A Comparative Analysis of Test Automation Frameworks Performance for Functional Testing in Android-Based Applications using the Distance to the Ideal Alternative Method

Calysta Merina
Department of Informatics
Syarif Hidayatullah State Islamic University
Jakarta, Indonesia
calysta.merina13@mhs.uinjkt.ac.id

Nenny Anggraini
Department of Informatics
Syarif Hidayatullah State Islamic University
Jakarta, Indonesia
nenny.anggaraini@uinjkt.ac.id

Nashrul Hakiem
Department of Informatics
Syarif Hidayatullah State Islamic University
Jakarta, Indonesia
hakiem@uinjkt.ac.id

Sandra H. Afrizal
Department of Public Health Science
Universitas Indonesia
Depok, Indonesia
sandra62@ui.ac.id

Abstract—**The importance of functional testing is to guarantee the quality of an Android application to enhance the user experience so as to be more secure and reliable under market pressure, forcing many companies to build high quality and high performance applications that can be released in a short time. Thus, the testing process should be automated to make sure that applications run quickly and effectively. There are several test automation frameworks available, specifically designed for Android applications which offer features and efficient performance to test applications more effectively, such as Espresso, Calabash and Appium. The purpose of this research is to find the best framework based on performance with certain parameters and sub-parameters. The authors ran function tests on each framework with several test cases and provisions from each framework in an Android-based application. Reports were generated from each of the frameworks to collect data in the form of test cases covered, time complexity, execution speed, and element inspection for automated testing progress parameters. These were compared with a matrix that defines tool usability parameters using The Distance to the Ideal Alternative method (DIA) with a weight for each parameter and sub-parameter. The results show that Calabash has the best performance among the other frameworks with a 0 value for the DIA Method that indicates that Calabash has zero distance to the ideal alternative.**

Keywords— Software Testing, Android Testing, Functional Testing, Android Application, Test Automation Framework, The Distance to the Ideal Alternative method

I. INTRODUCTION

Mobile applications simplify the daily activities of users and provide many conveniences in many aspects, as shown by over 1 million apps available in the Google Play Store as well as other sources [1]. Due to growing user demand mobile applications have become more complex. The real obstacle is not building the application itself, but to ensure its effectivity and stability [2]. Some mobile applications are developed in the face of many limitations, such as budget, time, and resources. Applications are often released in a limited time with pressure from market demand. Mobile applications are also required to be able to handle a variety of systems and various user actions correctly, and to test all these aspects take a lot of time. It is not a surprise that applications often experience constraints and react badly to unexpected user actions. Such things can reduce the trust of users, reduce the users perceived convenience, and harm mobile application developer companies [1].

Therefore, it is necessary to apply a functional test or acceptance test during the development of mobile applications, in order to improve the security and reliability and to define and apply new testing techniques specifically designed for mobile applications [3]. Testers have two options, being manual testing or automation using test scripts [2]. Due to the market demand for stability and effectivity of mobile application to become available quickly and efficiently, it is not possible to achieve these goals through manual testing. Hence testers prefer automated testing methodology [4].

In the automation testing framework, the method plays a main function in determining how to configure automation jobs in order to maximise software products. It assists in accepting what, where, how and when to apply the factors needed for automation. One of the main goals for the automation testing method is finding precisely the best test framework [4]. According to [1] and [5], there are five best test automation frameworks to perform functional tests on Android-based applications. The current authors chose the three most commonly used test automation frameworks, namely, Espresso, Calabash, and Appium that will be tested by the authors one by one.

The authors are performing functional tests on the Weedu Application, an Android-based application developed by PT Nusantara Beta Studio, using the three best test automation frameworks and the most frequently used based on performance. These are Espresso, Calabash, and Appium. The parameters for testing the performance of the frameworks are Automated Testing Progress and Tool Usability [6]. The authors will use The Distance to the Ideal Alternative (DIA) method as a decision support method in determining the best test automation framework in terms of performance because according to research done by [7], the DIA method outperforms the Technique for Order of Preference by

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

Similarity to an Ideal Solution (TOPSIS) method in terms of ranking abnormalities. DIA also outperforms Simple Additive Weighting (SAW) and Weighted Product (WP) methods in terms of the difference in rank values, and according to research conducted by [8] the DIA method has a greater percentage change of rank compared to SAW.

II. Literature Review

A. *Software Testing*

Software testing is a part of software engineering methodology. The Software testing has been used make a qualified software effectively and efficiently [9]. There are two types of software testing, namely white-box testing and black-box testing [10].

