

PENGUJIAN KUALITAS PERANGKAT LUNAK SISTEM INFORMASI INVENTORY PADA KOPERASI DOSEN DAN KARYAWAN UNIVERSITAS DR. SOETOMO

Cempaka Ananggadipa Swastyastu¹, Ratna Nur Tiara Shanty², Rika Puspita Sari³, Anggit Wikanningrum⁴
^{1,2,3,4} Universitas Dr. Soetomo

¹cempaka@unitomo.ac.id, ²ratnanurtiara@unitoo.ac.id, ³anggit.wikanningrum@unitomo.ac.id

Naskah diterima: 15 Pebruari 2025 ; Direvisi : 26 Mei 2025 ; Disetujui : 28 Mei 2025

Abstrak (Indonesia)

Pengujian perangkat lunak merupakan tahap esensial dalam software development life cycle (SDLC) untuk memastikan sistem berfungsi sesuai kebutuhan dan tidak menghambat operasional organisasi. Sistem Informasi Inventory yang telah dikembangkan pada Koperasi Dosen dan Karyawan Universitas Dr. Soetomo untuk mendukung operasional koperasi. Namun belum dilakukan pengujian kualitas perangkat lunak secara menyeluruh. Pengujian diperlukan untuk meminimalkan risiko kegagalan operasional yang dapat mengganggu proses bisnis. Penelitian ini bertujuan mengevaluasi kualitas sistem melalui pendekatan black box testing dengan menggabungkan teknik Equivalence Partitioning (EP) dan Boundary Value Analysis (BVA). Metode ini diimplementasikan pada use case seperti input stok barang dan transaksi penjualan. Hasil pengujian menunjukkan bahwa uji antarmuka (interface) mencapai efektivitas 93,33% (kategori sangat efektif), sedangkan uji fungsionalitas memperoleh 85,73% (cukup efektif). Temuan ini mengidentifikasi celah pada fungsionalitas yang memerlukan perbaikan, seperti validasi input dan konsistensi data. Penelitian ini memberikan rekomendasi kepada pengembang untuk meningkatkan kualitas sistem, sekaligus menjadi referensi bagi pengujian sistem inventory berbasis metode EP dan BVA. Hasil studi tidak hanya mendukung peningkatan sistem di koperasi tersebut, tetapi juga berkontribusi pada pengembangan sistem serupa di lingkup lebih luas melalui pendekatan pengujian yang efisien dan terstruktur.

Kata kunci: Pengujian Perangkat Lunak, Black Box Testing, Equivalence Partitioning, Boundary Value Analysis, Sistem Informasi Inventory

Abstract (English Version)

Software testing is an essential stage in the Software Development Life Cycle (SDLC) to ensure that the system functions according to requirements and does not hinder organizational operations. The Inventory Information System developed for the Cooperative of Lecturers and Employees at Dr. Soetomo University aims to support the cooperative's operations. However, comprehensive software quality testing has not yet been conducted. Testing is necessary to minimize the risk of operational failures that could disrupt business processes. This study aims to evaluate the system's quality through a black box testing approach, combining Equivalence Partitioning (EP) and Boundary Value Analysis (BVA) techniques. The method is implemented on use cases such as stock input and sales transactions. The test results show that the interface testing achieved an effectiveness of 93.33% (highly effective category), while functionality testing obtained 85.73% (moderately effective). These findings identify gaps in functionality that require improvements, such as input validation and data consistency. This study provides recommendations for developers to enhance system quality and serves as a reference for inventory system testing based on the EP and BVA methods. The results not only support system improvement in the cooperative but also contribute to the development of similar systems in a broader context through efficient and structured testing approaches.

