Boundary Value Analysis Testing on Augmented Reality of Indonesian Fruit Recognition at Mekarsari Tourist Park using Cloud Method on Android Mobile Devices

Dewi Agushinta R¹, Hustinawaty², Ihsan Jatnika³, Henny Medyawati⁴

¹, ⁴Information System, Faculty of Computer Science and Information Technology
², ³Informatics Engineering, Faculty of Industrial Technology
Gunadarma University
Jl. Margonda Raya No. 100 Pondok Cina, Depok 16424, West Java

Abstract. Software testing is a process of implementing a program with the aim of finding an error. A good test case is if the test has the possibility of finding an uncovered error. A successful test is if the test finds an error that was not initially found. One of the testing types available is black box testing. This paper proposes testing using black box testing technique. The black box testing method consists of several ways including equivalence partitioning, boundary value analysis, comparison testing, sample testing, robustness testing, and others. Among the many methods of testing, Boundary Value Analysis was chosen in this study. Boundary Value Analysis is a method of testing by determining the value of the lower limit and upper limit of the data that will be tested. This test is performed on the functions of Augmented Reality prototype of Indonesia fruit recognition by using the cloud method on Android mobile devices. From testing the distance of marker objects to Android mobile devices on cloud recognition using an Android camera shows that the higher the augmentable rating of the target image and the more the number of markers features detected, the easier the image will be tracked by AR. If the distance between the camera and the real object gets farther away, then the virtual object cannot be displayed. Testing with mobile devices using HSDPA, 3G and WIFI networks connecting a cloud database server to display virtual objects shows the average results of devices that use WIFI networks provide the fastest performance.

1. Introduction
Technology development in mobile phones recently is very rapid. Nowadays, a mobile phone is not only used as a voice communication tool, but also as a tool to facilitate users in their daily lives. The facilities for accessing the internet, e-mail, organizer, games and so on that can be directly used via mobile phone are some examples of services offered to Android-based Mobile users. In its development, Android can also support Augmented Reality technology, which is a technology that combines three-dimensional virtual objects into a real environment and displays them in real time [1]. Unlike virtual reality which completely replaces reality, Augmented Reality simply adds or completes the reality. Many developers make Augmented Reality-based applications as information, learning, promotion and marketing media. Augmented Reality prototype of Indonesia fruit recognition by using cloud method on Android mobile devices is one of AR application we developed. The method used is cloud technology [2]. The database of fruit metadata is stored in a cloud server database using Vuforia technology [3]. The implementation of the AR prototype has already made [4]. After the prototype was coded, it is necessary to do the test to make sure all processes are running as desired. Testing is a process of implementing a program with the aim of finding an error. A good test is if the test has the possibility of finding an uncovered error. A
successful test is if the test finds an error that was not initially found. The main purpose of testing is to design a test that systematically finds types of errors with minimum effort and time. Boundary Value Analysis has been used for testing some system and application. A Management Information System Counselling Final Assignment is tested using BVA [5]. BVA has investigated the problem of generating test cases for non-numerical values and fail to test non-numerical variables [6]. This paper investigated the BVA testing on the Augmented Reality prototype of Indonesia fruit recognition at Mekarsari tourist park using cloud method on Android mobile devices.

2. Literature Review

2.1. Software Testing
Software testing is urgently needed to ensure that software or applications that are being created can run in accordance with the expected functionality. Software developers or testers must prepare special sessions to test the programs that have been created so that errors or deficiencies can be detected early and corrected as soon as possible. The testing itself is a critical element of Software Quality Assurance (SQA) and is an integral part of the Software Development Life Cycle as well as analysis, design, and coding [7]. Software testing must be done in the process of software engineering. Some software testing strategies have been proposed. All of them provide testing templates for software makers. In this case, all must have general characteristics in the form of [8]:

- Testing starts at the module level and works out towards integration on computer-based systems
- A different testing technique according to different points in time
- Testing is carried out by software developers and for large projects by independent group testing
- Testing and debugging are different activities but debugging must be accommodated in each testing strategy.

Software testing is an element of a broader topic that is often interpreted as Verification and Validation (V&V). Verification refers to a collection of activities that ensure the software has implemented a specific function. Validation refers to a collection different from activities that ensure the software that has been built can be traced to customer needs. The definition of V&V includes many SQA activities, including formal technical reviews, quality and configuration audits, performance monitoring [8].

There are several types of software testing, which are [9]:

1. White box testing is a test based on checking the design details, using the control structure of the program design procedurally to divide the test into several test cases. At first glance, it can be concluded that white box testing is a guide to get the program 100% correct. The weakness of white box testing is of the large type software, this white box testing method is wasteful because it involves a lot of resources to do it [10]

2. Black box testing is a test that focuses on the functional specifications of the software, the tester can define a set of input conditions and test the functional specifications of the program.

