

Design and Implementation of Mobile Native Application

1st Erika Ramadhani

¹ Universitas Islam Indonesia

¹Jl. Kaliurang No.Km. 14,5, Krawitan, Umbulmartani, Kec. Ngemplak,
Kabupaten Sleman, Daerah Istimewa Yogyakarta 55584

email : 1st erika@uii.ac.id

Abstract—Virtual money is very popular in Indonesia, the integration of many type of virtual money into one application enable the user to check many type of virtual money application easily into one platform according to a mobile native application. Mobile applications that have not used native technology have problems, that is, if one of the services is a system failure, the other system is experiencing problems as well. The objective of this paper is to overcome the problem, we made an application using native technology. We analyze the system according to the performance of the system by comparing when using native system and conventional application. This paper has a several steps as the method to reach the result, such as literature review, requirement analysis, design, implementation, and testing. The use of native technology does not cause system failure if one of the modules has an error. The development of a mobile native application for e-money using Native React and Firebase in the Android platform. The service of the application of e-money is enable to check the balance of the money in several type of virtual money application such as Bukalapak, Jakmall, Paypal, and several type of government bank. API service is used to integrate many platforms named Finfini. Furthermore, this API enables the mobile e-money application to communicate with a cross platform virtual money. The results of testing, the e-monies system using native technology has a performance of 13% better in using resources when compared to conventional applications

Keywords : mobile application; mobile native application; e-money; android; virtual wallet; virtual money.

I. INTRODUCTION

In 2017 there are several activities that related to the Internet. According to the survey of Asosiasi Penyedia Jasa Internet Indonesia (APJII) in 2017, the Internet user in Indonesia reached 143,26 million users from the total population 262 million people. In other words, the use of Internet in Indonesia reached to 54,68%. Financial activity that use an Internet such as e-money, online transaction, etc. In 2017, more than 89,35% the use of Internet is for chatting while financial is 7,39% [1].

In the term of online transaction, e-money or electronic money is the means of payment with the value that has been saved in electronic way on the server or transaction card. E-money card is used to do a payment transaction in Internet with cooperation of the bank. E-money become popular because of the frequently use in a daily life even the public facilitation has used e-money as the payment such as toll payment, commuter line, parking, etc [2].

E-money become popular in Indonesia since the launch of many mobile native applications such as

Bukalapak, Gojek, Grab, Tokopedia, Shopee, Lazada, and others. One example of mobile native application that very popular in Indonesia is Gojek. Gojek is an online transportation service such as taxi car and taxi motorcycle. There are several services in Gojek, the one very popular is Gofood, a service that a customer can buy a food by delivery order. All of the services of Gojek enable the user to pay with virtual money accrodng to e-money. The increasing user of Gojek make the increasing in the use of virtual money. Gojek is the application that use e-money technology as a payment system activley. Furthermore, it effected to another application to use e-money based payment system such as Bukalapak, Tokopedia, Shopee, etc.

According to the previous explanation, the effect is there are many type of virtual wallet. In order to make easy the user to check the balance of the several type applications, we attempt to make an application based on mobile native application to ease the user to check the balance of their virtual wallet. Native application is specifically written and developed for a specific operating system. Today, three type of mobile phone operating

system are popular, i.e. Android, iOS, and Windows, which is they have their own programming language that fit. For example, Java programming language must be used in the developing native application for Android, while Objective C programming language for iOS. Windows phone used .NET framework in developed. The advantage of this type of native application are this application easy to integrate within the device hardware, provide all type of user interface, all those characteristics are provide in the respective mobile operating system environment [3].

An integrated virtual wallet, also known as a digital wallet or e-wallet, has become increasingly urgent as more people conduct transactions online and seek a convenient and secure way to store their payment information. An integrated virtual wallet can also be integrated with other services, such as budgeting and financial management apps. This can provide users with a comprehensive financial management solution. In summary, the urgency of an integrated virtual wallet has grown due to its convenience, security, mobile payment options, loyalty program integration, and potential integration with other services. As more people conduct transactions online and on mobile devices, an integrated virtual wallet has become an essential tool for managing payments and financial information.

The rest of the paper is organized as follows. Section II explains in details the method and material in developing the application of mobile native. Result and discussion of the application is given in the Section III, continue with section IV which discusses the conclusion and recommendation.

II. RESEARCH METHODS

First we do a literature review about current development of mobile native application in Indonesia especially in the term of e-money. Then we analyze the requirement of the research on how to develop the mobile native application in the e-money field according to android application. The result of the analysis is requirement of functionalities, non-functionalities, input, process,

output, and software. Figure 1. depict the summarize method of the research.

