JURNAL TEKNOLOGI DAN OPEN SOURCE
Vol. 8, No. 2, December 2025, pp. 1064~1070
e-ISSN: 2622-1659, accredited Four Grade by Kemenristekdikti, Decree No: 152/E/KPT/2023
DOI: https://doi.org/10.36378/jtos.v8i2.5084

Functional Testing of Web-Based Input Form Using Boundary Value Analysis Technique: A Case Study on TemanTernak

Akhmad Zufar Rafif[1], Muhammad Rafi Alexander Prayoga[2], Muhammad Faris Fadhil Islam[3], Raden Muhammad Raditya Rahman[4], Sofiyanti Indriasari[5], Muhammad Nasir[6]
[1-6] Software Engineering Technology, Vocational Studies of IPB University, Indonesia

Article Info

Article history:

Received 11 28, 2025
Revised 12 11, 2025
Accepted 12 20, 2025

Keywords:

Software Testing Life Cycle,
Black Box Testing,
Boundary Value Analysis,
Input Validation,
Web Application

ABSTRACT

Software reliability is highly dependent on robust data validation mechanisms to prevent data integrity issues. The main issue raised in this study is the risk of input errors in the "Create Doctor Invitation" form in the TemanTernak web application, which could potentially cause invalid data storage. This study aims to evaluate the quality of the system's input validation to ensure its compliance with the requirements specifications. The proposed solution is the implementation of Black Box Testing with the Boundary Value Analysis (BVA) technique. The testing methodology was carried out following the Software Testing Life Cycle (STLC) stages, in which 17 test scenarios were run to assess four critical elements: name and title, email, phone number, and invitation message. The test results revealed that the system only achieved a functional success rate of 41.2%. Although the application successfully processed standard valid inputs, the system failed to reject the majority of inputs that were outside the tolerance limits (off-boundary) and passed empty inputs in mandatory fields. In conclusion , the current validation mechanism is insufficient to handle extreme conditions. In the future, the implementation of strict server-side validation with range constraints is necessary to mitigate these vulnerabilities.

Corresponding Author:

Sofiyanti Indriasari
Software Engineering Technology
Vocational Studies of IPB University
Bogor, Indonesia
Email: sofiyanti@apps.ipb.ac.id

1. Introduction

Software quality is a fundamental aspect that determines the success of an information system in serving user needs. In modern software engineering, software quality assurance plays a critical role as an integral component in minimizing the risk of system failure [1]. One of the biggest vulnerabilities in web-based applications lies in the interface input validation mechanism, which often becomes a gateway for cyber attacks [2]. Errors in handling data constraints, such as data type mismatches and character length violations, are not just interface display issues, but serious threats to database integrity. If the system fails to filter invalid input on a form, this can lead to garbage data storage, information ambiguity, and even fatal security vulnerabilities. Therefore, specific testing methods are needed to verify system behavior under extreme conditions to prevent such data corruption.

The Black-Box Testing method is the industry standard for validating interface functionality without the need to dissect the internal code structure [3]. Among the various techniques in this method, Boundary

Value Analysis (BVA) is considered the most effective for detecting errors that occur at the "edges" of the input domain. Unlike other testing techniques, BVA specifically tests values at the edges of the input boundary, such as minimum and maximum values, to ensure the reliability of the application in handling extreme conditions [4]. This approach is based on the statistical principle that the majority of software errors (bugs) tend to hide at the boundaries of the data range rather than in the middle [5].

The effectiveness of the BVA technique has been proven in various recent case studies. Wicaksono A. [6] showed that a combination of BVA was able to detect validation gaps in the Nestify property management system with high accuracy. Similar findings were presented by Suharyono et al. [7] in a travel ticket application, where BVA successfully revealed system failures in handling user input character limits. In more complex systems such as expert diagnosis applications [8] and coding workshop platforms [9], this technique has proven reliable in ensuring that system algorithms do not process values outside the permitted range.

The application of BVA is also highly relevant in the administration and public service sectors. Studies on library systems [10] and community services [11] found that numeric input errors often slip through standard validation but can be caught through boundary test scenarios. This is reinforced by research on academic systems [12], which concluded that lower and upper boundary testing is crucial to prevent garbage data from being stored in databases. Even in comparative studies, BVA has been proven to have high accuracy in validating numerical data input in web-based systems [13], and is recommended as a crucial method for ensuring the robustness of applications against automatic input anomalies [14].

