Jurnal Teknologi Sistem Informasi dan Aplikasi **ISSN: 2654-3788**
Penerbit: Program Studi Teknik Informatika Universitas Pamulang **e-ISSN: 2654-4229**
Vol. 4, No. 1, Januari 2021 (58-64) **DOI: 10.32493/jtsi.v4i1.9074**

Pengujian Black Box pada Aplikasi Keamanan Data *Multimedia Message Service* (MMS) Berbasis Android Menggunakan Teknik *Equivalence Partitions*

Abdul Aziz[1]
[1]Pusat Teknologi Penerbangan, LAPAN, Bogor, Indonesia, 16350
e-mail: [1]abdul.aziz@lapan.go.id

Submitted Date: January 12th, 2020 Reviewed Date: January 13th, 2021
Revised Date: March 31st, 2021 Accepted Date: April 01st, 2021

Abstract

Currently, almost everyone has used electronic devices in which there are applications that have been installed. To meet user satisfaction, this application must go through a testing process to check all errors in the system so that they can be fixed later. This test is performed on a previously created Data Security Application. Software testing that has been done is using the Black Box method with the Equivalence Partition technique. From the test results, it was found that cases that need to be fixed are the fields in the Create Message Form.

Keywords: Application; *Testing*; *Black Box*; *Equivalence partitioning*

Abstrak

Saat ini hampir setiap orang telah menggunakan perangkat elektronik yang didalamnya terdapat aplikasi yang telah terpasang. Untuk memenuhi kepuasan penggunanya, aplikasi ini harus melalui proses pengujian untuk memeriksa semua kesalahan yang ada pada sistem agar dapat diperbaiki nantinya. Pengujian ini dilakukan pada Aplikasi Keamanan Data yang telah dibuat sebelumnya. Pengujian perangkat lunak yang telah dilakukan yaitu menggunakan metode *Black Box* dengan teknik *Equivalence Partition*. Dari hasil pengujian ditemukan kasus yang perlu diperbaiki yaitu kolom isian pada *Form* Buat Pesan.

Kata Kunci: Aplikasi; *Testing*; *Black Box*; *Equivalence partitioning*

1. Pendahuluan

Pada masa milenial ini, hampir semua orang telah mengenal *gadget* atau perangkat elektronik. Untuk dapat menjalankan sistem yang ada pada perangkat tersebut diperlukan namanya perangkat lunak. Perangkat lunak yang terdapat pada perangkat tersebut salah satunya yaitu aplikasi.

Untuk menentukan aplikasi dapat berjalan dengan baik salah satunya dengan melakukan pengujian. Pengujian perangkat lunak adalah cara untuk mendapatkan informasi mengenai kualitas dari perangkat lunak yang sedang diuji (Sulistyanto & Azhari, 2014). Pengujian pada sebuah program sangat penting untuk dilakukan untuk memeriksa semua kesalahan yang ada pada program tersebut agar tidak terjadi kerugian yang akan ditimbulkan dari kesalahan tersebut, sehingga sangat perlu untuk melakukan pengujian untuk mengurangi terjadinya kesalahan yang merugikan tersebut (Ningrum, Suherman, Aryanti, Prasetya, & Saifudin, 2019). Pengujian perangkat lunak dilakukan untuk mendeteksi adanya kesalahan, yang menyebabkan kegagalan perangkat lunak (Irawan, 2017). Pengujian perangkat lunak juga bertujuan untuk memperoleh produk yang berkualitas yang memberikan produktivitas tinggi. Dalam proses pengujian perangkat lunak, untuk setiap kasus yang akan diuji harus memiliki identitas dan mempunyai keterhubungan antara sekumpulan masukan dengan hasil yang diinginkan (Komarudin MZ, 2016). Kepuasan pelanggan tergantung pada kualitas perangkat lunak dan kualitas sejumlah perangkat lunak perlu dijaga dengan beberapa alasan (Cholifah, Yulianingsih, & Sagita, 2018).

