Performance Comparison of Native Android Application on MVP and MVVM

Bambang Wisnuadhi[1] Ghifari Munawar[2*] Ujang Wahyu[3]

[1,2,3]*Informatics and Computer Engineering Department, Politeknik Negeri Bandung, Bandung, Indonesia*
[*]*Corresponding author. Email: ghifari.munawar@polban.ac.id*

ABSTRACT
The performance of the android application is one factor that needs to be considered because an android device has limited power, memory and resources. This is a challenge for developers to improve performance so that applications can run optimally. There are several architectures that are commonly used, namely MVC, MVP, and MVVM. Previous studies have measured performance among the three architectures, and it can be concluded that the performance of MVC < (MVP = MVVM), but between MVP and MVVM which is better is still unclear. This study aims to compare the performance of MVP and MVVM architectures measured from 3 (three) aspects, namely CPU usage, memory usage, and execution time. Based on experiments, the results show that MVVM performance is better at CPU usage and execution time, while MVP is better at memory usage. CPU usage in MVVM applications is lower with an average difference of 0.55%. Execution time in MVVM applications is faster with an average difference of 126.21 ms, while memory usage in MVP applications is lower with an average difference of 0.92 Mb. This happens because the MVVM architecture has an additional library (in the form of a data-binding) that can increase application response so that CPU usage and execution time are better, but another impact is its memory usage is higher than MVP.

Keywords: *Mobile Development, Software Architecture, MVP, MVVM, Performance Comparison*

1. INTRODUCTION

Android is an open source mobile operating system (OS) with the most users in the world [1], but not all application performance is the same on every Android device, because each device has different computing capabilities. It has limitations in terms of power, memory and resources, which impact performance aspects [2]. In developing Android applications, architecture is an important thing to be considered because each application has different characteristics and approaches. Generally, new architectures are developed to solve problems in its previous. Therefore, there needs to be a scientific study to compare the several aspects of the software architecture to find out the advantages and disadvantages of each architectures [3].

There are three architectures that are generally used for android application development, including the MVC (Model View Controller) architecture, MVP (Model View Presenter), and MVVM (Model View View Presenter). The most widely used architecture is MVC because of its ease in the implementation process. However, this architecture also has a disadvantage is between the Controller and View has a high level of coupling [4]. In 2016, Google released the Android Architecture Blueprint project on GitHub [5]. This project provides two application examples developed using the MVP and MVVM architectures. This architecture introduces a new library to reduce code. It has been claimed that the MVP and MVVM architectures are better than MVC, but what about the performance? Which of the two architectures is better?

Performance represents a measure of how fast it runs on application, how fast it loads data, and overall connectivity to different operations. To evaluate the performance of an Android application, there are several aspects that need to be considered. L. Corral [6], and Sibarani [7] believe that CPU usage, execution time, and memory usage are the main aspects of measuring the performance. In this study, we use Android Profiler and Snapdragon Profiler to measuring it. Snapdragon Profiler is used to measure real-time the usage of CPU, and memory [8], while Android Profiler used to measure the execution time [9].

As an object of this study, we developed a Point of Sales (Pos) application on native android using the MVP and MVVM architectures. PoS applications are

related to the sales and transaction processes to increase the effectiveness and efficiency of a business unit [10]. The purpose of developing this application is to maintain the consistency of the architecture, where we built it in two versions, namely the MVP and MVVM versions, besides that the features of the PoS application will be used as a test case in performance testing which one of them can measure the impact of data volume. Performance comparison between MVP and MVVM measured by three aspects; namely CPU usage, memory usage, and execution time. The results of this test will then be reported as an evaluation of the performance of the two architectures.

1.1. Related Work

Performance analysis is an important thing to do, because performance is one of the major problems in application development. Teerath Das' study [11] noted that of the 2,443 commits made on Android apps, 180 of them contained 547 performance issues. Problems that arise include problems with the GUI, networking, memory management, and load images.