In this study, the BVA technique was applied to the TemanTernak application, a digital livestock consulting platform. The focus of the study was on the "Create Doctor Invitation" form, which contains critical data elements such as Mobile Number (with a specific digit limit) and Name and Title (with a character length limit). To date, testing on TemanTernak has not touched on the aspect of limit value validation in depth, so the risk of invalid data slipping through is still high. This study aims to fill this gap by designing limit test scenarios to identify whether the system is able to reject inputs that violate digit and character limits, thereby ensuring the integrity of communication data between farmers and veterinarians.

2. Research Method

This study applies the Black-Box Testing experimental method with a systematic Software Testing Life Cycle (STLC) approach. STLC was chosen as the main framework because it offers structured, sequential, and standardized testing stages, ensuring that every aspect of software functionality is thoroughly validated before release to end users [15]. The STLC stages used in this study are illustrated in Figure 1.

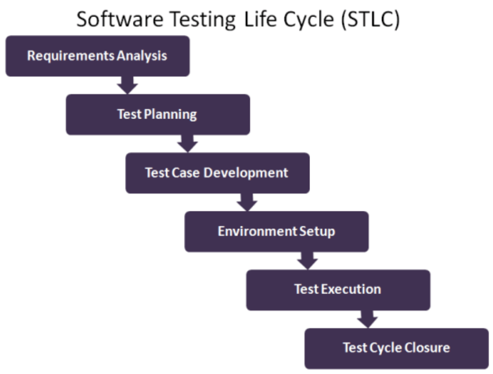

Figure 1. Software Testing Life Cycle (STLC) stages (Source: https://1000bh.wordpress.com/2019/01/13/software-testing-life-cycle-stlc)

2.1. Requirement Analysis

This stage focuses on analyzing functional requirements based on empirical observations of the user interface (UI). Given that testing was conducted using the Black Box method without access to internal specification documents, validation rules were determined by identifying input constraints applied to HTML form elements and general business logic standards. The results of this identification are used as a baseline for determining boundary values in the design of test scenarios [19].

2.2. Test Planning

At the planning stage, a comprehensive testing strategy was developed. This activity includes determining the scope of testing with a primary focus on functional validation of web-based applications as applied to similar online reporting systems [18]. In addition, tools such as web browsers and log recording software are selected, and the resources needed to ensure efficient testing are estimated.

2.3. Test Case Development

This stage involves designing detailed test cases based on the results of the requirements analysis. The scenarios are designed to cover various possible inputs, both valid and invalid data, to test the system's resilience to user errors. The determination of these test cases refers to the best practices of online registration system testing, which requires coverage of positive and negative scenarios [23]. In addition, the design of scenarios also considers specific input limitations to match the characteristics of the data to be tested [21].

2.4. Test Environment Setup

Before execution begins, the test environment is set up to resemble real-world conditions. The strategy applied is to conduct testing on an active deployment server that can be accessed via the public internet, rather than just in a local development environment (localhost). This approach aims to validate the system's performance and response to input validation in actual network conditions, as recommended in web-based information system testing procedures [25].

2.5. Test Execution

The execution stage is the core phase where all test scenarios are run on the application. At this stage, the Boundary Value Analysis (BVA) technique is applied in depth as the main verification method. BVA is a Black Box testing technique based on the principle that software errors (bugs) have the highest probability of appearing at the "edges" or boundaries of the input domain, rather than in the middle [16]. In its application, each input variable on the form is tested using four critical boundary values to ensure range validation accuracy:

1. Lower Bound Value (Min): The smallest value allowed by system specifications (Expected: Valid/Accepted)
2. Value Just Below the Limit (Min - 1): The value that is one unit below the minimum limit (Expected: Invalid/Rejected)
3. Upper Limit Value (Max): The largest value allowed by the system specifications (Expected: Valid/Accepted)
4. Value Exactly Above the Limit (Max + 1): A value that is one unit above the maximum limit (Expected: Invalid/Rejected)

In addition, testing also includes null value validation to ensure that mandatory fields cannot be skipped, as applied in testing the submission system application [22]. The main purpose of implementing this strict BVA is to verify whether the system is capable of providing the correct error message when receiving input outside the tolerance [20]. This method also aims to ensure that the system receives valid data without unnecessary technical obstacles [17].

