

## Pengujian *Black Box* pada Aplikasi Perpustakaan Menggunakan Teknik *Equivalence Partitions*

Bayu Aji Priyaungga<sup>1</sup>, Dwi Bayu Aji<sup>2</sup>, Mukron Syahroni<sup>3</sup>, Nurul Tri Sukma Aji<sup>4</sup>, Aries Saifudin<sup>5</sup>

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

e-mail: <sup>1</sup>Bayuuups98@gmail.com, <sup>2</sup>dwiangin@gmail.com, <sup>3</sup>msyahroni@gmail.com, <sup>4</sup>Nurultrisukmaaji@gmail.com, <sup>5</sup>aries.saifudin@unpam.ac.id

Submitted Date: June 12<sup>th</sup>, 2020  
Revised Date: August 01<sup>st</sup>, 2020

Reviewed Date: July 17<sup>th</sup>, 2020  
Accepted Date: August 02<sup>nd</sup>, 2020

### Abstract

*The library application is used to help manage and document transaction activities in the library which include borrowing and repaying. Library applications must be free of errors, because if they contain errors can cause harm to providers, managers, or members of the library. To provide a guarantee that the library application is free from errors, testing needs to be done. Software testing is an activity aimed at finding and finding errors and bugs in an application, which aims to minimize the losses that occur due to system errors. In this study, the proposed library application testing is Black Box. Black Box Testing is a test based on the appearance (interface) and functions of the software itself and not from the source code of the program. Black Box Testing has several methods, one of which is Equivalence Partitions, the method we use for testing the software. Equivalence Partitions are methods that discuss valid or not entered into the software, and observe the accuracy of the input. So that the side of the error is known. The test results have proven that the library application that has been developed is error free and meets all the requirements set.*

*Keywords: Black Box Testing; Equivalence Partitions; Library Applications*

### Abstrak

Aplikasi perpustakaan digunakan untuk membantu mengelola dan mendokumentasikan aktifitas transaksi di perpustakaan yang mencakup peminjaman dan pengembalian. Aplikasi perpustakaan harus bebas dari kesalahan, karena jika mengandung kesalahan dapat menyebabkan kerugian bagi penyedia, pengelola, atau anggota perpustakaan. Untuk memberikan jaminan bahwa aplikasi perpustakaan telah bebas dari kesalahan, maka perlu dilakukan pengujian. Pengujian perangkat lunak merupakan kegiatan bertujuan untuk mencari dan menemukan kesalahan dan *bugs* pada sebuah aplikasi, yang bertujuan untuk meminimalisir kerugian yang terjadi akibat kesalahan sistem. Pada penelitian ini diusulkan pengujian aplikasi perpustakaan secara *Black Box*. Pengujian secara *Black Box* merupakan pengujian berdasarkan dari tampilan (*interface*) dan fungsi-fungsi dari perangkat lunak itu sendiri serta bukan dari *source code* program tersebut. Pengujian *Black Box* memiliki beberapa metode salah satunya *Equivalence Partitions*, metode yang kami gunakan untuk pengujian perangkat lunak tersebut. *Equivalence Partitions* adalah metode yang membahas tentang valid atau tidak masukkan yang dimasukkan pada perangkat lunak, serta mengamati akurasi dari masukan tersebut. Sehingga diketahui sisi kesalahannya. Hasil pengujian telah membuktikan bahwa aplikasi perpustakaan yang telah dikembangkan bebas dari kesalahan dan sudah memenuhi semua persyaratan (*requirement*) yang ditetapkan.

