

## Pengujian *Black Box* pada Aplikasi Penjualan Berbasis Web Menggunakan Metode *Equivalents Partitions* (Studi Kasus: PT Arap Store)

Arief Kurniawan<sup>1</sup>, Arief Maulana<sup>2</sup>, Verdian Ramadika Sukma<sup>3</sup>, Wini Keumala<sup>4</sup>, Aries Saifudin<sup>5</sup>

Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

E-mail: <sup>1</sup>arifkurniawan1514@gmail.com, <sup>2</sup>ariefmaulana2244@gmail.com,  
<sup>3</sup>reapergrim06@gmail.com, <sup>4</sup>keumalawini@gmail.com, <sup>5</sup>aries.saifudin@unpam.ac.id

Submitted Date: January 30<sup>th</sup>, 2020  
Revised Date: January 31<sup>st</sup>, 2020

Reviewed Date: January 30<sup>th</sup>, 2020  
Accepted Date: January 31<sup>st</sup>, 2020

### **Abstract**

*In a software test, validation is a process of checking whether the software meets the specifications and intended purposes. Data validation process that has not been maximized can result in data that is stored in the database is not appropriate, including the login process that becomes an error. This can be detrimental to the owner and user in using this application. Therefore, it is necessary to improve the quality of validation more accurately, so that it can help the owner in the accuracy of the data.*

*Applications that are designed in such a way must go through a testing phase to ensure the quality of the software itself. A good test case is if the test has the possibility of finding an error that is not revealed. Among the many ways of testing Black Box, in this study the Equivalence Partition testing method was chosen. This method tests solving or dividing the input domain from the program into data classes so that a test case can be obtained, which is considered quite relevant to test sales applications at PT Arap Store. Stages of research carried out begins by determining the functionality to be tested, designing test scenarios, determining the data to be tested, determining input can be numeric values, range values, related value collections or Boolean conditions in accordance with the database structure that has been created, conducting experiments testing, documenting research results, and drawing conclusions.*

*Based on the results of tests that have been carried out, it gives a conclusion that this software can be used properly, after a repair has been made on the errors found. However, this test can be said to be imperfect, because it is only done by using several sample forms. From the results of this test it is expected that the application can be used according to user requirements.*

*Keywords: Software, Testing, Black Box, Equivalence partitioning*

### **Abstrak**

Dalam sebuah pengujian perangkat lunak, validasi merupakan suatu proses memeriksa apakah perangkat lunak memenuhi spesifikasi dan tujuan yang dimaksudkan. Proses validasi data yang belum maksimal dapat mengakibatkan data yang disimpan pada database tidak sesuai, di antara proses *login* yang menjadi *error*. Hal ini dapat merugikan pemilik dan pengguna dalam menggunakan aplikasi ini. Oleh karena itu, diperlukan perbaikan kualitas validasi yang lebih akurat, sehingga dapat membantu pemilik dalam keakuratan data.

Aplikasi yang dirancang sedemikian rupa harus melalui tahap pengujian untuk memastikan kualitas dari perangkat lunak itu sendiri. Suatu kasus pengujian yang baik adalah apabila pengujian tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Di antara sekian banyak cara pengujian *Black Box*, pada penelitian ini dipilih metode pengujian *Equivalence Partition*. Metode ini menguji memecahkan atau membagi domain input dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh, yang dinilai cukup relevan untuk menguji aplikasi penjualan pada PT Arap Store. Tahapan penelitian yang dilakukan dimulai dengan menentukan fungsionalitas yang akan diuji, merancang skenario pengujian, menentukan data yang akan diuji, menentukan *input* dapat berupa nilai *numeric*, *range* nilai, kumpulan nilai yg berhubungan atau kondisi *Boolean* sesuai dengan struktur basis

data yang telah dibuat, melakukan percobaan pengujian, mendokumentasikan hasil penelitian, dan menarik kesimpulan.

Berdasarkan hasil pengujian yang telah dilakukan, memberikan kesimpulan bahwa perangkat lunak ini dapat digunakan dengan baik, setelah dilakukan perbaikan pada kesalahan yang ditemukan. Namun pengujian tersebut dapat dikatakan belum sempurna, karena hanya dilakukan dengan menggunakan beberapa form sampel. Dari hasil pengujian ini diharapkan aplikasi dapat digunakan sesuai kebutuhan pengguna.

