JACIS : Journal Automation Computer Information System Volume: **4**, Nomor **2**, **2024**, *Hal*: 60-66

Pengujian Aplikasi Sistem Informasi Akademik menggunakan Metode Blackbox dengan Teknik Boundary Value Analysis

Ahmad Jailani*[1], Muhammad Ainul yaqin[2]

[1] Universitas Islam Negeri Maulana Malik Ibrahim Malang; 230605210020@student.uin-malang.ac.id
[2] Universitas Islam Negeri Maulana Malik Ibrahim Malang; yaqinov@ti.uin-malang.ac.id

Abstrak: Pengujian perangkat lunak adalah aspek krusial dalam riset untuk memastikan mutu dan perkembangan aplikasi. Dalam konteks pengujian sistem informasi akademik, pentingnya desain yang baik memungkinkan deteksi kesalahan yang cepat, menghemat waktu dalam proses pengujian, dan membangun kepercayaan pengguna terhadap aplikasi. Penelitian terkait mengidentifikasi masalah dalam aplikasi, seperti kesalahan input informasi, yang memicu strategi pengujian yang difokuskan pada memastikan kesesuaian dan akurasi informasi. Metode pengujian seperti Black Box Testing dengan Boundary Value Analysis digunakan untuk mengevaluasi kualitas aplikasi dan mengidentifikasi area untuk perbaikan. Kesimpulan dari penelitian ini menunjukkan bahwa aplikasi berhasil hanya sekitar 60% dari jumlah percobaan maka penting dilakukan peningkatan kualitas aplikasi, dengan fokus pada memastikan bahwa aplikasi berfungsi sebagaimana mestinya dan dapat mengatasi berbagai skenario penggunaan
Keywords: Sistem Informasi Akademik; Blackbox; Boundary Value Analysis

DOI: https://doi.org/10.47134/jacis.v4i2.78
*Correspondensi: Ahmad Jailani
Email: 230605210020@student.uin-malang.ac.id

Receive: 29 Mei 2024
Accepted: 29 Juni 2024
Published: 01 Juli 2024

Copyright: © 2021 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

Abstrak: Software testing is a crucial aspect in research to ensure the quality and development of applications. In the context of academic information system testing, the importance of a good design allows for quick error detection, saves time in the testing process, and builds user trust in the application. Relevant research identifies issues in the application, such as input information errors, which trigger testing strategies focused on ensuring the correctness and accuracy of information. Testing methods like Black Box Testing with Boundary Value Analysis are used to evaluate the quality of the application and identify areas for improvement. The conclusion of this research shows that the application was successful in only about 60% of the trials, highlighting the importance of improving application quality. The focus should be on ensuring that the application functions properly and can handle various usage scenarios
Keywords: Academic Information System; Blackbox; Boundary Value Analysis

PENDAHULUAN

Pengujian adalah aspek penting dalam penelitian untuk memastikan kualitas perangkat lunak dan kemajuannya. Proses ini mencakup tahapan seperti analisis, perancangan, dan implementasi kode yang bertujuan untuk memverifikasi bahwa perangkat lunak sesuai dengan persyaratan yang telah ditetapkan.[1]. Dalam pengujian, memiliki desain yang baik sangat penting agar kesalahan dapat terdeteksi dengan mudah selama proses, sehingga

memungkinkan perbaikan dilakukan dengan cepat[2]. Ini bisa menghemat waktu saat melakukan pengujian[3]. Pengujian yang cermat sangat penting untuk mengurangi risiko kegagalan atau ketidaksesuaian aplikasi dengan kebutuhan pengguna[4]. Dengan mengidentifikasi dan memperbaiki potensi masalah sebelum implementasi, pengujian memastikan aplikasi berfungsi sebagaimana mestinya, sehingga mencegah kerugian atau ketidaknyamanan akibat kinerja yang buruk[5]. Pengujian juga membantu membangun kepercayaan pengguna terhadap aplikasi, memastikan bahwa sistem informasi akademik berjalan dengan baik dan sesuai dengan tujuan yang telah ditetapkan[6].

