

## PENGGUNAAN METODE BLACK BOX PADA PENGUJIAN SISTEM INFORMASI SURAT KELUAR MASUK

Umi Hanifah<sup>1</sup>, Ronggo Alit<sup>2</sup>, Sugiarto<sup>3</sup>

<sup>123</sup>Program Studi Teknik Informatika Fakultas Teknologi Industri

Universitas Pembangunan Nasional “Veteran” Jawa Timur

Jl. Raya Rungkut Madya, Gunung Anyar, Surabaya, Jawa Timur 60294

hanyahanifah92@gmail.com<sup>1</sup>, ronggoa@gmail.com<sup>2</sup>, soegi@null.net<sup>3</sup>

**Abstrak.** Pengukuran kualitas pada perangkat lunak saat ini sangat diperlukan sebelum sebuah perangkat lunak akan digunakan oleh pengguna. Hal ini bertujuan untuk mengetahui kelemahan dari sistem agar data yang dihasilkan bisa sesuai dengan data yang di masukkan setelah data dieksekusi. Teknik dari penelitian ini yaitu sistem informasi surat keluar masuk kemudian dilakukan menggunakan pengujian Black Box. Metode ini berfokus pada masukkan data, tampilan sistem, pemakaian memori dan kecepatan eksekusi data sehingga jika masukkan data tidak sesuai dengan apa yang diharapkan maka sistem gagal. Hasil dari pengujian menggunakan metode ini dikatakan sangat baik karena dapat diketahui semua kelemahan pada sistem sebelum digunakan.

**Kata Kunci:** Sistem Informasi, Surat Keluar Masuk, Black Box

Perkembangan teknologi informasi juga menjamah proses dalam ruang lingkup universitas agar dapat meningkatkan efektifitas dan efisiensi kerja. Salah satu contoh adalah kinerja akademik bagian arsip yang kurang efisien karena masih menerapkan sistem pengarsipan yang konvensional. Arsip yang tersimpan ada bermacam-macam diantaranya surat masuk umum, arsip kemahasiswaan, arsip akademik dan surat keluar seperti penugasan, surat permohonan dan lain-lain. Perkembangan teknologi informasi juga menjamah proses dalam ruang lingkup universitas agar dapat meningkatkan efektifitas dan efisiensi kerja. Salah satu contoh adalah kinerja akademik bagian arsip yang kurang efisien karena masih menerapkan sistem pengarsipan yang konvensional.

Arsip yang tersimpan ada bermacam-macam diantaranya surat masuk umum, arsip kemahasiswaan, arsip akademik dan surat keluar seperti penugasan, surat permohonan dan lain-lain. Catatan surat masuk dan surat keluar dilakukan dengan mencatat aktifitas keluar masuk surat secara manual dan dengan cara mencatat aktifitas pada buku catatan arsip.

Pengarsipan surat masuk dan keluar dilakukan secara manual dengan mengelompokkan dokumen-dokumen yang sejenis (Junidar 2012).

### Pengujian Perangkat Lunak

Pengujian perangkat lunak yang dilaksanakan dengan tidak sempurna tentu akan membawa pengaruh yang kurang baik terhadap kualitas perangkat lunak yang dihasilkan. Pengujian perangkat lunak yang tidak efektif dan tidak lengkap dapat mengakibatkan berbagai masalah ketika perangkat lunak tersebut digunakan oleh *end-user*. Secara otomatis dapat meningkatkan efisiensi proses pengujian untuk mengidentifikasi bagian dari perangkat lunak yang rawan mengalami kegagalan (Catelani 2011). Dalam pengujian perangkat lunak terdapat berbagai metode yang bisa digunakan untuk melakukan pengujian, misalnya metode *Black Box Testing*. Umumnya, sebuah metode pengujian perangkat lunak ini hanya mencakup sebuah area dalam perangkat lunak (*Data Driven Testing*) hanya menguji kemampuan perangkat lunak menerima dan mengolah data masukan. Sistem pengujian perangkat lunak tentunya harus mampu menguji berbagai aspek dalam perangkat

lunak sehingga penggunaan lebih dari satu tipe sub pengujian sangat diharapkan (Ghalin 2014).