Kata Kunci: Perangkat Lunak, Pengujian, *Black Box*, *Equivalence partitioning*

## 1. Pendahuluan

Pengujian perangkat lunak adalah cara untuk mendapatkan informasi mengenai kualitas dari perangkat lunak yang sedang diuji (Sulistiyanto & Azhari, 2014), pengujian perangkat lunak dilakukan untuk mendeteksi adanya kesalahan, yang menyebabkan kegagalan perangkat lunak (Irawan, 2017). Proses pengujian untuk mengidentifikasi bagian dari perangkat lunak yang rawan mengalami kegagalan (Hanifah, Ronggo, & Sugiarto, 2016), sebuah perangkat lunak dinyatakan gagal, jika perangkat lunak tersebut tidak memenuhi spesifikasi (Komarudin, 2016).

Tujuan pengujian perangkat lunak dilakukan untuk mencari kesalahan perangkat lunak (Wibisono & Baskoro, 2016). Pengujian perangkat lunak juga bertujuan untuk memperoleh produk yang berkualitas yang memberikan produktivitas tinggi. Dalam proses pengujian perangkat lunak, untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan (Komarudin, 2016).

Pengujian yang dilakukan sebelumnya tidak efektif dan tidak lengkap mengakibatkan berbagai masalah ketika perangkat lunak tersebut digunakan, sehingga sistem dikatakan layak untuk digunakan (Nurudin, Jayanti, Saputro, Saputra, & Yulianti, 2019). Masalah yang timbul dalam sebuah perangkat lunak dapat menyebabkan kerugian terutama bagi pemilik, oleh karena itu dilakukan kembali pengujian untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak (Syaban, 2015).

Tujuan utama dari pengujian perangkat lunak sebenarnya sederhana yaitu untuk memastikan bahwa *software* yang dihasilkan sesuai dengan kebutuhan (*requirement*) yang sebelumnya ditentukan. Ketika *requirement* dari suatu sistem telah disusun maka semestinya sudah ada suatu pengujian perencanaan (*test plan*). Selain itu suatu proses testing membutuhkan tujuan akhir yang dapat dinilai sehingga pihak tester bisa berhenti melakukan suatu testing ketika

tujuan-tujuan itu tercapai (Krismadi, et al., 2019). Sehingga sangat perlu untuk melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019).

Pengujian yang dilakukan harus dirancang dengan baik agar dapat menemukan kesalahan secara sistematis dan dapat diperbaiki dalam waktu yang singkat. Pengujian *Black box* berbasis *Equivalence Partioning* berusaha untuk menemukan kesalahan fungsi-fungsi yang hilang atau salah, seperti kesalahan desain antarmuka atau tampilan, kesalahan dalam struktur data atau akses menuju *database* dan kesalahan performa (Febiharsa, Sudana, & Hudallah, 2018). Dalam proses pengujian perangkat lunak untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan.

Pengujian dilakukan untuk memastikan bahwa masing-masing fungsi beroperasi dengan sepenuhnya dan mencari kesalahan pada setiap fungsi yang berfokus pada memasukkan sistem data, tampilan, pemakaian memori dan kecepatan eksekusi data, sehingga jika memasukkan data tidak sesuai dengan apa yang diharapkan maka sistem gagal (Sidi, Fajri, & Hendra, 2015). Berdasarkan pengujian yang dilakukan maka akan diketahui kelemahan pada sistem informasi setelah dilakukan pengujian menggunakan metode *Black Box* berbasis *Equivalence Partioning* dan bagaimana untuk mengetahui hasil yang dianggap *valid*. Sedangkan untuk tujuan dari pengujian ini adalah mengetahui kelemahan dari sistem agar data yang dihasilkan sesuai dengan data yang dimasukkan setelah data dieksekusi dan menghindari kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh user. (Hidayat, 2018)

