



Pengujian Aplikasi SalaryMSI Menggunakan Metode Boundary Value Analysis dan Equivalence Partitioning

Anggi Dwi Cahyani¹, Leoni², Nahdah Aqilah Aulia³, Antonius Wahyu Sudrajat⁴

^{1,2,3,4} Program Studi Sistem Informasi, Fakultas Ilmu Komputer dan Rekayasa, Universitas Multi Data Palembang, Palembang, Indonesia

¹anggdwi1909@mhs.mdp.ac.id, ²leonilek421@mhs.mdp.ac.id, ³nahdah2021@mhs.mdp.ac.id, ⁴wahyu.sudjarat@mdp.ac.id

Abstract

Aplikasi SalaryMSI merupakan sistem yang dirancang untuk mengelola dan menghitung gaji karyawan, yang berperan krusial dalam operasional perusahaan. Penelitian ini bertujuan untuk menguji aplikasi SalaryMSI guna mengidentifikasi kesalahan atau kekurangan dengan menggunakan teknik pengujian Blackbox Testing, yaitu Boundary Value Analysis (BVA) dan Equivalence Partitioning (EP). Metode penelitian ini mengikuti alur Software Testing Life Cycle (STLC) yang mencakup analisis kebutuhan, pembuatan test case, pengujian test case, dan evaluasi hasil pengujian. Hasil pengujian menunjukkan bahwa aplikasi SalaryMSI mampu menangani input yang valid dan menolak input yang tidak valid dengan memberikan pesan kesalahan yang tepat. Pengujian dengan metode BVA memperlihatkan bahwa sistem berfungsi dengan baik pada titik-titik ekstrem input, sementara metode EP menunjukkan bahwa sistem mampu menangani variasi input yang berbeda dengan stabil. Hasil dari pengujian ini ialah berdasarkan pengujian menggunakan metode BVA dan EP, sama-sama memiliki kesesuaian 100%. Dimana menunjukkan bahwa sistem berfungsi dengan baik dalam pengujian kombinasi partisi dan nilai batas.

Keywords: Blackbox Testing; Boundary Value Analysis; Equivalence Partitioning; SalaryMSI;

1. Introduction

Kemajuan teknologi dan informasi yang pesat telah merambah ke segala aspek kehidupan, termasuk dunia bisnis. Teknologi informasi berperan penting dalam kehidupan masyarakat, khususnya dalam bidang bisnis[1]. Tak heran jika banyak perusahaan yang berlomba-lomba mengadopsi sistem terkomputerisasi untuk menangani berbagai kebutuhannya, termasuk CV Percetakan Media Sarana Inspirasi.

Dalam pengembangan perangkat lunak, kualitas dan keandalan aplikasi merupakan faktor krusial yang menentukan kesuksesan dan kepuasan pengguna. Salah satu aspek penting dalam menjamin kualitas perangkat lunak adalah pengujian, yang bertujuan untuk mengidentifikasi dan memperbaiki kesalahan atau *bug* sebelum aplikasi dirilis ke pengguna akhir. Pengujian perangkat lunak memerlukan pendekatan yang sistematis dan metodologi yang tepat untuk memastikan cakupan yang luas dan mendeteksi kesalahan secara efektif. Pengujian perangkat lunak adalah proses yang sistematis dan terstruktur yang dilakukan untuk memverifikasi dan validasi bahwa perangkat lunak atau aplikasi memiliki kualitas yang memadai sesuai dengan persyaratan yang ditetapkan, serta untuk mengidentifikasi dan mengatasi kesalahan atau cacat

yang mungkin ada sebelum perangkat lunak tersebut dirilis atau digunakan secara luas[2].

Aplikasi SalaryMSI adalah sebuah sistem yang dirancang untuk mengelola dan menghitung gaji karyawan. Aplikasi ini berperan penting dalam operasional perusahaan karena berkaitan langsung dengan penghitungan kompensasi karyawan yang harus akurat dan dapat diandalkan. Kesalahan dalam aplikasi SalaryMSI bisa berdampak signifikan, baik dari segi keuangan maupun kepuasan karyawan. Oleh karena itu, pengujian aplikasi ini menjadi sangat penting untuk memastikan fungsionalitasnya berjalan sesuai dengan yang diharapkan. Pengujian aplikasi juga berguna untuk menemukan kekurangan dan kesalahan pada aplikasi melalui *study literature* terkait.

