

## Pengujian Black Box Berbasis Equivalence Partitions Pada Aplikasi Administrasi Penjualan Furniture

**Yudha Ardiyansyah<sup>1</sup>, Ahmad Muhadi<sup>2</sup>, Renal Aji Pangestu<sup>3</sup>, Rino Bahtiar<sup>4</sup>, Aris Setiawan<sup>5</sup>,  
Joko Riyanto<sup>6</sup>**

<sup>1-6</sup>Universitas Pamulang; Jl. Raya Puspitek No. 46 buaran, serpong, Kota Tangerang Selatan. Provinsi Banten  
15310. (021) 741-2566 atau 7470 9855

<sup>1-6</sup>Jurusan Teknik Informatika, Fakultas Teknik, Universitas Pamulang

e-mail: <sup>1</sup>yudha.ardiyansyah07@gmail.com, <sup>2</sup>ahmadmuhadi62@gmail.com, <sup>3</sup>renalaji08@gmail.com,  
<sup>4</sup>rinbahtiar357@gmail.com, <sup>5</sup>aris.seti990@gmail.com, <sup>6</sup>jokoriyanto@unpam.ac.id..

---

### *Abstrak*

Demi menjamin aplikasi (*software*) Administrasi Penjualan Furniture agar bebas dari *error* maka perlu dilakukan pengujian pada kualitas aplikasi. Aplikasi ini memiliki masalah ketika menginput data yang terkadang tidak bisa menginput data secara berulang-ulang kecuali kita refresh terlebih dahulu, maka aplikasi akan diuji dengan Metode *Black Box* berbasis *Equivalence Partition* secara menyeluruh berkenaan dengan penggunaan, manfaat, dan hasil yang didapati dari pemanfaatan perangkat lunak tersebut. Dalam pengujian perangkat lunak Seleksi Promosi Kenaikan Jabatan menggunakan Metode *Weight Product* terdiri dari 5 form. Metode *Black Box* berbasis *Equivalence Partition* menguji kualitas aplikasi yang akan dilakukan dokumentasi pengujian perangkat lunak dengan ditemukannya kesalahan pada setiap form yang dibagi menjadi tiga model kesalahan, yaitu kesalahan pada fungsi, struktur data dan *interface*. Hasil dari pengujian ini, untuk menjamin kualitas aplikasi Administrasi Penjualan *Furniture* bebas dari kesalahan. Agar dapat digunakan dengan nyaman tanpa adanya kesalahan pada saat menggunakannya.

*Kata kunci: Pengujian, Software, Black Box, Equivalence Partition*

---

### I. PENDAHULUAN

Pengujian perangkat lunak merupakan hal yang perlu kita lakukan untuk mendapatkan informasi tentang kualitas dari perangkat lunak yang telah kita buat, pengujian perangkat lunak harus dilakukan setelah membuat sebuah perangkat lunak agar kita dapat mendeteksi adanya kesalahan pada perangkat lunak yang dapat menyebabkan perangkat lunak tidak berfungsi dengan baik. Tujuan utama dari pengujian perangkat lunak untuk mencari kesalahan yang ada dari perangkat lunak. Hal ini diperlukan untuk menjamin bawah *software* yang telah kita buat dapat berjalan dengan baik tanpa ada kegagalan sama sekali. Pada proses pengujian perangkat lunak setiap kasus yang di uji harus memiliki identitas dan saling terhubung antara sekumpulan masukkan dengan hasil yang diinginkan.

*Testing* dan *Debugging* adalah aktivitas yang berbeda tetapi *debugging* harus diakomodasikan pada setiap strategi testing. Jika ingin mengimplementasikan strategi testing *software* yang sukses Tom Gilb menyarankan prosedur yang digunakan adalah (Hidayat & Muttaqin, 2018):

- 1) Menetapkan seluruh kebutuhan produk *software* dalam perhitungan sebelum memulai testing.
- 2) Status obyek testing harus jelas.
- 3) Memahami pengguna *software* dan mengembangkan sebuah profil untuk setiap kategori user.
- 4) Mengembangkan rencana testing yang menekankan pada "*Rapid Cycle Testing*".
- 5) Membangun *software* yang sempurna yang didesain untuk menguji dirinya sendiri.
- 6) Menggunakan tinjauan ulang yang formal sebagai filter sebelum pengujian.

