



PENGUJIAN APLIKASI TRANSAKSI PERDAGANGAN MENGUNAKAN *BLACK BOX TESTING BOUNDARY VALUE ANALYSIS*

Agus Utomo¹, Yusuf Sutanto², Erna Tiningrum³, Eko Meiningsih Susilowati⁴

Email : agu_tomo@yahoo.co.id¹, pixel_cmp@yahoo.co.id², tiningrum63@yahoo.co.id³,
susilowatieko7@gmail.com⁴

Abstract

The information system is one of the needs that is often used to facilitate human work. Jobs that often use information systems are office jobs, both in government and private agencies. Companies generally need an information system to simplify work such as creating cooperation documents, contract documents with external parties, as well as to help make monitoring and evaluation sheets for internal management. Like a system, to be able to store and display data as it should. Information system testing is needed in order to know the shortcomings of a system. The drawbacks of a system can be in the form of features that don't work properly or there are differences between the features expected and the features available. One of the tests of information systems is testing the Black Box method system by means of Boundary Value Analysis. Boundary Value Analysis is a method of testing by determining the lower and upper limit values of the data to be tested. This test is carried out on the function of adding new goods data to the trade transaction application. The test results are still lacking when validating the data entered into the system, causing the data stored in the database to not match what was expected. The results of this test can be an input to improve the application so that the eligibility requirements of the information system are better in accordance with the expected data entered into the system.

Key words: information system, value analysis, boundary value analysis

Pendahuluan

Aplikasi transaksi perdagangan pengolahan data transaksi penjualan ini merupakan aplikasi yang di buat untuk membantu instansi swasta / wirausaha swasta untuk mencatat, menyimpan dan mengolah data untuk mengetahui preferensi dari pembelian barang yang dilakukan oleh pembeli toko. Dengan aplikasi ini diharapkan dapat membantu menganalisa data penjualan, sehingga omset optimal dari perusahaan distribusi tersebut bisa dicapai.

Aplikasi ini terbagi menjadi beberapa modul, antara lain modul pembelian, modul penjualan, modul pelunasan hutang, modul pembayaran piutang, modul

pengeluaran operasional, modul tanda terima barang, dan lain-lain. Modul-modul tersebut terintegrasi satu sama lain menjadi satu kesatuan dalam sebuah aplikasi transaksi perdagangan untuk menyajikan data yang akurat, tepat dan terpercaya.

Aplikasi yang sudah selesai di buat memerlukan pengujian untuk memastikan input nilai yang dimasukkan sudah sesuai harapan sebelum tersimpan di dalam database. Pengujian adalah suatu proses uji coba suatu program dengan tujuan untuk menemukan kesalahan sehingga potensi gagal dalam memasukkan data kedalam database bisa diminimalisir. Suatu uji coba yang baik adalah uji coba tersebut mempunyai kemungkinan menemukan sebuah kesalahan atau lebih yang tidak terungkap, sebuah uji coba juga mendapatkan predikat baik apabila menemukan suatu kesalahan yang awalnya tidak ditemukan. Tujuan utama dari uji coba ini adalah untuk mendesain pengetesan secara sistematis menemukan jenis kesalahan dengan usaha dan waktu minimum, sehingga pada titik kesalahan tersebut pihak yang melakukan uji coba dapat mencatat pesan kesalahan dan mengirimkan kepada pihak developer untuk melakukan perbaikan pada tahap perbaikan program selanjutnya.

Landasan Teori

A. Sistem Informasi

Sistem Informasi adalah suatu kesatuan yang terdiri dari manusia (*brainware*), perangkat keras (*hardware*), perangkat lunak (*software*), jaringan komputer dan sumber daya data yang mengumpulkan dan mendistribusikan informasi dalam suatu organisasi (O'brien, 2008).

