Pengujian *Black Box* Menggunakan Metode *Cause Effect Relationship Testing*

Nadya Safitri [1,*], Rully Pramudita [2]

[1] Sistem Informasi; STMIK Bina Insani; Jl. Siliwangi No 6 Rawa Panjang Bekasi Timur 17114 Indonesia. Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: nadyasafitri@binainsani.ac.id

[2] Teknik Informatika; STMIK Bina Insani; Jl. Siliwangi No 6 Rawa Panjang Bekasi Timur 17114 Indonesia. Telp. (021) 824 36 886 / (021) 824 36 996. Fax. (021) 824 009 24; e-mail: rullypramudita@binainsani.ac.id

* Korespondensi: e-mail: nadyasafitri@binainsani.ac.id

Diterima: 20 November 2018; Review: 24 November 2018 ; Disetujui: 28 November 2018

Cara sitasi: Safitri N, Pramudita R. 2018. Pengujian *Black Box* Menggunakan Metode *Cause-Effect Relationship Testing*. Information System For Educators and Professionals. 3 (1): 101 – 110.

Abstrak: Pengujian merupakan sebuah tahapan yang dilakukan oleh *tester* untuk menemukan kesalahan yang ada di suatu perangkat lunak. Salah satu teknik *testing* yang sering digunakan yaitu *black box testing*. Pada penelitian ini akan dicoba menerapkan teknik *black box testing*. Teknik *black box testing* terdiri dari beberapa cara, yaitu *Equivalence Partitioning, Boundary Value Analysis / Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing.Cause Effect Relationship* merupakan cara pengujian dengan melakukan ujicoba yang dilakukan berdasarkan kondisi logikal dan aksi yang berhubungan. Hasil pengujian ini memberikan kesimpulan mengenai kebenaran fungsi dari aplikasi yang diuji serta apa saja kesalahan yang masih terdapat didalamnya.

Kata kunci: Aplikasi, *Black Box, Cause, Effect,* Pengujian

Abstract: *Testing is the stage carried out by the tester to find errors that exist in the software. One testing technique that is often used is black box testing. In this study we will try to apply the black box testing technique. Black box testing techniques consist of several ways, namely Equivalence Partitioning, Boundary Value Analysis / Boundary Testing, Comparative Testing, Sample Testing, Robustness Testing, Behavioral Testing, Needs Testing, Performance Testing, Durability Testing, Testing of Cause and Effect Relations. how to test by conducting tests based on logical conditions and related actions. The results of this test provide conclusions about the functions of the applications that support and what is still in it.*

Keywords*: Application, Black box, Cause, Effect, Testing*

1. Pendahuluan

Aplikasi Revo *Uninstaller* merupakan aplikasi gratis berbasis desktop yang digunakan untuk melakukan penghapusan atau *uninstaller* sebuah aplikasi, selain itu juga merupakan aplikasi unistaller yang inovatif karena menggunakan algoritma tingkat lanjut yang cepat. Aplikasi ini memulai scan aplikasi yang ter*install* di komputer pada saat aplikasi mulai dijalankan.

Kelebihan lain aplikasi ini adalah setelah proses program *uninstaller* dijalankan, aplikasi akan menghapus *file* tambahan yang tidak diperlukan, *folder*, dan *registry* yang biasanya tersisa pada komputer. Bahkan jika pada proses *uninstaller* mengalami kegagalan aplikasi ini akan

E-ISSN: 2548-3587 ; 101 – 110

memindai data aplikasi pada drive dan dalam window *registry*, menampilkan semua *file* yang ditemukan, *folder* dan item *registry* sehingga dapat dengan mudah untuk menghapusnya.

