

Pengujian Black Box ada Aplikasi Hitung Nilai Mahasiswa Menggunakan Metode Equivalence Partitions

Sri Mulyati¹, Irpan Kusyadi², Muhammad Iqbal Ashara³, Prasetyo Adi Widodo⁴, Wahyudin⁵

Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek No.46, Buaran, Serpong, Kota Tangerang Selatan, Banten Indonesia 15310

email: ¹dosen00391@unpam.ac.id, ²dosen00673@unpam.ac.id ³iqbalashara3@gmail.com, ⁴prstyady@gmail.com, ⁵sinyoyuga@gmail.com

Submitted Date: December 23rd, 2021

Reviewed Date: January 08th, 2022

Revised Date: January 14th, 2022

Accepted Date: March 31st, 2022

Abstract

Black-box testing is very influential in finding flaws and errors in the software because it can identify errors in functions, interfaces, data models, and access to external information sources. During the testing process, the problem often arises that the tester is never sure that the software being tested has met the criteria for being eligible for testing. This condition occurs because sometimes there are still some execution paths that have never been tested. The examiner should be able to make every possible mix of input information for the test material. The selection of input information to detect errors is a problem for testers because it has a high gap, so that an automatic test trial model can be a solution to automatically generate a test trial model in software testing. The black-box method with the identical division technique requires a false awareness technique.

Keywords : Application; Black box; Testing; Quality; Software

Abstrak

Pengujian *black-box* sangat berpengaruh guna mencari kekurangan dan kesalahan dari perangkat lunak karena pengujian tersebut dapat mencari kesalahan pada fungsi, antar muka, model data, dan akses ke sumber informasi eksternal. Saat proses pengujian sering muncul masalah pengujian tidak pernah yakin dengan perangkat lunak yang di uji sudah memenuhi kriteria layak pada pengujian. Kondisi ini terjadi karena terkadang masih ada sebagian jalur eksekusi yang belum pernah dilakukan pengujian. Pengujian harus bisa membuat setiap kemungkinan campuran informasi masukan untuk bahan pengujian. Pemilihan informasi masukan untuk mendeteksi kesalahan menjadi masalah bagi pengujian karena memiliki celah yang tinggi, sehingga model percobaan pengujian secara otomatis bisa menjadi solusi untuk menghasilkan model percobaan pengujian secara otomatis pada pengujian perangkat lunak. Metode *black-box* dengan teknik pembagian identik dibutuhkan sebuah teknik kesadaran palsu.

Kata kunci : Aplikasi; *Black box*; *Testing*; kualitas; perangkat lunak

1 Pendahuluan

Pengujian pada Perangkat lunak berguna untuk memastikan semua fungsi berjalan dengan sempurna tanpa ada kesalahan pada perangkat lunak supaya tidak menyebabkan kerugian di karenakan kesalahan. Pengujian sistem bertujuan untuk memastikan bahwa semua proses sudah berfungsi sesuai dengan kebutuhan yang ditetapkan (Muslimin, et al., 2020). Oleh karena itu sangat penting untuk melaksanakan pengujian pada perangkat lunak untuk meminimalisir adanya

kesalahan yang dapat menyebabkan kerugian (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019).

Object yang akan kami uji adalah Aplikasi hitung nilai Mahasiswa menggunakan metode *Equivalence Partitions*, dalam pengujian ini kami menguji Perangkat lunak tersebut apakah fungsi perangkat lunak berjalan dengan baik dan layak untuk digunakan serta memeriksa apakah perangkat lunak tersebut terdapat kesalahan atau tidak.

Masalah yang terjadi biasanya tidak bisa Memasukan data, hasil yang tidak muncul ketika sudah memasukan data dan ada juga perangkat lunak yang eror ketika program dijalankan. Demikian perangkat lunak yang terdapat masalah tersebut dapat menyebabkan kerugian yaitu hasil yang muncul tidak sesuai dengan harapan dan tujuan dari pembuatan perangkat lunak. Sehingga Perangkat lunak tidak layak di gunakan untuk menyelesaikan pekerjaan. Masalah yang ada di perangkat lunak ini adalah tidak adanya *database* yang berfungsi menyimpan nilai mahasiswa yang telah dihitung. Masalah ini menyebabkan hilangnya data nilai mahasiswa tersebut saat Program telah selesai atau ditutup.

