Blackbox Testing on Virtual Reality *Gamelan Saron* Using Equivalence Partition Method

Richard Gunawan[1], Yohanes Priadi Wibisono[*2], Clara Hetty Primasari[3], Djoko Budiyanto[4], Mutiara Cininta[5]
Program Studi Sistem Informasi, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta, Sleman 55281, Daerah Istimewa Yogyakarta, Indonesia
Email: [1]richardgunawan100@gmail.com, [2]priadi.wibisono@uajy.ac.id, [3]clara.hetty@uajy.ac.id , [4]djoko.budiyanto@uajy.ac.id, [5]mutiara.cininta@uajy.ac.id

Abstrak. Pengujian Blackbox Pada Virtual Reality Gamelan Saron Menggunakan Metode Equivalence Partition. *Dalam pengembangan sebuah aplikasi, testing pada aplikasi sangat penting sebelum aplikasi dirilis. Testing berdasarkan kualitas dengan menggunakan Black Box mengutamakan pengujian fitur-fitur yang terdapat pada aplikasi, sehingga dapat menemukan permasalahan yang terjadi di aplikasi Gamelan VR Saron ini. Pada metode Black Box berbasis Equivalence Partition secara menyeluruh menguji dalam aspek penggunaan aplikasi Gamelan VR Saron. Dalam pengujian ini membutuhkan seleksi penggunaan berdasarkan test case, kemudian memastikan kualitas dari fitur-fitur yang tersedia serta menemukan error function yang bisa terjadi dalam aplikasi. Sehingga ujicoba pada aplikasi ini bisa menilai apakah telah sesuai dengan harapan dan direncanakan. Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa aplikasi ini dapat berjalan dengan baik tanpa adanya error sesuai dengan yang telah direncanakan. Hasil pengujian ini dapat dijadikan sebagai dokumentasi serta evaluasi untuk pengembangan aplikasi kedepannya.*
Kata Kunci: *black box, equivalence partition, test case, error function, user*

Abstract. Blackbox Testing on Virtual Reality Gamelan Saron Using the Equivalence Partition Method. *Testing is essential in application development because it helps identify and eliminate defects. One of the most used testing methods is Black Box testing, which involves deeply examining the application's functionality without knowing its internal workings. The Equivalence Partition method is frequently used in Black Box testing to divide input values into groups and select test cases from each group. Potential errors can be identified by testing the available features with appropriate test cases, and future improvements can be made to ensure seamless application performance. In addition, testing results also serve as documentation and research for future development. By using this method, the developers of the VR Gamelan Saron application can ensure that its quality meets user expectations to improve its quality to provide an optimal user experience. In summary, proper testing is crucial in application development, and the Equivalence Partition method is an effective tool for identifying and eliminating potential issues.*
Keywords: *black box, equivalence partition, test case, error function, user*

1. Introduction

Gamelan is one of Indonesia's traditional musical instruments from various unique musical instruments typical of Indonesian cultures, consisting of different percussion instruments such as *peking, Saron, kenong*, and *gong*, usually used as musical instruments to accompany *wayang* and *karawitan* [1]. *Gamelan* is popular in various countries. Due to its popularity, UNESCO stipulated *gamelan* as an Intangible Cultural Heritage / ICH or Intangible Cultural Heritage / WBTB in Paris's Intergovernmental Committee for Safeguarding the Intangible Cultural Heritage [2]. Therefore, *gamelan* is a valuable cultural asset for Indonesia that must be preserved. However, as time passes, most people rarely play this *gamelan* musical instrument [3], and even finding the instrument takes work. *Gamelan* is usually found in a particular community where people gather and play, making it inaccessible and pricey and avoiding other people who want to try it. Unfortunately, these factors make most young generation loses interest in it, not to mention technological developments that have introduced foreign cultures into Indonesia [4].

However, technological developments can still help Indonesian citizens become more familiar with the culture.