2.2. Black Box Testing
Black box testing focuses on functional specifications of software. The tester can define a set of input conditions and test the program’s functional specifications. Black box testing is not an alternative solution to white box testing but is more of a complement to testing things that are not covered by white box testing. Black box testing tends to find the following:

1. Functions that are incorrect or non-exist.
2. Interface errors.
3. Errors in data structures and database access.
4. Performance errors.
5. Initialization and termination errors.

There are currently many methods or techniques for implementing black box testing, such equivalence partitioning, boundary value analysis or limit testing, comparison testing, sample testing, robustness
testing, behavior testing, requirement testing, performance testing, endurance testing, and cause-effect relationship testing.

2.3. Boundary Value Analysis

Boundary value analysis is one of the black box testing that does the testing at the upper and lower value limit expected in the application. There are few principles underlying the boundary value analysis (BVA), such as [11]:

1. Many errors occur at input error
2. BVA allows selecting the testing case that tests the input value limit
3. BVA is a complement of equivalent partitioning. More to select the elements in the equivalent class at the limit value class.

3. Result and Discussion

This stage is a trial analysis process for Augmented Reality prototype of Indonesia fruit recognition by using cloud method on Android mobile devices. This trial analysis phase was conducted through some testing, the marker object distance and feature testing through an Android mobile device on a cloud by using an android camera and testing with mobile devices using HSDPA, 3G and WIFI networks connect the cloud database server to display virtual objects.

3.1. Marker Objects Distance Analysis against Mobile Devices

In this experiment, the real objects that are used as markers have a size of 23x15 cm, placed at a height of 120 cm above the floor and placed on the branch of the fruit tree being directed, while the camera is directed and placed at an altitude of 125 cm. In this test, the distance changing testing was carried out.

![Figure 1. Marker real object image.](image)

(a) Marker (b) Feature Distribution

The marker used is as in Figure 1(a) and the number of objects features calculation used is in Figure 1(b).

In the system analysis testing that has been made by researchers using several fruit collections as markers, each marker has its own feature value called Augmentable. This feature is automatically given by the Vuforia system through rating calculations when the image upload process becomes a marker. An augmentable rating defines how well an image can be detected and tracked using Vuforia SDK. This rating is displayed in the target manager and returned for each upload target via the web API. An augmentable rating can range from 0 to 5 for each given image. The higher the augmentable rating of the target image, the stronger the detection and tracking capabilities.
A zero rating indicates that the target is not tracked at all by the AR system, while the 5-star rating shows that an image is easily tracked by the AR system [3]. Based on the detection results of certain augmentable features, the smaller the augmentable value of the marker, the more difficult it is for the camera to detect marker object features. Table 1 explained the detected features which are directly proportional to the augmentable marker for markers. On the 50 mm to 100 mm distance testing, the marker image was detected completely so that it was still detected well. The features detected by the camera at farther distances will accumulate at the point that detects the most features, so the features stored by the system will be much different from what is seen by the camera. Based on the tests that have been carried out at 150 mm between the camera and real objects, virtual objects cannot be displayed. This is because at that distance the features of real objects that are seen by the camera have accumulated too much.

Table 1. Feature comparison detected by the camera.

| Marker Used | Augmentable Rating | The camera distance to the object (mm) | Total feature | Result |
|-------------|--------------------|---------------------------------------|---------------|--------|
| Starfruit   | ***                | 100                                   | 35            | Detected |
| Rambutan    | ****               | 100                                   | 63            | Detected |
| Pineapple   | ***                | 100                                   | 33            | Detected |
| Durian      | ****               | 100                                   | 79            | Detected |
| Guava       | ****               | 100                                   | 72            | Detected |
| Melon       | ****               | 100                                   | 82            | Detected |
| Passionfruit| ***                | 100                                   | 44            | Detected |
| Mango       | ***                | 100                                   | 31            | Detected |
| Orange      | **                 | 100                                   | 24            | Not Detected |
| Srikaya     | ***                | 100                                   | 34            | Detected |
| Jackfruit   | ****               | 100                                   | 76            | Detected |
| Mangosteen  | ***                | 100                                   | 42            | Detected |
| Longan      | **                 | 100                                   | 19            | Not Detected |
| Malay apple | **                 | 100                                   | 18            | Not Detected |
| Dukuh       | **                 | 100                                   | 28            | Detected |

Note: (*) augmentable rating

The lighting quality testing, colors, textures and images that are on the marker greatly affect the speed and accuracy of the data when it is detected by the camera. In addition to light intensity, the marker distance with the camera also takes huge effect in the process of running this program where the marker that has been detected by this camera will later be compared with the marker data that has become the reference. If the camera is too close or too far, the camera cannot read the marker properly, so the program cannot recognize the marker.