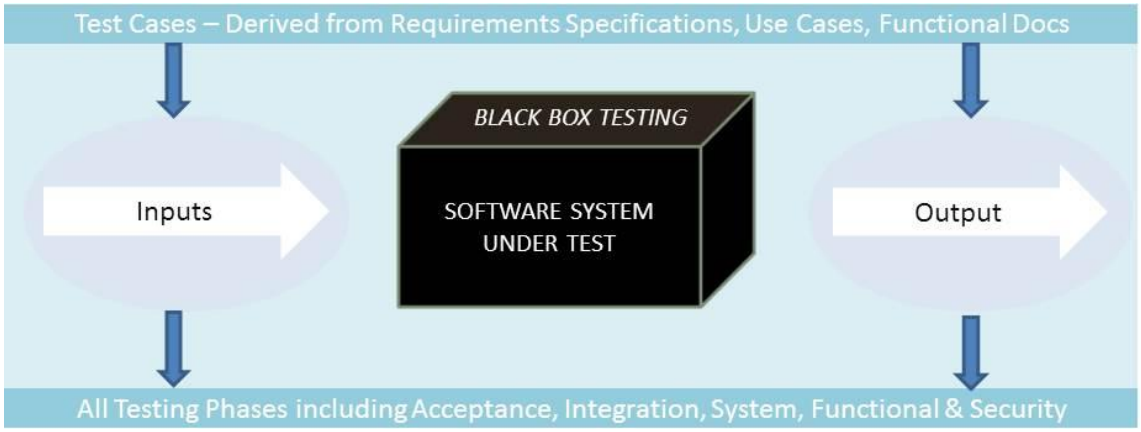
Pada aplikasi ini terdapat beberapa fungsi utama, diantaranya adalah unistall aplikasi, menghentikan aplikasi, menghapus aplikasi serta dapat pula menonaktifkan aplikasi yang sebelumnya telah ter*uninstaller* di komputer.

Aplikasi ini memiliki fitur-fitur diantaranya yaitu 1) *real time uninstaller action monitor* yang berfungsi untuk *uninstaller* secara keseluruhan, 2) *Forced Uninstaller* berfungsi untuk menghapus aplikasi yang sulit untuk di *uninstaller* , 3) *Quick / Multiple uninstaller* berfungsi untuk menghapus aplikasi dengan cepat dari satu atau lebih aplikasi, 4) *Junk File Cleaner* berfungsi untuk menghilangkan *file-file* tidak penting, 5) *Auto-run Manager* berfungsi untuk mengatur start awal windows, 6) *Advanced Scanning for Leftovers* berfungsi untuk menghapus sisa-sisa dari aplikasi, 7) *Manage uninstaller ation logs* berfungsi untuk edit, tinjau, bagikan (ekspor, impor) log, 8) *Multi-Level Backup System* berfungsi untuk menghapus program dengan cara aman, 9) *Windows Tools* merupakan alat Windows yang bermanfaat di satu tempat, 10) *Browsers Cleaner* berfungsi untuk hapus riwayat browser Internet, 11) *MS Office Cleaner* berfungsi untuk hapus riwayat microsoft office, 12) *Windows Cleaner* berfungsi untuk hapus *file* sampah dari Windows Anda, 13) *Evidence Remover* berfungsi untuk Penghilang Bukti, 14) *Unrecoverable Delete* berfungsi untuk menghapus *file* dan *folder* secara permanen.

Dengan adanya aplikasi tersebut, maka perlu dilakukannya pengujian untuk memastikan atau menguji fungsi yang tersedia berjalan sesuai dengan yang diinginkan. Namun untuk pengujian ini hanya akan dilakukan terhadap lima fungsi saja yaitu fungsi *search, uninstaller , quick uninstaller , force uninstaller* dan *junk file cleaner*. Pengujian adalah suatu proses pelaksanaan atau pengecekan suatu program dengan tujuan menemukan suatu kesalahan [Mustaqbal et al., 2015]. Black box *testing* penting dilakukan jika kode perangkat lunak tidak tersedia selama fase pengujian [Khanna, 2017]. Pengujian *black box* merupakan salah satu jenis metode pengujian yang memperlakukan perangkat lunak yang tidak diketahui kinerja internalnya [Salamah and Khasanah, 2017].

2. Metode Penelitian

Metode yang digunakan di penelitian ini yaitu *black box testing*. *Black box testing* berfokus pada pengujian dari masing-masing spesifikasi fungsional perangkat lunak. Seorang *tester* dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada fungsional perangkat lunak [Mustaqbal et al., 2015].