Related study has been conducted by C.Aygun [12] by analyzing the performance comparison of MVC and MVP architectures on application running time. This study uses a simple CRUD application as the object of the experiment and the number of concurrent users as the variable. The results show that MVP has better performance than MVC, and is easier to test. This study tested performance by looking at one aspect only, namely running time. Another study, L. Tian [4] by comparing three architectures: MVC, MVP and MVVM on aspects of modifiability, testability, and performance in native android applications. But the performance aspect measured its only memory usage. The results stated that the performance of MVC < (MVP = MVVM). This means that the performance of MVP and MVVM is better than MVC, but between the two do not have significant differences (tend to be the same). F. Sholichin [13] in his research has also compared performance in four architectures: MVC, MVP, MVVM, and VIPER on the iOS application. It also measured aspects of modifiability, testability, and performance, but the performance aspects measured was only memory usage, and CPU usage. These results show that the performance of MVVM and VIPER is better than the other two architectures (MVC, and MVP). MVVM is better in terms of memory usage, while VIPER is better in terms of CPU usage.

From the results of research that has been done, it is stated that the MVP and MVVM architectures are superior to MVC in several aspects of performance in android applications, but between the two (MVP and MVVM) it is still unclear which one is better, besides that the performance aspects measured are CPU usage and memory usage only. L. Corral [6], and Sibarani [7] believe that CPU usage, execution time, and memory usage are the main aspects of measuring the performance, so that in this study we try to compare the performance of the two based on these three aspects.

One important aspect of performance testing is the application object to be tested. L. Tian [4] also explains that an application suitable for performance testing is an application whose performance process is mostly carried out in the application itself, the more complex the application structure (not only text), the more data volume that is processed, the more it will affect the application performance. Therefore, in this study, we developed an offline PoS application to ensure that the test can be measured optimally.

1.2. Our Contribution

This study aims to compare the performance of the MVP and MVVM architectures in native android applications that measure three aspects: (1) CPU usage, (2) memory usage, and (3) execution time. The first part of this study is to build the android app (Point Of Sales) on the MVP and MVVM architectures use the Java language on Android Studio IDE. The second part then measures its performance through several iterations of testing using Snapdragon profiler and Android profiler tools. The test results are then compared and analyzed, which performance is better between MVP and MVVM. The results can be a consideration for android developers in using which architecture is better in terms of performance.

2. METHODOLOGY

This study is divided into two major stages, namely (1) implementation phase, and (2) evaluation phase. The implementation phase is the stage for building an Android point of sales (PoS) application on the MVP and MVVM architectures. In the evaluation phase, the application that has been built will be measured and analyzed, the results of testing based on predetermined metrics. The following is the methodology:

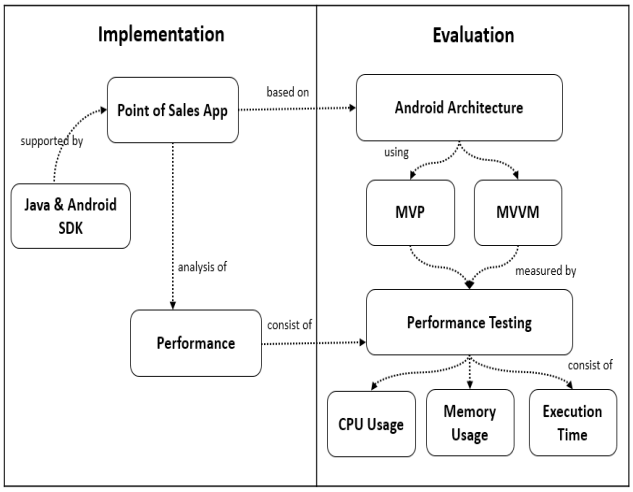

Figure 1. Methodology

2.1 Implementation Phase

This phase is conducted by building the POS application as an experimental object. Application development uses the software development life cycle (SDLC) method. This stage aims to analyze the needs of the application, design the overall system, carry out implementation and perform functional testing to ensure that the developed application is running according to the needs.

2.1.1 Usecase Diagram

The functional requirements of the application are modeled in the form of use cases as shown in Figure 2. The use-case diagram illustrates the interaction between actors and systems. In general, the PoS application developed has features to view product catalogs, shopping carts, reports, transaction management, product search, etc. The types of businesses that can be managed by this PoS application are small and medium-businesses that sell daily life products.