Keywords: Software Testing, Black Box Testing, Equivalence Partitioning, Boundary Value Analysis, Inventory Information System

PENDAHULUAN

Sistem informasi inventory memiliki peran yang penting dalam pengelolaan barang dan pencatatan transaksi di berbagai organisasi, termasuk koperasi. Koperasi Dosen dan Karyawan Universitas Dr. Soetomo telah mengembangkan Sistem Informasi Inventory untuk mendukung operasional koperasi, khususnya dalam pengelolaan stok barang dan transaksi penjualan [1][2]. Namun, meskipun sistem informasi ini telah selesai dikembangkan, belum dilakukan pengujian kualitas perangkat lunak secara menyeluruh. Pengujian perangkat lunak merupakan tahap esensial dalam siklus hidup pengembangan perangkat lunak untuk memastikan bahwa sistem yang dihasilkan memenuhi kebutuhan pengguna dan dapat beroperasi dengan baik. Tanpa pengujian yang memadai, sistem informasi berisiko mengalami kegagalan operasional yang dapat berdampak negatif dalam mendukung proses bisnisnya [3] [4]. Oleh karena itu, diperlukan pengujian yang sistematis untuk mengidentifikasi potensi kesalahan dan memastikan bahwa sistem dapat berjalan dengan optimal[5].

Salah satu cara untuk menguji sistem adalah dengan menggunakan metode black box testing, yaitu menguji fungsionalitas sistem tanpa melihat kode programnya. Dalam penelitian ini, digunakan dua teknik utama dalam pengujian black box, yaitu Equivalence Partitioning (EP) dan Boundary Value Analysis

(BVA). Kombinasi kedua metode tersebut dapat meningkatkan efisiensi dan efektivitas pengujian perangkat lunak[6] [7]. EP membagi input data menjadi partisi-partisi yang setara, memungkinkan pengurangan jumlah kasus uji tanpa mengurangi cakupan pengujian. Sementara itu, BVA berfokus pada pengujian nilai batas dari partisi-partisi tersebut, karena kesalahan sering terjadi pada nilai-nilai ekstrem.

Hasil dari penelitian ini diharapkan dapat menjadi catatan bagi pengembang sistem agar dapat meningkatkan kualitas perangkat lunak yang digunakan di koperasi. Selain itu, penelitian ini juga dapat menjadi referensi bagi pengembang lain yang ingin menerapkan teknik serupa dalam pengujian sistem berbasis inventaris. Dengan demikian tidak hanya berkontribusi terhadap peningkatan kualitas sistem informasi di koperasi, namun juga memberikan wawasan bagi pengembangan sistem informasi inventory secara lebih luas.

METODE

Proses pelaksanaan penelitian ini mengikuti beberapa tahapan yang telah disesuaikan dengan metode Equivalence Partitioning dan Boundary Value Analysis[8][9]. Metode ini merupakan metode pengujian, dimana tahap pengujian itu sendiri merupakan salah satu tahapan pada siklus pengembangan perangkat lunak dimana berisi rangkaian kegiatan yang dilaksanakan untuk menemukan kesalahan, ketidaksesuaian ataupun kekurangan dari

sebuah sistem atau perangkat lunak[10] [11].

Metode ini berfungsi untuk memeriksa jenis dan kondisi input yang kemudian akan dapat dibagi menjadi beberapa kelas ekuivalensi yaitu yang bersifat valid dan tidak valid. Pembagian sistem menjadi sejumlah kelas ini memiliki fungsi yaitu jika pada satu kasus uji dalam salah satu kelas menghasilkan kesalahan, maka kelas yang lain juga akan menghasilkan kesalahan[12]. Artinya, asumsi ini dirancang untuk satu kelas mewakili nilai input pada seluruh kelas ekuivalensi.

1. Equivalence Partitioning

Teknik Equivalence Partitioning (EP) membagi domain input dari suatu program ke dalam partisi-partisi yang dianggap ekuivalen, di mana setiap partisi diharapkan menghasilkan perilaku sistem yang serupa[13]. Pada setiap kasus uji, jika satu kasus uji di kelas ekivalensi pertama mendeteksi kesalahan, semua kasus uji lainnya di kelas ekivalensi kedua diharapkan menemukan kesalahan yang sama[14]. Sebaliknya, jika kasus uji tidak mendeteksi kesalahan, maka seharusnya tidak ada kasus uji lain dalam kelas ekivalensi pertama yang akan masuk dalam kelas ekivalensi yang lain, karena kelas ekivalensi mungkin saling tumpang tindih. Kedua kelas ini membentuk pertimbangan dimana, pertimbangan pertama digunakan untuk mengembangkan serangkaian kasus uji minimal yang mencakup kondisi ini dan pertimbangan kedua digunakan untuk mengembangkan serangkaian kondisi yang menarik untuk diuji [15] [16]. Pengujian dilakukan dengan cara mendesain

beberapa kasus uji yang disesuaikan pada fungsionalitas sistem. Penetapan batas pengujian, jenis pengujian, dan hasil yang diinginkan untuk diuji berdasarkan jenis yang direncanakan dimaksudkan untuk memperoleh data seperti dokumentasi metode Equivalence Partitioning dan untuk mengetahui keefektifan pengujian dengan Metode Equivalence Partitioning.

