

## Implementasi *Black Box* Testing pada Aplikasi Sistem Kasir dengan Menggunakan Teknik *Equivalence Partitions*

M. Nurul Huda<sup>1</sup>, Muhammad Burhan<sup>2</sup>, Ahmad Satibi<sup>3</sup>, Helmy Agus Pradita<sup>4</sup>, Aries Saifudin<sup>5</sup>, Irpan Kusyadi<sup>6</sup>

Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek No.46, Buaran, Serpong, Kota Tangerang Selatan, Banten Indonesia 15310  
e-mail: <sup>1</sup>muhammadhuda694@gmail.com, <sup>2</sup>muhammadburhan160@gmail.com, <sup>3</sup>blognyasatibi@gmail.com, <sup>4</sup>helmy.vestoria@gmail.com, <sup>5</sup>aries.saifudin@unpam.ac.id, <sup>6</sup>dosen00673@unpam.ac.id

Submitted Date: January 20<sup>th</sup>, 2022

Reviewed Date: February 01<sup>st</sup>, 2022

Revised Date: February 03<sup>rd</sup>, 2022

Accepted Date: April 30<sup>st</sup>, 2022

### Abstract

Testing an application serves to check whether an application / program is running with what is desired or whether there are still errors / errors that need to be corrected so that the program created into a quality program. Software testing method consists of two types, namely Blackbox testing and Whitebox Testing where many are used in the time of the testers to check the program is appropriate or not with what they want. Testing this time the software that will be tested using the Blackbox testing method, this time testing only has the purpose to check the program being tested whether it is in accordance with the benefits that the manufacturer wants to the program without the tester having to know what program code is used in the application. Testing the Blackbox method is really a variety of one is Equivalence Partitions. which is utilized by us in the testing process this time. This technique is a trial in which the tester inputs data on all pages in the cashier application process, each page in the form of input will be tested and classified based on its function whether it is valid or not. What is desired in this test is the creation of the quality of the software produced in accordance with the benefits expected by the creator, and can also provide benefits to buyers who will use it.

Keywords: Black Box Testing; cashier application; Equivalence Partitions

### Abstrak

Pengujian terhadap sebuah aplikasi berfungsi untuk mengecek apakah sebuah aplikasi/program sudah berjalan dengan apa yang diinginkan atau masih adakah *error*/kesalahan yang perlu untuk diperbaiki supaya program yang diciptakan menjadi program yang berkualitas . Metode pengujian *software* terdiri atas dua jenis yaitu *Blackbox testing* dan *Whitebox Testing* di mana banyak digunakan di kalangan para pengujian untuk mengecek program telah sesuai atau belum dengan apa yang mereka inginkan. Pengujian kali ini *software* yang akan diuji menggunakan metode *Blackbox testing*, pengujian kali ini cuma mempunyai tujuan untuk mengecek program yang sedang diuji apakah sesuai dengan faedah yang di inginkan pembuat kepada programnya tanpa pengujian harus tahu kode program apa yang dipergunakan pada aplikasi tersebut. Pengujian metode *Blackbox* benar-benar banyak ragamnya satunya ialah *Equivalence Partitions*. yang dimanfaatkan oleh kami dalam proses pengujian kali ini. Teknik ini merupakan sebuah uji coba yang di mana pengujian menginput data pada semua halaman yang ada pada proses aplikasi kasir, setiap halaman yang berupa inputan akan dilaksanakan pengujian dan digolongkan berdasarkan fungsinya apakah itu berharga valid maupun tidak. Yang dikehendaki pada pengujian ini ialah terciptanya mutu *software* yang dihasilkan sesuai dengan faedah yang diharapkan pencipta, dan juga dapat memberikan keuntungan pada pembeli yang akan menggunakannya.

