Published *online* on the journal's web page : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

Comparison Analysis of Equivalence Class Partitioning and Boundary Value Analysis Techniques in Software Quality Testing of ReservasiPolnep Application

Zuhrie Alifiansyah[1], Syarifah Putri Agustini Alkadri[2], Rachmat Wahid Saleh Insani[3]

[1]*Informatics Engineering Study Program, Faculty of Engineering and Computer Science, Universitas Muhammadiyah Pontianak, Indonesia*

[1]*zuhriealif24@gmail.com,* [2]*agustini.putri@unmuhpnk.ac.id,* [3]*rachmat.wahid@unmuhpnk.ac.id*

ARTICLE INFORMATION

Article History:
Received: August 31, 2025
Last Revision: November 21, 2025
Published *Online*: November 30, 2025

KEYWORDS

Software Testing,
Black Box Testing,
Equivalence Class Partitioning,
Boundary Value Analysis,
ReservasiPolnep

CORRESPONDENCE

Phone: +62 878-4215-8371
E-mail: *zuhriealif24@gmail.com*

ABSTRACT

Software testing is a crucial phase before the official launch of an application to ensure its functionality and quality. This research compares two black box testing techniques Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) in identifying functional defects in the ReservasiPolnep application. Key application features were tested using both techniques, and results were measured using standard software testing metrics: test case coverage, success rates, test time, and cost per defect. The results showed that Equivalence Class Partitioning is more time and cost-efficient, requiring only 26 test cases and 15 minutes 27 seconds per test, with a cost of Rp30 per defect and an 84.6% success rate. In contrast, Boundary Value Analysis covers more test scenarios with 36 test cases, taking 27 minutes 5 seconds and costing Rp40 per defect, with a slightly higher success rate of 86.1%. The study concludes that each technique has advantages depending on the context, and highlights the need for input validation improvements in the application. This comparison provides empirical evidence for selecting the optimal testing strategy in academic web-based reservation systems, balancing resource efficiency (ECP) with comprehensive boundary defect detection (BVA).

1. INTRODUCTION

Politeknik Negeri Pontianak, as a higher education institution, continuously strives to improve service quality and the efficiency of resource management. One of the common challenges encountered is related to the borrowing of available facilities. The manual borrowing system previously used often caused difficulties such as limited access to facility availability information, a complicated borrowing process, and the potential for scheduling conflicts.

The ReservasiPolnep application is an information system developed by the Academic Support Unit for Information and Communication Technology (UPA-TIK) of Politeknik Negeri Pontianak with the primary goal of enhancing efficiency and convenience in the facility reservation process. This web-based application provides booking features and reservation status tracking, enabling users to monitor their requests more easily. However, before the application is officially launched, it is necessary to conduct comprehensive software quality testing to ensure that all system functions operate properly and as expected.

Software testing is one of the main tools in software quality assurance, designed to ensure product quality before it is delivered to users [1]. Over time, however, the objective of testing has evolved beyond merely ensuring that software is free of errors. It also serves to verify and validate overall product quality. Furthermore, software testing is intended to detect errors that have not yet been identified through various carefully designed test cases [2].

There are various methods available for software testing, one of which is black-box testing. This method plays a crucial role in ensuring that all system functions work correctly and in validating the functionality of the tested system [3]. Black-box testing is often used because it does not require access to source code or system architecture. Testing is conducted by interacting with the system through the user interface, without needing to understand its internal implementation [2].

Within black-box testing, there are several techniques such as Equivalence Class Partitioning (ECP) and

Boundary Value Analysis (BVA). Although both are categorized as black-box testing techniques, they emphasize different aspects. Equivalence Class Partitioning focuses on dividing input values into equivalent classes to reduce the number of tests without compromising coverage. In contrast, Boundary Value Analysis emphasizes testing the boundary values of each class, as these values are considered more prone to errors. Despite these differences, both techniques are equally important in ensuring software quality.

Equivalence Class Partitioning (ECP) is a testing technique based on the input data of each form or application menu. Each input is tested and grouped according to its function, whether valid or invalid [3],[4],[5],[6],[7]. This technique divides the input domain into equivalent partitions, making the testing process more efficient by reducing the number of test cases required without sacrificing coverage [2]. Boundary Value Analysis (BVA), on the other hand, is a black-box testing technique that focuses on testing the upper and lower boundary values of data inputs. The aim is to select test data near the domain boundaries, based on the assumption that implementation errors often occur at these points. This technique can detect failures caused by errors in handling boundary values and evaluate software quality by ensuring that boundary value testing does not result in errors stored in the database [8],[9],[10].