Sumber : Noeticforce (2017)

Gambar 1. *Black Box Testing*

Black box testing memiliki beberapa metode atau teknik dalam melakukan pengujiannya, diantaranya ialah *Equivalence Partitioning, Boundary Value Analysis atau Limit Testing, Comparison Testing, Sample Testing, Robustness Testing, Behavior Testing, Requirement Testing, Performance Testing, Endurance Testing, Cause-Effect Relationship Testing.*

Berdasarkan banyaknya metode *black box testing*, yang akan digunakan dalam pengujian ini yaitu menggunakan metode *cause effect relationship testing*.

Cause Effect Relationship Testing It is technique of software test design that includes identifying the cases (Input conditions) and the effects (Output conditions) [Babbar, 2017], merupakan teknik uji coba yang merepresentasikan kondisi logika serta aksi. Adapun tahapan yang diantaranya yaitu: 1) Tentukan kondisi input (cause) dan kondisi output (effect), 2) Buat contoh kasus untuk melakukan pengujian, 3) Membuat tabel uji.

3. Hasil dan Pembahasan

Dalam subbab ini akan dibahas mengenai hasil dari penelitian yang telah dilakukan. Pengujian dilakukan dengan menguji fungsi-fungsi yang terdapat dalam aplikasi. Beberapa fungsi yang akan diuji dapat dilihat di tabel 1.

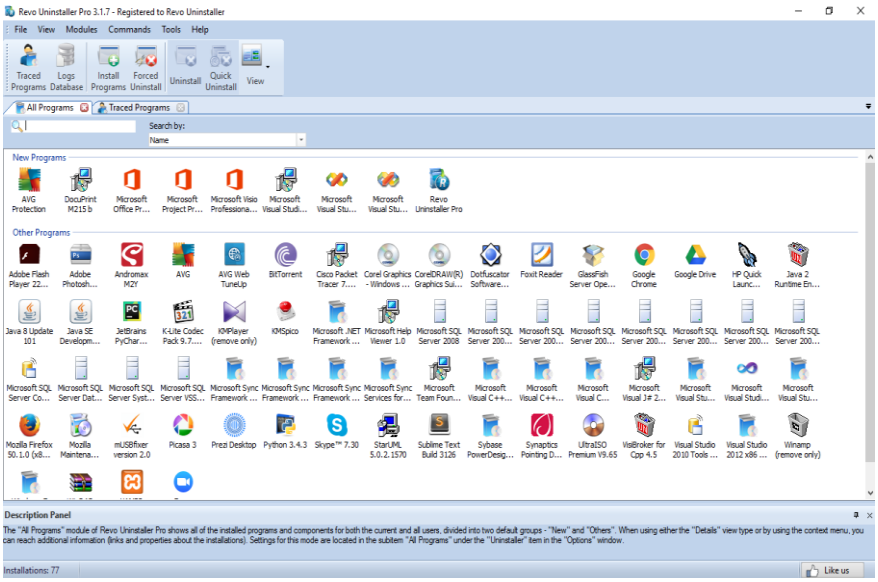
Tabel 1. Daftar Pengujian Fungsi

No	Nama Fungsi
1	*Search*
2	*Uninstaller*
3	Quick *Uninstaller*
4	Force *Uninstaller*
5	*Junk File* Cleaner

Sumber : Penelitian(2018)

3.1. Fungsi *Search*

Fungsi pertama yang akan diuji yaitu fungsi *search*/pencarian. Pada gambar 2 terlihat tampilan dari menu pencarian yang selanjutnya akan dilakukan pengujian.



Sumber : Penelitian(2018)

Gambar 2. Fungsi *Search*

Fungsi pencarian ini berfungsi untuk melakukan pencarian terhadap aplikasi-aplikasi yang ada di dalam sistem komputer yang digunakan.

a. Kondisi *Input dan output*

Tahap pertama yang dilakukan yaitu menentukan kondisi *input dan output*. Dalam tabel 2 dapat dilihat bahwa kondisi input ada satu yaitu *search* yang menandakan bahwa inputan pencarian bertipe char. Kondisi output ada dua yaitu 1) Kondisi dimana pencarian ditemukan dan 2) pencarian tidak ditemukan.