Menciptakan Perangkat lunak yang layak dan berkualitas harus memperhatikan kebutuhan dari penggunaanya, sehingga perangkat lunak mampu menyelesaikan permasalahan yang di hadapi oleh para penggunaanya. Perangkat lunak yang berkualitas pasti melalui banyak tahapan pengujian yang panjang agar kualitasnya terjamin. (Hidayat H. T., 2017). Pengujian dapat menyebabkan pengguna percaya bahwa fungsionalitas aplikasinya sudah berfungsi dengan baik dan tidak ragu untuk menggunakannya (Susanto, Biqirrosyad, Junaidi, Sudrajat, & Desyani, 2021). Pengujian sistem sangat penting untuk memberikan jaminan kualitasnya dan membuktikan bahwa fungsinya telah beroperasi dengan benar (Ijudin & Saifudin, 2020).

Pengujian perangkat lunak dari segi fungsional tidak dengan menguji model dan kode program untuk mengetahui kerja fungsi masukan dan keluaran Perangkat lunak sesuai dengan apa yang diperlukan. Pengujian ini Menggunakan metode *Black box*, pengujian black box dilakukan berdasarkan masukan dan luaran tanpa memperhatikan rincian program sehingga pengujian tidak perlu memiliki pengetahuan pemrograman (Shaleh, Prayogi, Pirdaus, Syawal, & Saifudin, 2021). Metode *Black box* merupakan metode yang paling mudah dilakukan karena hanya perlu batas bawah dan batas atas dari data yang diharapkan, banyak data yang diuji dapat dihitung melalui banyaknya *field* data entri pada *form* perangkat lunak tersebut, aturan entri yang harus di penuhi serta kasus batas atas dan bawah yang memenuhi. Dengan metode ini dapat Mendeteksi jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang *valid* (Cholifah, Yulianingsih, & Sagita, 2018).

Pengujian *Equivalence Partitions* adalah pengujian dengan memasukan kalimat berupa kombinasi huruf dan angka pada setiap *field* yang terdapat pada *form* perangkat lunak ini. Setiap *field* Masukan dilakukan pengujian melalui klasifikasi dan pengelompokan berdasarkan fungsi (Hidayat & Muttaqin, 2018).

Setiap menu masukan di isi dengan String berupa huruf dan angka yang akan dilakukan pengujian dan dikelompokan berdasarkan fungsinya berhasil atau gagal . pengujian dilakukan pada 1 *Form* karena memang Perangkat lunak ini hanya memiliki 1 *Form* yaitu *form* data mahasiswa, Aplikasi ini berhasil apabila dapat memunculkan seluruh masukan *form* pada kolom hasil.

2 Metodologi

Pengujian perangkat lunak memiliki peranan penting dalam suatu sistem informasi, dengan pengujian ini dapat diketahui Galat atau Error yang akan muncul pada perangkat lunak (Ahrizal, Miftah, Kurniawan, Zaelani, & Yulianti, 2020). Pengujian yang dilakukan pada Aplikasi Penghitung Nilai Mahasiswa dengan memasukan data acak berupa kalimat berisi kombinasi huruf dan angka untuk memastikan sistem menolak menyimpan data acak yang dimasukkan pada perangkat lunak, sehingga sistem dikatakan Layak untuk digunakan (Sinulingga, Zuhri, Mukti, Ziasyifa, & Saifudin, 2020). Pada Pengujian ini dilakukan dengan beberapa tahap. Pada tahap pertama diawali dengan mengumpulkan *Test Case* perangkat lunak yang akan diuji dengan metode *Equivalence Partitions*, kemudian Menganalisis standar *Grade* partition masukan dan keluaran. Perancangan kasus uji (test case) pada teknik *Equivalence Partitions* didasarkan pada evaluasi kelas yang menggambarkan kondisi ekuivalen masukan dari kumpulan kadaan yang valid dan tidak valid (Nugraha, Aditama, Arrofi, Ahmad, & Yulianti, 2020). Pengujian *Equivalence Partition* dilakukan dengan memecahkan atau membagi domain masukan ke dalam kelas-kelas data untuk membuat test case yang relevan untuk menguji aplikasi (Maulana, Kurniawan, Keumala, Sukma, & Saifudin, 2020). Hal ini dilakukan untuk mendapatkan data berupa dokumentasi pengujian dengan metode *Equivalence partitions* dan nilai tingkat efektifitas metode *Equivalence partitions*.