Penelitian[7] mengungkapkan bahwa masalah yang ditemukan pada framework ini adalah kesalahan dalam memasukkan nilai. Meskipun framework ini dapat menghitung dan menampilkan nilai di atas 100 atau di bawah 0, namun tidak dapat mengoreksi informasi yang telah dimasukkan ke dalam dataset. Strategi yang direkomendasikan untuk menguji karakteristik aplikasi ini adalah dengan menggunakan metode "black box" dan prosedur pengujian nilai batas. Pendekatan ini melibatkan pengujian nilai batas untuk menetapkan nilai terendah dan tertinggi dari informasi yang dievaluasi. Sedangkan peneliti[8] menemukan kesalahan dalam proses input informasi selama masa percobaan, yang menyebabkan ketidaksesuaian antara informasi normal dan set data. Untuk menghadapi masalah ini, fokus penelitian adalah memastikan bahwa informasi dalam aplikasi sesuai dengan jenis informasi yang dimaksudkan dan direpresentasikan seakurat mungkin. Selain itu penelitian[9] menekankan pentingnya pengujian perangkat lunak untuk memastikan aplikasi reservasi kamar hotel berfungsi dengan baik dan untuk menghindari bug. Pengujian perangkat lunak berdasarkan kualitasnya sangat penting. Beberapa penelitian menunjukkan bahwa hasil pengujian aplikasi hanya berhasil sekitar 70% hingga 100% dari waktu. Fokus penelitian ini adalah untuk menguji apakah kualitas sistem berfungsi sesuai yang diharapkan.

Pada penelitian ini metode yang digunakan adalah metode Blackbox Testing dengan teknik Boundary Value Analysis. Metode ini fokus pada pengujian nilai-nilai yang mendekati batas valid dan tidak valid, dengan tujuan menentukan jumlah maksimum dan minimum digit yang akan diuji[10]. Prinsip dasar dari Boundary Value Analisys meliputi identifikasi kesalahan input yang mungkin terjadi, pemilihan kasus uji yang memeriksa batas nilai input, serta melengkapi partisi ekuivalen dengan memilih elemen dalam kelas ekuivalen pada batas-batas tersebut[11]. Teknik ini sangat efektif untuk mengidentifikasi cacat pada nilai ekstrem, yaitu pada nilai yang mendekati atau tepat di batas maksimum dan minimum dari rentang input, sehingga memastikan aplikasi berfungsi dengan baik dalam kondisi ekstrem.

Langkah-langkah dalam melakukan pengujian ini melibatkan penetapan nilai batas atas dan bawah untuk setiap kolom melalui prosedur yang telah ditentukan, serta merancang kasus uji yang akan digunakan dalam pengujian[12]. Pengujian ini mencakup persiapan skenario uji untuk kondisi batas, pengujian respons aplikasi, dan evaluasi hasil untuk memastikan aplikasi bekerja dengan baik dalam berbagai kondisi input. Hal ini memungkinkan pengujian yang menyeluruh dan dapat diandalkan, membantu mendeteksi dan memperbaiki potensi masalah sejak tahap awal pengembangan

METODE

Metode penelitian yang digunakan dalam studi ini adalah Blackbox Testing dengan menggunakan teknik Boundary Value Analysis.

