

**Penggunaan Metode *Behavior Based Detection***

**Untuk Deteksi Ransomware Walware Dengan Cara Mengawasi Perilaku Applikasi Pada Data**

M. Ikhwan Nasrullah Elyas

14917214

*Tesis diajukan sebagai syarat untuk meraih gelar Magister Komputer*

*Konsentrasi Forensika Digital*

*Program Studi Magister Teknik Informatika*

*Program Pascasarjana Fakultas Teknologi Industri*

*Universitas Islam Indonesia*

2018

Daftar Isi

Daftar Isi i

Daftar Tabel iv

Daftar Gambar v

BAB 1 Pendahuluan 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 4

1.3 Batasan Maslah 4

1.4 Tujuan Penelitian 4

1.5 Manfaat Penelitian 4

1.6 *Review* Penelitian 5

1.7 Metodologi Penelitian 9

1.8 Sistematika Penulisan 10

BAB 2 Tinjauan Pustaka 12

2.1 Malware 12

2.1.1 Sejarah malware 12

2.1.2 Jenis-Jenis Malware 13

2.1.3 Perkembangan Malware 14

2.2 Metode-Metode Deteksi 17

2.2.1 *Signature Based Detection* 17

2.2.2 *Behavior Based Detection* 17

2.2.3 *Statistic Based Detection* 18

BAB 3 Metodologi Penelitian 19

3.1 Pendahuluan 19

3.2 Tinjauan Pustaka 19

3.3 Analisis 20

3.3.1 Malware dan Applikasi yang dianalisis 20

3.3.2 Sistem dan *Tools* Analisis 20

3.3.3 Analisis dan *Extract Behavior Features* 21

3.4 Perancangan Skema Deteksi 21

3.5 Implementasi dan Testing Skema Deteksi 23

3.5.1 Implementasi 23

3.5.2 Testing sistem 23

3.6 Hasil 23

3.7 Kesimpulan 24

BAB 4 Hasil dan Pembahasan 25

*4.1* Penerapan Metode *Behavior Approach* 25

4.1.1 *Behavior Operations* dan *Behavior Features* 26

4.1.2 User Sebagai Target *Behavior operations* 27

4.2 Komputer dan Sistem yang digunakan 27

4.3 Analisis 28

4.3.1 Malware yang digunakan 28

4.3.2 Sistem dan *Tools* yang Digunakan 28

4.3.3 Dinamik Analisis 28

4.3.4 *Extract Behavior Features* 32

4.4 Implementasi Sistem Deteksi Malware 32

4.4.1 Instalasi Python 2.7 32

4.4.2 Instalasi Modul Python 33

4.4.3 Implementasi kedalam Script Python 33

4.5 Pengujian Sistem dan Hasil 37

4.5.1 Testing Untuk Applikasi 38

4.5.2 Testing Untuk Malware Locky Ransomware 43

4.5.3 Hasil Testing Rata-Rata 44

4.6 Hasil dan Kesimpulan 45

4.6.1 Kesimpulan Hasil Testing 45

4.6.2 Hasil Penerapan Metode *Behavior* 46

4.6.3 Kesimpulan 46

BAB 5 Kesimpulan dan Saran 48

5.1 Kesimpulan 48

5.2 Saran 48

Daftar Pustaka 49

Daftar Tabel

[Tabel 1.1 Kelebihan dan kekurangan signature dan behavior (Souri & Hosseini, 2018) 1](#_Toc530584146)

[Tabel 1.2 Literatur Review Penelitian 6](#_Toc530584147)

[Tabel 4.1 Spesifikasi komputer yang digunakan 27](#_Toc535887780)

[Tabel 4.2 Data hasil testing application behavior *MsWord* 38](#_Toc535887781)

[Tabel 4.3 Data hasil testing application behavior *MsExcel* 40](#_Toc535887782)

[Tabel 4.4 Data hasil testing *application* *behavior MsPowerPoint* 41](#_Toc535887783)

[Tabel 4.5 Data hasil testing application *behaviour* *Adobe Acrobat Reader* 42](#_Toc535887784)

[Tabel 4.6 Data hasil testing application *behaviour Xn View Image* 42](#_Toc535887785)

[Tabel 4.7 Data hasil testing malware *Locky Ransomware* *behaviour* 43](#_Toc535887786)