Kata kunci: *Black Box Testing; Equivalence Partitions; Aplikasi Perpustakaan*

### 1 Pendahuluan

Pengujian terhadap perangkat lunak sangat penting dilakukan dengan tujuan untuk

memberikan jaminan kualitas perangkat lunak yang dihasilkan agar bebas dari terjadinya kesalahan (Debiyanti, Sutrisna, Budrio, Kamal, &

Yulianti, 2020). Pengujian perangkat lunak adalah suatu proses resmi yang sudah ditetapkan oleh kelompok penguji yang mencangkup item perangkat lunak, beberapa item perangkat lunak yang terhubung atau semua paket perangkat lunak yang diperiksa secara menyeluruh dengan mengoperasikan program di komputer (Utomo, Kurniawan, & Astuti, 2018). Pengujian perangkat lunak adalah prosedur mengoperasikan sebuah program dengan tujuan menemukan penyimpangan-penyimpangan error pada program tersebut. Prosedur untuk mengoperasikan program komputer serta membandingkan perilaku yang sebenarnya dengan yang diharapkan sehingga bisa menghasilkan produk yang bermutu tinggi.

Telah banyak alat bantu yang dikomputerisasi dalam kehidupan sehari-hari. Komputer telah mendominasi di berbagai bidang kerja atau aktifitas untuk meningkatkan kemudahan, efektifitas dan efisiensi sehingga memiliki peranan yang sangat penting (Taufiq, Magfiroh, Yusup, & Yulianti, 2020). Aplikasi perpustakaan merupakan salah satu contoh penerapan komputerisasi di sekolah untuk efisiensi dan mempercepat pengelolaan data (Salsabilah & Yulianti, 2019).

Software yang akan diuji adalah sebuah aplikasi perpustakaan. Aplikasi yang dapat digunakan untuk membantu petugas perpustakaan dalam mengatur peminjaman buku data buka dan menyimpan catatannya. Jika terjadi masalah pada software/program atau program error maka akan terjadi kerugian. Bagi petugas perpustakaan, jika aplikasi ini terjadi program error maka akan mengakibatkan hilangnya catatan peminjaman dan berdampak pada jumlah buku yang ada.

Metode yang kami gunakan untuk melakukan pengujian pada aplikasi perpustakaan adalah metode pengujian *Black Box*. Pengujian fungsional atau pengujian *Black Box* adalah strategi pengujian yang pengujiannya diturunkan dari rincian program atau item. Sistem adalah 'kotak hitam' yang perilakunya semata-mata bisa ditetapkan dengan menganalisa masukan dan keluaran yang berhubungan. Cara ini disebut dengan pengujian fungsional karena penguji tidak menerapkan perangkat lunak tapi hanya berkepentingan dengan fungsionalitasnya (MZ, 2016).

Rencana penelitian/pengujian pada aplikasi perpustakaan akan menggunakan metode pengujian *Black Box* dengan teknik Equivalence Partitions.

## 2 Metodologi

Pengujian terhadap perangkat lunak sangat penting dilakukan dengan tujuan untuk memberikan jaminan kualitas perangkat lunak yang dihasilkan agar bebas dari terjadinya kesalahan (Debiyanti, Sutrisna, Budrio, Kamal, & Yulianti, 2020). Dalam pengujian ini, teknik pengujian *Black Box* yang kami gunakan untuk melakukan pengujian Aplikasi Perpustakaan.

Ada beberapa macam pengujian perangkat lunak yaitu *White Box* dan *Black Box*. Pengujian *White Box* dan *Black Box* memiliki tujuan dan cara yang berbeda, sehingga keduanya saling melengkapi.

### 2.1 Pengujian *White Box*

Pengujian *White Box* ialah pengujian yang berfokus terhadap prosedur serta logika kode dari aplikasi yang dibuat. Pada pengujian *White Box*, penguji bertugas untuk menganalisa dan menemukan *error* pada kode program (Siagian, 2018).

Pengujian *White Box* mencakup analisa jalur dasar dengan membuat flow graph dari kode program dan menentukan nilai kompleksitas siklomatik untuk menentukan jumlah pengujian (Pratala, Asyer, Prayudi, & Saifudin, 2020). Kemudian dilanjutkan dengan membuat uji kasus (*test case*), membuat *source code* untuk *test unit*, dan mengeksekusi *test unit* yang telah dibuat, serta mengevaluasi hasilnya.

