

# PENGUJIAN APLIKASI PERHITUNGAN STOK BARANG DENGAN METODE BLACKBOX PADA CV. DELTA PILAR (CABANG BANDUNG)

## *SOFTWARE TESTING FOR INVENTORY STOCK APPLICATION WITH BLACKBOX METHOD AT CV. DELTA PILAR (BANDUNG BRANCH)*

Disa Arista Syuhada<sup>1</sup>, Agus Nursikuwagus<sup>2</sup>

<sup>1</sup> Universitas Komputer Indonesia

<sup>2</sup> Universitas Komputer Indonesia

Email : disaada@gmail.com

**Abstrak** - Aplikasi perhitungan stok barang merupakan aplikasi menghitung stok barang secara otomatis berdasarkan jumlah pemesanan. Dalam proses pengembangannya, aplikasi ini belum sampai kepada tahap pengujian. Jika melihat pada System Development Life Cycle (SDLC), pengujian merupakan tahap penting yang perlu dilakukan sebelum masuk ke tahap implementasi. Oleh karena itu, perlu adanya sistematisa pengujian pada aplikasi ini, mendokumentasikan seluruh proses pengujian yang kemudian menghasilkan rekomendasi perbaikan. Metode penelitiannya adalah kualitatif. Pengujian menggunakan metode Blackbox dengan model *Boundary Value Analysis (BVA) - Equivalence Partitioning (EP)* dan *Decision Table (DT)* sebagai pembanding. Hasil dari penelitian ini menunjukkan bahwa aplikasi memiliki kekurangan sebesar 43,4% dengan teknik BVA+EP dan 53,5% dengan teknik DT.

**Kata kunci** : pengujian, *blackbox*, *boundary value analysis*, *equivalence partitioning*, *decision table*

**Abstract** - *Inventory stock calculation application is an application to automatically calculate the stock of goods based on the number of orders. In the development process, this application has not yet arrived at the testing stage. If we look at the System Development Life Cycle (SDLC), testing is an important stage that needs to be done before entering the implementation stage. Therefore, there is a need for making a systematics testing on this application, documenting the entire testing process which then produces recommendations. The research method is qualitative. The test uses the Blackbox method with a Boundary Value Analysis (BVA) - Equivalence Partitioning (EP) and Decision Table (DT) models as a comparison. The results of this study indicate that the application has a deficiency of 43.4% with the technique of BVA + EP and 53.5% with DT techniques.*

**Keyword** : *testing*, *blackbox*, *boundary value analysis*, *equivalence partitioning*, *decision table*

## I. PENDAHULUAN

Hingga hari ini, spektrum pengujian aplikasi pada sebuah sistem informasi semakin meluas dan bervariasi. Pengujian bisa dilakukan pada *output*, *input*, atau proses aplikasi. Hampir semua model pengembangan rekayasa perangkat lunak mempunyai tahapan pengujian. Sehingga dapat dikatakan pengujian juga merupakan pembahasan yang vital dalam membangun sebuah aplikasi atau sistem informasi.

Aplikasi Perhitungan Stok Barang pada CV. Delta Pilar adalah aplikasi yang berfungsi untuk menghitung jumlah stok barang berdasarkan jumlah pesanan. Aplikasi ini membantu pegawai dalam menghitung jumlah stok yang tersisa berdasarkan jumlah pesanan yang ada. Aplikasi ini dirancang dengan metode *Waterfall* dan dibangun dengan pendekatan berbasis objek.

Jika kita melihat tahapan pembangunan sistem pada metode *Waterfall* pada tahapan pengkodean (*coding*), maka tahapan berikutnya dilanjutkan kepada tahapan pengujian (*testing*). Pada praktiknya, pembangunan aplikasi pada CV. Delta Pilar (Cabang Bandung) ini belumlah sampai pada tahap pengujian. Tahap pengujian perlu dilakukan agar aplikasi yang digunakan oleh karyawan instansi adalah aplikasi yang benar-benar siap pakai dan terbebas atau minimal kecil kemungkinan dari kesalahan teknis dan disfungsi aplikasi.

