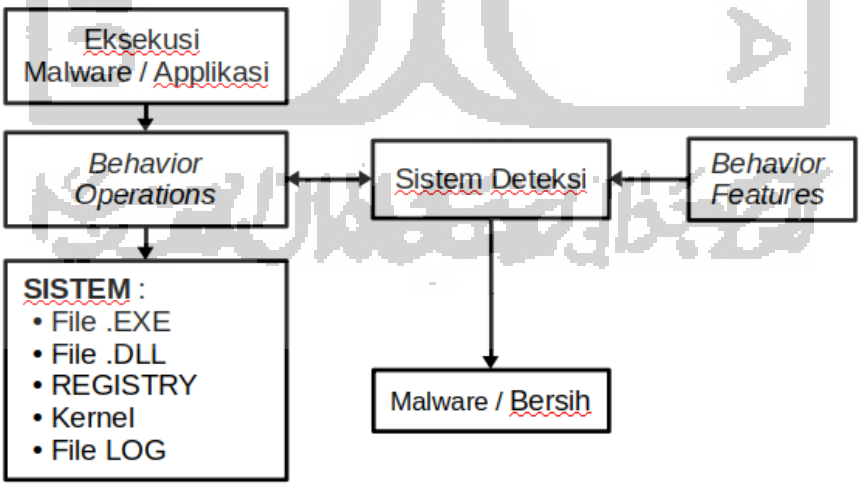
BAB 4
Hasil dan Pembahasan

Bab ini membahas tentang hasil dan pembahasannya untuk menjawab poin-poin pada rumusan masalah, beberapa tahapan yang telah disebutkan pada bab sebelumnya yaitu tahapan analisis yang dilakukan terhadap malware dan aplikasi pengolah data dan kedua yaitu, tahapan implementasi sistem deteksi malware menggunakan pemrograman python serta tahapan berikutnya adalah testing guna mendapatkan hasil, semua tahapan itu akan dijelaskan pada bab ini.

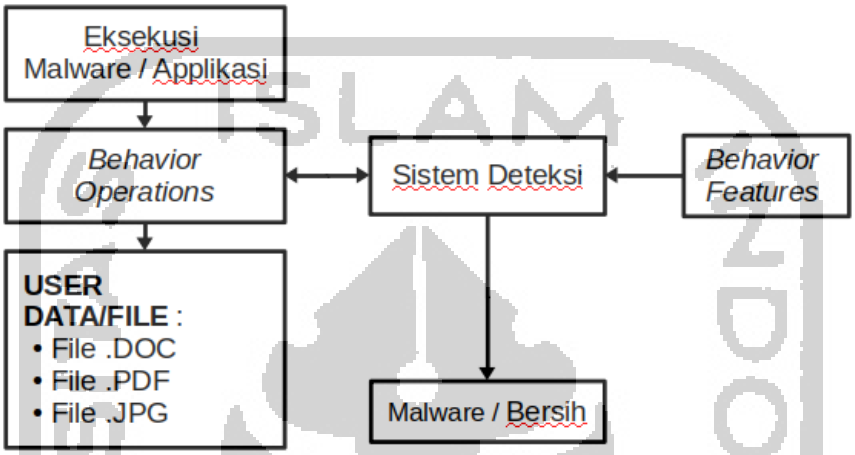
4.1 Penerapan Metode *Behavior Approach*

Deteksi malware menggunakan metode *behavior approach* adalah metode deteksi malware berdasarkan perilaku sebuah appalikasi yang diketahui setelah melalui analisis, untuk dapat mengetahui perilaku sebuah aplikasi secara tepat diperlukan analisis langsung terhadap aplikasi tersebut yaitu dinamik analisis, dengan demikian tahapan pertama dalam penerapan metode *behavior approach* adalah dinamik analisis, untuk dapat memehami perbedaan antara penerapan metode *behavior* pada penelitian yang sudah ada dan penerapannya pada penelitian ini, maka penjelasan dibawah merupakan gambaran umum secara ringkas bagaimana serta perbedaan metode *behavior* diterapkan :



Gambar 4.1 Target *behaviour operations* pada penelitian yang sudah ada

Gambar 4.1 diatas merupakan gambaran tentang penerapan metode *behavior* yang sudah ada, pada gambar diatas sistem merupakan target dari *behavior operation*, sedangkan penerapan metode *behavior approach* pada penelitian ini, sistem bukan menjadi target operation dari *behavior operations,* berikut gambaran penelitian yang akan dilakukan :

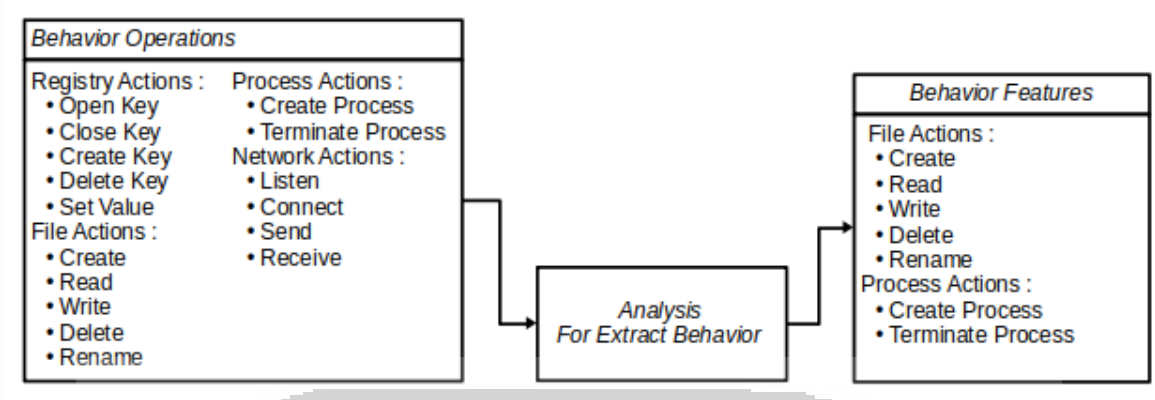


Gambar 4.2 Target *behaviour operations* pada penelitian yang diusulkan

Pada gambar 4.2 diatas penggunaan metode *behavior* yang berbeda adalah pada target *behavior operation*, dimana pada gambar 4.1 menunjukkan target *behavior operations* pada gambar tersebut adalah sistem dengan file dan configurasi yang ada pada sistem tersebut, sedangkan pada gambar 4.2 menunjukkan target *behavior operations* adalah user data atau user file.

4.1.1 *Behavior Operations* dan *Behavior Features*

Behavior operations merupakan operasi atau peoses dasar yang menyebabkan adanya perubahan pada status sistem oleh sebuah perangkat lunak, seperti membuat file, mengubah registry, membuat proses, dan sebagainya, sehingga semua operasi yang dilakukan oleh sebuah aplikasi disebut sebagai *behavior operstions*. Sedangkan *behavior features* merupakan istilah yang digunakan pada penelitian ini untuk menggambarkan *behavior operations* yang telah diekstrak dari sebuah aplikasi atau malware melalui proses analisis, *features* yang diekstrak merupakan *behavior* yang dianggap dapat digunakan untuk mengenali aplikasi atau malware tersebut. Gambar dibawah merupakan ilustrasi proses ekstrak dari *behavior operation* yang melalui proses analisis untuk mendapatkan *behavior features*:



Gambar 4.3 peoses untuk ekstrak behavior

Proses pada gambar 4.3 diatas merupakan proses ekstrak *behavior operations* melalui proses analisis untuk mendapatkan *behavior features, features* yang telah diekstrak merupakan *behavior* yang merepresentasikan aplikasi ataupun malware yang dianalisis.