2. Boundary Value Analysis

Metode Boundary Value Analysis (BVA) melakukan pengujian pada nilai atau data input aplikasi di batas atas dan batas bawah. Metode ini memiliki pencakupan yang efektif karena menguji semua instruksi atau menu yang ada [17][18]. Metode ini dirancang untuk melengkapi teknik Equivalence Partitioning, yang hanya fokus pada nilai input atau data uji, serta juga memperhatikan nilai keluaran, sehingga dapat menguji apakah data uji yang melebihi batas yang telah ditetapkan akan diolah oleh database atau sistem, memberikan peringatan jika melebihi, dan hanya memproses data uji yang berada di bawah batas atas tersebut [19] [20].

HASIL DAN PEMBAHASAN

Metode pelaksanaan uji coba yang pertama adalah menggunakan metode Equivalence Partitioning, dengan tahapan menentukan use case yang diuji, menentukan kriteria pengujian, mendefinisikan partisi, membuat data uji, menyusun kasus uji, melakukan pengujian, evaluasi hasil pengujian. Pada pengujian ini, di

dalamnya juga akan terdapat beberapa pengujian dengan menggunakan teknik boundary value analysis, terutama untuk use case yang berhubungan dengan input nilai yang punya batas atas dan batas bawah, atau suatu kekhususan di dalamnya.

1. Menentukan Use Case yang Diuji.

Pada Tahap pertama melibatkan identifikasi use case atau skenario penggunaan yang akan diuji. Use case ini menggambarkan interaksi pengguna dengan sistem dan hasil yang diharapkan dalam situasi tertentu, mencakup input, output, dan respon dari sistem. Daftar *use case interface testing* ditunjukkan oleh Tabel 1.

2. Menentukan Kriteria Pengujian.

Setelah menentukan use case, langkah selanjutnya adalah menentukan kriteria pengujian yang digunakan untuk menilai keberhasilan atau kegagalan pengujian. Kriteria ini biasanya berdasarkan spesifikasi fungsional yang telah ditetapkan. Daftar kriteria functional testing ditunjukkan oleh Tabel 2 dan Tabel 3.

3. Mendefinisikan Partisi.

Domain input sistem dibagi menjadi beberapa partisi atau kelas ekuivalen. Setiap partisi berisi sekumpulan input yang diharapkan menghasilkan perilaku sistem yang serupa. Misalnya, untuk input berupa angka, partisi dapat dibagi berdasarkan rentang nilai tertentu. Daftar definisi partisi dapat ditunjukkan pada Tabel 4.

4. Membuat Data Uji.

Data uji disiapkan pada Tabel 5 dan Tabel 6, untuk setiap partisi yang telah didefinisikan. Data ini dipilih untuk mewakili kondisi normal, kondisi batas, dan kondisi yang tidak valid dalam setiap partisi, mencakup berbagai kemungkinan input untuk memastikan pengujian yang menyeluruh.

5. Menyusun Kasus Uji.

Berdasarkan data uji yang telah disiapkan, kasus uji disusun pada Tabel 7. Kasus uji ini mencakup langkah-langkah pengujian, input yang harus dimasukkan, dan hasil yang diharapkan. Dokumentasi yang jelas dari setiap kasus uji memudahkan pelaksanaan dan evaluasi pengujian.

6. Melakukan Pengujian.

Kasus uji dieksekusi menggunakan data uji yang telah disiapkan dan ditunjukkan oleh Tabel 8. Selama pengujian, hasil aktual sistem dicatat dan dibandingkan dengan hasil yang diharapkan. Pengujian dianggap berhasil jika hasil aktual sesuai dengan yang diharapkan. Jika tidak, maka ditemukan bug atau kesalahan yang perlu diperbaiki.

