

---

## Development of an Android Based Personal Financial Management Application for Financial Monitoring and Student Expense Notifications

Rizki Alfa Reza<sup>1\*</sup>, Yuyun Khairunisa<sup>2</sup>, Nur Rahmansyah<sup>3</sup>

<sup>1,2,3</sup> State Polytechnic of Creative Media, Multimedia Engineering Technology, Jl. Srengseng Sawah No.17, RT.8/RW.3, Srengseng Sawah, Jagakarsa District, South Jakarta City, Special Capital Region of Jakarta 12630, Indonesia

---

### Keywords

Finance Application; Firebase; Mobile Application

**\*Corresponding Author:**  
[Rizkireza63@gmail.com](mailto:Rizkireza63@gmail.com)

### Abstract

This development was carried out because many students still experience difficulties in managing their finances, resulting in waste and uncontrolled financial conditions. This is usually caused by a lack of awareness in financial recording and the absence of adequate tools to monitor income and expenses regularly. As a result, students often experience difficulties in meeting daily needs or sudden needs. The purpose of this research is to develop an Android-based personal financial management application to help students manage their finances better, in a structured, and efficient manner. With this application, it is hoped that students can make financial recording easier and more structured. The research was conducted by collecting data through observation and interviews to identify user needs in managing finances and understand the problems often faced by students in daily financial activities. The method used in this development is the Agile method, using the Scrum framework, which allows the development process to be carried out in stages and flexibly so that it can adapt to user needs. This application was developed on the Android platform by utilizing Firebase Authentication which is used as a security system to maintain the confidentiality of user data, while the Firebase Cloud Firestore service helps in data storage so as to reduce the risk of data loss. Black Box Testing showed that all application features functioned successfully according to system requirements. Based on usability testing using the System Usability Scale (SUS) involving 22 respondents, the application obtained a score of 79.09 which indicates that the application has good usability and is acceptable to users. Based on these results, the developed application can help students better manage their finances.

---

### 1. Introduction

Personal financial management is essential in daily life, especially for college students. Good financial management plays a crucial role in helping students maintain financial stability throughout their college years. [1] In addition, many students still struggle to manage their finances due to a lack of record-keeping and ineffective budgeting. Several factors influence students' financial situations, including environmental

factors, needs, and lifestyle choices, which often lead students to spend recklessly without considering their financial circumstances. This leads to uncontrolled spending and makes it difficult to assess students' spending habits, resulting in wasteful spending. Good financial management can help students achieve their goals in meeting their basic needs. Meanwhile, poor financial management has negative consequences that can lead students to face financial problems such as loans and debt [1].

According to the latest findings from a survey conducted by the OJK in 2025, financial literacy among the Indonesian public stands at 66.64%, while financial access currently stands at 80.51% [2]. The aforementioned data demonstrates that financial services are expanding, but financial literacy remains suboptimal. This impacts young people, who, even if familiar with digital financial applications, may not possess the skills to manage their finances effectively. College students, on the other hand, have poor financial management skills, a problem often evident in poor habits such as impulsive spending and infrequently recording expenses and saving [3]. Many students lack the skills to create a budget. They often don't record income and expenses, making it difficult to monitor their finances and identify excessive spending. Consequently, students struggle to control their finances, leading to waste [4]. Financial management is essential for students to become more financially independent. With a financial recording app, students can clearly see their income and expenses, making it easier to understand their financial situation. Understanding the importance of financial management is crucial, as it's a reality everyone faces in life [5].

Previous studies have developed various Android-based personal financial management applications to simplify financial management through mobile devices. These applications provide users with convenient tools for recording and monitoring financial transactions. However, most existing applications still rely on local data storage using SQLite, which restricts data availability to a single device and increases the risk of data loss when the device is reset, replaced, or damaged [6]. This limitation highlights the need for a more reliable cloud-based data storage solution. To address this gap, the present study integrates Firebase Cloud Firestore as the primary database to enable secure cloud storage, real-time data synchronization, and cross-device accessibility, thereby reducing the risk of data loss while improving data availability and reliability.

Based on further research, community service activities through training on designing a prototype for an Android personal financial management application were successfully implemented in accordance with the activity's objectives. This training was able to improve participants' understanding of personal financial management and the use of digital technology as a means of financial recording and evaluation. Through a prototype design approach, participants were able to understand the application's usage flow and the interrelationships between features more systematically. The results of the activity showed that participants were able to produce an application prototype design that included budgeting features, transaction tagging, asset management, and automatic monthly financial reports. The resulting prototype had a clear flow and a simple interface, thus potentially becoming the basis for developing an easy-to-use personal financial management application. Thus, this community service activity made a positive contribution to improving digital financial literacy and technology utilization skills in the community [7]. In this research, there are limitations, in this research it is still in the form of a prototype that has not been developed in the form of an application, therefore in this development, the focus will be on developing the application as a whole and implementing its features, so that it can be used by users.

Table 1. Research Gap

Research	Limitations	Novelty
Android Based Personal Finance Manager Application Design[6].	Only uses local storage (SQLite), making data vulnerable to loss.	Firestore Cloud Firestore, Firebase Authentication
Android Application Prototype Design Training for Personal Financial Management with Budgeting, Tagging, Assets and Automated Monthly Reports Features [7]	In this research it is still in the form of a prototype that has not been developed in the form of an application	develop the application as a whole and implement its features, so that it can be used by users.