Berdasarkan metode yang digunakan maka akan diketahui kelemahan pada sistem informasi setelah dilakukan pengujian menggunakan metode *Black Box* dan bagaimana untuk mengetahui *output* yang dianggap valid (menurut kebutuhan kermawa), Sedangkan untuk tujuan dari penelitian ini adalah mengetahui kelemahan dari sistem agar data yang dihasilkan sesuai dengan data yang di masukkan setelah data dieksekusi dan menghindari kekurangan dan kesalahan pada aplikasi sebelum digunakan oleh user.

## I. Metodologi

### Metode Black Box

Pengujian sistem bertujuan untuk melihat apakah sistem yang telah dibuat sudah sesuai dengan tujuan awal pembuatan dan layak untuk dipergunakan. Pengujian pada sistem menggunakan metode *Black Box*, tujuannya untuk mengetahui bahwa bagian-bagian dalam sistem aplikasi telah benar menampilkan pesan-pesan kesalahan jika terjadi kesalahan dalam penginputan data (Sandy 2015). *Black Box Testing* sendiri merupakan pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak. Pengujian black box ini menitik beratkan pada fungsi system (Rizki 2015).

Berikut adalah 10 tipe pengujian dari metode *Black Box* menurut (Julian 2015) :

1. *Equivalence Partitioning* :  
Membagi inputan menjadi kelas data yang dapat digunakan untuk menggenerasi kasus uji.
2. *Boundary Value Analysis / Limit Testing*:  
Mengijinkan untuk menyeleksi kasus uji yang menguji batasan nilai input, Merupakan komplemen dari *Equivalence Partitioning*.

3. *Comparison Testing* :  
Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.
4. *Sample Testing* :  
Melibatkan beberapa nilai yang terpilih dari sebuah kelas ekivalen.
5. *Robustness Testing* :  
Data input dipilih diluar spesifikasi yang telah didefinisikan, Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid
6. *Behavior Testing* :  
Hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur data *stack*.
7. *Performance Testing* :  
Mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya : aliran data, ukuran pemakaian memori, kecepatan eksekusi.
8. *Requirement Testing* :  
Spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak diidentifikasi pada tahap spesifikasi kebutuhan dan desain.
9. *Endurance Testing* :  
Melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu.
10. *Cause – Effect Relationship Testing* :  
Bagi-bagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja

### Skenario Uji Coba

Pengujian menggunakan masukan data acak bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan pada database, sehingga sistem dikatakan layak untuk digunakan. Berikut cara pengujian setiap tipe dari metode *Black Box* :

1. *Equivalence Partitioning*  
Pengujian ini dilakukan pada form yang sudah ada pada sistem informasi surat keluar masuk dengan memasukkan data

yang tidak sesuai dengan tipe data atau memasukkan data acak.

2. *Boundary Value Analysis*  
Pengujian ini untuk memastikan bahwa masukkan data yang melebihi batas yang sudah ditentukan tidak dapat tersimpan dengan baik pada database.
3. *Comparison Testing*  
Membandingkan tampilan *interfaces* sistem pada web browser yang berbeda
4. *Sample Testing*  
Pengujian ini untuk memastikan nilai yang terpilih dapat menghasilkan data yang baik dan sesuai dengan data masukkan dari *user*.
5. *Robustness Testing*  
Penguji akan memasukkan data acak untuk membuktikan bahwa tidak ada kesalahan jika masukan tidak valid.
6. *Behavior Testing*  
Pengujian ini dilakukan dengan cara membuat data baru secara berkali – kali untuk menghindari data *stack*.
7. *Performance Testing*  
Pengujian ini mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari aliran pemakaian memori,
8. *Requirement Testing*  
Tipe ini hanya melihat spesifikasi kebutuhan dari sistem mulai dari sistem pembuatan sampai pengujian.
9. *Endurance Testing*  
Tipe ini untuk memastikan apakah hasil operasi matematika pada sistem ini benar atau salah.
10. *Cause – Effect Relationship Testing*  
Pengujian yang melibatkan kondisi input dan aliran data mulai dari *Input, View, Update, Delete* dan *Search*.