E-ISSN: 2548-3587 ; 101 – 110

Tabel 2. Kondisi *Input dan output*

Input State:	Output State:
1. *Search* = char	A. Tampil hasil pencarian B. Muncul text pesan “Found 0 program”

Sumber: Penelitian (2018)

b. Pengujian Kasus

Tahap kedua yaitu menentukan uji kasus yang akan digunakan untuk menguji program. Yang nantinya akan diuji berdasarkan kondisi *input dan output* yang telah ditentukan diawal. Kasus yang dijadikan contoh ada tiga diantaranya yaitu 1) Tes 1 : AVG, 2) Tes 2 : guguk, 3) Tes 3 : _

c. Tabel Uji

Langkah terkahir yaitu mebuat tabel uji yang akan menerapkan uji kasus berdasarkan kondisi *input dan output* ditahap sebelumnya serta kasus-kasus uji yang sudah ditentukan. Agar lebih jelasnya maka akan dibuatkan dalam sebuah tabel matriks uji fungsi. Berikut ini tabel 3 yang menjelaskan tabel uji dari fungsi *search*.

Tabel 3. Tabel Uji Fungsi *Search*

	Tes 1	**Tes 2**	**Tes 3**
1	1	1	1
A	1	0	0
B	0	1	1

Sumber: Penelitian (2018)

3.2. *Uninstaller*

Fungsi kedua yang akan diuji yaitu fungsi *uninstaller* . Pada gambar 3 terlihat tampilan dari menu *uninstaller* yang selanjutnya akan dilakukan pengujian. Menu *uninstaller* ini berfungsi untuk melakukan penghapusan sejumlah program yang ada di komputer jika sudah tidak diperlukan kembali.

Fungsi *uninstaller* yang ada di aplikasi revo ini memiliki kelebihan dibandingkan dengan fungsi sejenis yang ada di bawaan sebuah sistem operasi yang berada di sebuah komputer, yaitu memiliki kemampuan penghapusan yang jauh lebih bersih, sehingga tidak menyisakan *file-file* temporary ataupun *file* sampah akibat proses penggunaan aplikasi di komputer tersebut. Namun daripada itu tetap saja diperlukan pengujian terhadap fungsi ini agar lebih memastikan kebenaran dari fungsi tersebut. Selanjutnya akan dilakukan pengujian sesuai tahapan yang ada di *cause effect testing*, dimulai menentukan kondisi *input dan output*, menentukan uji kasus dan membuat tabel uji fungsi.

a. Kondisi *Input dan output*

Tahap pertama yang dilakukan yaitu menentukan kondisi *input dan output*. Dalam tabel 4 dapat dilihat bahwa kondisi input ada satu yaitu *uninstaller* yang menandakan bahwa inputan pencarian bertipe boolean. Kondisi output ada dua yaitu kondisi dimana unistall dilakukan atau batal dilakukan.

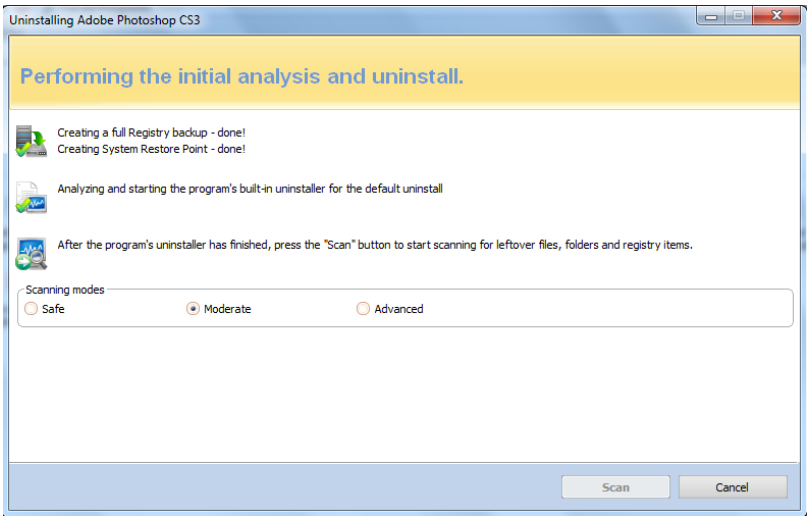
Tabel 4. Kondisi *Input dan output*

Input State :	Output State :
1. *Uninstaller* = boolean	A. Tampil lanjutkan *uninstaller* program B. Batalkan *uninstaller*

Sumber: Penelitian (2018)

b. Pengujian Kasus

Tahap kedua yaitu menentukan uji kasus yang akan digunakan untuk menguji program. Yang nantinya akan diuji berdasarkan kondisi *input dan output* yang telah ditentukan diawal. Kasus yang dijadikan contoh ada dua diantaranya yaitu 1) Tes 1 : Klik Continue, 2) Tes 2 : Klik Cancel.