[Tabel 4.8 Data rangkuman hasil testing rata-rata per applikasi dan malware 44](#_Toc535887787)

Daftar Gambar

Gambar 1.1 Gambaran penelitian yang sudah ada (Liu et al., 2011), (Veeramani & Rai, 2012) dan yang akan diusulkan. 2

Gambar 1.2 Alur Metode Penelitian. 10

Gambar 2.1 Malware baru dalam angka sejak tahun 1984 – 2017 (www.av-test.org, 2017). 15

Gambar 2.2 Total Malware dalam kurun waktu 10 (sepuluh) tahun terakhir (www.av-test.org, 2017) 16

Gambar 3.1 Alur metode penelitian 19

Gambar 3.2 Skema sistem deteksi malware 21

Gambar 3.3 Monitoring proses 22

Gambar 3.4 Proses Komparasi *Behavior* 23

[Gambar 4.1 Target *behaviour operations* pada penelitian yang sudah ada 25](#_Toc530584609)

[Gambar 4.2 Target *behaviour operations* pada penelitian yang diusulkan 26](#_Toc530584610)

[Gambar 4.3 peoses untuk ekstrak behavior 27](#_Toc530584611)

[Gambar 4.4 Perintah *cuckoo* untuk menjalankan *cuckoo sandbox* 30](#_Toc530584612)

[Gambar 4.5 Screenshot hasil analisis *cuckoo* pada malware 31](#_Toc530584613)

[Gambar 4.6 Proses debugging pada malware di cmd 32](#_Toc530584614)

[Gambar 4.7 Flowchart dari script utama sistem deteksi malware 36](#_Toc530584615)

# BAB 1 Pendahuluan

## Latar Belakang

Malware merupakan program yang sengaja dibuat untuk membahayakan dan merugikan sistem operasi atau data pada komputer (Siddiqui, 2008). Langkah untuk deteksi terhadap serangan malware telah dilakukan dengan menerapkan metode-motode terkini yang dirasa mampu untuk deteksi malware masuk dan merusak sistem, salah satu metode tersebut adalah *behavior based detection*.

Pendeteksian malware berbasis *behaviour based detection* adalah metode deteksi malware berdasarkan pengamatan secara terus-menerus terhadap perilaku applikasi untuk menentukan apakah perilaku itu berbahaya atau tidak (Alqurashi & Batarfi, 2016), dalam metode ini perilaku malware dan program biasa dipelajari dan dilakukan pemantauan serta analisis, setelah itu hasil analisis dapat dijadikan acuan bahwa suatu program dapat dikategorikan sebagai malware jika memiliki perilaku yang meyimpang dari perilaku normal program pada umumnya.

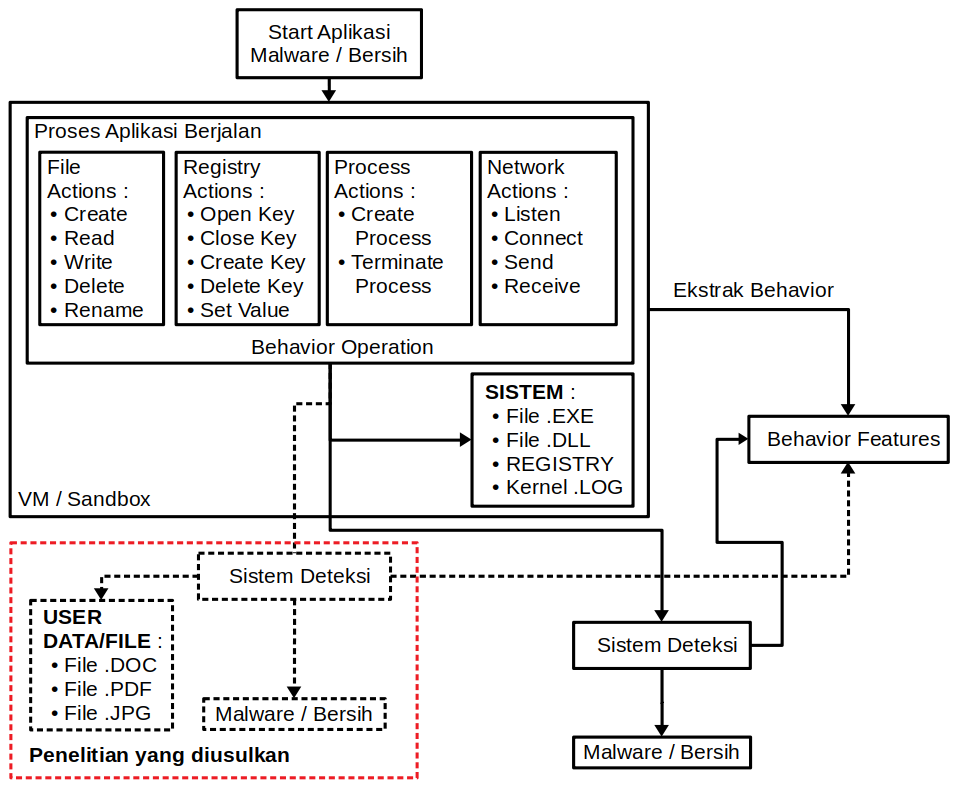
Metode pendeteksian malware yang umum digunakan saat ini adalah *signature based,* metode ini digunakan pada *tools* anti virus saat ini, dalam metode ini terdapat dua teknik yang biasa digunakan yaitu *byte pattern* dan file *checksum* ataunilai *hash.*