Pada penelitian dengan jenis kualitatif ini akan dilakukan pengujian pada Aplikasi Perhitungan Stok Barang dengan model pengujian V-Model, metode pengujian Blackbox, dan batasan pengujian yang sudah ditentukan. Hasil akhir dari penelitian ini adalah perhitungan, evaluasi, rekomendasi untuk Aplikasi Perhitungan Stok Barang. Hasil penelitian ini dapat digunakan untuk kepentingan akademis maupun praktis.

## II. KAJIAN PUSTAKA

### A. Penelitian Terdahulu

Aplikasi Perhitungan Stok Barang merupakan hasil kerja praktek yang dilakukan oleh Ardhiana Nur Hakim dan Disa Arista Syuhada pada CV. Delta Pilar (Cabang Bandung) dengan judul laporan *Aplikasi Perhitungan Stok Barang Berbasis Website Pada CV. Delta Pilar (Cabang Bandung)*. Pembuatan aplikasi ini berawal dari identifikasi masalah pada instansi dimana instansi belum memiliki sistem yang terintegrasi dengan *internet*, sehingga proses bisnisnya masih dengan pencatatan manual.

Aktivitas yang berjalan seperti ketika ada pelanggan yang melakukan order, salah satu staff admin hanya mencatat di catatan *handphone* yang berisi data pembeli, produk yang diorder, dan jumlah order, tulisan order masih ditulis dalam nota, salah satu staff *packing* menghitung setiap produk berapa jumlah yang disorder, lalu dikurangkan dengan stok yang ada, lalu dihitung jumlah barang apakah sesuai dengan hasil perhitungan jumlah barang sebelumnya, pada langkah ini salah satu dari staf melakukan kesalahan dalam menghitung, sehingga terjadi tidak sesuai jumlah barang yang ada dalam catatan.[1]

Penelitian berjudul *Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus: Kantor Digital Politeknik Negeri Lampung)* oleh Tri Snadhika Jaya bertujuan untuk menguji perangkat lunak kantor digital di Politeknik Negeri Lampung. Metode yang digunakan adalah Blackbox dengan dengan teknik *Boundary Value Analysis*. Hasil penelitian menunjukkan bahwa aplikasi mampu menangani data, baik data normal maupun data tidak normal ataupun tidak normal dengan presentase keberhasilan 91,67%. [2]

Penelitian yang dilakukan oleh M. Sidi Mustaqbal, Roeri Fajri Firdaus, dan Hendra Rahmadi yang berjudul *'Pengujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)'* bertujuan untuk menemukan kesalahan pada Aplikasi Prediksi Kelulusan SNMPT. Peneliti menggunakan metode Blackbox dengan teknik *Boundary Value Analysis*. [3]

Hasil penelitian menunjukkan bahwa form Tambah Kelas berfungsi dengan baik, namun masih menerima masukan data yang tidak diharapkan sehingga menyebabkan data yang tersimpan kurang valid. Penelitian ini dengan penelitian yang dilakukan oleh M. Sidi Mustaqbal, Roeri Fajri Firdaus, dan Hendra Rahmadi sama-sama menggunakan *Boundary Value Analysis* sebagai teknik pengujian. Perbedaannya adalah penelitian ini juga menggunakan teknik *Decision Table* sebagai pembandingnya.

