

UJI BLACK BOX TES APLIKASI SOFTWARE DEVELOPMENT SYSTEM INFORMATION (SODEVI) PT. DIMATA SORA JAYATE MENGGUNAKAN KATALON STUDIO

Eland Vaskita Nugraha¹, Renna Yanwastika Ariyana^{2*}, Erna Kumalasari Nurnawati³

^{1,2,3}Program Studi Informatika., Institut Sains & Teknologi AKPRIND Yogyakarta
e-mail :¹evnugraha@gmail.com ,²renna@akprind.ac.id, ³ernakumala@akprind.ac.id
Corresponding Author: renna@akprind.ac.id

ABSTRACT

Software Development System Information (SoDevi) is a web-based application used by programmers and testers to work on projects at PT. In the eyes of Sora Jayate. This application is used to perform enhancements to the UI/UX web display to improve web performance and layout. It is necessary to test the SoDevi application, in order to improve the quality of the application developed by PT. In the eyes of Sora Jayate to minimize bugs or errors in the application. In previous applications that were developed there were problems, namely, when testing manually, human errors often occurred which resulted in the test results not being carried out optimally. By doing automated testing in search of bugs or errors in the system, it is very necessary to do, where one of the tests that can be done is to do a black box test. Performing a black box test will be very influential in testing a system, where this test will look for deficiencies and errors from the system that has been built. Testing is carried out in order to get an assessment, so that the system that has been built is in accordance with expectations and later there is no need to do repeated testing. A system can be said to be valid if the testing process has been carried out, because by testing it will be known with certainty the error function of the process in the system or not. In the black box test, the focus of the test is only on the functional test of the system, not on testing the model or program code. In the system test carried out on the SoDevi application, an automation tool called Katalon Studio will be used which will record the testing and scripting steps to run the test with the test focus only on the create, read, update and delete (CRUD) processes. The results of the black box test carried out on the SoDevi application have been running according to function and no bugs or errors were found from the system. From the black box test which is done automatically using Katalon Studio, it can be concluded that the system is running well where errors that occur in the application are a factor of human error.

Keywords : black box, CRUD, functional, human error, SoDevi

INTISARI

Software Development System Information (SoDevi) adalah aplikasi berbasis web yang digunakan oleh programmer dan tester untuk mengerjakan project di PT. Dimata Sora Jayate. Aplikasi ini digunakan untuk melakukan enhancement pada tampilan UI/UX web untuk meningkatkan performa dan layout web. Perlu adanya pengujian terhadap aplikasi SoDevi, guna meningkatkan kualitas dari aplikasi yang dikembangkan PT. Dimata Sora Jayate guna meminimalisir bug ataupun kesalahan yang ada didalam aplikasi. Pada aplikasi - aplikasi sebelumnya yang dikembangkan terdapat permasalahan yaitu, ketika melakukan pengujian secara manual sering terjadinya human error yang mengakibatkan hasil pengujian tidak dapat dilakukan secara maksimal. Dengan melakukan pengujian secara automasi dalam mencari bug atau kesalahan didalam sistem sangat perlu dilakukan, dimana salah satu pengujian yang dapat dilakukan yaitu dengan melakukan uji black box. Melakukan Uji black box akan sangat berpengaruh dalam menguji sebuah sistem, dimana uji ini akan mencari kekurangan dan kesalahan dari sistem yang telah dibangun. Pengujian dilakukan guna mendapatkan penilaian, agar sistem yang telah dibangun sesuai dengan harapan dan nantinya tidak perlu melakukan pengujian berulang. Sebuah sistem dapat dikatakan valid jika telah dilakukan proses pengujian, karena dengan pengujian akan diketahui secara pasti error fungsi dari proses dalam sistem atau tidak. Pada uji black box fokus pengujian hanya pada uji fungsional sistem bukan pada pengujian model ataupun kode program. Pada uji sistem yang dilakukan pada aplikasi SoDevi digunakan alat automasi dengan nama Katalon Studio yang nantinya akan merekam langkah pengujian dan scripting untuk menjalankan pengujian dengan fokus pengujian hanya pada proses create, read, update dan delete (CRUD). Hasil uji black box yang dilakukan pada aplikasi SoDevi sudah berjalan sesuai dengan fungsi dan tidak ditemukan bug atau error dari sistem. Dari uji black box yang dilakukan secara automasi menggunakan Katalon Studio dapat disimpulkan bahwa sistem sudah berjalan dengan baik dimana kesalahan yang terjadi dalam aplikasi merupakan faktor kesalahan human error.