White-box testing is a test that processes the details of design, using the structure of the program design based on the program code. The source code testing may be measured by using graph, data flow testing, and Cyclomatic complexity calculation.

While black-box testing or functional testing is a software testing method that tests application functionality. Test cases are built around specifications and requirements of the system. There is no need to know about the internal structure of the test object.

Software testing method can be implemented into many platforms or operating systems, e.g. Web based, mobile applications and other platforms. It also can be implemented into several level of testing which are the unit testing, integration testing, acceptance testing [3].

B. *Automated Testing*

Automation testing is an extension of the manual testing method with special software testing [11]. The automation performs testing with little or no intervention at all from the software tester. An automation testing framework is a layered structure which provides a mechanism for interacting in orders to achieve common objectives. The framework also provides a standardization to modify, add, and remove scripts and their functions. This goal can be achieved by choosing an automated software testing framework. Selecting the right framework may help the budget, time, and other resources' saving [12]. Some of the well-known frameworks for reliability are Espresso, Calabash, and Appium.

C. *Comparison Framework Matrix*

There are several differences in each test automation framework (Espresso, Calabash, and Appium) in terms of Tool Usability which can be seen in the Table I.

TABLE I. Comparison of each Framework Matrix in terms of Tool Usability

Feature	Espresso	Calabash	Appium
Android	Yes	Yes	Yes
IOS	No	Yes	Yes
Mobile Web	No	Yes	Yes
Scripting Language	Java	Ruby	Almost Everything
Parallel Execution	Yes	No	Yes

D. *Android*

Android can simply be defined as software used on a mobile device that includes the operating system, middleware, and key applications released by Google. Android covers the entire application, from the operating system to the development of the application itself. Application development on this Android platform uses the basic Java programming language. The Android application development platform is open source [13].

E. *The Distance to the Ideal Alternative*

The Distance to the Ideal Alternative (DIA) method is a part of the Multiple Attribute Decision Making (MADM) method. The MADM is a method which has been used to find the optimal alternative from several alternatives with some criteria. There are some types of MADM namely Simple Additive Weighting (SAW), Weighted Product (WP), ELimination and Choice Expressing REality (ELECTRE), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), and Analytic Hierarchy Process (AHP) [14]. The DIA method is an extension of the TOPSIS, it also determines a positive and negative ideal values of each attribute. The DIA determines the Positive Ideal Alternatives (PIA) by using Manhattan distance, it has minimum (D_j^+) and maximum (D_j^-) of R_i as a ranking conclusion [8].

The following are the main steps of the DIA algorithm [7]:

- Determine the decision matrix with assigned weight.
- Construct the normalised decision matrix R. Each element r_{ij} of the Euclidean normalised decision matrix R can be calculated as follows:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \quad (1)$$

- Construct the weighted normalised decision matrix. This matrix V is calculated by multiplying each column of the matrix R with its associated weight wi.

$$V = \begin{bmatrix} v_{11} & . & . & v_{1m} \\ . & . & v_{ij} & . \\ . & . & . & . \\ v_{1n} & . & . & v_{nm} \end{bmatrix} = \begin{bmatrix} r_{11} * w_1 & . & . & r_{1m} * w_m \\ . & . & r_{ij} * w_i & . \\ . & . & . & . \\ r_{n1} * w_1 & . & . & r_{nm} * w_m \end{bmatrix} \quad (2)$$

- Determine positive and negative ideal values of each attribute.

$$A^+ = \max_j v_{ij} = \left[v_1^+, v_2^+, \ldots, v_i^+, \ldots, v_m^+\right] \quad (3)$$

$$A^- = \min_i v_{ij} = \left[v_1^-, v_2^-, \ldots, v_i^-, \ldots, v_m^-\right] \quad (4)$$

- Calculate the Manhattan distance to the positive and negative attribute.

$$D^+{}_j = \sum_{i=1}^{m} \left| v_{ij} - a^+{}_i \right| \quad (5)$$

$$D^-{}_j = \sum_{i=1}^{m} \left| v_{ij} - a^-{}_i \right| \quad (6)$$

- Determine the (PIA) which has minimum D^+, and maximum D^-.

$$PIA = \left\{ \min(D_j^+), \max(D_j^- \right\} \quad (7)$$

- The distance of an alternative to the PIA is calculated as follows:

$$R_i = \sqrt{(D_i^+ - \min(D_i^+))^2 + (D_i^- - \max(D_i^-))^2} \quad (8)$$

A set of alternatives can now be ranked according to the increasing order of R_i.