## 2. Research Method

This research uses the Research and Development (R&D) method. In its development stage, this research uses the agile method. Agile. In Sugiyono's view, it is stated that R&D is an applicable method. This research not only stops at the theoretical stage, but continues to produce a new product which is then tested to prove its benefits and effectiveness [8]. Agile Development is a software development method whose process is gradual and allows developers to adapt to various changes [9]. This study employs a Research and Development (R&D) approach using the AGILE methodology. The AGILE methodology is a software development approach that involves incremental, phased development and prioritizes flexibility in adapting to changes. This method allows for an incremental development process, where implemented features are adapted based on user needs or user feedback, then tested and refined [10].

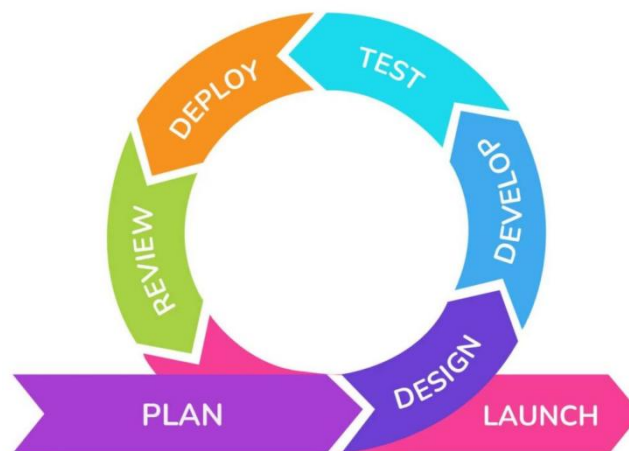


Figure 1. Agile Method

Source : [11]

The AGILE method is an approach to software development that is carried out quickly and incrementally. It emphasizes communication and the ability to adapt to changes during the development process. The phases of the AGILE method include Planning, Design, Development, Testing, Deployment and Review.

### 2.1 Planning

System planning is the initial stage in development using the AGILE methodology. At this stage, developers and users collaborate to create a design tailored to user needs. Activities at this stage include observation, interviews, and reviewing existing documentation [12].

## 2.2 Design

The design phase is carried out in stages and on an ongoing basis to produce features that meet user needs and are ready for testing. User interface (UI) and user experience (UX) design is conducted with consideration of user input obtained from feedback[13]. This prototype can be used to get feedback from users regarding the layout, functionality, and user experience of applications that do not yet meet user needs.[11]

## 2.3 Development

The development phase is the process of creating the application currently under development based on the previously designed application. This phase involves creating application features and managing the database to ensure the system runs smoothly and meets user needs. In the development process, code creation or coding, system integration, and initial testing are carried out to ensure that each part of the system functions properly and meets user needs[11].

## 2.4 Testing

The testing phase is the process of ensuring that the system in the developed application functions properly according to user needs. At this stage, application features are tested to determine whether there are errors in the application system. Testing in this research uses the Blackbox Testing and System Usability Scale (SUS) methods. The Blackbox Testing method aims to find problems with the application system, such as application malfunctions and features that do not function properly [14].

## 2.5 Deployment

The deployment phase is the process during which an application is rolled out and begins to be used by users. In Agile methodologies, deployment is an iterative process. Feedback from users following initial use is crucial for refining and improving the system in subsequent development cycles [15].

## 2.6 Review

During this evaluation phase, developers gather user feedback. This feedback is used to identify any deficiencies or issues with the application. The evaluation results then become the basis for improvements and further development in the next phase.

This research uses the Agile development method by applying the Scrum framework in the application development process. Using a scrum framework can help the software development process gradually so that it is easier to adjust user needs during the development process. Development of this application was carried out in 3 sprints, where each sprint lasted for 2 weeks, so the total development time was 6 weeks. The distribution of sprints in developing this application is as follows:

Table 2. Sprint

Sprint	Activities	Duration
Sprint 1	User needs analysis and identification of application features based on user requirements	2 weeks
Sprint 2	Development of core application features.	2 weeks
Sprint 3	Implementation of Firebase cloud synchronization features and Notification.	2 weeks

## 3. Result and Discussions

### 3.1 Use Case Diagram

Use case diagrams facilitate understanding of the interactions between users and the system and define the features that will be included on the website. Use case diagrams enable developers to identify user needs, clearly communicate requirements, and guide system development toward user needs and expectations [16]. Use case scenarios are described in textual form depending on the format requirements, namely informal, brief, and complete [17]. A use case diagram provides an overview of the system and helps clarify the roles of

each actor in operating the designed system. With this diagram, the system's functional flow can be fully understood before the implementation process begins [18].

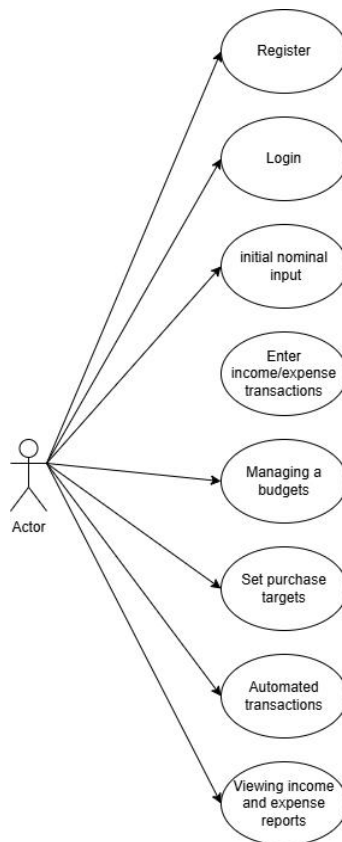


Figure 2. Use Case Diagram

In Figure 2. This diagram illustrates the interaction of one chord with the system, where users can carry out activities such as registration, login, enter initial balances, record income and expenditure transactions, manage budgets, determine purchase targets, view routine bills, and can view income and expenditure reports that have been made by the user.