7. Evaluasi Hasil Pengujian.

Setelah pengujian selesai, hasilnya dievaluasi untuk mendeteksi kesalahan dalam setiap partisi. Evaluasi ini mencakup analisis kesalahan yang ditemukan, dokumentasi kesalahan tersebut, dan rekomendasi perbaikan. Evaluasi ini penting untuk memberikan umpan balik

kepada tim pengembang dan memastikan perbaikan sistem sesuai temuan pengujian. Berdasarkan standar acuan dari Litbang Depdagri 1991, nilai efektivitas suatu sistem dapat diukur seperti pada Tabel 9 yang menunjukkan standar ukuran efektivitas dalam bentuk range nilai dan tingkat capaiannya. Kemudian sistem akan dinilai berdasarkan nilai efektivitasnya, apakah termasuk sangat tidak efektif, tidak efektif, cukup efektif, atau sangat efektif.

Adapun menurut [10], rumus untuk menghitung nilai efektivitas sistem ditunjukkan oleh formula (1) dan (2).

Nilai Efektivitas Tabel =

$$\left(\frac{\Sigma \text{ Pengujian Sesuai}}{\Sigma \text{ Jumlah Skenario Pengujian}} \right) \times 100 \quad (1)$$

Rumus perhitungan efektivitas dari setiap tabel pengujian adalah dengan menghitung hasil pengujian yang berhasil kemudian membaginya dengan total skenario pengujian dalam tabel tersebut, lalu hasilnya dikalikan 100.

$$\text{Nilai Efektivitas Sistem} = \left(\frac{\text{Nilai Tabel A} + \text{Nilai Tabel B} + \dots + \text{Nilai Tabel n}}{\Sigma \text{ Jumlah Tabel yang diuji}} \right) \quad (2)$$

Setelah mengetahui nilai efektivitas dari setiap tabel, selanjutnya menghitung nilai akhir efektivitas sistem. Penghitungan dilakukan dengan mengetahui keseluruhan nilai dari setiap tabel pengujian yang kemudian semua nilai ditambahkan lalu dibagi dengan jumlah seluruh tabel yang diuji.

Hasil Uji Interface Testing:

Jumlah seluruh kasus Uji =	15
As Expected =	14
Unexpected =	1
Persentase =	93.33

Hasil Uji Functional Testing:

Jumlah seluruh kasus uji =	32
As Expected =	25
Unexpected =	7
Persentase =	78.13

Nilai Efektivitas Sistem :

$$\frac{(93.3 + 78.13)}{2} = 85.73 \quad (3)$$

Dari hasil tersebut dapat dinyatakan bahwa pengujian yang dilaksanakan untuk uji interface adalah sangat efektif, dan untuk uji fungsionalitas adalah cukup efektif.

Tabel 1. Daftar Use Case Interface Testing

Use Case	Deskripsi
Login	Fitur untuk autentikasi pengguna agar dapat mengakses sistem berdasarkan username dan password.
Barang	Fitur untuk menambah, mengedit, menghapus, dan melihat daftar barang dalam sistem inventory.
Kasir	Modul untuk melakukan transaksi penjualan secara langsung kepada customer (anggota koperasi).
Pembelian	Fitur untuk mencatat dan mengelola data pembelian barang dari supplier.
Penjualan	Modul untuk mencatat dan mengelola data penjualan barang kepada pelanggan.
Gudang	Fitur untuk mengelola data gudang penyimpanan barang
Supplier	Modul untuk mengelola data supplier, termasuk informasi kontak dan transaksi.

Tabel 2. Daftar Kriteria Functional Testing

Strategi Pengujian	Langkah-langkah
Test Objective	1. Memastikan fitur-fitur dapat berjalan dengan baik. 2. Mengisi form input sehingga data yang dimasukkan dapat masuk dalam database
Technique	1. Melakukan pengujian pada fitur dan komponen aplikasi. 2. Menguji tombol pada fitur dalam aplikasi. 3. Mengisi form input dengan beberapa kondisi.
Completion Criteria	Form input dapat melakukan input data ke dalam database.