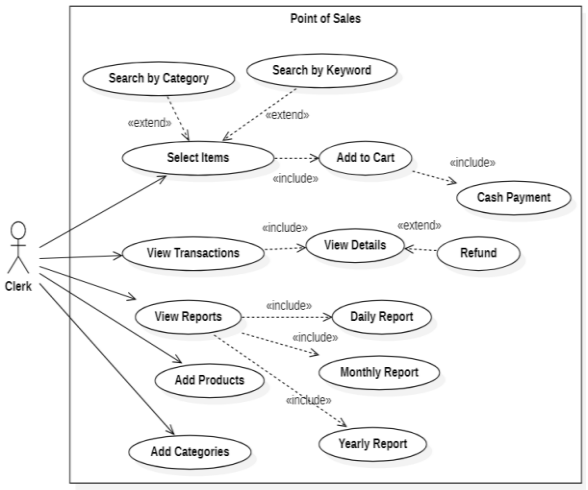

Figure 2. Point of Sales (PoS) Usecase Diagram

2.1.2 Architechture Diagram

An architecture diagram is needed to illustrate the design of the application to be built. This diagram shows the relationships between components and other technologies needed. More details can be seen in Figure 3 and Figure 4 which illustrate the MVP and MVVM diagram architecture. This architecture is used as a reference in writing program code to match each layer.

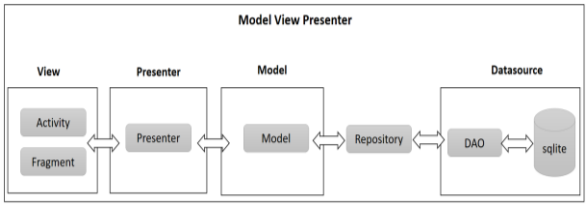

Figure 3. MVP Architecture of PoS Application

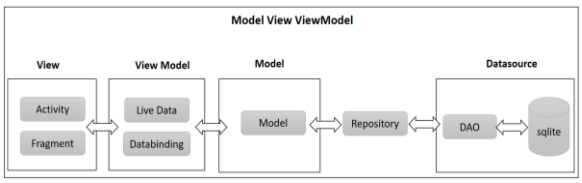

Figure 4. MVVM Architecture of PoS Application

2.2 Evaluation Phase

In the evaluation phase, the MVP and MVVM applications that have been built will be tested for performance with the following experimental stages:

1) Run the MVP and MVVM applications on the Android device.
2) Each test case will be measured using CPU usage, memory usage, and execution time.
3) Every measurement will be monitored through Snapdragon Profiler tools.
4) The measurement results will be exported as CSV files.
5) The value of each metric will be averaged according to the measurement results.
6) Measurements are made using two scenarios:
   a. Measurement based on the test case
   b. Measurement based on the data volume
7) Each scenario will be carried out 5 times, and the results will be averaged.

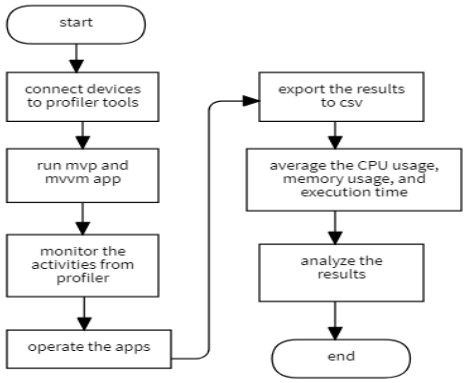

Figure 5. Experiment Steps

Before running an experiment, there are several conditions that need to be done on the device so that the measurement results are not affected by other processes. Conditions that must be met are:

1) There are no other applications running on Android devices
2) Disable internet connection and all notifications on the Android device

Experiments are carried out through performance measurement or commonly called profiling in each MVP and MVVM application. Profiling is obtained using snapdragon profiler tools that can record CPU usage in percent, memory usage in megabytes (mb), and execution time in milliseconds (ms). The

experimental scenario was carried out with two measurements, namely (1) measurement based on the test case and (2) measurement based on the data volume. This is done to see the impact of these three aspects of performance on data volume, whether they have the same or different result patterns. Table 1 shows the test case to be tested, and Table 2 shows the data volume to be processed. The specifications of the android device that will be used in the experiment are as follows:

- Brand : Asus Zenfone Max 3
- OS : Android 8.0
- RAM : 3GB
- Processor : Snapdragon 625
- CPU Cores : 8