Pengujian *White Box* dilakukan mengikuti tahapan berikut ini:

- Menganalisa kode program untuk mengidentifikasi jalur eksekusi dengan membuat *flow graph*, menghitung kompleksitas siklomatik, dan mengidentifikasi jalur.
- Membuat *test case* untuk pencarian *error* pada kode program yang dibuat oleh orang yang mengerti program atau biasa disebut programmer.
- Membuat kode program pengujian unit (*unit test*)

### 2.2 Pengujian *Black Box*

Pengujian *Black Box* adalah pengujian yang memverifikasi hasil eksekusi aplikasi berdasarkan masukan yang diberikan (data uji) untuk memastikan fungsional dari aplikasi sudah sesuai dengan persyaratan (*requirement*) (Febrian, Ramadhan, Faisal, & Saifudin, 2020). Pengujian *Black Box* ialah pengujian yang berfokus pada interface atau tampilan dan pengujian fungsional

yang terdapat pada aplikasi, serta kesesuaian pada alur fungsi yang dibutuhkan oleh *user*. Pengujian *Black Box* tidak menguji berdasarkan *source code* program (Siagian, 2018).

Pengujian *Black Box* dilakukan mengikuti tahapan berikut ini:

- Membuat *test case* untuk pengujian fungsi-fungsi yang terdapat di aplikasi
- Membuat *test case* untuk pengujian kesesuaian *flow* atau alur dari kerja suatu fungsi pada program cocok dengan apa yang dibutuhkan dan permintaan dari pengguna
- Mencari *bugs/error* berdasarkan tampilan (*interface*) pada aplikasi

Dalam melakukan pengujian harus memilih teknik yang tepat, yaitu teknik yang dapat menemukan kesalahan yang belum terdeteksi sehingga dapat meningkatkan kualitas software (Hendri, Manurung, Ferian, Hanaatmoko, & Yulianti, 2020). Pada penelitian ini digunakan metode *Equivalence Partitions* untuk melakukan pengujian. *Equivalence Partitions* merupakan pengujian yang didasarkan pada masukan data di setiap form (Maturidi, 2014). Metode pengujian *Equivalence Partitions* membagi domain masukan dari program ke dalam masing-masing kelas data. *Equivalence Partitioning* merupakan sebuah pengujian berdasarkan memasukkan data pada setiap form yang memecah domain masukan ke dalam kelompok valid dan tidak valid (Aziz, Setiawan, Khanh, Nurdiansyah, & Yulianti, 2020).

Pengujian ini menggunakan beberapa tahapan, tahapan yang pertama dimulai dengan membuat uji kasus (*test case*) dari perangkat lunak yang akan diuji. Selanjutnya mengeksekusi setiap uji kasus yang dibuat dan mendokumentasikan pengujian untuk menganalisa keefektifannya (Jaya,

Gumilang, Tresnawati, Andersen, & Desyani, 2019). Dari pengujian yang dilakukan dapat dilihat pada tabel *test case* dibuat sebagai acuan berhasil tidaknya pengujian tersebut (Arochman, Darmawan, & Santoso, 2016).

Kode Buku	Judul Buku	Tanggal Terbit	Pengarang
1234	Teknik Informatika	12-12-2015	Jimmy Putra, S.Kom, M.k

Gambar 1 Form Data Buku

Berdasarkan form pada Gambar 1, terdapat beberapa rencana pengujian, rencana pengujian KODE BUKU, data akan valid jika diisi dengan memasukkan angka (contohnya "123") dan tidak boleh kosong. Sebaliknya data tidak valid jika diisi dengan huruf (contohnya "abc"). Kolom JUDUL BUKU diisi dengan memasukkan judul buku dan harus memasukkan huruf dan tidak boleh kosong. Kolom TANGGAL TERBIT akan valid jika diisi dengan format tanggal dd-mm-yyyy dan tidak boleh kosong. Kolom NAMA PENGARANG diisi dengan nama pengarang dengan memasukkan huruf dan tidak boleh kosong.