4.1.2 User Sebagai Target *Behavior operations*

Target *behavior* merupakan target dari sebuah operasi *behavior* yang terjadi pada sebuah aplikasi, pada pnelitian ini target behavior operation merupakan user data yang dibatasi dalam beberapa type data yang umum dimiliki oleh user komputer. Hal yang mendasari ini adalah karena malware yang ramai dibahas saat ini adalah malware ransomware, yang mana salah satu tipe dari malware tersebut adalah locky ransomware, malware ini merupakan malware yang dirancang untuk mencegah akses terhadap data yang telah dikunci sampai pemilik data melakukan pembayaran kepada penyerang atau yang membuat malware tersebut.

4.2 Komputer dan Sistem yang digunakan

Komputer dan sistem yang digunakan untuk melakukan pernelitian dan untuk melakukan analisis dan merancang sistem deteksis serta testing adalah sebuah komputer merk HP *all in one* 23 *inch,* berikut detail spesifikasinya :

Tabel 4. 1 Spesifikasi komputer yang digunakan

Merk /Seri	HP/Pavilion 23-g135x All-in-One Desktop
Resolusi Layar	1920 x 1080
Ukuran Layar	23 Inch
Tipe Layar	Widescreen WLED Backlit

Tabel 4. 1 Spesifikasi komputer yang digunakan (Lanjutan)

CPU	Intel Core i5 4590T, 3.0 GHz
Memori/RAM	8 GB
Harddisk	1 TB Sata
Koneksi	Bluetooth 4.0, Wireless LAN 802.11b/g/n, LAN 1000BASE-T
Port USB	2 x Port USB 3.0, 4 x Port USB 2.0
Webcam	Ya Webcam Terintegrasi
Berat	7.20 kg.
OS	Linux Ubuntu Mate 18.04 (Windows 7 on VM)

4.3 Analisis

Langkah pertama yang dilakukan sebelum masuk pada tahapan analisis adalah mempersiapkan semua hal-hal yang dibutuhkan meliputi sistem, *tool*s dan *environment* yang mendukung proses analisis, persiapan ini dapat dijelaskan dalam tahapan-tahapan proses dibawah.

4.3.1 Malware yang digunakan

Malware yang digunakan sebagai sampel adalah malware locky ransomware dengan beberapa variannya, sampel malware menggunakan sampel malware yang terdapat pada situs komunitas *open source* yaitu *github.com,* dengan alamat tautan https://github.com/ytisf/theZoo.

4.3.2 Sistem dan *Tools* yang Digunakan

Sistem operasi yang digunakan adalah system operasi Ubuntu Linux Mate sebagai host dan Windows 7 yang berjalan pada mesin virtual, adapun tool-tool yang digunakan untuk melakukan analisis malware adalah cuckoo sandoboxs, *Proces Explorer, Process Hacker, Process Monitor* dan modul python *WinAppDbg*. Tool-tool ini digunakan untuk mendapatkan *behavior features,* hal ini nantinya digunakan sebagai dasar untuk mengimplementasikan system deteksi malware.

4.3.3 Analisis Dinamis

Dinamik analisis merupakan tahapan melakukan analisis pada malware dengan cara menjalankan malware secara langsung pada sebuah mesin virtual *virtualbox,* hal ini dilakukan untuk mengamati perilaku malware guna mendapatkan perilaku malware yang dapat dijadikan dasar untuk mengimplementasikan sistem deteksi malware. Berikut

merupakan proses analisis menggunakan tool-tool tersebut beserta tahapan dan penjelasannya :

a. Mesin Virtual

Mesin virtual yang digunakan sebagai tempat untuk menjalankan malware adalah virtualbox versi 5.2.10 dari Oracle yang berjalan diatas sistem operasi linux Ubuntu Mate 18.04, dan sistem operasi yang berjalan pada mesin virtual adalah sistem operasi windows 7. Agar mesin virtual aman digunakan sebagai *environment* untuk menguji malware maka mesin virtual sebagai *environment* harus disetting sedemikian rupa agar tidak menimbulkan hal yang tidak diinginkan seperti malware yang dapat menyerang *host* melalui celah konfigurasi network yang lemah, *sharing* file dan folder ataupun kelemahan konfigurasi lainnya, selanjutnya adalah melakukan *snapshot*, yaitu menyimpan semua kondisi atau keadaan sebuah mesin virtual untuk dapat digunakan diwaktu yang akan dating, *snapshot* dilakukan pada sistem operasi yang terinstall pada mesin virtual, hal ini selain karena merupakan sebuah standar dalam melakukan analisis malware, *snapshot* juga merupakan salah satu standar yang diperlukan untuk melakukan analisis menggunakan tool *cuckoo sandbox*.

b. *Process Explorer, Process Hacker, Process Monitor*

Process analisis menggunakan *process explorer, process hacker* dan *process monitor* adalah tahapan pertama dari analisis yang dilakukan, analisis menggunakan ketiga *tools* ini dilakukan untuk melihat serta mengumpulkan informasi aktivitas prosess yang berjalan pada sistem operasi windows 7. Tool utama yang digunakan disini adalah *process explorer*, setelah *process explorer* dijalankan, selanjutnya malware *locky ransomware* dijalankan untuk dipantau aktivitasnya, fitur-fitur yang ada di *process explorer* seperti *Tree-view, check Virus Total* memudahkan untuk mendapatkan informasi seputar *child process* dan *parent process* serta aktivitas malware mengakses data di komputer. Tool *process hacker* digunakan untuk melihat aktivitas malware mengakses jaringan, sedangkan tool *process monitor* digunakan untuk melihat *real-time file system actifity, registry actifity* dan *process/thread actifity*.

c. *Cuckoo Sandbox*

Tahapan analisis menggunakan *Cuckoo sandbox* adalah proses untuk mendapatkan informasi dari sebuah file yang di-eksekusi pada sebuah mesih virtual virtualbox. Proses analisis menggunakan cuckoo diawali dengan meng-install *cuckoo sandbox* dilanjutkan dengan *setting* konfigurasi sampai *cuckoo* siap digunakan, selanjutnya dijelaskan tahapan

