



Effect Of MVVM Architecture Pattern on Android Based Application Performance

Hammamul Achdan Epiloksa^{*}, Dana Sulistyo Kusumo, Monterico Adrian

School of Computing, Informatics Study Program, Bandung, Indonesia

Email: ^{1,*}epiloksaachdan@student.telkomuniversity.ac.id, ²danakusumo@telkomuniversity.ac.id,

³monterico@telkomuniversity.ac.id

Correspondence Author Email: epiloksaachdan@student.telkomuniversity.ac.id

Abstrak– Kinerja aplikasi mobile berbasis android sangat penting untuk diperhatikan karena kinerja akan berkaitan dengan bagaimana pengguna akan merasakan pengalaman terhadap aplikasi tersebut. Hal ini berkaitan dengan penggunaan memori dan penggunaan CPU pada perangkat android ketika menggunakan aplikasi. Terdapat beberapa cara yang digunakan dalam upaya meningkatkan kinerja aplikasi mobile berbasis android, salah satunya yaitu dengan menerapkan architecture pattern. Penggunaan architecture yang tepat diharapkan menghasilkan kinerja yang baik bagi perangkat android ketika menjalankan sebuah aplikasi. Oleh karena itu penulis melakukan penelitian apakah penerapan architecture pattern MVVM dengan melakukan perubahan kode program dengan metode refactoring akan berpengaruh terhadap penggunaan memori, CPU, dan waktu eksekusi code pada aplikasi berbasis android. Hal tersebut diimplementasikan menggunakan aplikasi Android Studio dan menggunakan tools android profiler untuk mengukur kinerja aplikasi tersebut.

Kata Kunci: Performance; CPU Usage; Memory Usage; Execution Time; MVVM (Model View ViewModel); MVC (Model View Controller)

Abstract–The performance of android-based mobile applications is very important to pay attention to because performance will be related to how users will feel the experience of the application. This relates to memory usage and CPU usage on android devices when using applications. There are several ways used to improve the performance of android-based mobile applications, one of which is by applying an architecture pattern. The use of the right architecture is expected to produce good performance for android devices when running an application. Therefore, the author conducted a study on whether the application of the MVVM architecture pattern by changing the program code with the refactoring method will affect memory usage, CPU, and code execution time in Android-based applications. This is implemented using the Android Studio application and using android profiler tools to measure the performance of the application.

Keywords: Performance; CPU Usage; Memory Usage; Execution Time; MVVM (Model View ViewModel); MVC (Model View Controller)

1. INTRODUCTION

The development of the world of technology has led to the emergence of many android applications that help humans in obtaining convenience in living life. There are 2,797,581 applications in the Play Store as of September 2021 [1]. As time goes by, developers continue to look for ways so that the applications they create can provide good results and experiences for users. Performance is a significant thing in an android application because it will be directly related to power usage, process speed in an application, and how the application interacts with users [70]. One of the things that affect the performance of an application is the Architecture Pattern [2]. There are several architecture patterns that are used when building an android application, some of which are MVP (Model, View, Presenter), MVVM (Model, View, View-Model), and MVC (Model, View, Controller) [3]. Seven out of ten developers encounter problems in developing or building applications, and these problems can be solved by applying the architecture pattern or design pattern [4]. Android is an open-source Linux-based operating system developed by Google. The android operating system is mainly designed for smartphones and gadgets. Android is a favorite for users and developers because it is open source because developers will be able to modify and add features to this operating system [5]. In designing an android-based application, the application of the architecture pattern in writing code can be more easily understood, this will affect a case when a view on the application will be reused or if there are changes to the code, the developer will easily understand the flow and usability of the code to be used. changed [3].