Kata Kunci: Pengujian *Black Box*; Aplikasi Kasir; Teknik *Equivalence Partitions*

## 1 Pendahuluan

Pengujian merupakan tahapan penting yang harus dilakukan untuk memberikan jaminan terhadap kualitas perangkat lunak yang dikembangkan (Muslimin, et al., 2020). Pengujian program ialah suatu proses yang ditujukan guna mengidentifikasi kebenaran, kelengkapan dan kualitas suatu program/aplikasi yang sedang digarap ataupun sedang dikembangkan. Tujuan dari pengujian adalah untuk memastikan bahwa sistem yang dihasilkan sesuai dengan kebutuhan dan layak untuk digunakan (Arwaz, Putra, Putra, Kusumawijaya, & Saifudin, 2019). Pengujian sistem sangat penting untuk memberikan jaminan kualitasnya dan membuktikan bahwa fungsinya telah beroperasi dengan benar (Ijudin & Saifudin, 2020). Proses pengujian *software* adalah di mana kita menjalankan suatu *software* dalam wilayah simulasi ataupun wilayah nyata, menggunakan data input yang sudah dikehendaki dengan menggunakan cara yang telah disepakati. Pengujian *software* sebenarnya memiliki tujuan utama yang sederhana, yaitu untuk menjamin *software* yang dihasilkan sesuai dengan kebutuhan (requirement) yang ditetapkan sebelumnya (Maulana, Kurniawan, Keumala, Sukma, & Saifudin, 2020). Dengan begitu, pengujian *Software* adalah sebuah kegiatan untuk mengecek benar atau tidaknya suatu sistem perangkat lunak bebas dari kesalahan. Proses uji pada sebuah perangkat lunak ini ditujukan untuk menemukan kesalahan sehingga program dapat disempurnakan sebelum produk dirilis ke pemesan program nantinya.

Masalah sebuah *software* pada dasarnya terdiri dari tiga hal, antara lain ialah sistem yang bersangkutan, persyaratan keluaran, dan persyaratan masukan. Pengujian sebuah *software* ialah sebuah elemen penting dari terjaminnya kualitas perangkat lunak dan merupakan sebuah review akhir terhadap kemampuan, desain dan pembuatan sebuah program. Proses pengujian perangkat lunak biasanya memerlukan upaya 30-40% dari total sebuah pekerjaan proyek. Untuk proyek yang mempunyai tingkat resiko tinggi menyangkut nyawa manusia, biaya pengujian ini bisa 3-5 kali dari proyek biasa.

Kesalahan terhadap sementara pembuatan *software* dapat terlampaui fatal nantinya. Pengeditan data tidak efektif sementara pengeditan data dilakukan dan berfungsi dengan benar, namun masih gagal menahan data yang salah dimasukkan ke dalam sistem. Orang

mengalami secara langsung rasa frustrasi memanfaatkan perangkat lunak yang rumit, sukar dinavigasi, dan perlu lebih dari satu langkah untuk melakukan tugas-tugas sederhana.

Pengujian suatu *software* dari sisi kemampuan fungsional tanpa harus menguji tampilan desain dan kode program untuk bisa mengerti apakah fungsi, input dan output yang berasal dari perangkat lunak yang sedang dibuat sesuai dengan kemampuan yang diinginkan/butuhkan. Metode *Blackbox testing* merupakan salah satu metode yang terbilang simple untuk diterapkan, hanya perlu batas bawah dan batas atas yang berasal dari data yang diinginkan. Pengujian secara *black box* merupakan pengujian perangkat lunak tanpa harus memperhatikan detail proses perangkat lunak (Sinulingga, Zuhri, Mukti, Syifa, & Saifudin, 2020). Pengujian *black box* dilakukan berdasarkan masukan dan luaran tanpa memperhatikan rincian program sehingga penguji tidak perlu memiliki pengetahuan pemrograman (Shaleh, Prayogi, Pirdaus, Syawal, & Saifudin, 2021). Estimasi dari banyaknya jumlah data yang diuji dapat dihitung lewat banyaknya field data entri yang diuji, ketetapan entri data wajib dilengkapi dan juga kasus batas atas dan batas bawah yang harus memenuhi ketentuan.

Ada banyak langkah yang dapat dimanfaatkan di saat menguji memanfaatkan metode *Black Box Testing* antara lain ialah menerapkan teknik *Equivalence Partitions*. *Equivalence Partitioning* merupakan sebuah pengujian berdasarkan masukkan data pada setiap form yang memecah domain masukan ke dalam kelompok valid dan tidak valid (Aziz, Setiawan, Khanh, Nurdiyansyah, & Yulianti, 2020). *Equivalence Partitions* adalah teknik pengujian yang didasari dengan memasukkan data terhadap tiap-tiap halaman yang tersedia terhadap aplikasi, tiap-tiap menu masukan akan dilakukan uji coba dan kumpulkan berdasarkan fungsinya baik itu memiliki nilai valid ataupun tidak valid. Perancangan kasus uji (test case) pada teknik *Equivalence Partitions* didasarkan pada evaluasi kelas yang menggambarkan kondisi ekuivalen masukan dari kumpulan keadaan yang valid dan tidak valid (Nugraha, Aditama, Arrofi, Ahmad, & Yulianti, 2020). Pada penelitian ini terkandung lebih dari satu tahapan, di mana di mulai dengan memilih *Test case* yang akan diujicobakan, lalu membaginya dengan lebih dari satu partisi masukan dan keluaran. Hal ini

dilakukan untuk mendapatkan dokumentasi pengujian.