Table 1. Testcase on performance test

Test-case *ID*	Description
TC-A-01	Displaying a list of items
TC-A-02	Search for items by keywords
TC-A-03	Select an item category and display a list of items.
TC-A-04	Place an order by selecting one item to be purchased.
TC-A-05	See a list of items in the basket.
TC-A-06	Make payments in cash by entering the keyword the amount of money paid
TC-A-07	Add new items by inputting keywords, item names, descriptions, buying prices, selling prices, quantities, units, categories, and images
TC-A-08	Displays a list of sales receipts or transaction history
TC-A-09	Displays sales reports

Data volume testing aims to measure the performance of applications with large amounts of data. We divided it into two data volumes (1000 data, and 10,000 data) to see a pattern as the data volume increased.

Table 2. Data volume on performance test

Test-case ID	Data Volume
TC-A-01	1,000 data
	10,000 data

To calculate the average percentage difference between MVP and MVVM, we use the formula of means difference and derive it based on data volume:

$$\mu_{M1-M2} = \mu_1 - \mu_2 \quad [14]$$

M1 represents the MVVM, while *M2* represents the MVP. *a* calculate the percentage of difference in 1,000 data, and *b* in 10,000 data:

a = 1,000 x (|**MVVM – MVP**| / 100);
b = 10,000 x (|**MVVM – MVP**| / 100);

The average difference = (*a* + *b*) / (1,000 + 10,000) * 100;

The data that has been obtained in the form of CPU usage, memory usage, and execution time values in each scenario will be compared in results between applications that use the MVP and MVVM architectures. The lower the value of CPU usage and memory usage means the better, the faster the execution time means the better. Thus, it can be concluded which architecture is better between MVP and MVVM in terms of performance (CPU usage, memory usage, and execution time).

3. RESULT AND EVALUATION

3.1 Implementation

The PoS application is implemented on the Android Studio IDE with the Java programming language. The application was built on two projects, namely the MVP and MVVM applications. The apk file generated by the two applications is not much different, which is 30.3 MB in the MVP application and 30.4 MB in the MVVM application. The file structure in MVVM has additional data-binding folders as shown in Figure 6, this causes a slight increase in file size. The application module has been implemented as needed; examples of the user interface display can be seen in Figure 7 and Figure 8.

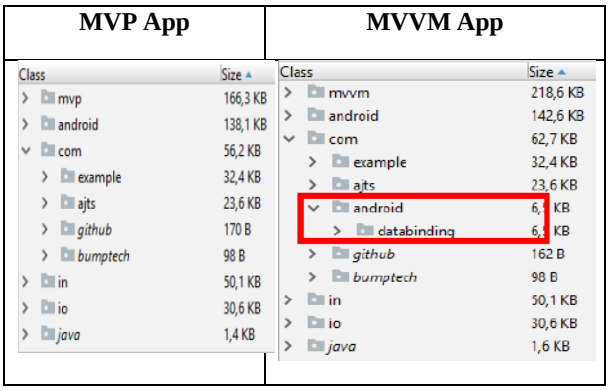

Figure 6. MVP and MVVM files structure

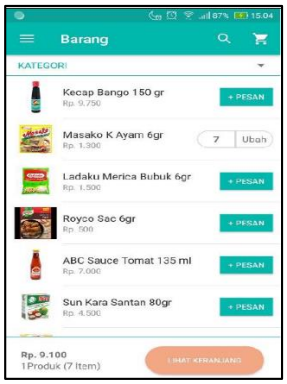

Figure 7. Catalog Page

Figure 8. Report Page

To test the application features that have been built to meet the needs, functional testing is performed. This test is done in a black box by testing the functionality of each test case. The functional application has run well and is ready to an experimental object at the evaluation stage. The test results are shown in Table 3.

Table 3. Functional Test Result

Test-case *ID*	Status	
	MVP Apps	**MVVM Apps**
TC-A-01	*Pass*	*Pass*
TC-A-02	*Pass*	*Pass*
TC-A-03	*Pass*	*Pass*
TC-A-04	*Pass*	*Pass*
TC-A-05	*Pass*	*Pass*
TC-A-06	*Pass*	*Pass*
TC-A-07	*Pass*	*Pass*
TC-A-08	*Pass*	*Pass*
TC-A-09	*Pass*	*Pass*

3.2 Performance Testing

Applications will be measured on CPU usage, memory usage, and execution time based on 9 prepared test cases. In addition to measuring the average of each test case, testing is also done by measuring the difference in the data volume; this is needed to see whether the pattern of test results has the same or different results. Each of the sections below explains the measurement results.

