

## PENGUJIAN FUNGSIONAL PERANGKAT LUNAK SISTEM INFORMASI PERPUSTAKAAN DENGAN METODE BLACK BOX TESTING BAGI PEMULA

Uminingsih, Muhamad Nur Ichsanudin, Muhammad Yusuf, Suraya

Jurusan Rekayasa Sistem Komputer<sup>(1,2,4)</sup>, Jurusan Teknik Industri<sup>(3)</sup>, IST AKPRIND Yogyakarta

Email: umy\_bin@yahoo.com

---

### INFO ARTIKEL

---

#### Riwayat Artikel :

Diterima : 15 Mei 2022

Disetujui : 19 Mei 2022

---

#### Kata Kunci :

Black Box Testing, Equivalen  
partition, Sistem Informasi  
Perpustakaan.

---

### ABSTRAK

Sistem Informasi yang dibuat sebelum di gunakan secara formal harus melewati pengujian terhadap perangkat lunaknya terlebih dahulu. Banyak metode pengujian yaitu white Box Testing dan Black Box Testing. Disini di pilih Black Box Testing karena untuk pemula yaitu yang tidak mengharuskan menguasai Bahasa pemrograman tertentu .Teknik pengujian pada Black Box Testing ada beberapa cara antara lain Equivalen Partitions Testing, Boundary Value Analysis Testing, Comparison Testing, Sample Testing, Robustness Testing dan lainnya. Disini dipilih menggunakan Teknik Equivalence Partitions yang hanya untuk menguji apakah ada kesalahan pada: fungsi sistem, Interface, struktur data atau akses data, performansi , inisialisasi. Obyek yang dilakukan pengujian adalah Sistem Informasi Perpustakaan melalui web. Jumlah form yang diuji sebanyak 3 buah yaitu Form Inisialisasi , Form Peminjaman dan Form Pengembalian ,yang mana masing-masing form diuji dengan memasukkan beberapa data input yang sering terjadi(ditemukan). Hasil penelitian menunjukkan bahwa pengujian perangkat lunak menggunakan metode Black Box Testing dengan Teknik Equivalence Partitions pada system informasi perpustakaan masih ada form yang hasil outputnya belum sesuai dengan yang diharapkan dalam perancangan karena terjadi error. Dari hasil perhitungan validitas sistem diperoleh nilai 75% .Kemungkinan error yang terjadi dikarenakan adanya kesalahan pada baris kode (syntax). Untuk itu perlu dilakukan update pada syntax untuk memperbaiki bug tersebut.

---

### ARTICLE INFO

---

#### Article History :

Received : May 15, 2022

Accepted : May 19, 2022

---

#### Keywords:

Black Box Testing, Equivalent  
Partition, Library Information  
System

---

### ABSTRACT

*Information systems which has been made formally must pass testing on the software first before it is used. There are many testing methods, namely white box testing and black box testing. Here, Black Box Testing is chosen because it is for beginners that does not require mastering certain programming languages. There are several ways to test the Black Box Testing technique, including Equivalent Partitions Testing, Boundary Value Analysis Testing, Comparison Testing, Sample Testing, Robustness Testing and others. Here selected using the Equivalence Partitions technique which is only to test whether there are errors in: system functions, interfaces, data structures or data access, performance, initialization. The object being tested is the library information system via the web. There are 3 forms tested, namely Initialization Form, Loan Form and Return Form, where each form is tested by entering some input data that*

*often occurs (found). The results showed that software testing using the Black Box Testing method with the Equivalence Partitions technique in the library information system still has a form whose output is not as expected in the design because an error occurred. From the results of the calculation of the validity of the system obtained a value of 75%. Possible errors that occur due to errors in the line of code (syntax). So, it is necessary to update the syntax to fix the bug.*

---

## 1. PENDAHULUAN

Dalam setiap melakukan perancangan system aplikasi, maka tahap terakhir sebelum dipublikasikan ke masyarakat pengguna, tentunya harus dilakukan pengecekan/testing pada software yang dibuat. *Software Testing* merupakan sebuah metode untuk mengetahui apakah *software* yang dibuat, berfungsi dengan baik dan benar. Tanpa adanya *software testing* kita tidak dapat mengetahui apakah sebuah software sudah memenuhi kriteria yang dibutuhkan untuk user.