2.6. Test Cycle Closure

The closing stage of the testing cycle focuses on evaluating the execution results. All actual results are compared with the expected results to calculate the success metrics. At this stage, test reports are compiled, anomalies or bugs are documented, and final conclusions regarding the system's reliability are drawn as recommendations for improvement for developers [24].

3. Results and Discussion

This section comprehensively describes the results obtained from applying the Black-Box Testing method to the TemanTernak application, a web-based digital platform that facilitates animal health consultations. The main focus of the discussion is the evaluation of the reliability of the "Create Doctor Invitation" feature. This test aims to verify whether the validation mechanism in the form is adequate in handling user input variations to prevent data integrity risks.

The presentation of results is systematically organized following the six stages of the Software Testing Life Cycle (STLC) as defined in the research methodology. The discussion flow moves from Requirement Analysis, Test Planning, Test Case Development, Test Environment Setup, Test Execution, to

Test Cycle Closure. The entire series of testing activities was carried out directly on the active hosting environment to ensure that the evaluation data obtained reflects the actual performance and response of the system when accessed by end users.

3.1. Requirement Analysis

At this stage, validation logic mapping was carried out through form element inspection on the TemanTernak application. The researchers determined the testing parameters based on the implicit specifications seen on the interface. Based on the form inspection results (see Figure 2), the following input variable constraints (IVC) were obtained:

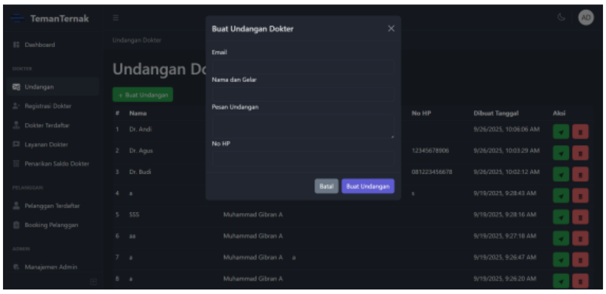

Figure 2. "Create Doctor Invitation" Input Form Display on the TemanTernak Web Application

The first variable is Name & Title, which is mandatory with a string data type and a minimum character length of 3 and a maximum of 50 characters. The second variable is Email, which is also mandatory with validation focused on character length limits in accordance with database storage allocation standards. The third variable, Mobile Number, requires numeric input with a digit length range of 10 to 13 digits in accordance with mobile operator standards. Finally, Invitation Message is a mandatory text field with a specific character length limit to prevent technical risks such as buffer overflow.

3.2. Test Planning

The output of the planning stage is the establishment of a manual testing strategy using the Black Box Testing approach. This strategy focuses on the functional validation of web-based applications, including client-side and server-side validation checks, as applied in the design of similar web-based information systems. The tools used include the Google Chrome browser for interface execution and Microsoft Excel for recording test logs.

3.3. Test Case Development

Based on the variables found in the analysis stage, 17 test cases were successfully designed. This design table serves as the main guide for test execution. The following is a summary of the scenario structure that has been developed:

Table 1. Test Case Design

ID	Section Name	Scenario Description
TC-01	Name & Title	Testing empty input
TC-02	Name & Title	Testing 2-character input (min-1)
TC-03	Name & Title	Testing 3-character input (min)
TC-04	Name & Title	Testing 50-character input (max)
TC-05	Name & Title	Testing input 51 characters (max+1)
TC-06	Email	Testing empty input
TC-07	Email	Testing 6-character input (lower limit)
TC-08	Email	Testing 255-character input (upper limit)
TC-09	Phone number	Testing empty input
TC-10	Phone number	Testing 9-digit input (min-1)
TC-11	Phone number	Testing 10-digit input (min)
TC-12	Phone number	Testing 13-digit input (max)
TC-13	Phone number	Testing 14 digits input (max+1)
TC-14	Invitation message	Testing empty input
TC-15	Invitation Message	Testing 4-character input (min-1)
TC-16	Invitation Message	Testing 11-character input (min)
TC-17	Invitation Message	Testing input 256 characters (max+1)

3.4. Test Environment Setup

The test environment has been successfully configured on the production server (live server) and verified as ready for use on November 14, 2025. The TemanTernak test version application has been deployed and can be accessed stably via the domain http://admin-temanternak.test.h14.my.id:8888/. Pre-execution checks on that date ensured that the database and server connections were running normally without any technical issues that could skew the test results.