B. Pengujian software

Software memerlukan pengujian untuk memastikan bahwa *software* / aplikasi yang sudah atau sedang di buat dapat berjalan sesuai dengan fungsionalitas yang di harapkan. *Developer* atau pengembang harus menyiapkan seperangkat uji coba untuk menguji program yang sudah selesai di buat agar kekurangan atau kesalahan dapat di deteksi sejak awal dan di lakukan perbaikan pada siklus berikutnya. Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisahkan dari siklus hidup pengembangan perangkat lunak seperti halnya analisis, desain, dan pengkodean (Shi, 2010). Ada beberapa jenis pengujian perangkat lunak, antara lain (Khan, 2011):

1. Pengujian *white box* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara prosedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Secara sekilas dapat diambil kesimpulan *white box testing* merupakan petunjuk untuk mendapatkan program yang benar secara 100%,
2. *Black-Box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

C. Pengujian Black Box

Menurut pressmann (2010) metode uji coba *black box* memfokuskan pada keperluan fungsional dari *software*. Karena itu uji coba *black box* memungkinkan pengembang *software* untuk membuat himpunan kondisi input yang akan melatih seluruh syarat – syarat fungsional suatu program. Uji coba *black box* bukan merupakan alternatif dari uji coba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi

Tidak seperti metode *white box* yang dilaksanakan di awal proses, uji coba *black box* diaplikasikan di beberapa tahapan berikutnya. Karena uji coba *black box* dengan sengaja mengabaikan struktur kontrol, sehingga perhatiannya di fokuskan pada informasi domain.

Uji coba *black box* di desain untuk dapat menjawab pertanyaan-pertanyaan berikut :

- a. Bagaimana validitas fungsionalnya diuji ?
- b. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik?
- c. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
- d. Bagaimana batasan-batasan kelas data di isolasi?
- e. Berapa rasio data dan jumlah data yang dapat di toleransi oleh sistem ?
- f. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem ?

Dengan mengaplikasikan uji coba *black box*, diharapkan dapat menghasilkan sekumpulan kasus uji yang memenuhi kriteria berikut :

- a. Kasus uji yang berkurang, jika jumlahnya lebih dari 1, maka jumlah dari uji kasus tambahan harus di desain untuk mencapai uji coba yang cukup beralasan.
- D. Kasus uji yang memberitahukan sesuatu tentang keberadaan atau tidaknya suatu jenis kesalahan, dari pada kesalahan yang terhubung hanya dengan suatu uji coba yang spesifik.

E. Pengujian *Black Box Boundary Value Analysis*

Boundary value analisis adalah salah satu teknik *black box testing* yang melakukan pengujian pada batas atas dan batas bawah nilai yang diisikan pada aplikasi. Beberapa prinsip yang mendasari pada *boundary value analysis* (BVA) yaitu :

- a. Banyak kesalahan terjadi pada kesalahan masukan.
- b. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input.

- c. BVA merupakan komplemen dari *equivalence partitioning*. Lebih pada memilih elemen-elemen di dalam kelas ekivalen pada bagian sisi batas dari kelas.

Sebagai Contoh :

- a. Untuk rentang yang dibatasi a dan b maka uji (a-1), a, (a+1), (b-1), b, (b+1).
- b. Jika kondisi input mensyaratkan sejumlah n nilai maka uji dengan sejumlah (n-1), n dan (n+1) nilai.
- c. Aplikasikan dua aturan sebelumnya pada kondisi output (buat table pengujian hasil outputnya untuk nilai maksimal dan minimal).
- d. Jika struktur data internal dari program memiliki cakupan (misal: ukuran *buffer*, batas array) gunakan data input yang menguji batas cakupan.
- e. Secara umum, aplikasi BVA dapat dikerjakan secara *generic*. Bentuk dasar implementasi BVA adalah untuk menjaga agar satu variable berada pada nilai nominal (normal atau rata-rata) dan mengijinkan variable lain diisikan dengan nilai ekstrimnya. Nilai yang digunakan untuk menguji keekstriman data adalah:

Min ----- minimal

Min+ ----- di atas minimal

Nom ----- rata-rata

Max- ----- tepat di bawah maksimum

Max ----- Maksimum

Sebagai contoh, misalnya akan dimasukkan data tanggal. Data tanggal memiliki tiga variable yaitu tanggal, bulan dan tahun. Maka untuk ketiga variable tersebut, dapat diambil kondisi berikut :

$$1 \leq \text{tanggal} \leq 31$$

$$1 \leq \text{bulan} \leq 12$$

$$1812 \leq \text{tahun} \leq 2016$$

Maka untuk setiap entri data di luar angka di atas akan menampilkan pesan “Tanggal yang anda isikan salah”. (M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi, 2015)