Blackbox Testing

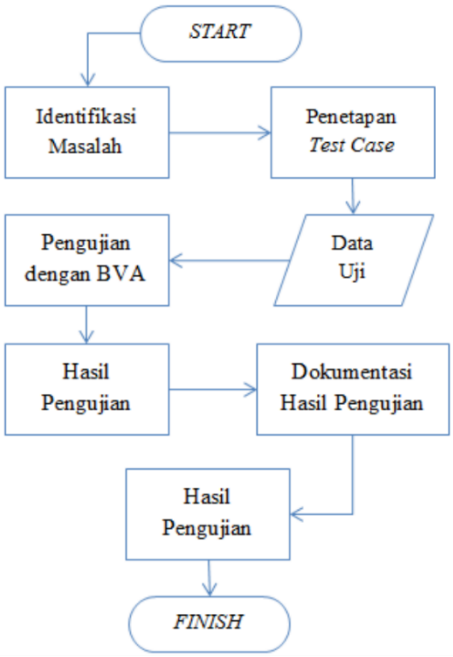
Blackbox Testing adalah metode yang memverifikasi hasil eksekusi aplikasi berdasarkan data masukan untuk memastikan bahwa fungsionalitas aplikasi sesuai dengan persyaratannya[13]. Pengujian ini berfokus pada antarmuka aplikasi, fungsionalitas yang ada, serta kesesuaian alur kerja dengan kebutuhan pengguna. Metode ini tidak melibatkan pemeriksaan kode sumber program. Tahapan dalam blackbox testing meliputi:

a. Membuat test case untuk menguji berbagai fungsi dalam aplikasi.
b. Mengembangkan test case untuk mengevaluasi sejauh mana alur kerja suatu fungsi sesuai dengan kebutuhan dan permintaan pengguna.
c. Mengidentifikasi bug atau kesalahan berdasarkan antarmuka aplikasi.

Proses pengujian harus menggunakan teknik yang sesuai untuk dapat mendeteksi kesalahan yang belum terdeteksi sebelumnya guna meningkatkan kualitas perangkat lunak.

Boundary Value Analysis

Metode Blackbox Testing dengan teknik Boundary Value Analysis melibatkan beberapa tahapan dalam alur pengujian yang dirancang untuk memastikan aplikasi diuji secara menyeluruh dan akurat[10]. Proses ini dimulai dengan mengidentifikasi masalah yang akan diuji untuk menentukan area aplikasi yang memerlukan perhatian khusus dan batas nilai yang akan diuji. Setelah masalah diidentifikasi, langkah berikutnya adalah mendefinisikan test case, di mana setiap elemen yang akan diuji didefinisikan secara rinci untuk memastikan cakupan pengujian yang lengkap. Selanjutnya, elemen-elemen test case yang telah didefinisikan dimasukkan ke dalam aplikasi, dan pengujian dilakukan menggunakan teknik Boundary Value Analysis untuk memastikan aplikasi berfungsi dengan benar di sekitar batas nilai yang ditentukan. Tahap terakhir melibatkan mendokumentasikan hasil pengujian dan menarik kesimpulan, di mana dokumentasi ini mencatat temuan selama pengujian dan memberikan dasar bagi rekomendasi perbaikan. Kesimpulan yang ditarik membantu mengarahkan langkah-langkah berikutnya dalam pengembangan aplikasi. Alur penelitian ini diilustrasikan dalam Gambar 1.



Gambar 1. Alur Pengujian Black Box dengan Teknik Boundary Value Analysis

Proses pengujian penting sebagai panduan agar tester dapat melakukan pengujian secara terstruktur dari awal sampai akhir[14]. Berdasarkan Gambar 1, urutan pengujian yang telah ditetapkan dapat dijelaskan sebagai berikut:

a. Langkah awal dalam proses pengujian melibatkan pemilihan metode dan teknik yang akan digunakan. Dalam konteks ini, pengujian menggunakan metode Black Box Testing dengan teknik Boundary Value Analysis. Pemilihan metode ini dilakukan untuk memastikan bahwa pengujian dapat mendeteksi masalah pada batas nilai input yang mungkin tidak terdeteksi oleh metode pengujian lainnya.
b. Masalah utama yang diidentifikasi untuk diuji adalah validitas data dalam aplikasi pencatatan peminjaman buku. Khususnya, fokus pengujian adalah pada situasi di mana data masukan tidak konsisten dengan data yang tersimpan dalam database. Identifikasi masalah ini penting untuk memastikan integritas dan keakuratan data dalam sistem.
c. Penetapan test case merupakan langkah berikutnya, yang mencakup pemilihan data uji untuk beberapa elemen kunci dalam aplikasi. Elemen-elemen ini termasuk halaman login, halaman pendaftaran peminjaman buku, dan halaman formulir peminjaman buku. Dengan menetapkan test case ini, pengujian dapat difokuskan pada area yang paling kritis dalam aplikasi.
d. Data uji terdiri dari berbagai skenario yang telah dirancang untuk dievaluasi menggunakan metode Black Box Testing dengan teknik Boundary Value Analysis. Setiap skenario dirancang untuk menguji batas nilai input pada test case yang telah ditetapkan sebelumnya. Hal ini memungkinkan pengujian yang komprehensif dan memastikan bahwa aplikasi berfungsi dengan baik di berbagai kondisi.
e. Hasil pengujian akan memberikan gambaran tentang kecocokan dan kinerja aplikasi. Dokumentasi hasil ini penting untuk mencatat apakah aplikasi berfungsi sesuai dengan kriteria yang telah ditentukan dan untuk memberikan rekomendasi perbaikan jika diperlukan. Dengan demikian, hasil pengujian ini akan menjadi dasar untuk memastikan aplikasi siap digunakan dan memenuhi kebutuhan pengguna.

Langkah selanjutnya adalah penetapan test case, yang melibatkan pemilihan data uji untuk beberapa elemen kunci dalam aplikasi. Elemen-elemen ini termasuk halaman dalam aplikasi, dan pengujian dilakukan menggunakan teknik Boundary Value Analysis untuk memastikan aplikasi berfungsi dengan benar di sekitar batas nilai yang ditentukan. Seperti yang di jelaskan pada tabel 1.

Tabel 1. Rencana test case form input nilai

ID	Deskripsi Pengujian	Hasil Yang di harapkan
TC1	Input Valid di Batas Bawah	Nilai harus di terima sebagai input yang valid
TC2	Input Valid di Batas Atas	Nilai harus di terima sebagai input nilai yang valid
TC3	Input di Luar Batas Bawah (Invalid)	Nilai harus di tolak sebagai input nilai yang tidak valid
TC4	Input di Luar Batas Atas (Invalid)	Nilai harus di tolak sebagai input nilai yang tidak valid
TC5	Input di Atas Batas Bawah	Nilai harus di terima sebagai nilai yang valid
TC6	Input di Bawah Batas Atas	Nilai harus di terima sebagai nilai yang valid

Tabel rancangan test case pengujian black box adalah tabel yang memuat daftar test case yang dirancang untuk menguji fungsi atau fitur perangkat lunak. Deskripsi tabel ini mencakup beberapa kolom, seperti:

a. ID: Kode atau identifikasi unik untuk setiap test case.

b. Deskripsi Pengujian: Deskripsi singkat atau judul yang menjelaskan apa yang diuji oleh test case tersebut.
c. Hasil yang diharapkan: Hasil atau respons yang diharapkan dari perangkat lunak setelah menjalankan aksi yang telah ditentukan.

Tabel 1 digunakan sebagai panduan untuk mengatur dan melacak berbagai skenario pengujian yang telah dirancang, guna memastikan bahwa perangkat lunak berfungsi dengan benar sesuai dengan spesifikasi fungsional yang ditetapkan

HASIL DAN PEMBAHASAN

Dalam proses pengujian ini, salah satu fungsi yang dipilih dari Sistem Informasi Akademik adalah "input nilai," yang dianggap mampu mewakili fungsi-fungsi lain dalam aplikasi tersebut. Hasil dari pengujian kemudian didokumentasikan dan disajikan dalam Tabel 2. Tabel ini berisi hasil dari test case yang telah dilakukan, termasuk deskripsi pengujian, hasil yang diharapkan, dan hasil aktual dari pengujian tersebut. Tabel test case black box digunakan sebagai instrumen untuk merencanakan, melacak, dan mengevaluasi test case yang dirancang untuk menguji fungsionalitas perangkat lunak. Setiap test case memiliki ID unik, deskripsi pengujian, hasil yang diharapkan, dan status hasil pengujian (Pass/Fail).

