieve the file URL from Google Cloud SQL, then display the PDF using the PDF viewer integrated into the app. To enhance security and prevent unauthorized distribution, the application implements screenshot protection, restricting users from capturing screenshots while viewing the kitab. The display of the book reading feature can be seen in Figure 10.



Figure 9 Read Book

d. Search book

This search feature allows users to find books by title, author, or category. With this feature, users can access the kitab they need more quickly. The interface of the book search can be seen in Figure 11.

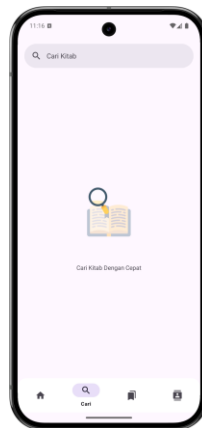


Figure 10 Search Book

e. Add Bookmark

This feature allows users to mark certain books as bookmarks, so that frequently accessed books can be easily found at a later date. These bookmarks are stored locally on the user's device. The display of the bookmark feature can be seen in Figure 12.

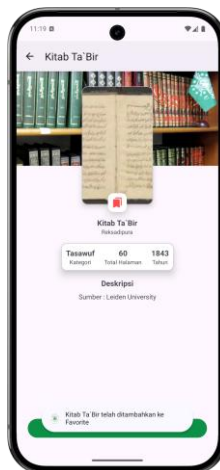


Figure 11 Add bookmark

B. Testing

Based on the results of black-box testing, the system has been verified to function as expected. Testing ensures that each feature that has been implemented works well according to predetermined needs. Detailed test results can be seen in Table 1 for the admin system and Table 2 for the user system.

Table 1 Admin System

No	Test Scenario	Expected Result	Test Result
1.	Login with valid credentials	Login successfully and enter the admin dashboard	Successful
2.	Login with incorrect username and password	The message 'Username and Password are incorrect' appears	Successful
3.	Login with empty or invalid username and password	'Invalid data' message appears	Successful
4.	Upload kitab data with invalid file format	The message 'File format not supported (PDF only)' appears	Successful
5.	Upload kitab data without filling in the title	Appears message 'Complete Form' Successful	Successful
6.	Upload kitab data without filling in the author	Appears message 'Complete Form' Successful	Successful
7.	Upload kitab data without filling in the category	Appears message 'Complete Form' Successful	Successful
8.	Edit book title	The message "Book successfully updated!" appears	Successful
9.	Edit book file with invalid format	The message "Error" appears	Successful
10.	Edit book description without filling in author	The message "Complete Form" appears	Successful
11.	Edit book description without filling in title	The message "Complete Form" appears	Successful
12.	Admin selects book and confirms deletion	Book data successfully deleted from list	Successful
13.	Admin selects book but cancels deletion	Book data remains in list	Successful

Table 2 User System

No	Test Scenario	Expected Result	Test Result
1.	Registration with complete data	Account successfully created, and the user is redirected to the login page	Successful
2.	Registration with an already registered email	A message appears: "Email is already in use."	Successful
3.	Registration without filling in all required fields	A message appears in the form: "Please complete all required fields."	Successful
4.	Registration with a password shorter than 8 characters	A message appears: "Password must be at least 8 characters long."	Successful
5.	Registration with an incorrect email format	A message appears: "Invalid email format."	Successful
6.	Login with a valid email and password	The user successfully logs in and is redirected to the homepage.	Successful
7.	Login with an unregistered email	A message appears: "Account not found."	Successful
8.	Login with an incorrect password	A message appears: "Incorrect password."	Successful

No	Test Scenario	Expected Result	Test Result
9.	Login with an invalid email format	A message appears: "Invalid email format."	Successful
10.	Login without entering an email and password	A message appears: "Please enter email and password."	Successful
11.	Login using Sign in with Google	The user successfully logs in with a Google account and is redirected to the homepage.	Successful
12.	Sign in with Google is canceled	The user remains on the login page without accessing the application.	Successful
13.	Opening a book without an internet connection	A message appears: "Internet connection is required."	Successful
14.	Searching for a specific book in the collection	The searched book appears and can be read.	Successful
15.	Reading a book with page navigation options	Pages can be navigated smoothly without issues.	Successful
16.	Adding a bookmark on a book page	The system saves the page as a bookmark and displays a success notification.	Successful
17.	Viewing the bookmark list	The system displays a list of bookmarked pages in the correct order.	Successful
18.	Deleting a specific bookmark	The system removes the bookmark from the list and displays a success notification.	Successful

3. **System Evaluation Model**

The acceptance test of the Nahdlatut Turots application was conducted using the Technology Acceptance Model (TAM) to evaluate the benefits, ease of use, and user intention in using it. Data collection was conducted through a questionnaire based on three main dimensions of TAM: Perceived Usefulness (PU), Perceived Ease of Use (PEU), and Behavioral Intention to Use (BI). The results of the questionnaire analysis were used to determine the level of user acceptance of the application. The questionnaire results obtained are presented in Table 3, Table 4, and Table 5, each of which contains questions related to Perceived Usefulness (PU), Perceived Ease of Use (PEU), and Behavioral Intention to Use (BI). Meanwhile, a summary of the questionnaire evaluation results can be seen in Table 6.