Dalam pengujian menghasilkan table *Test Case* yang berfungsi untuk memutuskan apakah

sistem berjalan dengan baik atau tidak. Berikut adalah penjelasan dari tabel *Test Case*.

Gambar.1 Form Data Mahasiswa

Dari gambar 1 ada beberapa macam rencana pengujian. Kolom NIM diisi dengan NIM mahasiswa, kolom Nama diisi dengan nama mahasiswa, kolom Jenis kelamin di pilih berdasarkan jens kelamin mahasiswa, kolom Jurusan dipilih berdasarkan Jurusan mahasiswa dan akan memunculkan biaya/bulan, kolom Semester diisi dengan semester mahasiswa, kolom Kehadiran diisi dengan jumlah kehadiran yang mahasiswa hadiri, kolom Nilai Tugas diisi dengan nilai tugas mahasiswa, kolom Nilai UTS

diisi dengan nilai UTS mahasiswa, kolom Nilai UAS diisi dengan nilai UAS mahasiswa, Kolom Nilai Pemrograman II dan *Grade* akan muncul setelah mengisi seluruh kolom, kolom (registrasi, UTS, UAS) dicentang akan menampilkan biayanya, dan terakhir kolom kosong akan berisi hasil dari masukan yang telah dilakukan. masukan yang dapat dimasukan dalam aplikasi ini adalah *String* (angka atau huruf), berikut adalah tabelnya.

Tabel 1. Rancangan *Test Case Form Data Mahasiswa*

ID	Deskripsi	Hasil yang diharapkan
A01	Mengisi NIM dengan kalimat berupa huruf atau angka	NIM muncul dikolom hasil
A02	Mengisi Nim, dan Nama dengan kalimat berupa huruf atau angka	Nim dan nama muncul dikolom hasil
A03	Mengisi Nim, Nama dan Jenis kelamin dengan kalimat berupa huruf atau angka	Nim, Nama dan Jenis kelamin muncul dikolom hasil
A04	Mengisi Nim, Nama, Jenis Kelamin dan Jurusan dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan muncul dikolom hasil
A05	Mengisi Nim, Nama, Jenis kelamin, Jurusan dan Semester dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan dan semester muncul dikolom hasil
A06	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester dan Nilai Kehadiran dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, , semester dan Nilai Kehadiran muncul dikolom hasil

ID	Deskripsi	Hasil yang diharapkan
A07	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, dan Nilai Tugas dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas muncul dikolom hasil
A08	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas, dan Nilai UTS muncul dikolom hasil
A09	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS dan Nilai UAS dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, dan Nilai UAS muncul dikolom hasil
A10	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka dan <i>men-ceklist</i> Registrasi	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Total biaya muncul dikolom hasil
A11	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka <i>men-ceklist</i> Registrasi dan UTS	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Total biaya muncul dikolom hasil
A12	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka, <i>men-ceklist</i> Registrasi, UTS dan UAS	Nim, Nama, Jenis kelamin, Jurusan, semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Total biaya muncul dikolom hasil

3 Hasil dan Pembahasan

Setelah rancangan Pengujian, selanjutnya menguji langsung perangkat lunak. Pengujian dilaksanakan dengan menerapkan deskripsi pengujian. Hasilnya akan dicatat dikolom hasil pengujian. Kesimpulan dibuat berdasarkan hasil kesesuaian antara hasil yang diharapkan dengan hasil pengujian. Jika hasil pengujian sesuai

dengan hasil yang diharapkan maka kesimpulannya berhasil, jika sebaliknya maka kesimpulan Gagal. Hasil pengujian yang telah dilakukan akan ditunjukkan pada tabel 2 Hasil pengujian. *Input* yang dapat dimasukkan dalam aplikasi ini adalah String (angka atau huruf), berikut adalah tabelnya.