3.2.1 *CPU Usage*

The measurement of CPU usage in an application is aimed at Figure 9. In each test, the MVVM application is always lower than the MVP, with an average difference of 0.60%. For testing the data volume also has the same pattern where MVVM is lower than MVP with an average difference of 0.55% as seen in Figure 10.

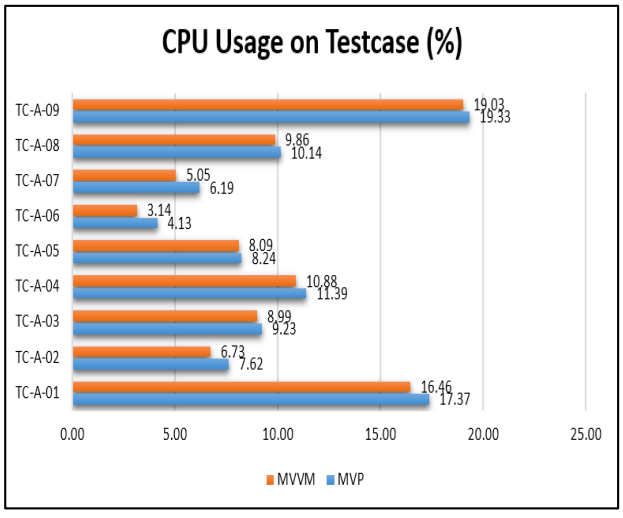

Figure 9. CPU usage result on testcase

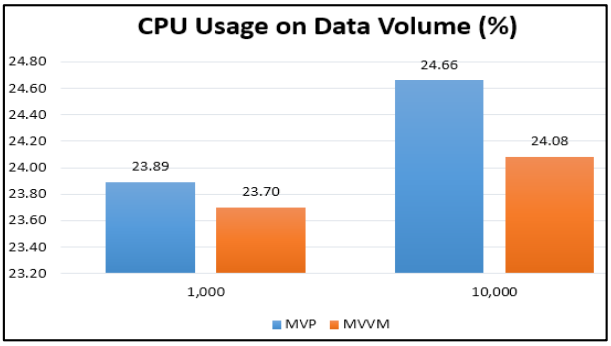

Figure 10. CPU usage result on data volume

The calculation of the average difference of 0.55% is obtained using the formula described in section 2.2:

1,000 x 0,20 / 100 = 2; 10,000 x 0,59 / 100 = 59;
2 + 59 = 61;
61 / (1,000 + 10,000) * 100 = **0,55%**;

3.2.2 *Memory Usage*

In the measurement of memory usage, the results got differ from CPU usage, where the MVP application has lower memory usage compared to MVVM with an average difference of 0.51 mb as listed in Figure 11. The same pattern is also obtained when done testing of the data volume; MVP has a lower value than MVVM with an average difference of 0.92 mb as seen in Figure 12.