III. Related Works

The authors used several sources for the literature study. Paper [4] discussed the advantages and disadvantages in general of test automation frameworks such as Robotium, Espresso, Calabash, Ui Automator and Appium, including the features and capabilities they have in common. In that paper Appium stood out in terms of platform independence, support for hybrids, web applications and support for multiple programming languages.

Paper [2] explained the use of the Calabash framework in detail from general information to an explanation of an example of making a test case in the Calabash framework. The authors of that paper compared Calabash with other test automation frameworks in terms of features in general, and the result was that Calabash had a higher quality than the other test automation frameworks.

Paper [1] explained the translation of each test automation framework in the Table I. The paper did not clearly explain the specific test framework suggested for use in testing. It only explained that to find the best test automation framework, it is necessary to develop criteria such as the ability to perform multi device testing, testing can be done on real devices or a simulator, test reports can be provided with good format, and others.

Research in [8] created a guideline for authors to study and implement the Distance to the Ideal Alternative method (DIA) as the author of that research implemented this method along with the SAW method to determine scholarship grantee in a website system, and concluded that the DIA method is a better MADM approach compared to SAW.

IV. Methodology

The flow of analysis in this work began with data collection using the literature study, then applying an implementation method which identified the application on which to perform automatic functional testing. This led to design of the test cases and continued with the emulator settings for test preparation.

The current authors performed a functional testing process automatically on Android-based mobile applications with the Espresso Framework, Calabash Framework, and Appium Framework with various stages and conditions of each framework. From the results of the automated functional tests and reports produced by each framework, the authors analysed and compared the results of the automated functional testing of each framework in terms of performance with The Distance to the Ideal Alternative method. The whole of the above stages are represented in Figure 1.

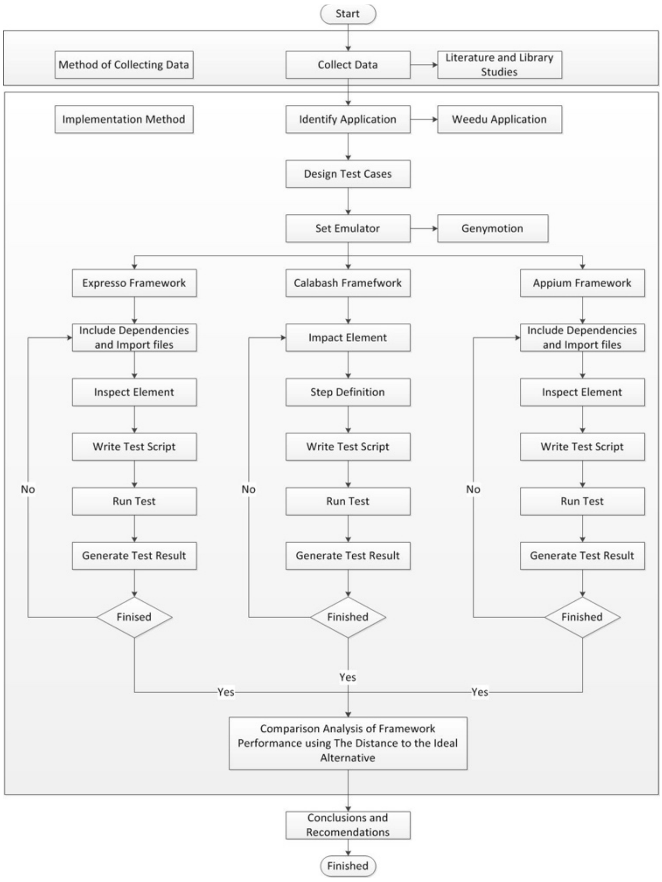

Fig. 1. The Stages of Analysis

V. Implementation

A. Identify The Application

The application that is used in this research is the Weedu Application which is an Android-based application developed by PT Nusantara Beta Studio. The Weedu application is an educational app that provides a feature to learn online. It has a simple interface and is easy to understand by the user. In this research, the authors chose three features in the Weedu application, namely the login feature, contact us feature, and the edit profile feature which will be tested in a functional test. The authors chose these three features because they are common features in every Android application.

B. Design Test Cases

In designing the test cases for each feature, the authors divided them into two forms, positive test cases and negative test cases. For this application, there are 10 test cases which are (login success and login failed) for the login feature, (contact us success, contact us empty, contact us subject empty, contact us message empty) for the contact us feature, and (edit profile success, edit profile username empty, edit profile telephone empty, edit profile empty) for the edit profile feature.

C. Setting the Emulator

The Android emulator used by authors is Genymotion as it is simple and the most common emulator used for automated testing.