Table 3 Perceived Usefulness (PU)

No	Question
1.	Does the Nahdlatut Turots application help you access manuscript books more efficiently?
2.	Does this application make it easier for you to study turots books?
3.	To what extent does this application improve your productivity in finding references for turots books?

Table 4 Perceived Ease of Use

No	Question
1.	Does this application contribute positively to your understanding of turots books?
2.	Are the features in the Nahdlatut Turots application easy to understand without much assistance?
3.	Is the application's navigation intuitive and not confusing?
4.	Do you find it easy to master all the features available in this application?

Table 5 Behavioral Intention to Use

No	Question
1.	Can this application be used without requiring much effort or training?
2.	Do you intend to continue using the Nahdlatut Turots application in the future?
3.	How likely are you to recommend this application to others who also study turots books?

Table 6 Questionnaire Evaluation Results

Age	Category	Score										Average
		PU 1	PU 2	PU 3	PEU 1	PEU 2	PEU 3	PEU 4	BI 1	BI 2	BI 3	
23	General Public	3	5	2	3	4	4	4	4	4	4	3,7
26	General Public	3	3	3	3	3	3	3	3	5	5	3,4
20	Student	3	3	3	3	3	3	3	3	3	3	3
19	Student	3	5	4	5	2	5	2	5	5	5	4,1
19	Student	3	4	3	3	3	4	3	3	4	3	3,3
20	Student	3	3	3	3	3	3	3	3	3	3	3
23	Student	3	4	3	4	3	4	3	3	4	4	3,5
24	General Public	4	5	4	5	4	4	5	5	5	2	4,3
24	Student	4	5	3	4	4	4	5	4	4	4	4,1
22	General Public	4	4	4	3	3	3	3	3	3	3	3,3
21	Student	4	4	3	4	4	4	3	3	3	4	3,6
20	Student	4	4	3	4	4	3	4	4	4	4	3,8
23	Student	4	3	3	4	3	4	3	3	4	3	3,4
24	General Public	4	4	3	3	4	3	4	3	3	3	3,4
18	Student	4	4	3	4	4	3	5	4	4	5	4
20	Student	4	3	4	4	4	4	4	5	4	5	4,1
17	Student	4	5	3	5	4	3	4	5	4	5	4,2
16	Student	4	3	5	2	4	3	5	5	4	4	3,9
17	Student	4	3	5	3	4	4	5	5	3	4	4
18	General Public	4	4	3	5	3	4	4	4	3	5	3,9
18	Student	4	4	4	4	4	4	4	3	4	4	3,9
24	General Public	5	5	5	5	5	5	5	5	5	5	5
30	General Public	5	5	5	5	5	5	5	5	5	5	5
20	Student	5	5	5	5	5	5	5	5	5	5	5
18	Student	5	4	4	3	3	3	3	4	2	4	3,5
32	General Public	5	5	5	5	5	5	5	5	5	5	5
38	General Public	5	5	5	5	5	5	5	5	5	5	5
20	Student	5	5	4	4	4	4	4	4	4	4	4,2
16	Student	5	5	5	5	4	2	4	3	5	5	4,3
25	General Public	5	4	4	5	4	5	5	4	4	5	4,5
Average												3,98

Based on the questionnaire data, the percentage of each response is calculated using the following formula:

$$\text{Percentage} = \frac{\text{Maximum Scale}}{\text{Average Score}} \times 100\%$$

The calculation results show that the average score for the Perceived Usefulness (PU) dimension is 3.86, equivalent to 77.2%, Perceived Ease of Use (PEU) is 3.84, equivalent to 76.8%, and Behavioral Intention (BI) is 3.94, equivalent to 78.8% out of a maximum scale of 5.

Meanwhile, the overall TAM average is 3.98, equivalent to 79.6%, indicating a high level of user acceptance. With these results, the Nahdlatut Turots application is considered to provide significant benefits, be easy to use, and have great potential for widespread adoption by both primary users and the general public.

IV. CONCLUSIONS AND RECOMMENDATIONS

The development of the Android-based Nahdlatut Turots application has been successfully accomplished, featuring a robust database of turots books. This ensures the authenticity and accuracy of data through an effective verification system. However, challenges remain in obtaining licensing agreements from manuscript owners, which need to be addressed to further expand the available collection.