Sehingga dari cara pengujian diatas sudah dijelaskan untuk melakukan pengujian dari beberapa tipe pengujian yang nantinya akan di implementasikan pada sistem informasi tersebut, dapat diketahui hasil pada sebuah tabel. Diharapkan tidak ada kesalahan masukkan selama pengujian berlangsung untuk memastikan sistem menghasilkan data yang baik.

## II. Hasil dan Pembahasan

### Implementasi dan Uji Coba

Implementasi dan uji coba dilakukan pada sistem informasi surat keluar masuk dengan tujuan untuk mengetahui kekurangan yang ada pada sistem sebelum sistem digunakan oleh pengguna. Hasil dari pengujian jika ditemukan kelemahan pada bagian sistem maka pengguna bisa mengembangkan pada bagian sistem tersebut yang dianggap lemah. Uji coba yang dilakukan berdasarkan 10 tipe pengujian dari metode *Black Box* yaitu :

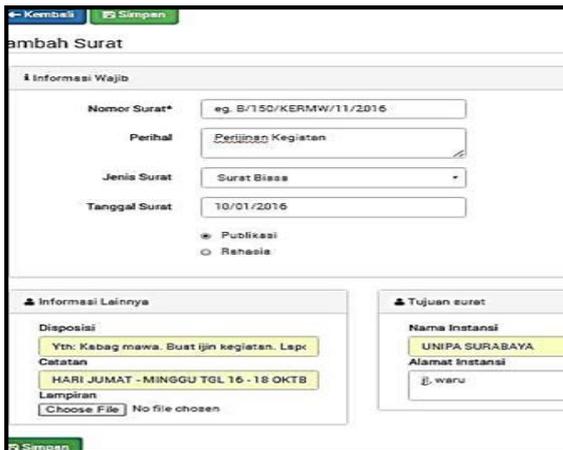
1. Memasukkan data benar sebagai contoh sistem menerima data masukan untuk disimpan pada database.
2. Memasukkan data berupa data acak untuk memastikan sistem menolak untuk menyimpan data masukan pada database.

Pada hasil pengujian terdapat tabel *test case* yang berfungsi untuk menyimpulkan apakah sistem berhasil dalam pengujian tipe tersebut atau tidak. Berikut terdapat beberapa penjelasan dari tabel pengujian yang akan digunakan :

1. *Input* adalah penjelasan tentang masukkan data benar atau data acak.
2. Hasil harapan adalah hasil dari harus terjadi ketika dalam proses pengujian.
3. *Output* adalah hasil dari pengujian setelah sistem selesai diujikan
4. Kesimpulan adalah simpulan apakah sistem tersebut dinyatakan berhasil atau gagal.

### Pengujian Data Benar ( *Data Valid* )

Pengujian menggunakan data benar oleh mahasiswa *non IT* dengan harapan sistem juga mampu menerima data tanpa adanya kesalahan baik.