Architecture Pattern is a pattern that is applied repeatedly to the creation of an application or software in order to build a dynamic and easy-to-understand software design construction [6]. The diversity of architectural models and databases is a challenge for developers so research is needed to find out which architecture is most optimally applied, because it can produce different performances [7]. The use of architecture patterns can help accelerate the development of a software because it uses patterns that have been tested for usefulness [8]. The MVVM architecture pattern is an alternative to the previous architectural patterns, namely MVP and MVC. In its application in mobile app-based programming, the architectural pattern used by default is MCV [9]. Meanwhile, compared to MVVM the complexity of the code in the use of the MVC architectural pattern will be higher which will cause the code to be more difficult to debug. In a study conducted by Tian Lou that for performance evaluation, MVVM is superior to MVC [10]. In android applications, of course, the MVVM architectural pattern will present various advantages such as implementing the LiveData feature on the ViewModel. ViewModel will store data in



the form of Live Data which will solve the problem of crashes in the application due to the cessation of activity [4]. Using Live Data will also solve the problem of manually setting the lifecycle, reducing code complexity.

In short, the application of MVVM to the android application code will reduce the complexity of the program code which affects the use of the code more efficiently. In the previous study conducted by Tian Lou from Eindhoven University of Technology [9], performance measurement was only carried out by measuring memory usage, while according to ISO / IEC 25010: 2011 performance efficiency was measured by 3 variables, namely memory usage, CPU usage and execution time. This research was conducted in order to find out whether it affects the performance of the android application. The purpose of this study is to produce performance values measured from CPU usage, memory usage, and execution time of an android application after the application of the MVVM architecture pattern to the application and to find out whether the advantages that exist in the MVVM architecture pattern make the performance of the Android application to be better. The issue raised is whether the use of the MVVM architecture pattern can affect the performance of Android applications.

In the process of implementation, the programming language used is Kotlin. Kotlin itself is a programming language developed by JetBrains which was created mainly to develop smartphone applications based on Android, therefore Kotlin is also integrated with Android Studio. This programming language runs on the JVM (Java Virtual Machine) platform then all the libraries in java can also be used on kotlin [11], this programming language also brings many advantages such as a higher level of security.. Kotlin is also LLVM as a compiler and can be compiled into Javascript code [12]. In 2017 Google announced that Kotlin was becoming one of the official languages in the development of android-based applications [13].

2. RESEARCH METHODOLOGY

2.1 Research Stages

The research stages show what steps we took in conducting this research, it can be seen in figure 1 that there are four stages that we have carried out from beginning to end so as to get the results of performance measurements.



Figure 1. Research Stages

The process is carried out with the stages shown in Figure 1. starting with the selection of the base code that will be applied to the MVVM architecture pattern. Furthermore, the measurement of performance values in applications that have not applied the MVVM architecture pattern. The next step is to apply the MVVM architecture pattern to the application and again perform performance measurements on applications that have applied the MVVM architecture pattern.

2.2 Base Application Selection

In selecting the application to be executed, We looked for a simple open source application on GitHub with the consideration of whether the application has implemented the architecture pattern or not. The selected application is "news app" which is an application that is used to read and store news obtained from the API.

2.3 MVVM Implementation

The implementation of MVVM in the application was done by implementing supporting classes and libraries. In mvvm implementation Each component was be separated into several different packages containing each component. The components in MVVM are described in the flow shown in Figure 2.

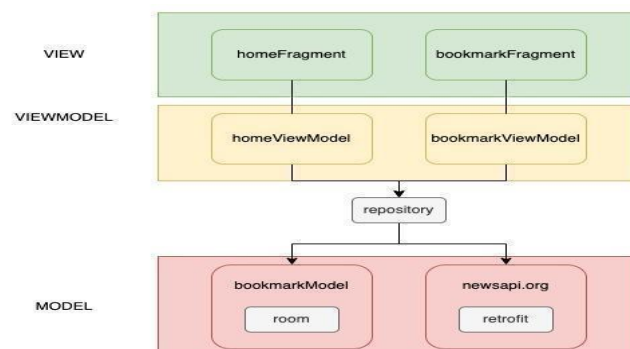


Figure 2. MVVM Application Components



In the process of implementing the MVVM architecture pattern, the authors performs code refactoring, In the software evolution context, refactoring is a re-engineering technique or the process of changing a software system that aims at reorganizing a program to improve its quality without changing its external behavior [14], [15]. This is done with the aim of changing the architectural structure of the application which was previously a Model View Controller (MVC) to a Model View Viewmodel (MVVM) while maintaining the functionality of the application. The following is an overview of the mapping of each class from MVC to MVVM:

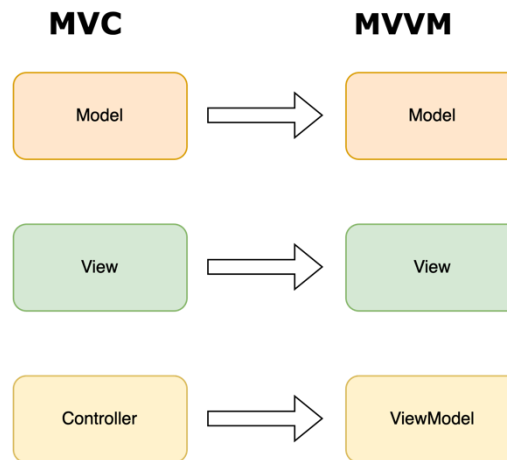


Figure 3. MVC to MVVM mapping

2.3.1 View

The view component includes both the activity class and the fragment that is responsible for the display. The View will observe the data in the ViewModel component and make updates when the data changes

2.3.2 ViewModel

ViewModel is a component that represents how a view interacts with the data in the Model, so that the view can represent and process the data contained in the Model class [16]. In this case ViewModel acts as a component that stores data in the form of livedata so that the view component can observe changes that occur in existing data. The viewmodel component also communicates with the data repository.

2.3.3 Model

The model components will be directly related to the data to be used in the application. The repository will be responsible for the data needed by the application. Data will be obtained in two ways, namely by using restAPI remotely with the help of a retrofit library and with a Room database in the form of local data [17].

2.3 Performance Measurement

To get optimal results, application performance measurements was carried out with real devices that use the Android operating system. This was done because if it is only done with an emulator in Android Studio, the measurement results can be larger and take longer to execute due to memory limitations on the emulator.

The measurement was done using a tool available in android studio called android profiler which will be able to measure CPU usage, specifically the android profiler system will measure and show a percentage of the total available CPU time

On the measurement of memory usage android profiler will show the amount of memory used by each memory category. As for the execution time, it was measured by android studio itself when running applications on android emulators. According to ISO/IEC 25010 shown in Figure 4, the parameters for software performance measurement are time behavior, resource utilization, and capacity.

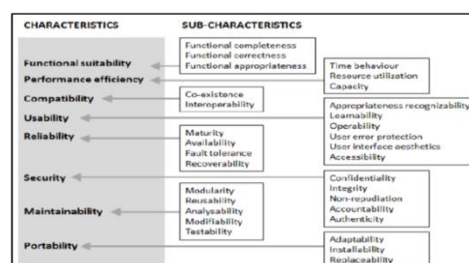


Figure 4. Application Quality Characteristic [18]



3. RESULT AND DISCUSSION

In this study, testing was carried out by first testing the functional of each case that could occur in the application such as displaying news, choosing categories and others. The test scenario was divided into two parts, namely in the MVC architecture pattern and in the MVVM architecture pattern, while the parameters tested in each architectural pattern are CPU usage, Memory usage, and Execution time. The average value of each test result was recapitulated as material for analysis which will later determine the final result of the test. After getting the results of the analysis, it was concluded whether the implementation of the MVVM architecture pattern affects the performance of the Android application as measured by several predetermined parameters, and we will also find out the average value of application performance in each architectural pattern.