- 7) Melakukan tinjauan ulang secara formal untuk menilai strategi tes dan kasus tes itu sendiri.
- 8) Mengembangkan pendekatan peningkatan yang berkelanjutan untuk proses *testing*.

Program yang akan kita uji pada pengkajian ini merupakan Aplikasi Administrasi Penjualan *Furniture* dengan metode *Blackbox Testing*, di mana tujuan dari pengujian ini untuk melihat program tersebut dapat berjalan sesuai dengan tugas program tersebut tanpa mengetahui kode program yang dipakai. Tahapan pertama pada pengujian dengan menggunakan *Blackbox Testing* adalah mengidentifikasi masukkan lalu diuji agar kita mengetahui letak dari kesalahannya. Pengujian menggunakan *Blackbox Testing* merupakan sebuah pengujian yang digunakan untuk melengkapi pengujian sebelumnya yaitu *Whitebox Testing* agar aplikasi yang kita buat memiliki kualitas yang baik serta waktu yang digunakan akan lebih efektif, sehingga dapat menguntungkan bagi perusahaan (Sinulingga et al., 2020). *Equivalence Partitions* merupakan sebuah pengujian berdasarkan masukkan data pada setiap form yang ada pada Aplikasi Administrasi Penjualan *Furniture*, setiap menu yang dimasukkan akan dilakukan pengujian dan dikelompokkan berdasarkan fungsinya baik itu bernilai valid ataupun tidak valid (Sinulingga et al., 2020).

II. METODE PELAKSANAAN

Pengujian sistem merupakan langkah terpenting dalam pembuatan sistem. pengujian sistem dilakukan karena setiap orang pasti bisa membuat kesalahan apa lagi saat membuat sistem yang rumit. Langkah ini dilakukan untuk menguji apakah sistem yang telah kita buat sudah layak untuk digunakan dan sesuai dengan kebutuhan, sehingga sangat perlu dilakukan pengujian untuk mengurangi kesalahan pada sistem yang dibangun yang dapat menyebabkan kerugian. *Black box testing* merupakan teknik pengujian software yang fokus pada spesifikasi fungsi-fungsi yang ada pada perangkat lunak yang dikembangkan. *Black box testing* cenderung dapat menemukan beberapa hal seperti fungsional yang tidak benar atau tidak ada, kesalahan struktur data, kesalahan akses basis data, kesalahan antar muka, kesalahan *performance* serta kesalahan inialisasi dan terminasi (Rahadi & Vikasari, 2020) Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan oleh perusahaan (Ningrum et al., 2020).

*Equivalence partitioning* merupakan salah satu metode *black box testing* yang akan membagi *domain* input dari suatu program ke dalam kelas-kelas data, dimana *test case* dapat diturunkan. Metode *black box equivalence partitioning* berdasarkan pada *premis* masukan dan keluaran dari suatu komponen yang dipartisi ke dalam kelas-kelas, menurut spesifikasi dari komponen tersebut, yang akan diperlakukan sama (*ekuivalen*) oleh komponen tersebut, dan dapat diasumsikan bahwa masukan yang sama akan menghasilkan respon yang sama (Rahadi & Vikasari, 2020). Dalam tahap pengujian *Equivalence Partitions* yang dilakukan adalah sebagai berikut:

1. Buatlah suatu rancangan *test case* yang berdasarkan fungsi di dalam pengujian perangkat lunak.
2. Buatlah batasan pengujian *Equivalence Partitions*.
3. Buatlah model pengujian berdasarkan skenario pengujian dan hasil yang diharapkan.
4. Melakukan pengujian berdasarkan rancangan model yang telah dibuat.