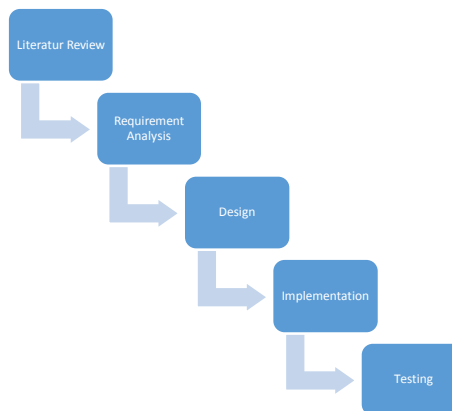


Figure 1. Method in the development of mobile native application

A literature review is a critical analysis of published literature on a particular topic or research question. It aims to identify, evaluate, and summarize relevant research in the field, and to highlight the gaps, contradictions, and areas for further investigation. A well-conducted literature review is an important step in the research process, as it provides a comprehensive overview of the existing literature, helps to identify research gaps and opportunities, and provides a foundation for developing research hypotheses and objectives.

Requirement analysis is a critical component of software development, as it helps to ensure that the system meets the needs of its users and stakeholders. By identifying and documenting the requirements, developers can design and implement a system that meets the functional and non-functional requirements of the system.

The design process involves several steps, including problem definition, research, ideation, prototyping, testing, and refinement. The ultimate goal of design is to create a solution that is functional, aesthetically pleasing, and easy to use.

In implementation stage, involves developing the code, building the hardware, configuring the software, and integrating the various components into a working system. The implementation phase follows the design phase and typically involves several steps, including coding, testing, deployment, and maintenance.

Testing is a process of verifying and validating a software application or system to ensure that it meets the specified requirements and works as intended. Testing helps to identify defects, errors, or gaps in the software, which can be fixed before the software is released to end-users. The testing process is typically carried out in multiple stages, from unit testing (testing individual units or components of the software) to system testing (testing the software as a whole). By testing at each stage, issues can be identified and fixed early, which helps to reduce the cost and time associated with fixing defects later in the development cycle.

2.1. Literature Review

The important thing in the mobile native application is an Application Programming Interface (API). API consist of function, command, and protocol which is a programmer may use it in developing a software with a certain operating system, also a programmer may use the standart function to interact with an operating system. A developer may use other application without knowing the inside of the application by using an API. API also can speed up the process of development of an application, it also provide a separate function [4]. The design of API depict in Figure 2. which the data retrieve from the database to android handheld using API.

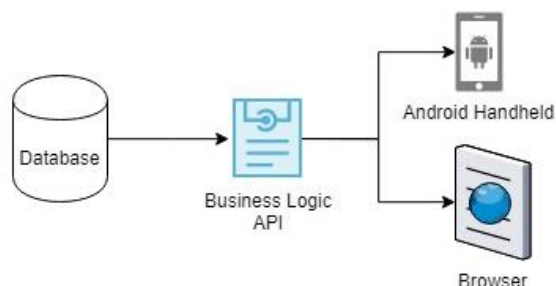


Figure 2. Firebase database and API design

React Native is the framework that we use in the development of mobile native application. The framework is based on JavaScript developed by Facebook which can be use in Android and iOS mobile platform. React js is the type of javascript library in React Native. The advantage of React Native is in a simple code, good performance

because it use native component and native API make it easy to convert into native iOS or native Android. Process recompiling is faster because it can reload the new code more fast with a hot reloading feature [5].

Firebase is a database storage based on cloud and NoSQL with JSON format when the data is saved, JSON format is sync in real time to each client who accessed it. In the development of mobile e-money, we use a type of Firebase Realtime Database and Firebase Email Authentication.

2.2. Requirement Analysis

The method that we use in the development of mobile native application is waterfall method [6]. We communicate with customer candidate to gain the information and data about the application. In this stage, we summarize the requirement of data specification from the project manager which is project manager has interviewed the customer candidate before. The result is we have the data about the description of the flow work of the application, programming language that is used, then the features that may consist in the application

- 1) Functionalities Requirement: there are several requirements in this stage i.e. the application provide the registration, login, display a money balance from internet banking, logout feature, profile editing, and display information about the last transaction.
- 2) Non-functionalities Requirement: consist of the interface of the application by using a simple material design, application can be use in handheld that has an android and IOS operating system.
- 3) Input Requirement: the input stage on the application are the action of input the user data for registration, authentication, and the data of the internet banking account.
- 4) Process Requirement: the requirement of this stage is the signup process, request process from API, refresh process for balance e-money, login process, updating the user data, and the logout process.
- 5) Output Requirement: this stage requirement is the output of information about the user profile, list of bank name integrated with API, balance

information from the internet banking, and transaction information.