The main focus of this research is to compare the effectiveness of Equivalence Class Partitioning and Boundary Value Analysis as software testing techniques in identifying functional errors within the ReservasiPolnep application, considering the complexity of its features that involve user data management, scheduling, and facility availability. These two techniques were selected because of their relevance and ability to uncover various types of errors. This research employs Standard Software Testing Metrics as parameters to measure the total number of executed test cases (Test Case Executed), success rate (Test Case Passed), failure rate (Test Case Failed), test coverage, and cost of finding defects (Cost of Finding Defect) [3],[8],[11],[12],[13].

Previous studies have demonstrated that both techniques have their respective strengths and limitations. Most comparisons focus on generic or commercial systems without considering the complexity of institutional reservation features, such as user roles, time-bound submissions, and multi-step approvals. Therefore, this study aims to fill that gap by evaluating the effectiveness and efficiency of Equivalence Class Partitioning and Boundary Value Analysis techniques using the ReservasiPolnep application as a case study. The research intends to provide empirical evidence on which testing technique performs better in identifying functional defects in an academic reservation system, using software testing metrics as evaluation parameters.

2. Related Work

Several previous studies have explored the application of black box testing techniques such as Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) to evaluate the quality of web-based systems. Aryandana, Permanasari, and Adji [12] conducted a comparative study of Equivalence Class Partitioning and Boundary Value Analysis on a medicine management module. Their findings showed that Equivalence Class Partitioning had superior performance in detecting bugs, with a success rate of 83.33% compared to BVA's 50%, and a lower defect leakage rate. The study also revealed that Equivalence Class Partitioning enabled more efficient test case execution. Ahrizal et al. [10] applied Boundary Value Analysis in testing a PlayStation rental system and found the success rate to be around 70%. Their research highlighted the importance of input completeness and the need for validation improvements in the system. In another study, Putri et al.[8] compared several black box techniques on a reservation website and concluded that Equivalence Class Partitioning was the most effective and efficient technique, outperforming both Boundary Value Analysis and Decision Table methods in identifying bugs with minimal test cases.

Furthermore, Nurfathullah and Purnamasari [14] implemented Equivalence Class Partitioning in a sales order system, which resulted in zero errors from 17 test cases, indicating high system reliability. This finding reinforces the effectiveness of Equivalence Class Partitioning in structured input environments. Meanwhile, Susilo and Suharto [15] used Selenium IDE for automated testing of an employee attendance system and achieved a 100% success rate, demonstrating the tool's capability in web-based testing scenarios.

Although previous studies have explored Equivalence Class Partitioning and Boundary Value Analysis individually or in generic commercial systems , there is limited research that directly compares the effectiveness and efficiency of both techniques specifically within the context of academic reservation systems like ReservasiPolnep. Features such as user roles, time-bound submissions, and multi-step approvals inherent in institutional reservation platforms introduce a complexity not addressed in generic comparisons. Therefore, this research novel contribution is providing empirical evidence on the comparative performance (effectiveness and efficiency) of Equivalence Class Partitioning and Boundary Value Analysis in identifying functional defects within a specific, complex academic reservation system (ReservasiPolnep), utilizing comprehensive software testing metrics as evaluation parameters. This comparison offers practical recommendations for future testing strategies in similar educational contexts.

3. METHODOLOGY

The methodology is structured into several stages, as illustrated in Figure 1. This research employs a quantitative experimental approach to compare the effectiveness of the Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) techniques in detecting functional errors in the ReservasiPolnep application.

3.1 Literature Study

The first stage involves a comprehensive literature review to gain insights into the principles, concepts, and best practices of software testing, particularly black box testing techniques such as Equivalence Class Partitioning and Boundary Value Analysis. Sources include textbooks,

scientific journals, conference papers, and prior relevant studies.