Pada penelitian yang berjudul "Pengujian Aplikasi dengan Metode Blackbox Testing: Analisis Boundary Value dan Equivalence Partitioning pada Aplikasi Sistem Pakar Kucing" menggunakan metode Boundary Value Analysis dan Equivalence Partitioning. Dari pengujian dengan metode BVA terdapat 14 *case* yang diuji dan memiliki kesesuaian 100%. Dari pengujian dengan metode Equivalence Partitioning terdapat 8 *case* dan tidak ditemukan kesalahan dalam pengujian tersebut[3]. Pada penelitian yang berjudul "Pengujian Aplikasi *e-farmer* dalam perhitungan keuntungan

dengan metode *blackbox testing boundary value analysis*". Penelitian ini menggunakan metode *boundary value analysis* dimana hasil pengujian sesuai 88,89%. Terdapat tiga bidang yang perlu ditingkatkan agar dapat mengelola data dengan baik[4].

Black Box testing adalah teknik pengujian perangkat lunak yang fokus menguji fungsi aplikasi tanpa perlu memeriksa kinerja atau struktur internalnya. Teknik ini dapat diterapkan pada hampir setiap tingkat pengujian perangkat lunak, termasuk pengujian penerimaan, integrasi, sistem, dan unit, untuk memastikan bahwa aplikasi berperilaku sesuai dengan spesifikasi fungsional yang telah ditetapkan[5].

Pengujian ekuivalensi didasarkan pada analisis nilai batas dan merupakan metode yang banyak digunakan dalam *Black Box Testing* untuk menguji perangkat lunak. Pendekatan ini mendasarkan pengujian pada kasus uji yang mencakup nilai ekstrim dari data uji, memastikan bahwa perilaku program diuji dengan berbagai kondisi yang mencakup segala kemungkinan nilai yang mungkin diterima atau diproses oleh perangkat lunak tersebut[6].

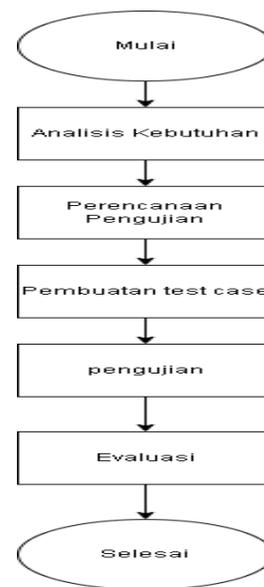
Equivalence Partitions (EP) merupakan teknik pengujian *software* yang fokus pada pembagian data masukan ke dalam kelompok-kelompok berdasarkan fungsinya. Pada setiap kelompok, dilakukan pengujian untuk memastikan bahwa sistem seleksi *sales* terbaik berfungsi dengan baik[7]. Dengan menggunakan teknik *Equivalence Class Partitioning* (ECP), kita hanya perlu menguji satu syarat dari setiap *subset* atau kelas yang dipartisi, yang membantu mengurangi jumlah kasus uji yang diperlukan. Pada dasarnya, teknik ini berfungsi untuk membagi domain input program menurut nilai-nilai yang ekuivalen ke dalam kelas-kelas kesetaraan, yang mewakili berbagai kondisi yang seharusnya diuji untuk memvalidasi fungsi atau perilaku perangkat lunak secara menyeluruh[8].

Penelitian ini menggunakan metode *Bondary Value Analysis* dan *Equivalence Partitioning* untuk pengujian. Peneliti dapat melakukan pengujian menyeluruh dan menyeluruh terhadap aplikasi tersebut dengan menggunakan EP untuk membagi data masukan menjadi partisi dengan nilai valid dan tidak valid, dan BVA untuk menguji nilai-nilai batas dari masukan sistem. Diharapkan bahwa ini akan membantu aplikasi beroperasi dengan kualitas yang tinggi dengan mengidentifikasi dan mengatasi kesalahan.