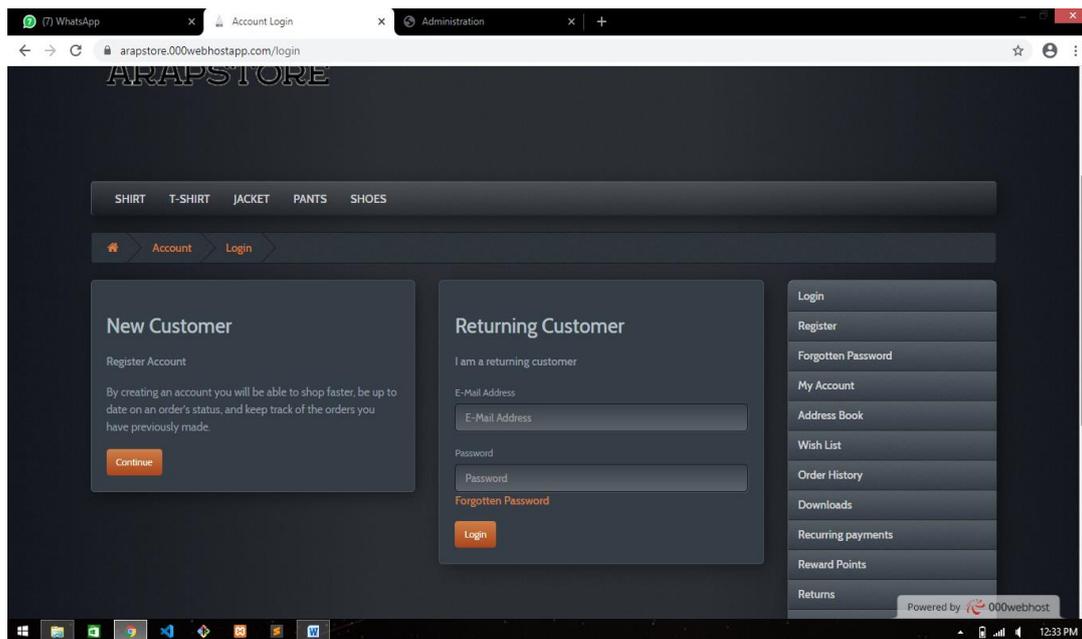
## 2. Metodologi

Tujuan utama dari pengujian perangkat lunak sebenarnya sederhana yaitu untuk memastikan bahwa *software* yang dihasilkan sesuai dengan kebutuhan (*requirement*) yang

sebelumnya ditentukan. Ketika requirement dari suatu sistem telah disusun maka semestinya sudah ada suatu pengujian perencanaan (*test plan*). Selain itu suatu proses testing membutuhkan tujuan akhir yang dapat dinilai sehingga pihak tester bisa berhenti melakukan suatu testing ketika tujuan-tujuan itu tercapai (Krismadi, et al., 2019).

Dalam tahap pengujian *Equivalence Partitioning* langkah pertama yang harus dilakukan adalah membuat rancangan *test case* berdasarkan fungsi yang ada dalam pengujian perangkat lunak.

Kemudian membuat batasan pengujian *Equivalence Partitioning*, setelah membuat Batasan pengujian, langkah selanjutnya adalah membuat model pengujian dari skenario pengujian dan hasil yang diharapkan, dan yang terakhir melakukan pengujian berdasarkan model yang dirancang. Hal ini dilakukan untuk mendapatkan data berupa dokumentasi pengujian dengan metode *Equivalence Partitions* dan nilai tingkat efektifitas metode *Equivalence Partitions* (Jaya, Gumilang, Wati, Andersen, & Desyani, 2019).



