JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 8, No. 2, December 2025, pp. 776~787
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: https://doi.org/10.36378/jtos.v8i2.5028

Implementation of Equivalence Partitioning and Boundary Value Analysis in Black Box Testing: A Case Study on the Admin Website of PAUD KB Al Husna

Adelia Tiara Putri[1], Aisya Arline Husnaya[2], Dzaky Fachri Hadafi[3], Hasan Ismail Abdulmalik[4], Muhammad Nasir[5], Sofiyanti Indriasari[6]
[123456]Software Engineering Technology, College of Vocational Studies, IPB University, Bogor, Indonesia

Article Info

Article history:

Received 10 30, 2025
Revised 11 20, 2025
Accepted 12 02, 2025

Keywords:

Black Box Testing
Boundary Value Analysis
Equivalence Partitioning
Software Testing
System Validation

ABSTRACT

In order to guarantee system functioning and quality, software testing is an essential stage in the Software Development Life Cycle (SDLC). This study implements black box testing using Equivalency Partitioning (EP) and Boundary Value Analysis (BVA) on the Admin Website PAUD KB Al Husna to identify functional defects, focusing on input validation and error handling. In this study, 82 test scenarios were created for five functional modules (Class Management, Teacher Management, FAQ, Schedule, Gallery) and three non-functional factors (Security, Compatibility, and Responsiveness). Test execution was conducted manually based on BVA test cases (Min-1, Min, Max, Max+1) and EP cases (valid and invalid classes). Results showed a 32.9% failure rate (27 of 82 scenarios). Critical findings include: (1) Systemic failure in upper-boundary validation (BVA Max+1) with 256-character inputs; (2) EP-Invalid validation failures revealing Cross-Site Scripting (XSS) vulnerabilities from HTML input; and (3) Absence of brute-force protection on login pages. The study concludes that BVA and EP techniques effectively identify critical data validation and security defects, providing essential improvement recommendations for developers.

Corresponding Author:

Muhammad Nasir
Software Engineering Technology
College of Vocational Studies
IPB University
Bogor, Indonesia
Email: m_nasir@apps.ipb.ac.id

1. Introduction

Web-based information systems have gained significant popularity in the digital era, including within early childhood education institutions such as PAUD KB Al Husna. Testing is a crucial stage in the Software Development Life Cycle (SDLC) to guarantee software quality [1], [2]. Among various methodologies, black box testing is an approach focused on evaluating functionality without examining internal code structure [3], [4]. Boundary Value Analysis (BVA) and Equivalency Partitioning (EP) are common methods used in black box testing. EP divides the input domain into valid and invalid equivalence classes, enabling the selection of representative test cases from each category [5], [6]. BVA examines values at minimum boundaries, maximum boundaries, and beyond input domain limits [7], [8]. The combination of EP and BVA is expected to comprehensively capture input anomalies and invalid data classes.

Previous studies have demonstrated the effectiveness of these techniques in diverse contexts. For instance, Santi et al. applied EP and BVA to test the Academic Information System (SIA) at University of Mataram and identified 80 defects from 322 scenarios [9]. Similarly, Sholeh et al. implemented the same techniques on the ukmbantul.com e-commerce platform, concluding that the system effectively enforced input boundary constraints [10]. Kartono et al. conducted BVA testing on the *Osha Snack* e-commerce site and reported a 96% success rate in handling valid input cases [11]. Conversely, Hardika et al. performed EP-based testing on a fisheries website (*Garuda Farm*) and recorded a validity rate of only 56.25%, indicating issues in date and image upload validation [12]. Another study by Amrulloh et al. on *ternaku.id* revealed that, among 11 tested functions, three failed and eight succeeded [13]. Collectively, these studies affirm that the combined application of EP and BVA can effectively detect diverse functional and validation-related issues across various system types.

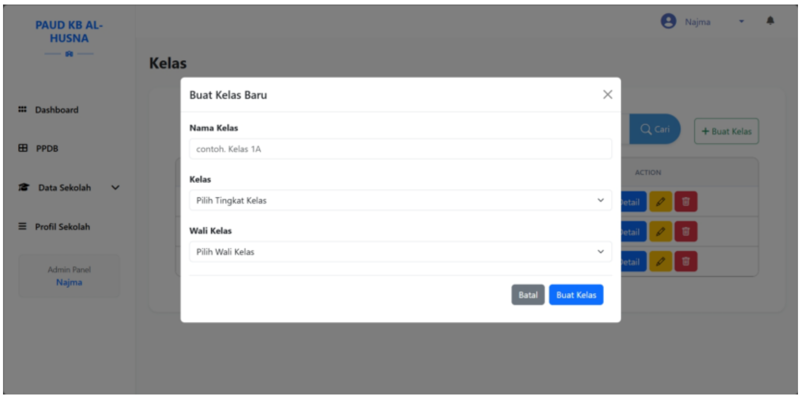
However, most prior research has focused on large-scale or enterprise-level information systems such as academic portals or e-commerce applications [14]. In contrast, validation and security testing for small-scale web-based content management systems (CMS), particularly those used in educational institutions, remain underexplored [15], [16]. These smaller systems often rely heavily on client-side validation and lack adequate server-side mechanisms, leading to potential vulnerabilities such as input overflow, malformed data entries, and security threats like *Cross-Site Scripting* (XSS) and *brute-force attacks* [17], [18]. Given that such systems manage sensitive student and staff information, validation failures can severely compromise data integrity and institutional credibility [19].

