

## Pengujian Aplikasi Berbasis Website Dengan *Black Box Testing Metode Boundary Value Analysis Dan Responsive Testing*

Doddy Satrya Perbawa <sup>1)</sup>, Galih Setiawan Nurohim <sup>2)</sup>  
Universitas Bina Sarana Informatika <sup>1)</sup>  
Doddy.dwp@bsi.ac.id <sup>1)</sup>, Galih.glt@bsi.ac.id <sup>2)</sup>

**Abstract** - Website is one of the information systems that is very much needed by every institution, both government and private. Information systems are very important because they can be used to recap academic data and provide online services to all people who are currently familiar with the internet and digital world. Like a system, testing is necessary in order to know the weaknesses and weaknesses of a system. Various system testing methods have been widely used to determine the advantages and disadvantages of website-based information systems. One of them 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 you want to test. This test is carried out on the function of adding patient data on the doctor's practice medical record website. The test results show that there are still many shortcomings when validating the data to be entered, so that it can cause the data stored in the database to not match the expected data. The test results can be used as input to improve the application because in the eligibility requirements of the information system it is necessary to test in order to know the weaknesses and strengths of the information system.

**Keywords:** Website, Boundary value analysis, Black box analysis

**Abstrak** – Website merupakan salah satu system informasi yang sudah sangat dibutuhkan setiap instansi baik pemerintahan maupun swasta, hal ini sudah terbukti mulai dari bidang instansi pendidikan negeri maupun swasta sudah tidak asing dengan system informasi berbasis website. Sistem informasi sangat penting karena dapat digunakan untuk rekap data akademik dan memberikan layanan secara online kepada setiap masyarakat yang saat ini sudah tidak asing dengan dunia internet dan digital. Layaknya sebuah system perlu dilakukan pengujian agar dapat diketahui sisi kekurangan dan kelemahan suatu system. Berbagai metode pengujian system sudah banyak digunakan untuk mengetahui kekurangan dan kelebihan system informasi berbasis website. Salah satunya adalah pengujian system metode black box dengan cara Boundary Value Analys. Boundary Value Analysis merupakan cara pengujian dengan menentukan nilai batas bawah dan batas atas dari data yang akan ingin diuji. Pengujian ini dilakukan pada fungsi tambah data pasien pada website rekam medis praktek dokter . Hasil pengujian menunjukkan bahwa masih terdapat banyak kekurangan saat melakukan validasi data yang akan dimasukkan, sehingga dapat menyebabkan data yang disimpan pada database tidak sesuai dengan data yang diharapkan. Hasil pengujian dapat dijadikan masukan untuk memperbaiki aplikasi karena dalam persyaratan kelayakan system informasi perlu dilakukan pengujian agar dapat diketahui kekurangan dan kelebihan system informasi.

**Kata kunci:** Website, Boundary value analysis, Black box analysis

### 1. Latar Belakang

Aplikasi website praktek dokter pengolahan data pasien ini merupakan aplikasi yang dibuat untuk membantu para petugas medis pada tempat praktek dokter untuk konsultasi pasien dan pendaftaran pasien. Dengan aplikasi ini, diharapkan dapat membantu pasien dalam konsultasi dan mendaftar dalam pemeriksaan kesehatan.

Pada aplikasi ini terdapat beberapa modul, antara lain modul data pasien, kartu berobat pasien

riwayat kesehatan, kunjungan pasien, diagnosa penyakit pasien, pembayaran, laporan dan lain-lain.

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. Suatu kasus test yang baik adalah apabila test tersebut mempunyai kemungkinan menemukan sebuah kesalahan yang tidak

terungkap. Suatu test yang sukses adalah bila 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 minimum.