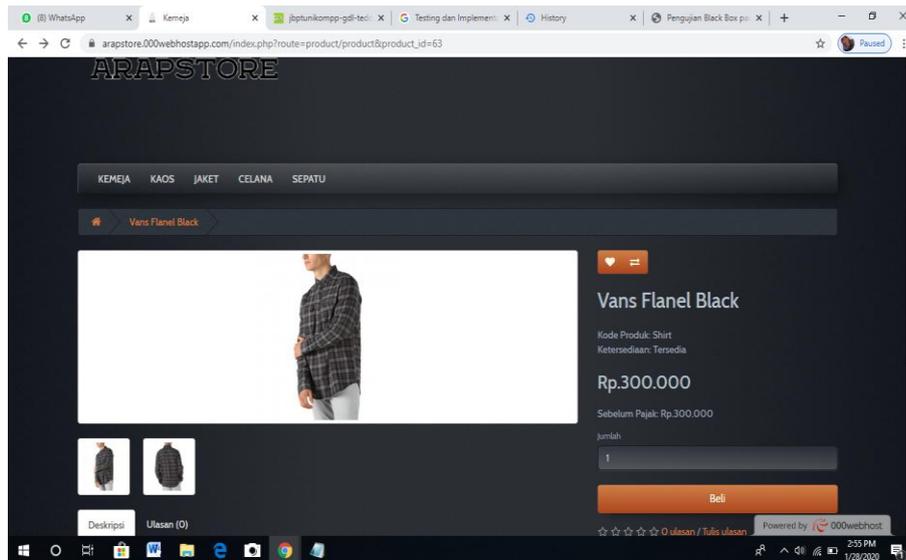
Gambar 1 Form Login

Berdasarkan form pada Gambar 1 terdapat beberapa rencana pengujian. Pada rencana pengujian *E-mail* data akan valid jika *E-mail* harus diisi dengan memasukkan kata (contohnya "Ariefmaulana9898@gmail.com") dengan *password* ("guasiapa98"), setelah mengisi form maka Sistem menerima dan menampilkan

pemberitahuan ("*Success*"), apabila jika *password* diisi dengan *password* yang salah contohnya ("123") maka sistem akan menolak dan menampilkan pemberitahuan ("*Alamat E-mail dan password tidak cocok*") dan begitu juga apabila mengosongkan *E-mail* dan *password*.

Tabel 1 Rancangan Test Case Form login

Id	Deskripsi pengujian	Hasil yang diharapkan
A001	Mengisi <i>E-mail</i> dengan "ariefmaulana9898@gmail.com dan <i>Password</i> dengan "guasiapa98" lalu menekan tombol " <i>Login</i> "	Sistem menerima dan menampilkan pemberitahuan " <i>Success</i> " Kemudian tampil halaman beranda
A002	Mengisi <i>E-mail</i> dengan "ariefmaulana9898@gmail.com dan <i>Password</i> dengan "123" lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan pemberitahuan " <i>Alamat E-mail dan Password tidak cocok</i> "
A003	Mengkosongkan <i>E-mail</i> dan <i>Password</i> lalu menekan tombol " <i>Login</i> "	Sistem akan menolak dan menampilkan " <i>Alamat E-mail dan Password tidak cocok</i> "



Gambar 2 Form Pembelian Barang

Berdasarkan form pada Gambar 2 terdapat beberapa rencana pengujian. Pada rencana pengujian form pembelian barang, pembeli harus mengisi jumlah barang yang ingin dipesan contohnya (“1”), kemudian menekan tombol beli dan sistem menerima dan menampilkan

pemberitahuan (“Anda telah menambahkan Barang ke keranjang belanja anda!”). Apabila pembeli mengosongkan jumlah barang yang ingin dipesan maka sistem akan menolak dan menampilkan pemberitahuan (“Daftar belanja anda kosong!”).

Tabel 2 Rancangan Test Case Form Pembelian Barang

Id	Deskripsi pengujian	Hasil yang diharapkan
B001	Mengisi jumlah barang yang ingin dipesan “1” lalu menekan tombol “Beli”	Sistem menerima dan menampilkan pemberitahuan “Anda telah menambahkan Barang ke keranjang belanja anda!” Kemudian tampil halaman keranjang
B002	Kosongkan jumlah barang yang ingin dipesan “0” lalu menekan tombol “Beli”	Sistem akan menolak dan menampilkan pemberitahuan “Daftar belanja anda kosong!”

 A screenshot of a web form titled 'Kasir' for customer registration. The form is divided into two steps: 'Langkah 1: Pilihan Checkout' and 'Langkah 2: Rincian Penagihan'. The registration fields include:
 

- Nama depan: Arief
- Nama Belakang: Maulana
- Perusahaan: DistrolJye
- Alamat 1: jl.gembor no 26
- Alamat 2: Alamat 2
- Kota: Tangerang
- Kode Pos: 12312
- Negara: Indonesia
- Wilayah / Propinsi: Banten

 A 'Lanjutkan' button is located at the bottom right of the form.