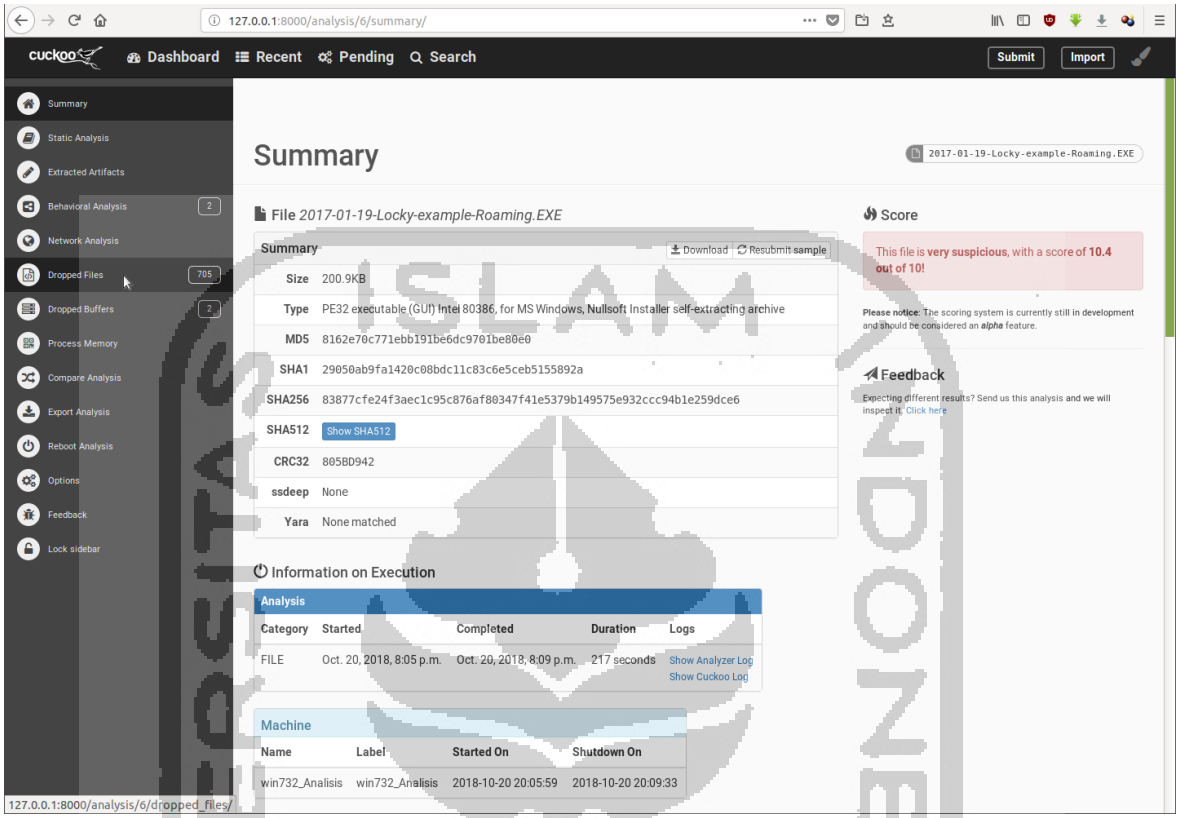
dari menjalankan *cuckoo* sampai mendapatkan informasi detail tentang malware yang di-*submit* ke mesin virtual, berikut tahapannya :

1. Menjalankan cuckoo sundbox melalui jendela terminal pada sistem operasi linux dengan perintah ***"cuckoo"***, berikut schreenshot ketika perintah *cuckoo* dijalankan di terminal :

Gambar 4.4 Perintah *cuckoo* untuk menjalankan *cuckoo sandbox*

2. Menjalankan *cuckoo web interface* dengan perintah "***cuckoo web runserver***" yang bertujuan agar informasi hasil analisis dapat diakses melalui browser pada mesin lokal atau *localhost*.
3. Proses selanjutnya melakukan *submit* file malware yang telah disiapkan dengan perintah "***cuckoo submit 2017-01-19-Locky-example-Roaming.EXE***", proses submit akan memakan waktu karena proses cuckoo meng-*upload* malware ke mesin virtual (windows 7) untuk dijalankan yang selanjutnya akan dianalisis. Proses submit juga dapat dilakukan menggunakan *web interface* yang disediakan oleh *cuckoo*, hal ini dapat dilakukan dengan membuka browser untuk mengakses alamat *http://127.0.0.1:*8000 selanjutnya dapat memilih tombol submit yang disediakan pada halaman tersebut.
4. *Cuckoo* akan menapilkan pesan ketika proses analisis telah selesai, pesan pemberitahuan akan tampil pada terminal ketika prose berhasil ataupun proses gagal.
5. Hasil yang dari analisis yang dilakukan oleh cuckoo adalah berupa laporan dalam bebberapa format yang disediakan oleh *cuckoo sandbox* seperti dalam format text atau txt, dalam format html dan format lain seperti json file. Hasil analisis dapat dilihat menggunakan browser dengan mengakses alamat *http://127.0.0.1:8000*. Berikut

screenshot hasil analisis malware locky dengan nama file "*2017-01-19-Locky-example-Roaming.EXE*" :



Gambar 4.5 Screenshot hasil analisis *cuckoo* pada malware

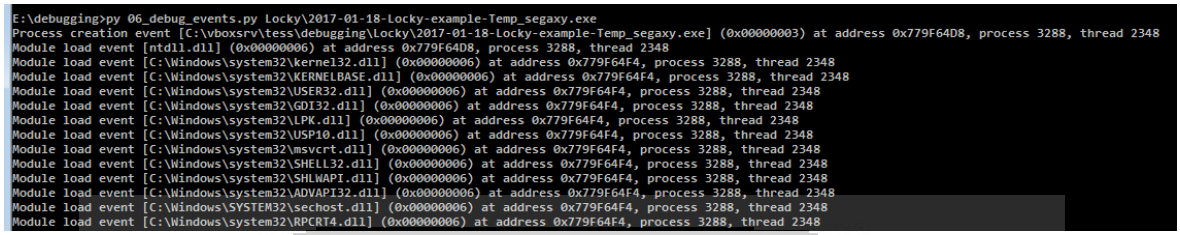
Hasil analisis dari cuckoo pada gambar 4.2 diatas menjelaskan secara detail dari file malware yang dianalisis, menampilkan infomasi standar seperti nilai hash dari beberapa tipe algoritma seperti md5 sampai dengan crc32, waktu lamanya aplikasi dijalankan, berapa jumlah file yang di-drop, serta memberikan score pada malware, nilai score yang tinggi mengindikasikan file yang dianalisis adalah malware.

d. Python *WinAppDbg*

Python *winappdbg* merupakan modul python yang dibangun untuk developer yang bertujuan agar developer dapat dengan cepat membuat scripts di python khusus dilingkungan system operasi windows. Terdapat banyak fungsi Win32 API *call* yang berhubungan dengan *debugging* pada modul ini, selain sangat powerful pada lapisan manipulasi *threads, libraries, process, trace execution, hook API calls, handle actions*, dan *attach process to debugger*.

Penggunaan modul *winappdbg* pada tahapan ini adalah untuk melakukan *debugging* langsung pada malware, hal ini bertujuan untuk mendapatkan informasi *libraries* yang

diload, actions dan *threads* serta informasi lainnya, berikut hasil screenshot process debugging malware :



Gambar 4.6 Proses debugging pada malware di cmd

Proses debugging pada gambar 4.3 diatas merupakan proses debugging actions, proses ini bertujuan untuk melihat lebih detail event-event saat file aplikasi dijalankan menggunakan script python debugging, informasi yang didapat adalah *event load* dan *event unload* module ke memori, dan *event process creation*, serta *event process terminated*.