2. Research Methods

Pengujian ini menggunakan *Life Cycle Software Testing* (STLC) sebagai pedoman untuk mengikuti alur pengujian perangkat lunak. Aliran STLC digambarkan pada *Figure.1. Diagram Software Testing Life Cycle* (STLC) terdiri dari lima langkah utama. Pertama, Analisis Persyaratan dilakukan untuk memahami kebutuhan fungsional dan non-fungsional dari perangkat lunak yang akan dikembangkan. Ini

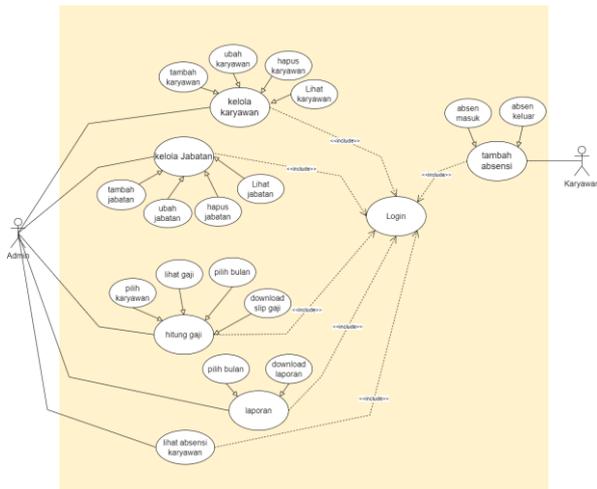
mencakup mengevaluasi seberapa banyak sumber daya web yang dibutuhkan dalam kondisi normal penggunaan. Selanjutnya, *Test Case Development* melibatkan pembuatan skrip atau skenario pengujian yang akan digunakan oleh alat seperti JMeter untuk melakukan pengujian yang terstruktur dan menyeluruh. Terakhir, *Test Planning* melibatkan perencanaan strategi dan pendekatan pengujian untuk memastikan bahwa semua aspek dari aplikasi atau sistem diuji secara efektif dan efisien sepanjang siklus pengembangan perangkat lunak[9].



Gambar 1. Software Testing Life Cycle

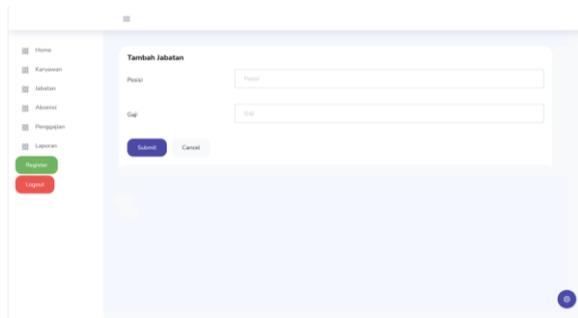
2.1 Analisis Kebutuhan

Analisis kebutuhan sistem termasuk perangkat lunak yang akan diuji, fiturnya, *use case* diagram, dan persyaratan dan spesifikasinya. Dalam penelitian ini, perangkat lunak yang diuji adalah aplikasi *salaryMSI*. *Figure 2* menunjukkan *use case* diagram untuk aplikasi. Sistem akan digunakan oleh pengguna melalui interaksi antara pengguna dan sistem, yang meliputi langkah-langkah atau tindakan yang harus dilakukan untuk mencapai tujuan tertentu dalam penggunaan sistem tersebut[10].

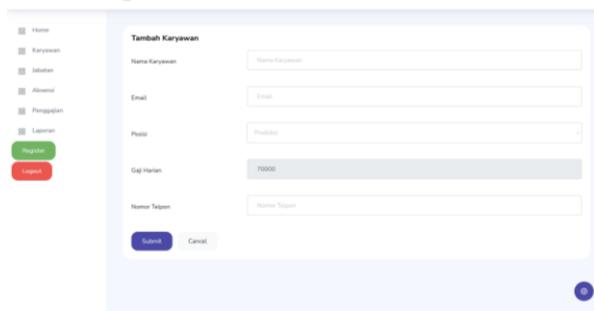


Gambar 2. Use case diagram SalaryMSI

Dari figure 2 peneliti akan menguji fitur yang ada pada aplikasi salaryMSI yaitu kelola karyawan dan kelola jabatan.



Gambar 3. Antarmuka Fitur Data Jabatan



Gambar 4. Antarmuka Fitur Data Karyawan

Tabel 1 dan Tabel 2 menunjukkan persyaratan dan spesifikasi sistem untuk fitur data jabatan dan data karyawan pada aplikasi salaryMSI. Analisis kebutuhan tambahan mencakup kriteria dan spesifikasi sistem untuk fitur yang akan diuji. Ini berfungsi sebagai standar untuk pengembangan *test case* pada tahap berikutnya.