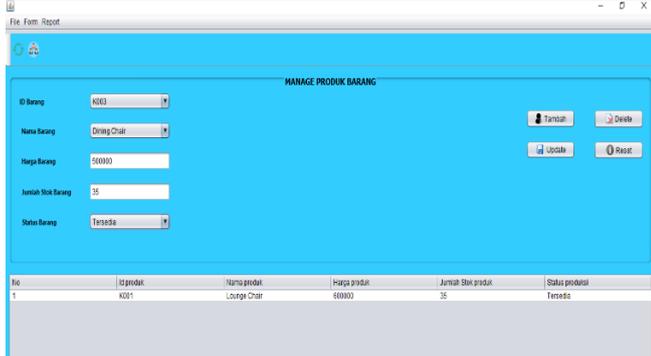
Gambar 2. 1 Tampilan Form Login

Pengujian ini dimulai dengan *Form Login*. Rencana pengujian *Username* data akan valid jika data yang dimasukan sama dengan data yang ada di dalam *database*, karena sistem pada *form login* ini harus memasukan data yang sama dengan data yang sudah tersimpan di dalam *database* jika sebaliknya maka tidak akan bisa masuk kedalam aplikasi. Begitu juga dengan *Password* yang dimasukan harus sama dengan yang sudah ada di *database*.

Tabel 2. 1 Rancangan *Test Case Form Login*

ID	Deskripsi Pengujian	Hasil yang diharapkan
L01	Menginput <i>Username</i> dengan	Akses <i>Login</i> berhasil, lalu

	“yuda22” dan <i>Password</i> “kelompok3” dengan ketentuan data tersebut sudah tersimpan di database kemudian klik tombol Masuk.	sistem akan pindah kedalam tampilan <i>Home</i> .
L02	Menginput <i>Username</i> dengan “aji66” dan <i>Password</i> “ayahadi21” dengan ketentuan data tersebut belum tersimpan di <i>database</i> kemudian klik tombol Masuk.	Akses <i>Login</i> Gagal, sistem tidak dapat membuka menu <i>Home</i> dan tampilan tetap pada form <i>Login</i> .



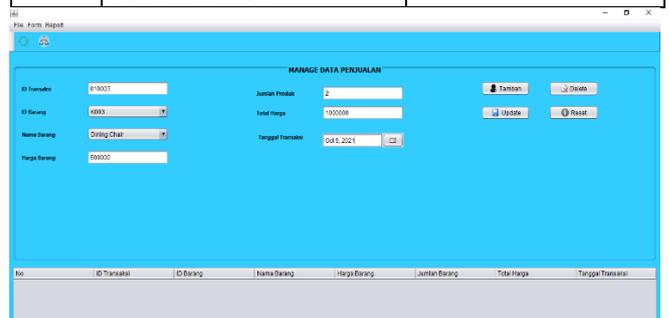
Gambar 2. 2 Tampilan Form Produk Barang

Pengujian berikutnya dilakukan pada *form* Produk Barang, pada *form* ini kita akan merencanakan pengujian pada *field* harga barang dan jumlah stok barang karena pada kedua *field* itu merupakan *type integer* dimana data yang dimasukkan harus berupa angka agar *valid*, dan sebaliknya jika diisi dengan huruf misal “abc” maka tidak akan *valid* atau tidak akan bisa tersimpan ke dalam sistem. Disini kita akan menguji *field* harga barang dengan menambahkan “Rp” dibagian depan, lalu pada jumlah stok barang akan di isi dengan kata “Tujuh” dan kita akan lihat apakah data dapat tersimpan atau tidak.

Tabel 2. 2 Rancangan Test Case Form Produk Barang

ID	Deskripsi Pengujian	Hasil Yang Diharapkan
B01	Pilih id “K003”, lalu Nama barang “Dining Chair” Mengisi Harga	Berhasil, dan data akan tersimpan pada <i>database</i> lalu

	barang “500000” dan jumlah stok barang “35” dengan status barang “Tersedia” jika sudah terisi semua selanjutnya kita tekan tambah.	data secara otomatis akan tampil pada table <i>alternative</i> .
B02	Pilih id “M001”, lalu Nama barang “Bar Table” Mengisi Harga barang “Rp. 3000000” dan jumlah stok barang “5” dengan status barang “Tersedia” jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal,dan data tidak akan tersimpan kedalam database serta data tidak akan tampil pada <i>table alternative</i> .
B03	Pilih id “L002”, lalu Nama barang “Lemari Document” Mengisi Harga barang “800000” dan jumlah stok barang “Tujuh” dengan status barang “Tersedia” jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal,dan data tidak akan tersimpan kedalam database serta data tidak akan tampil pada <i>table alternative</i> .