This study aims to apply black box testing techniques, specifically EP and BVA, to evaluate the functional reliability and input validation robustness of the Admin Website PAUD KB Al Husna. Unlike prior works that emphasize performance or scalability testing, this study emphasizes input validation, security robustness, and defect discovery within small-scale administrative systems. The primary contributions of this research include: (1) identifying critical defects in input handling, (2) analyzing root causes such as insufficient server-side validation, and (3) uncovering real-world vulnerabilities related to XSS and brute-force risks. These findings are expected to deliver practical recommendations to improve data validation practices and strengthen system security for similar web-based educational systems.

2. Research Method

2.1 Research Object

The research object in this study is the web-based application “Admin PAUD KB Al Husna.” This platform functions as an internal Content Management System (CMS) that enables school administrators to manage various types of content and operational data presented on the institution’s main website. The testing process focused on five primary functional modules that support daily administrative workflows: Class Management (“Add Class”), Teacher Management (“Add Teacher Data”), FAQ Management, Schedule Management (“Add Schedule”), and Gallery Management (“Add Gallery”). These modules were examined to ensure that each feature operated correctly, handled input efficiently, and produced outputs as expected. In addition to functional testing, three essential non-functional aspects were also evaluated: system security through login authentication, browser compatibility to ensure stable performance across multiple platforms, and responsive design to verify that the interface adapts smoothly to different screen sizes. Collectively, these tests provide a comprehensive assessment of the system’s reliability and usability.



Buat Kelas Baru

Nama Kelas

contoh. Kelas 1A

Kelas

Pilih Tingkat Kelas

Wali Kelas

Pilih Wali Kelas

Figure 1. Class Management Menu

The Class Management menu is used to create and organize class data for the school. Administrators can add a new class through the *Create New Class* modal, which requires entering the class name, selecting the class level, and choosing the homeroom teacher responsible for the class. Existing classes can be edited when changes occur, such as updating the class name or assigning a new homeroom teacher. This feature ensures that the structure of each class is well-documented, supporting smoother management of students. teaching assignments, and academic activities.

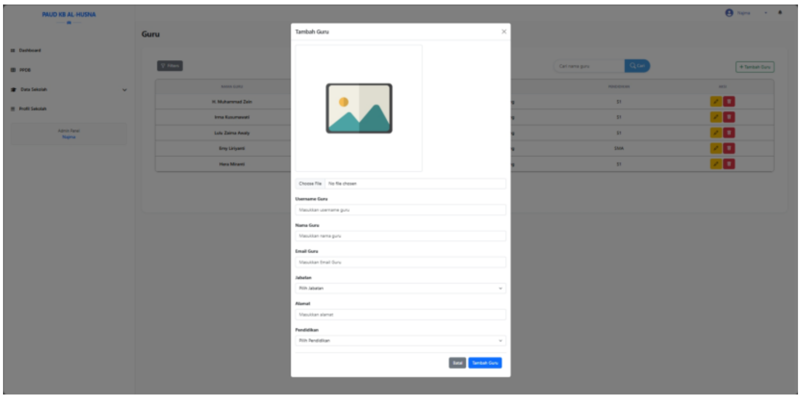


Figure 2. Teacher Management Menu

The Teacher Management menu serves as the central hub for managing all teacher-related data within the school system. The administrator can add new teachers by entering essential information such as profile photo, username, full name, email address, job position, home address, and educational background. The *Add Teacher* button opens a structured form modal that simplifies data entry. Administrators can also edit existing teacher information to ensure data accuracy or delete teacher records that are no longer active. A search bar is provided to help quickly locate specific teachers, making the process of managing staff data more efficient, organized, and user-friendly.

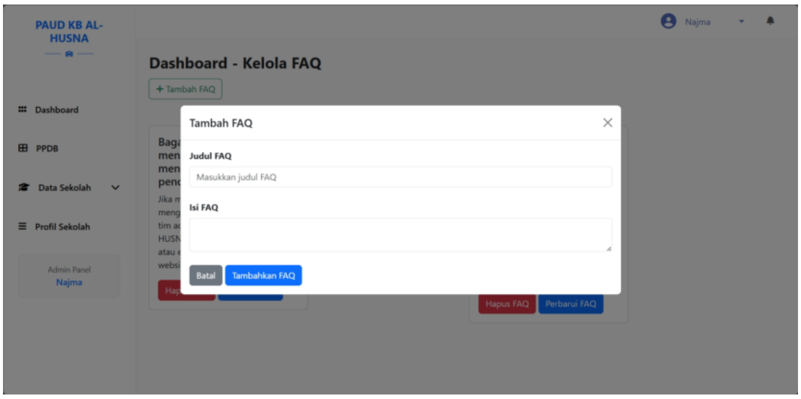


Figure 3. FAQ Management Menu

The FAQ Management menu enables the administrator to manage frequently asked questions that are displayed on the school's website. Using the *Add FAQ* modal, the admin can create a new FAQ entry by providing a clear question title and a detailed answer. This feature is especially useful for addressing common inquiries regarding admissions, school policies, or general information, reducing the need for direct communication. Administrators can also update outdated FAQs or remove those that are no longer relevant.