Metaverse has become a proof of development in Information Technology. It has started to be developed and used for various purposes. Metaverse has the potential to expand the physical world using augmented and virtual reality technologies that allow users to interact with virtual environments using avatars and holograms [5], providing an excellent opportunity to develop *gamelan* metaverse technology and introduce it to young generations. *Gamelan* Metaverse is a game simulation designed in a virtual world. This application is for everyone who wants to try or learn *gamelan* musical instruments because not all places have *gamelan*, and its prices are not affordable to own at home. Virtual reality can improve user experience, save time, and save costs. Virtual reality can also provide more interactive learning to users [6]. Playing a musical instrument in the natural and virtual worlds are different experiences. However, not all applications developed in the virtual world can be similar to the real world. Therefore, for the application to meet user satisfaction, the application of quality assurance or software testing is required.

Software testing is a stage of searching for information about software. This process is done by looking for errors in the software [7] to ensure the software meets the specifications. Software testing can be successful if no errors occur during the software testing process. One of the methods for conducting software testing is black box testing. Black box testing does not pay attention to the application's internal logic. Still, it focuses on functional software, which will be tested and checked to determine the output [8]. Black box testing has several techniques for testing software quality, one of which is equivalence partitions. Equivalence partitions reduce the number of cases tested and divide the input data into several ranges of values [9], which has several input types where the input will be divided into several partitions or based on the input that is similar to the output [10]. Equivalence partitions break the input domain of software being tested into several classes to obtain test cases [11]. This testing aims to verify that each application function operates according to the user's expectations. The author has chosen this testing approach due to its direct evaluation from the user's perspective, facilitating a simulated experience of the actual system use. By obtaining user experience or perspective, the author expects to provide constructive inputs to the application developers on the functionality that requires enhancements and those that meet the requisite standards.

2. Literature Study

2.1. Virtual Reality

Virtual reality is defined as stimulating a three-dimensional space with multisensory, immersive, and real-time characteristics that the user feels through input and output devices from three-dimensional devices [12]. Virtual reality technology is generally used worldwide through devices [13], so virtual reality requires supporting facilities during the application process. Based on this, the development of VR technology is widely used for various aspects of life [14].

2.2. Software Testing

Testing is an evaluation process to ensure the quality of the application, whether it meets the specified requirements or not. Testing is important to ensure the application is free from errors and bugs. The testing process has several techniques, including black, white, and grey box testing. These techniques will produce different test cases during the testing process. After the test case creation process proceeds to the testing stage in application testing according to the test cases that have been made, the results of this testing process will be a report indicating whether the application meets the requirements or still requires improvement in the system [15]. In the Software Development Life Cycle (SDLC), an application can be considered complete after the testing stages.

2.3. Black Box Testing

Black box testing is a test to complement white box testing with results that tend to be different from white box testing. Black box testing allows the tester to see the software's input and output without knowing the software code's structure. This test was conducted at the end of

software development to determine whether the software could work properly. The testers only know the input of a system and the expected or required output. They did not need to know the workings of the logic in the system. Black box testing has several advantages: testing is efficient, testers can do testing simply because testers do not need to know about the logic of the system, testers do not need to know about programming languages, and testing is seen from the end user's point of view. In black box testing, several techniques include equivalence partition, boundary value analysis, fuzzing, cause-effect graphs, orthogonal array testing, all-pair testing, and state transition testing [16].

2.4 Equivalence Partitions

Equivalence partition focuses on input and output values derived from a function. Testing using the equivalence partition divides an input from the system into equal classes, all designed to test the system based on all input values. Equivalence partition into test cases will produce a valid or invalid output where valid output occurs when a function being tested follows a predetermined test plan. In contrast, invalid results happen when the tested function does not follow the test plan [17].

3. Methodology

3.1. Research Stage

When conducting research, the author first sets the objectives and methods to be used. The author uses the Blackbox method with the equivalence partitions technique in this research. Then, the author complies with the test cases according to the purpose and method and conducts the test on the application. The results are collected and analyzed.

3.1.1. Determining Purpose and Method

3.1.1.1. Determining Purpose

The purpose of the research conducted by the author is to ensure that every function in the VR *gamelan Saron* application can run as planned before being released to the public. Equivalence partition-based black box is used to test each function in an application. The testing focuses on verifying the output generated from the input provided. This testing is considered fast because it does not look at the backend code [11]. By doing this research, the author can determine whether the VR *gamelan Saron* application still has errors and bugs or follows what has been planned.