Gambar 3 Form Biodata Pembeli

Berdasarkan form pada gambar 3 terdapat beberapa rencana pengujian. Pada rencana pengujian form biodata pembeli, pembeli harus mengisi kolom Nama Depan dengan contoh (“Arief”), mengisi kolom Nama belakang dengan contoh (“Maulana”), mengisi kolom perusahaan dengan contoh (“DistroUye”), mengisi kolom Alamat 1 dengan contoh (“Jl.gembor no 26”), mengisi kolom Kota dengan contoh (“Tangerang”) mengisi kolom Kode Pos dengan contoh

(“12312”), mengisi kolom Negara dengan contoh (“Indonesia”), mengisi kolom Wilayah/Propinsi dengan contoh (“Banten”), setelah mengisi semua form maka Sistem mampu menyimpan data ke dalam database dan menampilkan pemberitahuan (“Data sudah tersimpan!”). Apabila pembeli tidak mengisi data pada kolom Kode Pos di form biodata pembeli maka sistem akan menolak untuk menyimpan data dan menampilkan pemberitahuan (“Data tidak boleh kosong!”).

Tabel 3 Rancangan Test Case Form Biodata Pembeli

Id	Deskripsi Pengujian	Hasil Yang Diharapkan
C001	Mengisi Nama depan dengan “Arief”, Nama belakang dengan “Maulana”, Perusahaan dengan “DistroUye”, Alamat 1 dengan “Jl.gembor no 26”, Kota dengan “Tangerang” Kode Pos dengan “12312”, Negara dengan “Indonesia”, Wilayah/Propinsi dengan “Banten”	Sistem mampu menyimpan data ke dalam database dan menampilkan pemberitahuan “Data sudah tersimpan!”
C002	Tidak mengisi data Kode Pos di form biodata pembeli	Sistem menolak untuk menyimpan data dan menampilkan pemberitahuan “Data tidak boleh kosong!”

### 3. Hasil dan Pembahasan

Pengujian dilakukan pada sistem informasi penjualan dengan tujuan untuk mengetahui kekurangan yang ada pada sistem sebelum digunakan oleh pengguna. Hasil dari pengujian jika ditemukan kelemahan dalam sistem maka pengguna bisa mengembangkan pada bagian

tersebut yang dianggap lemah. Pada hasil pengujian terdapat tabel *test case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian atau tidak. Pengujian akan dilakukan dengan metode *Black Box* berbasis *Equivalence Partitioning*.

Tabel 4 Hasil Pengujian *Equivalence Partitioning*

Id	Deskripsi Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A001	Mengisi <i>E-mail</i> dengan “ariefmaulana9898@gmail.com dan <i>Password</i> dengan “guasiapa98” lalu menekan tombol “Login”	Sistem menerima dan menampilkan pemberitahuan “Success” Kemudian tampil halaman beranda	Sistem menampilkan pemberitahuan “Success” Kemudian tampil halaman beranda	Berhasil
A002	Mengisi <i>E-mail</i> dengan “ariefmaulana9898@gmail.com dan <i>Password</i> dengan “123” lalu menekan tombol “Login”	Sistem akan menolak dan menampilkan pemberitahuan “Alamat <i>E-mail</i> dan <i>Password</i> tidak cocok”	Sistem menampilkan pemberitahuan “Alamat <i>E-mail</i> dan <i>Password</i> tidak cocok”	Berhasil
A003	Kosongkan <i>E-mail</i> dan <i>Password</i> lalu menekan tombol “Login”	Sistem akan menolak dan menampilkan “Alamat <i>E-mail</i> dan <i>Password</i> tidak cocok”	Sistem menampilkan pemberitahuan “Alamat <i>E-mail</i> dan <i>Password</i> tidak cocok”	Berhasil
B001	Kosongkan jumlah barang yang ingin dipesan “1” lalu menekan	Sistem akan menolak dan menampilkan	Sistem menampilkan pemberitahuan	Berhasil