Tabel 2. Hasil Test Case Form Nilai

ID	Deskripsi Pengujian	Hasil Yang di harapkan	Hasil
TC1	Input Valid di Batas Bawah	Nilai harus di terima sebagai input yang valid	Pass
TC2	Input Valid di Batas Atas	Nilai harus di terima sebagai input nilai yang valid	Pass
TC3	Input di Luar Batas Bawah (Invalid)	Nilai harus di tolak sebagai input nilai yang tidak valid	Fail
TC4	Input di Luar Batas Atas (Invalid)	Nilai harus di tolak sebagai input nilai yang tidak valid	Fail
TC5	Input di Atas Batas Bawah	Nilai harus di terima sebagai nilai yang valid	Pass
TC6	Input di Bawah Batas Atas	Nilai harus di terima sebagai nilai yang valid	Pass

Tabel 2 merupakan hasil dari test case form nilai yang dilakukan dalam pengujian aplikasi Sistem Informasi Akademik. Tabel ini berisi informasi yang penting untuk evaluasi hasil pengujian, yaitu:

a. ID Deskripsi Pengujian: Setiap test case memiliki ID unik yang digunakan untuk mengidentifikasi test case secara spesifik. Deskripsi pengujian menjelaskan secara rinci apa yang diuji oleh test case tersebut dan bagaimana proses pengujian dilakukan.
b. Hasil Yang diharapkan: Bagian ini mencantumkan hasil atau respons yang diharapkan dari perangkat lunak setelah menjalankan test case. Ini membantu dalam menetapkan standar atau kriteria keberhasilan untuk setiap test case.
c. Hasil: Bagian ini mencantumkan hasil aktual dari pengujian yang dilakukan. Hasil ini dapat berupa "Pass" jika test case berhasil atau "Fail" jika test case gagal. Status hasil

pengujian ini menunjukkan kesesuaian output yang dihasilkan dengan yang diharapkan.

Dengan informasi yang terdapat dalam Tabel 2, para peneliti dan pengembang perangkat lunak dapat melacak dan mengevaluasi hasil pengujian dengan jelas. Tabel ini membantu dalam memahami kinerja aplikasi, mengidentifikasi area yang memerlukan perbaikan, dan memberikan dasar untuk rekomendasi perbaikan guna meningkatkan kualitas dan kehandalan aplikasi Sistem Informasi Akademik

SIMPULAN

Dari hasil pengujian yang telah dilakukan, simpulannya adalah bahwa kita fokus pada input dan output aplikasi untuk menilai sejauh mana aplikasi tersebut memenuhi harapan. Hasilnya menunjukkan bahwa aplikasi berhasil hanya sekitar 60% dari jumlah percobaan. Meskipun aplikasi dapat melakukan perhitungan bahkan jika ada bagian yang kosong, namun kualitasnya masih perlu ditingkatkan agar dapat menghitung nilai dari 0 hingga 100 dengan lebih baik

DAFTAR PUSTAKA

[1] D. B. Muslimin, D. Kusmanto, K. F. Amilia, M. S. Ariffin, S. Mardiana, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 19, 2020, doi: 10.32493/informatika.v5i1.3778.

[2] S. Putri, Farsya Asfian Saputra, Ferdy Dwiansyah, and Ionia Veritawati, "Pengujian Sistem Informasi Akademik (NeoSiak) Berbasis Website Menggunakan Equivalence Partitioning dan Metode Black Box," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 3, no. 1, pp. 18–26, 2024, doi: 10.55123/storage.v3i1.3133.