4.3.4 *Extract Behavior Features*

Tahap *extract behavior features* ini merupakan proses menyimpulkan atas semua informasi yang didapat dari tahapan analisis, selanjutnya informasi ini akan dijadikan sebagai data dasar untuk membangun sistem deteksi malware. Fokus *extract behavior features* adalah mengambil perilaku malware berdasarkan pada operasi file yaitu *create, read, write, rename, delete, open* dan operasi proses yaitu *create process, terminate process.*

4.4 Implementasi Sistem Deteksi Malware

Tahapan implementasi sistem deteksi adalah tahapan untuk meng-implementasikan sistem deteksi malware berdasarkan perilaku aplikasi terhadap data melalui pendekatan behavior menggunakan bahasa pemrograman python. Tahapan ini akan menjelaskan secara detail tahapan implementasi rancangan kedalam bahasa pemrograman python. Hal yang pertama dilakukan adalah mempersiapkan *software* yang diperlukan terdiri dari python 2.7, modul-modul python, dan editor.

4.4.1 Instalasi Python 2.7

Python yang digunakan adalah python 2.7, *software* python 2.7 dapat diunduh langsung di situs resminya yaitu https://www.python.org. file instalasi yang berbasis windows tersedia dalam beberapa format yang dapat didownload, file dapat didownload pada alamat tautan https://www.python.org/downloads/windows/. Beberapa modul python juga dibutuhkan

untuk diinstall guna mendukung proses implementasi, diatara modul yang perlu diinstall adalah modul *psutil*, *pywin32* dan modul *winappdbg.*

4.4.2 Implementasi kedalam Script Python

Implementasi sistem deteksi malware kedalam script python dilakukan dengan mempersiapkan beberapa file script meliputi *main script* atau script utama dan script pendukung, untuk memperjelas fungsi dari setiap script maka dibawah akan dijelaskan fungsi dan fitur dari setiap script serta keterkaitan antara masing-masing script antara script utama dan script pendukung.

a. File Script Pendukung

Persiapan untuk membuat sistem deteksi malware diawali dengan mempersiapkan informasi yang dikumpulkan dari tahapan sebelumnya yaitu tahapan analisis, informasi ini dijadikan sebagai data yang akan diimplementasikan kedalam script dalam bentuk pendeklarasian variabel-variabel dalam file script python. Script pendukung terdiri dari script yang akan menampung variable-variabel dan script yang akan menampung fungsi-fungsi yang disediakan untuk digunakan oleh script utama. Terdapat beberapa file script pendukung yang perlu untuk dijelaskan apa yang terdapat didalam script dan fungsinya terhadap file script utama, berikut masing-masing file script tersebut :

1. File *myvar.py,* merupakan file yang dibuat dan digunakan untuk mendeklarasikan dan menampung variabel-variabel yang diperlukan pada script utama, beberapa variabel perlu dijelaskan fungsi dan kegunaannya seperti variabel filter file, terdapat variabel lainnya dalam file *myvar.py* namun penjelasan hanya untuk memberi gambaran file pendukung dan apa yang terdapat didalam file tersebut.
2. File *mymodul.py* merupakan file yang dibuat untuk menampung fungsi-fungsi yang diperlukan pada script utama, diataranya fungsi manipulasi proses, fungsi monitoring file, fungsi file *open* dan fungsi lainnya. Fungsi-fungsi ini merupakan beberapa fungsi utama yang terdapat pada file *mymodul.py* yang berguna untuk mendukung file script utama.

b. File script utama

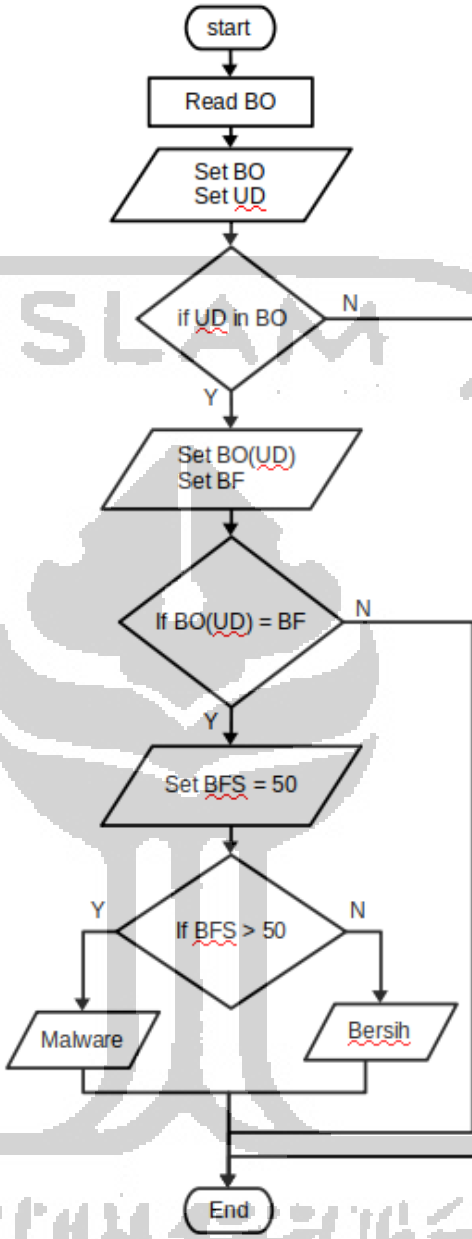
File script utama adalah file script python yang akan dijalankan untuk melakukan proses deteksi malware menggunakan *command line* atau cmd.exe, nama filenya adalah *malde.py*. Tahapan proses yang terdapat dalam script utama meliputi tiga proses utama yaitu *monitoring*, komparasi, dan *decision*. Berikut merupakan penjelasan alur tahapa proses yang ada dalam script tersebut kedalam algoritma :

1. Start, mulai menjalankan program

2. *Read behavior operations,* proses membaca *behavior operations*
3. *Set BO = behavior operation,* buat variabel BO untuk menampung *behavior operations*
4. *Set variable UD = user data type,* buat variabel UD untuk menampung tipe data yang akan difilter
5. *If UD in BO = false,* proses filter memeriksa jika *behavior* tidak mengandung data user maka proses langsung ke *end*
6. *If UD in BO = true,* proses filter memeriksa jika *behavior* mengandung data user maka akan dilanjutkan keproses selanjutnya
7. *Set BO(UD) = BO,* buat variable BO(UD) untuk menampung *behavior* yang telah difilter
8. *Set BF = Behavior Features,* langkah untuk menentukan *behavior features* yang akan diperiksa
9. *If BF in BO(UD) = false,* memeriksa jika *behavior* yang sudah difilter tidak sama dengan *behavior features,* maka proses langsung ke *end*
10. *If BF in BO(UD) = true,* memeriksa jika *behavior* yang sudah difilter sama dengan *behavior features,* maka proses selanjutnya
11. *Set normal standard of BF, BFS = 50,* menentukan standar normal dari jumlah *behavior features* yaitu BFS = 50
12. *If BFS > 50 = ture,* memeriksa jika jumlah BFS melebihi standar maka tampilkan laporan bahwa malware terdeteksi
13. *If BFS > 50 = false,* memeriksa jika jumlah BFS tidak melebihi standar maka tampilkan laporan data saja.
14. *End,* proses selesai.