Kata kunci : black box, CRUD, fungsional, human error, SoDevi

1. PENDAHULUAN

Melakukan pengujian untuk menghasilkan sebuah sistem yang berbobot merupakan salah satu proses yang tidak boleh terlewat dalam pengembangan sistem, hal ini digunakan untuk mengetahui kualitas dari sistem

yang telah dibangun. Apabila tidak dilakukan pengujian tentu pengembang tidak dapat mengetahui kualitas dari sistem yang nantinya memungkinkan terjadinya kesalahan yang akan merugikan sistem tersebut (Ijudin & Saifudin, 2020). Pengujian sistem memiliki fungsi yang sangat penting dalam proses pengembangan sistem untuk menemukan kesalahan atau cacat yang disebabkan perbedaan hasil perancangan dan implementasi dari sistem yang dikembangkan. Pengujian terhadap sistem, merupakan elemen yang digunakan untuk merepresentasikan hasil analisis, perancangan dan implementasi guna menjamin kualitas dari sistem yang dikembangkan. Pengujian sistem dilakukan untuk menemukan kesalahan yang terjadi dalam sistem sebagai dasar dalam menemukan solusi, dan meningkatkan kualitas dari sistem (Pratama, Ristiano, Prayogo, Nasrullah, & Saifudin, 2020).

Untuk mengurangi kesalahan yang terjadi, maka perlu dilakukan sebuah pengujian agar menghindari kerugian (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019). Untuk meminimalisir kesalahan berupa *error* dari sistem tentu diperlukan tindakan preventif berupa pengujian untuk mendeteksi sedini mungkin kesalahan dan kekurangan, sehingga sesegera mungkin sistem diperbaiki sebelum digunakan oleh *user* (Dewi, Pratama, Putera, & Carudin, 2022). Pengujian sistem dilakukan guna mendapatkan sebuah penilaian, agar nantinya tidak perlu dilakukan pengujian berulang ketika tujuan pengujian telah tercapai. Sistem dikatakan gagal apabila saat proses pengoperasian data tidak sesuai dengan harapan. Kelemahan sistem dan hasil yang dianggap valid akan diketahui setelah proses pengujian dilakukan. Dalam pengujian akan diketahui secara pasti *error* dan fungsi dari masing-masing proses di dalam sistem apakah telah berjalan dengan baik seluruhnya (Wulandari, Saepudin, Kinanti, Sudesi, Saifudin, & Yulianti, 2022).

Pengujian *black-box* merupakan salah satu metode pengujian sistem, dimana metode ini berfokus pada pencarian kesalahan dan kekurangan pada fungsional perangkat lunak seperti fungsi, antarmuka, model data, dan akses ke sumber informasi eksternal (Mulyati, Kusyadi, Ashara, Widodo, & Wahyudin, 2022). Uji *black box*, melakukan pengecekan terhadap fungsi sistem sudah dapat berjalan dengan baik atau tidak. Pengujian menggunakan *black box* hanya mampu menguji sistem dari segi fungsional dan tidak menguji model dan kode program untuk mengetahui fungsi input dan output dari sistem. Uji *black box* dilakukan berdasarkan masukan dan luaran tanpa memperhatikan kode program, sehingga pengujian tidak perlu memiliki pengetahuan terkait pemrograman (Shaleh, Yogi, Pirdaus, Syawal, & Saifudin, 2021). Pengujian *black box* merupakan pengujian yang paling mudah dilakukan, karena pengguna hanya perlu mengisikan batas atas dan batas bawah dari data yang diharapkan. Banyaknya data yang di uji dihitung dari banyak *field* data entri pada *form* dalam sistem, aturan entri yang harus dipenuhi serta harus memenuhi batas bawah dan batas atas. Dengan metode *black box* akan mampu mendeteksi fungsionalitas masih dapat menerima masukan data yang tidak diharapkan, sehingga menyebabkan data yang tersimpan kurang valid (Cholifah, Yulianingsih, & Sagita, 2018) (Mustaqbal, Firdaus, & Rahmadi, 2015).