Tabel 1. Persyaratan Fitur

Kode	Bagian yang diuji	Kriteria
K01	Posisi	Mengisi data sesuai formatnya

K02	Gaji	Mengisi data sesuai formatnya
K03	ID Karyawan	Menambahkan ID Karyawan sesuai dengan ketentuan
K04	Nama Karyawan	Mengisi data sesuai formatnya
K05	Email	Mengisi data sesuai formatnya
K06	Nomor Telpun	Mengisi data sesuai formatnya

Tabel 2. Spesifikasi Fitur

Kode	Bagian yang diuji	Kriteria
K07	Posisi	Berisikan antara 1-255 karakter
K08	Gaji	Berisikan angka dengan ketentuan 1-11 karakter
K09	ID Karyawan	Menambahkan ID Karyawan dengan syarat minimal (1-4)
K10	Nama Karyawan	Berisikan antara 1-255 karakter
K11	Email	Berisikan antara 1-255 karakter
K12	Nomor Telpun	Berisikan antara 1-255 karakter

Metode Equivalence Partitioning (EP) membagi *input* ke dalam berbagai partisi, yang kemudian dikelompokkan berdasarkan kriteria valid (*true*) dan tidak valid (*false*). Pembagian partisi ini dilakukan dengan mempertimbangkan kebutuhan dan spesifikasi sistem, sehingga setiap partisi mencerminkan kondisi yang berbeda dalam konteks operasional sistem tersebut. Di sisi lain, *Boundary Value Analysis (BVA)* menguji elemen-elemen spesifikasi sistem dengan menggunakan nilai-nilai batas. Nilai-nilai batas ini diidentifikasi sebagai titik-titik kritis di sekitar batas-batas partisi yang dibentuk oleh metode EP, dan pengujian ini bertujuan untuk memastikan bahwa sistem berfungsi dengan benar pada titik-titik ekstrem atau batas tersebut.

2.2 Pembuatan Test Case

Pada langkah ini, proses perencanaan pengujian digunakan untuk membuat berbagai kasus uji yang komprehensif dan relevan. Hasil dari pembuatan kasus uji tersebut kemudian disusun dan disajikan secara rinci dalam Tabel 3 dan Tabel 4, yang memuat informasi penting tentang setiap kasus uji, termasuk tujuan, kondisi awal, langkah-langkah yang harus diikuti, serta hasil yang diharapkan dari setiap pengujian. Penyajian ini bertujuan untuk memberikan gambaran yang jelas dan terstruktur mengenai berbagai skenario pengujian yang dirancang untuk memastikan bahwa sistem berfungsi sesuai dengan spesifikasi dan persyaratan yang telah ditetapkan.

2.3. Pengujian Test Case

Pada tahap ini, peneliti akan melaksanakan pengujian terhadap kasus-kasus uji yang telah dibuat dan dirancang sebelumnya. Setiap kasus uji akan dijalankan secara menyeluruh untuk memverifikasi apakah hasil yang diperoleh sesuai dengan hasil yang diharapkan berdasarkan spesifikasi dan persyaratan sistem. Jika hasil pengujian tersebut sesuai dengan yang diharapkan, maka pengujian dianggap berhasil dan validasi sistem dianggap memadai. Namun, jika hasil pengujian tidak sesuai dengan yang diharapkan, maka pengujian perlu dievaluasi kembali dan dilakukan perbaikan yang diperlukan. Proses perbaikan ini mencakup identifikasi dan analisis penyebab ketidaksesuaian, serta modifikasi kasus uji atau sistem yang diuji untuk memastikan bahwa pengujian berikutnya dapat menghasilkan hasil yang sesuai dengan harapan.