Tabel 3. Daftar Kriteria User Interface Testing

Strategi Pengujian	Langkah-langkah
Test Objective	Memastikan komponen aplikasi dapat berjalan sesuai fungsionalitas dengan mudah.
Technique	Testing pada fungsi dan akses pada komponen dan tombol yang ada dalam aplikasi.
Completion Criteria	Pengguna dapat dengan mudah menggunakan aplikasi ini

Tabel 4. Daftar Definisi Partisi

Use Case	Interface Testing	Fungsionalitas Testing
Login	Menampilkan halaman username dan password serta tombol masuk	Uji validitas username dan password dan keberhasilan masuk ke sistem
Barang	Menampilkan daftar barang dalam tabel UI.	Uji keberhasilan mengelola data barang
Kasir	Menampilkan interface transaksi.	Uji keberhasilan kelola transaksi penjualan dan pengaruh terhadap stock barang.
	Menampilkan daftar invoice	
	Menampilkan daftar adjustment stock	Uji keberhasilan penyesuaian stock barang
Pembelian	Menampilkan daftar invoice pembelian barang dari supplier	Uji keberhasilan kelola transaksi pembelian dari supplier dan pengaruh terhadap stock barang.
	Menampilkan daftar pembelian barang per invoice	
Penjualan	Menampilkan daftar penjualan	Uji keberhasilan kelola data penjualan
	Menampilkan daftar penjualan tempo	
	Menampilkan penjualan per tanggal	
	Menampilkan penjualan per item	
Gudang	Menampilkan daftar gudang penyimpanan	Uji keberhasilan kelola data daftar gudang penyimpanan
Supplier	Menampilkan daftar supplier (pemasok) barang	Uji keberhasilan kelola data daftar supplier barang

Tabel 5. Data Uji Interface Testing

Use Case	Scenario ID	Test Scenario
Login	TI-01	Menampilkan field username dan password serta tombol masuk
Identitas	TI-02	Menampilkan setting identitas untuk ditampilkan pada invoice dan nota penjualan
Barang	TI-03	Menampilkan daftar barang dalam tabel UI.
Kasir	TI-04-01	Menampilkan interface transaksi.
	TI-04-02	Menampilkan daftar invoice
	TI-04-03	Menampilkan daftar adjustment stock
Pembelian	TI-05-01	Menampilkan daftar invoice pembelian barang dari supplier
	TI-05-02	Menampilkan daftar pembelian barang per invoice
Penjualan	TI-06-01	Menampilkan daftar penjualan
	TI-06-02	Menampilkan daftar penjualan tempo
	TI-06-03	Menampilkan penjualan per tanggal
	TI-06-04	Menampilkan penjualan per item
Gudang	TI-07	Menampilkan daftar gudang penyimpanan
Supplier	TI-08	Menampilkan daftar supplier (pemasok) barang

Tabel 6. Data Uji Fucntional Testing

Use Case	Fucntional Testing	Scenario ID	Test Scenario	Scenario Type
Login	Uji validitas username dan password dan keberhasilan masuk ke sistem	TF-01-01	Validasi login dengan kombinasi username dan password valid.	Positive
		TF-01-02	Validasi login dengan kombinasi username dan password tidak valid.	Negative
Barang	Uji keberhasilan mengelola data barang	TF-03-01	User menambahkan data secara manual dengan data yang lengkap	Positive
		TF-03-02	User menambahkan data secara manual dengan data yang tidak lengkap	Negative
		TF-03-03	User menambahkan data barang dengan memasukkan harga pada batas bawah (0) dan sedikit di atas batas bawah (1)	Negative
		TF-03-04	User menambahkan barang dengan stok pada batas bawah (0) dan sedikit di atas batas bawah (1).	Negative
Kasir	Uji keberhasilan kelola transaksi penjualan dan pengaruh terhadap stock barang.	TF-04-01	User memilih barang dan memasukkan kuantitas, jumlah bayar, dan klik tombol bayar	Positive