PT. Dimata Sora Jayate merupakan perusahaan yang bergerak di bidang teknologi informasi, menyediakan layanan dalam membuat aplikasi, *website*, *e-commerce* dan berbagai produk digital yang dibutuhkan untuk bisnis maupun pribadi dengan fitur lengkap dan dapat ditambahkan atau disesuaikan sesuai dengan kebutuhan pelanggan. PT. Dimata Sora Jayate memiliki *Software Development System Information* (SoDevi) yang merupakan aplikasi berbasis *web* yang digunakan oleh *programmer* dan *tester* dalam berkolaborasi mengerjakan *project*. Aplikasi SoDevi sendiri masih menggunakan tampilan yang sangat sederhana dan belum pernah dilakukan uji *black box* sehingga perlu dilakukan pengujian untuk mencari kesalahan-kesalahan yang terjadi ketika dilakukan *enhancement* pada aplikasi SoDevi. Selain itu juga uji *black box* yang dilakukan digunakan untuk memastikan bahwa fungsionalitas yang sudah dibuat oleh pengembang dapat berjalan dengan baik guna menjamin aplikasi yang dihasilkan sesuai dengan kebutuhan (*requirement*) yang ditetapkan sebelumnya (Maulana, Kurniawan, Keumala, Sukma, & Saifudin, 2020).

Pengujian *black box* pada aplikasi SoDevi PT. Dimata Sora Jayate menggunakan alat automasi bernama Katalon Studio. Dimana katalon studio sendiri merupakan *software* yang digunakan untuk menguji sistem yang tersedia secara gratis dengan mendukung pengujian *web*, *desktop* hingga seluler. Cara kerja katalon studio yaitu dengan merekam langkah pengujian dan *scripting* untuk menjalankan pengujian (Prakoso & Sujarwo, 2022). Pada aplikasi SoDevi dilakukan uji fungsionalitas rekayasa data dengan proses *create*, *read*, *update* dan *delete* (CRUD). Uji fungsionalitas berupa CRUD merupakan pengujian yang digunakan untuk melakukan aksi yang menghubungkan tampilan antar muka sebagai fasilitator dengan perubahan data atau tampilan informasi yang terjadi dalam *database* berbentuk tabel atau laporan yang nantinya akan ditampilkan dalam halaman sistem.

2. METODE PENELITIAN

2.1. Analisis Data

Proses analisis data yang dilakukan dalam penelitian ini nantinya akan digunakan untuk menguji aplikasi SoDevi dari PT. Dimata Sora Jayate. Adapun metode yang digunakan dalam proses analisis data dalam penelitian yang dilakukan dengan:

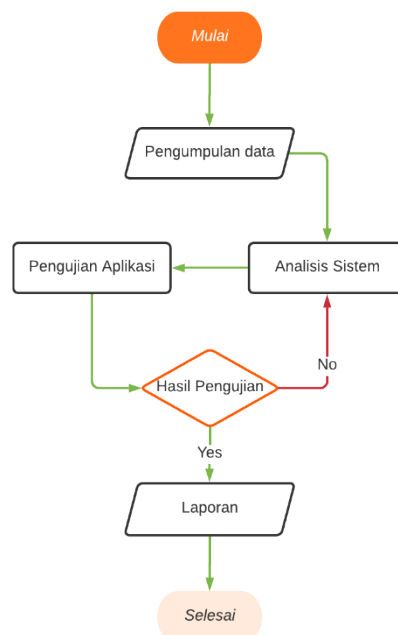
1. *Data collection*, yaitu data dikumpulkan melalui wawancara yang dilakukan dengan *Head of Production* PT. Dimata Sora Jayate selaku pihak yang paham tentang aplikasi SoDevi yang telah di kembangkan dan melakukan dokumentasi terkait aplikasi tersebut. Selain melakukan wawancara dan dokumentasi *data*

collection juga dikumpulkan melalui study literatur dengan mengumpulkan sumber data berupa buku maupun referensi lain terkait pengujian *black box test* yang dilakukan.