Gambar 1. Penambahan Data Surat Masuk Baru

| # | Tanggal    | No Surat               | Perihal        | Asal     | Tujuan   |
|---|------------|------------------------|----------------|----------|----------|
| 1 | 19-12-2015 | eg. B403K/KERMW/1/2015 | Biasa          | REKTORAT | Kemawa   |
| 2 | 19-12-2015 | eg. B402Z/KERMW/1/2015 | Biasa          | III      | FTI      |
| 3 | 19-12-2015 | eg. B402X/KERMW/1/2015 | Biasa          | REKTORAT | Kemawa   |
| 4 | 19-12-2015 | eg. B402Y/KERMW/1/2015 | Sangat Penting | REKTORAT | Kemawa   |
| 5 | 19-12-2015 | eg. B403K/KERMW/1/2015 | Biasa          | FTI      | Kemawa   |
| 6 | 19-12-2015 | eg. B200K/KERMW/1/2015 | Biasa          | FTI      | Kemawa   |
| 7 | 18-12-2015 | eg. B401Y/KERMW/1/2015 | Biasa          | Kemawa   | Rektorat |
| 8 | 18-12-2015 | eg. B402X/KERMW/1/2015 | Biasa          | FTI      | Kemawa   |
| 9 | 18-12-2015 | eg. B405K/KERMW/1/2015 | Biasa          | REKTORAT | Kemawa   |

Gambar 2. Export Keseluruhan Data Surat Masuk

Pada gambar 1 adalah proses pengujian data benar dengan memasukkan data yang diperlukan untuk pembuatan surat masuk. Pada gambar 2 adalah proses pengujian *export* data surat masuk.

Tabel 1 Hasil Pengujian Data Benar

| Input                            | Hasil Harapan                              | Output  | Kesimpulan                |
|----------------------------------|--|---|---------------------------|
| Isikan data pada informasi wajib | Sistem menerima semua data                 | Sistem menerima seluruh data masukkan dengan <i>type</i> data yang sesuai | [√] Berhasil<br>[ ] Gagal |
| Export seluruh data surat masuk  | Data surat masuk berhasil di <i>export</i> | Sistem berhasil menampilkan hasil <i>export</i> surat masuk               | [√] Berhasil<br>[ ] Gagal |

### Pengujian Data Acak ( Data Tidak Valid )

Pengujian menggunakan masukan data acak bertujuan untuk memastikan sistem menolak untuk menyimpan data masukan pada database, sehingga sistem dikatakan layak untuk digunakan. Pengujian akan dilakukan dengan 10 tipe metode *Black Box* :

#### 1. Equivalence Partitioning

Pengujian ini dilakukan pada form yang sudah ada pada sistem informasi surat keluar masuk dengan memasukkan data yang tidak sesuai dengan *type* data atau memasukkan data acak.

Tabel 2 Hasil Pengujian *Equivalence Partitioning*

| Input   | Hasil Harapan   | Output  | Kesimpulan                |
|---|---|---|---------------------------|
| Masukkan ID : 1234567891 Password : 123917 acak | Sistem menolak untuk menyimpan data masukkan  | Sistem kembali ke menu awal dan tidak dapat masuk untuk mengelola data surat              | [√] Berhasil<br>[ ] Gagal |
| Masukkan input dengan <i>text html</i>          | Sistem menolak untuk menyimpan karena data berupa <i>text html</i> dan sistem tidak mengalami kesalahan <i>interfaces</i> | Sistem mampu menyimpan data kedalam database dan mengalami perubahan tampilan pada sistem | [ ] Berhasil<br>[√] Gagal |

#### 2. Boundary Value Analysis

Pengujian ini untuk memastikan bahwa masukan data yang melebihi batas yang sudah ditentukan tidak dapat tersimpan dengan baik pada database, dan sistem hanya memunculkan data yang kurang dari batas data.

Tabel 3 Hasil Pengujian *Boundary Value Analysis*

| Input  | Hasil Harapan  | Output   | Kesimpulan                |
|--|--|--|---------------------------|
| Isikan pada form informasi wajib melebihi batasan data | Sistem menyimpan data dan menampilkan data sesuai dengan batas | Sistem menampilkan data sesuai dengan batas data | [√] Berhasil<br>[ ] Gagal |

### 3. Comparison Testing

Tipe ini nantinya tidak mencari kesalahan pada masukan, melainkan membedakan tampilan *interfaces* sistem pada web browser yang berbeda, tipe ini hanya untuk redundansi perangkat lunak dan untuk menguji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama.