Use Case	Fucntional Testing	Scenario ID	Test Scenario	Scenario Type
Pembelian		TF-04-02	User menghapus item barang yang dibeli	Positive
		TF-04-03	User memilih barang dan memasukkan kuantitas pembelian = 0	Negative
		TF-04-04	User memilih barang dan memasukkan kuantitas di bawah 0 (minus)	Negative
		TF-04-05	User memasukkan kuantitas barang di atas jumlah stock yang tersedia	Negative
	Uji keberhasilan penyesuaian stock barang	TF-05-01	User menyesuaikan stock barang dengan menginput jumlah dan tanggal adjustment lengkap	Positive
		TF-05-02	User menyesuaikan stock barang dengan menginput jumlah tanpa tanggal adjustment	Negative
		TF-05-03	User menyesuaikan stock barang dengan menginput jumlah di bawah 0	Negative
		TF-05-04	User menyesuaikan stock barang dengan menginput jumlah diatas 0	Positive
	Uji keberhasilan kelola transaksi pembelian dari supplier dan pengaruh terhadap stock barang.	TF-06-01	User menambahkan data pembelian dengan memasukkan semua data secara lengkap	Positive
		TF-06-02	User menambahkan data pembelian dengan cara mengosongkan data pokok (nomor invoice, tanggal invoice, barang, harga satuan, kuantitas, tanggal terima)	Negative
		TF-06-03	User menambahkan data pembelian dengan cara menginput jumlah di bawah 0 (minus)	Negative
Penjualan	Uji keberhasilan kelola data penjualan	TF-07-01	User menambahkan data penjualan dengan memasukkan data secara lengkap	Positive
Gudang	Uji keberhasilan kelola data daftar gudang penyimpanan	TF-08-01	User menambahkan data nama gudang dan lokasi dengan lengkap	Positive
		TF-08-02	User menambahkan data gudang dengan mengosongkan lokasi	Negative
		TF-08-03	User memasukkan data gudang dengan mengosongkan nama	Negative

Use Case	Fucntional Testing	Scenario ID	Test Scenario	Scenario Type
Supplier	Uji keberhasilan kelola data daftar supplier barang	TF-09-01	gudang User memasukkan data supplier dengan memasukkan data secara lengkap	Positive
		TF-09-02	User memasukkan data supplier dengan mengosongkan nama supplier	Negative
		TF-09-03	User memasukkan data supplier dengan mengosongkan alamat supplier dan nomor telpon	Negative

Tabel 9. Standar Nilai Efektivitas

Rasio Efektivitas	Tingkat Capaian
0 – 40	Sangat tidak efektif
40 – 55.99	Tidak efektif
60 – 79.99	Cukup efektif
80 – 100	Sangat efektif

PENUTUP

Berdasarkan hasil pengujian dan evaluasi yang telah dilakukan, dapat disimpulkan bahwa

1. Metode Equivalence Partitioning yang dipadukan dengan teknik Boundary Value Analysis terbukti efektif dalam menguji berbagai use case sistem.
2. Proses pengujian yang sistematis meliputi identifikasi use case, penentuan kriteria pengujian, pembuatan partisi dan data uji, penyusunan kasus uji, pelaksanaan pengujian, hingga evaluasi hasil.
3. Evaluasi pengujian dilakukan melalui analisis kesalahan, dokumentasi, dan pemberian rekomendasi perbaikan sebagai umpan balik bagi tim pengembang.

4. Pengujian antarmuka menunjukkan efektivitas yang sangat tinggi (93,33%), sedangkan pengujian fungsionalitas mencapai tingkat efektivitas yang cukup (85,73%).

DAFTAR PUSTAKA

[1] C. A. Swastyastu, R. N. T. Shanty, R. P. Sari, and A. Wikanningrum, "Agile Implementation for Inventory (Case study: Business Unit of Private University)," *J. Inf. Technol. Cyber Secur.*, vol. 2, no. 1, Art. no. 1, Feb. 2024, doi: 10.30996/jitcs.10060.

[2] N. Honest, "Role of Testing in Software Development Life Cycle," *Int. J. Comput. Sci. Eng.*, vol. 7, pp. 886–889, May 2019, doi: 10.26438/ijcse/v7i5.886889.

[3] R. Abdillah, R. Hermawan, W. Hermawansyah, I. Adkha, and H. Arifin, "Pengujian Perangkat Lunak Sistem Informasi Inventori pada Usaha Jasa Pengiriman Paket," *Polyg. J. Ilmu Komput. Dan Ilmu Pengetah. Alam*, vol. 2, no. 4, pp. 166–175, Jul. 2024, doi: 10.62383/polygon.v2i4.199.