2. *Data managing*, yaitu melakukan pengolahan terhadap hasil dokumentasi, wawancara dan studi literatur yang dilakukan menjadi informasi yang sesuai dengan kebutuhan pengujian *black box test* yang dilakukan pada aplikasi SoDevi PT. Dimata Sora Jayate.

2.2. Diagram Alir Penelitian

Pada diagram alir penelitian, akan digambarkan proses demi proses yang dilakukan dalam penelitian. Adapun gambar diagram alir dari penelitian yang dilakukan ditunjukkan pada gambar 1.



Gambar 1. Diagram Alir Penelitian Uji Black Box Test Aplikasi SoDevi

Keterangan:

1. Pengumpulan data : merupakan tahapan pertama yang dilakukan dalam penelitian ini, dimana tahapan pengumpulan data dilakukan proses wawancara dengan ibu Kadek Rida Dwi Wahyuni, S.T selaku *Head of Production* dan kepada I Gede Gunadi Wirawan Amd. Kom selaku *Head of Intelligent System Research & Development Division* PT. Dimata Sora Jayate. Selain wawancara dilakukan pula dokumentasi dan study literatur terkait aplikasi dan proses pengujian yang dilakukan pada aplikasi SoDevi.
2. Analisis sistem : selanjutnya dilakukan analisis sistem dengan mempelajari aplikasi SoDevi dan menentukan *tools* yang akan digunakan dalam menguji aplikasi SoDevi. Pada tahapan ini, ditentukan *software* katalon studio sebagai *tools* yang akan digunakan dalam menguji aplikasi dengan pertimbangan, pengujian yang dilakukan tidak akan memakan waktu yang lama.
3. Pengujian Aplikasi : pada tahapan ini dilakukan pengujian pada aplikasi SoDevi menggunakan *automation testing tool*, dari *software* Katalon Studio.
4. Penyusunan Laporan : selanjutnya penyusunan laporan dilakukan sebagai tahapan akhir dari proses penelitian. Dalam tahapan ini akan disusun sebuah laporan sesuai dengan format yang ditentukan oleh perguruan tinggi sebagai dokumentasi kegiatan penelitian yang telah dilakukan.

2.3. Langkah Alir Pengujian CRUD Aplikasi SoDevi

Tahapan yang dilakukan dalam uji *black box tes* dari aplikasi SoDevi, fokus pada pengujian *Create, Read, Update, dan Delete* (CRUD) produk adapun langkah -langkah dalam pengujian CRUD pada aplikasi SoDevi dilakukan sebagai berikut.

1. Membuka aplikasi Katalon Studio : merupakan tahapan awal yang dilakukan ketika akan melakukan pengujian terhadap aplikasi SoDevi.
2. Menggunakan fitur *web recorder* : Fitur ini digunakan untuk merekam segala aktivitas user baik dari pengisian, klik, dan lain sebagainya yang nantinya aktivitas tersebut akan direkam pada aplikasi Katalon Studio untuk diuji.
3. Melakukan pengujian : Pengujian ini dilakukan dengan memanfaatkan *teks record* dan *klik record* yang sudah tersimpan di Katalon Studio. Setelah terisi semua, Katalon Studio akan menjalankan aplikasi SoDevi secara otomatis pada *browser* yang sudah terkontrol dengan menekan tombol Run. Pada Katalon Studio jika program berjalan dengan baik maka akan muncul status PASSED dengan *highlight* berwarna hijau, dan jika gagal akan muncul status FAILED dengan *highlight* berwarna merah.