## 2 Metodologi

Pengujian aplikasi/*software* sangat diperlukan untuk dilakukan sebab setiap pembuat *software* pasti pernah melakukan kesalahan dalam masa pembuatannya. 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). Kesalahan sebuah *software* dan *software* lainnya akan berbeda pada setiap *software* (Akiladevi, Vidhupriya, & Sudha, 2018). Itulah mengapa perlunya melakukan pengujian *software* untuk bisa melakukan sebuah verifikasi dan validasi bahwa aplikasi dibuat berdasarkan kebutuhan pengguna. Pengujian yang akan dipakai ialah *blackbox testing* agar *software* yang dihasilkan memiliki kualitas yang sangat baik. Metode 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).

Metode *Blackbox testing* ialah sebuah metode yang banyak digunakan dalam menguji sebuah *software* tanpa wajib mencermati dengan teliti *software* tersebut. Pengujian ini cuma memfokuskan terhadap nilai keluaran berdasarkan nilai masukan masing-masing. Tidak tersedia usaha untuk sadar kode program apa yang digunakan. Proses

pengujian *Black Box Testing* menggunakan langkah mencoba sebuah program yang telah dibuat dengan mencoba menginput data pada setiap halaman yang tersedia. Pengujian ini dibutuhkan guna mengetahui program tersebut telah berjalan sesuai dengan yang dibutuhkan oleh perusahaan atau belum (Wahyudi, Utami, & Arief, 2016).

*Equivalence Partitioning* (EP) adalah satu dari sekian banyak teknik pengujian pada metode *Blackbox testing* yang kami terapkan guna melakukan uji coba pada bagian input dan juga mengelompokkannya berdasarkan masing-masing dari fungsinya. Dan akan menghasilkan sebuah *test case* menjadi akurat. Penelitian kali ini akan dilaksanakan beberapa langkah. Langkah awal ialah kami memilih *Test case* untuk perangkat lunak dan nantinya akan kami uji coba menggunakan metode *Equivalence Partitions*. lantas menginisialisasi standar *grade partition* pada bagian masukan dan keluarannya. Semua itu dilaksanakan supaya kami mendapatkan hasil berupa dataset yang berbentuk dokumentasi pengujian bersama metode *Equivalence Partitions*. dan untuk pula mendapatkan nilai tingkat efektifitas terhadap metode *Equivalence Partitions*.

Akan ada sebuah tabel rencana *Test case* berguna pada pelaksanaan pemeriksaan program telah sesuai yang diinginkan atau masih wajib disempurnakan supaya menghasilkan program yang memiliki kualitas yang tinggi. Di bawah ini akan kami jabarkan mengenai tabel Rancangan *Test case* berdasarkan teknik *Equivalence Partitions*..



Gambar 1 halaman Login kasir & admin

Pada uji coba kali ini dilaksanakan di halaman Menu *login* Admin menggunakan cara menginput data yang salah yang tidak sesuai dengan *type* data yang sudah di buat. Hasil uji pada bagian *Username* akan valid jika *Username* diinput menggunakan huruf atau angka maupun kombinasi dari huruf dan angka yang maksimal input karakternya adalah berjumlah 100 karakter. Begitupun data tidak akan valid semisal *username* di *input* menggunakan lebih dari 100 karakter. Hasil uji pada bagian *password* akan valid jika *Password* diinput menggunakan huruf atau angka maupun kombinasi dari keduanya dengan maksimal input karakter berjumlah 100. Begitupun data tidak akan valid semisal *Password* di *input* menggunakan lebih dari 100 karakter.