Ada beberapa metode pengujian software testing yaitu *white Box Testing* dan *Black Box Testing* (Bois Bezier). Untuk *white Box Testing* merupakan salah satu cara untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisa kode program apakah ada yang salah atau tidak. Sedangkan untuk *Black Box Testing*, melakukan pengujian berdasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi dan kesesuaian alur fungsi dengan sistem kerja yang diinginkan perancangannya.

Jenis-jenis pengujian *Black Box testing* adalah: a). *Fungsional Testing*., b). *Non Fungsional Testing* dan c). *Regulation Testing*. Disini dipilih jenis pengujiannya adalah jenis *Fungsional Testing* dengan pertimbangan bahwa ini untuk pemula jadi : a). Penguji tidak perlu memiliki pengetahuan tentang Bahasa pemrograman tertentu karena ini untuk pemula. b). Pengujian dilakukan berdasarkan sudut pandang user agar dapat mengungkapkan inkonsistensi dan ambiguitas data spesifikasi. c). Programmer dan tester memiliki ketergantungan satu sama lain.

### Sistem Informasi perpustakaan

Perpustakaan sebagai suatu tempat pengunjung dapat memperoleh akses terhadap informasi dan pengetahuan melalui penyediaan

bahan Pustaka dan pelayanan peminjaman dan pengembalian Pustaka.

Sistem Informasi perpustakaan adalah suatu sistem di dalam organisasi *public* yang mempertemukan kebutuhan pengelolaan transaksi (Muslimin dkk).

### Web Browser dan Web Server

Web browser atau sering juga disebut Internet Browser berfungsi sebagai jembatan bagi pengguna computer dalam menjelajah dunia maya. Internet browser merupakan sebuah aplikasi atau software yang digunakan untuk mengolah data yang ditransfer dari world Wide Web (lebih dikenal dengan istilah www) ke komputer dan menampilkannya secara visual agar mudah dimengerti oleh pengguna internet. (Setyaji dkk). Browser web adalah *software* yang digunakan untuk menampilkan informasi web server". Sedangkan aplikasi web adalah aplikasi yang diakses melalui jaringan internet. Dengan menggunakan aplikasi web, kita, hanya perlu menempatkan aplikasi dalam sebuah server sehingga aplikasi tersebut dapat diakses dari manapun asal pengguna dapat mengakses web-servernya.

*Web server* adalah server yang melayani semua permintaan aplikasi web. Dalam hal ini untuk penulisan pemrogramannya adalah *specific* yaitu menggunakan Bahasa pemrograman HTML (*Hypertext Marku Language*) (Sidik dkk)

### Kualitas Perangkat Lunak

Pengujian dilakukan untuk mendapatkan perangkat lunak dengan kualitas yang baik. Kualitas adalah karakteristik atau atribut dari sesuatu. Kualitas perangkat lunak adalah suatu keadaan yang secara jelas menyatakan permintaan dari fungsi dan kinerja, yang secara eksplisit dituliskan ke dalam dokumen standar pembangunan dan secara implisit menyatakan karakteristik yang diharapkan oleh semua

pengembang *software*. Perancangan perangkat lunak harus mempunyai kualitas yaitu: *Operability, Observability, Controllability, Decomposability, Simplicity, Stability, dan Understandability*.

Pengertian kualitas perangkat lunak terbagi dua tingkat yaitu: Kualitas (*intrinsic*) produk yaitu pengukuran dilakukan dengan menggunakan jumlah *defect* yang terjadi dalam suatu perangkat lunak atau dengan perkiraan berapa lama perangkat lunak masih dapat berfungsi sebelum menjadi *crash*. Dan kepuasan *customer* ialah pengukuran yang dilakukan dengan memperhatikan permasalahan yang dihadapi *customer* dan tingkat kepuasan *customer* selama menggunakan perangkat lunak tersebut.

Proses verifikasi dan validasi adalah keseluruhan proses daur hidup. Verifikasi dan validasi harus diterapkan pada tahapan dalam proses perangkat lunak. Proses verifikasi dan validasi mempunyai dua obyektif *principal* yaitu (Richard Helm) :

1. Verifikasi statis, yaitu berhubungan dengan analisis representasi sistematis untuk menemukan masalah, biasa disebut dengan *software inspection*.
2. Verifikasi dinamis, yaitu berhubungan dengan pelaksanaan dan memperhatikan perilaku produk, biasa disebut *software testing*.