Penelitian yang berjudul *'Pengujian Sistem Informasi Akuntansi Biaya Operasional Sekolah Dengan Black Box Testing'* oleh Mursyidah dan Hari Toha Hidayat bertujuan untuk menguji Aplikasi SIA BOS. Hasil dari penelitian ini menunjukkan bahwa tidak ditemukan permasalahan pada sistem. [4]

Penelitian yang dilakukan oleh Umi Salamah dan Fata Nidaul Khasanah dengan judul *'Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing'* bertujuan untuk menguji SIP Undangan Pernikahan. Hasil penelitian menunjukkan bahwa sistem berjalan sesuai harapan / sesuai dengan tujuan. [5]

### B. Pengujian aplikasi

Pengujian itu sangat penting karena merupakan bagian dari pengembangan perangkat lunak. Pengujian digunakan untuk memastikan kualitas aplikasi yang telah terbuat. Tujuan lain dalam pengujian adalah untuk memastikan bahwa aplikasi berjalan dengan baik tanpa kesalahan dan memungkinkannya dibangun kembali. [6]

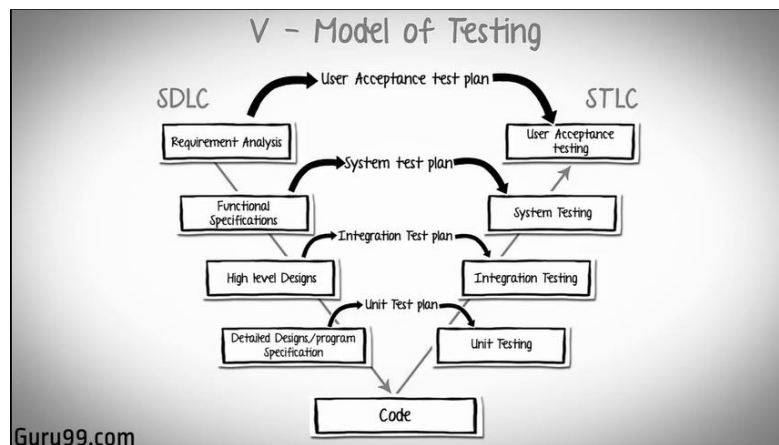
*International Software Testing Qualification Board (ISTQB)* menjelaskan definisi dari pengujian aplikasi sebagai sebuah proses (*process*) yang mencakup semua siklus hidup aktivitas (*all life cycle activities*) dalam pembangunan aplikasi, baik statis maupun dinamis, dimana fokus utama dari pengujian adalah pada perencanaan pengujian (*planning*), persiapan (*preparing*) dan evaluasi aplikasi (*evaluation of software product*) dan hubungannya dengan kinerja aplikasi (*related work product*) untuk menentukan syarat-syarat spesifik kepuasan pengguna dalam menggunakan aplikasi, kemudian mendemonstrasikan aplikasi tersebut agar aplikasi sesuai dengan persyaratan-persyaratan yang sudah ditentukan, dan kemudian mencari kecacatan didalamnya. [7]

1. Proses (*process*). Pengujian aplikasi disebut sebagai proses karena ia bukanlah aktivitas tunggal (*single activity*), melainkan ada beberapa tahapan-tahapan aktivitas yang harus dijalankan.
2. Mencakup semua siklus hidup aktivitas (*all life cycle activities*). Sepanjang siklus hidup pengembangan sistem (*Software Development Life Cycle*), pengujian aplikasi benar-benar mengambil seluruh tahapan dalam SDLC. Pengujian dilakukan dari tahapan awal dalam membangun sistem yaitu tahap Perencanaan hingga tahapan terakhir yaitu Implementasi. Bayangkan jika pengujian hanya dilakukan di tahap Implementasi saja, maka akan semakin besar pembiayaan yang dibutuhkan untuk memperbaiki kecacatan. Jumlah kecacatannya akan jauh lebih banyak dibanding yang sudah melakukan pengujian sejak tahap awal.
3. Baik statis maupun dinamis (*both static and dynamic*). Kita dapat melakukan pengujian dengan cara menjalankan aplikasi (*dynamic testing*) atau tanpa harus menjalankan aplikasi tersebut (*static testing*)

4. Perencanaan (*planning*). Kita perlu merencanakan pengujian. Contohnya, merencanakan aplikasi apa yang akan diuji, apa saja tahapan yang diperlukan, kapan membuat laporan proses pengujian dan kondisi aplikasi selama diuji, kemudian kapan harus mengakhiri pengujian.
5. Persiapan (*preparation*). Kita harus memilih jenis pengujian yang dilakukan dan mempersiapkan test case.
6. Evaluasi (*evaluation*). Mengecek hasil akhir sehingga dapat memutuskan mana yang lulus uji dan mana yang perlu diperbaiki.
7. Produk aplikasi dan hubungannya dengan kinerja produk (*software product and related work product*). Tidak hanya melakukan pengujian koding aplikasi, namun juga diuji apakah kinerja aplikasi sesuai dengan kebutuhan pengguna.