3.1.1.2. Determining Method

Black Box testing is used as a software testing technique without having to investigate the logic of an application based on its specifications. This approach enables testing to directly test each functionality of the tested application [7]. Equivalence partitions represent one of the black box testing techniques, in which every function of an application is subdivided into several test cases [9]. In this research, each application function will be divided into a table of test cases to determine whether every application function is already in accordance with the plan or requires further enhancements.

3.1.2. Determining Test Case

The author tested sixteen functions divided into four test cases: sound with three functions, collision with three functions, main menu with three functions, and terrain with seven functions.

Figure 1 is the first test case: sound testing using the Unity application's velocity measurements and sound volume. Velocity calculations are obtained using the formula:

$$v = dx / dt \quad (1)$$

where dx is the frame and dt is the time. For example, if the *gamelan* is hit at a low speed (velocity <= 3.3 fps), it will make a slight sound (volume 0-0.33). If the *gamelan* is hit at medium speed (velocity> 3.3fps & velocity <=6.6fps), it will make a medium sound (sound volume 0.33-0.66). When the *gamelan* is hit at high speed (velocity >6.6fps& velocity <=10fps) it will make a loud sound (sound volume 0.66-1), as shown in Table 1.

Figure 1. *Gamelan Saron* with the beater

Table 1. Sound Test Case on VR *Gamelan Saron*

ID	Testing Descriptions	Expected Results
SA01	*Gamelan* is hit at low speed / v <= 3.3fps	The sound volume output is low/low sound. (Volume 0-0.33)
SA02	*Gamelan* is hit at medium speed (not too slow and not too fast) / v>3.3fps & v<=6.6fps	The sound volume output is medium sound. (Volume 0.33-0.66)
SA03	*Gamelan* is hit at high speed / hit hard / v>6.6fps & v<=10fps	The sound volume output is a high or loud sound. (Volume 0.66-1)

Figure 2. Saron Environment

Figure 2 shows that the VR *gamelan Saron* application has an environment similar to Figure 2. The test plan for the environmental tests the collisions implemented during the application development process by walking toward the wall and seeing the results. In Table 2, it is shown that there are three ways of walking: using the joystick found on the controller in VR, walking in the real world, and walking using the joystick and in the real world simultaneously, with the expected result being that the player cannot penetrate or pass through the wall.

Table 2. Test Case Collision in the VR *Gamelan Saron* Environment

ID	Testing Descriptions	Expected Results
EB01	Players walk using the joystick on the controller.	Players cannot penetrate or pass-through walls.
EB02	Players walk in the real world and do not use a joystick.	Players cannot penetrate or pass-through walls.
EB03	Players walk using the joystick and walk in the real world.	Players cannot penetrate or pass-through walls.

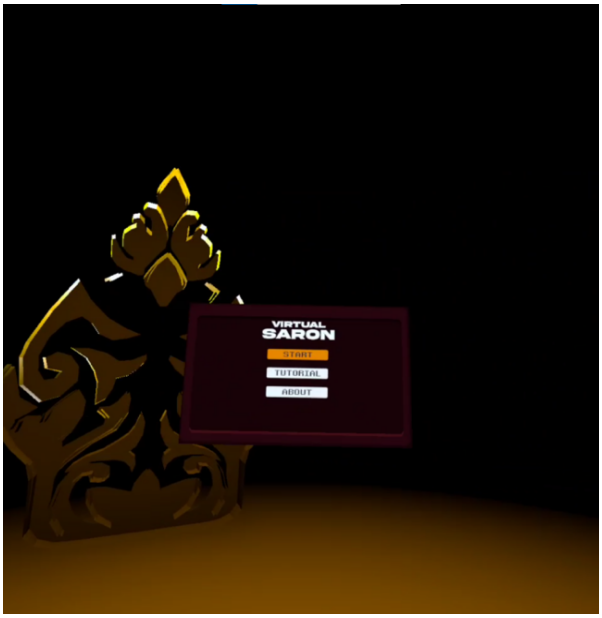

Figure 3. Main Menu on VR Gamelan Saron

Figure 3 shows that the VR *gamelan Saron* application has a main menu display that contains the "Start," "Tutorial," and "About" menus. The test plan ensures the system functions and runs as planned, as shown in Table 3. For example, when pressing the "Start" menu, the player will switch from the main menu to the *gamelan Saron* environment. When the player selects the "tutorial" menu, the player will be faced with a tutorial menu containing pictures and descriptions of how to play the VR *gamelan Saron* application. When selecting the "About" menu, a menu will be displayed, including the names of VR *gamelan Saron* developers.