Tabel 2. Hasil Pengujian

ID	Deskripsi	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A01	Mengisi NIM dengan kalimat berupa huruf atau angka	NIM muncul dikolom hasil	Menampilkan <i>warning</i> Nama anda belum diisi!	Tidak berhasil
A02	Mengisi Nim, dan Nama dengan kalimat berupa huruf atau angka	Nim dan nama muncul dikolom hasil	Menampilkan <i>warning</i> jenis kelamin anda belum dipilih	Tidak berhasil
A03	Mengisi Nim, Nama dan Jenis kelamin dengan kalimat berupa huruf atau angka	Nim, Nama dan Jenis kelamin muncul dikolom hasil	Menampilkan <i>Warning</i> jurusan anda belum dipilih	Tidak berhasil
A04	Mengisi Nim, Nama, Jenis Kelamin dan Jurusan dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan muncul dikolom hasil	Menampilkan <i>warning</i> semester anda belum diisi	Tidak berhasil
A05	Mengisi Nim, Nama, Jenis kelamin, Jurusan dan Semester dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, dan semester muncul dikolom hasil	Menampilkan <i>warning</i> Nilai kehadiran belum diisi	Tidak berhasil
A06	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester dan Nilai Kehadiran dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, semester dan Nilai Kehadiran muncul dikolom hasil	Menampilkan <i>warning</i> Nilai Tugas belum diisi	Tidak berhasil

ID	Deskripsi	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A07	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, dan Nilai Tugas dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan , semester,Nilai Kehadiran, Nilai Tugas muncul dikolom hasil	Menampilkan <i>Warning</i> Nilai UTS belum diisi	Tidak berhasil
A08	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan, semester,Nilai Kehadiran, Nilai Tugas, dan Nilai UTS muncul dikolom hasil	Menampilkan <i>warning</i> Nilai UAS belum diisi	Tidak berhasil
A09	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS dan Nilai UAS dengan kalimat berupa huruf atau angka	Nim, Nama, Jenis kelamin, Jurusan , semester,Nilai Kehadiran, Nilai Tugas, Nilai UTS, dan Nilai UAS serta memunculkan Nilai Pemrograman II dan <i>Grade</i> dikolom hasil	Berhasil Memunculkan Seluruh <i>Form</i> yang diisi serta memunculkan Nilai pemrograman dan <i>Grade</i> tetapi total biaya berisi nol	Berhasil
A10	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka, dan Men- <i>ceklist</i> Registrasi	Nim, Nama, Jenis kelamin, Jurusan , semester,Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Nilai pemrograman, <i>grade</i> dan total biaya muncul dikolom hasil	Berhasil Memunculkan Seluruh <i>Form</i> yang diisi serta memunculkan Nilai pemrograman <i>Grade</i> , dan total biaya tetapi hanya biaya registrasi yang terhitung	Berhasil
A11	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka, Men- <i>ceklist</i> Registrasi dan UTS	Nim, Nama, Jenis kelamin, Jurusan , semester,Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Nilai pemrograman, <i>grade</i> dan total biaya muncul dikolom hasil	Berhasil Memunculkan Seluruh <i>Form</i> yang diisi serta memunculkan Nilai pemrograman <i>Grade</i> , dan total biaya tetapi hanya biaya registrasi dan UTS yang terhitung	Berhasil
A12	Mengisi Nim, Nama, Jenis kelamin, Jurusan, Semester, Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS dengan kalimat berupa huruf atau angka, dan Men- <i>ceklist</i> Registrasi	Nim, Nama, Jenis kelamin, Jurusan , semester,Nilai Kehadiran, Nilai Tugas, Nilai UTS, Nilai UAS, Nilai pemrograman, <i>grade</i> dan total biaya muncul dikolom hasil	Berhasil Memunculkan Seluruh <i>Form</i> yang diisi serta memunculkan Nilai pemrograman <i>Grade</i> , dan total biaya terhitung program telah berjalan sempurna	Berhasil