4. Memunculkan status pengujian : status pengujian akan dimunculkan pada katalon Studio jika status PASSED pada penginputan *create*, menandakan bahwa aplikasi tidak mengalami masalah apapun sehingga dapat terbaca dengan baik (*read*) namun jika status FAILED menandakan aplikasi mengalami masalah.

3. HASIL DAN PEMBAHASAN

Sistem ini hanya dapat diakses oleh *programmer* dan *tester* PT. Dimata Sora Jayate untuk berkolaborasi dalam menciptakan suatu produk. Hak akses yang dapat dilakukan adalah melakukan CRUD produk pada aplikasi web yang nantinya akan ditampilkan pada laman *website*.

a. Pengujian Halaman *Login*

Gambar 2 merupakan tampilan halaman *login* yang merupakan tampilan awal ketika mengakses sistem. Dalam halaman ini terdapat *form* yang digunakan untuk masuk ke dalam sistem. Pada halaman *login* admin sebagai pengguna dapat memasukkan *username* dan *password* untuk dapat mengakses fitur dari sistem pada aplikasi SoDevi.

Software Development Information System

Gambar 2. Tampilan Halaman *Login* aplikasi SoDevi

Tabel pengujian *blackbox tes* pada halaman *login* dapat dilihat pada Tabel 1 dimana pada hasil pengujian yang dilakukan pada halaman *login* ini tidak terdapat kesalahan atau *error* yang ditemukan.

Tabel 1. Hasil Pengujian *black box test* pada halaman *login* aplikasi SoDevi

<i>Test Case ID</i>	<i>Detail Test Case</i>	<i>Nama Fitur</i>	<i>Langkah pengujian</i>	<i>Hasil yang diharapkan</i>	<i>Hasil yang terjadi</i>	<i>Status</i>
01	<i>User login</i> dengan memasukkan <i>Login name</i> dan <i>password</i> .	<i>Login</i>	<ol style="list-style-type: none"> 1. Membuka web SoDevi 2. Masukkan <i>Login name</i> 3. Masukkan <i>password</i> 4. Klik tombol <i>Login</i> 	<i>Login</i> berhasil dan web menampilkan <i>dashboard</i>	Berhasil sesuai yang diharapkan	<i>PASSED</i>
02	<i>User login</i> dengan tidak memasukkan kolom isian <i>Login name</i> .	<i>Login</i>	<ol style="list-style-type: none"> 1. Membuka web SoDevi 2. Masukkan <i>password</i> 3. Klik tombol <i>Login</i> 	Muncul <i>error</i> , pengguna harus memasukkan kolom isian <i>Login name</i> sebelum <i>login</i> .	Berhasil sesuai yang diharapkan	<i>PASSED</i>
03	<i>User login</i> dengan tidak memasukkan kolom isian <i>password</i> .	<i>Login</i>	<ol style="list-style-type: none"> 1. Membuka web SoDevi 2. Masukkan <i>Login name</i> 3. Klik tombol <i>Login</i> 	Muncul <i>error</i> , pengguna harus memasukkan kolom isian <i>password</i> sebelum <i>login</i> .	Berhasil sesuai yang diharapkan	<i>PASSED</i>
04	<i>User login</i> dengan memasukkan sintaks SQL Injection pada kedua <i>Login name</i> dan <i>password</i> .	<i>Login</i>	<ol style="list-style-type: none"> 1. Membuka web SoDevi 2. Masukkan <i>Login name</i> 3. Masukkan <i>password</i> 4. Klik tombol <i>Login</i> 	Muncul <i>error</i> , pengguna harus memasukkan kolom isian <i>Login name</i> dan <i>password</i> dengan benar sebelum <i>login</i> .	Berhasil sesuai yang diharapkan	<i>PASSED</i>

b. Pengujian Halaman Produk

Gambar 3 Merupakan halaman input barang yang digunakan untuk menginputkan barang yang akan di jual di aplikasi SoDevi. Produk yang telah diinputkan akan langsung masuk ke daftar *list* produk. Pada *list* produk admin dapat menambah, mengubah dan menghapus data.