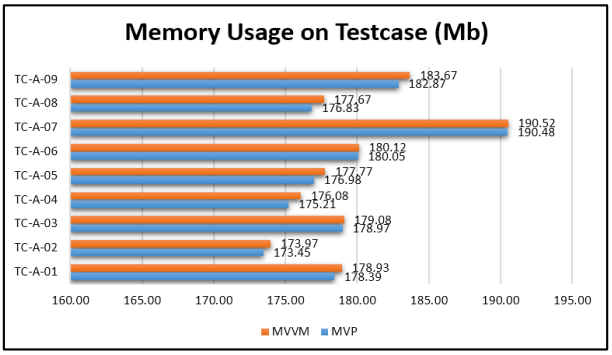

Figure 11. Memory usage result of Testcase

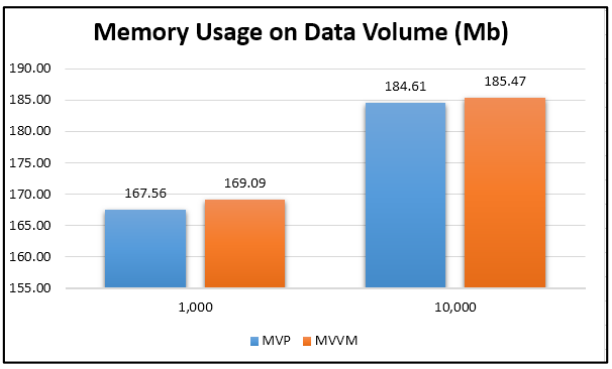

Figure 12. Memory usage result on data volume

3.2.3 *Execution Time*

The results of execution time measurements on android devices are shown in Figure 13, where the MVVM application has a faster execution time than MVP with an average difference of 28.67 ms. Similarly, the results of testing on the data volume, where MVVM is superior to a significant average difference, which is 126.21 ms as seen in Figure 14.

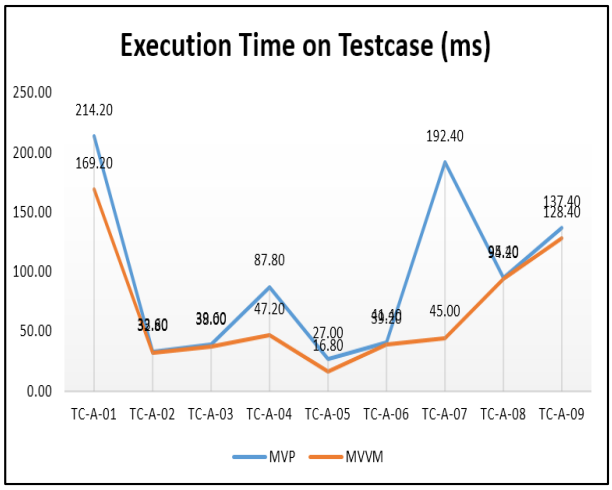

Figure 13. Execution time result of Testcase

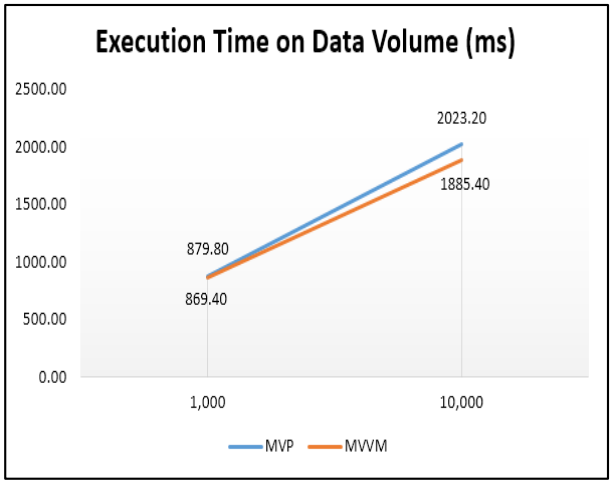

Figure 12. Execution time result on data volume

3.3 Evaluation

The use of both architectures has different output effects on Android applications. In this study, the impact that occurs on the measured aspects is as follows: (1) at CPU usage, MVVM applications are lower than MVP, with an average difference of 0.55%. (2) At memory usage, MVP applications have smaller results than MVVM with an average difference of 0.92 mb. (3) at execution time, the test results show that the MVVM application execution time is faster than MVP with an average difference of 126.21 ms.

Based on the experimental results, we can conclude it that the MVVM architecture produces better performance values than MVP, although not in all aspects. MVVM architecture is better in the aspects of CPU usage and execution time, while the MVP architecture is better in terms of memory usage. This happens because the MVVM architecture has additional libraries (in the form of data-binding) that can improve application responses so that CPU usage and execution times are better. Another impact of using this library is that its memory usage is greater than MVP. Thus, it can be said that in terms of performance **MVVM** > **MVP**. Using MVVM architecture is recommended for android application development if the factors considered are CPU usage, and better execution time.