3.1 Application Functional Testing

This test was performed to ensure that all the functions available in the application can run in order to measure the performance of the application on the function. Table 1 containing the expected results of each functional application:

Table 1. Functional Test Requirement

Test.ID	Deskripsi
T.01	Splash Screen
T.02	Display a list of news categories in the "sports" category
T.03	View selected news

Table 2 is the result of testing the functionality of the application that has been carried out:

Table 2. Functional Test Result

No.	Functionality	MVC apps	MVVM apps
1	T.01	✓	✓
2	T.02	✓	✓
3	T.03	✓	✓

3.2 Application Performance Measurement

The test was conducted using a Samsung Galaxy A9 device (SM-A920F) that uses the Android version 10 operating system (Android Q).

3.2.1 CPU Usage

Testing was performed on the device by using the android profiler. The CPU Profiler displays the CPU usage and thread activity of your application. At first glance, it has a complex user interface, but it's an extremely powerful tool for identifying performance issues [19]. The measurement of CPU usage in the application is carried out by running a predetermined feature, the results of the CPU usage test are shown in the following figure:

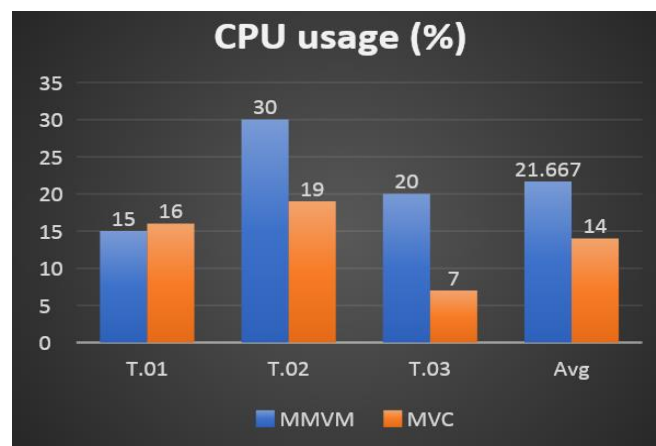


Figure 5. CPU Usage

3.2.2 Memory Usage

Testing is performed on the device by using the android profiler. The measurement of memory usage in the application is carried out by running a predetermined feature, the results of the memory usage test are shown in the following figure:

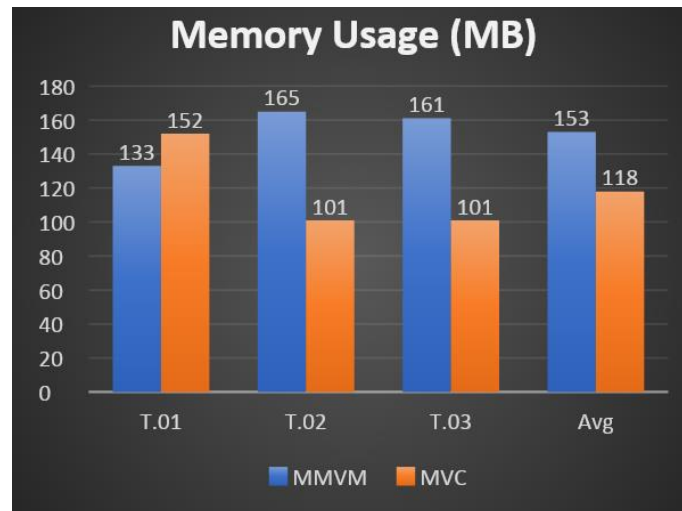


Figure 6. Memory Usage

3.2.3 Execution Time

Testing is performed on the device by using the android profiler. The measurement of execution time on the application is carried out by running a predetermined feature, the results of the execution time test are shown in the following figure:

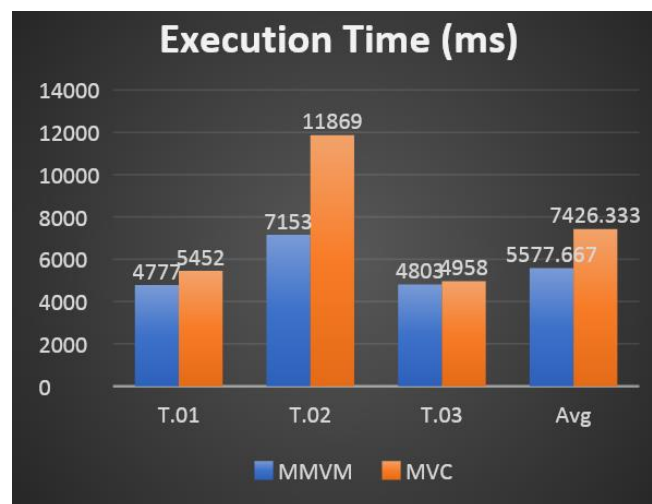


Figure 7. Execution Time

3.3 Analysis

Based on the measurements that have been made on predetermined application features by paying attention to CPU usage, memory usage, and program execution time. Minimal CPU usage will certainly affect the improvement of user experience quality and minimize battery usage which will increase user comfort when running applications [20]. To determine whether the MVVM architecture pattern affected the performance of the android application and determine which architectural pattern has better performance when implemented into the android application. The following are the results of the recapitulation of all measurement results that have been carried out:

Table 3. Performance Measurement Recapitulation

Parameter	Result	
	MVC	MVVM
CPU usage (%)	14	21,667
Memory Usage (MB)	118	153
Execution Time (ms)	7426,333	5577,667



After refactoring the application with the MVC architecture pattern to MVVM, the pattern and layout of some codes have changed, even in the MVVM architecture pattern, the code that must be written becomes more so that it affects the process of running the application. Based on the results that have been obtained, it can be seen that the use of MVVM architectural pattern affects each specified measurement variable (CPU Usage, Memory Usage, Execution Time). The MVC architecture pattern gets a good performance value with a smaller CPU and memory usage value compared to MVVM architecture pattern, the measurement results show that the MVC architecture pattern obtains better performance values on 2 variables (CPU usage and Memory usage), but in contrast to the program code execution time, in MVVM architecture pattern get a smaller value compared to MVC architecture pattern, this suggests that the program code can be executed faster than the MVC architecture pattern. Thus judging from the change in the value of the measurement results of the two architectural patterns can show that the MVVM architecture pattern can affect the performance of android applications.

4. CONCLUSION

MVVM implementations are performed by refactoring source applications that use the MVC architecture pattern into the MVVM architecture pattern and measuring CPU usage, memory usage, and program code execution time for each application before and after the MVVM implementation. The results of measuring CPU usage show that applications with the MVC architecture pattern get an average value of 14%. This value is smaller than applications with the MVVM architecture pattern, which has an average CPU usage value of 21.667%. While the difference between the two values is 7.667%. The results of the measurement of memory usage showed that applications with the MVC architecture pattern got an average value of 118 mb. This value is smaller than applications with the MVVM architecture pattern, which have an average CPU usage value of 153mb. While the difference between the two values is 35 mb. While measurements of the speed of program execution time show that the MVVM architecture pattern can execute program code faster by obtaining an average value of 5577.667 ms. This value is smaller compared to the MVC architecture pattern which executes program code longer and gets an average score of 7426,667 ms. It can be seen from the results of the android application performance measurement in this case study that it can be better when written with the MVC architecture pattern compared to the MVVM architecture pattern because it outperforms the performance measurement value on two of the three measurement variables. Judging from the difference in measurement results, it shows that the implementation of the MVVM architecture pattern can affect the performance of Android applications as measured by CPU usage (CPU Usage), Memory Usage (Memory Usage), and program code execution time (Execution Time).