Pada uji coba kali ini dilaksanakan di halaman Menu *login* Kasir menggunakan cara menginput data yang salah yang tidak sesuai dengan *type* data yang sudah di buat. Hasil uji pada bagian *Username* akan valid jika *Username* diinput menggunakan huruf atau angka maupun kombinasi dari huruf dan angka yang maksimal input karakternya adalah berjumlah 100 karakter. Begitupun data tidak akan valid semisal *username* di *input* menggunakan lebih dari 100 karakter. Hasil uji pada bagian *password* akan valid jika *Password* diinput menggunakan huruf atau angka maupun kombinasi dari keduanya dengan maksimal input karakter berjumlah 100. Begitupun data tidak akan valid semisal

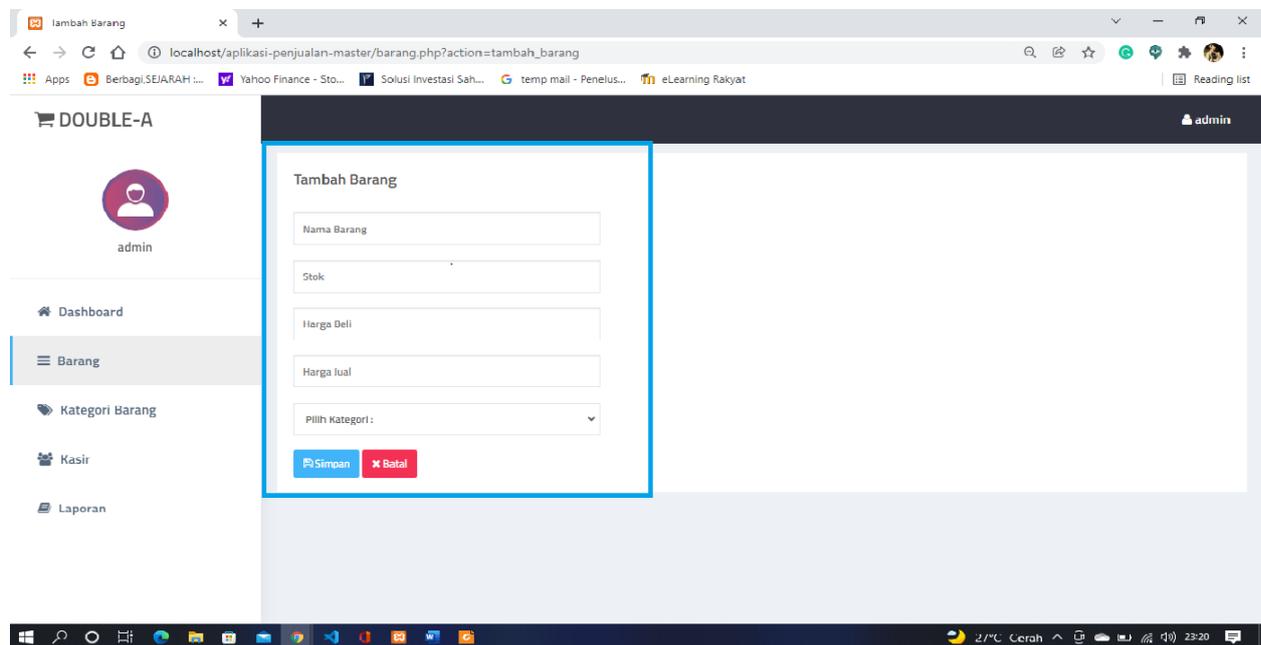
*Password* di *input* menggunakan lebih dari 100 karakter.

Tabel 1 Tabel *test case form login* admin & kasir

<b>Id</b>	<b>Langkah Pengujian</b>	<b>Hasil yang Diinginkan</b>
L01	Menginput <i>username</i> login admin menggunakan “unpam123”, <i>Password</i> diinput dengan “informatika123” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>Login</i> akan berhasil, secara otomatis sistem menutup <i>form login</i> dan masuk ke halaman utama menu admin.
L02	Menginput <i>username</i> admin dengan “unpmm123”, <i>Password</i> diinput dengan “informatika321” di	<i>login</i> gagal dan akan tetap berada di <i>form login</i> .

	mana data yang diinput bukanlah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	
L03	Menginput <i>username</i> kasir dengan “kasir1”, <i>Password</i> diinput dengan “1234” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>Login</i> akan berhasil, secara otomatis sistem menutup <i>form login</i> dan masuk ke halaman

		transaksi kasir.
L04	Menginput <i>username</i> kasir admin dengan “kasir2”, <i>Password</i> diinput dengan “kasur123” di mana data yang diinput bukanlah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>login</i> gagal dan akan tetap berada di <i>form login</i> .