3.5. Test Execution

At this stage, 17 test scenarios were run to verify the system response (Actual Result) to various boundary input conditions. The results were then validated against the expected results (Expected Output) to determine the pass status of the scenario. A summary of the test results comparison is summarized in Table 2.

Table 2. Summary of Test Results for the "Create Doctor Invitation" Form

ID	Section Name	Input Scenario (Boundary Value)	Boundary Value Type (BVA)	Expected Output	Actual Result	Status
TC-01	Name & Title	Input: Empty (0 Characters)	Null / Min-1	Rejected	System rejects input	Valid
TC-02	Name & Title	Input: "Ab" (2 Characters)	Min - 1	Rejected	System accepts input	Invalid
TC-03	Name & Title	Input: "Abu" (3 Characters)	Min	Accepted	The system accepts the input	Valid
TC-04	Name & Title	Input: 50-character string	Max	Accepted	The system accepts input	Valid
TC-05	Name & Title	Input: 51-character string	Max + 1	Rejected	System accepts input	Invalid
TC-06	Email	Input: Empty (0 characters)	Null / Min-1	Rejected	The system rejected the input	Valid
TC-07	Email	Input: "d@g.co" (6 characters)	Minimum Valid Length	Rejected	System accepts input	Invalid
TC-08	Email	Input: String 255 characters	Max Length	Rejected	System accepts input	Invalid
TC-09	Phone number	Input: Empty (0 Digits)	Null / Min-1	Rejected	System accepts input	Invalid
TC-10	Phone number	Input: "081234567" (9 digits)	Min - 1	Rejected	System accepts input	Invalid
TC-11	Phone Number	Input: "0812345678" (10 digits)	Min	Accepted	System accepts input	Valid
TC-12	Phone Number	Input: "0812345678901" (13 digits)	Max	Accepted	The system accepts the input	Valid
TC-13	Phone Number	Input: "08123456789012" (14 digits)	Max + 1	Rejected	System accepts input	Invalid
TC-14	Invitation Message	Input: Empty (0 characters)	Null / Min-1	Rejected	System accepts input	Invalid
TC-15	Invitation Message	Input: "Hello" (4 characters)	Min - 1	Rejected	System accepts input	Invalid
TC-16	Invitation Message	Input: "Hello Doctor" (11 characters)	Min	Accepted	The system accepts the input	Valid
TC-17	Invitation Message	Input: 256-character string	Max + 1	Rejected	System accepts input	Invalid

3.6. Test Cycle Closure

The test cycle closure phase focuses on a comprehensive evaluation of the execution results to determine the quality of the software objectively. Based on the data recapitulation in Table 2, the system validation reliability level was calculated using the Pass/Fail Rate metric. Of the total 17 test scenarios executed, only 7 scenarios were declared valid (in accordance with specifications), while the other 10 scenarios were declared invalid (failed). Thus, the test success rate can be calculated as follows:

$$Success\ Rate = \left(\frac{Valid\ Scenario}{Total\ Test\ Case}\right) \times 100\% \tag{1}$$

$$Success\ Rate = \left(\frac{7}{17}\right) \times 100\% \approx 41{,}2\%$$

(2)

The success rate of 41.2% indicates that the majority of the input validation mechanisms in the "Create Doctor Invitation" form (58.8%) failed to handle test data variations. A more in-depth analysis of the failed scenarios revealed a pattern of critical vulnerabilities, where the system was found to accept inputs that were outside the tolerance range (off-boundary), such as characters below the minimum limit and above the maximum limit in the Name, Mobile Number, and Message variables. In addition, inconsistencies were found in the handling of mandatory fields, where critical fields could be left blank without triggering an error message.

The fact that all invalid data was successfully processed and stored in the system indicates the absence of a defense layer on the backend (lack of server-side validation). This condition poses a serious risk to data integrity. Therefore, this study recommends that the development team immediately implement validation logic improvements, specifically the application of strict range constraints on the server side to prevent garbage data from entering the database.