F. Pengujian Black Box Boundary Value Analysis

Berdasarkan pada penjelasan awal bab sebelumnya, penerapan teknik BVA telah di coba untuk pengujian aplikasi yaitu aplikasi transaksi perdagangan pengolahan data barang pada perusahaan dagang. Pada aplikasi tersebut terdapat beberapa modul, tetapi pada artikel ini akan dibahas fokus pada pengujian modul “Penambahan master barang” yang di jual.



Login

General | Connection

Nama Operator

Password

Login Batal

Gambar 1. Form Login Admin'

Barang

Golongan: Accessories

Sub Golongan: Adaptor

Kode: 20190721001192 Barcode: 20190721001192

Nama:

Supplier:

Satuan: #1 BIJI 1 Harga Jual #1 0

#2 0 Harga Jual #2 0

#3 0 Harga Jual #3 0

#4 0 Harga Jual #4 0

Stok Awal: 0

Stok: 0

Harga Beli: 0

Minimum Stok: 0

Maksimal Stok: 0

Garansi:

Poin:

Lokasi: GDG1 - GUDANG JOGA

Diskon Member

General: 0

Silver: 0

Gold: 0

☐ Boleh ganti harga jual ☐ Wajib Isi Imei

F2 = Simpan ESC = Batal

Gambar 2. Form Input Kode Barang baru

Tabel 1. Struktur tabel kunjungan pasien

Kode (Varchar 25)	Nama (Varchar 50)	Satuan (Varchar 10)	Stok (Varchar 9)	Min_Stok (int 13)	Max_Stok (Int 13)	Harga_Beli (Int 18)
Harga_Jual (Int 18)	Golongan_id (varchar 6)	Lokasi_id (varchar 6)	Supplier_id (varchar 10)	K_Barcode (varchar 20)	Stok_Awal (Int 13)	Diskon_Rp (Int 13)
Garansi (varchar 10)	Sub_Gol_id (varchar 6)	Biji 1 (Int 5)	Satuan 2 (varchar 10)	Biji 2 (Int 5)	Satuan 3 (varchar 10)	Biji 3 (Int 5)
Satuan 4 (varchar 10)	Biji 4 (Int 5)	Diskon_Gen (Int 13)	Diskon_Silvr (Int 13)	Diskon_Gold (Int 13)	Point (Int 13)	Wjb_Isi_Imei (Int 5)

Berdasarkan struktur table diatas, kemudian dilakukan pengujian dengan menyiapkan beberapa data uji. Dari bentuk form diatas, contoh pengujian akan dilakukan pada satu field yaitu Satuan 1, Satuan 2, Satuan 3, Satuan 4 pada Form Barang.

Pengujian field Satuan 1

Aturan entri data A1 terdiri karakter huruf

Tabel 2. Pengujian Baris Satuan 1 dengan huruf

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
Lusin	T	T	Sukses
Lu8sin	F	T	Sukses
Lu sin	F	T	Sukses

Aturan entri data A1 terdiri karakter angka

Tabel 3. Pengujian Baris Satuan 1 dengan angka

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
123456	F	T	Sukses
123_456	F	T	Sukses
1 2345	F	T	Sukses

Pengujian field Satuan 2

Aturan entri data A1 terdiri karakter huruf

Tabel 4. Pengujian Baris Satuan 2 dengan huruf

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
Kodi	T	T	Sukses
Ko9di	F	T	Sukses
K odi	F	T	Sukses

Aturan entri data A1 terdiri karakter angka

Tabel 5. Pengujian Baris Satuan 2 dengan angka

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
1357	F	T	Sukses
12_789	F	T	Sukses
1 2357	F	T	Sukses