REFERENCES

- [1] L. Ceci, "Number of available applications in the Google Play Store from December 2009 to March 2022," *statista.com*, Jul. 13, 2022.
- [2] N. Mani, D. C. Petriu, and M. Woodside, "Studying the impact of design patterns on the performance analysis of service oriented architecture," in *Proceedings - 37th EUROMICRO Conference on Software Engineering and Advanced Applications, SEAA 2011*, 2011, pp. 12–19. doi: 10.1109/SEAA.2011.13.
- [3] W. Sun, , haohui Chen, and W. Yu, "The Exploration and Practice of MVVM Pattern on Android Platform," 2017.
- [4] B. Rizki, P. Surya, A. Putra Kharisma, and N. Yudistira, "Perbandingan Kinerja Pola Perancangan MVC, MVP, dan MVVM Pada Aplikasi Berbasis Android (Studi kasus : Aplikasi Laporan Hasil Belajar Siswa SMA BSS)," 2020. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] R. Singh, "An Overview of Android Operating System and Its Security Features," 2014. [Online]. Available: <http://ptcoresec.eu/2013/05/02/part-1-getting>
- [6] "Examples to Accompany: Design Patterns."
- [7] M. Kustino Muharram, Z. Ariachandra, B. Wisnuadhi, and G. Munawar, "Prosiding The 12 th Industrial Research Workshop and National Seminar Bandung," 2021.
- [8] "THE IMPACT OF DESIGN PATTERNS IN REFACTORING TECHNIQUE TO MEASURE PERFORMANCE EFFICIENCY."
- [9] T. Lou, "A comparison of Android Native App Architecture MVC, MVP and MVVM."
- [10] J. Estdale and E. Georgiadou, "Applying the ISO/IEC 25010 Quality Models to Software Product," in *Communications in Computer and Information Science*, 2018, vol. 896, pp. 492–503. doi: 10.1007/978-3-319-97925-0_42.
- [11] A. SOSHIN, *HANDS-ON DESIGN PATTERNS WITH KOTLIN; BUILD SCALABLE APPLICATIONS USING TRADITIONAL, REACTIVE, AND CONCURRENT DESIGN PATTERNS IN KOTLIN.*, vol. 0. Birmingham: PACKT PUBLISHING., 2018.
- [12] I. N. Karlina, "Kotlin, Bahasa Pemrograman Aplikasi Berbasis Android," *qwords.com*, Jan. 07, 2021.
- [13] T. Ismi, "Kenalan sama Kotlin, Bahasa Pemrograman Android yang Mudah Dipahami," *glints.com*, Jan. 13, 2021.
- [14] P. Bourque, J. W. Moore, L. Tripp, K. Shyne, B. Pflug, and G. Tremblay, "Guide to the Software Engineering Body of Knowledge A Straw Man Version," 1998.
- [15] K. Langsari, S. Rochimah, and R. J. Akbar, "Measuring Performance Efficiency of Application applying Design Patterns and Refactoring Method," *IPTEK Journal of Proceedings Series*, vol. 4, no. 1, p. 149, Jan. 2018, doi: 10.12962/j23546026.y2018i1.3527.
- [16] J. Kouraklis, "MVVM as Design Pattern," in *MVVM in Delphi*, Apress, 2016, pp. 1–12. doi: 10.1007/978-1-4842-2214-0_1.



- [17] B. A. Santoso, "Arsitektur MVC vs MVP vs MVVM di Pemrograman Android," *lobothijau.medium.com*, Oct. 18, 2019.
- [18] "ISO/IEC 25010:2011(en) System and software quality models," *iso.org*, 2011.
- [19] I. KUCHERENKO, *MASTERING HIGH PERFORMANCE WITH KOTLIN; OVERCOME PERFORMANCE DIFFICULTIES IN KOTLIN WITH A RANGE OF EXCITING TECHNIQUES AND SOLUTIONS*. Birmingham: PACKT PUBLISHING, 2018.
- [20] U. Wahyu, "PERBANDINGAN PERFORMA APLIKASI ANDROID NATIVE MENGGUNAKAN ARSITEKTUR MVP DAN MVVM," POLITEKNIK NEGERI BANDUNG, Bandung, 2019.