4. Conclusion

This study concludes that the evaluation of the reliability of input validation on the "Create Doctor Invitation" form in the TemanTernak application shows a significant data integrity gap when tested using the Boundary Value Analysis (BVA) technique. Based on the execution of 17 boundary test scenarios, the system only achieved a functional success rate of 41.2%. A critical vulnerability was identified in the system's failure to handle inputs outside the tolerance limits (off-boundary), where data with a character length below the minimum (min-1) and above the maximum (max+1), as well as empty inputs in required fields, were still processed by the system. To mitigate the risk of garbage in data storage, further development must prioritize the implementation of server-side validation that strictly enforces range constraints. In addition, future studies are recommended to integrate security testing and automated testing so that boundary value anomaly detection can be performed more comprehensively and efficiently.

.

Acknowledgement

The author would like to express his highest appreciation to IPB University for the facilities and support provided so that this research could be completed successfully.

References

[1] B. Respiar, A. Fernanda, Taryadi, F. Maulana, and A. H. Anshor, “Peran Penting Software Quality Assurance Dalam Pengembangan Aplikasi,” *Journal Scientific of Mandalika*, vol. 5, no. 12, pp. 535–540, 2024.

[2] N. Hidayat and M. A. Nugroho, “Analisis Celah Keamanan Pada Website SMA NEGERI 3 BERAU Dengan Metode Penetration TESTING,” *Journal of Information System Management (JOISM)*, vol. 6, no. 2, pp. 102–108, Nov. 2025, doi: 10.24076/joism.2025v6i2.1858.

[3] S. D. Pratama, L. Lasimin, and M. N. Dadaprawira, “Pengujian Black Box Testing Pada Aplikasi Edu Digital Berbasis Website Menggunakan Metode Equivalence Dan Boundary Value,” *J-SISKO TECH (Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD)*, vol. 6, no. 2, p. 560, Nov. 2023, doi: 10.53513/jsk.v6i2.8166.

[4] R. P. Fajar, “Teknik Boundary Value Analysis pada Blackbox Testing untuk Aplikasi Buku Catatan Harian,” *Jurnal Repositor*, vol. 6, no. 1, Nov. 2024, doi: 10.22219/repositor.v6i1.31852.

[5] A. S. Wulandari, A. Saepudin, M. P. Kinanti, Z. Sudesi, A. Saifudin, and Y. Yulianti, “Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning,” *Jurnal Teknologi Sistem Informasi dan Aplikasi*, vol. 5, no. 2, p. 102, Nov. 2022, doi: 10.32493/jtsi.v5i2.17561.

[6] A. R. Wicaksono, “Analisis Kinerja Pengujian Black Box Terhadap Website Nestify Setelah Implementasi Equivalence Partitioning dan Boundary Value Analysis,” *Jurnal Informatika dan Teknologi Pendidikan*, vol. 5, no. 1, pp. 77–98, Nov. 2025, doi: 10.59395/jitp.v5i1.131.

[7] F. W. G. Suharyono, K. Kartini, and A. Junaidi, “Penerapan Metode Boundary Value Analysis dan Equivalence Partitioning Dalam Pengujian Black Box Untuk Aplikasi SIADITA,” *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 1, pp. 1013–1020, Nov. 2024, doi: 10.36040/jati.v8i1.8921.

[8] N. Mahrozi and M. A. Yaqin, "PENGUJIAN APLIKASI DENGAN METODE BLACKBOX TESTING: ANALISIS BOUNDARY VALUE DAN EQUIVALENCE PARTITIONING PADA APLIKASI SISTEM PAKAR KUCING," 2024.

[9] C. E. E. S. Putri and A. Susanto, "Feasibility Analysis of Bengkel Koding Website Using Black Box Testing and Boundary Value Analysis," *Sinkron*, vol. 8, no. 2, pp. 764–776, Nov. 2024, doi: 10.33395/sinkron.v8i2.13589.

[10] M. S. R. Langit, A. Voutama, and A. A. Ridha, "Black Box Testing pada Website Sistem Perpustakaan Menggunakan Metode Equivalence Partitioning dan Analisis Boundary Value," *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, vol. 8, no. 3, p. 355, Nov. 2024, doi: 10.30998/string.v8i3.19529.

[11] E. Z. Selviana, H. E. Wahanani, and F. P. Aditiawan, "Black Box Testing in Community Service Systems Using Boundary Value Analysis and Cause Effect Graph Methods (Case Study of Jombang District Community and Village Empowerment Service)," *East Asian Journal of Multidisciplinary Research*, vol. 2, no. 10, pp. 4161–4184, Nov. 2023, doi: 10.55927/eajmr.v2i10.6435.