- 6) Software Requirement: software that is used in this paper are JetBrains WebStorm, Mozilla Firefox, Genymotion Android Emulator, and Postman.

2.3. Design

JSON is the format of storage structure in Firebase. Below is the structure of database of e-money application. E-monies consist of the data tabled in Table I. which are about the tree of bank that provide e-money such as bank name, bank id number, etc.

Table 1. Document e-monies

Key	Type	Description
emoneyKey	Objectid	Collection primary key
alias_name	String	Alias name
automatic_fetch	String	Automatic fetch
country_code	String	Country code
forum_url	String	Forum url
home_url	String	Home url
id	Object	Bank id number
image	Array	Picture of user
instructions	String	Instruction
label_username	String	Username
login_url	String	Login
mode	Object	Mode
name	String	Name
refresh_timeout	String	Refresh timeout
status	Object	Status
vendor_type	Object	Vendor type

The storage structure for user tabled in Table 2. which are the function to store data of e-money application such as username, address, email, password, and e-monies data.

Table 2. Document data user

Key	Type	Description
userKey	Objectid	Collection primary key
email	String	Email user
emonies	String	E money

emoniesKey	Object	Type E money
balamce	Int	Balamce
bankName	String	Bank name
key	Object	Id
number	Int	Number
requestBody	String	Request body
password	String	Password user
username	String	Username user
vendor_id	String	Vendor id
status	Object	Status
transactions	Object	Type of transaction
key	Object	Id
additional_info	String	Additional info
reference_number	String	Reference number
ammount	String	The amount of balance
date	Date	Date of access
note	String	Note
homeAddress	String	Home address
id	String	Id
name	String	Name
password	String	Password
phoneNumber	String	Phone number
photoUrl	Array	Picture of user

Use case diagram consist six use case in the application of mobile native. The use case diagram is created from six functionalities requirement which depict in the Figure 3.

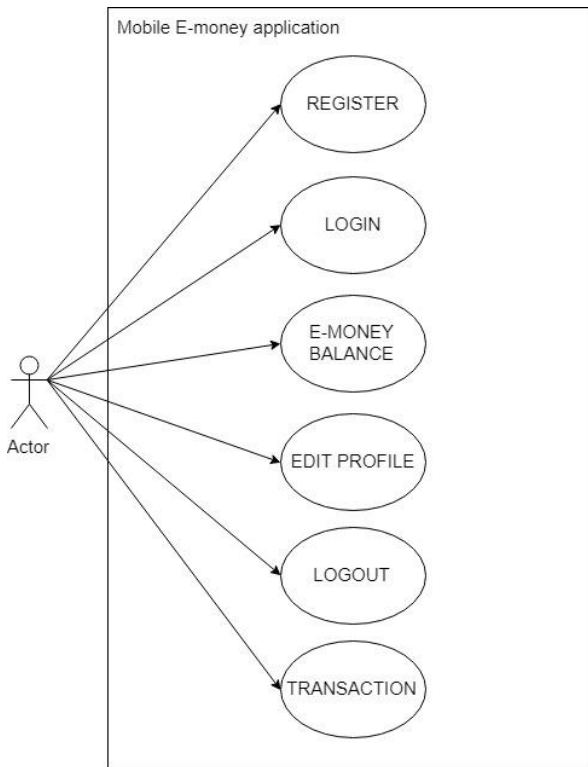


Figure 3. Use case diagram

The activity diagram modeled according to each functionalities of mobile e-money application. Table 3 and Table 4 are the activity diagram to display e-money data and display transaction information.