Gambar 2. 3 Tampilan Form Data Penjualan

Rancangan berikutnya dilakukan pada *form* Data Penjualan, pada *form* ini kita akan merencanakan pengujian pada *field* Id Transaksi, Harga barang, jumlah produk dan total harga dimana pada *field-field* itu merupakan *type integer* dimana data yang dimasukkan harus berupa angka, dan sebaliknya jika diisi dengan huruf misal “abc” atau simbol maka tidak akan *valid*

atau tidak akan bisa tersimpan ke dalam sistem. Disini kita akan menguji *field* Id transaksi dengan memasukan huruf misal “abc” begitu juga *field* yang lain untuk melihat apakah akan berhasil tersimpan atau tidak.

Tabel 2. 3 Rancangan Test Case Form Data Penjualan

ID	Deskripsi Pengujian	Hasil Yang Diharapkan
P01	Memasukan id dengan “010003”, lalu Pilih id “K003”, lalu Nama barang “Dining Chair” Mengisi Harga barang “500000” dan jumlah barang “2” dengan total harga “1000000”, lalu masukan tanggal transaksi “Oct 5,2021” jika sudah terisi semua selanjutnya kita tekan tambah.	Berhasil, dan data akan tersimpan pada <i>database</i> lalu data secara otomatis akan tampil pada <i>table alternative</i> .
P02	Memasukan id dengan “T003”, lalu Pilih id “K003”, lalu Nama barang “Dining Chair” Mengisi Harga barang “500000” dan jumlah barang “2” dengan total harga “1000000”, lalu masukan tanggal transaksi “Oct 5,2021” jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal,dan data tidak akan tersimpan kedalam <i>database</i> serta data tidak akan tampil pada <i>table alternative</i> .
P03	Memasukan id dengan “010003”, lalu Pilih id “K003”, lalu Nama barang “Dining Chair” Mengisi Harga barang “Rp. 500000” dan jumlah barang “2” dengan total	Gagal,dan data tidak akan tersimpan kedalam <i>database</i> serta data tidak akan tampil pada <i>table alternative</i> .

	harga “Rp. 1000000”, lalu masukan tanggal transaksi “Oct 5,2021” jika sudah terisi semua selanjutnya kita tekan tambah.	
P04	Memasukan id dengan “010003”, lalu Pilih id “K003”, lalu Nama barang “Dining Chair” Mengisi Harga barang “500000” dan jumlah barang “Dua” dengan total harga “1000000”, lalu masukan tanggal transaksi “Oct 5,2021” jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal,dan data tidak akan tersimpan kedalam <i>database</i> serta data tidak akan tampil pada <i>table alternative</i> .
P05	Memasukan id dengan “030004”, lalu Pilih id “L004”, lalu Nama barang “Lemari Buku” Mengisi Harga barang “2000000” dan jumlah barang “2” dengan total harga “4000000”, lalu masukan tanggal transaksi “Sep 8,2021” jika sudah terisi semua selanjutnya kita tekan tambah.	Berhasil, dan data akan tersimpan pada <i>database</i> lalu data secara otomatis akan tampil pada <i>table alternative</i> .

### III. HASIL DAN PEMBAHASAN

Berdasarkan Rancangan Uji kasus yang sudah dilakukan pada Tabel 2.1, 2.2 dan 2.3, maka akan mendapatkan Hasil Pengujian. Hasilnya dapat dilihat pada Tabel 3.1 Hasil Pengujian dibawah ini.