Algoritma diatas merupakan algoritma dari prose utama yaitu tahapan langkah-langkah untuk proses deteksi malware pada script utama, beberapa singkatan yang digunakan untuk merepresentasikan variable, input, proses dan, output yang perlu dijelaskan yaitu BO (*Behavior Operations*) merupakan proses mendapatkan semua *behavior operation* dengan cara monitoring dan menyimpannya pada variabel, UD (*User Data*) merupakan jenis data user yang dijadikan sebagai filter untuk BO, BO(UD) (*Behavior Operations Contai User Data*) merupakan BO yang telah difilter sehingga BO(UD) merupakan BO yang hanya mengandung data user, BF (*Behavior Features*) merupakan fitur-fitur perilaku yang ditetapkan berdasarkan hasil analisis, dan BFS (*Behavior Features Standar*) merupakan standar yang ditetapkan berdasarkan hasil

analisis. Gambar 4.7 dibawah menjelaskan tentang alur algoritma diatas, berikut merupakan flowchart dari algoritma diatas :



Gambar 4.7 Flowchart dari script utama sistem deteksi malware.

Flowchart diatas merupakan gambaran tahapan-tahapan proses dari proses monitoring sampai proses membuat keputusan pada script utama dari sistem deteksi malware yang dibuat, secara umum terdapat tiga tahapan utama pada flowchart diatas, dimana masing-masing tahapan mempunyai proses yang saling berkaitan satu dengan yang lainnya diawali dengan proses monitoring, berikutnya komparasi dan terakhir keputusan.

Monitoring merupakan proses sistem deteksi mengawasi setiap *behavior operations.* Dalam proses monitoring terdapat proses filter, yaitu proses untuk mengambil *behavior operation* yang hanya mengandung data user atau tipe file berdasarkan dengan ekstensi yang telah ditetapkan, dan mengabaikan data selainnya. Bagian selanjutnya adalah proses melakukan verifikasai file dari direktori pada *actions*, yaitu proses untuk memastikan hanya *actions* yang mengandung atau dinyatakan sebagai file yang akan diperiksa dan bukan direktori.

Terakhir dari proses filter, yaitu proses memeriksa *events* yang mengandung file berdasarkan ekstensi file yang diambil dari variabel yang telah dideklarasikan pada file *myvar.py*, selanjutnya data yang memiliki kesamaan type ekstensi dengan variabel ditampung kedalam variabel yang disediakan.

Komparasi, merupakan proses untuk membandingkan *behavior* yaitu, atara *behavior* yang diambil dari proses monitoring dengan *behavior features* yang telah ditetapkan. Pada proses komparasi terdapat tahapan proses yaitu, tahapan mengambil data *BO (behavior operations)* dari variabel *string, sdat, gtag, pid* dan *exe* untuk dibandingkan dengan *BF (behavior features)* yang telah ditetapkan, proses membandingkan antara *BO* dan *BF* dilakukan berdasarkan pada banyaknya *actions* antara interval waktu yang ditetapkan, fungsi yang melakukan pemeriksaan setiap interval waktu dari saat *start* sampai dengan lima detik berikutnya, dalam interval waktu tersebut proses melakukan penggabungan data, memilah data berdasarkan *behavior*, menghitung data berdasarkan *behavior*, menghitung total file yang diakses, proses menghitug setiap *actions* dan menyimpannya pada variabel-variabel global, selanjutnya *behavior* dan Jumlah data *behavior* diambil dari variabel global tersebut oleh fungsi lain yang bertugas untuk komparasi, selanjutnya data *behavior* ini dikomparasi dengan *behavior* pada saat *system* komputer berjalan dalam keaadaan normal.

Keputusan, merupakan tahapan akhir dari sistem deteksi yaitu, proses mengambil keputusan berdasarkan input dari tahapan-tahapan proses sebelumnya. Proses putusan dilakukan setelah melakukan komparasi antara *behavior* normal dan *behavior* yang tidak normal, ketika sistem deteksi mendapati kondisi *behavior* yang tidak normal maka sistem deteksi menetapkan proses yang tidak normal tersebut sebagai malware.

4.5 Pengujian Sistem dan Hasil

Tahapan pengujian sistem deteksi adalah tahapan untuk menguji hasil penerapan skema yang bertujuan untuk mendeteksi seberapa jauh sistem dapat bekerja dan juga untuk

mendeteksi kegagalan sistem dalam mendeteksi malware sehingga dapat dilakukan perbaikan. Tahapan ini adalah menguji hasil implementasi dari rancangan sistem deteksi malware berupa script python pada sistem operasi windows 7 yang berjalan diatas *virtual machine* virtualbox versi 5.2.10 dari Oracle.

Pengujian script dilakukan dengan menjalankan script pada *command line* atau cmd.exe, hal ini dilakukan setelah melakukan pemeriksaan kembali daftar perangkat pendukung untuk pengujian script ini yaitu sistem dan lingkungan yang akan dilakukan testing seperti, konfigurasi mesin virtual, upload contoh data-data berupa file dokumen dan file gambar serta file pdf, setelah semuanya siap maka dilakukan pengujian script python untuk sistem deteksi malware tersebtu.

4.5.1 Testing Untuk Aplikasi

Testing untuk appliksi dilakukan terhadap aplikasi yang umumnya digunakan oleh pengguna komputer untuk mengakses atau membuka data mereka seperti file dokumen, file pdf, ataupun file gambar, beberapa aplikasi yang dijadikan sebagai aplikasi untuk dilakukan testing merupakan aplikasi yang biasa digunakan khususnya yang berhubungan dengan ketiga jenis file tersebut, 5 (lima) aplikasi normal seperti aplikasi pengolah kata *MsOffice* yang meliputi *MsWord, MsExcel, dan MsPowerPoint*, dan aplikasi untuk membaca file pdf seperti *Adobe Reader* serta aplikasi untuk membuka gambar seperti *Xn View Image*. Berikut merupakan proses testing pada masing-masing aplikasi yang disertai dengan pembahasan tentang hasil dari aplikasi yang ditesting.

a. Testing untuk *MsOffice Word*

Testing pertama yang dilakukan adalah testing terhadap aplikasi pengolah kata *MsOffice Word,* testing dilakukan dengan cara membuka data user berupa file dokumen MsWord yang berekstensi .docx untuk melihat *behavior* yang terrekam oleh sistem deteksi, berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat aplikasi *MsWord* dijalankan untuk mengakses file dokumen yang dirangkum dalam tabel berikut :

Tabel 4. 2 Data hasil testing *application behavior* MsWord

NO.	Daftar *Behavior*	Data *Behavior* Per Testing			Jumlah Rata-Rata
		Test 1	Test 2	Test 3	
1.	*Total Actions*	58	57	24	46.33
2.	*Open*	0	0	0	0.00
3.	*Created*	20	22	4	15.33

Tabel 4. 2 Data hasil testing *application behavior* MsWord (Lanjutan)

NO.	Daftar *Behavior*	Data *Behavior* Per Testing			Jumlah Rata-Rata
		Test 1	Test 2	Test 3	
4.	*Changed*	30	24	10	21.33
5.	*Deleted*	8	11	10	9.67
6.	*Renamed*	0	0	0	0.00
7.	*Total Files*	11	14	11	12.00
8.	*Files Exist*	10	8	2	6.67
9.	*Files Missing*	1	6	9	5.33
10.	*Signature Know*	1	1	0	0.67
11.	*Signature Unknow*	10	13	11	11.33