### C. Pengujian pada V-Model

V-Model merupakan model pengembangan sistem yang sifatnya adalah perpanjangan dari *Waterfall*. Perbedaannya dari *Waterfall*, V-Model menambahkan STLC (*Software Testing Life Cycle*) bersamaan dengan adanya SDLC. [7]



**Gambar 1.** Alur Pengujian Model  
(Sumber : guru99.com)

### D. Pengujian metode Blackbox

Pengujian Black-box adalah salah satu metode dalam pengujian yang berfokus pada fungsional aplikasi. Sehingga, pengujian Black-box bisa mendapatkan rangkaian *input case* pada aplikasi yang bersifat fungsional pada sebuah program. Pengujian Blackbox bukan tidak dapat dibandingkan dengan pengujian Whitebox, melainkan mereka adalah metode pengujian yang saling melengkapi satu sama lain. [8]

### E. Boundary Value Analysis

*Boundary Value Analysis* (BVA) adalah teknik uji dari metode Black-box yang melengkapi pengujian *Equivalence Partitioning* (EP). Alih-alih memilih elemen apa pun dari kelas ekivalensi, BVA mengarah ke pemilihan *test case* di "edge" kelas. Daripada semata-mata pada kondisi input, BVA mendapatkan kasus uji dari *domain output* juga. [9]

### F. Equivalence Partitioning

Teknik ini membantu kita mengurangi jumlah tes kasus. Teknik ini pada dasarnya bekerja pada pembagian domain input program berdasarkan nilai input ke kelas kesetaraan. Kasing uji dihasilkan dari kelas kesetaraan ini yang berasal dari input domain. [10]

Kasus pengujiannya dibuat berdasarkan nilai minimum dan nilai maksimum yang sudah ditentukan sebelumnya. Misalkan, suatu *input* telah ditentukan hanya akan bisa diisi dengan nilai minimum 1 dan nilai maksimum 99, maka diluar dari nilai yang sudah ditentukan maka dapat dikatakan bahwa inputan tersebut tidak sesuai dengan kebutuhan.

**Tabel 1.** Batas Nilai Minimum dan Maksimum

tidak valid		Valid		Tidak valid
	0	1	99	100

### G. Decision Table

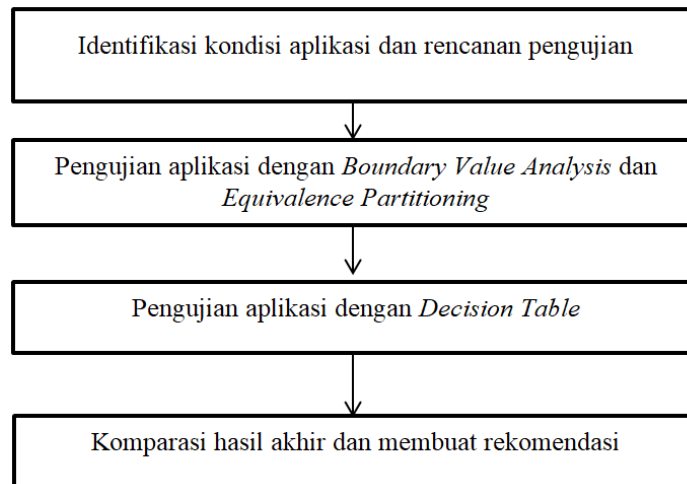
Decision Table adalah pengujian dengan menjadikan inputan dan proses menjadi kasus ujinya. Sebelumnya dalam membuat kasus uji, kita membuat beberapa kemungkinan-kemungkinan input yang dilakukan oleh pengguna lalu hasil akhir dapat ditemukan dari reaksi sistem berdasarkan inputan yang sudah ditentukan. Contohnya adalah kasus uji perwalian dan pemilihan kelas dengan 3 macam kemungkinan input akan menghasilkan reaksi yang berbeda. [11]