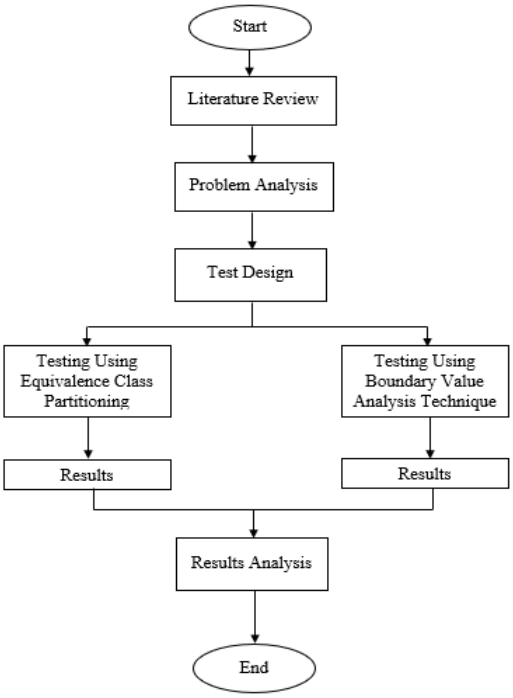

FIGURE 1. FLOW CHART DIAGRAM

3.2 Problem Analysis

After establishing a theoretical foundation, the next step is to analyze the application's requirements and features. This includes identifying key functions of the ReservasiPolnep application to be tested, such as reservation form validation, input processing, and reservation tracking.

ReservasiPolnep is an information system application developed by the Academic Support Unit for Information and Communication Technology (UPA-TIK) of Politeknik Negeri Pontianak to facilitate the reservation process for various services at the institution. This application allows both internal and external users of the campus to book a variety of available services, such as reserving buildings for specific events and booking transportation services like buses. With its intuitive interface, ReservasiPolnep provides features for making reservations as well as tracking the status of submitted requests, enabling users to easily monitor their applications. The system is expected to enhance efficiency and convenience in the process of reserving facilities provided by Politeknik Negeri Pontianak.

The workflow of the reservation submission process in the ReservasiPolnep application is illustrated in Figure 2.

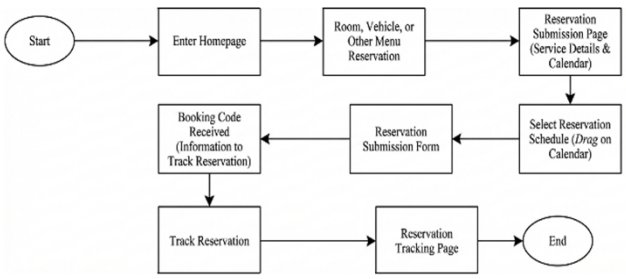

FIGURE 2. RESERVATION SUBMISSION FLOW

3.3 Test Case Design

This stage includes designing test scenarios and test cases based on Equivalence Class Partitioning (ECP) and Boundary Value Analysis techniques. Equivalence Class Partitioning groups input values into valid and invalid equivalence classes to reduce the number of test cases without sacrificing coverage [16], [4], [5], [6], [7]. At this stage, the researcher applies testing using the Equivalence Class Partitioning technique on the functions, features, and predefined scenarios. The steps carried out in this testing are as follows:

a. Identify and determine the features to be tested.
b. Identify the input variables.
c. Define the equivalence classes.
d. Design the test cases.
e. Execute the test cases.
f. Analyze the results.

Table 1 present examples of test results using Equivalence Class Partitioning (ECP).

TABLE 1. EXAMPLE OF ECP CLASSIFICATION

Input	Equivalence Class	Valid/Invalid
Username	≤ 30 character	Valid
	>30 character	Invalid
	Empty Field	Invalid
Handphone	≤ 12 character	Valid
	>12 character	Invalid
	Empty Field	Invalid

Meanwhile, Boundary Value Analysis focuses on testing input values at the lower and upper boundaries of the input domain and their surrounding values [8], [9], [10], [17]. At this stage, the researcher applies testing using the Boundary Value Analysis technique on the functions, features, and predefined scenarios. The steps carried out in this testing are as follows:

a. Identify and determine the features to be tested.
b. Identify the input variables and their value ranges.
c. Define the boundary values.
d. Design the test cases.
e. Execute the test cases.

Table 2 present examples of test results using Boundary Value Analysis (BVA).

TABLE 2. EXAMPLE OF BVA CLASSIFICATION

Input	Boundary Value	Valid/Invalid
Username	1 character	Valid
	30 character	Valid
	0 character	Invalid
	31 character	Invalid
	Empty Field	Invalid
Handphone	1 character	Valid
	12 character	Valid
	0 character	Invalid
	13 character	Invalid
	Empty Field	Invalid

3.4 Application Testing

Before conducting testing, it's important to develop a test scenario to ensure no features are overlooked. This scenario includes a list of features to be tested and the

expected results. Any defects or bugs encountered during testing will be documented immediately.