## 2. Kajian Pustaka

### a) Pengujian Software

Pengujian pada software sangat diperlukan bertujuan untuk memastikan bahwa software/aplikasi yang sudah/sedang dibuat dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Pengembang atau penguji software harus menyiapkan sesi khusus untuk menguji program yang sudah dibuat agar kesalahan ataupun kekurangan dapat dideteksi sejak awal dan dikoreksi secepatnya. Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean. (Shi, 2010)

Pengujian software haruslah dilakukan dalam proses rekayasa perangkat lunak atau software engineering. Sejumlah strategi pengujian software telah diusulkan dalam literatur. Semuanya menyediakan template untuk pengujian bagi pembuat software. Dalam hal ini, semuanya harus memiliki karakteristik umum berupa (Bhat and Quadri, 2015) :

1. Testing dimulai pada level modul dan bekerja keluar ke arah integrasi pada sistem berbasis komputer
2. Teknik testing yang berbeda sesuai dengan poin-poin yang berbeda pada waktunya
3. Testing diadakan oleh pembuat/pengembang software dan untuk proyek yang besar oleh group testing yang independent
4. Testing dan Debugging adalah aktivitas yang berbeda tetapi debugging harus diakomodasikan pada setiap strategi testing

Pengujian software adalah satu elemen dari sebuah topik yang lebih luas yang sering diartikan sebagai Verifikasi dan Validasi (V&V)

- Verifikasi: menunjuk kepada kumpulan aktifitas yang memastikan bahwa software telah mengimplementasi sebuah fungsi spesifik.
- Validasi: menunjuk kepada sebuah kumpulan berbeda dari aktivitas yang memastikan bahwa

software yang telah dibangun dapat ditelusuri terhadap kebutuhan customer.

Definisi V&V meliputi banyak aktifitas SQA (software quality assurance), termasuk review teknis formal, kualitas dan audit konfigurasi, monitor performance. Terdapat beberapa tipe yang berbeda dalam pengujian software yang meliputi studi kelayakan dan simulasi. (Bhat and Quadri, 2015):

1. Metode software engineering menyediakan dasar dari mutu yang mana yang akan dipakai.
2. Metode Analysis, design and Construction berupa tindakan untuk meningkatkan kualitas dengan menyediakan teknik yang seragam dan hasil yang sesuai dengan keinginan.
3. Metode Formal Technical Reviews menolong untuk memastikan kualitas kerja produk merupakan hasil konsekuensi dari setiap langkah software engineering.
4. Metode Measurement diberlakukan pada setiap elemen dari konfigurasi software
5. Metode Standards and Procedures membantu untuk memastikan keseragaman dan formalitas dari SQA untuk menguatkan dasar “filosofi kualitas total”
6. Metode Testing menyediakan cara terakhir dari tingkat kualitas mana yang dapat dicapai dan dengan praktis dapat mengetahui letak error.

Daivids menyarankan satu set prinsip pengujian:

1. Semua test harus dapat dilacak ke kebutuhan pelanggan.
2. Test harus direncanakan dengan baik sebelum pengujian mulai.
  - a. Prinsip Pareto berlaku untuk pengujian
  - b. 80% dari semua kesalahan yang terungkap selama pengujian akan mudah dapat dilacak dari 20% semua modul program.
3. Pengujian seharusnya mulai “dari yang kecil” dan pengujian perkembangan ke arah “yang besar”.
4. Pengujian menyeluruh adalah tidak mungkin. Paling efektif, pengujian harus diselenggarakan oleh suatu pihak ketiga mandiri.

Langkah-langkah pengujian software ada 4 yaitu:

1. Unit testing-testing per unit yaitu mencoba alur yang spesifik pada struktur modul kontrol untuk memastikan pelengkapan secara penuh dan pendeteksian error secara maksimum

2.Integration testing – testing per penggabungan unit yaitu pengalamatan dari isu-isu yang diasosiasikan dengan masalah ganda pada verifikasi dan konstruksi program

3.High-order test yaitu terjadi ketika software telah selesai diintegrasikan atau dibangun menjadi satu –tidak terpisah-pisah

4.Validation test yaitu menyediakan jaminan akhir bahwa software memenuhi semua kebutuhan fungsional, kepribadian dan performa.