Sumber: Penelitian (2018)

Gambar 3. Fungsi *Uninstaller*

Terlihat pada gambar 3, fitur *uninstaller* yang ada pada aplikasi revo untuk dilakukan pengujian *black box* terhadap fitur tersebut.

c. Tabel Uji

Langkah terkahir yaitu mebuat tabel uji yang akan menerapkan uji kasus berdasarkan kondisi *input dan output* ditahap sebelumnya. Berikut ini tabel 5 yang menjelaskan tabel uji dari fungsi *uninstaller* .

Tabel 5. Tabel Uji Fungsi *Uninstaller*

	Tes 1	Tes 2
1	1	1
A	1	0
B	0	1

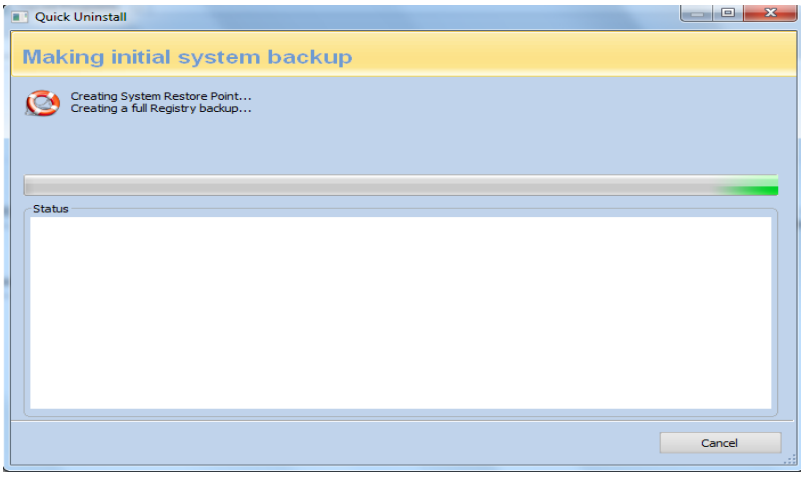
Sumber: Penelitian (2018)

3.3. Quick *Uninstaller*

Fungsi ketiga yang akan diuji yaitu fungsi quick *uninstaller* . Pada gambar 4 terlihat tampilan dari menu *uninstaller* yang selanjutnya akan dilakukan pengujian. Menu *uninstaller* ini berfungsi untuk melakukan penghapusan sejumlah program yang ada di komputer jika sudah tidak diperlukan kembali. Fungsi quick *uninstaller* yang ada di aplikasi revo ini memiliki kelebihan dibandingkan dengan fungsi sejenis yang ada di bawaan sebuah sistem operasi yang berada di sebuah komputer, yaitu memiliki kemampuan penghapusan yang jauh lebih bersih dan cepat dalam pemrosesan, sehingga tidak menyisakan *file-file* temporary ataupun *file* sampah akibat proses penggunaan aplikasi di komputer tersebut. Namun daripada itu tetap saja diperlukan pengujian terhadap fungsi ini agar lebih memastikan kebenaran dari fungsi tersebut.

E-ISSN: 2548-3587 ; 101 – 110

Selanjutnya akan dilakukan pengujian sesuai tahapan yang ada di *cause effect testing*, dimulai menentukan kondisi *input dan output*, menentukan uji kasus dan membuat tabel uji fungsi.



Sumber: Penelitian (2018)

Gambar 4. Fungsi Quick *Uninstaller*

a. Kondisi *Input dan output*

Tahap pertama yang dilakukan yaitu menentukan kondisi *input dan output*. Dalam tabel 6 dapat dilihat bahwa kondisi input ada satu yaitu quick yang menandakan bahwa inputan pencarian bertipe boolean. Kondisi output ada dua yaitu kondisi dimana tampilan untuk melanjutkan ke unistall program dan batal *uninstaller* .