The testing process is carried out using Selenium IDE as an automation tool [15], [18], [19]. Each scenario is executed automatically, and data such as test duration, success rate, and number of defects detected is recorded systematically. The number of test cases varies for each technique due to the nature of Equivalence Class Partitioning and Boundary Value Analysis [20], [21]. To illustrate, Tables 3 present examples of test results using Equivalence Class Partitioning and Boundary Value Analysis techniques.

TABLE 3. EXAMPLE OF TEST RESULTS

ID	Description	Expected	Actual	Conclusion
EP01	Test scenario EP01	Successfully	Successfully	Match
EP02	Test scenario EP02	Successfully	Failed	Not Match
BV01	Test scenario BV02	Successfully	Successfully	Match
BV02	Test scenario BV02	Successfully	Failed	Not Match

3.5 Evaluation and Data Analysis

To assess testing results, Software Testing Metrics are used [11], [12], [13], Software Testing Metrics are essential instruments for evaluating the quality and reliability of software. These metrics can be categorized into three main types: Product Metrics, which focus on the quality of the software product; Process Metrics, which measure the effectiveness and efficiency of the software development process; and Project Metrics, which are used to monitor and control software projects. Each metric provides an objective benchmark for assessing various aspects of the testing process, such as test coverage, defect density, execution time, effort, and cost [13].

Testing metrics allow developers to systematically monitor project progress, identify areas that require improvement, and ensure that the delivered software meets the established quality standards [8], [11], [12]. In this study, the metrics used are in Table 4.

TABLE 4. SOFTWARE TESTING METRICS

Metric	Formula
Test Case Executed %	$\left(\frac{\text{Test Case Executed}}{\text{Test Case Written}}\right) \times 100\%$
Test Case Passed %	$\left(\frac{\text{Test Case Passed}}{\text{Total Test Case}}\right) \times 100\%$
Test Case Failed %	$\left(\frac{\text{Test Case Failed}}{\text{Total Test Case}}\right) \times 100\%$
Cost of Finding Defect	$\left(\frac{\text{Effort Spent on Testing}}{\text{Defect Found on Testing}}\right)$

3.6 Conclusion Drawing

The final stage involves interpreting the testing results to draw conclusions about the comparative effectiveness and efficiency of the two techniques. The results are evaluated based on the number of defects found, test execution time, number of test cases, and overall testing cost.

4. RESULT AND DISCUSSION

The testing of the ReservasiPolnep application using Equivalence Class Partitioning and Boundary Value Analysis revealed several important findings regarding the effectiveness of both techniques in detecting errors. The analysis focused on the types of errors identified, the causes of test case failures, and their implications for application quality.

The Equivalence Class Partitioning technique proved effective in detecting common validation errors, such as empty input, incorrect data formats, and character lengths exceeding limits. However, its main weakness lies in the inability to identify errors at input boundaries, which often results in missed cases at extreme values. In contrast, Boundary Value Analysis demonstrated strong effectiveness in detecting errors at minimum and maximum input boundaries, particularly for numeric data. Nonetheless, Boundary Value Analysis showed limitations when applied to non-numeric inputs or data without explicit boundaries, such as dropdown selections or semantic input. Several test case failures occurred due to inconsistent validation. In some cases, the system stored data even when the input was invalid, indicating that validation was only partially implemented on the client side and not comprehensively on the server side. This highlights weaknesses in the application's validation logic.

These findings carry significant implications for application quality. From a reliability perspective, weak validation undermines system dependability. In terms of security, unvalidated inputs expose the system to potential exploits, including injection attacks. From the user experience standpoint, discrepancies between expected and actual outcomes lead to confusion and reduced user satisfaction.This section presents the implementation of the planned methodology, including the testing process using Equivalence Class Partitioning and Boundary Value Analysis, followed by an analysis and comparison of the test results using several software testing metrics.

4.1 Test Execution Using Equivalence Class Partitioning (ECP)

Testing was conducted on the reservation submission and tracking system to identify system weaknesses and ensure that all tested features were error-free or bug-free. Test results are presented in a test case table, which is used to assess the system's success and compliance with the test plan. In this test, 26 test cases were designed based on input equivalence classes identified from the application's reservation form. These test cases covered both valid and invalid input data. The automated test execution using Selenium IDE was completed in 15 minutes and 27 seconds. The results are summarized in Table 5.