Tabel 1 Rancangan Test Case Form Data Buku

ID	Deskripsi Pengujian	Hasil yang Diharapkan
A01	Mengisi KODE BUKU dengan "1234567890", mengisi JUDUL BUKU dengan "30 Hari Mahir Membuat Software dengan Java", mengisi TANGGAL TERBIT dengan "20-01-2020", mengisi NAMA PENGARANG dengan "Nurul Aji S.Kom, M.Kom" kemudian klik tombol TAMBAH	Data buku dengan kode buku 1234567890 akan tersimpan ke database
A02	Mengisi KODE BUKU dengan "BBB" atau tidak diisi, mengisi JUDUL BUKU dengan "123123" atau tidak diisi, mengisi TANGGAL TERBIT dengan "2020-01-20" atau tidak diisi, mengisi NAMA PENGARANG dengan "123456" atau tidak diisi, kemudian klik tombol TAMBAH	Data dengan kode buku BBB gagal disimpan ke database dan akan tampil pesan kode buku harus berupa angka

No Bp	Nama	Tempat Lahir	Tanggal Lahir	Jurusan	Tanggal Ma...
20133	Risma Fitri	Jakarta	12-09-1997	Teknik Infor...	20-11-2015

Pada Gambar 2 terdapat beberapa rencana pengujian, kolom NO BP akan valid jika diisi dengan angka (contoh "1234") dan tidak akan valid jika diisi dengan huruf (contoh "bbb") dan tidak boleh kosong. Kolom NAMA akan valid jika diisi dengan huruf dan tidak akan valid jika diisi dengan angka dan tidak boleh kosong. Kolom TEMPAT LAHIR akan valid jika diisi dengan huruf dan tidak valid jika diisi dengan angka dan tidak boleh kosong. Kolom TANGGAL LAHIR akan valid jika diisi dengan format tanggal DD-MM-YYYY dan tidak boleh kosong. Kolom JURUSAN akan valid jika diisi dengan huruf dan tidak akan valid jika diisi dengan angka dan tidak boleh kosong. Kolom TANGGAL MASUK akan valid jika diisi dengan format tanggal DD-MM-YYYY dan tidak boleh kosong.

Gambar 2 Form Data Mahasiswa

Tabel 2 Rancangan Test Case Form Data Mahasiswa

ID	Deskripsi Pengujian	Hasil yang Diharapkan
B01	Mengisi NO BP dengan "123456", mengisi NAMA dengan "Ahmad", mengisi TEMPAT LAHIR dengan "JAKARTA", mengisi TANGGAL LAHIR dengan "18-10-1990", mengisi JURUSAN dengan "Teknik Informatika", mengisi TANGGAL MASUK dengan "12-03-2019"	Data mahasiswa dengan no BP 123456 akan tersimpan ke dalam database
B02	Mengisi NO BP dengan "AAAA" atau tidak diisi, mengisi NAMA dengan "123" atau tidak diisi, mengisi TEMPAT LAHIR dengan "123" atau tidak diisi, mengisi TANGGAL LAHIR dengan "18-10-1990" atau tidak diisi, mengisi JURUSAN dengan "123" atau tidak diisi, mengisi TANGGAL MASUK dengan "12-03-2019" atau tidak diisi	Data mahasiswa dengan no BP AAAA gagal disimpan ke dalam database dan menampilkan pesan no BP harus angka

Gambar 3 Form Data Peminjaman

Pada gambar 3 terdapat beberapa rencana pengujian, kolom NO BP akan valid jika diisi dengan angka dan tidak akan valid jika diisi dengan huruf dan tidak boleh kosong. Kolom NAMA akan valid jika diisi dengan huruf dan tidak akan valid jika diisi dengan angka dan tidak boleh kosong. Kolom JURUSAN akan valid jika diisi dengan huruf dan tidak akan valid jika diisi dengan angka dan tidak boleh kosong. Kolom KODE BUKU akan valid jika diisi dengan angka dan tidak akan

valid jika diisi dengan huruf dan tidak boleh kosong. Kolom JUDUL BUKU akan valid jika diisi dengan huruf dan tidak akan valid jika diisi dengan angka dan tidak boleh kosong. Kolom TANGGAL TERBIT akan valid jika diisi dengan format tanggal DD-MM-YYYY dan tidak boleh kosong. Kolom TANGGAL PINJAM akan valid jika diisi dengan format tanggal DD-MM-YYYY dan tidak boleh kosong.