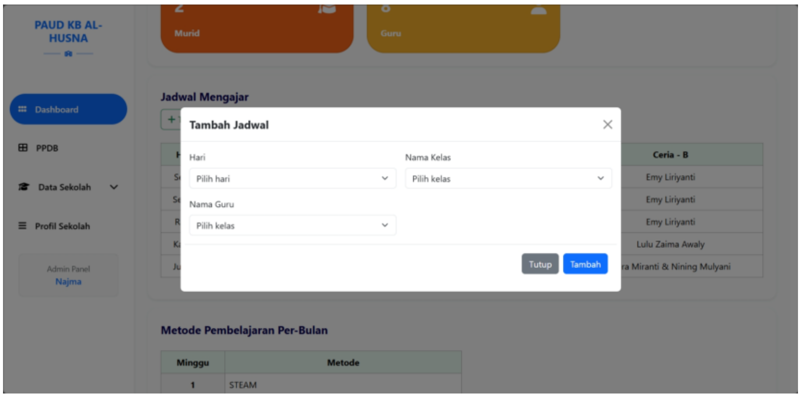


Figure 4. Schedule Management Menu

The Schedule Management menu enables administrators to set and organize teaching schedules for each class and teacher. Using the Add Schedule modal, the admin can select the day, class name, and the teacher assigned for that particular session. All created schedules are displayed in a structured table for easy monitoring. Administrators can edit or delete schedules when adjustments are needed, ensuring the timetable remains accurate and conflict-free. This feature helps maintain a smooth teaching process by clearly managing which teacher handles which class on each day.

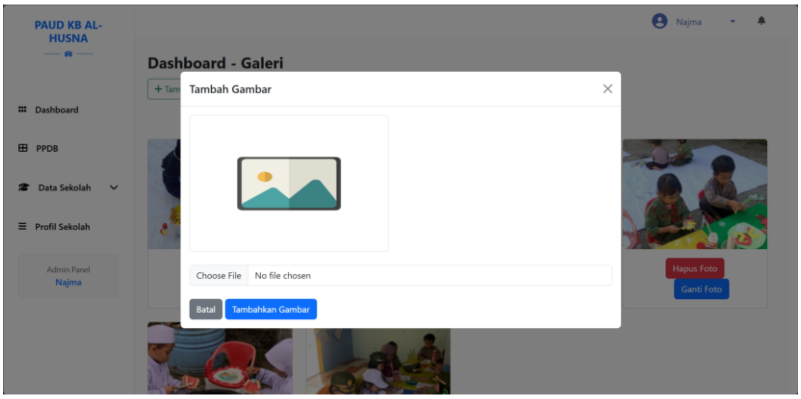


Figure 5. Gallery Management Menu

The Gallery Management menu allows the administrator to manage photo documentation of school activities. Through the Add Image modal, the admin can upload new pictures directly into the system. Each image displayed in the gallery can also be updated or removed when necessary. This feature is essential for showcasing school events, student achievements, learning activities, and other important moments to parents and the public. By keeping the gallery well-maintained and up-to-date, the school can present a visually appealing and informative online presence.

2.2 Test Case Design

This study employed a combination of two primary black box testing techniques: Equivalence Partitioning (EP) and Boundary Value Analysis (BVA). This combined approach was selected to provide broader and more efficient test coverage, particularly for form-based systems like the Admin Website PAUD KB Al Husna. The primary objective was to develop test cases that fully represent all input domains, encompassing both valid and invalid equivalence classes, and to rigorously evaluate system behavior at boundary thresholds, including minimum and maximum values [20]. Based on the functional requirements specifications for each module, all test designs were completed before execution.

2.2.1 Equivalence Partitioning (EP)

The Equivalency Partitioning (EP) technique divides input domains into equivalence classes that include categories that are both valid and invalid. Data that the system should accept is contained in valid classes, but data that should be rejected is included in invalid classes [21], [22]. Because each class is represented by one or more representative values, this approach reduces the number of test cases without sacrificing efficacy.

EP was applied to a number of text fields and select inputs in the Admin Website PAUD KB Al Husna modules, including Class, Teacher, FAQ, Schedule, and Gallery. Verifying that the system only takes correctly formatted inputs while rejecting all incorrect data types, such as symbols, foreign characters (Unicode), duplicates, or attempts at script injection, was the main goal. The designed equivalency classes for important input fields are summarized in the following table

Table 1. Equivalence Partitioning (EP) Class Design for Several Input Fields

Field	Valid Class	Invalid Class	Description
Class Name	Alphabetic text (e.g., “Bahagia”)	Symbols (“@#$”), numbers (“123”), Unicode characters (“□”), duplicate names, and HTML tags	Tests input sanitization and format validation
Homeroom Teacher	Selected from the list of teachers	Not selected	Tests mandatory input validation for *select* fields

Teacher Name	Alphabetic text	Numbers, symbols, Unicode characters, duplicate names	Tests format validation and duplication handling
Teacher Email	Valid email format (e.g., “name@gmail.com”)	Missing “@”, excessive symbols, HTML tags, or plain text	Tests email format and input sanitization
Address	Free text ≤ 255 characters	Empty, whitespace only, random symbols, or Unicode characters	Tests consistency of address data format
Day (Schedule)	Selected from list (Monday–Saturday)	Not selected	Tests mandatory input validation
Image File	JPG/PNG format	Non-image files (PDF/TXT)	Tests file type validation

This equivalence class design served as the foundation for developing initial test cases. This approach ensures that each test scenario demonstrates specific system behavior without redundancy.

2.2.2 Boundary Value Analysis (BVA)

The Boundary Value Analysis (BVA) technique was implemented to evaluate system behavior at input domain boundaries [23]. The core premise is that system defects are more likely to occur at minimum or maximum boundaries rather than within the central input domain [24]. Each BVA test case was designed to examine four critical boundary variations: Min−1 (one unit below minimum, expected rejection), Min (exact minimum, expected acceptance), Max (exact maximum, expected acceptance), and Max+1 (one unit above maximum, expected rejection) [25]. This methodology enables focused testing on error-prone boundary regions [26].