[3] H. Hendri, J. W. Hasiholan Manurung, R. A. Ferian, W. F. Hanaatmoko, and Y. Yulianti, "Pengujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 2, p. 107, 2020, doi: 10.32493/jtsi.v3i2.4694.

[4] A. M. Astripat, M. Faisal, and A. Saifudin, "Pengujian Software Sistem Informasi Akademik dengan Black Box," *JRIIN J. Ris. Inform. dan Inov.*, vol. 1, no. 4, pp. 1–9, 2023, [Online]. Available: https://jurnalmahasiswa.com/index.php/jriin

[5] Y. Yulianti, T. Desyani, R. R. Chaniago, H. Iswanto, E. Suroso, and T. S. Hermanto, "Pengujian Aplikasi Sistem Informasi Akademik Berbasis Website Menggunakan Teknik Equivalence Partitioning dan Metode Black Box," *J. Inform. Univ. Pamulang*, vol. 7, no. 1, pp. 145–150, 2022.

[6] A. Fahrezi, F. N. Salam, G. M. Ibrahim, R. R. Syaiful, and A. Saifudin, "Pengujian Black Box Testing pada Aplikasi Inventori Barang Berbasis Web di PT. AINO Indonesia," *J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 1, pp. 1–5, 2022, [Online]. Available: https://journal.mediapublikasi.id/index.php/logic

[7] A. Firmansyah, M. A. Arief, M. Daffa, F. Falah, O. D. Dharmawan, and J. Riyanto, "Pengujian Aplikasi Sistem Penilaian Mahasiswa Dengan Menggunakan Teknik Boundary Value Analysis," *Sci. Sacra J. Sains*, vol. 2, no. 1, pp. 175–179, 2022, [Online].

Available: http://pijarpemikiran.com/index.php/Scientia

[8] E. Nirmala, A. Saifudin, D. Tri Ovandi, R. Nursafitri, and Z. Andris, "Pengujian Aplikasi Penyewaan Lapangan Futsal Berbasis Dekstop dengan Metode Black Box Testing Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 7, no. 1, pp. 2622–4615, 2022, [Online]. Available: http://openjournal.unpam.ac.id/index.php/informatika

[9] A. Yani, D. Setiawan, N. E. Sofian, R. Subagja, and T. Desyani, "Pengujian Aplikasi Reservasi Hotel di LeGreen Hotel & Suite dengan Metode Black Box Testing Boundary Value Analysis," *J. Teknol. Sist. Inf. dan Apl.*, vol. 3, no. 2, p. 114, 2020, doi: 10.32493/jtsi.v3i2.4686.

[10] 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," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 129, 2020, doi: 10.32493/informatika.v5i2.5366.

[11] B. A. Maulana, E. Mawarni, M. Y. Hidayattuloh, V. Suryawijaya, and A. Saifudin, "Pengujian Black Box pada Sistem Informasi Barang Berbasis Web Menggunakan Metode Boundary Value Analysis," *OKTAL J. Ilmu Komput. dan Sci.* , vol. 2, no. 6, pp. 1747–1753, 2023.

[12] D. Debiyanti, S. Sutrisna, B. Budrio, A. K. Kamal, and Y. Yulianti, "Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis," *J. Inform. Univ. Pamulang*, vol. 5, no. 2, p. 162, 2020, doi: 10.32493/informatika.v5i2.5446.

[13] M. Mintarsih, "Pengujian Black Box Dengan Teknik Transition Pada Sistem Informasi Perpustakaan Berbasis Web Dengan Metode Waterfall Pada SMC Foundation," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 1, pp. 33–35, 2023, doi: 10.47233/jteksis.v5i1.727.

[14] E. H. Kusuma Dewi, I. S. Pratama, A. S. Putera, and C. Carudin, "Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analysis," *STRING (Satuan Tulisan Ris. dan Inov. Teknol.*, vol. 6, no. 3, p. 315, 2022, doi: 10.30998/string.v6i3.11958.