Tabel 4 Hasil Pengujian *Comparison Testing*

| Input                           | Hasil Pengamatan  | Chrome                    | Internet Explorer         |
|---------------------------------|---|---------------------------|---------------------------|
| Tampilan <i>export as Excel</i> | Hasil <i>export</i> data surat masuk tidak teratur      | [ ] Berhasil<br>[√] Gagal | [ ] Berhasil<br>[√] Gagal |
| Tampilan <i>export as PDF</i>   | Terjadi penumpukan data surat jika data terlalu panjang | [ ] Berhasil<br>[√] Gagal | [ ] Berhasil<br>[√] Gagal |

### 4. Sample Testing

Pengujian ini untuk memastikan nilai yang terpilih dapat menghasilkan data yang baik dan sesuai dengan data masukkan dari *user*.

Tabel 5 Hasil Pengujian *Sample Testing*

| Input   | Hasil Harapan                  | Output  | Kesimpulan                |
|---|--------------------------------|---|---------------------------|
| Masukkan data pencarian untuk memastikan nilai terpilih, Contoh : surat | Sistem berhasil menemukan data | Sistem berhasil menampilkan data yang berhubungan dengan nama surat | [√] Berhasil<br>[ ] Gagal |

### 5. Robustness Testing

Pengujian data acak dimana penguji akan memasukkan data acak untuk membuktikan bahwa tidak ada kesalahan jika masukan tidak *valid*. Ketika sistem menampilkan hasil keluaran tidak *valid* maka sistem tidak dikatakan mengalami kegagalan dalam hal pengujian ini.

Tabel 6 Hasil Pengujian *Robustness Testing*

| Input  | Hasil Harapan                           | Output  | Kesimpulan                |
|--|---|---|---------------------------|
| Edit data pegawai dengan data simbol * menjadi # | Sistem menerima penambahan data pegawai | Sistem berhasil menerima penambahan data dengan data acak | [√] Berhasil<br>[ ] Gagal |

### 6. Behavior Testing

Pengujian ini dilakukan dengan cara membuat data baru secara berkali – kali untuk menghindari data *stack* dan sistem dapat menerima data dengan jumlah lebih dari 50.

Tabel 7 Hasil Pengujian *Behavior Testing*

| Input   | Hasil Harapan  | Output   | Kesimpulan                |
|---|--|--|---------------------------|
| Show hasil penambahan data <i>users</i> secara berulang – ulang menjadi # | Sistem berhasil menyimpan data dan tidak mengalami data <i>stack</i> Jumlah = 11 halaman 107 <i>users</i> berhasil ditambahkan | Sistem berhasil menyimpan data sebanyak lebih dari 50 dan sistem tidak mengalami kendala | [√] Berhasil<br>[ ] Gagal |

### 7. Performance Testing

Pengujian ini mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari aliran pemakaian memori, aliran data dan kecepatan eksekusi. Ujicoba pemakaian memori dilakukan pada web browser yang sejenis.

Tabel 8 Hasil Pengujian *Performance Testing*

| Input                         | Hasil Harapan  | Output   | Kesimpulan                |
|-------------------------------|--|--|---------------------------|
| Search dengan kata “ penting” | Sistem menampilkan seluruh data surat yang berhubungan dengan kata penting | Sistem berhasil menampilkan data surat yang dicari | [√] Berhasil<br>[ ] Gagal |

**8. Requirement Testing**

Tipe ini tidak dinamakan pengujian karena tipe ini hanya melihat spesifikasi kebutuhan dari SIKEMAS mulai dari sistem pembuatan sampai pengujian. Kebutuhan perangkat lunak dan perangkat keras yang digunakan akan dijelaskan pada tabel di tipe ini.