For the Admin Website PAUD KB Al Husna testing, BVA was primarily applied to text fields with character limitations, including Class Name, Teacher Name, Address, FAQ Title, and FAQ Answer. The maximum character limit followed system design specifications at 255 characters, while the minimum boundary was typically 1 character. The following table presents the boundary testing scheme designed for each field:

Table 2. Boundary Value Analysis (BVA) Design for Input Fields

Field	Minimum Boundary	Maximum Boundary	BVA Test Values	Purpose
Class Name	1 character	255 characters	Min−1: “ ”, Min: “a”, Max: 255×“a”, Max+1: 256×“a”	To test input length validation
Teacher Name	1 character	255 characters	Same as Class Name	To ensure input length consistency across forms
Teacher Email	1 character (valid format)	255 characters	Min−1: empty, Max+1: 256 characters	To test boundary limits for email length
Address	1 character	255 characters	Min−1: “ ”, Max+1: 256×“a”	To validate the maximum text boundary
FAQ Title	1 character	255 characters	Min−1: empty, Max+1: 256 characters	To evaluate error handling for input length
FAQ Answer	1 character	255 characters	Min−1: empty, Max+1: 256 characters	To test validation consistency within the FAQ section

This testing design validates system resilience under extreme input conditions, including both empty inputs and data exceeding specified limits. The subsequent Result and Discussion section details empirical findings, examining the system's effectiveness in blocking boundary-violating inputs and evaluating the precision of corresponding error notifications.

2.3 Testing Procedure and Execution

The testing of the “Admin PAUD KB Al Husna” web application was carried out through manual testing conducted by the Testerius team from the Software Engineering Technology study program at IPB University. The evaluation process adhered to structured testing standards to ensure that every component of the system met functional and non-functional requirements. The testing workflow was divided into three primary stages: general testing procedures (Figure 6), functional testing procedures (Figure 7), and non-functional testing procedures (Figure 8). Each stage contained detailed steps designed to validate system behavior, verify data processing accuracy, and ensure that user interactions operated as intended.

To guarantee that the application performed reliably across diverse environments, compatibility testing was conducted on multiple operating systems, specifically Windows and Linux. This cross-platform approach ensured that the system remained stable regardless of the user's device configuration. Furthermore, testing was performed using several widely used web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge, allowing evaluators to identify potential inconsistencies in rendering, responsiveness, or feature behavior. By combining systematic procedures, cross-platform assessment, and multi-browser testing, the evaluation provided a comprehensive overview of the application's quality, ensuring that the system is dependable, user-friendly, and ready for operational deployment.

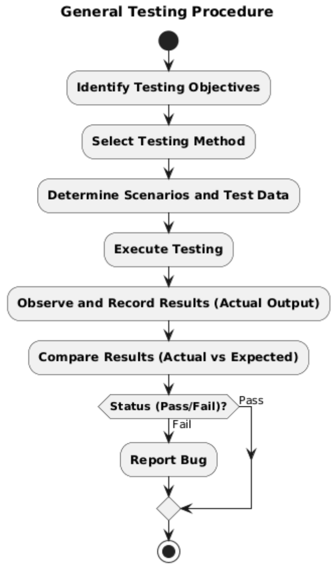

General Testing Procedure
Identify Testing Objectives
Select Testing Method
Determine Scenarios and Test Data
Execute Testing
Observe and Record Results (Actual Output)
Compare Results (Actual vs Expected)
Status (Pass/Fail)?
Pass
Fail
Report Bug

Figure 6. General testing procedures

Identification of testing objectives is the first step in this process, which is then followed by the choice of suitable testing techniques (such as EP or BVA for functional testing). Before the system is executed, test scenarios and data must be defined. After implementation, testers watch and record real outputs, which are then contrasted with anticipated outcomes. A pass/fail status is established based on this comparative study. When tests fail, defects are found and must be formally reported and fixed before the test is deemed complete.

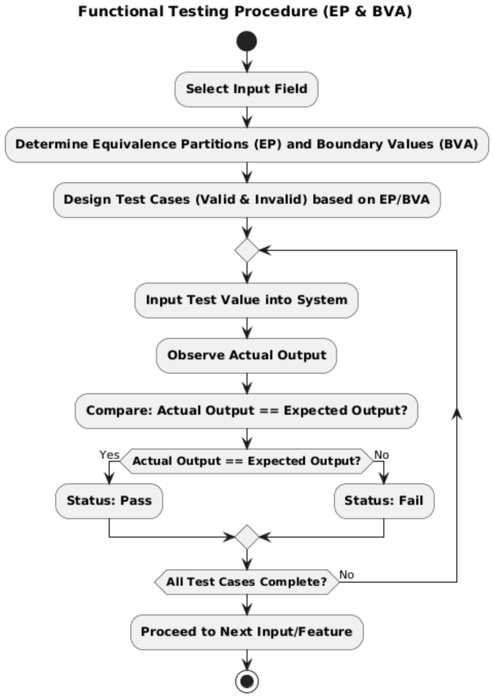

Functional Testing Procedure (EP & BVA)
Select Input Field
Determine Equivalence Partitions (EP) and Boundary Values (BVA)
Design Test Cases (Valid & Invalid) based on EP/BVA
Input Test Value into System
Observe Actual Output
Compare: Actual Output == Expected Output?
Yes
Actual Output == Expected Output?
No
Status: Pass
Status: Fail
All Test Cases Complete?
No
Proceed to Next Input/Feature

Figure 7. Functional testing procedure

This procedure specifically focuses on data input validation. It begins with selecting an input field (e.g., Class Name, Teacher Name), followed by defining Equivalence Partitions (EP) and Boundary Values (BVA). The application of EP and BVA frameworks enables systematic test case design, encompassing Valid values, Invalid inputs (such as symbols or duplicates), Minimum (Min) values, values just below minimum (Min-1), Maximum (Max) values, and values just above maximum (Max+1). This process is performed repeatedly as testers input test values into the system, review the system’s actual response, and compare it with the predetermined expected results. A matching result yields a Pass status; otherwise, a Fail status is recorded. Before moving on to the next input or functional feature, this cycle is repeated until all test cases have been finished.