- [4] Kusum, P. Talwar, A. Puri, and G. Kumar, "Overview of software testing," *Glob. J. Eng. Technol. Adv.*, vol. 19, no. 1, pp. 104-112, 2024, doi: 10.30574/gjeta.2024.19.1.0060.
- [5] 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 J. Mhs. Tek. Inform.*, vol. 8, no. 1, Art. no. 1, Mar. 2024, doi: 10.36040/jati.v8i1.8921.
- [6] R. Pramudita, "Pengujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning," *Inform. Educ. Prof. J. Inform.*, vol. 4, no. 2, pp. 193-202, Jun. 2020, doi: 10.51211/itbi.v4i2.1347.
- [7] R. Amalia, S. Syaodah, S. Susilawati, and R. G. Suyatna, "Penerapan Digital Marketing Sebagai Strategi Pemasaran UMKM Studi Kasus Batik Tedjo," *J. Manuhara Pus. Penelit. Ilmu Manaj. Dan Bisnis*, vol. 2, no. 1, Art. no. 1, 2024, doi: 10.61132/manuhara.v2i1.568.
- [8] G. J. Myers, C. Sandler, and T. Badgett, *The Art of Software Testing*. John Wiley & Sons, 2011.
- [9] A. Ardilla, F. P. Aditiawan, and Y. V. Via, "PENERAPAN TEKNIK EQUIVALENCE PARTITIONING DAN BOUNDARY VALUE ANALYSIS DALAM PENGUJIAN BLACK BOX APLIKASI KEDIRI SINGLE WINDOW FOR INVESTMENT:," *JATI J. Mhs. Tek. Inform.*, vol. 8, no. 3, Art. no. 3, May 2024, doi: 10.36040/jati.v8i3.9739.
- [10] B. Homès, *Fundamentals of Software Testing*. John Wiley & Sons, 2024.
- [11] A. M. Memon, "A Comprehensive Survey on Software Testing Techniques," *Int. J. Comput. Sci. Issues*, vol. 9, no. 2, pp. 145-150, Mar. 2012.
- [11] A. S. Nugroho, "Analisis Pengujian Perangkat Lunak Berbasis Black Box," *J. Teknol. Inf. dan Komput.*, vol. 9, no. 1, pp. 12-19, 2023.
- [12] M. H. Putra dan D. R. Wahyuni, "Penerapan Boundary Value Analysis pada Sistem Informasi Akademik," *J. Sist. Inf.*, vol. 10, no. 2, pp. 45-52, 2022.
- [13] [1] A. M. Memon, "An Analysis of Equivalence Partitioning and Boundary Value Analysis in Software Testing," *Int. J. Comput. Sci. Issues*, vol. 12, no. 1, pp. 45-50, Jan. 2015..
- [14] [2] A. M. Ammann and J. Offutt, "Using Mutation Faults in Test Requirements to Improve Equivalence Class Partitioning," *Empir. Softw. Eng.*, vol. 10, no. 1, pp. 34-62, Mar. 2005, doi: 10.1007/s10664-004-6287-5.
- [15] [3] M. D. Myers and C. Sandler, *The Art of Software Testing*, 3rd ed., John Wiley & Sons, 2011.
- [16] A. P. Lestari, "Perancangan Kasus Uji Black Box pada Aplikasi Inventori," *J. Teknol. Komput. dan Inf.*, vol. 7, no. 1, pp. 20-26, 2021.
- [17] [1] A. M. Memon, "Boundary Value Analysis: A Black-Box Testing Technique," *Int. J. Comput. Appl.*, vol. 108, no. 15, pp. 1-5, Dec. 2014.
- [18] A. P. Lestari, "Perancangan Kasus Uji Black Box pada Aplikasi Inventori," *J. Teknol. Komput. dan Inf.*, vol. 7, no. 1, pp. 20-26, 2021.
- [19] A. Ardilla, F. P. Aditiawan, and Y. V. Via, "Penerapan Teknik Equivalence Partitioning dan Boundary Value Analysis dalam Pengujian Black Box Aplikasi Kediri Single Window for Investment," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 8, no. 3, May 2024, doi: 10.36040/jati.v8i3.9739
- [20] D. K. Utama, "Evaluasi Sistem Informasi Kepegawaian Menggunakan Black Box Testing," *J. Sist. dan Teknol. Inf.*, vol. 10, no. 4, pp. 88-95, 2023