Pengujian aplikasi ini menggunakan metode *Black Box*. *Black Box* adalah teknik pengujian yang berfokus pada spesifikasi fungsional dari

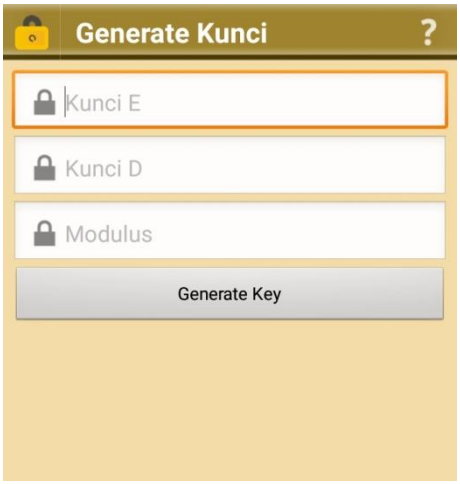
perangkat lunak, penguji dapat mendefinisikan kumpulan kondisi masukan dan melakukan pengetesan pada spesifikasi fungsional program (Hidayat & Muttaqin, 2018). Metode *Black Box Testing* adalah sebuah metode yang digunakan untuk menguji sebuah perangkat lunak tanpa harus memperhatikan hal detail perangkat lunak. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing (Hanifah, Alit, & Sugiarto, 2016). Tujuan *Black Box Testing* untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang telah diharapkan dan apakah informasi yang disimpan serta eksternal selalu dijaga kemutakhirannya (Maharani & Merlina, 2014).

Dalam melakukan pengujian black box menggunakan teknik *Equivalence Partitions*. *Metode Equivalence Partitions* adalah metode pengujian *Black Box* yg memecah atau membagi domain masukan dari program ke dalam kelas-kelas data sehingga *Test Case* dapat diperoleh (Krismadi, et al., 2019).

2. Metodologi

Dalam penelitian ini akan dilakukan beberapa tahapan. Pada tahapan pertama yaitu dengan membuat rancangan *test case* berdasarkan fungsi yang ada dalam pengujian aplikasi. Lalu membuat batasan pengujian *Equivalence Partioning*, setelah itu membuat Batasan pengujian, dan langkah selanjutnya adalah membuat model pengujian dari skenario pengujian dan hasil yang diharapkan, dan yang terakhir yaitu melakukan pengujian berdasarkan model yang telah dirancang pada rancangan *test case*. Hal ini dilakukan untuk mendapatkan data berupa dokumentasi pengujian dengan metode *Equivalence Partitions* dan nilai tingkat efektifitas metode *Equivalence Partitions* (Jaya, Gumilang, Wati, Andersen, & Desyani, 2019).

Pada Gambar 1 menampilkan *Form Generate Key* yang berfungsi untuk membangkitkan kunci secara otomatis oleh sistem yang akan digunakan untuk proses enkripsi dan dekripsi kunci *blowfish*. Kunci yang dibangkitkan yaitu kunci e, kunci d dan modulus. Kunci e dan modulus akan digunakan untuk proses enkripsi kunci *blowfish* menggunakan algoritma RSA. Sedangkan untuk proses dekripsi kunci *blowfish* menggunakan kunci d dan modulus.