4 Kesimpulan

Berdasarkan pengujian yang telah kami lakukan, dapat kami simpulkan pengujian Perangkat lunak dengan metode *Black box* fokus pada apa yang kita input dan menghasilkan output yang Memberitahukan kualitas Perangkat lunak yang dikembangkan dengan spesifikasi sistem yang ditentukan. Teknik *Equivalence* partitions dapat membantu proses pembuatan kasus pengujian dan ketentuan kualitas serta menemukan masalah yang ada serta menjamin perangkat lunak yang diuji sesuai dengan fungsi

yang diinginkan. Pembuatan perangkat lunak ini mempunyai tujuan untuk memudahkan mengumpulkan data dari masing-masing mahasiswa sehingga data tersusun dan tersimpan dengan baik.

5 Saran

Berdasarkan pengujian dan hasil pengujian Aplikasi Sistem perhitungan mahasiswa masih perlu untuk dikembangkan. Adapun saran yang kami anjurkan yaitu pengujian tidak hanya dengan metode *Black Box* agar tidak hanya

masuk dan pengeluaran melainkan sistem didalam *code* yang dirancang dan terstruktur dapat diuji, untuk Perangkat lunak yang dibuat ini pun masih ada kekurangan yaitu data yang telah dihitung tidak tersimpan dalam database jadi Aplikasi ini hanya untuk menghitung tidak untuk

disimpan datanya. Kami harap pembuat aplikasi dapat mengembangkan perangkat lunak ini lebih banyak fungsi lagi seperti menggunakan *database* agar data yang sudah dihitung dapat tersimpan.

References

- Ahrizal, D., Miftah, M. K., Kurniawan, R., Zaelani, T., & Yulianti, Y. (2020). Pengujian Perangkat Lunak Sistem Informasi Peminjaman PlayStation dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. *Jurnal Informatika Universitas Pamulang*, 5(1), 73-77. doi:10.32493/informatika.v5i1.4338
- Cholifah, W. N., Yulianingsih, Y., & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(2), 206-210. doi:10.30998/string.v3i2.3048
- Hidayat, H. T. (2017). Pengujian kualitas kelayakan perangkat lunak dengan penerapan perancangan model Rapid Application Development. *Elinvo (Electronics, Informatics, and Vocational Education)*, 2(2), 121-129. doi:10.21831/elinvo.v2i2.17304
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jutis (Jurnal Teknik Informatika)*, 6(1), 25-29. doi:10.33592/jutis.Vol6.Iss1.38
- 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
- Maulana, A., Kurniawan, A., Keumala, W., Sukma, V. R., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalent Partitions (Studi Kasus: PT Arap Store). *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(1), 50-56. doi:10.32493/jtsi.v3i1.4307
- Muslimin, D. B., Kusmanto, D., Amilia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-25. doi:10.32493/informatika.v5i1.3778
- 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.
- Nugraha, B. F., Aditama, F., Arrofi, M., Ahmad, S. U., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Penghitungan Parkir Swalayan ADA Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 5(2), 146-151. doi:10.32493/informatika.v5i2.5350
- Shaleh, I. A., Prayogi, J., 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:10.32493/jtsi.v4i1.8960
- Sinulingga, A. R., Zuhri, M., Mukti, R. B., Ziasyifa, & Saifudin, A. (2020). *Pengujian Black box pada sistem aplikasi informasi data kinerja menggunakan teknik Equivalence Partitions*, 2.
- Susanto, J., Biqirrosyad, B., Junaidi, M. M., Sudrajat, Y., & Desyani, T. (2021). Pengujian Black Box pada Aplikasi Desktop Penjualan Elektronik Menggunakan Metode Equivalence Partitioning. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 4(1), 38-45. doi:10.32493/jtsi.v4i1.8960