TABLE 5. SUMMARY OF ECP TESTING RESULTS

Metric	Result
Total Test Case Executed	26
Test Case Passed	22
Test Case Failed	4
Test Execution Time	15m 27s

Several test failures occurred due to inconsistencies in input validation, especially in the fields for date, NIP/NIM/NIK, and reservation category. This indicates that some validation logic needs improvement to handle edge cases more effectively.

4.2 Test Execution Using Boundary Value Analysis

Testing was conducted on the reservation submission and tracking features to identify system weaknesses and

ensure all tested functions ran without errors or bugs. The test results are presented in a test case table, which is used to evaluate the system's success rate and compliance with the test plan. For Boundary Value Analysis (BVA), 36 test cases were generated by identifying boundary conditions from the same input fields. This technique focuses on testing the minimum and maximum values around the allowed input ranges. The total execution time recorded was 27 minutes and 5 seconds. The results are summarized in Table 6.

TABLE 6. SUMMARY OF BVA TESTING RESULTS

Metric	Result
Total Test Case Executed	36
Test Case Passed	31
Test Case Failed	5
Test Execution Time	27m 5s

Although Boundary Value Analysis (BVA) produced a slightly higher success rate (86.1%), it required more time and effort due to the higher number of test cases. The technique proved to be beneficial in revealing defects in fields where input limits were not handled correctly.

4.3 Comparative Analysis

The main focus of this research is to compare Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) as software quality testing techniques. Therefore, this study employs Standard Software Testing Metrics as parameters to measure various aspects of the testing process, ranging from test coverage to the severity level of the detected defects. Based on the testing that has been conducted, the next step is to calculate the percentage of the Standard Software Testing Metrics, as follows in Table 7.

TABLE 7. COMPARATIVE ANALYSIS

Metric	ECP	BVA
Test Case Executed %	26	36
Test Case Passed %	84.6%	86.1%
Test Case Failed %	15.3%	13.8%
Test Execution Time	15m 27s	27m 05s
Cost of Finding Defect	Rp30	Rp40

Comparison of Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) techniques according to the results of the Standard Software Testing Metrics percentage calculations and other test result data is visualized using graphs, as follows.

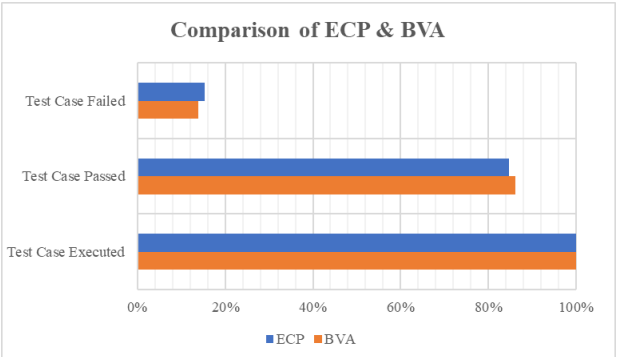

FIGURE 3. PERFORMANCE OF COMPARISON BETWEEN ECP AND BVA

Figure 3 shows that both techniques have the same high test coverage with a test case execution rate of 100%. However, the passing test case success rate is slightly higher in Boundary Value Analysis (86.1%) than in Equivalence Class Partitioning (84.6%), while the percentage of failed test cases is lower in Boundary Value Analysis (13.8%) than in Equivalence Class Partitioning (15.3%).

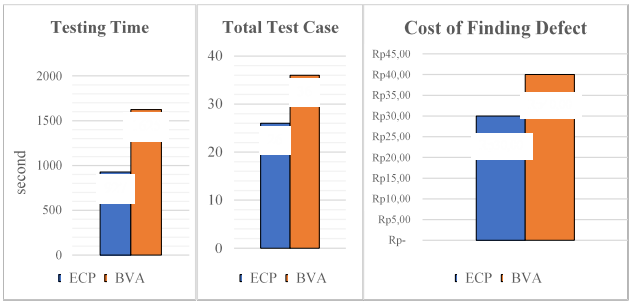

FIGURE 4. COMPARISON TESTING TIME, TOTAL TEST CASE AND COFD

Figure 4 shows that in terms of efficiency, the time required for testing using Equivalence Class Partitioning is shorter, namely 927 seconds (15 minutes 27 seconds) compared to Boundary Value Analysis which takes 1625 seconds (27 minutes 5 seconds) with a wider Boundary Value Analysis testing scope of 36 test cases while Equivalence Class Partitioning has 26 test cases. Because of this, in terms of Cost of Finding Defect (CoFD), testing with Boundary Value Analysis requires a more expensive cost of around Rp. 40 to find defects, compared to Equivalence Class Partitioning which requires a cheaper cost of around Rp. 30 which shows that Equivalence Class Partitioning is slightly more efficient in finding defects than Boundary Value Analysis.