2.4 Evaluasi

Evaluasi merupakan langkah terakhir dalam keseluruhan proses ini. Pada tahap evaluasi, dilakukan analisis mendalam terhadap hasil dari semua kasus uji yang telah dijalankan. Langkah ini bertujuan untuk menyimpulkan apakah aplikasi sistem pakar berfungsi dengan baik sesuai dengan spesifikasi dan persyaratan yang telah ditetapkan. Evaluasi mencakup pemeriksaan menyeluruh terhadap kinerja sistem, identifikasi keberhasilan pengujian, dan deteksi potensi masalah atau ketidaksesuaian yang mungkin masih ada. Berdasarkan analisis ini, ditentukan apakah sistem dapat diimplementasikan dalam lingkungan operasionalnya atau apakah diperlukan perbaikan lebih lanjut. Jika ditemukan masalah atau kekurangan, langkah-langkah perbaikan dan penyesuaian akan diidentifikasi dan direncanakan untuk memastikan bahwa sistem berfungsi secara optimal sebelum digunakan secara penuh.

3. Hasil dan Pembahasan

Tabel 3. Pengujian Equivalence Partitioning

No Uji	Deskripsi Pengujian	Nilai	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
U01	Memasukkan posisi yang sesuai	Valid	Sistem akan menyimpan posisi	Posisi akan ditambahkan	Sesuai
U02	Mengosongkan posisi	Invalid	Sistem akan memberikan pesan error	Posisi gagal ditambahkan	Sesuai
U03	Memasukkan gaji berupa angka	Valid	Sistem yang menyimpan data gaji	Gaji akan ditambahkan	Sesuai
U04	Mengosongkan gaji	Invalid	Sistem memberikan pesan error	Gaji gagal ditambahkan	Sesuai

U05	Memasukkan ID Karyawan yang sesuai	Valid	Sistem akan menyimpan ID Karyawan	ID Karyawan akan ditambahkan	Sesuai
U06	Mengosongkan ID Karyawan	Invalid	Sistem akan memberikan pesan error	ID Karyawan gagal ditambahkan	Sesuai
U07	Memasukkan nama karyawan yang sesuai	Valid	Sistem akan menyimpan nama karyawan	Nama karyawan akan ditambahkan	Sesuai
U08	Mengosongkan nama karyawan	Invalid	Sistem akan memberikan pesan error	Nama karyawan gagal ditambahkan	Sesuai
U09	Memasukkan E-mail yang sesuai	Valid	Sistem akan menyimpan e-mail	E-mail akan ditambahkan	Sesuai
U10	Mengosongkan E-mail	Invalid	Sistem akan memberikan pesan error	E-mail gagal ditambahkan	Sesuai
U11	Memasukkan Nomor Telp	Valid	Sistem akan menyimpan nomor telp	Nomor telp akan ditambahkan	Sesuai
U12	Mengosongkan nomor telp	Invalid	Sistem akan memberikan pesan error	Nomor telp gagal ditambahkan	Sesuai

Tabel 4. Pengujian Boundary Value Analysis

No Uji	Deskripsi Pengujian	Nilai	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
U13	Memasukkan ID Karyawan 0 karakter	BLB	Sistem akan memberikan pesan error yang mengharuskan mengisi ID Karyawan	ID Karyawan gagal ditambahkan	Sesuai
U14	Memasukkan ID Karyawan 1 karakter	LB	Sistem menerima data ID Karyawan	ID Karyawan diterima sistem	Sesuai
U15	Memasukkan ID Karyawan 4 karakter	UB	Sistem menerima data ID Karyawan	ID Karyawan diterima sistem	Sesuai
U16	Memasukkan	AUB	Sistem akan	Terdapat pesan error	Sesuai