## 2. METODE

Metode *Black Box Testing* merupakan pengujian untuk menunjukkan kesalahan pada sistem aplikasi seperti kesalahan pada fungsi sistem aplikasi, serta menu aplikasi yang hilang. Jadi *Black Box testing* merupakan metode uji fungsionalitas sistem aplikasi. Dalam melakukan pengujian menggunakan masukan data acak dengan tujuan untuk mendapatkan hasil yang pasti. Dikatakan pasti artinya bila salah, maka di tolak oleh sistem informasi atau data input tersebut tidak dapat disimpan dalam data base, sedangkan bila data input benar maka dapat di terima / masuk di database sistem informasi.

Teknik pengujian pada *Black Box testing* ada banyak macamnya yaitu: a). *Teknik Equivalence Partitioning* yaitu dengan cara

melakukan *partition* atau pembagian menjadi beberapa partisi dari input data. b). *Teknik Boundary Value Analysis* yaitu dengan cara mencari adakah error dari luar atau sisi dalam *software*, minimum maupun maximum nilai dari error yang di temukan. c). *Teknik Fuzzing* yaitu merupakan teknik untuk mencari Bug/gangguan dari *software* dengan menggunakan injeksi data yang terbilang cacat. d). *Teknik Cause-Effect Graph* ialah suatu Teknik testing dimana menggunakan *graphic* sebagai acuannya. Dimana dalam grafik ini menggambarkan relasi diantara efek dan penyebabnya. e). *Teknik Orthogonal Array Testing* adalah jenis Teknik yang digunakan jika input domain yang relative terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar. f). *Teknik All Pair Testing* yaitu semua pasangan dari test case di desain sedemikian rupa agar dapat di eksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar input parameternya, Tujuan testing ini adalah memiliki pasangan test case yang mencakup semua pasangan tersebut. g). *Teknik state Transition*. Teknik ini berguna untuk melakukan pengetestan terhadap kondisi dari mesin dan navigasi dalam bentuk grafik. (Boris Besar, 2008).

Pada penelitian ini teknik Pengujian yang dilakukan pada aplikasi Sistem Informasi perpustakaan adalah Teknik *Equivalence Partitioning*.

Teknik *Equivalence partitioning* yaitu teknik yang membagi data masukan dari unit perangkat lunak menjadi beberapa partisi data dari mana test case dapat diturunkan. Pada prinsipnya, uji kasus dirancang untuk menutupi setiap partisi minimal sekali. Teknik ini mencoba untuk mendefinisikan kasus uji yang mengungkap kelas kesalahan, sehingga mengurangi jumlah kasus uji yang harus dikembangkan. (Jaya M.S Gumilang dkk).

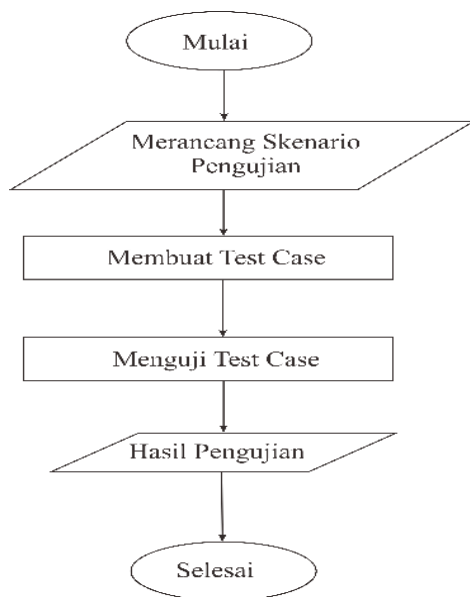
Tujuan menggunakan Teknik ini untuk mencari kesalahan pada:

- a) Fungsi yg salah atau hilang,
- b) Kesalahan pada interface,
- c) Kesalahan pada struktur data atau akses database,
- d) Kesalahan performansi,
- e) Kesalahan inisialisasi dan tujuan akhir

Pada penelitian ini teknik Pengujian Equivalence partition dilakukan pada aplikasi Sistem Informasi Perpustakaan berbasis web. Sebagai masukannya, digunakan input data acak yang bertujuan untuk melakukan test dengan pasti apakah system yang dibuat menolak dengan masukan data/informasi yang tidak sesuai dengan yang telah di formatkan dalam perancangan, namun bila data sesuai perancangan hasilnya adalah akan sebaliknya. (Ahrizal dkk).