Gambar 2 Halaman Tambah Barang

Langkah uji coba selanjutnya akan ditujukan ke *form* tambah barang di mana kami akan memasukkan data yang tidak sesuai dengan *type* data. Hasil uji coba pada halaman tambah

barang akan valid jika semua kolom informasi barang yang ada diisi dan tidak boleh ada kolom yang kosong/ tidak di isi. Dan data akan tidak valid jika ada kolom kosong/tidak diisi.

Table 2 Tabel *test case* halaman Tambah Barang

<b>Id</b>	<b>Langkah Pengujian</b>	<b>Hasil yang Diinginkan</b>
M01	Menginput semua kolom pada halaman tambah barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik Simpan	Barang akan berhasil di tambah lalu akan muncul <i>notice</i> bahwa “Data berhasil ditambah” dan Data akan tersimpan
M02	Tidak mengisi semua kolom pada halaman tambah barang dengan tidak menggunakan ketentuan pada <i>database</i> , Lalu klik Simpan	Data tidak akan tersimpan dan akan tetap berada di halaman tambah barang serta tidak akan mendapatkan <i>notice</i> “ Data berhasil di Tambah”

Gambar 3 Tampilan *Form* transaksi

Langkah Pengujian selanjutnya diterapkan pada halaman transaksi kami akan memasukan data yang salah atau tidak sesuai dengan type data. Pengujian pada halaman transaksi data akan

valid jika semua kolom informasi barang yang akan di proses diisi dan tidak boleh ada kolom yang kosong/ tidak di isi. Dan data akan tidak valid jika ada kolom kosong/tidak diisi.

Tabel 3 Tabel *test case form* Transaksi

<b>Id</b>	<b>Langkah Pengujian</b>	<b>Hasil yang Diinginkan</b>
X01	Mengisi semua kolom pada halaman Transaksi barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik tombol proses transaksi	Data akan tersimpan lalu akan di arahkan ke halaman daftar transaksi
X02	Tidak mengisi semua kolom pada halaman Transaksi barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik tombol proses transaksi	Data tidak akan tersimpan dan akan tetap berada di halaman transaksi

### 3 Hasil dan Pembahasan

Berdasarkan dasi hasil rancangan uji kasus (*test case*) yang sudah dilakukan pada tabel

pertama sampai ketiga, maka bisa dilakukan pengujian. Hasil dari pengujian ditunjukkan pada tabel dibawah ini.

<b>Id</b>	<b>Langkah Pengujian</b>	<b>Hasil yang Diinginkan</b>	<b>Hasil dari Pengujian</b>	<b>Kesimpulan</b>
L01	Menginput <i>username</i> login admin menggunakan “ unpm123”, <i>Password</i> diinput dengan “informatika123” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>Login</i> akan berhasil, secara otomatis sistem menutup <i>form login</i> dan masuk ke halaman utama menu admin	<i>Login</i> berhasil dan diarahkan langsung ke Menu Utama	Sesuai
L02	Menginput <i>username</i> admin dengan “unpmm123”, <i>Password</i> diinput dengan “informatika321” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>login</i> gagal dan akan tetap berada di <i>form login</i> .	Akan ada <i>notice</i> “Maaf sepertinya <i>username</i> atau <i>password</i> anda salah “ dan tampilan tetap pada halaman <i>login</i> admin	Sesuai