Table 3. Test Case Main menu on VR *Gamelan Saron*

ID	Testing Descriptions	Expected Results
MC01	Players select the "START" menu option.	Players will be directed to an environment with a *Gamelan Saron*.
MC02	Players select the "TUTORIAL" menu option.	The system will display the Tutorial menu.
MC03	Players select the "ABOUT" menu option.	The system will display the About menu.

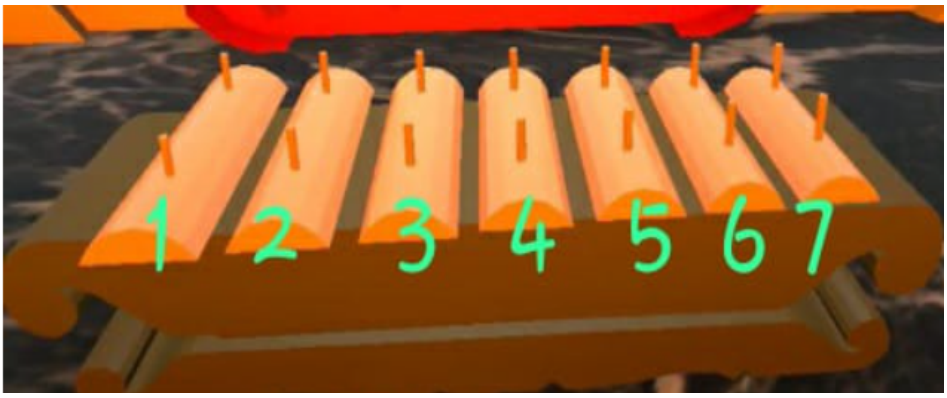

Figure 4. *Wilahan* on *Gamelan Saron*

In Figure 4, the test plan ensures that the *Wilahan* in the *gamelan* makes the appropriate sound sequentially from one to seven, as shown in Table 4. Wilahan in *gamelan Saron* can be sorted from the largest *Wilahan* to the smallest *Wilahan*, with the order of *Wilahan* number one to number seven with the tone "ji," "ro," "lu," "pat," "mo," "nem," "tu".

Table 4. Wilahan VR Gamelan Saron Sound Test Case

ID	Testing Descriptions	Expected Results
WD01	Players will hit the first “Wilahan” section.	"Wilahan" will issue a sound called *"ji"*
WD02	Players will hit the second “Wilahan” section.	"Wilahan" will issue a sound called *“ro”*
WD03	Players will hit the third “Wilahan” section.	"Wilahan" will issue a sound called *“lu”*
WD04	Players will hit the fourth “Wilahan” section.	"Wilahan" will issue a sound called *“pat”*
WD05	Players will hit the fifth “Wilahan” section.	"Wilahan" will issue a sound called *“mo”*
WD06	Players will hit the sixth “Wilahan” section.	"Wilahan" will issue a sound called *“nem”*
WD07	Players will hit the seventh “Wilahan” section.	"Wilahan" will issue a sound called *“tu”*

3.1.3. Testing

This step aims to obtain data based on the test cases prepared by testing the VR *gamelan Saron*. To obtain the user's actual experience, the test case is run on the VR *gamelan Saron* application installed on the Oculus Quest 2 device. The author conducted tests with the team's assistance to ensure the test case's accuracy with the tests performed in the field.

3.1.4. Collecting Testing Results

This step collects data based on testing results from the VR *gamelan Saron*. The first test scheme involves connecting the headset to the Unity software's debugger section to determine the speed and volume output suitability. The second is done natively on the Oculus Quest 2 headset and the other tests to gauge potential users’ experiences with the *gamelan Saron* VR application. The results of this test will be recorded in the test table.