Code	Name	Abbreviation	Description
001	EARSIP	elektronik arsip	value
001	JAISO	Accounting Information Online System	
002	HANOMAN	Hotel Reservation and Online Management System	
003	HARISMA	Human Resources Management System	
004	PROCHAIN POS	System Retail and Distribution	
005	KASIR	KASIR	
006	SODEVI	Software Development Information System	
007	LOG BOOK	Log Book	
008	IRS	DIMATA REWARD SYSTEM	
009	E- HOTEL	E- Hotel	

Gambar 3. Tampilan Halaman Produk Aplikasi SoDevi

Hasil uji *blackbox* pada halaman produk yang ditunjukkan pada Tabel 2 dimana pada proses input pada edit produk gagal hal ini dikarenakan pemanggilan *mysql database* pada table *code* bersifat *primary*, sehingga produk yang dipilih tidak bisa disunting ketika melakukan input menggunakan *code* yang sudah tersimpan pada *mysql database*. Begitu pula ketika produk ditambahkan menggunakan *code* yang sudah tersimpan pada *mysql database*, hasil menunjukkan penambahan produk tidak dapat ditambahkan.

Tabel 2. Pengujian *Black box test* pada halaman produk aplikasi SoDevi

Test Case ID	Detail Test Case	Nama Fitur	Langkah pengujian	Hasil yang diharapkan	Hasil yang terjadi	Status
01	Pemanggilan <i>mysql</i> menggunakan metode <i>select by code (primary)</i> pada database untuk <i>edit</i> data.	<i>Edit Product</i>	<ol style="list-style-type: none"> 1. Klik <i>Product</i> pada <i>dropdown Software</i> 2. Klik pada salah satu <i>code</i> yang akan di <i>edit</i> 3. Melakukan penyuntingan data pada <i>code</i> dengan nomor yang sudah tersimpan pada database. 4. Klik tombol <i>Save product</i> 	Gagal karena <i>code</i> adalah <i>primary key</i> .	Tidak sesuai yang diharapkan	<i>FAILED</i>
02	Pemanggilan <i>mysql</i> menggunakan metode <i>select by code (primary)</i> pada database untuk <i>edit</i> data.	<i>Edit Product</i>	<ol style="list-style-type: none"> 1. Klik <i>Product</i> pada <i>dropdown Software</i> 2. Klik pada salah satu <i>code</i> yang akan di <i>edit</i> 3. Melakukan penyuntingan data pada <i>code</i> dengan nomor yang belum tersimpan pada database. 4. Klik tombol <i>Save Product</i> 	<i>Edit</i> berhasil, web melakukan <i>refresh</i> halaman.	Berhasil sesuai yang diharapkan	<i>PASSED</i>
02	Pemanggilan <i>mysql</i> menggunakan metode <i>select by code (primary)</i> pada database untuk delete data.	<i>Delete Product</i>	<ol style="list-style-type: none"> 1. Klik <i>Product</i> pada <i>dropdown Software</i> 2. Klik pada salah satu <i>code</i> yang akan di <i>delete</i> 3. Memilih data pada <i>code</i> yang akan dihapus 4. Klik tombol <i>Delete Product</i> 	<i>Delete</i> berhasil, web melakukan <i>refresh</i> halaman.	Berhasil sesuai yang diharapkan	<i>PASSED</i>
03	Melakukan input data product baru dengan menginputkan <i>code (primary)</i> yang sudah tersimpan pada database dan mengisi semua <i>required input</i> .	<i>Input Product</i>	<ol style="list-style-type: none"> 1. Klik <i>Product</i> pada <i>dropdown Software</i> 2. Klik <i>Add New Product</i> 3. Mengisi semua data dan mengisi <i>code</i> dengan <i>value</i> yang sudah ada pada database 4. Klik <i>Save Product</i> 	Gagal karena <i>code</i> adalah <i>primary</i>	Tidak sesuai yang diharapkan	<i>FAILED</i>