### 3.2 Activity Diagram

An activity diagram is a depiction of the flow of activities in the form of a series of actions, how the system starts and runs, until the final process of the system [19]. An Activity Diagram is a type of diagram in the Unified Modeling Language (UML) used to depict workflows or processes within a system. This diagram illustrates the sequence of activities, actions, and decisions that occur from the beginning to the end of a process [20]. Activity diagrams help clarify how the interaction between the user and the system takes place, and how the system processes data from start to finish, producing output in the form of information [21].

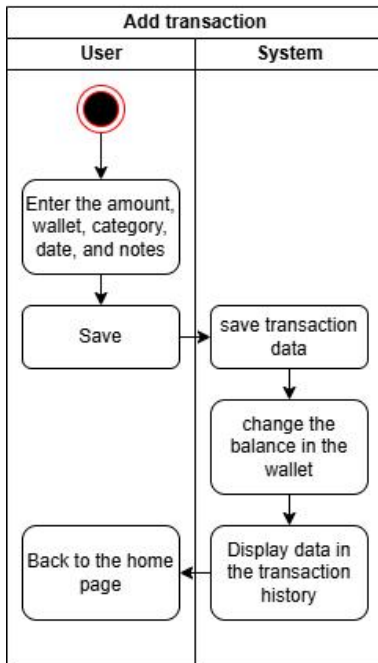


Figure 3. Transaction Addition Activity Diagram

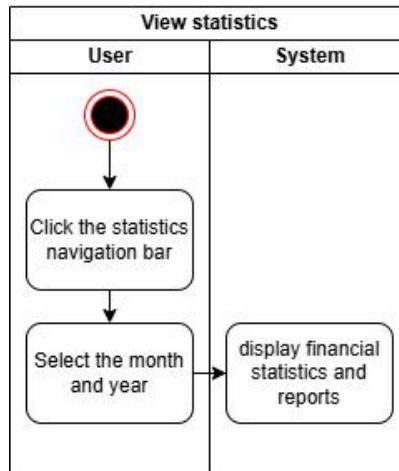


Figure 4. Activity Diagram View Report

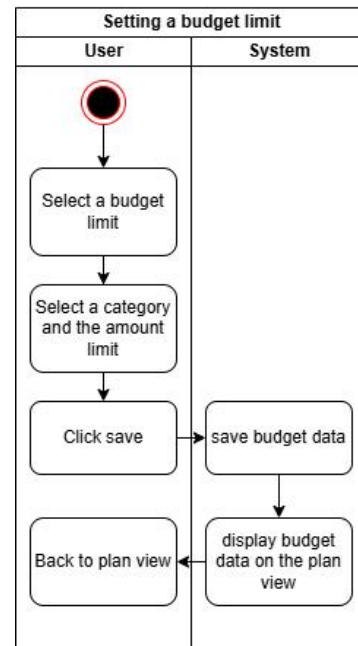


Figure 5. Activity Diagram Setting Budget Limit

Figure 3 illustrates the user's flow for recording financial expenses. The user enters the expense amount, selects a category, and adds notes if necessary. After pressing the save button, the system saves the expense data and displays a list of expenses on the transaction history page. Figure 4 shows the user's process of viewing financial report data to determine their income and expenses. The user selects the statistics navigation menu button, then selects the month and year for which they wish to view data. The system then displays the report data for the specified month and year. Figure 5 for setting budget limits explains the steps a user takes to set a spending limit. The user opens the planning navigation button and then selects the manage budget feature. Next, the user selects a category and enters the budget limit amount. Once completed, the user clicks the save button. The system saves the budget data and displays it on the planning page. The system also notifies the user if spending exceeds the specified budget limit.

Figure 6 explains the user's flow in setting a purchase target and saving for the target they wish to purchase. The process begins when the user opens the planning menu and selects the purchase target feature. The user then fills in the purchase target data, such as the item name, target price, and desired purchase time. After the target data is saved, the system will save and display the purchase target on the planning page. After that, the user can save by selecting the purchase target they have created and then entering the amount to be saved. After the user presses the save button, the system will record the savings amount and update the progress of the purchase target achievement. This process can be repeated until the purchase target is reached. All target and savings data will be displayed on the planning page so that users can easily monitor their savings progress. Figure 7 illustrates the user flow in managing and paying monthly recurring bills in a financial management application. The process begins when the user opens the planning navigation menu and selects the recurring billing feature. The user then enters billing data such as bill name, bill amount, and due date. After the billing data is saved, the system will save and display the recurring billing information on the planning page. When the due date approaches, the system will display a reminder notification to the user. The user can select the bill to be paid via the recurring billing card on the planning page and press the pay button on the monthly recurring billing card. The system will automatically deduct the amount from the user's specified wallet balance, which will then display it as an expense in the transaction history.