Tabel 6. Kondisi *Input dan output*

Input State:	Output State:
1. Quick = boolean	A. Tampil lanjutkan ke *uninstaller* program B. Batal *uninstaller*

Sumber: Penelitian (2018)

b. Pengujian Kasus

Tahap kedua yaitu menentukan uji kasus yang akan digunakan untuk menguji program. Yang nantinya akan diuji berdasarkan kondisi *input dan output* yang telah ditentukan diawal. Kasus yang dijadikan contoh ada dua diantaranya yaitu 1) Tes 1 : Klik Continue Quick *Uninstaller,* 2) Tes 2 : Klik Cancel

c. Tabel Uji

Langkah terkahir yaitu mebuat tabel uji yang akan menerapkan uji kasus berdasarkan kondisi *input dan output* ditahap sebelumnya. Berikut ini tabel 7 yang menjelaskan tabel uji dari fungsi quick *uninstaller* .

Tabel 7. Tabel Uji Fungsi Quick *Uninstaller*

	Tes 1	**Tes 2**
1	1	1
A	1	0
B	0	1

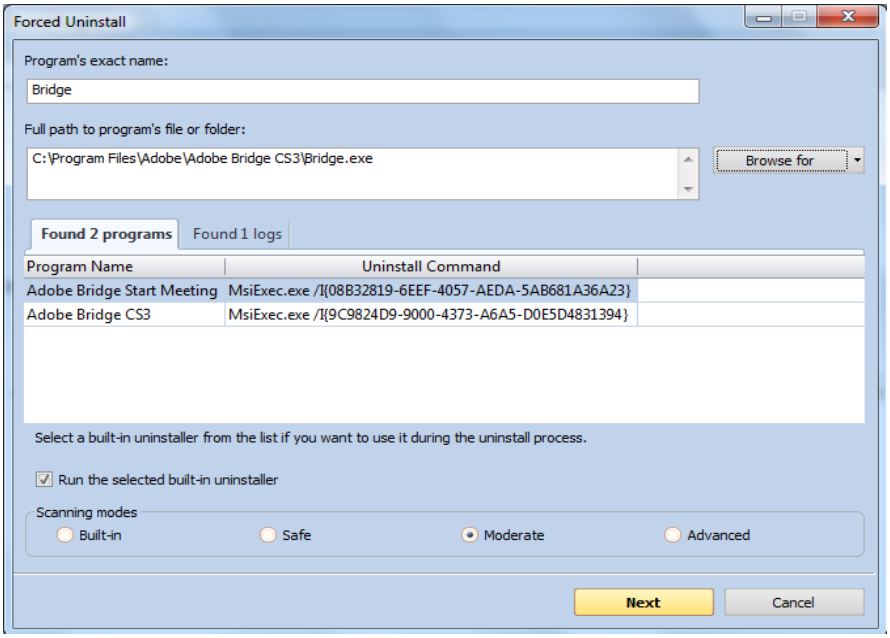
Sumber: Penelitian (2018)

3.4. Force *Uninstaller*

Fungsi keempat yang akan diuji yaitu fungsi force *uninstaller* . Pada gambar 5 terlihat tampilan dari menu *uninstaller* yang selanjutnya akan dilakukan pengujian. Menu *uninstaller* ini berfungsi untuk melakukan penghapusan sejumlah program yang ada di komputer jika sudah tidak diperlukan kembali.

Fungsi force *uninstaller* yang ada di aplikasi revo ini memiliki kelebihan dibandingkan dengan fungsi sejenis yang ada di bawaan sebuah sistem operasi yang berada di sebuah komputer, yaitu memiliki kemampuan penghapusan yang jauh lebih bersih dan cepat dalam pemrosesan, sehingga tidak menyisakan *file-file* temporary ataupun *file* sampah akibat proses penggunaan aplikasi di komputer tersebut. Namun daripada itu tetap saja diperlukan pengujian terhadap fungsi ini agar lebih memastikan kebenaran dari fungsi tersebut.

Selanjutnya akan dilakukan pengujian sesuai tahapan yang ada di *cause effect testing*, dimulai menentukan kondisi *input dan output*, menentukan uji kasus dan membuat tabel uji fungsi.