**Tabel 2.** Decision Table Dengan 3 Case

Kondisi	Case 1	Case 2	Case 3
Sudah melakukan perwalian	Valid	Valid	Tidak Valid
Sudah melakukan pemilihan kelas	Valid	Tidak Valid	Tidak Valid
Aksi			
Keluarkan jadwal kelas	Success	Error	Error

## III. METODE PENELITIAN

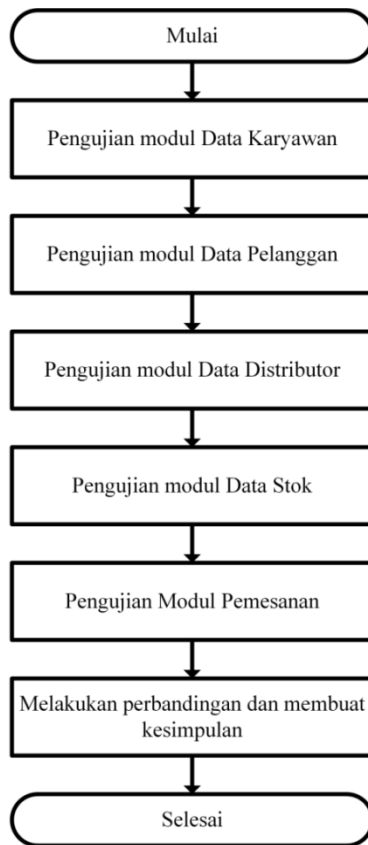
### A. Desain penelitian dan metode pengumpulan data



**Gambar 2.** Alur Penelitian

1. **Identifikasi kondisi aplikasi dan rencana pengujian.** Ini adalah tahapan pengumpulan data untuk kebutuhan pengujian pada aplikasi. Dalam pengumpulan data, terdapat beberapa metode :
  - a. **Wawancara.** Peneliti melakukan wawancara terhadap CV. Delta Pilar dalam rangka mengumpulkan kebutuhan instansi terhadap aplikasi.
  - b. **Observasi.** Pengamatan dilakukan pada aplikasi perhitungan stok barang yang akan diuji.
  - c. **Studi Literatur.** Melakukan pencarian studi literatur tentang pengujian metode Black-box model *Boundary Value Analysis - Equivalence Partitioning* dan *Decision Table*.

Adapun alur pengujian adalah sebagai berikut :



**Gambar 3.** Alur Pengujian

2. **Pengujian aplikasi dengan *Boundary Value Analysis* dan *Equivalent Partitioning*.** Melakukan pengujian seluruh form yang ada pada aplikasi dengan metode *Boundary Value Analysis* dan *Equivalent Partitioning*. Pengujian ini memiliki tiga tahapan yaitu **menentukan nilai ekivalen, membuat test case, dan pengujian.**
3. **Pengujian aplikasi dengan *Decision Table*.** Melakukan pengujian seluruh form yang ada pada aplikasi dengan metode *Decision Table*. Pengujian ini juga memiliki tiga tahapan yaitu **pembuatan test scenario, test case, dan pengujian.**
4. **Perbandingan hasil akhir dan membuat rekomendasi.** Tahapan ini melakukan perbandingan hasil antara metode *Boundary Value Analysis – Equivalent Partitioning* dengan *Decision Table*.