4. CONCLUSION

Software architecture becomes very important in developing an application on both small and large scale, especially in the development of android applications. The emergence of MVP and MVVM architecture can be a consideration for android developers to migrate from MVC architecture to MVP or MVVM architecture. Besides productivity factors, this architecture can also affect performance factors. This research has compared the effect of using MVP and MVVM architectures on the performance of android applications by measuring CPU usage, memory usage, and execution time. Based on the experimental results it can be concluded that the MVVM architecture has better performance than MVP although not in all aspects. MVVM is better in terms of CPU usage and execution time, while MVP is better in terms of memory usage. One factor that influence is the existence of additional libraries in the form of data-binding that can improve MVVM performance in terms of system response. So that CPU usage and execution time are better, but the other impact is higher memory usage.

Measurements made in this study are limited to data processing internally in the application (offline). Application objects that are used as experiments only for the needs of proof of the concept, so these results need to be further reviewed in applications with the higher complexity, including the influence of several factors such as access to external networks (online), the use of API / Library from other systems (parties third), back-end / service technology used, etc. In addition, further research can also compare the use of MVP or MVVM architecture in other native Android languages such as kotlin, whether the results will also apply the same or are there any other influences with the use of the language, this will also be something interesting to discuss.

ACKNOWLEDGEMENT

This research was conducted under the group of software engineering expertise majoring in computer engineering and informatics in Politeknik Negeri Bandung (POLBAN). Acknowledgments are conveyed to the research and community unit (P3M) POLBAN who have provided their support in increasing the quantity and quality of research, and granting publication assistance so that this article can be published.

REFERENCES

[1] R. van der Meulen and Thomas McCall, "Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017," *Gartner, Inc*, 2018. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017. [Accessed: 28-Jan-2019].

[2] D. Sillars, *High Performance Android Apps: improving ratings with speed, optimizations, and testing*. 2015.

[3] V. Humeniuk, "Android Architecture Comparison : MVP vs . VIPER," 2018.

[4] L. Tian, "A Comparison of Android Native App Architecture – MVC , MVP and MVVM," 2016.

[5] Github, "Android Architecture Blueprints v2," 2016. [Online]. Available: https://github.com/googlesamples/android-architecture. [Accessed: 01-Dec-2018].

[6] L. Corral, A. Sillitti, and G. Succi, "Mobile multiplatform development: An experiment for performance analysis," *Procedia Comput. Sci.*, vol. 10, pp. 736–743, 2012, doi: 10.1016/j.procs.2012.06.094.

[7] N. S. Sibarani, G. Munawar, and B. Wisnuadhi, "Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Analisis Performa Aplikasi Android Pada Bahasa Pemrograman Java dan Kotlin," *9th Ind. Res. Work. Natl. Semin.*, no. Juli, pp. 319–324, 2018.

[8] Qualcomm, *Qualcomm ® Snapdragon ™ Profiler*, G. San Diego, U.S.A.: Qualcomm Technologies, Inc., 2020.

[9] developers.android.com, "Measure app performance with Android Profiler," 2020. [Online]. Available: https://developer.android.com/studio/profile/android-profiler. [Accessed: 09-Jan-2020].

[10] D. Setiaji, "Kumpulan Startup Penyedia Layanan Point of Sales (POS) di Indonesia," *Technisia*, 2017. [Online]. Available: https://id.techinasia.com/kumpulan-startup-pos-di-indonesia. [Accessed: 05-Mar-2019].

[11] T. Das, M. Di Penta, and I. Malavolta, "A quantitative and qualitative investigation of performance-related commits in android apps," *Proc. - 2016 IEEE Int. Conf. Softw. Maint. Evol. ICSME 2016*, no. January 2018, pp. 443–447, 2017, doi: 10.1109/ICSME.2016.49.

[12] C. Aygun and E. Kazan, "The Performance Analysis of Applications Written Using MVP and MVC," *Int. J. Sci. Res. Inf. Syst. Eng.*, vol. Volume 1, no. Issue 2, p. 8, 2015.

[13] F. Sholichin, M. Adham, S. A. Halim, and M. Firdaus Bin Harun, "Review of Ios Architectural Pattern for Testability," *J. Theor. Appl. Inf. Technol.*, vol. 15, no. 15, pp. 4021–4035, 2019.

[14] D. M. Lane, D. Scott, M. Hebl, R. Guerra, D. Osherson, and H. Zimmer, "Sampling Distribution of Difference Between Means," *Introduction to Statistics*, 2013. [Online]. Available: http://onlinestatbook.com/2/sampling_distributions/samplingdist_diff_means.html. [Accessed: 16-Apr-2020].