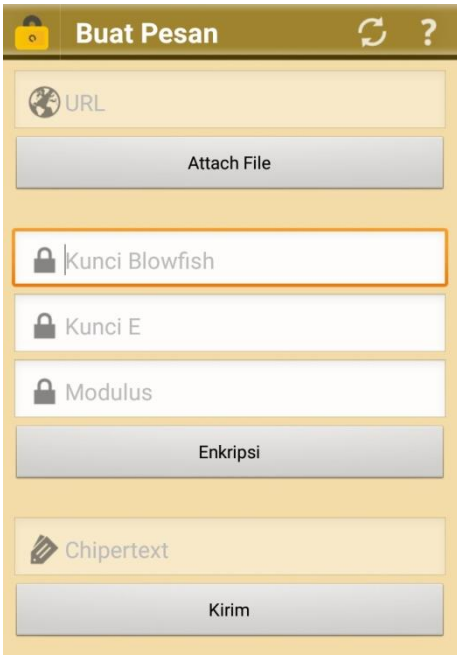
Gambar 1. Form Generate Key

Berdasarkan form pada Gambar 1 terdapat rencana pengujian yaitu dengan menekan tombol “Generate Key”, dan sistem secara otomatis akan menampilkan kunci e, kunci d dan modulus sesuai kolom yang tersedia. Jika salah satu kolom tidak muncul kunci, maka terdapat *error* pada sistem tersebut.

Table 1. Rancangan Test Case Form Generate Key

Id	Deskripsi	Hasil yang diharapkan
A01	Klik tombol “Generate Key”	Sistem akan menampilkan Kunci E, Kunci D dan Modulus

Pada Gambar 2 menampilkan *Form* Buat Pesan yang berfungsi untuk melakukan proses enkripsi sebelum pesan dikirimkan. Proses enkripsi yang dilakukan oleh sistem pertama kali yaitu dengan mengenkripsi *file* menggunakan algoritma *blowfish*, lalu kunci yang digunakan untuk mengenkripsi file, dienkripsi menggunakan algoritma RSA.



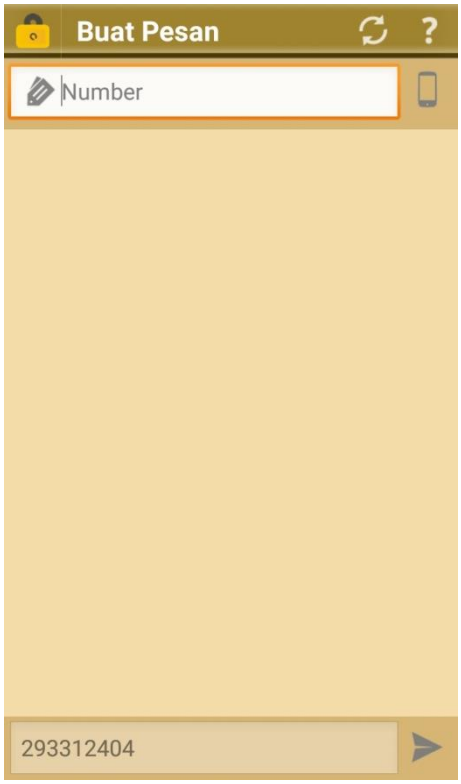
Gambar 2. Form Buat Pesan

Berdasarkan Gambar 2 terdapat beberapa rencana pengujian. Semua kolom isian tidak boleh dibiarkan kosong. Pada kolom URL diisi dengan berkas *microsoft office* seperti *microsoft word* dan *microsoft power point*. Pada kolom kunci *blowfish* diisi teks dengan maksimal 4 karakter. Pada kolom kunci e dan modulus diisi dengan angka yang telah dibangkitkan sebelumnya pada *form generate key*.

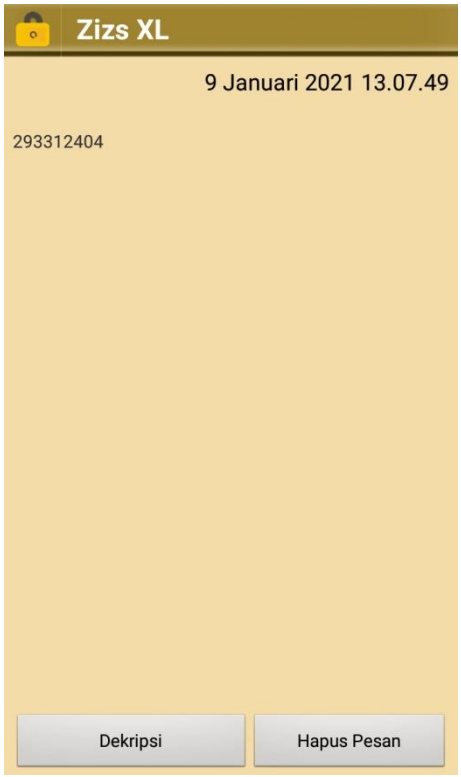
Table 2. Rancangan Test Case Form Buat Pesan