Table 3. Activity diagram data e-monies

Display E-money Data	
User	Application

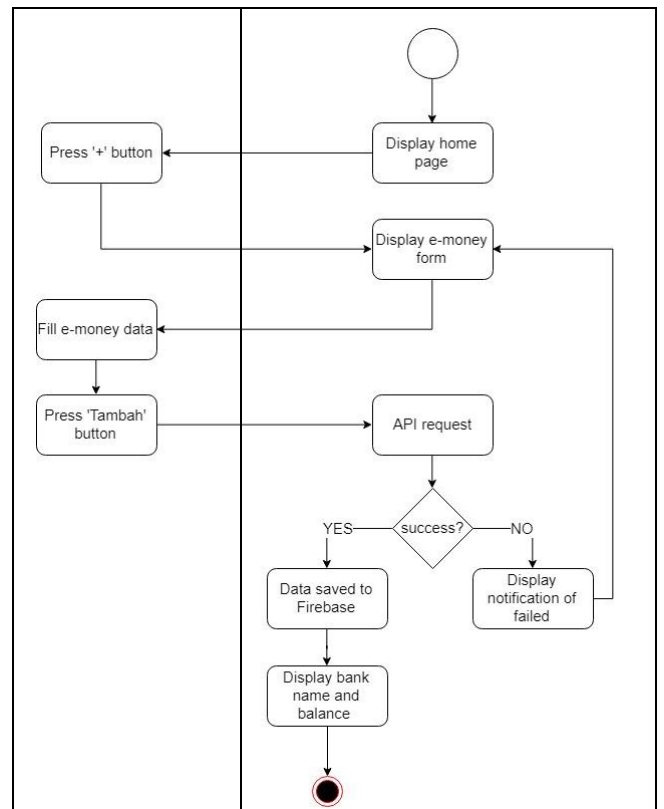
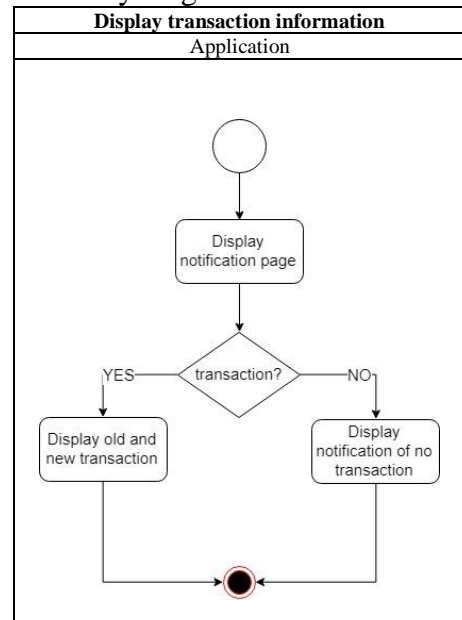


Table 4. Activity diagram of data transaction



III. RESULT AND ANALYSIS

3.1 Implementation

The implementation on how to integrate Firebase with the application mobile e-money then connect the application with the API. To integrate Firebase with the application, we need to provide the component of React Native and the dependencies module first, in order the application can work perfectly. Furthermore, we connect the

application using an API by request a service to API Finfini. Finfini is the API service from PT. Javasingna Intermedia which provide endpoint, this endpoint will give the return data of bank e-money. Data login and data password is needed to do a request to Finfini. The endpoint to access Finfini is in the Table 5.

Table 5. Endpoint of API

Data	Endpoint URL
List of data bank	https://finfini.com/api/v2/vendors
Data of e-money	https://finfini.com/api/v2/emoney/sync

Firestore has the real time database [7], this advantage lead to the system enable to generate the balance of e-money in real time as well. Figure 5 depict the method to display the balance of e-money. Figure 6 depict the method to check the availability of transactions.

```

isExists() {
  const userID = firebase.auth().currentUser.uid;
  firebase.database().ref('users/' + userID + '/eMonies/').once("value").then((snapshot) => {
    console.log("Apakah akun ini ada EMoniesnya ? " + snapshot.exists());
    console.log("Number of EMonies: " + snapshot.numChildren());
    this.setState({isMoney: +snapshot.exists()});
  }).catch((error) => {
    console.log(error);
  });
}

```

Figure 4. Method to check the balance information

```

isTransactions(){
  return f.database().ref("dummyData/data/transactions").once("value").then(snapshot => {
    this.setState({isTransactionsData: snapshot.exists()});
  }).catch(error => {
    console.error(error);
  });
}

```

Figure 5. Method to check the availability transaction information

The result of the implementation depict in Fig. 7. Which are the display of home screen and notification screen. In the home screen, information such as bank name are displayed, while in the notification screen, we can see the list of transactions of the user.