Pengujian field Satuan 3

Aturan entri data A1 terdiri karakter huruf

Tabel 6. Pengujian Baris Satuan 3 dengan huruf

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
Karung	T	T	Sukses
Karu7ng	F	T	Sukses
Karu ng	F	T	Sukses

Aturan entri data A1 terdiri karakter angka

Tabel 7. Pengujian Baris Satuan 3 dengan angka

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
56789	F	T	Sukses
345_456	F	T	Sukses
335 2345	F	T	Sukses

Pengujian field Satuan 4

Aturan entri data A1 terdiri karakter huruf

Tabel 8. Pengujian Baris Satuan 4 dengan huruf

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
Gross	T	T	Sukses
Gr1oss	F	T	Sukses
Gro oss	F	T	Sukses

Aturan entri data A1 terdiri karakter angka

Tabel 9. Pengujian Baris Satuan 4 dengan angka

Contoh Data	Seharusnya	Hasil Nyata	Kondisi
15889	F	T	Sukses
567_897	F	T	Sukses
67 8989	F	T	Sukses

Berdasarkan hasil uji pada satu form di atas, dapat disiapkan beberapa kasus data uji. Pada contoh diatas, terdapat satu field yang akan diuji. Masing-masing field pada tabel minimal memuat dua aturan untuk di uji. Pada satu tabel, perlu disiapkan 3 data yang diuji, sehingga total data yang akan di uji pada soal kasus diatas adalah 1 baris x 2 aturan x 3 data uji, totalnya 6 data uji. Pada penelitian ini menguji 4 tabel, jadi data yang diuji menjadi 4 tabel x 1 baris x 2 aturan x 3 data uji, totalnya 24 data uji.

Berdasarkan penjelasan diatas menyebutkan, jumlah data uji yang harus disiapkan untuk melakukan *blackbox testing* dengan metode BVA. Hasil pengujian memperlihatkan bahwa aplikasi ini masih memiliki kekurangan, yaitu belum sempurnanya validasi data sehingga masih perlu di perbaiki dengan menambah fungsi validasi.

G. Kesimpulan

Setelah melakukan pengujian pada input satuan barang dengan menggunakan metode pengujian *BlackBox Testing Boundary Value Analysis* dapat di tarik kesimpulan bahwa :

1. Metode *Black Box Testing Boundary Value Analysis* merupakan salah satu metode pengujian yang mudah dipahami karena hanya memerlukan batas atas dan batas bawah dari data yang di gunakan.
2. Hasil uji coba dengan metode *Boundary Value Analysis* menunjukkan bahwa fungsi input data masih perlu di sempurnakan dengan beberapa fitur validasi data untuk mengatasi kekurangan kesalahan supaya data yang dimasukkan kedalam sistem menjadi benar, seperti halnya pada waktu melakukan input huruf salah menjadi input angka.
3. Perkiraan banyaknya data yang diuji bisa dihitung melalui banyaknya baris input data yang akan diuji, aturan input harus di penuhi serta batasan nilai atas dan nilai bawah yang memenuhi.
4. Setelah melakukan beberapa uji coba di dapatkan hasil bahwa fungsionalitas masih bisa berjalan namun masih dapat menerima masukan data yang tidak diharapkan, sehingga mengakibatkan data yang di simpan dalam database menjadi kurang tepat / tidak sesuai yang di harapkan.

Daftar Pustaka

O' Brien, James, 2008. *Pengantar Sistem Informasi*. Jakarta : Salemba Empat.

Khan, Mohd Ehmer, 2011, Different Approach to Blackbox Testing Technique for Finding Error, *International Journal of Software Engineering & Applications (IJSEA)*, Vol.2, No.4.

Mustaqbal M.S.; Firdaus R.F.; Rahmadi H. 2015. *Jurnal Ilmiah Teknologi Informasi Terapan*. Volume I, No 3.

Pressman, Roger S, 2010, *Pendekatan Praktisi Rekayasa Perangkat Lunak*. Edisi 7. Yogyakarta . Penerbit Andi.

Shi, Mingtao, 2010, Software Functional Testing from the Perspective of Business Practice Computer and Information Science, www.ccsenet.org/cis