Tom Gilb menyatakan bahwa prosedur yang harus digunakan jika ingin mengimplementasikan strategi testing software yang sukses (Bhat and Quadri, 2015):

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

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.

#### b) Pengujian Black Box

Black Box Testing merupakan pengujian sistem yang berfokus pada spesifikasi fungsional dari perangkat lunak tersebut. Tester dapat mendefinisikan kumpulan kondisi input dan

melakukan pengetesan pada spesifikasi fungsional program.

Black Box Testing merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh pengujian White Box Testing.

Black Box Testing cenderung untuk menemukan hal-hal berikut:

1.Fungsi yang tidak benar atau tidak ada

2.Kesalahan pada struktur data dan akses basis data.

3.Kesalahan inisialisasi dan terminasi.

4.Kesalahan antarmuka (interface errors).

5.Kesalahan performansi (performance errors).

Pengujian didesain untuk menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana fungsi-fungsi diuji agar dapat dinyatakan valid?

2. Input seperti apa yang dapat menjadi bahan kasus uji yang baik?

3. Apakah sistem sensitif pada input-input tertentu?

4. Bagaimana sekumpulan data dapat diisolasi?

5. Berapa banyak rata-rata data dan jumlah data yang dapat ditangani sistem?

6. Efek apa yang dapat membuat kombinasi data ditangani spesifik pada operasi sistem?

Saat ini terdapat banyak metoda atau teknik untuk melaksanakan Black Box Testing, antara lain:

1.Equivalence Partitioning

2.Boundary Value Analysis/Limit Testing

3.Comparison Testing

4.Sample Testing

5.Robustness Testing

6.Behavior Testing

7.Requirement Testing

8.Performance Testing

9.Uji Ketahanan (Endurance Testing)

10.Uji Sebab-Akibat (Cause-Effect Relationship Testing)

(M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi, 2015)

#### c) Pengujian Black Box Boundary Value Analysis

Boundary value analysis 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 :

1. Banyak kesalahan terjadi pada kesalahan masukan.
2. BVA mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input.
3. 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.

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 diisi 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

Nama Pasien (Char)	Jenkel (Char)	Alamat (Char)	Kota (Char)	Tel (Int 13)	Ket (Char 30)

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

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

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

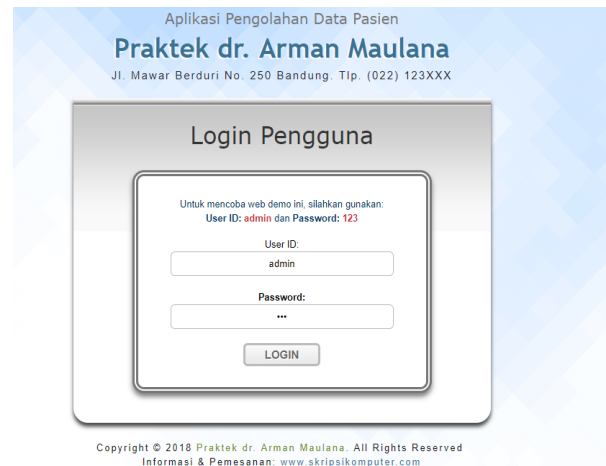
$$1812 \leq \text{tahun} \leq 2016$$

Maka untuk setiap entri data di luar angka di atas akan menampilkan pesan “Tanggal yang anda

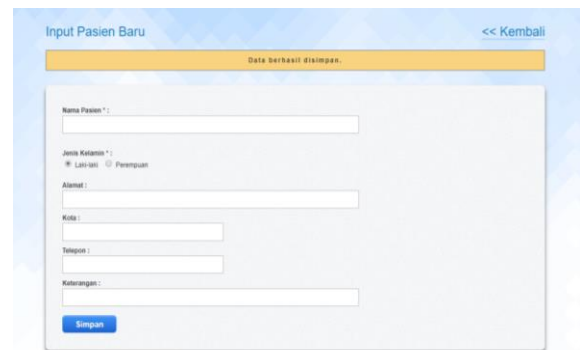
isikan salah”. (M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi, 2015)