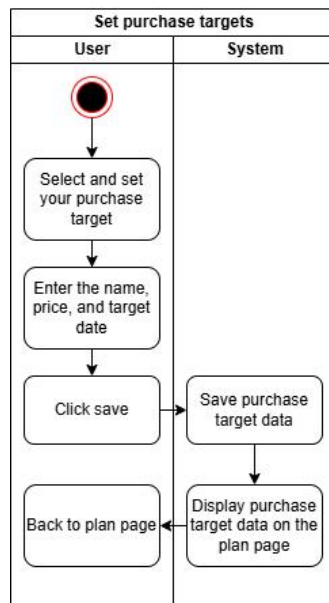


Figure 6. Purchase Target Activity Diagram

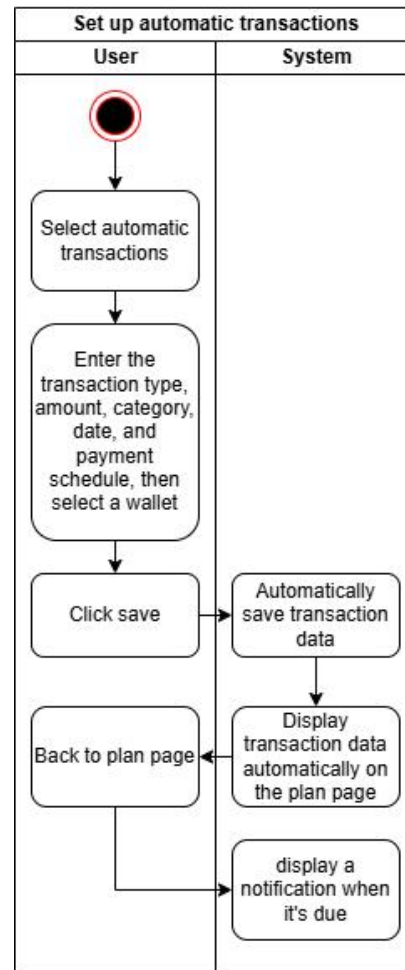


Figure 7. Activity diagram for automated transactions

### 3.3 Application Display

Display add income, in Figure 8, the user can add income transactions, first the user can enter the nominal amount to be recorded, then the user selects the source of funds or wallet, after that the user selects the income category such as salary, scholarship, pocket money, then the user can set the income date and can also add notes if necessary. Figure 9 view displays a list of all incoming and outgoing transactions. On this page, users can set the transaction history date, filter the wallets and categories they wish to view, and filter by incoming, outgoing, or all transactions. Figure 10 view displays several features such as setting and viewing the progress of budget limits where users can set limits on monthly expenses, if the user approaches the budget limit or exceeds the budget limit then the user will get a notification, after that setting and viewing the progress of the purchase target this feature can make it easier for users to save according to what the user wants, in this feature there is also progress if the user wants to save so the user can know how much savings have been collected, as well as setting scheduled transactions this feature makes it easier for users to not need to record income and expenses anymore, the system will automatically record according to the specified date, in this feature there is also an automatic or manual option.

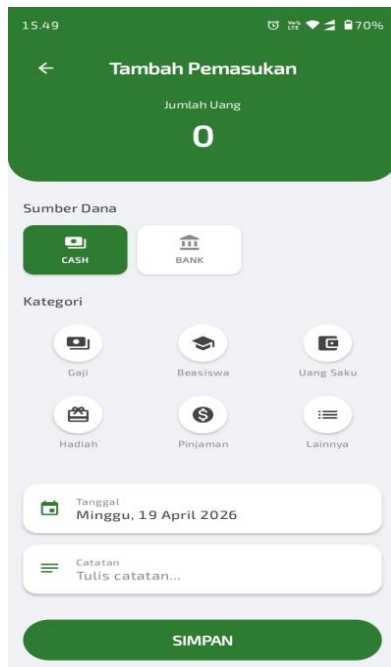


Figure 8. Add Income View

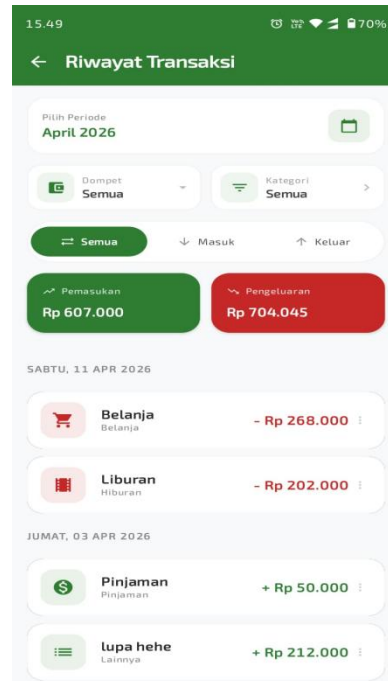


Figure 9. Transaction History

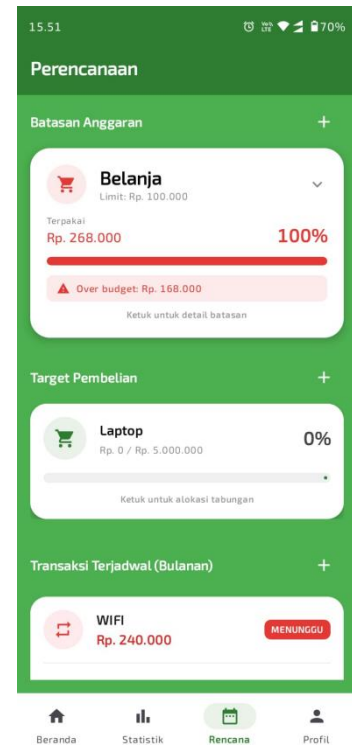


Figure 10. Plan View

### 3.4 Blackbox Testing

Black Box testing is a testing method that only focuses on the outside of the software, especially on existing functions according to system needs and specifications. Testing is carried out without looking at the design or code of the application program. The main focus on this method is checking input, output, and ensuring features can run properly [22]. The Black Box Testing method is carried out to find errors in the software so that they can be prevented or corrected if errors occur in the system [23].