D. Automated Functional Testing

In this stage, the authors performed functional tests for each test automation framework in accordance with the provisions and rules of each framework. The following are the steps of each framework in Figure 2, Figure 3, and Figure 4.

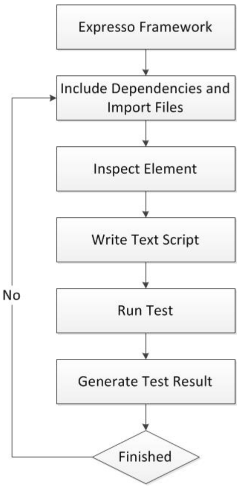

Fig. 2. Testing Stages in Espresso

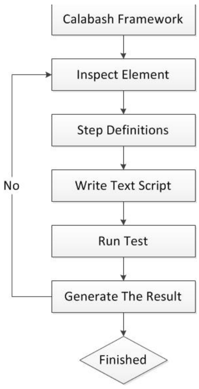

Fig. 3. Testing Stages in Calabash

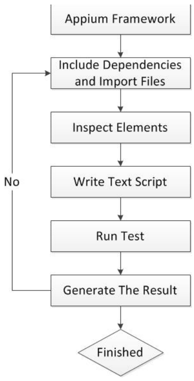

Fig. 4. Testing Stages in Appium

E. Generate Test Result

From these stages, the authors obtained data as seen in the Table II that will be compared to determine the best test automation framework using The Distance to the Ideal Alternative method.

TABLE II. TEST RESULT DATA

Framework Name	Test Case Covered	Time Complexity (seconds)	Execution Speed (seconds)
Espresso	3	19.342	0.509
Calabash	10	267.361	3.3007530864
Appium	10	431.021	3.652720339

VI. ANALYSIS AND DISCUSSION

After the authors finished performing the automated functional testing on the three frameworks, an analysis was performed of the performance comparison using The Distance to the Ideal Alternative (DIA) method with the following steps as depicted in Figure 5.

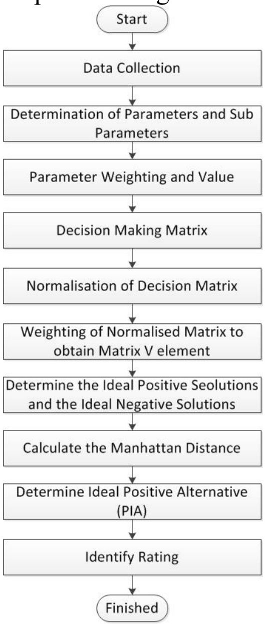
Start
Data Collection
Determination of Parameters and Sub Parameters
Parameter Weighting and Value
Decision Making Matrix
Normalisation of Decision Matrix
Weighting of Normalised Matrix to obtain Matrix V element
Determine the Ideal Positive Seolutions and the Ideal Negative Solutions
Calculate the Manhattan Distance
Determine Ideal Positive Alternative (PIA)
Identify Rating
Finished

Fig. 5. Stages of DIA Method

The complete equation for each stage in The Distance to the Ideal Alternative (DIA) method is already shown in the Literature Review. The authors defined the parameters and sub-parameters based on the literature studies written by [6] and [15]. The automated testing progress parameter has a weight of 0.6, whereas the tool usability parameter has a weight of 0.4. The following figure describes the parameters and sub parameters as a benchmark comparison between the frameworks.

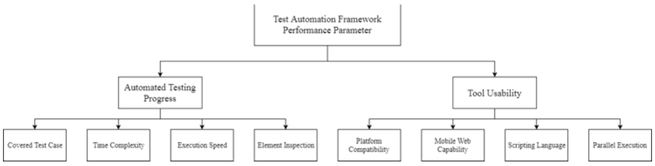

Fig. 6. Parameters and Sub-Parameters

After the analysis, the authors obtained the results of the comparison test automation frameworks using The Distance to the Ideal Alternative method on each of the parameters in the Table III and Table IV.

TABLE III. AUTOMATED TESTING PROGRESS RESULT

Framework Name	R_i	Ranking
Espresso	0.00567	**2**
Calabash	0	**1**
Appium	0.18973	**3**

After obtaining the Ri value and the ranking of the two predefined parameters, the authors undertook a re-analysis to compare the values of the two parameters using The Distance to the Ideal Alternative (DIA) method to produce a more valid final value to measure the performance from each test automation framework. The result is shown in Table V.