[12] A. Jailani and M. A. Yaqin, "Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis," *Journal Automation Computer Information System*, vol. 4, no. 2, pp. 60–66, Nov. 2024, doi: 10.47134/jacis.v4i2.78.

[13] F. K. Kartono *et al.*, "Pengujian Black Box Testing Pada Sistem Website Osha Snack: Pendekatan Teknik Boundary Value Analysis," *Jurnal Kridatama Sains dan Teknologi,* vol. 6, no. 2, pp. 754-766, Des. 2024, doi: 10.53863/kst.v6i02.1407.

[14] F. Dobslaw, R. Feldt, and F. G. D. O. Neto, "Automated black-box boundary value detection," *PeerJ Comput Sci*, vol. 9, p. e1625, Nov. 2023, doi: 10.7717/peerj-cs.1625.

[15] T. S. Jaya, P. S. M. Informatika, J. E. dan Bisnis, and P. N. L. JlnSoekarno, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*, vol. 03, no. 02, 2018.

[16] S. R. Yulistina, T. Nurmala, R. M. A. T. Supriawan, S. H. I. Juni, and A. Saifudin, "Penerapan Teknik Boundary Value Analysis untuk Pengujian Aplikasi Penjualan Menggunakan Metode Black Box Testing," *Jurnal Informatika Universitas Pamulang*, vol. 5, no. 2, p. 129, Nov. 2020, doi: 10.32493/informatika.v5i2.5366.

[17] J. Nadhifah *et al.*, "Black Box Testing on the Wingpos Website Using the Equivalence Partitioning Technique," *International Journal of Information Engineering and Science*, vol. 1, no. 4, pp. 81–88, Nov. 2024, doi: 10.62951/ijies.v1i4.128.

[18] T. A. I. Alvayet and E. V. Barrichelo, "Perancangan Sistem Informasi Pengolahan Data Laporan Pajak Bulanan Berbasis Web Pada Depo Unilever Padang," *Jurnal Sains Informatika Terapan*, vol. 2, no. 3, pp. 108–113, Nov. 2023, doi: 10.62357/jsit.v2i3.202.

[19] G. W. Sasmito, "Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 2, no. 1, pp. 6–12, Nov. 2017, doi: 10.30591/jpit.v2i1.435.

[20] M. Sutomo and F. E. B. Simamora, "Pengujian Black Box Aplikasi Inventaris dengan Boundary Value Analysis di PT Duo Roda Motor," *DEVICE : JOURNAL OF INFORMATION SYSTEM, COMPUTER SCIENCE AND INFORMATION TECHNOLOGY*, vol. 5, no. 1, pp. 202–216, Nov. 2024, doi: 10.46576/device.v5i1.4557.

[21] A. M. Wilhelmus and D. Marselio, "Penerapan BVA dan EP pada Aplikasi Pencatatan Bahan dan Produksi," *Jurnal Sistem dan Teknologi Informasi Komunikasi*, vol. 8, no. 1, pp. 1–9, Nov. 2024, doi: 10.32524/jusitik.v8i1.1180.

[22] C. Kartiko, "BLACK BOX TESTING BOUNDARY VALUE ANALYSIS PADA APLIKASI SUBMISSION SYSTEM," *Edik Informatika*, vol. 6, no. 2, pp. 15–22, Nov. 2020, doi: 10.22202/ei.2020.v6i2.3995.

[23] M. Rizal and H. Endah, "PENGUJIAN BLACK BOX SISTEM PENDAFTARAN ONLINE DENGAN METODE EQUIVALENCE PARTITIONING DAN BOUNDARY VALUE ANALYSIS," *Jurnal Informatika dan Sistem Informasi (JIFoSI)*, vol. 1, no. 2, 2020.

[24] A. M. Astripat, M. Faisal, and A. Saifudin, "Pengujian Software Sistem Informasi Akademik dengan Black Box," *Jurnal Riset Informatika dan Inovasi*, vol. 1, no. 4, 2023.

[25] R. J. Meliala *et al.*, "Web-Based Financial Information System Testing of PT Perta Sakti Abadi Using the Black Box Testing Method," *International Journal of Computer Technology and Science*, vol. 2, no. 1, pp. 58–66, Nov. 2024, doi: 10.62951/ijcts.v2i1.129.