This finding aligns with the broader purpose of Black Box Testing, which is to surface malfunctions and non-functioning features through input-output checks rather than code-level inspection [22], [23]. All twenty test scenarios in the present study, spanning authentication, transaction handling, budget limits, scheduled transactions, and offline-to-online synchronization, returned a Passed status, indicating that the feature set behaves as specified across both normal and edge-case conditions. A comparable Black Box evaluation of a learning management application also reported that every core feature ran according to specification, yet the corresponding usability score for that system fell into the marginal range [24]. This contrast is instructive: functional correctness and perceived usability are related but not interchangeable outcomes, and a system can pass every functional test while still frustrating its users. The present application's combination of a fully passed Black Box test with a SUS score well above the marginal range suggests that the functional reliability achieved in development carried through into an interface users actually found easy to use, rather than the two outcomes diverging as they did in the comparison case.

Table 3. Blackbox Testing

Test Type	Testing Steps	Expected results	Status
Login with valid account	<ol style="list-style-type: none"> <li>1. Open the application</li> <li>2. The user enters email and password</li> <li>3. Press the login button</li> </ol>	The system successfully verified the user and directed them to the home page.	Passed

Test Type	Testing Steps	Expected results	Status
login with the wrong email and password	<ol style="list-style-type: none"> <li>1. Open the application</li> <li>2. the user entered the wrong email and password</li> </ol>	the system will verify whether the account is registered, otherwise there will be a "unregistered account" notification	Passed
Add Transaction	<ol style="list-style-type: none"> <li>1. User selects income/expenses</li> <li>2. Write the nominal, wallet, category, date, notes</li> <li>3. Press the save button</li> </ol>	The system saves the data, and displays it on the transaction history page.	Passed
adding transactions with insufficient balance	<ol style="list-style-type: none"> <li>1. Users choose revenue/expenditure</li> <li>2. Write the nominal amount beyond the contents of the wallet, select wallet, category, date, note</li> <li>3. Press the save button</li> </ol>	the system will check whether the current balance is sufficient to carry out the transaction, otherwise there will be insufficient balance notification	Passed
Edit and Delete Transactions	<ol style="list-style-type: none"> <li>1. Press the transaction history button</li> <li>2. Then swipe left or right to edit and delete transaction history</li> </ol>	The system will automatically delete or go to the transaction edit page.	Passed
Edit/Delete Confirmation Popup	<ol style="list-style-type: none"> <li>1. Press the transaction history button</li> <li>2. Then swipe left or right to edit and delete transaction history</li> </ol>	the system will display a confirmation notification whether the user wants to delete/edit	Passed
Setting Budget limit	<ol style="list-style-type: none"> <li>1. Press plan navigation</li> <li>2. Add budget limit</li> <li>3. Enter category and nominal</li> <li>4. Save</li> </ol>	The system will automatically add progress to the spending limit when the user makes a transaction according to the category.	Passed
setting budget limits without writing down nominal	Users enter categories without entering nominal	the system will double check whether there are any parts that have not been filled in, if there are any, an alert will appear that the nominal must be filled in	Passed
View Statistics	<ol style="list-style-type: none"> <li>1. User adds transactions with different categories</li> <li>2. Press statistics navigation</li> </ol>	The system will display a summary of income and expenses, such as the total nominal income and expenses and what categories they fall into.	Passed
View Statistics with no transaction data	User opens statistics page without transaction history	The system displays blank statistical information and asks the user to make a transaction.	Passed
Transaction History Filter	<ol style="list-style-type: none"> <li>1. Enter the transaction history page</li> <li>2. Select filters such as date, category filter, wallet filter, and transaction type filter</li> </ol>	The system will automatically filter according to what the user specifies, and display transaction history according to the specified filter.	Passed

Test Type	Testing Steps	Expected results	Status
Transaction History Filter with No Transactions in a Specific Category	The user selects a transaction category that does not contain any transaction data.	The system displays information that there has been no transaction.	Passed
Set Purchase Targets	1. Press the plan navigation 2. Add a purchase target 3. Enter the item name, nominal and date 4. Save 5. Press the tube button	The system will automatically add savings progress if the user presses the save button.	Passed
set a purchase target by emptying one of the columns	Users enter data without entering the nominal item	The system displays the warning message "All fields must be filled in" and data is not stored.	Passed
Manage Scheduled Transactions	1. Press plan navigation 2. Add scheduled transaction 3. Fill in the nominal data, description, category, execution date, schedule type, select wallet	The automatic system will add transactions automatically according to the specified date, and can be done manually with one press.	Succeed
Manage Scheduled Transactions without entering a nominal amount	Users fill in the category and date, but don't enter the nominal, then click Save.	The system displays the message "Nominal must not be empty" and prevents storage.	Passed
Auto Sync	Add new transactions when the internet is active.	Data is stored in real time in the Firebase database, and can be used on more than 1 device, to avoid data loss.	Passed
Sync Without Internet	Add transactions when your cellphone is offline.	The system stores data in a local database (SQLite) and when the user is online, the data in the local storage will automatically be stored in a firebase	Passed
Notifications	Users wait for the specified notification at 12 and 9 pm or the user exceeds the budget limit.	The system will display notifications according to the specified hours, and provide notifications if expenses approach the limit.	Passed
does not allow notifications in the application	Turn off app notification permissions via Android settings, then wait for the notification reminder schedule.	The system still performs tasks in the background, even if notifications do not appear.	Passed

### 3.5 System Usability Scale (SUS) Testing

One method commonly used in measuring usability is the System Usability Scale (SUS), which is a questionnaire consisting of ten statements using a likert scale but can provide a quantitative picture of the user experience. Apart from that, usability evaluation has an important role in improving the quality of the learning process because the measurement results can show the level of effectiveness of the system in

supporting the achievement of goals [24]. System Usability Scale (SUS) is a method for measuring the level of ease of use of an application quickly and simply. SUS consists of 10 questions that users answer using a Likert scale from 1 (strongly disagree) to 5 (strongly agree). The resulting value from SUS is used to find out how good the usability is in the application and can be compared with other usability standards. The advantages of the SUS method are that it is easy to use, easy to understand, and helps in finding parts of the application that still need improvement [25].