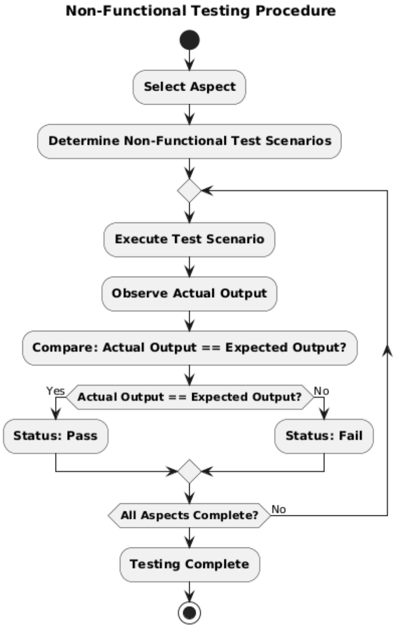

Non-Functional Testing Procedure
Select Aspect
Determine Non-Functional Test Scenarios
Execute Test Scenario
Observe Actual Output
Compare: Actual Output == Expected Output?
Yes
Actual Output == Expected Output?
No
Status: Pass
Status: Fail
All Aspects Complete?
No
Testing Complete

Figure 8. Non-functional testing procedure

This procedure is designed to assess the system's quality beyond its functional aspects. The process commences with the selection of non-functional attributes to be evaluated (such as login security, compatibility, or performance) and the definition of pertinent test scenarios. The tester subsequently executes these scenarios (e.g., attempting access without authentication or opening the application in different browsers). The observed actual outcomes are then compared against the expected results. A Pass status is assigned when the outcomes align (e.g., the dashboard loads in ≤ 3 seconds or the display remains consistent across all browsers), whereas a Fail status is recorded in cases of discrepancy (e.g., the account remains unlocked after five consecutive incorrect password attempts or the layout fails to be responsive). This iterative process continues until all designated nonfunctional aspects have been thoroughly examined.

3. Results and Discussion

3.1 Summary of Overall Testing Results

Testing was conducted on 82 test scenarios covering both functional modules (Class Management, Teacher Data, FAQ, Schedule, and Gallery) and non-functional modules (Login Security, Compatibility, and Performance). The testing results for each module are presented in the following table:

Table 3. Summary of Testing Results per Module

Test Module	Number of Scenarios	Pass Status	Fail Status	Success Rate
Class Management	15	9	6	60%
Teacher Data	22	14	8	63.6%
FAQ	15	9	6	60%
Schedule	6	5	1	83.3%
Gallery	3	3	0	100%

Login Security	2	1	1	50%
Compatibility and Responsiveness	5	4	1	80%
Total	82	45	27	67.1%

A total of 27 defects were detected out of 82 test scenarios executed, yielding a failure rate of 32.9%, calculated using the following formula (Figure 9):

[redacted] (1)

Figure 9. Failure Rate Formula

Overall, the system demonstrated satisfactory performance in its core components, particularly in data management and basic input validation. However, several significant issues were identified, notably concerning maximum boundary validation and the security of the system. The observed failure rate indicates that approximately one-third of the tested functional and non-functional requirements did not meet optimal input validation standards or failed to operate as expected. Findings related to weaknesses in data input validation mechanisms (BVA/EP-Invalid) and system security elements were the primary contributors to the elevated failure rate.

3.2 Discussion of Functional Testing Results

Functional testing using the Equivalence Partitioning (EP) and Boundary Value Analysis (BVA) approaches successfully identified several critical weaknesses in the system's data input validation mechanisms across various core modules.

3.2.1 Equivalence Partitioning (EP) Analysis

The system worked successfully while processing legitimate input, according to testing using the EP technique; nevertheless, when handling incorrect data, it showed significant failures, as seen in the following table:

Table 4. Summary of Equivalence Partitioning (EP) Testing Results

Field	Valid Class	Invalid Class	Status	Findings
Class Name	Alphabetic (Valid)	Symbols, HTML, Unicode	Partially Failed	The system accepts HTML tags <h1> without filtering
Teacher Name	Alphabetic (Valid)	Symbols, Unicode, Duplication	Partially Failed	Duplicate entries are not rejected
Teacher Email	Valid email format	Missing "@", excessive symbols, HTML	Mostly Passed	Email format validation is effective, but injection checking remains suboptimal
FAQ (Title & Answer)	Alphabetic	HTML, Unicode, Symbols	Partially Failed	Script injection is not yet sanitized
Schedule & Gallery	Validation of selection and file format	Empty input / incorrect format	Passed	All validations function as expected

Several critical weaknesses were identified, including the system's inability to detect and block HTML tags such as <h1> and <p> within the *Class Name* and *FAQ* fields, thereby exposing potential vulnerabilities to Cross-Site Scripting (XSS) attacks. In addition, validation against special symbols (e.g., @, #, $) and Unicode characters in the *Teacher Name* and *Address* fields proved to be weak due to the absence of adequate error-handling mechanisms. Overall, the lack of comprehensive input sanitization renders the system susceptible to accepting various forms of malicious data that should have been rejected at an early stage, consequently increasing the potential risk of XSS exploits.