### Proses Melakukan Pengujian

Dalam melakukan *software testing* ada beberapa tahapan-tahapan kegiatan yang harus di laksanakan. Hal ini dilukiskan melalui diagram alir pada gambar 1:



Gambar :1. Diagram alir kegiatan *software testing*

Keterangan :

1. Tahap perancangan scenario, merancang perintah pada form yang akan dilakukan pengujian.
2. Tahap membuat *test case*, yaitu mencatat kolom-kolom yang akan diuji pada form penginputan.
3. Tahap menguji *test case*, melakukan pengujian dengan melakukan perintah pada skenario pengujian.
4. Tahap hasil pengujian, menuliskan seluruh alur pengujian dalam bentuk laporan beserta metode yang digunakan.

### Rancangan Test Case Form

Dalam penelitian dilakukann pengujian terhadap 3 buah form yaitu: form Inisialisasi(login), form Peminjaman Buku dan Form Pengembalian Buku. Adapun pertimbangan pengambilan form masing-masing dapat diterangkan sebagai berikut:

#### 1. Pengujian Form Inisialisasi(LOGIN)



Gambar 2: Form Login

Halaman ini diuji karena halaman login merupakan tampilan awal yang akan dijumpai oleh pengguna, sehingga kestabilannya harus sudah teruji sebelum aplikasi diperkenalkan ke publik.

Pada Form ini dirancang akan dilakukan test untuk empat macam bentuk pengisian data input seperti yang ditampilkan pada tabel: 1

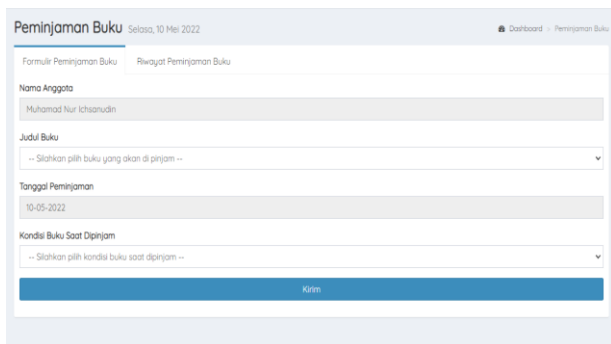
Tabel 1: Rancangan input data pada form Login

Id	Deskripsi Pengujian	Hasil yang Diharapkan
A01	Mengosongkan semua isian data	Sistem akan menolak akses <i>login</i> dengan menampilkan pesan nama pengguna harus diisi
A02	Mengisi username dengan benar dan mengosongkan isian <i>password</i>	Sistem akan menolak akses <i>login</i> dan menampilkan pesan kata sandi harus diisi

A03	Mengisikan isian dengan salah satu data yang salah	Sistem akan menampilkan pemberitahuan bahwa username atau kata sandi salah
A04	Mengisikan isian dengan data yang benar	Sistem akan menerima akses login dan akan mengarahkan langsung ke dashboard pengguna

B04	Mengisikan isian dengan data yang benar	Sistem akan menampilkan pesan bahwa peminjaman buku berhasil
-----	---	--

**2. Pengujian Form Peminjaman buku**  
 Form Peminjaman buku menjadi pilihan untuk diuji karena



Gambar 3: Form peminjaman buku

Form peminjaman buku adalah salah satu form yang paling banyak digunakan oleh pengguna aplikasi. Form ini memiliki dua isian yang harus diberikan oleh pengguna yaitu judul buku dan kondisi buku saat dipinjam. Rancangan pengujiannya ialah dengan memasukkan 4 variasi masukan data yang sering terjadi. Hal tersebut dilukiskan pada tabel 2 sebagai berikut:

Id	Deskripsi Pengujian	Hasil yang Diharapkan
B01	Mengosongkan semua isian data	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dipinjam
B02	Mengisi judul buku dan mengosongkan kondisi buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan kondisi buku
B03	Mengisi kondisi buku dan mengosongkan judul buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dipinjam

**3. Pengujian Form Pengembalian buku**  
 Form Pengembalian buku dipilih dilakukan pengujian karena sering digunakan., seperti dilukiskan pada gambar 4



Gambar 4: Form Pengembalian Buku

Form pengembalian buku adalah form yang paling sering digunakan setelah form peminjaman buku. Form ini akan dilakukan serangkaian pengujian dengan perancangan yaitu dengan memberikan 4 macam kondisi input data yang sering terjadi. Hal ini dilukiskan pada Tabel 3:

Tabel 3: rancangan input data pada form Pengembalian buku

Id	Deskripsi Pengujian	Hasil yang Diharapkan
C01	Mengosongkan semua isian data	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dikembalikan
C02	Mengisi judul buku dan mengosongkan kondisi buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan kondisi buku yang akan dikembalikan
C03	Mengisi kondisi buku dan mengosongkan judul buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dikembalikan
C04	Mengisikan isian dengan data yang benar	Sistem akan menampilkan pesan bahwa pengembalian buku berhasil

**3. HASIL DAN PEMBAHASAN**

Setelah data input kita masukkan pada sistem, maka kita lihat bagaimana hasilnya apakah sesuai dengan yang diharapkan atau tidak. Dengan memberikan 4 variasi masukan pada masing-masing form yang diuji, hasilnya dapat dilihat pada tabel 5, 6 dan 7.

Setelah Rancangan input data pengujian pada masing-masing form yang akan di uji selesai dibuat, maka selanjutnya adalah melakukan pengujian pada system informasi perpustakaan yang telah dibuat.

Dari hasil pengujian untuk *Form Login* menunjukkan bahwa apa yang diharapkan dalam perancangan sesuai dengan hasil outputnya. Demikian pula untuk Form Peminjaman buku hasilnya sesuai dengan yang diharapkan baik untuk masukan yang salah maupun masukan yang benar.

Selanjutnya untuk Form Pengembalian buku terdapat ketidak sesuaian antara hasil yang diharapkan dengan output yang dihasilkan. Hal ini terjadi karena sistem masih belum stabil dikarenakan adanya kesalahan pada baris kode(*syntax*). Oleh karena itu perlu dilakukan update pada *syntax* untuk memperbaiki *bug* tersebut.

Secara keseluruhan dilakukan 12 skenario pengujian pada halaman yang berbeda. Pengujian tersebut menunjukkan hasil yang cukup memuaskan namun terdapat tiga skenario pengujian yang tidak sesuai dengan harapan. Pengujian pada form pengembalian buku ditemukan tiga hasil yang tidak sesuai harapan. Berdasarkan data tersebut didapatkan perhitungan sebagai berikut:

$$\text{Validitas sistem} = \frac{(JSK - JTSH)}{TSP} \times 100\%$$

$$\text{Validitas sistem} = \frac{12-3}{12} \times 100\% = 75\%$$

dengsn JSK=jjumlsk skenstop pengujisn, JTSH=jumlsk tfsk sesuai hrapan, TSP=totsl skenirio pengujisn

**a. Hasil Pengujian Form Login**

Tabel 4. Hasil Pengujian Form Login

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil Pengujian</b>	<b>Kesimpulan</b>
A01	Mengosongkan	Sistem akan menolak	Sistem menolak	Sesuai

	semua isian data	akses login dengan menampikan pesan nama pengguna harus diisi	pesan bahwa nama pengguna harus diisi	
A02	Mengisi username dengan benar dan mengosongkan isian <i>password</i>	Sistem akan menolak akses login dan menampikan pesan kata sandi harus diisi	Sistem menampikan pesan bahwa kata sandi harus diisi	Sesuai
A03	Mengisikan isian dengan salah satu data yang salah	Sistem akan menampikan pemberitahuan bahwa username atau kata sandi salah	Sistem menampikan peringatan bahwa nama pengguna atau kata sandi salah	Sesuai
A04	Mengisikan data yang benar	Sistem akan menerima akses login dan akan mengarahkan langsung ke dashboard pengguna	Sistem mengalihkan halaman dashboard setelah login	Sesuai