Application testing was carried out using the System Usability Scale (SUS) method involving 22 respondents. This test aims to find out whether young applications are used and acceptable to users.

SUS score calculations are carried out using the rules:

Odd numbered questions are calculated using the formula :

$$X - 1 \quad (1)$$

Even numbered questions are calculated using the formula :

$$5 - X \quad (2)$$

All scores are then added and multiplied by 2.5 using the formula :

$$SUS = (\sum \text{score}) \times 2.5 \quad (3)$$

After the score of each respondent is known, the next step is to find the average score by adding up all the score results and dividing by the number of respondents present. This calculation can be seen with the following formula :

$$X = \sum x / n \quad (4)$$

Description:

X: Average score

$\sum x$ : Number of system usability scale scores

N: Number of respondents

Table 4. System Usability Scale Testing

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jml	Skor = (Jml x 2,5)
R1	4	2	5	2	4	1	5	1	4	2	34	85
R2	5	1	5	1	5	2	5	1	4	1	38	95
R3	5	4	4	5	5	4	3	4	5	4	21	52,5
R4	4	2	3	1	4	2	3	2	4	5	26	65
R5	5	1	5	1	5	1	5	1	5	1	40	100
R6	5	1	5	1	5	1	5	1	5	1	40	100
R7	4	2	4	1	4	2	5	1	5	1	35	87,5
R8	4	2	4	3	4	2	4	3	4	1	29	72,5
R9	4	2	4	2	4	2	4	2	4	2	30	75
R10	4	3	4	3	4	3	3	3	3	3	25	62,5
R11	5	2	4	2	4	2	4	1	4	2	32	80
R12	4	3	4	2	4	3	3	2	3	3	26	65
R13	5	1	4	2	5	4	4	1	4	2	29	72,5
R14	5	1	5	1	5	1	5	1	5	1	40	100
R15	4	3	3	3	4	2	4	2	4	2	27	67,5
R16	4	2	5	2	5	1	5	2	5	1	36	90
R17	1	1	5	4	5	2	5	1	5	5	22	55

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jml	Skor = (Jml x 2,5)
R18	5	2	4	2	5	2	4	2	5	4	30	75
R19	4	2	4	1	4	1	4	2	5	1	35	87,5
R20	5	2	5	1	4	2	5	2	5	3	32	80
R21	5	2	4	2	5	2	4	2	4	3	29	72,5
R22	4	2	4	2	5	3	4	2	4	2	27	67,5
Total											696	1740
Average SUS												79,09

Application testing was carried out using the System Usability Scale (SUS) method involving 22 respondents. This test aims to find out whether the application is easy to use and acceptable to users. Based on the results of the tests that have been carried out, the application obtained a SUS score of 79.09. These results show that the application is considered to be fairly easy to use, and well understood, and acceptable to the user.

This SUS score places the application within the "good" usability range, a result consistent with the gap this study set out to close. Earlier Android-based personal finance applications relied on local SQLite storage, a structural weakness that limits cross-device access and raises the risk of data loss [6]. The present application instead uses Firebase Cloud Firestore for real-time synchronization, and the resulting usability score suggests that this architectural shift did not come at the cost of ease of use. The finding also extends an earlier community service program that produced a budgeting and reporting prototype but stopped short of a working, user-tested system [7]. By carrying that prototype concept through to a fully implemented and field-tested application, this study demonstrates that the budgeting, tagging, and automated reporting features envisioned at the design stage remain usable once built, addressing the very limitation the earlier work acknowledged.

Viewed against broader usability benchmarks, the application's performance is more than adequate rather than merely passable. Lewis's review of the System Usability Scale reports that across hundreds of studies the average SUS score is 68 [26], meaning the developed application scores notably above the typical system. Kortum and Sorber's comparison of mobile applications on phone and tablet platforms found a mean of 77.7 among the ten most usable apps tested [27], a figure the present application also surpasses. This comparison matters because student-facing financial tools compete for attention with applications users already consider intuitive; a SUS score below this range would have signaled that the cloud-based architecture introduced friction the prototype-stage designs in [6] and [7] never had to confront. The combination of a fully passed Black Box test and an above-benchmark SUS score instead indicates that the added complexity of authentication, encryption, and real-time synchronization was absorbed without degrading the user experience, which is the central risk such an architectural upgrade carries.

### 3.6 User Data Security

The security of users' financial data in this application is maintained with a multi-layered protection system. For the login process, the application uses Firebase Authentication so that only users who have an account can access the system. All financial data sent from the user's device to the server has been secured using the HTTPS/TLS (Encryption in Transit) protocol, so that important information is not easily intercepted by other parties. In addition, data is stored securely in Firebase Firestore which automatically uses AES-256 (Encryption at Rest) encryption. Data privacy between users is also maintained through the use of unique UUIDs checked by Firebase Security Rules. This system works automatically on the server side to deny access if the user ID does not match. That way, each user can only access their own data without the risk of data leaking to other accounts.