#### IV. HASIL DAN PEMBAHASAN

Berikut rekapitulasi pengujian dari seluruh modul

**Tabel 3.** Perbandingan Hasil Pengujian Seluruh Form

Nama Form	Hasil BVA+EP	Hasil DT
Tambah Level Karyawan	43%	60%
Edit Level Karyawan	43%	60%
Tambah Karyawan	38%	67%
Edit Karyawan	38%	67%
Tambah Pelanggan	55%	67%
Edit Pelanggan	55%	67%
Tambah Distributor	44%	60%

Edit Distributor	44%	60%
Tambah Stok	25%	0%
Edit Stok	25%	0%
Tambah Pemesanan	67%	80%
<b>Rata-rata Kesalahan</b>	<b>43,4%</b>	<b>53,5%</b>

Pengujian dengan DT menghitung kesalahan seluruh form dengan presentase 53,5% sedangkan presentase teknik BVA+EP adalah sebesar 43,4%. Berikut perinciannya :

1. Modul **Karyawan** memiliki kekurangan pada validasi jenis karakter yang masuk, hal itu menyebabkan modul ini bisa mencapai kesalahan hingga 60% dengan teknik DT karena menurut teknik ini, *output* hampir tidak berjalan sesuai skenario. Namun jika tanpa memperhatikan *output*, modul dapat memvalidasi inputan kosong dengan baik sehingga teknik BVA+EP menemukan kesalahan pada modul ini dengan presentase 43%.
2. Modul **Pelanggan** dan **Distributor** juga memiliki kekurangan validasi yang sama dengan modul nomor 1 sehingga mereka memiliki presentase yang hampir sama pada teknik DT yaitu sebesar 60%-67% sedangkan pada teknik BVA+EP memiliki presentase yang lebih rendah yakni sekitar 55%-39%
3. Modul **Stok** tidak ditemukan kesalahan jika diuji dengan teknik DT, terlihat dari presentase yang diperoleh adalah sebesar 0%. Namun teknik BVA+EP justru menemukan sebaliknya, bahwa modul ini memiliki kesalahan dengan presentase 25%. Ini menunjukkan bahwa meskipun jika secara detail modul ini masih memiliki kekurangan dalam validasi jenis karakter pada *field* Jumlah, namun kekurangan validasi tersebut tidak mempengaruhi *output*.
4. Modul **Pemesanan** memiliki presentase kesalahan paling besar dalam teknik DT yaitu sebesar 80%. Juga dengan teknik uji BVA+EP, kesalahan yang ditemukan masih di angka yang sama tinggi yaitu sebesar 67%. Hal ini menunjukkan bahwa modul memiliki tingkat kekurangan validasi yang tinggi, dan pengaruhnya juga begitu fatal pada *output*.

Sehingga perlu adanya perbaikan dengan meningkatkan kemampuan memvalidasi inputan. Adapun bagian-bagian yang perlu diperbaiki maka peneliti sajikan dalam bentuk tabel. Warna yang gelap menunjukkan bagian yang perlu diperbaiki :

**Tabel 4. Bagian-bagian yang Perlu Diperbaiki**

No	Nama Modul	Form Field	Validasi Inputan Kosong	Validasi Jumlah Karakter	Validasi Jenis Karakter
1	Karyawan (Level Karyawan)	Level Pengguna	√	X	X
2		Deskripsi	√	X	X
3	Karyawan (Biodata Karyawan)	Nama Lengkap	√	X	X
4		Username	√	X	X
5		Password	√	X	X
6	Pelanggan	Nama Lengkap	√	X	X
7		No. HP	√	X	X
8		Alamat Lengkap	√	X	X
9	Distributor	Nama Lengkap	√	X	X
10		No. HP	√	X	X
11	Stok	Jumlah	√	X	X
12	Pemesanan	Jumlah	X	X	X

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

Dari hasil pengujian baik dengan teknik *Boundary Value Analysis – Equivalent Partitioning* dengan *Decision Table*, secara umum form mampu untuk memvalidasi inputan kosong, namun belum mampu memvalidasi jenis-jenis inputan yang tidak diinginkan. Form belum mampu memvalidasi jumlah karakter dan jenis karakter yang masuk baik

itu jenis alphabet (a...z), angka (0...9), atau simbol. Maka rekomendasi dari penelitian ini bagi instansi adalah **untuk menambah kemampuan form dalam memvalidasi jumlah dan jenis karakter..**