**b. Hasil Pengujian Form Peminjaman Buku**

Tabel 6. Hasil pengujian form peminjaman buku

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang</b>	<b>Hasil Pengujian</b>	<b>Kesimpulan</b>

		<b>Diharapkan</b>		
B01	Mengosongkan semua isian data	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dipinjam	Sistem menampilkan pesan bahwa pengguna belum memasukkan buku yang akan dipinjam	Sesuai
B02	Mengisi judul buku dan mengosongkan kondisi buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan kondisi buku	Sistem menampilkan pesan bahwa pengguna belum memasukkan kondisi buku yang akan dipinjam	sesuai
B03	Mengisi kondisi buku dan mengosongkan judul buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dipinjam	Sistem menampilkan pesan bahwa pengguna belum memilih buku yang akan dipinjam	Sesuai
B04	Mengisikan isian dengan data yang benar	Sistem akan menampilkan pesan bahwa peminjaman buku berhasil	Sistem menampilkan pesan bahwa buku berhasil dipinjam	Sesuai

**c. Hasil Pengujian Form Pengembalian Buku**

Tabel 7. Hasil Pengujian Form Pengembalian Buku

<b>Id</b>	<b>Deskripsi Pengujian</b>	<b>Hasil yang Diharapkan</b>	<b>Hasil Pengujian</b>	<b>Kesimpulan</b>
C01	Mengosongkan semua isian data	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dikembalikan	Sistem gagal menampilkan pesan peringatan	Tidak sesuai
C02	Mengisi judul buku dan mengosongkan kondisi buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan kondisi buku yang akan dikembalikan	Sistem menampilkan pesan pengembalian buku berhasil	Tidak sesuai
C03	Mengisi kondisi buku dan mengosongkan judul buku	Sistem akan memberikan pesan error bahwa pengguna belum memasukkan buku yang akan dikembalikan	Sistem gagal menampilkan pesan peringatan	Tidak sesuai

		yang akan dikembalikan		
C 04	Mengisikan isian dengan data yang benar	Sistem akan menampilkan pesan bahwa pengembalian buku berhasil	Sistem menampilkan pesan pengembalian buku berhasil	Sesuai

#### 4. PENUTUP

##### Kesimpulan

Pengujian perangkat lunak menggunakan Metode Black Box Testing dengan Teknik Equivalence Partition cocok bagi pemula karena lebih sederhana caranya. Hasil pengujian menunjukkan output beberapa form dari system informasi perpustakaan yang dibuat masih terdapat error yaitu tidak sesuai dengan yang diharapkan pada perancangan dengan validitas 75%. Untuk itu perlu dilakukan refisi atau edit ulang tentang syntax pada form yang error tersebut.

#### 5. DAFTAR PUSTAKA

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

Andrian Agustian, Imas Andryani, Sita Khoerunisa, Adjie Pangestu, Aries Saifudin(2022), Implementasi Teknik Equivalence Partitioning pada Pengujian Aplikasi E-learning Berbasis Web, *Jurnal Teknologi System Informasi dan Aplikasi*, Program Studi Teknik Informatika Universitas Pamulang, Vol 5, No 1 (2022)

Boris Beizer, *Software Testing Techniques*, Van Nostrand Reinhold; 2nd edition (June 1, 2008)

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.

Muslimin, D. B., Kusmanto, D., Amelia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-26. doi:10.32493/informatika.v5i1.3778

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

Rosalina, A., Rassi, A. A., Hadi, G. Y., Ubaidillah, R., & Desyani, T. (2020). Pengujian Black Box pada Sistem Informasi Penjualan HI Shoe Store Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 5, 26-29. doi:10.32493/informatika.v5i1.3964

Richard Helm, Erich Gamma Ralph Johnson, John Vlissides, and Grady Booch., *Elements of Reusable Object-Oriented Software*, Addison -Wesley, 1977

Sidik, Betha, Husni Pohan, 2012, *Pemrograman web dengan HTML*, Bandung-Informatika

Setiaji, Jarot, 2010, *Buku pintar Menguasai Komputer dan Lap top*, Jakarta, Mediakita.