Tabel 3 Rancangan Test Case Form Data Peminjaman

ID	Deskripsi Pengujian	Hasil yang Diharapkan
C01	Mengisi Data Mahasiswa, mengisi NO BP dengan "123456", mengisi NAMA dengan "Ahmad", mengisi JURUSAN dengan "Teknik" Mengisi Data Buku, mengisi KODE BUKU dengan "1234", Mengisi JUDUL BUKU dengan "AAA", mengisi TANGGAL TERBIT DENGAN "12-02-2020", mengisi TANGGAL PINJAM dengan "01-03-2020"	Data peminjaman dengan nomor BP 123456 dan kode buku 1234 akan tersimpan ke database
C02	Mengisi Data Mahasiswa, mengisi NO BP dengan "AAAA" atau tidak diisi, mengisi NAMA dengan "123" atau tidak diisi, mengisi JURUSAN dengan "5555" atau tidak diisi Mengisi Data Buku, mengisi KODE BUKU dengan "BBB" atau tidak diisi, Mengisi JUDUL BUKU dengan "777" atau tidak diisi, mengisi TANGGAL TERBIT DENGAN "2020-02-01" atau tidak diisi, mengisi TANGGAL PINJAM dengan "2020-03-01" atau tidak diisi	Data peminjaman dengan nomor BP AAA dan kode buku BBB gagal tersimpan ke database

### 3 Hasil dan Pembahasan

Rancangan pengujian sudah dibuat, lalu dilakukan pengujian. Hasil dari pengujian kemudian dicatat di dalam kolom pengujian.

Setelah itu dibuat sebuah kesimpulan yang berdasarkan penilaian antara keluaran yang diharapkan dengan keluaran hasil eksekusi aplikasi. Apabila keluaran yang diharapkan sama

dengan keluaran yang dihasilkan maka pengujian dianggap berhasil. Hasil pengujian dapat dilihat pada Tabel 4.

