

IMPLEMENTASI PENGUJIAN APLIKASI PERHITUNGAN BIAYA PRODUKSI PADA RUMAH MAKAN PADANG MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS

Vindi Kusuma Putri Arifin¹, Rochmawati², Monterico Adrian³

^{1,2,3}D3 Sistem Informasi Akuntansi, Fakultas Ilmu Terapan, Universitas Telkom

Email: [1vindiarku@gmail.ac.id](mailto:vindiarku@gmail.ac.id), [2rochmawati@tass.telkomuniversity.ac.id](mailto:rochmawati@tass.telkomuniversity.ac.id), [3monterico.adrian@gmail.com](mailto:monterico.adrian@gmail.com).

(Naskah masuk: dd mmm yyyy, diterima untuk diterbitkan: dd mmm yyyy)

Abstrak

Sebuah proses yang harus dilakukan setelah aplikasi selesai dibangun adalah pengujian yang dilakukan dengan menjalankan program untuk menemukannya kesalahan. Kasus uji yang baik ditunjukkan dengan kasus uji tersebut dapat membongkar kesalahan sistem yang awalnya tidak ditemukan. *Black box testing* adalah salah satu dari teknik pengujian yang ada. Pada penelitian ini diterapkan pengujian dengan teknik *black box testing*. Yaitu sebuah teknik yang terdiri dari beberapa cara, diantara metode yang ada pada pengujian ini menggunakan metode *boundary value analysis* (BVA). *Boundary value analysis* (BVA) adalah metode pengujian dengan menentukan batas bawah dan batas atas suatu data yang akan diuji. Pada penelitian ini dilakukan pengujian pada fungsionalitas menambah master data bahan baku dalam aplikasi perhitungan biaya produksi pada rumah makan padang. Aplikasi yang diuji berikut berbasis web dengan bahasa pemrograman PHP. Hasil dari pengujian yang telah dilaksanakan menunjukkan bahwa terdapat kesalahan yaitu saat melakukan validasi data yang dimasukkan, hal ini menyebabkan ketidaksesuaian data yang tersimpan pada basis data dengan data yang diharapkan. Oleh karena itu hasil pengujian ini bisa menjadi masukan pada sistem.

Kata kunci: pengujian, *black box testing*, biaya produksi, *boundary value analysis*

IMPLEMENTATION OF APPLICATION TESTING OF PRODUCTION COST CALCULATION AT PADANG RESTAURANT USING BLACK BOX TESTING BOUNDARY VALUE ANALYSIS

Abstract

A process that must be carried out after the application is completed is the testing carried out by running the program to find errors. A good test case indicated by the test case can uncover system errors that were not initially found. *Black box testing* is one of the existing testing techniques. In this research, testing was carried out with *black box testing* techniques. That is a technique that consists of several ways, among the methods that exist in this test using the *boundary value analysis* (BVA) method. *Boundary value analysis* (BVA) is a testing method by determining the lower and upper limits of the data to be tested. In this study, testing the functionality of adding master data on raw materials in the application of the calculation of production costs at the Padang restaurant. The following applications are web-based with the PHP programming language. The results of the tests that have been carried out show that there is an error that is when validating the data entered, this causes incompatibility of data stored in the database with the expected data. Therefore the results of this test can be recommendation to the system.

Keywords: testing, *black box testing*, costing, *boundary value analysis*

1. PENDAHULUAN

Aplikasi Perhitungan Biaya Produksi pada Rumah Makan Padang adalah aplikasi berbasis web yang dibuat untuk membantu pemilik rumah makan padang dalam menghitung biaya produksi setiap satuan jenis produk/makanan yang dihasilkan. Dengan aplikasi ini setiap komponen biaya produksi dicatat sehingga menghasilkan biaya produksi yang

tepat. Pada aplikasi ini terdapat 2 tahapan dalam menghitung biaya produksi, yaitu tahap menambahkan komponen biaya produksi dan tahap menghitung *unit ekivalensi*. Perhitungan biaya produksi yang tidak tepat berimbas terhadap penentuan harga jual yang tidak tepat pula. Oleh karena itu aplikasi ini sangat bermanfaat untuk menentukan biaya produksi setiap satu unit masakan pada rumah makan padang.