While Equivalence Class Partitioning (ECP) demonstrated superior efficiency in terms of execution time (15 min 27 sec) and CoFD (Rp30) , it achieved this with fewer test cases (26 total) compared to BVA's 36. In practical terms, this difference signifies ECP's ability to validate the majority of functional paths (valid and invalid equivalence classes) quickly and cost-effectively, making it ideal for rapid smoke tests or iterative development cycles where resource limits are tight and core functionality validation is the priority. However, the lower number of test cases means Equivalence Class Partitioning inherently has lower coverage of critical boundary conditions. Conversely, BVA's higher case count (36) and slightly longer time (27 min 5 sec) are necessary to explicitly cover values at, just above, and just below the input limits, which are often the most defect-prone areas. Therefore, the trade-off in choosing Equivalence Class Partitioning is sacrificing the comprehensive, detail-oriented defect detection at the boundaries for significant gains in testing speed and cost.

The outcome of this research where Equivalence Class Partitioning proved more efficient (lower CoFD and time) but Boundary Value Analysis (BVA) showed slightly higher effectiveness (success rate 86.1% vs. 84.6%) partially aligns with and extends previous research. Aryandana, Permanasari, and Adji [12], found Equivalence Class Partitioning (ECP) to be superior in both efficiency and effectiveness (bug detection rate), which supports the efficiency findings of our study. For instance, Putri et al. [8], reported ECP's success rate at 51.8%, higher than BVA's 33.3%, which reinforces ECP's strength in identifying bugs with minimal cases. However, our finding that Boundary Value Analysis achieved a slightly higher

success rate (86.1%) for boundary coverage in the ReservasiPolnep system provides a more nuanced view than previous works. This indicates that in complex systems with critical input boundaries, BVA's focused coverage offers a marginal but valuable increase in defect detection, whereas in more straightforward or non-boundary-critical systems (like the medicine management or Lars website studied previously), ECP's broad-stroke efficiency is the dominant factor. The study by Ahrizal et al. [10] which highlighted BVA's limitations with non-boundary-related empty fields, also aligns with the observed need for comprehensive Equivalence Class Partitioning coverage alongside Boundary Value Analysis.

4.4 Findings and Implications

The main findings indicate that both techniques are capable of identifying defects effectively, but they exhibit different strengths and limitations. Equivalence Class Partitioning (ECP) is proven to be more suitable when time and cost are constrained, as it allows testers to design a smaller set of representative test cases while still maintaining adequate coverage of functional requirements. This efficiency makes Equivalence Class Partitioning particularly valuable in projects with limited resources, strict deadlines, or iterative development processes where rapid validation is required. In such contexts, Equivalence Class Partitioning enables testers to detect defects without overburdening the testing process with excessive complexity.

On the other hand, Boundary Value Analysis (BVA) demonstrates its importance in scenarios where precision at data limits is critical. Since many defects often occur at the boundaries of input domains, Boundary Value Analysis ensures that these critical points are thoroughly validated. While it may require more time and resources compared to Equivalence Class Partitioning, Boundary Value Analysis reduces the likelihood of boundary-related defects that could lead to significant failures in real-world use. This makes Boundary Value Analysis particularly relevant for safety-critical applications, financial systems, or environments where strict data integrity must be preserved.

The comparative analysis suggests that Equivalence Class Partitioning and Boundary Value Analysis should not be seen as competing techniques, but rather as complementary approaches. By integrating both methods, testers can balance efficiency with thoroughness, achieving optimal defect detection while managing resource constraints. For example, Equivalence Class Partitioning can be employed to cover the majority of functional validation efficiently, while Boundary Value Analysis can be selectively applied to critical boundary conditions where higher risk of defects is anticipated. Such a hybrid strategy would maximize test coverage while maintaining reasonable execution costs and time.

In the context of the ReservasiPolnep application, these findings have significant implications. As a web-based reservation system used in academic settings, it must operate reliably under varying input conditions from diverse users. Equivalence Class Partitioning ensures that the core functionalities, such as booking and scheduling, are validated efficiently, enabling rapid deployment and updates. Meanwhile, Boundary Value Analysis ensures that potential issues at input limits—such as maximum reservation slots, date ranges, or numeric values—are thoroughly examined, minimizing the possibility of system failures or user dissatisfaction.