	ID Karyawan 5 karakter		memberikan pesan error			U27	Masukkan Nomor Telepon 255 karakter	UB	Sistem menyimpan Nomor Telepin	Nomor Telepon ditambahkan	Sesuai
U17	Memasukkan Nama Karyawan 0 karakter	BLB	Sistem akan memberikan pesan error yang mengharuskan mengisi Nama Karyawan	Nama Karyawan gagal ditambahkan	Sesuai	U28	Masukkan Nomor Telepon 256 karakter	AUB	Sistem akan memberikan pesan error	Nomor Telepon gagal ditambahkan	Sesuai
U18	Memasukkan Nama Karyawan 1 karakter	LB	Sistem menerima data Nama Karyawan	Nama Karyawan diterima sistem	Sesuai						
U19	Memasukkan Nama Karyawan 255 karakter	UB	Sistem menerima data Nama Karyawan	Nama Karyawan diterima sistem	Sesuai						
U20	Memasukkan Nama Karyawan 256 karakter	AUB	Sistem akan memberikan pesan error	Terdapat pesan error	Sesuai						
U21	Memasukkan E-Mail 0 karakter	BLB	Sistem akan memberikan pesan error yang mengharuskan mengisi E-Mail	E-Mail gagal ditambahkan	Sesuai						
U22	Memasukkan E-Mail 1 karakter	LB	Sistem menerima E-Mail	E-Mail diterima sistem	Sesuai						
U23	Memasukkan E-Mail 255 karakter	UB	Sistem menerima E-Mail	E-Mail diterima sistem	Sesuai						
U24	Memasukkan E-Mail 256 karakter	AUB	Sistem akan memberikan pesan error	Terdapat pesan error	Sesuai						
U25	Memasukkan Nomor Telepon 0 karakter	BLB	Sistem akan memberikan pesan error yang mengharuskan mengisi Nomor Telepon	Nomor Telepon gagal ditambahkan	Sesuai						
U26	Masukkan Nomor Telepon 1 karakter	LB	Sistem menyimpan Nomor Telepon	Nomor Telepon ditambahkan	Sesuai						

4. Conclusions

Berdasarkan hasil pengujian *black box* dengan menggunakan metode nilai batas dan pembagian kesesuaian, dapat disimpulkan bahwa Pengujian *black box* dengan analisis nilai batas pada tabel 4 menunjukkan bahwa dari 16 kasus yang diuji *metode Boundary Value Analysis* ada kesesuaian 100%. Kemudian, pengujian *metode Equivalence Partitioning* memiliki kesesuaian menunjukkan bahwa dari 12 kasus yang diuji, tidak ada kesalahan sistem

References

- [1] T. S. Jaya, "Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)," *J. Inform. J. Pengemb. IT*, vol. 3, no. 1, pp. 45–48, 2018, doi: 10.30591/jpit.v3i1.647.
- [2] V. Febrian, M. R. Ramadhan, M. Faisal, and A. Saifudin, "Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox," *J. Inform. Univ. Pamulang*, vol. 5, no. 1, p. 61, 2020, doi: 10.32493/informatika.v5i1.4340.
- [3] N. Mahrozi and Muhammad Ainul Yaqin, "PENGUJIAN APLIKASI DENGAN METODE BLACKBOX TESTING: ANALISIS BOUNDARY VALUE DAN EQUIVALENCE PARTITIONING PADA APLIKASI SISTEM PAKAR KUCING," vol. 2, pp. 257–265, 2024.
- [4] D. Novianti and D. Anjani, "Pengujian Aplikasi E-Farmer Dalam Perhitungan Keuntungan Dengan Metode Blackbox Testing Boundary Value Analysis," *JUNIF - J. Nas. Inform.*, vol. 1, no. 2, pp. 76–81, 2020.
- [5] 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," *Log. J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 1, pp. 1–5, 2022, [Online]. Available:

<https://journal.mediapublikasi.id/index.php/logic>

- [6] B. A. Maulana, E. Mawarni, M. Y. Hidayattuloh, V. Suryawijaya, and A. Saifudin, "Penguujian 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.
- [7] A. A. Arwaz, T. Kusumawijaya, R. Putra, K. Putra, and A. Saifudin, "Penguujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions," *J. Teknol. Sist. Inf. dan Apl.*, vol. 2, no. 4, p. 130, 2019, doi: 10.32493/jtsi.v2i4.3708.
- [8] R. Pramudita, "Penguujian Black Box pada Aplikasi Ecampus Menggunakan Metode Equivalence Partitioning," *INFORMATICS Educ. Prof. J. Informatics*, vol. 4, no. 2, p. 193, 2020, doi: 10.51211/itbi.v4i2.1347.
- [9] I. G. N. P. D. D. D. Diastama, I. M. Sukarsa, and N. K. A. Wirdiani, "Pengembangan Test Script Untuk Load Testing Web Dengan Metode Software Testing Life Cycle," *J. Ilm. Teknol. dan Komput.*, vol. 2, no. 1, pp. 311–318, 2021.
- [10] S. Perhitungan *et al.*, "Penerapan Metode Equivalence Partitioning Pada Penguujian Otomatis," pp. 219–223.