Id	Deskripsi	Hasil yang diharapkan
B01	Mengisi URL dengan file "Test 1.doc", lalu masukkan sandi "teknik " pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Enkripsi"	File dan sandi terenkripsi menjadi chipertext
B02	Mengosongkan URL, lalu masukkan sandi "teknik " pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Enkripsi"	Sistem tidak akan melakukan enkripsi
B03	Mengisi URL dengan file "Test 1.doc", lalu mengosongkan sandi pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Enkripsi"	Sistem tidak akan melakukan enkripsi
B04	Jika proses berhasil dilakukan, lalu klik tombol "Kirim"	Sistem akan menampilkan halaman Kirim Pesan

Pada Gambar 3 menampilkan *Form* Kirim Pesan yang berfungsi untuk mengirimkan pesan *chipertext* dari hasil enkripsi pada form buat pesan.



Gambar 3. Form Kirim Pesan



Gambar 4. Form Kotak Masuk

Berdasarkan form pada Gambar 3 terdapat beberapa rencana pengujian. Pada kolom number diisi dengan nomor telepon yang akan dituju. Kolom number tidak boleh dibiarkan kosong. Sedangkan pada kolom pesan merupakan *chipertext* yang telah dienkripsi sebelumnya.

Table 3. Rancangan Test Case Form Kirim Pesan

Id	Deskripsi	Hasil yang diharapkan
C01	Mengisi "0896xxxxxxxx" pada "Number", setelah itu tekan ikon tombol kirim	Pesan dapat terkirim pada nomor yang dituju
C02	Mengosongkan "Number", setelah itu tekan ikon tombol kirim	Sistem akan memberi informasi peringatan

Pada Gambar 4 menampilkan Form Kotak Masuk yang berfungsi untuk menerima *chipertext* kunci blowfish untuk digunakan nantinya pada proses dekripsi *file*.

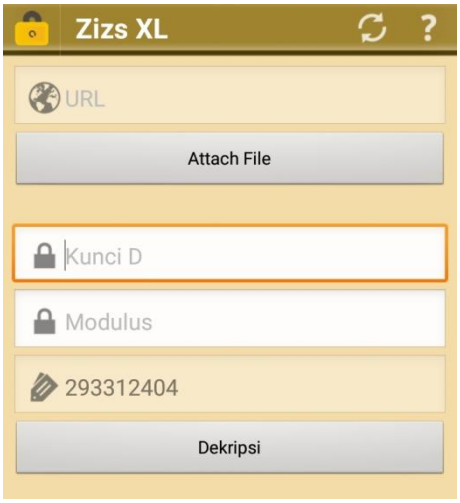
Berdasarkan halaman pada Gambar 4 terdapat rencana pengujian yaitu dengan menekan tombol "Dekripsi", dan sistem secara otomatis akan menampilkan halaman Form Dekripsi Pesan.

Table 4. Rancangan Test Case Form Kotak Masuk

Id	Deskripsi	Hasil yang diharapkan
D01	Klik tombol "Dekripsi"	Sistem akan menampilkan halaman Dekripsi Pesan

Pada Gambar 5 menampilkan *Form* Dekripsi Pesan yang berfungsi untuk melakukan proses dekripsi *file* dengan sandi yang telah diterima. Sandi yang diterima tersebut dalam bentuk *chipertext* atau dalam kondisi terenkripsi dan diperlukan proses dekripsi terlebih dahulu menggunakan algoritma RSA. Setelah sandi

berhasil didekripsi maka sandi dapat digunakan untuk mendekripsi file yang telah terenkripsi.