	tombol “Beli”	pemberitahuan “Daftar belanja anda kosong!”	“Anda telah menambahkan Barang ke keranjang belanja anda!”	
B002	Kosongkan jumlah barang yang ingin dipesan “0” lalu menekan tombol “Beli”	Sistem akan menolak dan menampilkan pemberitahuan “Daftar belanja anda kosong!”	Sistem menampilkan pemberitahuan “Daftar belanja anda kosong!”	Berhasil
C001	Mengisi Nama depan dengan “Arief”, Nama belakang dengan “Maulana”, Perusahaan dengan “DistroUye”, Alamat 1 dengan “Jl.gembor no 26”, Kota dengan “Tangerang” Kode Pos dengan “12312”, Negara dengan “Indonesia”, Wilayah/Propinsi dengan “Banten”	Sistem mampu menyimpan data ke dalam database dan menampilkan pemberitahuan “Data sudah tersimpan!”	Sistem menampilkan pemberitahuan “Data sudah tersimpan!”	Berhasil
C002	Tidak mengisi data Kode Pos di form biodata pembeli	Sistem menolak untuk menyimpan data dan menampilkan pemberitahuan “Data tidak boleh kosong!”	Sistem menampilkan pemberitahuan “Data tidak boleh kosong!”	Berhasil

#### 4. Kesimpulan

Pemeriksaan yang dilakukan untuk memastikan bahwa fungsi sistem bekerja dengan sebaik-baiknya dan memeriksa kesalahannya. Dalam menentukan tingkat akurasi agar semua parameter akurasi yang terkait seperti tahapan kerahasiaan, integritas data, dan ketersediaan data dapat sesuai dengan kebutuhan, maka diperlukannya proses pengujian metode lain guna sebagai nilai standar keamanan informasi, selain menggunakan metode *Equivalence Partitions*. Setelah dilakukan pengujian *Black Box* berbasis *Equivalence Partioning* terhadap sistem informasi penjualan pada Arap Store dapat disimpulkan dalam pengujian tidak ditemukan kesalahan.

#### 5. Saran

Aplikasi Arap Store diharapkan dapat dikembangkan agar aplikasi ini dapat memberikan informasi yang akurat seputar barang yang tersedia di Arap Store. Pengujian harus dirancang dengan baik agar dapat menemukan kesalahan secara sistematis dan dapat diperbaiki dalam waktu yang singkat. Pengujian dalam penelitian ini juga dapat dilakukan dengan teknik-teknik lain misalnya menggunakan pendekatan *white box testing*.

#### Referensi

Febiharsa, D., Sudana, I. M., & Hudallah, N. (2018). Uji Fungsionalitas (Black Box Testing) Sistem Informasi Lembaga Sertifikasi Profesi

(SILSP) Batik Dengan Apperfect Web Test Dan Uji Pengguna. *Joined Journal*, 1-10.

- Hanifah, Ronggo, A., & Sugiarto. (2016). Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk. *Teknik Informatika Fakultas Teknologi Industri*, 1-8.
- Hidayat, T. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurnal Teknik Informatika UNIS*, 1-5.
- Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja UPT BLK Kabupaten Kudus dengan Metode Whitebox Testing. *Sentra Penelitian Engineering dan Edukasi*, 1-5.
- Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 131-136.
- Komarudin, M. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Mikrotik*, 1-18.
- Krismadi, A., Lestari, A. F., Pitriyah, A., Mardangga, I. W., Astuti, M., & Aries, S. (2019). Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 2(4), 155-161.

- Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 125-130.
- Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 4(4), 143-148.
- Sidi, M., Fajri, F., & Hendra, R. (2015). Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis. *Ilmiah Teknologi Informasi Terapan*, 1-6.
- Sulistiyanto, H., & Azhari. (2014). Urgensi Pengujian Pada Kemajuan Perangkat Lunak Dalam Multi Perspektif. *Jurnal Komunikasi dan Teknologi Informasi*, 1-10.
- Syaban, R. M. (2015). Pengembangan Sistem Informasi Pengelolaan Surat Masuk dan Keluar Berbasis Web di Dinas Sosial Tenaga Kerja dan Transmigrasi Kabupaten Garut Menggunakan Framework PHP. *Jurnal Algoritma*, 1-11.
- Wibisono, W., & Baskoro, F. (2016). Pngujian Perangkat Lunak Dengan menggunakan model Behavior UML. *Jurnal Ilmiah Tekonologi Informasi*, 1-8.