TABLE IV. TOOL USABILITY RESULT

Framework Name	R_i	Ranking
Espresso	0.98284	**3**
Calabash	0.38284	**2**
Appium	0	**1**

TABLE V. THE FINAL RESULT

Framework Name	R_i	Ranking
Espresso	0.33941	**2**
Calabash	0	**1**
Appium	0.48686	**3**

Figure 7 describes that Calabash has the best performance among other test automation frameworks because of its ability to perform tests more effectively and faster based on the two parameters, and because the Automated Testing Parameter is the more relevant and important parameter to measure test automation framework performance.

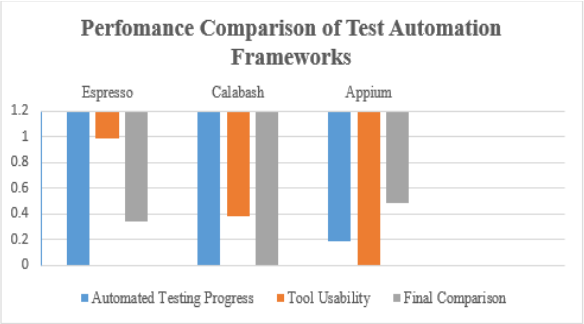

Fig. 7. Performance Comparison of Test Automation Frameworks

VII. CONCLUSION

The authors have successfully designed the test cases and run functional testing on all three frameworks. Calabash is in the first rank with a Ri value of 0, which indicates no distance between these values and an ideal alternative. Thus it can be concluded that Calabash is an ideal alternative test automation framework in terms of performance with automated testing progress and tool usability parameters. The second rank is occupied by Espresso with a Ri value of 0.33941, and the last rank is Appium with a Ri value of 0.48686.

For further research, the authors suggest to undertake a comparison with other test automation frameworks, especially for testing mobile applications, to perform functional testing on other platforms, to do functional testing with test automation frameworks that are integrated into a Continuous Integration Tool such as Jenkins, CI Circle or Travis CI.

REFERENCES

[1] P. Pareek, R. Chaturvedi, and H. Bhargava, "A Comparative Study of Mobile Application Testing Frameworks," in *BICON 2015*, 2015, pp. 4–5.

[2] M. K. Kulkarni and P. S. A, "Deployment of Calabash Automation Framework to Analyze the Performance of an Android Application," *J. Res.*, vol. 02, no. 03, pp. 70–75, 2016.

[3] P. Blundell and D. T. Milano, *Learning Android Application Testing*, vol. 1. Packt Publishing Ltd., 2015.

[4] A. M and S. KN, "Comparative Study on Different Mobile Application Frameworks," *Int. Res. J. Eng. Technol.*, pp. 1299–1300, 2017.

[5] L. Shao, "Top 5 Android Testing Frameworks with Examples," 2015. .

[6] S. Khoria and P. Upadhyay, "Performance Evaluation and Comparison of Software Testing Tools," *VSRD Int. J. Comput. Sci. Inf. Technol.*, vol. 2, no. 10, pp. 801–808, 2012.

[7] P. N. Tran and N. Boukhatem, "The Distance To the Ideal Alternative (DiA) Algorithm for Interface Selection in Heterogeneous Wireless Networks," *Proc. 6th ACM Int. Symp. Mobil. Manag. Wirel. access - MobiWac '08*, p. 61, 2008.

[8] R. A. Purnamasari, "Penentuan Penerima Beasiswa dengan Metode Simple Additive Weighting dan Metode The Distance to The Ideal Alternative," Universitas Jember, 2015.

[9] S. Dalal and R. S. Chhillar, "Software Testing-Three P ' S Paradigm and Limitations," *Int. J. Comput. Appl. (0975 – 8887)*, vol. 54, no. 12, pp. 49–54, 2012.

[10] R. Pressman and B. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill, 2014.

[11] Technopedia, "What is Automated Testing? - Definition from Techopedia," 2017. .

[12] A. Bhargava, *Designing and Implementing Test Automation Frameworks with QTP*. Packt Publishing Ltd., 2013.

[13] T. EMS, *Pemrograman Android dalam Sehari (Indonesian Edition)*. Jakarta: : PT Alex Media Komputindo, 2015.

[14] Fadwa Gamal Mohammed Al-Azab and M. A. Ayu, "Web based multi criteria decision making using AHP method," in *Proceeding of the 3rd International Conference on Information and Communication Technology for the Moslem World (ICT4M) 2010*, 2010, pp. A6–A12.

[15] R. Rattan and Shallu, "Performance Evaluation and Comparison of Software Testing Tool," *Int. J. Inf. Comput. Technol.*, vol. 3, no. 7, pp. 711–716, 2013.