Teknik pengujian antara *Boundary Value Analysis – Equivalent Partitioning* dengan *Decision Table* memiliki titik fokus yang berbeda. BVA dan EP menitikberatkan pada kedetailan jenis dan jumlah karakter yang masuk tanpa memperhatikan *output* yang dihasilkan. Sedangkan DT fokus pada pengaruh *input* pada *output* sehingga itulah sebabnya teknik DT juga memperhatikan seperti apa *output* yang dihasilkan. BVA dan EP menguji validitas form, sedangkan DT menguji kualitas form.

Perbedaan titik fokus ini akan berguna untuk pengujian yang memiliki kebutuhan tertentu. Jika kebutuhan pengujian adalah untuk menguji validitas form secara lengkap, maka teknik BVA dan EP adalah pilihan yang tepat. Jika kebutuhan pengujian adalah untuk menguji kualitas form dan pengaruhnya pada *output*, maka teknik DT adalah pilihan yang tepat.

## B. Saran

Pengujian dapat dilakukan kembali setelah instansi melakukan rekomendasi yang sudah disediakan. Penelitian ini masih dapat dikembangkan jika ada penambahan atau perubahan nilai ekivalen (tergantung pada kebutuhan sistem) atau pada saat merubah *test case* dan menggunakan jenis-jenis *test case* lain yang lebih variatif. Seiring dengan dikembangkannya secara terus-menerus aplikasi ini, maka pengujian akan selalu dibutuhkan.

## DAFTAR PUSTAKA

- [1] Hakim, Ardiana Nur and Syuhada, Disa Arista, “Aplikasi Perhitungan Stok Barang Berbasis Website Pada CV. Delta Pilar (Cabang Bandung)”, unpublished.
- [2] Jaya, Tri Snadhika, “Pengujian Aplikasi dengan Metode Blackbox Testing Boundary Value Analysis (Studi Kasus : Kantor Digital Politeknik Negeri Lampung)”, *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*. Vol.03, pp. 45-48, Jan. 2018.
- [3] Mustaqbal, M. Sidi, Firdaus, Roeri Fajri and Rahmadi, Hendra, “PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)”, *Jurnal Ilmiah Teknologi Informasi Terapan (JITTER)*. Vol. I, pp. 31-36, Agu. 2015.
- [4] Mursyidah and Hidayat, Hari Toha, “Pengujian Sistem Informasi Akuntansi Biaya Operasional Sekolah Dengan Black Box Testing”, *Jurnal Infomedia*. Vol. 2, pp 7-14, Des. 2017.
- [5] Salamah, Umi and Khasanah, Fata Nidaul, “Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box Testing”, *Information Management For Educators And Professionals*. Vol. 2, pp. 35-46, Des. 2017.
- [6] A.Nursikuwagus, L.Melian and P.Andrianto, “E-Health as a Service Software of Medical”, *International Journal of New Media Technology (IJMNT)*. Vol. 4, pp. 99-104, Des. 2017.
- [7] Graham, Dorothy, et al., *Foundation of Software Testing ISTQB Certification*. London : Thomson Learning, 2007.
- [8] Guru99. *What is V Model? Learn with a Case Study using SDLC & STLC* [Online]. Available: <https://www.guru99.com/software-testing-lifecycle.html>.
- [9] Pressman, Roger S., *Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku Satu)*. Yogyakarta : Penerbit ANDI, 2010.
- [10] —., *Software Engineering : A Practitioner's Approach*. Singapore : The McGraw-Hill Companies, 2010.
- [11] Jan, Syed Roohullah, et al, "An Innovative Approach to Investigate Various Software", *IJSRSET*. Vol. 2, pp. 682-689, Apr. 2016.