#### 4. Conclusions and Future Works

Based on the design and testing results of the study, entitled "Development of an Android-Based Personal Financial Management Application for Financial Monitoring and Notifications," it can be concluded that the application successfully met the research objectives. This application is able to assist users in managing their personal finances, such as recording income and expenses and effectively monitoring their financial condition. Furthermore, the application is equipped with a notification feature that serves as a reminder to record daily financial records so that users do not forget to record their finances, and notifications serve as a reminder to manage their finances to avoid waste. This application also features a Firebase Authentication database feature that serves to maintain user data security and prevent data loss. Based on the results of the Black Box test, all application system functions, starting from the login process as well as financial recording and monitoring features stored in the database, run well without encountering problems with the test process. This shows that the application system developed is in accordance with the user's design and needs. In addition, based on the results of usability testing using the System Usability Scale (SUS) involving 22 respondents, the application obtained an at-grade score of 79.09 which is included in the good category and can be accepted by users. Based on these results, this application can be an effective solution to help users manage finances and make it easier to record income and expenses more regularly.

Even though the application has fulfilled its main function, this research still has several limitations, such as the process of recording transactions which is still carried out manually and the appearance of financial reports which are still simple and in text form. Further development will therefore be focused on the integration of Bank APIs to allow transaction data to be automatically and in real time synchronized into the application. Apart from that, the application will also be developed with a recording feature using photo receipts to make it easier for users to record transactions more quickly and practically. Interactive data visualization features such as expenditure charts and monthly balance trends will also be added to make financial information easier for users to understand.

#### 5. References

- [1] H. Hatimatunnisani, A. Pradipta, W. Zahra, M. Amalia, and L. Putri, "Pengaruh Gaya Hidup Hedonisme Terhadap Pengelolaan Keuangan Mahasiswa Perguruan Tinggi di Bandung," *J. Ekon. Bisnis dan Manaj. EKO-BISMA*, vol. 3, no. 1, pp. 263–268, 2024, doi: 10.58268/eb.v3i1.87.
- [2] Otoritas Jasa Keuangan, "Buku Statistik Survei Nasional Literasi dan Inklusi Keuangan (SNLIK) Tahun 2025," 2025. [Online]. Available: [https://ojk.go.id/id/Fungsi-Utama/Perilaku-Pelaku-Usaha-Jasa-Kuangan/SNLIK/Documents/Buku\\_Statistik\\_Survei\\_Nasional\\_Literasi\\_dan\\_Inklusi\\_Kuangan\\_%28SNLIK%29\\_Tahun\\_2025.pdf](https://ojk.go.id/id/Fungsi-Utama/Perilaku-Pelaku-Usaha-Jasa-Kuangan/SNLIK/Documents/Buku_Statistik_Survei_Nasional_Literasi_dan_Inklusi_Kuangan_%28SNLIK%29_Tahun_2025.pdf)
- [3] C. D. Cristy, A. Eddy, and F. Harimurti, "Financial Attitude dan Financial Literacy terhadap Pengelolaan Keuangan Mahasiswa: Studi Literatur Berbasis Theory of Planned Behavior," 2024.
- [4] S. Addin, A. Hidayat, N. Herawati, and L. D. Warpindyastuti, "Pendidikan Keuangan Untuk Mahasiswa: Tantangan dan Dampaknya terhadap Pengelolaan Keuangan Pribadi," *JIIP - J. Ilm. Ilmu Pendidik.*, vol. 7, no. 12, pp. 14108–14114, 2024, doi: 10.54371/jiip.v7i12.6570.
- [5] J. H. Napitupulu, N. Ellyawati, and R. F. Astuti, "Pengaruh Literasi Keuangan dan Sikap Keuangan Terhadap Perilaku Pengelolaan Keuangan Mahasiswa Kota Samarinda," *J. Pendidik. Ekon. JUPE*, vol. 9, no. 3, pp. 138–144, 2021, doi: 10.26740/jupe.v9n3.p138-144.
- [6] E. Trivaika and M. A. Senubekti, "Perancangan Aplikasi Pengelola Keuangan Pribadi Berbasis Android," *Nuansa Inform.*, vol. 16, no. 1, pp. 33–40, 2022, doi: 10.25134/nuansa.v16i1.4670.
- [7] R. H. Mawarti and T. Ibadi, "Pelatihan Perancangan Prototipe Aplikasi Android untuk Manajemen Keuangan Pribadi dengan Fitur Budgeting, Tagging, Aset, dan Laporan Bulanan Otomatis," *J. Pengabd. Masy. Bangsa*, vol. 3, no. 11, pp. 6799–6806, 2026, doi: 10.59837/jpmba.v3i11.3930.
- [8] D. P. Ratmiyanti, M. T. A. Bangsa, and Y. Ramadhani, "Rancang Bangun Sistem Informasi E-Learning