### 3. Pengujian Black Box Boundary Value Analys

Berdasarkan pada penjelsan sebelumnya, telah dicoba penerapan teknik BVA untuk pengujian aplikasi yaitu aplikasi website pengolahann data pasien praktek dokter. Aplikasi ini terdapat beberapa modul, namun pada artikel ini akan dibahas sebagai contoh pengujian pada modul “Input pasien baru”



Gambar 1. Form Login Admin



Gambar 2. Form Input pasien baru

Berdasarkan form di atas, kemudian dilakukan pengujian dengan menyiapkan beberapa data uji. Dari bentuk form di atas, contoh pengujian akan dilakukan pada satu field yaitu Nama Pasien.

#### a) Pengujian field Nama Pasien

Aturan entri data A.1 Terdiri karakter huruf

Sample data	Expected Result	Result	Condition
Arum	T	T	Success
Art7m	F	T	Success
Ar_um	F	T	Success

Aturan entri data A.1 Terdiri angka

Sample data	Expected Result	Result	Condition
099999	F	T	Success
9_889	F	T	Success
4_999	F	T	Success

Berdasarkan hasil uji pada satu form di atas, dapat disiapkan beberapa kasus data uji. Pada contoh di atas, terdapat satu field yang akan diuji, dengan masing-masing field memuat minimal 2 aturan uji. Untuk satu aturan uji, perlu disiapkan minimal 3 data uji, sehingga total data uji yang perlu disiapkan untuk kasus di atas adalah 1 field x 2 aturan x 3 data uji yaitu 6 data uji. Hal ini dapat memberikan gambaran jumlah data uji yang harus disiapkan untuk melakukan blackbox testing dengan metode BVA. Hasil pengujian memperlihatkan bahwa aplikasi ini masih memiliki beberapa kekurangan yaitu belum lengkapnya proses validasi data sehingga masih perlu penyempurnaan dengan menambah fungsi validasi.

#### 4. Kesimpulan

- Setelah melakukan pengujian pada fungsionalitas tambah kelas dengan menggunakan metode pengujian Blackbox Testing Boundary Value Analysis dapat ditarik kesimpulan bahwa:
  - Metode Blackbox Testing merupakan salah satu metode yang mudah digunakan karena
  - hanya memerlukan batas bawah dan batas atas dari data yang di harapkan,
  - Berdasarkan hasil uji dengan metoda BVA maka fungsi entri data perlu dilengkapi dengan beberapa proses validasi data untuk menjamin akurasi entri data sesuai fungsional yang diinginkan. Seperti halnya saat melakukan input nama salah melakukan input yaitu input angka.
  - Estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan

diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi.

- Setelah melakukan pengujian diketahui bahwa fungsionalitas masih bisa berjalan namun masih dapat menerima masukan data yang tidak diharapkan sehingga dapat menyebabkan data yang disimpan kurang valid.

#### Daftar Pustaka

- Bhat, A, and Quadri, S.M.K, 2015, Equivalence Class Partitioning and Boundary Value Analysis = A review, 2nd International Conference on Computing for Sustainable Global Development (INDIACom)
- Khan, Mohd Ehmer, 2011, Different Approach to Blackbox Testing Technique for Finding Error, International Journal of Software Engineering & Applications (IJSEA), Vol.2, No.4, October 2011
- Nidhra, Srinivas, and Dondeti, Jagruthi, 2012, Blackbox and Whitebox Testing Techniques - A Literature Review, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012
- M. Sidi Mustaqbal, Roeri Fajri Firdaus, Hendra Rahmadi Jurnal Ilmiah Teknologi Informasi Terapan Volume I, No 3, 10 Agustus 2015
- Shi, Mingtao, 2010, Software Functional Testing from the Perspective of Business Practice Computer and Information Science, www.ccsenet.org/cis