3.1.5. Data Analysis

This stage aims to formulate conclusions by analyzing the data obtained from the results of tests carried out previously. The data obtained from the previous stages will then be matched with the test case to see whether the data is valid or not. The results of this data matching will be used later by the author to determine the suitability of VR *gamelan Saron* in simulating the sensation of playing *Saron* in the real world.

4. Result and Discussion

The tests on the VR *gamelan Saron* application conform to the description of the testing planned in the test case. The results of the tests will be listed in the results table, where if the results follow expectations, it can be concluded that the results are valid. If the results do not match the expectations, the results can be concluded as invalid.

4.1. Sound Testing Results on the VR *Gamelan Saron*

This test determines whether the sound meets expectations. Sound testing in this VR *gamelan* application used velocity and sound volume range found in the Unity application with a low-velocity range v <= 3.3 fps, medium velocity v > 3.3 fps and v <= 6.6 fps, and high-velocity v > 6.6 fps and v <= 10 fps with sound output with a scale of 0-1 with a range of low volume 0 - 0.33, medium volume 0.33 - 0.66 and high volume (0.66 - 1 volume).

Table 5. Sound Testing Results on the VR *Gamelan Saron*

ID	Testing Descriptions	Results	Expected Results	Conclusion
SA01	*Gamelan* is hit at low speed /v <= 3.3fps.	*Gamelan* makes a low sound after being hit. (Volume 0 - 0.33)	The sound volume output is a small/low sound. (Volume 0 - 0.33)	VALID
SA02	*Gamelan* is hit at medium speed (not too slow and not too fast) / v > 3.3fps and v <= 6.6fps	*Gamelan* is hit at medium speed (not too slow and not too fast) / v > 3.3fps and v <= 6.6fps	The sound volume output is a medium sound. (Volume 0.33 - 0.66)	VALID
SA03	*Gamelan* is hit at high speed / hit hard / v > 6.6fps and v <= 10fps	*Gamelan* makes a loud sound after being hit. (Volume 0.66 - 1)	The sound volume output is a high or loud sound. (Volume 0.66 - 1)	VALID

According to the test on sound as shown in Table 5, when the *gamelan* is hit at low speed (v <= 3.3fps) the sound output is (0 - 0.33 volume). When the *gamelan* is hit at medium speed (v > 3.3fps and v <= 6.6fps) the sound output is (0.33- 0.66 volume), and when the *gamelan* is hit at high speed (v > 6.6fps and v <= 10fps) the sound output is (0.66 - 1 volume) have met expectations in the test cases that have been made.

4.2. Collision Results in the *Gamelan Saron* VR Environment

The results shown in Table 6, obtained from the "EB01", "EB02", and "EB03" tests, are valid, which means that the collision in the environment in the VR *gamelan Saron* has been implemented correctly and the players cannot penetrate or walk into the wall. The results fit the expectations set in the test case.

Table 6. Collision Results in the *Gamelan Saron* VR Environment

ID	Testing Descriptions	Results	Expected Results	Conclusion
EB01	Players walk using the joystick on the controller.	Collision in the Environment works well, and players cannot walk through walls.	Players cannot penetrate walls.	VALID
EB02	Players walk in the real world and do not use a joystick.	Collision in the Environment works well, and players cannot walk through walls.	Players cannot penetrate walls.	VALID
EB03	Players walk using the joystick and walk in the real world.	Collision in the Environment works well, and players cannot walk through walls.	Players cannot penetrate walls.	VALID

The results shown in Table 6, obtained from the "EB01", "EB02," and "EB03" tests, are valid, which means that the collision in the environment in the VR *gamelan Saron* has been implemented accordingly and the players cannot penetrate or walk into the wall. The results fit the expectations set in the test case.

4.3. Main Menu Testing Results on the VR *Gamelan Saron*

Testing the “Main Menu” in the VR *gamelan Saron* application ensures that the “Main Menu” buttons can function properly and work well. The buttons tested are the "Start" button, the "Tutorial" button, and the "About" button. All of the buttons can be interacted with only using VR controller.