Furthermore, the implications of this study extend to broader software quality assurance practices. The findings highlight the importance of selecting appropriate testing techniques based on project needs rather than applying a single method universally. In practice, adopting a context-driven approach to testing where Equivalence Class Partitioning is prioritized for efficiency and Boundary Value Analysis for accuracy can significantly improve the overall effectiveness of software validation. This perspective aligns with modern software engineering practices that emphasize adaptability, risk management, and cost-effectiveness in testing strategies.

Ultimately, the study reinforces that the effectiveness of software testing does not lie solely in the choice of one technique over another, but in the strategic combination of methods tailored to the characteristics of the system under test. This balanced approach may provide not only optimal coverage and efficiency but also ensure long-term system reliability and user trust, especially in complex web-based systems such as ReservasiPolnep.

5. CONCLUSIONS

The results of this research demonstrate that both Equivalence Class Partitioning (ECP) and Boundary Value Analysis (BVA) techniques are effective in identifying functional defects within the ReservasiPolnep application, although each has distinct advantages. Equivalence Class Partitioning proved to be more efficient in terms of execution time and cost, making it suitable for scenarios where rapid validation is required. On the other hand, Boundary Value Analysis provided slightly higher accuracy in detecting errors at input boundaries, despite requiring more time and effort due to the larger number of test cases. These findings indicate that Equivalence Class Partitioning is preferable when efficiency is prioritized, while Boundary Value Analysis is more appropriate when thorough testing of boundary conditions is critical.

For future work, combining both techniques may yield optimal results, ensuring efficiency while maintaining comprehensive coverage for complex academic reservation systems. However, this study also highlights several research gaps that may be addressed in subsequent investigations. First, the study was limited to functional defect detection and did not evaluate non-functional aspects such as system performance, security, and usability, which are equally important for real-world applications. Second, the comparison was conducted on a single case study (ReservasiPolnep), which may limit the generalizability of the findings; future research could replicate the study across different domains or larger-scale systems. Third, the integration of automated testing tools with Equivalence Class Partitioning and Boundary Value Analysis has not yet been explored in depth, and such integration could further improve testing efficiency and scalability.

Therefore, future studies are encouraged to extend this research by addressing non-functional testing dimensions, applying Equivalence Class Partitioning and Boundary Value Analysis in various application contexts, and exploring

automation-assisted hybrid approaches. These directions would contribute to a more comprehensive understanding of software testing effectiveness and provide practical insights for improving software quality assurance in diverse real-world scenarios.

REFERENCES

[1] D. Galin Galin, "Software Quality Assurance From theory to implementation Software Quality Assurance From theory to implementation Cyan Magenta Yellow Black." 2020. [Online]. Available: www.pearsoned.co.uk.

[2] N. Anwar and S. Kar, "Review Paper on Various Software Testing Techniques & Strategies," 2020.

[3] P. A. D. A. Santi, R. Afwani, Moh. A. Albar, S. E. Anjarwani, and A. Z. Mardiansyah, "Black Box Testing with Equivalence Partitioning and Boundary Value Analysis Methods," in *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022*, Atlantis Press International BV, 2022, pp. 207–219. doi: 10.2991/978-94-6463-084-8_19.

[4] R. Parlika, T. Ardhian Nisaa', S. M. Ningrum, and B. A. Haque, "Literature Study Of The Lack And Excess Of Testing The Black Box," *TEKNOMATIKA*, vol. 10, no. 02, pp. 1–5, 2020.

[5] E. Novalia and A. Voutama, "Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah," 2022.

[6] T. Desyani, S. Mulyati, E. Kurnianto, K. Kamaludin, N. Afifah, and S. N. I. Fauziah, "Pengujian Black Box menggunakan teknik Equivalence Partitions pada Aplikasi Sistem Pemilihan Karyawan Terbaik," *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 2, p. 110, May 2022, doi: 10.32493/jtsi.v5i2.17578.

[7] M. Zidan, S. Nur'aini, N. C. H. Wibowo, and M. A. Ulinuha, "Black Box Testing pada Aplikasi Single Sign On (SSO) di Diskominfostandi Menggunakan Teknik Equivalence Partitions," *Walisongo Journal of Information Technology*, vol. 4, no. 2, pp. 127–137, Nov. 2022, doi: 10.21580/wjit.2022.4.2.12135.