Gambar 5. Form Dekripsi Pesan

Berdasarkan Gambar 5 terdapat beberapa rencana pengujian. Semua kolom isian tidak boleh dibiarkan kosong. Pada kolom URL diisi dengan berkas *microsoft office* yang telah terenkripsi sebelumnya. Pada kunci d dan modulus diisi dengan kunci yang telah dibangkitkan oleh pengirim pesan.

Table 5. Rancangan Test Case Form Dekripsi Pesan

Id	Deskripsi	Hasil yang diharapkan
E01	Mengisi URL dengan file "Encrypt Test 1.doc", lalu masukkan "501487583030282559" pada kunci d, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Dekripsi"	Chipertext menjadi plaintext dan mendekripsi File menjadi plaintext
E02	Mengosongkan URL, lalu masukkan "501487583030282559" pada kunci d, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Dekripsi"	Sistem tidak akan melakukan dekripsi

3. Hasil dan Pembahasan

Setelah melakukan perancangan pengujian, maka tahapan selanjutnya yaitu melakukan pengujian untuk menentukan sistem dapat berjalan sesuai dengan harapan yang direncanakan.

Table 6. Hasil Pengujian *Equivalence Partitioning*

Id	Deskripsi	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
A01	Klik tombol "Generate Key"	Sistem akan menampilkan Kunci E, Kunci D dan Modulus		Berhasil
B01	Mengisi URL dengan file "Test 1.doc", lalu masukkan sandi "teknik " pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Enkripsi"	File dan sandi terenkripsi menjadi chipertext	File dan sandi terenkripsi. *Chipertext* sandi ditampilkan pada Form Buat Pesan	Berhasil
B02	Mengosongkan URL, lalu masukkan sandi "teknik " pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol	Sistem tidak akan melakukan enkripsi	Muncul *toast* "URL tidak boleh kosong"	Berhasil

	"Enkripsi"			
B03	Mengisi URL dengan file "Test 1.doc", lalu mengosongkan sandi pada "Kunci Blowfish", masukkan "35879" pada kunci e, dan masukkan "10128270772511630 1" pada modulus, setelah itu tekan tombol "Enkripsi"	Sistem tidak akan melakukan enkripsi	Muncul *toast* "Sandi tidak boleh kosong"	Berhasil
B04	Jika proses berhasil dilakukan, lalu klik tombol "Kirim"	Sistem akan menampilkan halaman Kirim Pesan	Muncul form Kirim Pesan yang berisi *chipertext* sandi pada isian pesan	Berhasil
C01	Mengisi "0896xxxxxxxx" pada "Number", setelah itu tekan ikon tombol kirim	Pesan dapat terkirim pada nomor yang dituju	Pesan berhasil terkirim dan dapat diterima oleh nomor yang dituju	Berhasil
C02	Mengosongkan "Number", setelah itu tekan ikon tombol kirim	Sistem akan memberi informasi peringatan	Muncul *toast* "Number tidak boleh kosong"	Berhasil
D01	Klik tombol "Dekripsi"	Sistem akan menampilkan halaman Dekripsi Pesan	Muncul form Dekripsi Pesan yang berisi terusan pesan dari kotak masuk	Berhasil
E01	Mengisi URL dengan file "Encrypt Test 1.doc", lalu masukkan "501487583030282559" pada kunci d, dan masukkan "101282707725116301" pada modulus, setelah itu tekan tombol "Dekripsi"	Chipertext menjadi plaintext dan mendekripsi File menjadi plaintext	File dapat dibuka dan terbaca	Berhasil
E02	Mengosongkan URL, lalu masukkan "501487583030282559" pada kunci d, dan masukkan "10128270772511630 1" pada modulus, setelah itu tekan tombol "Dekripsi"	Sistem tidak akan melakukan dekripsi	Muncul *toast* "URL tidak boleh kosong"	Berhasil