2 Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), Vol. x, No. x, April 2014, hlm. x-y

Penelitian sebelumnya yang berkaitan dengan penelitian ini adalah pengujian aplikasi menggunakan *black box testing boundary value analysis* (MUSTAQBAL, et al., 2015). Sedangkan penelitian mengenai biaya produksi adalah Aplikasi Perhitungan Biaya Produksi Menggunakan Metode Harga Pokok Proses Berbasis Web (Studi Kasus Di Ruby's Cake, Bekasi) (FEBRIANTO, et al., 2018).

Setelah aplikasi selesai dibuat, maka perlu dilakukan pengujian untuk memastikan semua proses berjalan sesuai dengan yang diinginkan. Pengujian adalah suatu proses pelaksanaan suatu program dengan tujuan menemukan suatu kesalahan. Kasus uji yang baik adalah apabila kasus uji tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak terungkap. Suatu test yang sukses adalah apabila *test* tersebut membongkar suatu kesalahan yang awalnya tidak ditemukan. Tujuan utama dari pengujian adalah untuk mendesain *test* yang secara sistematis membongkar jenis kesalahan dengan usaha dan waktu singkat.

2. KAJIAN TEORI

Berikut merupakan kajian teori yang digunakan dalam aplikasi:

2.1 Akuntansi

Akuntansi merupakan keterampilan dalam pencatatan, pengelompokan, pengikhtisaran dan pelaporan yang dibuat dalam laporan keuangan. Laporan keuangan berisi sumber-sumber informasi yang berguna untuk pengambilan keputusan bagi pihak-pihak yang membutuhkan (LUBIS, 2017).

2.2 Metode Biaya Produksi

Metode akuntansi yang digunakan dalam penelitian ini adalah *Process costing* yaitu sebuah metode perhitungan biaya yang digunakan pada saat produk yang dihasilkan bersifat homogen (sejenis) dan kontinu (berkelanjutan/terus menerus) (DEWI & KRISTANTO, 2013).

2.3 Black Box Testing

Black box testing merupakan pengujian secara fungsionalitas dari sistem yang bertujuan untuk mengecek penerapan dari spesifikasi sistem yang telah direncanakan sebelumnya. *Black box testing* cocok digunakan untuk *system and user acceptance test*. Template isian *black box testing* berbentuk tabel yang berisi ID kasus pengujian, kasus pengujian, data masukan, fungsionalitas yang diharapkan, hasil pengamatan, serta kesimpulan (DESAI & SRIVASTAVA, 2016).

Black Box Testing cenderung untuk menemukan hal-hal berikut (MUSTAQBAL, et al., 2015):

1. Fungsi yang tidak ada atau tidak sesuai.
2. *Interface errors* kesalahan antarmuka.
3. Kesalahan struktur data dan akses pada basis data.
4. *Performance errors* (kesalahan performansi).
5. Kesalahan untuk inialisasi dan terminasi.

Pengujian didesain guna menemukan jawaban atas pertanyaan-pertanyaan ini:

1. Bagaimana fungsionalitas-fungsionalitas diuji agar dinyatakan valid?
2. Masukan seperti apa yang bisa menjadi data kasus uji yang baik?
3. Apakah sistem sensitif terhadap masukan tertentu?
4. Apakah kumpulan data dapat diisolasi?
5. Berapa banyak jumlah data yang bisa ditangani sistem?
6. Efek apa yang membuat kombinasi data ditangani secara khusus pada saat operasi sistem?

Saat ini banyak metode untuk melaksanakan *Black Box Testing*, antara lain:

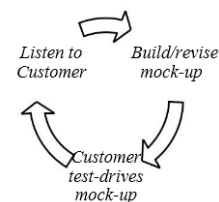
1. *Behavior Testing*
2. *Boundary Value Analysis/Limit Testing*
3. *Cause-Effect Relationship Testing*
4. *Comparison Testing*
5. *Endurance Testing*
6. *Equivalence Partitioning*
7. *Performance Testing*
8. *Sample Testing*
9. *Requirement Testing*
10. *Robustness Testing*

3. METODE

Berikut merupakan metode-metode yang digunakan dalam aplikasi:

3.1 Metode Penelitian

Pendekatan pengembangan sistem menggunakan metode prototipe (*prototyping model*). Metode *prototype* digunakan untuk menyelesaikan kesalahpahaman antara pengguna dan penulis yang timbul akibat pengguna tidak dapat mendefinisikan secara jelas kebutuhannya. Model ini diperbaiki secara terus-menerus sampai dengan kebutuhan pengguna terpenuhi (PRESMAN, 2012).