[8] S. J. Putri, D. Galih, P. Putri, W. Hayuhardhika, and N. Putra, "Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars)," *Journal of Internet and Software Engineering*, vol. 5, no. 1, 2024.

[9] F. D. Hartono and Y. Sugiarti, "Perbandingan Metode Equivalence Partitions dan Boundary Value Analisis Pada Pengujian Black Box Literatur Review," *Majalah Ilmiah METHODA*, vol. 12, no. 2, pp. 153–159, Aug. 2022, doi: 10.46880/methoda.Vol12No2.pp153-159.

[10] D. Ahrizal, M. Khaerul Miftah, R. Kurniawan, and T. Zaelani, "Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing," vol. 5, no. 1, 2020.

[11] P. Mathew Jacob and M. Prasanna Associate Professor, *A Comparative analysis on Black Box Testing Strategies, 2020*.

[12] I. G. S. Aryandana, A. E. Permanasari, and T. B. Adji, "Comparing method equivalence class partitioning and boundary value analysis with study case add medicine module," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, Jan. 2020. doi: 10.1088/1757-899X/732/1/012072.

[13] A. Petrov, "African Journal of Artificial Intelligence and Sustainable Development, Software Metrics and Measurement-Analysis and Applications: Studying software metrics and measurement techniques for quantifying software quality, productivity, and maintainability," 2024.

[14] M. Nurfathullah, "Pengujian Blackbox Pada Sistem Pemesanan Untuk Sales Order di PT Bukit Muria Jaya Berbasis Equivalence Partitions," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4174.

[15] A. Susilo, M. Iqbal, and F. Suharto, "Pengujian Otomatis Sistem Aplikasi Berbasis Web Dengan Menggunakan Selenium Ide Studi Kasus : Pengujian Sistem Aplikasi Absensi PT. Wilmar," *Jurnal Pelita Teknologi*, vol. 17, no. 6, pp. xx–xx, 2022.

[16] N. Anwar and S. Kar, "Review Paper on Various Software Testing Techniques & Strategies," 2019.

[17] M. A. Umar and C. Zhanfang, "A Comparative Study Of Dynamic Software Testing Techniques," *International Journal of Advanced Networking and Applications*, vol. 12, no. 03, pp. 4575–4584, 2020, doi: 10.35444/ijana.2020.12301.

[18] A. Alif Fahrizal, P. Arda Imawan, M. Abi Syihab, A. Majid, and A. Saifudin, "Pengujian Aplikasi Web Absensi Siswa Menggunakan Selenium IDE," *BIIKMA : Buletin Ilmiah Ilmu Komputer dan Multimedia*, 2023, [Online]. Available: https://jurnalmahasiswa.com/index.php/biikma

[19] F. Halawa and A. Saifudin, "Pengujian Fungsionalitas Aplikasi Kasir Berbasis Web dengan Metode Blackbox," *OKTAL : Jurnal Ilmu Komputer dan Science*, vol. 2, no. 6, 2023.

[20] A. Mori, "Anomaly Analyses to Guide Software Testing Activity," in *Proceedings - 2020 IEEE 13th International Conference on Software Testing, Verification and Validation, ICST 2020*, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 427–429. doi: 10.1109/ICST46399.2020.00055.

[21] G. Fraser and J. M. Rojas, "Software testing," in *Handbook of Software Engineering*, Springer International Publishing, 2019, pp. 123–192. doi: 10.1007/978-3-030-00262-6_4.

AUTHORS

Zuhrie Alifiansyah
Department of Informatics Engineering Study Program at Muhammadiyah University of Pontianak. He completed his Bachelor's degree in Informatics Engineering at Muhammadiyah University of Pontianak in 2025. He interested in information technology, software testing and human computer interaction.

Syarifah Putri Agustini Alkadri

She is a lecturer in the Department of Engineering and Informatics, Muhammadiyah University of Pontianak. She completed her Bachelor's degree at Tanjungpura University in Pontianak in 2011 and her Master's degree at the Nusa Mandiri College of Informatics and Computer Management in 2014.

Rachmat Wahid Saleh Insani

He is a lecturer in the Department of Engineering and Informatics, Muhammadiyah University of Pontianak. He completed his Bachelor's degree at STMIK Amikom in 2012 and his Master's degree at Gajah Mada University in 2015.