Sumber : Penelitian(2018)

Gambar 5. Fungsi Force *Uninstaller*

a. Kondisi *Input dan output*

Tahap pertama yang dilakukan yaitu menentukan kondisi *input dan output*. Dalam tabel 8 dapat dilihat bahwa kondisi input ada satu yaitu force yang menandakan bahwa inputan pencarian bertipe boolean. Kondisi output ada dua yaitu kondisi dimana tampilan untuk melanjutkan ke unistall program dan batal *uninstaller* .

Tabel 8. Kondisi *Input dan output*

Input State:	Output State:
1. Force = boolean	A. Tampil lanjutkan ke *uninstaller* program B. Batal *uninstaller*

Sumber: Penelitian (2018)

b. Pengujian Kasus

Tahap kedua yaitu menentukan uji kasus yang akan digunakan untuk menguji program. Yang nantinya akan diuji berdasarkan kondisi *input dan output* yang telah ditentukan diawal. Kasus yang dijadikan contoh ada dua diantaranya yaitu 1). Tes 1: Klik Continue Force *Uninstaller,* 2) Tes 2: Klik Cancel

E-ISSN: 2548-3587 ; 101 – 110

c. Tabel Uji

Langkah terkahir yaitu mebuat tabel uji yang akan menerapkan uji kasus berdasarkan kondisi *input dan output* ditahap sebelumnya. Berikut ini tabel 9 yang menjelaskan tabel uji dari fungsi force *uninstaller* .

Tabel 9. Tabel Uji Fungsi Force *Uninstaller*

	Tes 1	Tes 2
1	1	1
A	1	0
B	0	1

Sumber: Penelitian (2018)

3.5. Junk File Cleaner

Fungsi kelima yang akan diuji yaitu fungsi *junk file cleaner*. Pada gambar 6 terlihat tampilan dari menu *uninstaller* yang selanjutnya akan dilakukan pengujian. Menu *uninstaller* ini berfungsi untuk melakukan penghapusan sejumlah program yang ada di komputer jika sudah tidak diperlukan kembali.

Fungsi *junk file* cleaner yang ada di aplikasi revo ini memiliki kelebihan dibandingkan dengan fungsi sejenis yang ada di bawaan sebuah sistem operasi yang berada di sebuah komputer, yaitu memiliki kemampuan penghapusan *file* sampah yang jauh lebih bersih, detail dan cepat dalam pemrosesan, sehingga tidak menyisakan *file-file* temporary ataupun *file* sampah akibat proses penggunaan aplikasi di komputer tersebut. Namun daripada itu tetap saja diperlukan pengujian terhadap fungsi ini agar lebih memastikan kebenaran dari fungsi tersebut.

Selanjutnya akan dilakukan pengujian sesuai tahapan yang ada di *cause effect testing*, dimulai menentukan kondisi *input dan output*, menentukan uji kasus dan membuat tabel uji fungsi.

a. Kondisi *Input dan output*

Tahap pertama yang dilakukan yaitu menentukan kondisi *input dan output*. Dalam tabel 10 dapat dilihat bahwa kondisi input ada satu yaitu *junk file* cleaner yang menandakan bahwa inputan pencarian bertipe boolean. Kondisi output ada dua yaitu kondisi dimana muncul *file junk* atau tidak.