- Berbasis Web di MTs Laboratorium Jambi,” *JUKTISI*, vol. 4, no. 3, pp. 1603–1614, 2025, doi: 10.62712/juktisi.v4i3.686.
- [9] R. I. Melyani, R. Rosita, and S. Aji, “Pengembangan Sistem Informasi Penggajian Berbasis Web Menggunakan Framework Laravel dengan Metode Agile Software Development,” *JASIKA*, vol. 3, no. 1, pp. 31–36, 2023, doi: 10.31294/jasika.v3i01.2195.
- [10] D. Aryansyah, D. J. Author, and A. J. Author, “Pengembangan Sistem E-Commerce untuk Website Brand Finish One dengan Metode Agile,” *J. Inform. dan Tek. Elektro Terap.*, vol. 13, no. 3S1, 2025, doi: 10.23960/jitet.v13i3S1.7500.
- [11] S. B. Atim, “Permodelan Sistem Informasi Penjualan Barang Berbasis Website Menggunakan Metode Agile,” *JAITI*, vol. 2, no. 1, pp. 14–25, 2024, doi: 10.58602/jaiti.v2i1.104.
- [12] H. Handayani, A. M. Ayulya, K. U. Faizah, D. Wulan, and M. F. Rozan, “Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development,” *J. Test. dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 29–40, 2023, doi: 10.55583/jtisi.v1i1.324.
- [13] A. Ikhwan, A. A. Syahfitri, V. Ramadhani, and M. Y. Siregar, “Implementasi Agile dalam Perancangan UI/UX Sistem Informasi Absensi Siswa di MTS Muhammadiyah 01 Medan,” *Polygon*, vol. 3, no. 1, pp. 40–49, 2025, doi: 10.62383/polygon.v3i1.393.
- [14] Uminingsih, M. N. Ichsanudin, M. Yusuf, and S. Suraya, “Pengujian Fungsional Perangkat Lunak Sistem Informasi Perpustakaan dengan Metode Black Box Testing bagi Pemula,” *STORAGE*, vol. 1, no. 2, pp. 1–8, 2022, doi: 10.55123/storage.v1i2.270.
- [15] R. Wandri, M. Fadhillah, P. R. Setiawan, S. Daulay, A. Hanafiah, and D. F. Qurniawan, “Optimalisasi Pengelolaan Data Sekolah Melalui Sistem Informasi Sekolah Berbasis Web dengan Pendekatan Agile,” *SAINSTEK*, vol. 13, no. 1, pp. 157–165, 2025, doi: 10.35583/js.v13i1.327.
- [16] Y. D. Cahyono, L. W. Widiyanti, and S. I. Pramuningsih, “Pengujian Black Box pada Website Sistem Informasi Pengolahan Data Teritorial Menggunakan Metode Use Case Testing untuk Operasi Militer,” *JATI*, vol. 9, no. 2, pp. 2250–2256, 2025, doi: 10.36040/jati.v9i2.12915.
- [17] G. F. Fitriana, “Pengujian Aplikasi Pengenalan Tulisan Tangan Menggunakan Model Behaviour Use Case,” *JATISI*, vol. 7, no. 2, pp. 200–213, 2020, doi: 10.35957/jatisi.v7i2.390.
- [18] A. Ikhwan, Z. N. S. Lubis, and R. Z. Luthfi, “Perancangan Sistem Informasi Pengelolaan Komponen KWh Meter Berbasis Web pada PT PLN (Persero) ULP Medan Selatan,” *RIGGS*, vol. 4, no. 4, pp. 8456–8465, 2026, doi: 10.31004/riggs.v4i4.4942.
- [19] A. Ramadani, “Sistem Informasi Cuti Kepegawaian pada Rumah Sakit Umum Daerah Kabupaten Batu Bara,” *Modem*, vol. 3, no. 1, pp. 67–75, 2025, doi: 10.62951/modem.v3i1.350.
- [20] S. Oktaviandra, W. N. Syawalia, M. Al Fatah, M. F. Hadiwinata, and S. Sahrizal, “Rancang Bangun Sistem di CV Bumi Surya Beton dengan Metode Agile untuk Pengolahan Data Absensi Karyawan,” *RIGGS*, vol. 4, no. 4, pp. 13887–13896, 2026, doi: 10.31004/riggs.v4i4.5877.
- [21] I. S. P. Ihsan and D. Mallisza, “Sistem Informasi Pelanggaran Siswa Berbasis Web pada SMP Negeri 42 Padang,” *J. Manaj. Teknol. Inf.*, vol. 3, no. 2, pp. 80–89, 2025, doi: 10.70038/jentik.v3i2.118.
- [22] Y. Salim, M. A. Paris, D. Tjitradi, E. Eliatun, and E. Herliani, “Pengujian Aplikasi PPH21 Karyawan PT.X Menggunakan Metode Black Box Testing Equivalence Partitioning,” *Go Infotech*, vol. 31, no. 1, pp. 100–111, 2025, doi: 10.36309/goi.v31i1.365.
- [23] F. A. Sakinah, F. P. Aditiawan, and A. L. Nurlaili, “Pengujian pada Aplikasi Manajemen Aset Menggunakan Black Box Testing,” *JATI*, vol. 8, no. 3, pp. 2814–2823, 2024, doi: 10.36040/jati.v8i3.9524.
- [24] A. Gavrila and J. Parhusip, “Analisis Usability Aplikasi Google Classroom Menggunakan Metode System

- Usability Scale (SUS) dan Pengujian Black Box,” *JATI*, vol. 10, no. 1, pp. 310–316, 2026, doi: 10.36040/jati.v10i1.16680.
- [25] G. L. Ginting, R. K. Hondro, and A. Fau, “Implementasi System Usability Scale Guna Analisis Usability Aplikasi Moka,” *Informatics*, vol. 1, no. 1, pp. 6–14, 2025, doi: 10.63215/Informatics.v1i1.10.
- [26] J. R. Lewis, “The System Usability Scale: Past, Present, and Future,” *International Journal of Human-Computer Interaction*, vol. 34, no. 7, pp. 577–590, 2018, doi: 10.1080/10447318.2018.1455307.
- [27] P. Kortum and M. Sorber, “Measuring the Usability of Mobile Applications for Phones and Tablets,” *International Journal of Human-Computer Interaction*, vol. 31, no. 8, pp. 518–529, 2015, doi: 10.1080/10447318.2015.1064658.