Gambar 1 Siklus Prototipe

a. Listen to Customer

Adalah tahap mendengarkan dan berkomunikasi dengan pelanggan (*client*) dalam mendefinisikan

kebutuhan pengguna yang akan dibangun pada perangkat lunak.

b. *Build/Revise Mockup*

Adalah tahap membuat rancangan yang dapat mewakili seluruh bagian perangkat lunak.

c. *Customer Test Drives Mockup*

Adalah tahap saat pengguna menguji aplikasi yang sudah dibuat dan mengevaluasi *prototype* yang dibuat untuk memperjelas kebutuhan perangkat lunak.

3.2 Boundary Value Analysis

Salah satu metode atau teknik dalam pengujian *black box testing* adalah *boundary value analysis* yaitu pengujian dengan menentukan batas atas dan batas bawah nilai yang diisikan pada aplikasi sebagai dasar pengujian. Prinsip-prinsip yang mendasari pengujian *boundary value analysis* (BVA) adalah:

1. Banyak kesalahan/*errors* yang terjadi pada kesalahan *input/masukan*.

2. BVA memberikan izin untuk menyeleksi kasus uji (*test case*) yang menguji batasan nilai *input/masukan*.

3. BVA merupakan komplemen dari salah satu metode atau teknik *black box testing* yaitu *equivalence partitioning*. Terlihat pada saat memilih elemen di dalam kelas ekivalen pada bagian batas dari kelas.

4. Contoh:

a. Aplikasikan dua aturan sebelumnya pada keluaran/*output* (membuat tabel pengujian hasil keluaran untuk nilai maksimal dan minimal).

b. Untuk rentang yang dibatasi a dan b maka uji (a-1), a, (a+1), (b-1), b, (b+1).

c. Jika kondisi input mensyaratkan jumlah n nilai maka uji dengan jumlah (n-1), n dan (n+1) nilai.

d. Jika struktur data internal dari program memiliki cakupan misalnya batas *array* maka digunakan data *input/masukan* yang menguji batas cakupan yang digunakan untuk menguji keekstriman data adalah:

Table 1 Dasar Uji Keekstriman Data

Min	Minimal
Min+	Di atas minimal
Nom	Rata-rata
Max-	Tepat dibawah maksimum
Max	Maksimum

Berikut merupakan contoh, akan dimasukkan data umur. Data umur memiliki tiga variabel yaitu tanggal, bulan dan tahun. Maka untuk ketiga variabel tersebut, dapat diambil kondisi seperti ini:

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

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

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

Maka untuk setiap masukan data di luar angka di atas akan menampilkan pesan kesalahan "Tanggal yang

anda isikan tidak sesuai" (MUSTAQBAL, et al., 2015).

4. IMPLEMENTASI

Berikut merupakan implementasi aplikasi:

1.

Gambar 2 Antarmuka Entri Data Bahan Baku

Berdasarkan penjelasan sebelumnya, dicoba diterapkan teknik BVA untuk menguji aplikasi yaitu "Aplikasi Perhitungan Biaya Produksi pada Rumah Makan Padang". Aplikasi ini sendiri memuat beberapa fungsionalitas, namun pada penelitian ini sebagai contoh akan dibahas hasil pengujian pada salah satu fungsionalitas yaitu "Menambah master data bahan baku" yang terdiri atas satu tampilan entri data seperti pada gambar 2. Pada form ini terdapat 3 *field* entri data yaitu nama bahan baku, satuan bahan baku dan harga.

Berikut ini merupakan batasan dan tipe data pada fungsi menambah master data bahan baku:

Tabel 1 Batasan Data

NAMA KOLOM	TIPE	BATASAN DATA
Nama Bahan Baku	Varchar	Data bisa diisi berupa karakter dan angka, minimal karakter 1, maksimal berisi 25 karakter
Satuan Bahan Baku	Varchar	Satuan hanya bisa dipilih, tidak <i>diinput</i>
Harga	Int	Harga harus terisi, tidak boleh kosong

Tabel diatas menjelaskan tipe data yang digunakan dan batasan data dari tipe data tersebut. Misalnya pada kolom nama bahan baku mempunyai tipe data *varchar* dan batasan data minimal panjang data satu karakter dengan batas panjang data maksimal 25 karakter.