ID	Des kripsi Peng ujian	Hasil yang Diharap kan	Hasil Peng ujian	Kesim pulan
L01	Meng input <i>Username</i> dengan "yuda22" dan <i>Password</i> "kelom pok3" dengan ketentuan data tersebut sudah tersimpan di <i>database</i> kemudian klik tombol Masuk.	Akses <i>Login</i> berhasil, lalu sistem akan pindah kedalam tampilan <i>Home</i> .	Sistem akan langsung masuk kedalam <i>form Home</i>	Sesuai
L02	Meng input <i>Username</i> dengan "aji66" dan <i>Password</i> "ayah adi21" dengan ketentuan data tersebut belum tersimpan di <i>database</i> kemudian klik tombol Masuk.	Akses <i>Login</i> Gagal, sistem tidak dapat mem buka menu <i>Home</i> dan tampilan tetap pada <i>form Login</i> .	Sistem akan me nampil kan pesan " <i>user name</i> atau <i>pass word</i> salah" dan akan tetap pada <i>form Login</i>	Sesuai

B01	Pilih id "K003", lalu Nama barang " <i>Dining Chair</i> " Mengisi Harga barang "500000" dan jumlah stok barang "35" dengan status barang "Tersedia" jika sudah terisi semua selanjut nya kita tekan tambah.	Berhasil, dan data akan ter simpan pada <i>database</i> lalu data secara otomatis akan tampil pada <i>table alter native</i> .	Data berhasil ter simpan di dalam <i>database</i> lalu akan muncul pesan "data berhasil ter simpan" dan data akan muncul pada <i>table alter native</i>	Sesuai
B02	Pilih id "M001", lalu Nama barang " <i>Bar Table</i> " Mengisi Harga barang "Rp. 3000000" dan jumlah stok barang "5" dengan status barang "Tersedia" jika	Gagal, dan data tidak akan ter simpan kedalam <i>database</i> serta data tidak akan tampil pada <i>table alter native</i> .	Data gagal ter simpan lalu akan muncul pesan "terjadi ke salahan pada <i>database</i> " dan data tidak akan muncul pada <i>table alter native</i>	Sesuai

	sudah terisi semua selanjutnya kita tekan tambah.			
B03	Pilih id "L002", lalu Nama barang "Lemari Document" Mengisi Harga barang "800000" dan jumlah stok barang "Tujuh" dengan status barang "Tersedia" jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal, dan data tidak akan ter simpan ke dalam <i>database</i> serta data tidak akan tampil pada <i>table alter native</i> .	Data gagal ter simpan lalu akan muncul pesan "terjadi ke salahan pada <i>database</i> " dan data tidak akan muncul pada <i>table alter native</i>	Sesuai
P01	Memasukkan id dengan "010003", lalu Pilih id "K003", lalu Nama barang "Dining Chair" Mengisi Harga barang	<i>Berhasil, dan data akan ter simpan pada database lalu data secara otomatis akan tampil pada table alter</i>	Data berhasil ter simpan di dalam <i>database</i> lalu akan muncul pesan "data berhasil ter simpan" dan data	Sesuai

	"500000" dan jumlah barang "2" dengan total harga "1000000", lalu masukan tanggal transaksi "Oct 5,2021" jika sudah terisi semua selanjutnya kita tekan tambah.	<i>native.</i>	akan muncul pada <i>table alter native</i>	
P02	Memasukkan id dengan "T003", lalu Pilih id "K003", lalu Nama barang "Dining Chair" Mengisi Harga barang "500000" dan jumlah barang "2" dengan total harga "1000000", lalu masukan tanggal	Gagal, dan data tidak akan ter simpan kedalam <i>database</i> serta data tidak akan tampil pada <i>table alter native</i> .	Data gagal ter simpan lalu akan muncul pesan "terjadi ke salahan pada <i>database</i> " dan data tidak akan muncul pada <i>table alter native</i>	Sesuai

	transaksi "Oct 5,2021" jika sudah terisi semua selanjut nya kita tekan tambah.			
P03	Memasukan id dengan "010003", lalu Pilih id "K003", lalu Nama barang "Dining Chair" Mengisi Harga barang "Rp. 500000" dan jumlah barang "2" dengan total harga "Rp. 1000000", lalu masukan tanggal transaksi "Oct 5,2021" jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal, dan data tidak akan tersimpan ke dalam <i>database</i> serta data tidak akan tampil pada <i>table alter native</i> .	Data gagal tersimpan lalu akan muncul pesan "terjadi kesalahan pada <i>database</i> " dan data tidak akan muncul pada <i>table alter native</i>	Sesuai