4. KESIMPULAN

Pada penelitian yang dilakukan berupa uji *black box* berupa pengujian *automasi* menggunakan Katalon

Studio dengan menguji fungsi CRUD dapat disimpulkan bahwa aplikasi SoDevi sudah berjalan sesuai dengan fungsi dan tidak ditemukan *bug* atau *error* dari sistem. Kesalahan yang terjadi pada pengujian sistem hanyalah faktor dari salah memasukkan *code* dengan *value* yang telah ada di dalam *database* yang merupakan kesalahan *human error*.

UCAPAN TERIMA KASIH

Terimakasih penulis ucapkan kepada PT. Dimata Sora Jayate yang telah memberikan waktu dan kesempatan dalam melaksanakan kegiatan penelitian. Serta terimakasih juga penulis ucapkan kepada Jurusan Informatika Institut Sains & Teknologi AKPRIND Yogyakarta dalam mendukung kelancaran kegiatan penelitian yang penulis lakukan.

DAFTAR PUSTAKA

- Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *Jurnal String; Satuan Tulisan Riset dan Inovasi Teknologi*, 3(2), 206-210. doi:<http://dx.doi.org/10.30998/string.v3i2.3048>
- Dewi, E. H., Pratama, I. S., Putera, A. S., & Carudin. (2022). Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analisis. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 6(3), 315-324. doi:<http://dx.doi.org/10.30998/string.v6i3.11958>
- Ijudin, A., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Berita Online dengan Menggunakan Metode Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 5(1), 8-12. doi:[10.32493/informatika.v5i1.3717](http://dx.doi.org/10.32493/informatika.v5i1.3717)
- Maulana, A., Kurniawan, A., Keumala, W., Sukma, V. R., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store). *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(1), 50-56. doi:<http://dx.doi.org/10.32493/jtsi.v3i1.4307>
- Mulyati, S., Kusyadi, I., Ashara, M. I., Widodo, A. P., & Wahyudin. (2022). Pengujian Black Box ada Aplikasi Hitung Nilai Mahasiswa Menggunakan Metode Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 7(1), 83-88. doi:[10.32493/informatika.v7i1.16224](http://dx.doi.org/10.32493/informatika.v7i1.16224)
- Mustaqbal, M., Firdaus, R. F., & Rahmadi, H. (2015). Pengujian Aplikasi Menggunakan Black Box testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *JITTER: Jurnal Ilmiah Teknologi Informasi Terapan*, 1(3), 32-36.
- 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. doi:[10.32493/informatika.v4i4.3782](http://dx.doi.org/10.32493/informatika.v4i4.3782)
- Prakoso, B., & Sujarwo, A. (2022). Perancangan Automated Testing Pada Studi Kasus Website Indicar. *Jurnal Ilmiah Teknologi Informasi dan Robotika*, 4(1), 29-32. doi:<https://doi.org/10.33005/jifti.v4i1.69>
- Pratama, B. P., Ristiano, I. B., Prayogo, I. A., Nasrullah, & Saifudin, A. (2020). Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. *Journal of Artificial Intelligence and Innovative Applications*, 1(1), 32-36.
- Shaleh, I. A., Yogi, J. P., Pirdaus, P., Syawal, R., & Saifudin, A. (2021). Pengujian Black Box pada Sistem Informasi Penjualan Buku Berbasis Web dengan Teknik Equivalent Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 4(1), 38-45. doi:<http://dx.doi.org/10.32493/jtsi.v4i1.8960>
- Wulandari, A. S., Saepudin, A., Kinanti, M. P., Sudesi, Z., Saifudin, A., & Yulianti. (2022). Pengujian Aplikasi Sistem Informasi Akademik Berbasis Web Menggunakan Metode Black Box Testing Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 5(2), 102-109. doi:[10.32493/jtsi.v5i2.17561](http://dx.doi.org/10.32493/jtsi.v5i2.17561)