Tabel 9 Hasil Pengujian *Requirement Testing*

| Perangkat Keras                                  | Database | Bahasa Pemrograman | Metode Pengujian                          |
|--|----------|--------------------|---|
| Prosesor Intel inside core i3<br><br>Memory 4 GB | Mysql    | Php                | Blax Box Testing dengan 10 tipe pengujian |

**9. Endurance Testing**

Tipe ini untuk memastikan apakah hasil operasi matematika pada sistem SIKEMAS ini benar atau salah, dilihat dari setiap menu atau sub menu yang terdapat jumlah atau hasil keseluruhan surat

Tabel 10 Hasil Pengujian *Endurance Testing*

| Input   | Hasil Harapan                         | Output  | Kesimpulan                |
|---|---------------------------------------|---|---------------------------|
| Jumlah keseluruhan dari sistem sama dengan hasil manual | Hasil penjumlahan sama yaitu 68 surat | Hasil penjumlahan sama dengan hasil yang diharapkan | [√] Berhasil<br>[ ] Gagal |

**10. Cause-Effect Relationship Testing**

Pengujian yang melibatkan kondisi input dan aliran data mulai dari *Input, View, Update, Delete* dan *Search*.

Tabel 11 Hasil Pengujian Cause-Effect Relationship Testing

| Input  | Hasil Harapan  | Output  | Kesimpulan                |
|--|--|---|---------------------------|
| Hapus data surat dengan no surat @#. %%/ *# @ /# @ !%/ % %/ * & % \$ | Sistem berhasil menghapus data dan muncul notifikasi “ data berhasil dihapus “ | Sistem memunculkan notifikasi dan data nomor surat @#. %%/ *# @ /# @ !%/ % %/ * & % \$ tidak terhapus | [ ] Berhasil<br>[√] Gagal |

**III. Simpulan**

Setelah dilakukan pembahasan dan pengujian terhadap sistem informasi surat keluar masuk, maka beberapa kesimpulan sebagai berikut :

1. Pengujian ini dapat membantu menemukan kesalahan pada sistem informasi yang sudah dibuat sebelum sistem informasi digunakan.
2. Mempermudah seseorang pengembang sistem untuk mengembangkan kesalahan yang telah ditemukan pada pengujian.
3. Pada sistem informasi ini digunakan metode *Black Box* dengan 10 jenis tipe pengujian dengan hasil yang baik, sehingga beberapa kesalahan atau kelemahan dalam sistem informasi dapat ditemukan.

**IV. Daftar Pustaka**

[1] Andika, Radenal. “Penerapan CI Dalam Pengembangan Sistem Informasi Manajemen Surat dan Pengarsipan“ Studi Kasus : PT. Semen Padang. 2011

[2] Ferdinandus, Sandy. Perancangan Aplikasi Surat Masuk Dan Surat Keluar Pada PT. PLN (Persero) Wilayah Suluttenggo. 2015.

[3] Junidar. Perancangan informasi arsip surat menyurat di universitas u’budiyah indonesia. 2012.

[4] Kurniawan, Yoga. “Rancang Bangun Perangkat Lunak Untuk Workflow Pengelolaan Surat Menyurat Dinas Bagian Surat Masuk di Kabupaten Buton Utara”. 2012.

- [5] Rouf, Abdul. Pengujian perangkat lunak dengan menggunakan metode white box dan black box. 2012.
- [6] Santoso, Arum Tungga Dewi. Sistem informasi administrasi surat masuk dan surat keluar pada badan kepegawaian daerah kota semarang. 2014.
- [7] Sutono, D. Sistem Informasi manajemen. 2007.
- [8] Supardi, Julian. Materi Kuliah Black-Box Testing. 2015.
- [9] Syaban, Rizki Maulana Syaban, H.Bunyamin. “Pengembangan sistem informasi pengelolaan surat masuk dan keluar berbasis web di dinas sosial tenaga kerja dan transmigrasi kabupaten garut menggunakan framework php”. 2015.
- [10] Zulkifli. Model Prediksi Berbasis Neural Network untuk Pengujian Perangkat Lunak Metode Black-Box. “Seminar Nasional Aplikasi Teknologi Informasi (SNATI)”.2013.

Halaman ini sengaja dikosongkan.