3.2.2 Boundary Value Analysis (BVA) Analysis

Testing using the BVA technique revealed inconsistent system behavior when handling boundary character values. The system effectively managed lower-bound conditions, successfully rejecting inputs below the minimum limit (Min−1) with appropriate warning messages, and correctly accepting inputs that matched the minimum boundary (Min). Conversely, significant weaknesses were observed at the upper boundary, particularly when the input exceeded the maximum limit (Max+1), as presented in the following table.

Table 5. Summary of Boundary Value Analysis (BVA) Testing Results

Field	Minimum Boundary	Maximum Boundary	Critical Test Case	Result	Findings
Class Name	1 character	255 characters	256 characters (Max+1)	Fail	No error message displayed
Teacher Name	1 character	255 characters	256 characters (Max+1)	Fail	Error handling did not appear
Teacher Email	1 character	255 characters	256 characters (Max+1)	Fail	No length restriction applied
Address	1 character	255 characters	256 characters (Max+1)	Fail	Input not rejected
FAQ (Title & Answer)	1 character	255 characters	256 characters (Max+1)	Fail	No upper-bound warning shown

In the *Class Name*, *Teacher Name*, *Address*, and *FAQ* modules, the system failed to provide clear error notifications or reject the excessive input. Although the surplus data might be automatically truncated, the absence of proactive user feedback represents a critical quality defect and indicates weak server-side validation. The root cause appears to stem from the lack of explicit input length restrictions implemented in the backend layer, a vulnerability that could lead to potential security risks such as buffer overflow and data storage inefficiency.

3.3 Discussion of Non-Functional Testing Results

Non-functional testing targeted three key elements: login security, cross-browser compatibility, and responsive design. The results revealed two critical defects in security and compatibility aspects, while performance met the basic standards but was not comprehensively evaluated.

3.3.1 Security Analysis

The security testing involved simulating unauthorized access to the admin page and performing five consecutive failed login attempts. The system successfully blocked access without authentication (Pass); however, it failed to enforce login attempt limitations. The absence of temporary account lockout or CAPTCHA after multiple failed attempts leaves the system vulnerable to brute-force attacks. This deficiency represents a high-risk vulnerability, as it enables attackers to repeatedly test password combinations without restriction.

3.3.2 Compatibility Analysis

Compatibility testing was conducted across major web browsers, including Chrome, Firefox, and Edge. The testing also encompassed various device types, such as desktop computers, laptops, and smartphones. Results indicated consistent rendering and functionality across all browsers, with all essential features performing as expected. However, minor inconsistencies in font scaling and element alignment were observed, particularly on certain browser versions. While these inconsistencies do not affect system functionality, they may impact the perceived interface quality and visual coherence. This finding indicates the necessity for further interface optimization across different platforms.

3.3.3 Responsiveness Analysis

Responsiveness testing focused on evaluating the system's adaptability to different screen resolutions and device orientations. While the application maintained functional accessibility across all devices, it exhibited significant layout degradation on smaller screens. Navigation menus were occasionally truncated or difficult to interact with, reducing overall User Experience (UX) and accessibility for mobile users. Although these issues did not hinder access to core functionalities, they negatively impacted the interface aesthetics and usability, suggesting the necessity for improved responsive design implementation and grid-based layout adjustments. The following table summarizes the non-functional testing results:

Table 6. Summary of Non-Functional Testing Results

Aspect	Test Scenario	Result	Remarks
Security	Five consecutive failed login attempts	Fail	No account lockout or CAPTCHA implemented
Compatibility	Testing across four major browsers	Pass	Consistent interface display

Responsiveness	Testing on mobile devices	Fail	Navigation layout not proportionally displayed

3.4 Analysis of Critical Defect Findings and Implications

Both functional and non-functional testing identified two types of serious flaws that had the most effects on the Admin Website PAUD KB Al Husna system's quality: security flaws and input validation flaws. A comprehensive study reveals that Equivalency Partitioning (EP) and Boundary Value Analysis (BVA) techniques successfully uncovered hidden vulnerabilities that have a direct impact on the security, data integrity, and reliability of the application.

3.4.1 Root Cause Analysis

Most defects originated from excessive reliance on client-side validation without adequate server-side revalidation. JavaScript-based validation can be easily bypassed by manipulating HTTP requests (e.g., through DevTools or cURL), allowing malicious inputs to be accepted. Furthermore, inconsistencies in character length specifications across forms and the absence of input length verification at the database layer contributed to failures in handling Max+1 boundary cases in BVA testing. This issue enables invalid data to be stored even though it may appear to be automatically truncated.

3.4.2 Critical Security Finding: Cross-Site Scripting (XSS) Vulnerability

The EP-Invalid testing revealed that the system failed to reject inputs containing HTML tags, such as <h1>class name</h1> in the *Class Name* form and <p> tags in the *FAQ* section. These inputs were accepted and could potentially be executed when rendered on the admin page, thereby creating a Cross-Site Scripting (XSS) vulnerability. The exploitable XSS risks include:

- Theft of admin session cookies through malicious scripts.
- Unauthorized modification of page content (defacement).
- Execution of unauthorized commands in the user’s browser.
- Institutional reputation damage due to manipulated content.