Hasil testing diatas merupakan hasil dari tiga kali melakukan testing untuk membuka atau mengakses user data berupa file dokumen berekstensi .docx yang dibuka menggunakan aplikasi *MsWord,* dari hasil yang terdapat pada table 4.2 diatas menunjukkan data *behavior* beragam dari sebelas poin pada tabel. Data pada tabel 4.2 terdiri dari infomasi *actions* dan infomasi file, total *actions* merupakan jumlah total *actions* yang merangkum beberapa *actions* dibawahnya yaitu *open actions, created actions, changed actions, deleted actions,* dan *renamed actions,* dari tiga kali testing total *actions* menunjukkan angka rata-rata 46.33, data testing total *actions* menunjukkan data yang konsisten kecuali pada testing ketiga yang hanya setengah dari dua data testing sebelumnya, dari lima *actions* hanya tiga *actions* yang menunjukkan angka diatas 1 yaitu *created, changed,* dan *deleted* sedangkan dua *actions* lainya bernilai 0. Total file adalah total file yang diakses, data diatas menunjukkan total file yang diakses dari tiga kali testing tidak berbeda kecuali pada testing kedua yaitu 14 file yang diakses. *Files exist* dan *files missing*, merupakan informasi dari pemeriksaan keberadaan file berdasarkan nama file, jika terjadi *event renamed* atau *event deleted* pada file maka file dilaporkan *missing* sebaliknya file dilaporkan *exist,* data pada tabel 4.2 menunjukkan file *missing* lebih besar yaitu 2 *exist* banding 9 *missing* pada testing ketiga. Terakhir adalah *signature known* dan *signature unknown,* merupakan infomasi file *signature* yaitu deretan byte unik yang terdapat pada awal setiap file, hasil perbandingan file *signature* dengan file *signature* yang dimiliki sistem menghasilkan *signature known* jika sama, sebaliknya maka menghasilkan

signature unknown jika tidak ada kesamaan, data pada tabel 4.2 menunjukkan *signature unknown* lebih tinggi karena file yang diperiksa umumnya merupakan file *temporary*.

b. Testing untuk *MsOffice Excel*

Testing kedua dilakukan terhadpa aplikasi pengolah data yaitu *MsOffice Excel*, testing dilakukan dengan membuka documen file berekstensi .xlsx, berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat aplikasi *MsExcel* dijalankan untuk mengakses file dokemen .xlsx, berikut merupakan hasil yang dirangkum dalam tabel :

Tabel 4. 3 Data hasil testing *application behavior* MsExcel

NO.	Daftar *Behavior*	Data *Behavior* Per Testing			Jumlah Rata-Rata
		Test 1	*Test 2*	*Test 3*	
1.	*Total Actions*	37	22	24	27.67
2.	*Open*	0	0	0	0.00
3.	*Created*	10	8	8	8.67
4.	*Changed*	12	10	10	10.67
5.	*Deleted*	6	4	6	5.33
6.	*Renamed*	2	0	0	0.67
7.	*Total Files*	4	3	3	3.33
8.	*Files Exist*	3	3	3	3.00
9.	*Files Missing*	1	0	0	0.33
10.	*Signature Known*	1	0	0	0.33
11.	*Signature Unknown*	3	3	3	3.00

Data pada tabel 4.3 merupakan hasil testing tiga kali untuk aplikasi MsExcel, data menunjukkan pola data yang hampir sama dengan pola data pada hasil testing sebelumnya pada aplikasi MsWord, data rata-rata *actions* menunjukkan penurunan hampir setengah jika dibandingkan data total *actions* pada tabel 4.2 dengan total *actions* pada tabel 4.3 yaitu 46.33 berbanding 27.67, hal yang sama pada total file yang jumlah rata-rata menurun dan berbeda jaun dari 12.00 menjadi 3.33, sedangkan data file *exist* menunjukkan konsistensi pada angka 3 disetiap testing sama halnya dengan data *signature unknown* yang juga menunjukkan angka yang sama, sehingga secara umum data pada tabel 4.2 dan tabel 4.3 tidak menunjukkan berbedaan angka yang cukup jauh.

c. Testing untuk *MsOffice PowerPoint*

Testing aplikasi ketiga adalah testing terhadpat aplikasi *MsOffice PowerPoint,* testing dilakukan dengan membuka dokumen .pptx menggunakan aplikasi *MsOffice PowerPoint,* penjelasan berikut merupakan hasil dari beberapa kali melakukan pengujian sistem deteksi malware saat aplikasi *MsPowerPoint* dijalankan (lihat Lampiran A.1).

Data pada hasil testing tiga kali untuk aplikasi *MsPowerPoint,* data menunjukkan pola data yang juga hampir sama dengan pola data pada hasil testing sebelumnya pada aplikasi *MsWord,* data rata-rata *actions* menunjukkan penurunan jika dibandingkan data total *actions* pada tabel 4.3 yaitu 27.67 dibanding data pada table 4.4 dengan total *actions* 21.33, hal yang sama adalah data pada total file yang menunjukkan jumlah rata-rata yang sama yaitu 3.33, sedangkan data file *exist , signature unknown* tidak menunjukkan perbedaan angka yang jauh, sehingga secara umum data pada tabel 4.3 dan tabel 4.4 tidak menunjukkan berbedaan data yang cukup jauh. Data-data yang dihasilkan dari testing terhadap aplikasi *MsOffice* pada tabel 4.2, dan tabel 4.3 serta data pada *MsPowerPoint* menunjukkan perbedaan data yang tidak terlalu jauh.

d. Testing untuk *Adobe Acrobat Reader*

Testing aplikasi keempat adalah testing terhadpat aplikasi *Adobe Acrobat Reader,* testing dilakukan terhadpa aplikasi pembaca file pdf dengan membuka file berekstensi .pdf, penjelasan berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat aplikasi *Adobe Acrobat Reader* dijalankan (lihat Lampiran A.2).

Data pada hasil testing menunjukkan perbedaan yang jauh dibandinganken dengan data yang ada pada beberapa hasil testing sebelumnya, jika data pada ketiga hasil testing sebelumnya jumlah total *actions* rata-rata diatas angka 21.33 maka data total *actions* yang dihasilkan dari aplikasi *Adobe Acrobat Reader* hanya 1.67 *actions,* data total file hanya 1 file dengan file yang tidak dikenali masing-masing 1 file setiap testing.

e. Testing untuk *Xn View Image*

Testing aplikasi kelima adalah testing terhadpat aplikasi *Xn View Image,* testing dilakukan terhadpa aplikasi pembuka atau pengelola gambar, testing dilakukan dengan membuka file gambar berekstensi .jpg, berikut merupakan penjelasan hasil dari beberapa kali pengujian sistem deteksi malware saat aplikasi *Xn View Image* dijalankan (lihat Lampiran A.3).