P04	Memasukan id dengan "010003", lalu Pilih id "K003", lalu Nama barang "Dining Chair" Mengisi Harga barang "500000" dan jumlah barang "Dua" dengan total harga "1000000", lalu masukan tanggal transaksi "Oct 5,2021" jika sudah terisi semua selanjutnya kita tekan tambah.	Gagal, dan data tidak akan tersimpan ke dalam <i>database</i> serta data tidak akan tampil pada <i>table alter native</i> .	Data gagal tersimpan lalu akan muncul pesan "terjadi kesalahan pada <i>database</i> " dan data tidak akan muncul pada <i>table alter native</i>	Sesuai
P05	Memasukan id dengan "030004", lalu Pilih id "L004", lalu Nama barang "Lemari Buku" Mengisi	Berhasil, dan data akan tersimpan pada <i>database</i> lalu data secara otomatis akan tampil pada	Data berhasil tersimpan di dalam <i>database</i> lalu akan muncul pesan "data berhasil ter	Sesuai

	Harga barang "2000000" dan jumlah barang "2" dengan total harga "4000000", lalu masukan tanggal transaksi "Sep 8,2021" jika sudah terisi semua selanjut nya kita tekan tambah.	<i>table alter native.</i>	simpan" dan data akan muncul pada <i>table alter native</i>	
--	--	----------------------------	---	--

Jumlah percobaan yang diuji ada 10, dengan masing-masing percobaan yang diuji yaitu *form login* diuji sebanyak 2 kali, form produk barang diuji sebanyak 3 kali dan *form data penjualan* diuji sebanyak 5 kali. Adapun fungsi tombol yang juga diuji yaitu ada 4 fungsi yaitu fungsi masuk, tambah, *update*, *delete*. Dengan masing-masing fungsi yang diuji yaitu masuk 2 kali, *update* 2 kali, *delete* 2 kali, tambah 8 kali dengan total fungsi yang dicoba yaitu 14 kali. Jumlah *error* yang ditemukan sebanyak 0 fungsi dan fungsi yang tidak ditemukan *error* 4 fungsi.

#### IV. SIMPULAN

Pengujian aplikasi dengan *Blackbox Testing* bertujuan mencoba menjalankan program untuk mengetahui sistem kerja program tersebut tanpa harus mengetahui kode program yang dipakai. Berdasarkan pengujian kualitas aplikasi *furniture*, dapat ditarik kesimpulan bahwa pengujian dengan metode *Black Box* berbasis *Equivalence Partitions* dapat membantu proses pembuatan case pengujian, uji kualitas dan menemukan kesalahan yang tidak terdeteksi yang disebabkan oleh kesalahan pengetikan pada code aplikasi atau database. Dalam pengujian aplikasi

*furniture* tidak ditemukan kesalahan dalam fungsi tombol maupun percobaan yang dilakukan pada data yang sudah di masukan ke dalam aplikasi *furniture* ini. Untuk saat ini aplikasi *furniture* sudah dicoba dijalankan dengan hasil sebagaimana fungsinya, sehingga Diharapkan aplikasi ini dapat digunakan atau dikembangkan agar bisa bermanfaat untuk menyimpan data produk dan penjualan *furniture*.

#### DAFTAR PUSTAKA

- 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. *Jurnal Teknik Informatika UNIS JUTIS*, 6(1), 2252–5351. [www.ccsenet.org/cis](http://www.ccsenet.org/cis)
- Ningrum, F. C., Suherman, D., Aryanti, S., & Prasetya, H. A. (2020). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. 4(4), 125–130.
- Rahadi, N. W., & Vikasari, C. (2020). Pengujian Software Aplikasi Perawatan Barang Milik Negara Menggunakan Metode Black Box Testing Equivalence Partitions. *Infotekmesin*, 11(1), 57–61. <https://doi.org/10.35970/infotekmesin.v11i1.124>
- 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. <https://doi.org/10.32493/jtsi.v3i1.4303>