Table 7. Main Menu Testing Results on the VR *Gamelan Saron*

ID	Testing Descriptions	Results	Expected Results	Conclusion
MC01	Players select the “START” menu option.	When players select the "Start" menu, the player immediately enters *the Gamelan Saron* environment.	Players will be directed to an environment with a *Gamelan Saron*.	VALID
MC02	Players select the “TUTORIAL” menu option.	When players select the "Tutorial" menu, a tutorial menu appears in images and text.	The system will display the “Tutorial” menu.	VALID
MC03	Players select the “ABOUT” menu option.	When players select the "About" menu, a menu appears with some text.	The system will display the “About” menu.	VALID

The results in Table 7 show that when the "Start" button was pressed or selected by players, they would be directed directly into the environment and could immediately play *gamelan Saron*. When the "Tutorial" menu was pressed or chosen by players, they would be shown a picture with text on how to play *gamelan Saron*. When players pressed the "About" button or selected it, they would be shown text about the names of VR *gamelan Saron* developers. Based on the test cases, the results of testing the main menu fit the expectations.

4.4. *Wilahan* Testing Results on the VR *Gamelan Saron*

Testing of *Wilahan* on the VR *gamelan Saron* ensures that each *Wilahan* is issued a sound output like the original *gamelan Saron*. The *Wilahan* in *gamelan Saron* consisted of "ji," “ro,” “lu,” “pat,” “mo,” nem,” and “tu”. The order is based on the size of the *Wilahan* from the largest to the smallest.

Table 8. Wilahan Testing Results on the VR *Gamelan Saron*

ID	Testing Descriptions	Results	Expected Results	Conclusion
WD01	Players will hit the first "Wilahan" section.	Gamelan will issue a sound called *"ji."*	"Wilahan" will issue a sound called *"ji."*	VALID
WD02	Players will hit the second "Wilahan" section.	Gamelan will issue a sound called *"ro."*	"Wilahan" will issue a sound called *"ro."*	VALID
WD03	Players will hit the third "Wilahan" section.	Gamelan will issue a sound called *"lu."*	"Wilahan" will issue a sound called *"lu."*	VALID
WD04	Players will hit the fourth "Wilahan" section.	Gamelan will issue a sound called *"pat."*	"Wilahan" will issue a sound called *"pat."*	VALID
WD05	Players will hit the fifth "Wilahan" section.	Gamelan will issue a sound called *"mo."*	"Wilahan" will issue a sound called *"mo."*	VALID
WD06	Players will hit the sixth "Wilahan" section.	Gamelan will issue a sound called "nem."	"Wilahan" will issue a sound called *"nem."*	VALID
WD07	Players will hit the seventh "Wilahan" section.	Gamelan will issue a sound called "tu."	"Wilahan" will issue a sound called *"tu."*	VALID

In Table 8, the test results "WD1", "WD2", "WD3", "WD4", "WD5", "WD6", and "WD7" are valid, which means that when the *Wilahan* is hit, it produces a sound like the original *gamelan Saron*. The results showed that the output matched expectations and the test cases.

5. Conclusion and Recommendation

The VR *gamelan Saron* application can be developed more to enhance the virtual gamelan-playing experience by providing more features to achieve a more comfortable and accurate experience. Based on the discussion, the use of black box testing based on the equivalence partition technique on the VR *gamelan Saron* application shows that each function operates following predetermined expectations. Nevertheless, alternative testing techniques can be used in conjunction with the Equivalence Partitions method in this study to enhance the accuracy of each test case area within the VR *gamelan Saron* application.

References

[1] Ahmad Triaji, "Pembuatan Aplikasi Augmented Reality Sebagai Media Pengenalan Alat Musik Gamelan Jawa Berbasis Android," *J. Multi Media dan IT*, vol. 5, no. 2, Dec. 2021, doi: https://doi.org/10.46961/jommit.v5i2.446.

[2] Pengelola Web Kemdikbud, "Gamelan Jadi Warisan Budaya Dunia, Mendikbudristek Sampaikan Apresiasi Kepada Pegiat Budaya ," *kemdikbud.go.id*, Dec. 16, 2021.