Data pada hasil testing menunjukkan perbedaan yang tipis dengan data pada hasil testing *Adobe Acrobat Reader* sebelumnya, data total *actions* yang dihasilkan dari aplikasi *Adobe Acrobat Reader* hanya 1.67 *actions* sedangkan pada aplikasi *Xn View Image* diatas menunjukkan 1.33 total *actions* perbedaan hanya pada angka debelakang koma, data total

file hanya 1 file dengan file yang tidak dikenali masing-masing 1 file setiap testing, kedua tabel 4.5 dan tabel 4.6 menunjukkan data hampir sama disemua itemnya.

4.5.2 Testing Untuk Malware Locky Ransomware

Proses selanjutnya adalah melakukan pengujian sistem deteksi malware dengan menjalankan malware locky ransomware secara langsung setelah sistem deteksi dijalankan untuk monitoring, hasil berupa laporan dari sistem deteksi setelah malware locky ransomware dijalankan dapat dilihat sebagaimana screenshot dibawah :

Tabel 4. 4 Data hasil testing malware Locky Ransomware behaviour

NO.	Daftar *Behavior*	Data *Behavior* Per Testing			Jumlah Rata-Rata
		Test 1	*Test 2*	*Test 3*	
1.	*Total Actions*	784	1769	643	1065.33
2.	*Open*	0	0	0	0.00
3.	*Created*	4	7	0	3.67
4.	*Changed*	437	818	278	511.00
5.	*Deleted*	0	0	0	0.00
6.	*Renamed*	235	573	180	329.33
7.	*Total Files*	185	755	153	364.33
8.	*Files Exist*	135	538	110	261.00
9.	*Files Missing*	50	217	43	103.33
10.	*Signature Known*	0	0	0	0.00
11.	*Signature Unknown*	185	755	153	364.33

Data pada table 4.4 diatas adalah data hasil testing terhadap malware locky ransomware, jika melihat data tersebut menunjukkan perbedaan data yang sangat tinggi jika dibandingkan dengan data-data hasil testing terhadap aplikasi sebelumnya, tingginya perbedaan data tidak hanya pada satu poin melainkan semua poin pada tabel kecuali poin *created actions*, dari poin pertama yaitu total *actions* jika rata-rata data tertinggi pada tabel 4.2 sampai tabel 4.4 hanya mencapai angka 46.66, maka angka rata-rata total *actions* dari hasil testing malware pada tabel 4.4 menunjukkan peningkatan 23 kali lipat dari data tertinggi total *actions* hasil sebelumnya yaitu pada tabel 4.2, peningkatan data lebih dari 10 kali lipat juga terjadi pada poin lain yaitu poin *changed actions*, *renamed actions, files exist, file missing,* dan poin *signature unknow*. Perbedaan data angka rata-rata yang sangat

tinggi antara malware dengan aplikasi pada setiap poinnya dapat dilihat dengan lebih jelas pada penjelasn berikutnya yaitu tabel 4.5.

4.5.3 Hasil Testing Rata-Rata

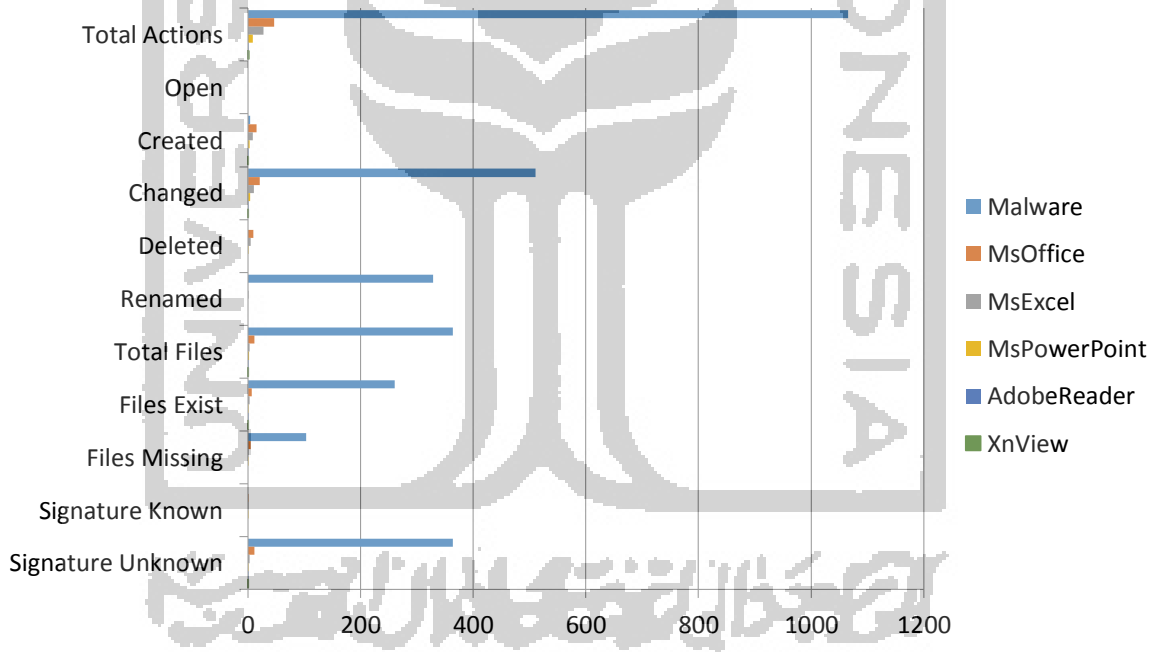
Hasil testing pada 6 aplikasi yang terdiri dari 5 aplikasi normal dan 1 malware menunjukkan hasil yang berbeda, pada 5 aplikasi data hasil testing tertinggi adalah data pada aplikasi MsWord sedangkan data terendah adalah XnView, sedangkan rata-rata hasil testing pada malware menunjukkan perbedaan data yang sangat tinggi jika disbandingkan dengan data pada 5 aplikasi normal sebelumnya terutama pada beberapa *action* yang akan dijelaskan selanjutnya, berikut merupakan hasil rangkuman data dari 6 (enam) aplikasi terdiri dari 5 (lima) aplikasi normal dan 1 (satu) malware yang dirangkum menjadi satu tabel :

Tabel 4. 5 Data rangkuman hasil testing rata-rata aplikasi normal dan malware

NO.	Daftar *Behavior*	Data *Behavior* Rata-Rata					
		MsWord	MsExcel	MsPower Point	Adobe Reader	XnView Image	Locky Malware
1.	*Total Actions*	46.33	27.67	8.67	1.67	1.33	1065.33
2.	*Open*	0.00	0.00	0.00	0.00	0.00	0.00
3.	*Created*	15.33	8.67	2.67	1.00	1.00	3.67
4.	*Changed*	21.33	10.67	4.33	0.67	0.33	511.00
5.	*Deleted*	9.67	5.33	1.67	0.00	0.00	0.00
6.	*Renamed*	0.00	0.67	0.00	0.00	0.00	329.33
7.	*Total Files*	12.00	3.33	2.00	1.00	1.00	364.33
8.	*Files Exist*	6.67	3.00	1.67	1.00	1.00	261.00
9.	*Files Missing*	5.33	0.33	0.33	0.00	0.00	103.33
10.	*Signature Known*	0.67	0.33	0.33	0.00	0.00	0.00
11.	*Signature Unknown*	11.33	3.00	1.67	1.00	1.00	364.33