3.2. Mobile Device Connection Testing
The Augmented Reality prototype needs the internet network to display virtual objects because all data features are stored on the Vuforia server database. The program is designed to download virtual objects that will be displayed when it detects the similarity of the object features seen by the device's camera and features stored on the Vuforia server. The marker feature database is stored separately from the main program, so adding or modifying data feature and content can be done easily. Figure 2 is a GUI real object uploading image that will be used as a marker.
Testing the device connected to the Vuforia server used three different network modes to display virtual objects shown in Table 2.

| # Testing | HSDPA | 3G  | Wi-Fi |
|-----------|-------|-----|-------|
| 1         | 8.2   | 7.9 | 3.8   |
| 2         | 8.1   | 7.7 | 4.7   |
| 3         | 6.7   | 7.2 | 4.4   |
| 4         | 5.3   | 7.6 | 5.8   |
| 5         | 9.8   | 6.8 | 5.2   |
| 6         | 6.5   | 6.3 | 5.5   |
| 7         | 8.7   | 7.9 | 3.1   |
| 8         | 6.1   | 8.2 | 5.3   |
| 9         | 8.4   | 10.8| 5.3   |
| 10        | 8.8   | 9.8 | 4.5   |
| Average time (s) | 7.66 | 8.02 | 4.76 |

Based on the tests carried out in Table 2 using three different internet networks, the average results for displaying virtual objects were different, devices that use HSDPA networks obtained an average of 7.66 seconds; devices that use 3G networks obtained an average of 8.02 seconds; devices that use Wi-Fi networks have an average of 4.76 seconds. The testing is done at almost the same time. Network factors that connect Android devices to servers over the internet network take huge effect. The mobile device testing showed that WIFI connection gives the fastest average time.

4. Conclusion
Boundary Value Analysis testing was carried out at the functionality Augmented Reality prototype of Indonesia fruit recognition by using the cloud method on Android mobile devices. It showed that the black box testing method is one method that is easy to use because it only requires different average results for displaying virtual objects. The results of the marker object distance and feature testing and mobile devices testing give a good performance.

5. Reference
[1] Ronald T A 1997 A Survey of Augmented Reality In Presence: Teleoperators and Virtual Environments 6(4) 355-85
[2] Dewi A R, Ihsan J, Henny M and Hustinawaty 2017 A Method of Cloud and Image Based Tracking for Indonesia Fruit Recognition Proceeding The IEEE 3rd International Conference on Engineering, Technologies and Social Sciences Paper ID: 119
[3] Dewi A R, Ihsan J, Henny M and Hustinawaty 2017 Implementasi Database Cloud Buah
Pada Vuforia Prosiding Seminar Nasional Hasil Penelitian dan Pengabdian Lembaga Penelitian Pengembangan Pembelajaran & Pengabdian Kepada Masyarakat IIB DARMAJAYA 502-12

[4] Dewi A R, Ihsan J, Henny M and Hustinawaty 2018 Augmented Reality Design of Indonesia Fruit Recognition International Journal of Electrical and Computer Engineering (IJECE) 8(6) Part I

[5] Doni A 2018 Pengujian Kotak Hitam Boundary Value Analysis Pada Sistem Informasi Manajemen Konseling Tugas Akhir Indonesian Journal on Networking and Security 7(1) 13-18

[6] Anupriya J, Sachin S, Seema S and Deepti J 2010 Boundary Value Analysis for Non-Numerical Variables: Strings Oriental Journal of Computer Science & Technology 3(2) 323-30

[7] Mingtao S 2010 Software Functional Testing from the Perspective of Business Practice Computer and Information Science 3(4) 49-52

[8] Bhat A and Quadri S M K 2015 Equivalence Class Partitioning and Boundary Value Analysis - A Review 2nd International Conference on Computing for Sustainable Global Development (INDIACom) 1557-62

[9] Mohd Ehmer K 2011 Different Approach to Blackbox Testing Technique for Finding Error International Journal of Software Engineering & Applications (IJSEA) 2(4) 31-40

[10] Srinivas N and Jagruthi D 2012 Blackbox and Whitebox Testing Techniques - A Literature Review International Journal of Embedded Systems and Applications (IJESA) 2(2) 29-50

[11] David J C 2008 A Review of Boundary Value Analysis Techniques The Journal of Defense Software Engineering 19-22

Acknowledgments
This work is partially supported by Indonesia Ministry of Research Technology and Higher Education-Kemenristekdikti Grant No. 023 / KM / PNT / 2018.