[3] Redaktur, "Jadi Warisan Budaya Dunia, Gamelan Yang Terlupakan Pewarisnya," *inijabar.com*, Mar. 22, 2022.

[4] E. Irmania, A. Trisiana, and C. Salsabila, "Upaya mengatasi pengaruh negatif budaya asing terhadap generasi muda di Indonesia," *Din. Sos. Budaya*, vol. 23, no. 1, pp. 148–160, 2021, [Online]. Available: http://journals.usm.ac.id/index.php/jdsb.

[5] Y. K. Dwivedi *et al.*, "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *Int. J. Inf. Manage.*, vol. 66, Oct. 2022, doi: 10.1016/j.ijinfomgt.2022.102542.

[6] S. G. Fussell and D. Truong, "Using virtual reality for dynamic learning: an extended technology acceptance model," *Virtual Real.*, vol. 26, no. 1, pp. 249–267, Mar. 2022, doi: 10.1007/s10055-021-00554-x.

[7] A. Kurniawan, A. Maulana, V. R. Sukma, W. Keumala, and A. Saifudin, "Jurnal Teknologi Sistem Informasi dan Aplikasi Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store)," vol. 3, no. 1, pp. 2654–4229, 2020, [Online]. Available: http://openjournal.unpam.ac.id/index.php/JTSI50

[8] U. Nugraha and T. Sianturi, "Blackbox Testing On E-Commerce System Web-Based Evermos (Feature: Registration Experiment & Revamp)," 2021.
[9] D. Yulistyanti, T. Y. Akhirina, T. Afrizal, A. Paramita, and N. Farkhatin, "Testing Learning Media for English Learning Applications Using BlackBox Testing Based on Equivalence Partitions," *Scope J. English Lang. Teach.*, vol. 6, no. 2, p. 73, 2022, doi: 10.30998/scope.v6i2.12845.
[10] S. Thapliyal and R. Bahuguna, "A Newly Proposed Ambidextrous Software Testing Model Based on Conventional Black Box Testing Strategy Using the Applications of Gaussian Distribution," *Int. Res. J. Innov. Eng. Technol.*, vol. 5, no. 8, pp. 94–101, 2021, doi: 10.47001/IRJIET/2021.508016.
[11] M. Syah Anwar Kesuma Jaya, P. Gumilang, Y. Philipus Andersen, and dan Teti Desyani, "Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions," vol. 4, no. 4, pp. 2622–4615, 2019, [Online]. Available: http://openjournal.unpam.ac.id/index.php/informatika
[12] B. I. Fox and B. G. Felkey, "Virtual reality and pharmacy: Opportunities and challenges," *Hosp. Pharm.*, vol. 52, no. 2, pp. 160–161, 2017, doi: 10.1310/hpj5202-160.
[13] J. Martín-Gutiérrez, C. E. Mora, B. Añorbe-Díaz, and A. González-Marrero, "Virtual technologies trends in education," *Eurasia J. Math. Sci. Technol. Educ.*, vol. 13, no. 2, pp. 469–486, 2017, doi: 10.12973/eurasia.2017.00626a.
[14] G. YILDIRIM, M. ELBAN, and S. YILDIRIM, "Analysis of Use of Virtual Reality Technologies in History Education: A Case Study," *Asian J. Educ. Train.*, vol. 4, no. 2, pp. 62–69, 2018, doi: 10.20448/journal.522.2018.42.62.69.
[15] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software testing techniques: A literature review," *Proc. - 6th Int. Conf. Inf. Commun. Technol. Muslim World, ICT4M 2016*, no. November, pp. 177–182, 2017, doi: 10.1109/ICT4M.2016.40.
[16] M. Albarka Umar and C. Zhanfang, "A Comparative Study of Dynamic Software Testing Techniques," *Int. J. Adv. Netw. Appl.*, pp. 4575–4584, 2020.
[17] Meiselino Ansfridus, "Implementasi Black Box Testing pada Sistem Techno Expertise Academy (TEA) (Studi Kasus: Astra Credit Companies)," pp. 16-17, Nov. 2021.