Tabel 4 Hasil Pengujian

ID	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
A01	Mengisi KODE BUKU dengan "1234567890" , mengisi JUDUL BUKU dengan "30 Hari Mahir Membuat Software dengan Java", mengisi TANGGAL TERBIT dengan "20-01-2020" , mengisi NAMA PENGARANG dengan "Nurul Aji S.Kom, M.Kom" kemudian klik tombol TAMBAH	Data buku dengan kode buku 1234567890 akan tersimpan ke database	Data buku dengan kode buku 1234567890 tersimpan ke database	Sesuai
A02	Mengisi KODE BUKU dengan "BBB" atau tidak diisi , mengisi JUDUL BUKU dengan "123123" atau tidak diisi, mengisi TANGGAL TERBIT dengan "2020-01-20" atau tidak diisi, mengisi NAMA PENGARANG dengan "123456" atai tidak diisi, kemudian klik tombol TAMBAH	Data dengan kode buku BBB gagal disimpan ke database dan akan tampil pesan kode buku harus berupa angka	Data buku dengan kode buku BBB gagal tersimpan ke database dan keluar pesan kode buku harus berupa angka	Sesuai
B01	Mengisi NO BP dengan "123456", mengisi NAMA dengan "Ahmad", mengisi TEMPAT LAHIR dengan "JAKARTA", mengisi TANGGAL LAHIR dengan "18-10-1990", mengisi JURUSAN dengan "Teknik Informatika", mengisi TANGGAL MASUK dengan "12-03-2019"	Data mahasiswa dengan no BP 123456 akan tersimpan ke dalam database	Data mahasiswa dengan no BP 123456 tersimpan ke database	Sesuai
B02	Mengisi NO BP dengan "AAAA" atau tidak diisi , mengisi NAMA dengan "123" atau tidak diisi, mengisi TEMPAT LAHIR dengan "123" atau tidak diisi, mengisi TANGGAL LAHIR dengan "18-10-1990" atau tidak diisi, mengisi JURUSAN dengan "123" atau tidak diisi, mengisi TANGGAL MASUK dengan "12-03-2019" atau tidak diisi	Data mahasiswa dengan no BP AAAA gagal disimpan ke dalam database dan menampilkan pesan no BP harus angka	Data mahasiswa dengan no BP AAAA gagal tersimpan ke database dan keluar pesan no BP harus berupa angka	Sesuai
C01	Mengisi Data Mahasiswa, mengisi NO BP dengan "123456", mengisi NAMA dengan "Ahmad", mengisi JURUSAN dengan "Teknik" Mengisi Data Buku, mengisi KODE BUKU dengan "1234", Mengisi JUDUL BUKU dengan "AAA", mengisi TANGGAL TERBIT DENGAN "12-02-2020", mengisi TANGGAL PINJAM dengan "01-03-2020"	Data peminjaman dengan nomor BP 123456 dan kode buku 1234 akan tersimpan ke database	Data peminjaman dengan nomor BP 123456 dan kode buku 1234 tersimpan ke database	Sesuai
C02	Mengisi Data Mahasiswa, mengisi NO BP dengan "AAAA" atau tidak diisi, mengisi	Data peminjaman	Data peminjaman	Sesuai

	NAMA dengan "123" atau tidak diisi, mengisi JURUSAN dengan "5555" atau tidak diisi Mengisi Data Buku, mengisi KODE BUKU dengan "BBB" atau tidak diisi, Mengisi JUDUL BUKU dengan "777" atau tidak diisi, mengisi TANGGAL TERBIT DENGAN "2020-02-01" atau tidak diisi, mengisi TANGGAL PINJAM dengan "2020-03-01" atau tidak diisi	dengan nomor BP AAA dan kode buku BBB gagal tersimpan ke database	dengan nomor BP AAA dan kode buku BBB gagal tersimpan ke database dan menampilkan pesan no BP dan kode buku harus berupa angka
--	---	---	--

Pengujian pada aplikasi perpustakaan dilakukan pengujian pada 3 form, yaitu Form Data Buku diuji sebanyak 2 kali, Form Data Mahasiswa diuji sebanyak 2 kali, Form Data Peminjaman diuji sebanyak 2 kali. Total keseluruhan pengujian ada 6 kali pengujian.

#### 4 Kesimpulan

Kesimpulan Pengujian aplikasi dengan *Black Box Testing* bertujuan melihat program tersebut sama dengan tugasnya tanpa mengetahui kode program yang dipakai. Berdasarkan pengujian kualitas aplikasi Perpustakaan, dapat ditarik kesimpulan bahwa pengujian dengan metode *Black Box* berbasis *Equivalence Partitions* dapat membantu proses pembuatan *test case* pengujian, uji kualitas dan menemukan kesalahan yang tidak terdeteksi yang disebabkan oleh kesalahan pengetikan. Dalam pengujian aplikasi Perpustakaan ditemukan kesalahan dalam mengedit data (di mana jika kita salah dalam menggunakan tipe data seperti pada form [edit data buku] kode buku hanya dapat diisi dengan menggunakan Angka, jika kita menggunakan huruf/symbol maka akan terdapat ketidaksesuaian pada tipe data dan hal tersebut membuat data yang sudah diisi akan Gagal Tersimpan. Tetapi jika kita memasukan sesuai dengan tipe data (angka) maka data tersebut akan berhasil tersimpan di dalam data base. Pengujian ini menunjukkan bahwa terdapat celah *error* pada suatu form di aplikasi perpustakaan