From a user experience perspective, the application boasts an intuitive and user-friendly interface, making navigation seamless and enhancing the reading experience of kitab turots. Its minimalist yet functional design ensures accessibility for a wide audience, including both novices and academics.

Furthermore, the application's efficient search functionality enables users to locate books swiftly and accurately using specific keywords. To safeguard copyright and prevent unauthorized distribution of digital manuscripts, the app includes a content protection feature that disables screenshots.

Overall, the Nahdlatut Turots application is poised to serve as an effective tool in digitizing and preserving turots books, contributing significantly to their wider accessibility and cultural preservation.

V. REFERENCES

- [1] A. G. Puspita, "Identifikasi dan Preservasi Digital Manuskrip Turots oleh TIM Nahdlatut Turots dan Turots Center Universitas Yudharta di Wilayah Pasuruan," *J. Kaji. Inf. dan Perpust.*, vol. 16, no. 1, pp. 58–77, 2024, doi: 10.21154/pustakaloka.v16i1.7607.
- [2] M. Sampurno, "Nahdlatul Turats: Gerakan Filologi Islam dalam Melacak Jaringan Keilmuan Ulama Nusantara," *Muttaqien; Indones. J. Multidisciplinary Islam. Stud.*, vol. 5, no. 1, pp. 15–28, 2024, doi: 10.52593/mtq.05.1.02.
- [3] S. Pargaonkar, "A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering," *Int. J. Sci. Res. Publ.*, vol. 13, no. 8, pp. 120–124, 2023, doi: 10.29322/ijsrp.13.08.2023.p14015.
- [4] Harryanto, M. Muchran, and A. S. Ahmar, "Application of TAM model to the use of information technology," *Int. J. Eng. Technol.*, vol. 7, no. 2.9 Special Issue 9, pp. 37–40, 2018.
- [5] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012, doi: 10.5121/ijesa.2012.2204.
- [6] S. Challita *et al.*, "A Precise Model for Google Cloud Platform To cite this version : HAL Id : hal-01689659 A Precise Model for Google Cloud Platform," 2018.
- [7] A. Bucko, K. Vishi, B. Krasniqi, and B. Rexha, "Enhancing JWT Authentication and Authorization in Web Applications Based on User Behavior History," *Computers*, vol. 12, no. 4, pp. 1–18, 2023, doi: 10.3390/computers12040078.
- [8] I. P. A. E. Pratama and I. M. S. Raharja, "Node.js Performance Benchmarking and Analysis at Virtualbox, Docker, and Podman Environment Using Node-Bench Method," *Int. J. Informatics Vis.*, vol. 7, no. 4, pp. 2240–2246, 2023, doi: 10.30630/joiv.7.4.1762.
- [9] S. Ramdany, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web," *J. Ind. Eng. Syst.*, vol. 5, no. 1, 2024, doi: 10.31599/2e9afp31.
- [10] R. Fauzan, D. Siahaan, S. Rochimah, and E. Triandini, "Class Diagram Similarity Measurement: A Different Approach," pp. 215–219, 2019, doi: 10.1109/icitisee.2018.8721021.
- [11] M. Persson, "JavaScript DOM Manipulation Performance : Comparing Vanilla JavaScript and Leading JavaScript Front-end Frameworks," vol. Independen, p. 40, 2020, [Online]. Available: <http://bth.diva-portal.org/smash/get/diva2:1436661/FULLTEXT01.pdf%0Ahttp://urn.kb.se/resolve?urn=urn:nbn:se:bth-19531>
- [12] G. Munawar, B. Wisnuadhi, and N. Sibarani Sumanda, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," *9th Ind. Res. Work. Natl. Semin.*, no. December, 2018, [Online]. Available: <https://www.researchgate.net/publication/329525878>
- [13] Y. M. Putra, U. M. Buana, R. J. Putri, and U. M. Buana, "Sistem Informasi Akuntansi Pengaplikasian Dan Implementasi Konsep Basis Data Relasional Pada Sistem Pendapatan Dan Pengeluaran," no. June, p. 1, 2020.
- [14] S. Widyaningtyas and T. Wahyono, "Implementasi Rest Api Menggunakan Retrofit Pada Aplikasi Monitoring Grooming Berbasis Android," *IT-Explore J. Penerapan Teknol. Inf. dan Komun.*, vol. 3, no. 2, pp. 147–161, 2024, doi: 10.24246/itexplore.v3i2.2024.pp147-161.
- [15] A. T. Gaikwad, "Firebase - Overview and usage," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 3(12), no. 12, pp. 1178–1183, 2022.