4 Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK), Vol. x, No. x, April 2014, hlm. x-y

Berikut ini merupakan identifikasi batasan (BVA) dari setiap field masukan:

Tabel 2 Identifikasi BVA

No	Field name	Ket	BVA	Value		Input Data	Valid
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	Nama Bahan baku	BVA	Min	1	Min-1		T
					Min	b	V
		BVA	Max	25	Max-1	Daging sapi super import	V
					Max	Daging sapi super import.	V
					Max+1	Daging sapi super import xl	T
2	Satuan bahan baku	BVA	Tidak bisa diuji karena berupa dropdown	-	-	-	-
3.	Harga Satuan	BVA	Slab	1	Slab-1	0	T
					Slab	1	V
					Slab+1	2	V

Pada tabel identifikasi BVA berisi identifikasi BVA beserta nilai masukannya. Input data adalah nilai yang diidentifikasi, sedangkan hasil identifikasi terdapat pada kolom valid. Pada kolom valid keterangan T menunjukkan bahwa data *input* atau nilai masukan tidak valid yang seharusnya ditolak oleh sistem. Sedangkan keterangan V menunjukkan bahwa masukan valid atau data yang dapat diterima sistem.

Tahap terakhir yaitu, mencocokkan skenario program saat diinputkan data, pada tahap ini membandingkan kesesuaian antara hasil yang diharapkan dengan hasil aktual. Jika sesuai maka statusnya adalah berhasil dan jika tidak sesuai maka dikolom status ditulis gagal.

Tabel 3 Test Case (Kasus Uji)

Test No	Test Case	Input Data	Result		Status
			Expected	Actual	
1	Mengisi form nama		sistem akan menampilkan pesan error "isi form"	sistem menampilkan pesan error "isi form"	berhasil

Test No	Test Case	Input Data	Result		Status
			Expected	Actual	
	bahan baku		nama bahan baku"	nama bahan baku"	
		b	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		bb	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		daging sapi import	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
2	satuan bahan baku	tidak dapat diuji dengan BVA karena inputan berupa dropdown			
3	Mengisi form Harga	0	sistem tidak menerima masukan dan menampilkan pesan error "harga tidak boleh 0"	sistem akan menerima data dan menyimpan kedalam basis data	tidak berhasil
		1	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil
		2	sistem akan menerima data dan menyimpan kedalam basis data	sistem akan menerima data dan menyimpan kedalam basis data	berhasil

Pada tabel *Test Case* (Kasus Uji) terdapat 5 kolom. Kolom *Tes No.* menunjukkan nomer kasus uji (nama form yang diisi). Pada kolom *Test Case* berisi kasus uji yang dijalankan. Kolom *Input Data* adalah nilai yang diujikan. Kolom *Result* terbagi menjadi dua bagian yaitu *expected* (hasil yang diharapkan) dan *actual* (hasil sesungguhnya). Kolom status menunjukkan berhasil tidaknya sistem, status berhasil ditulis apabila *expected result* sesuai dengan *actual result*.

5. KESIMPULAN

Berdasarkan hasil pengujian manual dan pengujian aplikasi dengan menggunakan teknik *black box testing* dan metode *boundary value analysis*, hasil yang ditunjukkan menghasilkan hasil yang berbeda sebesar 22% atau 2 dari sembilan masukan yang diuji, sehingga dapat disimpulkan bahwa pengujian aplikasi ini berhasil karena dapat membongkar kesalahan sistem yang sebelumnya tidak diketahui.

6. DAFTAR PUSTAKA

- DESAI, S. & SRIVASTAVA, A., 2016. *Software Testing: A Practical Approach, Second Edition*. Delhi: phi learning Private Limited.
- FEBRIANTO, A., SUKAWATI, R. & KASTAMAN, 2018. *Aplikasi Perhitungan Biaya Produksi Menggunakan Metode Harga Pokok Proses Berbasis Web (Studi Kasus Di Ruby's Cake, Bekasi)*. Bandung: Proyek Akhir Tidak Dipublikasikan.
- LUBIS, R. H., 2017. *Pengantar Akuntansi Jasa Berbasis SAK IFRS dan ETAP*. Yogyakarta: Penerbit Gava Media.
- MUSTAQBAL, M. S., FIRDAUS, R. F. & RAHMADI, H., 2015. Pengujian Aplikasi Menggunakan Black Box Testing Boundry Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). *Jurnal Ilmiah Teknologi Informasi Terapan*, pp. 4-5.
- PRESMAN, R. S., 2012. *Rekayasa Perangkat Lunak*. Yogyakarta: Andi Publisher.