#### 5 Saran

Untuk ingin membuktikan kelayakan sistem lakukanlah minimal 2 kali pengujian, agar hasil dari pengujian tersebut dapat disamakan nilainya walaupun hanya ditemukan 1 kesalahan yang sama. Saran untuk pengembangan aplikasinya adalah dapat menggunakan teknik pengujian *Black Box* seperti *Boundary Value Analysis* atau *Error*

*Guessing* agar kita dapat menemukan kesalahan/error lainnya pada pengembangan aplikasi ini sehingga, kualitas aplikasi yang dihasilkan jauh lebih baik (mengingat sudah berulang kali meminimalisir kesalahan) yang nantinya jika system tersebut sudah bebas dari kesalahan maka sistem akan bekerja secara maksimal sekaligus mengoptimalkan kinerja dari user (pengguna)

#### Referensi

- Arochman, Darmawan, A. S., & Santoso, F. H. (2016). Otomatisasi Desain Test Case Pengujian Perangkat Lunak Metode Black-Box Testing Dengan Teknik Equivalence Partitioning Menggunakan Algoritma Genetika. 47-55.
- Aziz, I. A., Setiawan, B., Khanh, R., Nurdiyansyah, G., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Kasir Berbasis Website Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(2), 10.32493/jtsi.v3i2.4693. doi:10.32493/jtsi.v3i2.4693
- Debiyanti, D., Sutrisna, S., Budrio, B., Kamal, A. K., & Yulianti, Y. (2020). Pengujian Black Box pada Perangkat Lunak Sistem Penilaian Mahasiswa Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 5(2), 162-166. doi:10.32493/informatika.v5i2.5446
- Febrian, V., Ramadhan, M. R., Faisal, M., & Saifudin, A. (2020). Pengujian pada Aplikasi Penggajian Pegawai dengan menggunakan Metode Blackbox. *Jurnal Informatika Universitas Pamulang*, 5(1), 61-66. doi:10.32493/informatika.v5i1.4340
- Hendri, H., Manurung, J. W., Ferian, R. A., Hanaatmoko, W. F., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Pengelolaan Masjid Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(2), 107-113. doi:10.32493/jtsi.v3i2.4694

- Jaya, M. S., Gumilang, P., Tresnawati, Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *4*, 131-136.
- Maturidi, A. J. (2014). *Metode Penelitian Teknik Informatika*. Yogyakarta: deepublish.
- MZ, M. K. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*, 2-8.
- Pratala, C. T., Asyer, E. M., Prayudi, I., & Saifudin, A. (2020). Pengujian White Box pada Aplikasi Cash Flow Berbasis Android Menggunakan Teknik Basis Path. *Jurnal Informatika Universitas Pamulang*, 5(2), 111-119. doi:informatika.v5i2.4713
- Salsabilah, Z., & Yulianti, Y. (2019). Perancangan Aplikasi Perpustakaan Berbasis Web pada SMK Negeri 1 Rongkasbitung. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 2(1), 9-14. doi:10.32493/jtsi.v2i1.2401
- Siagian, L. J. (2018). *Otomatisasi Pengujian Perangkat Lunak: Software Test Automation*. Yogyakarta: deepublish.
- Taufiq, R., Magfiroh, D. A., Yusup, D., & Yulianti, Y. (2020). Analisis dan Desain Sistem Informasi Pembayaran Sumbangan Pembinaan Pendidikan (SPP) di SMK Avicena Rajeg. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(1), 15-21. doi:10.32493/jtsi.v3i1.4308
- Utomo, D. W., Kurniawan, D., & Astuti, Y. P. (2018). Teknik Pengujian Perangkat Lunak Dalam Evaluasi Sistem Layanan Mandiri Pemantauan Haji Pada Kementerian Agama Provinsi Jawa Tengah. *Simetris : Jurnal Teknik Mesin, Elektro dan Ilmu Komputer*, 9(2), 731-746. doi:10.24176/simet.v9i2