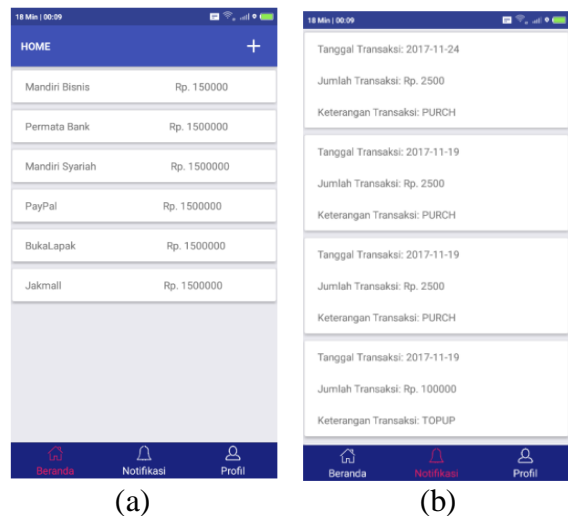


Figure 5. Implementation of native app (a) home screen (b) notification screen

The application display an error handling while the application accept a false input data. There are several condition when the system accept an action that act as an error to the system, such as the error login process when user input the incorrect username or password. The application will be notify the user that the input data is incorrect.

3.2 Testing

In this section we test the application by checking the error in the application that may occur. The test result is summarized in the Table 6. In test table we do a testing by checking the functionality of each scenario which we compare it to the expected result. If the expected result is related to the scenario then the functionality test succeeded.

Table 6. Test

No	Scenario	The Expected Result	Yes/No
1	User fill the false username and password	Display an error notification that show the user is not registered or wrong password	Yes
2	User tap the Login button without inputing the email and	Display error notification that show the false spell of email	Yes

	password in a login page.	or empty filling email.	
3	User click on the login button without filling the email and password in the registration page	Display error notification that show the email is false in the spelling or empty filling of email.	Yes
4	User filling the registered email that have registered before.	Display error notification that show the registered email has registered or has used in the application.	Yes
5	User tap the save button when edited the data profile otherwise the field is empty.	Display a notification to remind the user to fill in the data on the field.	Yes

The next test is measuring the system CPU usage between technologies natively with conventional technology. Conventional technology in question is a mobile application that still uses website-based technology.

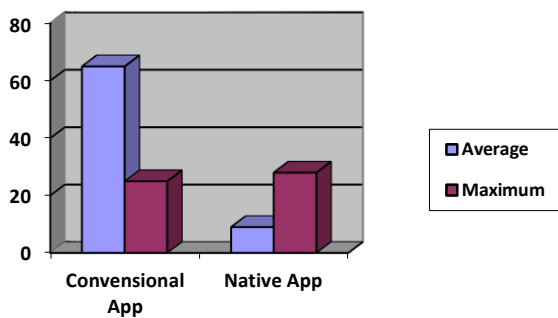


Figure 7. CPU utilization between conventional app and native app

In Figure 7. the test performed using the android device version 6. The device features are 1.89 GHz processor with four cores. The graphical show that CPU utilization when running android native apps is less compared then the convensional apps. As figured in the Figure 7. native applications has a

better performance then the convensional application.

IV. CONCLUSION

We conclude that React Native and Firebase can be used in developing a mobile native application for e-money according to mobile application. The performance given when using native technology is 13% better when compared to the use of conventional technology. The implementation is on how to integrate the communication between the application by using an API. Within an API, every cross platform type of internet banking can be integrated lightly. Although this paper only give an implementation on checking the balance of e-money, for future research, we can add another feature on the application that provide a transaction for transfer and check the balance in one integration of application.

REFERENCES

- [1] APJII, “Penetrasi & perilaku pengguna internet indonesia,” Indonesia, 2017.
- [2] S. Abadi *et al.*, “Design of online transaction model on traditional industry in order to increase turnover and benefits,” *Int. J. Eng. Technol.*, vol. 7, no. 2.27 Special Issue 27, pp. 231–237, 2018, doi: 10.14419/ijet.v7i2.31.1344
- [3] W. Jobe, “Native Apps vs . Mobile Web Apps,” vol. 7, no. 4, pp. 27–32, 2013.
- [4] B. Mulloy, “Web API Design – Crafting Interfaces that Developers Love,” p. 36, 2012.
- [5] K. Chinnathambi, *Learning React Native*, 1st ed. United States of America: O’Reilly, 2016.
- [6] Y. Bassil, “A Simulation Model for the Waterfall Software Development Life Cycle,” May 2012.
- [7] C. T. Google, “Firebase Realtime Database | Firebase Realtime Database [Firebase.]”[Online]. Available: <https://firebase.google.com/docs/database/>. [Accessed: 12-Jul-2018].