Tabel 10. Kondisi *Input dan output*

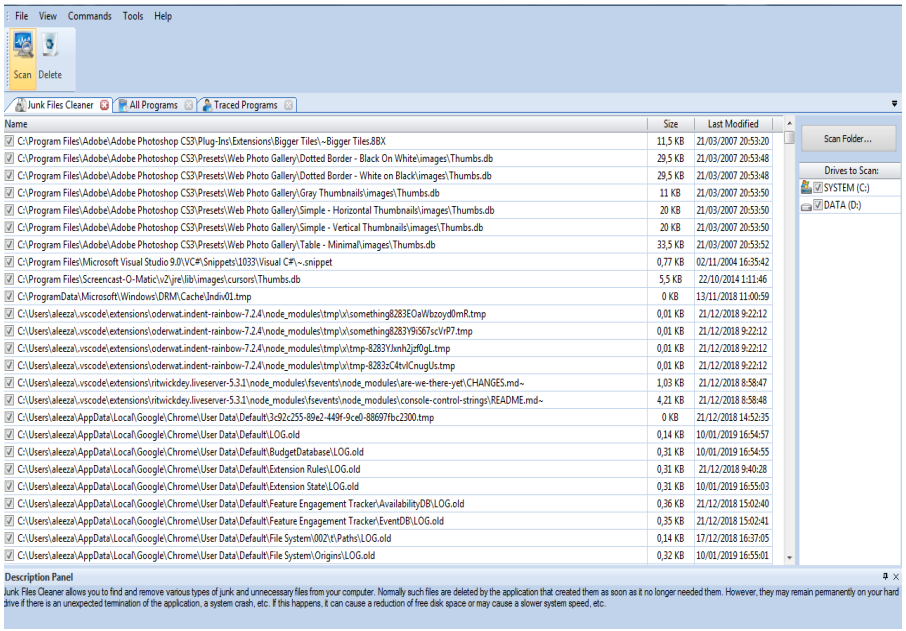
Input State:	Output State:
1. *Junk File* = boolean	A. Tampil hasil scan B. Tidak tampil hasil scan

Sumber: Penelitian (2018)

b. Pengujian Kasus

Tahap kedua yaitu menentukan uji kasus yang akan digunakan untuk menguji program. Yang nantinya akan diuji berdasarkan kondisi *input dan output* yang telah ditentukan diawal. Kasus yang dijadikan contoh ada dua diantaranya yaitu Tes 1 : Klik *scan*.

Terlihat pada gambar 6, fitur fungsi *junk file* cleaner yang ada pada aplikasi revo untuk dilakukan pengujian *black box* terhadap fitur tersebut.



Sumber: Penelitian (2018)

Gambar 6. Fungsi *Junk File* Cleaner

c. Tabel Uji

Langkah terkahir yaitu mebuat tabel uji yang akan menerapkan uji kasus berdasarkan kondisi *input dan output* ditahap sebelumnya. Berikut ini tabel 11 yang menjelaskan tabel uji dari fungsi *search*.

Tabel 11. Tabel Uji Fungsi *Junk File*

	Tes 1
1	1
A	1
B	0

Sumber: Penelitian (2018)

4. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan menggunakan tahapan-tahapan yang ada pada metode *cause-effect relationship testing*, dapat terlihat bahwa aplikasi atau program revo *uninstaller* yang menjadi contoh aplikasi yang diujikan memiliki fungsi-fungsi yang sudah berjalan dengan baik sesuai dengan yang diharapkan. Ini pun membuktikan bahwa metode cause-effect dapat digunakan dengan cukup mudah dalam melakukan pengujian aplikasi. Untuk penelitian selanjutnya disarankan dapat melakukan pengujian terhadap aplikasi lain menggunakan metode yang sama, atau dapat pula melakukan pengujian dengan metode black box *testing* yang lainnya agar dapat terlihat perbedaannya sehingga dapat dibandingkan metode mana yang memiliki tingkat kemudahan yang tinggi dalam melakukan pengujian sebuah aplikasi.

Referensi

Babbar H. 2017. Software *Testing* : Techniques. International Journal of Re*search* in Computer Applications and Robotics. 5 (3): 44–53.

Khanna E. 2017. On the applicability of Artificial Intelligence in Black Box *Testing*. International Journal On Computer Science And Engineering. 9 (5): 165–169.

E-ISSN: 2548-3587 ; 101 – 110

Mustaqbal MS, Firdaus RF, Rahmadi H. 2015. Pengujian Aplikasi Menggunakan Black Box *Testing* Boundary Value Analysis (Studi Kasus Aplikasi Prediksi Kelulusan SNMPTN). Jurnal Ilmiah Teknologi Informasi Terapan. 1 (3): 31–36.

Noeticforce. 2017. Black Box *Testing*: Introduction, Example and Techniques.

Salamah U, Khasanah FN. 2017. Pengujian Sistem Informasi Penjualan Undangan Pernikahan Online Berbasis Web Menggunakan Black Box *Testing*. Information Management For Educators And Professionals. 2 (1): 35–44.