Laporan data pada tabel 4.5 diatas merupakan data rata-rata hasil pengujian sistem deteksi malware terhadap aktivitas 6 aplikasi terdiri dari 5 aplikasi normal dan 1 malware locky ransomware dalam melakukan akses terhadap data user. Terdapat 11 daftar *action* yang digunakan untuk merepresentasikan perilaku aplikasi yang ditesting untuk mengukur

perilaku normal dan perilaku tidak normal, dan dari hasil testing terdapat beberapa *action* yang mempunyai nilai 0 dari hasi testing 5 aplikasi normal yaitu *open* dan *renamed*, hal ini dikarenakan testing pada aplikasi normal dilakukan dengan mengakses dokumen mengunakan aplikasi file explorer dan tidak membuka file melalui menu pada aplikasi, dan aplikasi tidak melakukan *action renamed* ketika membuka dokumen, sedangkan pada 1 malware *action* yang memiliki nilai 0 adalah *open, deleted* dan *signature known*, nilai pada *open action* bernilai 0 karena file dokumen tidak dibuka secara formal melalui aplikasinya akan tetapi malware mengakses dokumen menggunakan fungsi bawaannya sendiri, selain itu malware juga tidak melakukan penghapusan file dokumen akan tetapi mengubah isi dokumennya sehingga dengan demikian file dokumen tidak dikenali data *signature*-nya hal ini membuat data *signature known* menjadi 0. Berikut merupakan gambaran data dalam diagram yang akan memudahkan untuk melihat perbedaan antar data dari masing-masing aplikasi dan malware :



Gambar 4. 8 Data perbedaan *behaviour* aplikasi normal dan malware

Data pada tabel 4.5 sebelumnya diatas merupakan rangkuman data berupa angka rata-rata dari beberapa tabel sebelumnya yang bertujuan untuk mempermudah melihat perbedaan dari masing-masing tabel yang dihasilkan melalui testing terhadap aplikasi dan malware menggunakan sistem deteksi malware. Hasil rangkuman data rata-rata menunjukkan perbedaan sangat tinggi, data rata-rata tertinggi untuk poin total *actions*

selain malware adalah data dari aplikasi *MsWord* yaitu 46.33 dan terendah adalah data milik aplikasi *XnView Image* yaitu 1.33, sedangkan data total file yang diakses tertinggi selain malware adalah milik aplikasi *MsWord* yaitu 12 file dan yang terendah yaitu 1 file, dengan demikian data rata-rata dari beberapa aplikasi tersebut pada point total *actions* menunjukkan angka antara 1.33 sampai dengan 46.33 *actions* per testing, sedangkan total file yang diakses menunjukkan data rata-rata antara 1 file sampai dengan 12 file yang diakses per testing. Sedangkan data rata-rata yang dihasilkan dari testing terhadap malware menunjukkan peningkatan jumlah angka menjadi beberapa kali lipat, jika melihat pada tabel 4.7 data hasil tiga kali testing malware menunjukkan data terendah untuk point total *actions* adalah berjumlah 643 *actions* sedangkan tertinggi adalah berjumlah 1769 *actions,* adapun pada poin jumlah total file data terendah total file yang diakses adalah berjumlah 153 file dan tertingi adalah sebanyak 755 file yang diakses.

4.6 Analisis Hasil dan Penerapan Metode

4.6.1 Analisis Hasil Testing

Tingginya perbedaan jumlah data *actions,* jumlah file yang diakses, serta perbedaan data lainya yang dihasilkan dari pengujian antara aplikasi malware dengan beberapa aplikasi pengolah data diatas, menunjukkan perilaku aplikasi terhadap data berdasarkan hasil pada tabel-tabel dapat dijadikan sebagai cara untuk deteksi malware dengan menggunakan metode *behavior based detection approach,* tingginya perbedaan data antara malware dan aplikasi tersebut dapat untuk dijadikan sebagai sebuah standar *behavior* yaitu, *behavior* yang normal dan *behavior* yang tidak normal, sehingga dengan menggunakan standar *behavior* tersebut ketika sebuah aplikasi mengakses data user maka sistem apapun yang dibangun untuk deteksi malware dengan standar tersebut menggunakan metode pendekatan *behavior* dapat mendeteksi adanya perilaku yang tidak normal tersebut dan dinyatakan sebagai sebuah serangan malware.

Data hasil testing pada aplikasi dan malware diatas juga menunjukkan persentase yang berbeda pada keduanya yaitu pada aplikasi normal dan pada malware berdasarkan angka persentase *true negative* dan *false negative* untuk semua aplikasi serta angka persentase *true positive* dan *false positive* untuk beberapa kali pengujian terhadap malware, dalam hal ini data menunjukkan angka *true negative* 100% dan *false negative* 0% dalam mendeteksi aplikasi normal tingkat keberhasilan mencapti 100%, sedangkan angka *true positive* 99% dan *false positive* 1% ketika mendeteksi malware, dengan demikian data

menunjukkan angka keberhasilan mendeteksi malware adalah 99% akurat untuk *true positive.*

4.6.2 Analisis Hasil Penerapan Metode *Behavior*

Penggunaan metode *behavior based detection* untuk mendeteksi malware yang secara spesifik menjadikan data user sebagai target dan bukan sebagaimana malware pada umumnya yang menjadikan system sebagai targetnya merupakan yang ditekankan pada penelitian ini, pada tahap penerapannya penelitian ini menggunakan malware yang yang secara spesifik menjadikan data user sebagai target yaitu malware ransomware tipe locky dan, python untuk implementasi sistem deteksi malwarenya.

Penerapan penggunaan metode *behavior based detection* untuk deteksi malware pada penelitian ini menunjukkan keberhasilan dan hal ini sesuai dengan tujuan penelitian yaitu, penerpan metode ini untuk dapat mendeteksi walware berdasarkan perilaku sebuah aplikasi ketika melakukan akses terhadap data milik user. Keberhasilan ini tentu dapat dilihat atau diukur berdasarkan beberap poin yaitu, berhasil menerapkan metode *behavior* untuk deteksi malware yang diimplementasikan menjadi sistem deteksi malware menggunakan pemrograman python, dan sistem deteksi yang dibangun dapat bekerja sangat baik dalam mendeteksi aktifitas malware berdasarkan perilaku aplikasi yang mengakses data user.

4.6.3 Hasil

Pengujian terhadap sistem deteksi malware yang telah dilakukan pada beberapa aplikasi dan malware diatas menunjukkan hasil yang sesuai dari beberapa kali melakukan pengujian, hasil tersebut dapat disimpulkan sebagai indikasi keberhasilan sistem deteksi malware untuk mendeteksi malware menggunakan metode *behavior based detection*, poin yang menjadi indikasi keberhasilan adalah, hasil yang sesuai yaitu berhasil menerapkan metode *behavior based detection* untuk deteksi malware dan system yang dibagun berdasarkan pada metode *behavior* yaitu sistem deteksi malware dapat membedakan sebuah aplikasi dengan malware berdasarkan perilaku aplikasi terhadap data user, dan data hasil yang ditunjukkan pada tabel-tabel diatas menunjukkan poin keberhasil yang dimaksud.