Id	Langkah Pengujian	Hasil yang Diinginkan	Hasil dari Pengujian	Kesimpulan
L03	Menginput <i>username</i> kasir dengan “kasir1”, <i>Password</i> diinput dengan “1234” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>Login</i> akan berhasil, secara otomatis sistem menutup <i>form login</i> dan masuk ke transaksi kasir.	<i>Login</i> berhasil dan di arahkan langsung ke Halaman Transaksi	Sesuai
L04	Menginput <i>username</i> kasir admin dengan “kasir2”, <i>Password</i> diinput dengan “kasur123” di mana data yang diinput adalah data yang berada pada <i>database</i> lalu klik tombol <i>login</i> pada halaman.	<i>login</i> gagal dan akan tetap berada di <i>form login</i> .	Sistem akan muncul pesan “ Maaf sepertinya <i>username</i> atau <i>password</i> anda salah “ dan tampilan tetap pada halaman <i>login</i> Kasir	Sesuai
M01	Mengisi semua kolom pada halaman tambah barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik Simpan	Data akan tersimpan dan berhasil di tambah lalu akan muncul notice bahwa “ Data berhasil ditambah “	Sistem akan muncul “Data berhasil disimpan “ dan data berhasil disimpan ke <i>database</i> .	Sesuai
M02	Tidak mengisi semua kolom pada halaman tambah barang dengan tidak menggunakan ketentuan pada <i>database</i> , Lalu klik Simpan	Data tidak akan tersimpan dan akan tetap berada di halaman tambah barang serta tidak akan mendapatkan notice“ Data berhasil ditambah“	Sistem akan muncul “ Silahkan isi kolom ini “ dan data tidak tersimpan ke <i>database</i> .	Sesuai
X01	Mengisi semua kolom pada halaman Transaksi barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik tombol proses transaksi	Data akan tersimpan lalu akan diarahkan ke halaman daftar transaksi	Sistem akan muncul “ Data berhasil disimpan “ dan akan diarahkan ke halaman daftar transaksi .	Sesuai
X02	Tidak mengisi semua kolom pada halaman Transaksi barang dengan menggunakan ketentuan pada <i>database</i> , Lalu klik tombol proses transaksi	Data tidak akan tersimpan dan akan tetap berada di halaman transaksi	Sistem akan muncul “ Silahkan isi kolom ini “ dan akan tetap dihalaman transaksi	Sesuai

Total keseluruhan fungsi yang kami uji cobakan berjumlah 8 fungsi di mana fungsi *login* diuji sejumlah 4 kali percobaan, fungsi tambah barang diuji sejumlah 4 kali percobaan, fungsi Transaksi diujicoba sejumlah 2 kali percobaan. Jadi total keseluruhan pengujian sebanyak 10 kali percobaan. Dari semua uji coba fungsi tidak ditemukan *error* dalam pengujian.

#### 4 Kesimpulan

Pengujian pada sebuah aplikasi menggunakan metode *Blackbox testing* ditujukan

untuk memantau sudah sejalankah program itu dengan tujuan awal program itu dibuat tanpa harus kami tahu secara detail kode program yang digunakan. Hasil dari uji coba terhadap kualitas aplikasi kali ini, kami menyimpulkan pengujian dengan menggunakan metode *black box testing* berbasis *Equivalence Partitions*. akan mempermudah dalam pembuatan *case* pada uji coba, pengujian kualitas dan kesalahan akan lebih mudah terdeteksi yang mungkin akibatkan dari kelalaian pada saat pengetikan yang tidak terdeteksi sebelumnya . Dalam pengujian aplikasi

sistem kasir tidak di temukan *error* pada saat pengujian, menunjukkan bahwa aplikasi sudah bisa berjalan seperti apa yang di harapkan oleh pembuat aplikasi.

## 5 Saran

Kami memberi saran untuk pengembangan aplikasi berikutnya yaitu bisa menerapkan beberapa teknik pengujian *Blackbox testing* lainnya antara lain ialah *Error Guessing* atau *Boundary Value* agar nantinya bisa mengetahui kejanggalan *error* lainnya pada saat mengembangkan aplikasi ini supaya aplikasi yang dihasilkan mempunyai kualitas yang lebih baik lagi dan dapat memberikan kemudahan serta keuntungan bagi pengguna aplikasi nantinya.

## 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
- Akiladevi, R., Vidhupriya, P., & Sudha, V. (2018). A Study and Analysis on Software Testing tools. *International Journal of Pure and Applied Mathematics*, 118(18), 1783-1800.
- Arwaz, A. A., Putra, K., Putra, R., Kusumawijaya, T., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Pemenang Tender Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 2(4), 130-134. doi:10.32493/jtsi.v2i4.3708
- 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
- 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
- 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 Equivalents 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
- 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., Syifa, Z., & Saifudin, A. (2020). Pengujian Black Box pada Sistem Aplikasi Informasi Data Kinerja Menggunakan Teknik Equivalence Partitions. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 3(1), 9-14. doi:10.32493/jtsi.v3i1.4303
- Wahyudi, R., Utami, E., & Arief, M. R. (2016). Sistem Pakar E-Tourism Pada Dinas Pariwisata D.I.Y Menggunakan Metode Forward Chaining. *Jurnal Ilmiah DASI*, 17(2), 67-75.