3.4.3 Impact of Findings on System Quality

These validation and security weaknesses could compromise functionality and expose systemic risks that can affect the entire application ecosystem. The impact ranges from data corruption to security breaches, particularly concerning for an educational institution that manages sensitive information about students and staff. The following table presents a summary of the impacts categorized by key quality aspects:

Table 7. Summary of Critical Defect Impacts on the System

Aspect	Impact
Data Quality	Increased occurrence of unauthorized entries, irregular symbols, or embedded HTML content compromising database integrity.
System Security	Vulnerable to XSS and brute-force login attacks, potentially exposing sensitive user and administrator data.
Reliability & User Experience (UX)	Failure of error handling at maximum boundaries causes interface instability, server resource leaks, and poor user experience on mobile devices.
Institutional Reputation	For educational systems such as PAUD KB Al Husna, data breaches or content manipulation could severely undermine the trust of parents and staff.

Overall, the lack of server-side validation and input sanitization constitutes a critical vulnerability that must be promptly addressed to ensure the system is secure, reliable, and fit for deployment in a production environment.

4. Conclusion

This study aims to implement black box testing using the Equivalence Partitioning (EP) and Boundary Value Analysis (BVA) techniques on the Admin Website of PAUD KB Al Husna to evaluate its functional validity and input validation robustness. This objective was consistently achieved, as outlined in the Introduction chapter and reinforced through an in-depth analysis in the Results and Discussion chapter. Out of 82 executed test scenarios, 27 defects were identified, resulting in a failure rate of 32.9%. This indicates significant systemic quality issues, particularly common in small-scale information systems that often overlook validation and security aspects.

Key findings include Cross-Site Scripting (XSS) vulnerabilities detected through EP-Invalid testing, where the system failed to filter HTML tags such as <h1> or <p>, thereby allowing the injection of malicious scripts. Furthermore, a lack of protection against brute-force attacks was identified in the login module due to the absence of attempt limitations or CAPTCHA mechanisms. BVA testing further revealed systemic weaknesses in upper-bound validation (Max+1), where inputs exceeding the maximum length were neither rejected nor flagged with warnings, although data might be automatically truncated. The combination of these two techniques proved highly effective in detecting hidden defects with direct implications for security, data integrity, and user experience.

Based on these findings, there are high-priority improvements for developers include the implementation of rigorous server-side validation for data length and type, input sanitization and output escaping to prevent XSS, the addition of rate limiting or CAPTCHA in the login process, and the optimization of responsive design using modern CSS frameworks to enhance accessibility on mobile devices. These improvements will significantly enhance the system's reliability, security, and usability.

In the long term, this research opens avenues for further development, such as comparisons between manual and automated testing using Selenium, exploration of other black box techniques like Cause-Effect Testing or Decision Table Testing, and integration of White-Box Testing for source code analysis. The results also encourage the development of a reusable EP-BVA-based automated testing framework applicable to similar educational systems. Thus, this study not only contributes to improving the quality of the Admin Website PAUD KB Al Husna but also enriches the software testing literature in the context of small-scale educational institutions through a systematic and measurable approach.

Acknowledgement

The authors also extends heartfelt appreciation to all individuals who have offered support through encouragement, advice, or assistance during the process of conducting and finalizing this study. Although not every person can be mentioned by name, the authors acknowledges with deep appreciation all parties who, in ways both big and small, contributed to the smooth progression and completion of this work. Their presence, guidance, and goodwill have been invaluable, and the authors remains sincerely thankful for every contribution made along this research journey.

References

[1] Hozairi, Buhari, S. Alim, and Rofiudin, *Panduan Komprehensif Pengujian Perangkat Lunak*. Bandung, Indonesia: Penerbit Widina Media Utama, 2024.

[2] R. S. Pressman, *Software engineering: a practitioner's approach*, 7th ed. Dubuque, IA: McGraw-Hill, 2010.

[3] A. Agustian, I. Andryani, S. Khoerunisa, A. Pangestu, and A. Saifudin, "Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web," *J. Teknol. Sist. Inf. Dan Apl.*, vol. 3, no. 3, p. 178, Aug. 2020, doi: 10.32493/jtsi.v3i3.5371.

[4] R. P. Fajar, "Teknik Boundary Value Analysis pada Blackbox Testing untuk Aplikasi Buku Catatan Harian," *Repositor*, vol. 6, no. 1, pp. 69–78, Feb. 2024.

[5] H. A. S. Hutapea, Y. Priyadi, and E. Darwiyanto, "Analisis dan Pengujian dengan Menggunakan Metode Boundary Value Analysis dan Metode Equivalence Partitioning (Studi Kasus: Aplikasi Homelab)," *E-Proceeding Eng.*, vol. 9, no. 2, p. 746, Apr. 2022.

[6] Richard Gunawan, Yohanes Priadi Wibisono, Clara Hetty Primasari, and Djoko Budiyanto, "Blackbox Testing on Virtual Reality Gamelan Saron Using Equivalence Partition Method," *J. Buana Inform.*, vol. 14, no. 01, pp. 11–19, Apr. 2023, doi: 10.24002/jbi.v14i01.6606.

[7] F. Wardah Gracillaria Suharyono, K. Kartini, and A. Junaidi, "PENERAPAN METODE BOUNDARY VALUE ANALYSIS DAN EQUIVALENCE PARTITIONING DALAM PENGUJIAN BLACK BOX UNTUK APLIKASI SIADITA," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 1, pp. 1013–1020, Mar. 2024, doi: 10.36040/jati.v8i1.8921.

[8] D. Ahrizal, M. K. Miftah, R. Kurniawan, T. Zaelani, and Y. Yulianti, "Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 73, Mar. 2020, doi: 10.32493/informatika.v5i1.4338.

[9] 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 (Study Case: Academic Information System of Mataram University)," in *Proceedings of the First Mandalika International Multi-Conference on Science and Engineering 2022, MIMSE 2022 (Informatics and Computer Science)*, I. G. P. S. Wijaya, J. Hwang, A. M. Widodo, and B. Irawan, Eds., Dordrecht: Atlantis Press International BV, 2022, pp. 207–219. doi: 10.2991/978-94-6463-084-8_19.