Pada aplikasi keamanan data ini, telah dilakukan pengujian pada 5 form atau halaman aplikasi. Pada Form *Generate Key* dilakukan pengujian sebanyak sekali. Pada Form Buat Pesan dilakukan pengujian sebanyak 4 kali. Pada Form Kirim Pesan dilakukan pengujian sebanyak 2 kali. Pada Form Kotak Masuk dilakukan pengujian sebanyak sekali. Pada Form Dekripsi Pesan dilakukan pengujian sebanyak 2 kali. Jadi total pengujian yang telah dilakukan yaitu sebanyak 10 kali pengujian.

4. Kesimpulan

Dari hasil pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi keamanan data telah berjalan sesuai dengan harapan yang diinginkan. Namun saat proses pengujian berlangsung, terdapat *case* yang perlu diperbaiki agar sistem semakin baik yaitu pada form isian kunci *blowfish* yang hanya dapat menerima 4 karakter isian. Jadi pengujian *black box* dengan menggunakan teknik *equivalance partitions* dapat membantu proses pembuatan kasus pengujian dan menentukan kualitas sistem dan menemukan kesalahan yang ada, serta menjamin aplikasi yang diuji sesuai dengan harapan yang diinginkan.

5. Saran

Dari hasil pengujian yang dilakukan, teknik *equivalance partitions* dapat membantu untuk menentukan kualitas dari sistem dan menemukan kesalahan yang ada. Adapun saran untuk penelitian selanjutnya yaitu dengan memperbanyak kasus uji yang dilakukan, agar semakin banyak kesempatan celah yang dapat

ditemukan pada sistem aplikasi tersebut untuk dapat diperbaiki.

References

Cholifah, W. N., Yulianingsih, & Sagita, S. M. (2018). Pengujian Black Box Testing pada Aplikasi Action & Strategy Berbasis Android dengan Teknologi Phonegap. *Jurnal String*, 206-210.

Hanifah, U., Alit, R., & Sugiarto. (2016). Penggunaan Metode Black Box Pada Pengujian Sistem Informasi Surat Keluar Masuk. *SCAN*, 33-40.

Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jurnal Teknik Informatika UNIS*, 25-29.

Irawan, Y. (2017). Pengujian Sistem Informasi Pengelolaan Pelatihan Kerja Upt. BLK Kabupaten Kudus dengan Metode Whitebox Testing. *Sentra Penelitian Engineering dan Edukasi*, 1-5.

Jaya, M. S., Gumilang, P., Wati, T., Andersen, Y. P., & Desyani, T. (2019). Pengujian Black Box pada Aplikasi Sistem Penunjang Keputusan Seleksi Calon Pegawai Negeri Sipil Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 131-136.

Komarudin MZ, M. (2016). Pengujian Perangkat Lunak Metode Black-Box Berbasis Equivalence Partitions Pada Aplikasi Sistem Informasi Sekolah. *Jurnal Mikrotik*, 1-18.

Krismadi, A., Lestari, A. F., Pitriyah, A., Mardangga, I. W., Astuti, M., & Saifudin, A. (2019). Pengujian Black Box berbasis Equivalence Partitions pada Aplikasi Seleksi Promosi Kenaikan Jabatan. *Jurnal Teknologi Sistem Informasi dan Aplikasi*, 155-161.

Maharani, M., & Merlina, N. (2014). Penerapan Metode Straight Selection Pada Sistem Parkir Universitas Bina Nusantara. *Jurnal Pilar Nusa Mandiri*, 95-100.

Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 125-130.

Sulistyanto, H., & Azhari. (2014). Urgensi Pengujian Pada Kemajuan Perangkat Lunak Dalam Multi Perspektif. *Jurnal Komunikasi dan Teknologi Informasi*, 1-10.