[10] M. Sholeh, I. Gisfas, Cahiman, and M. A. Fauzi, "Black Box Testing on ukmbantul.com Page with Boundary Value Analysis and Equivalence Partitioning Methods," *J. Phys. Conf. Ser.*, vol. 1823, no. 1, p. 012029, Mar. 2021, doi: 10.1088/1742-6596/1823/1/012029.

[11] F. K. Kartono *et al.*, "Pengujian Black Box Testing Pada Sistem Website Osha Snack: Pendekatan Teknik Boundary Value Analysis," *J. KRIDATAMA SAINS DAN Teknol.*, vol. 6, no. 02, pp. 754–766, Dec. 2024, doi: 10.53863/kst.v6i02.1407.

[12] B. Hardika *et al.*, “Pengujian Blackbox Testing Website Garuda Farm Menggunakan Teknik Equivalence Partitioning,” *J. KRIDATAMA SAINS DAN Teknol.*, vol. 6, no. 02, pp. 740–753, Dec. 2024, doi: 10.53863/kst.v6i02.1420.

[13] A. Amrulloh, A. D. Septiadi, M. Septiara, and P. A. Wicaksono, “Black Box Testing Using the Equivalence Partitions Technique to Test the Functionality of the Ternaku.id Website,” *J. Multimed. Trend Technol.*, vol. 2, no. 3, pp. 171–178, Dec. 2023, doi: 10.35671/jmtt.v2i3.43.

[14] Y. Nam and S. Choi, “Analysis of Vulnerabilities in College Web-Based System,” *Electronics*, vol. 13, no. 12, p. 2261, June 2024, doi: 10.3390/electronics13122261.

[15] I. Indrianto and E. Edwar, “Vulnerability Evaluation for Student Enrollment at SMKS Pandawa Bali Global Abiansemal,” *Sebatik*, vol. 28, no. 2, Dec. 2024, doi: 10.46984/sebatik.v28i2.2510.

[16] S. Bose and A. K. Narayanan, “Security Analysis of CMS based Websites through CMSPY,” *ICRRD J.*, vol. 4, no. 4, pp. 162–174, 2023, doi: 10.53272/icrrd.

[17] G. Wassermann and Z. Su, “Static detection of cross-site scripting vulnerabilities,” in *Proceedings of the 13th international conference on Software engineering - ICSE ’08*, Leipzig, Germany: ACM Press, 2008, p. 171. doi: 10.1145/1368088.1368112.

[18] R. Verma, N. Dhanda, and V. Nagar, “Enhancing Security with In-Depth Analysis of Brute-Force Attack on Secure Hashing Algorithms,” in *Proceedings of Trends in Electronics and Health Informatics*, vol. 376, M. S. Kaiser, A. Bandyopadhyay, K. Ray, R. Singh, and V. Nagar, Eds., in Lecture Notes in Networks and Systems, vol. 376. , Singapore: Springer Nature Singapore, 2022, pp. 513–522. doi: 10.1007/978-981-16-8826-3_44.

[19] G. Wijaya, H. Winata, S. Aji, M. N. Faiz, and H. Haeruddin, “Website Security Analysis Using Vulnerability Assessment Method,” *J. Innov. Inf. Technol. Appl. JINITA*, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:275522979

[20] X. Guo, H. Okamura, and T. Dohi, “Optimal test case generation for boundary value analysis,” *Softw. Qual. J.*, vol. 32, no. 2, pp. 543–566, June 2024, doi: 10.1007/s11219-023-09659-9.

[21] P. Huriati, H. Azmi, Y. Wati, D. Meidelfi, and T. Lestari, “Black box testing on the online quiz application using the Equivalence Partitions method,” *Int. J. Adv. Sci. Comput. Eng.*, vol. 2, no. 2, pp. 51–56, Aug. 2020, doi: 10.62527/ijasce.2.2.48.

[22] M. Nasir *et al.*, “Implementasi Equivalence Partitioning Testing Pada Fitur Booking dan Jadwal Website Praktik Gigi Mandiri drg. Susilawati (https://frontend-webklinik.vercel.app/),” *STRING Satuan Tulisan Ris. Dan Inov. Teknol.*, vol. 9, no. 3, p. 371, Apr. 2025, doi: 10.30998/string.v9i3.26570.

[23] C. Vikasari, “Pengujian Sistem Informasi Magang Industri dengan Metode Blackbox Testing Boundary Value Analysis,” *Syntax J. Inform.*, vol. 7, no. 1, pp. 44–51, June 2018, doi: 10.35706/syji.v7i1.1291.

[24] S. J. Putri, D. G. P. Putri, and W. H. N. Putra, “Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars),” *J. Internet Softw. Eng.*, vol. 5, no. 1, pp. 23–28, May 2024, doi: 10.22146/jise.v5i1.9446.

[25] Mohd. Ehmer Khan, “Different Approaches To Black box Testing Technique For Finding Errors,” *Int. J. Softw. Eng. Appl.*, vol. 2, no. 4, pp. 31–40, Oct. 2011, doi: 10.5121/ijsea.2011.2404.

[26] I. E. Tsalatsah, D. Pratama, A. R. Hakim, L. A. Budiman, and J. Riyanto, “Penggunaan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Stok Barang,” *J. Teknol. Sist. Inf. Dan Apl.*, vol. 5, no. 1, p. 14, Jan. 2022, doi: 10.32493/jtsi.v5i1.14987.