Secara umum metode pendeteksian malware dibagi menjadi dua yaitu, *Signature-Based* dan *Behavior-Based* menurut (Deylami, Muniyandi, Ardekani, & Sarrafzadeh, 2016), dan sebagian besar *tools* keamanan khususnya *tool* anti malware saat ini masih menggunakan metode pendeteksian berbasis *signature-based* (Preda, Christodorescu, Jha, & Debray, 2008)*.* Selain kedua metode tersebut terdapat satu metode lainnya yang dapat digunakan untuk mendeteksi malware yaitu *Statistical Based Detection,* hal ini sebagaimana disampaikan oleh (Damodaran, 2015).

Tabel 1.1 Kelebihan dan kekurangan signature dan behavior (Souri & Hosseini, 2018)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | ***Signature Detection Method*** | | ***Behavior Detection Method*** | |
| **Kelebihan** | **Kekurangan** | **Kelebihan** | **Kekurangan** |
|  | Mudah dijalankan | Gagal untuk mendeteksi polimorfik malware | Dapat mendeteksi polimorfik malware | Kompleksitas penyimpanan untuk pola *behavior* |
|  | Cepat indentifikasi | Informasi disimpan dalam database yang sangat besar | Dapat mendeteksi jenis serangan malware yang tak dipahami | Kompleksitas waktu |
|  | Dapat diakses secara luas |  | Dapat mendeteksi ketergantungan aliran-data |  |

Dalam sistem operasi Windows, proses merupakan dasar dari sebuah unit eksekusi dan juga inisiator dari suatu *behavior*. Dalam penjelasannya (Liu, Ren, Liu, & Duan, 2011) menyebutkan bahwa *behavior operations* merupakan operasi dasar yang menyebabkan adanya perubahan pada status sistem oleh sebuah perangkat lunak, seperti membuat file, mengubah registry, membuat proses, dan sebagainya. Selain *behavior operations* hal lain yang menjadi bagian dari proses adalah *target behavior operations,* yaitu target operasi seperti file sistem, registry, port jaringan dan sebagainya, sistem merupakan target operasi pada umumnya. Gambar 1 dibawah menjelaska 4 (empat) *behavior operations* yang menjadi acuan untuk meng-ekstrak *behavior features* yaitu data *behavior* yang digunakan untuk deteksi malware berdasarkan kesamaan *features,* dan juga menjelaskan beberapa penelitian yang sudah ada dan yang akan diusulkan pada penelitan ini.



Gambar 1.1 Gambaran penelitian yang sudah ada (Liu et al., 2011), (Veeramani & Rai, 2012) dan yang akan diusulkan.

Penelitian yang berkembang saat ini lebih fokus pada *behavior operation* dimana sistem sebagai *target behavior operations*, contoh penelitian yang dilakukan oleh (Kheir, 2013), penelitian ini membahas tentang klasifikasi dan deteksi malware berdasarkan perilaku tidak normal pada *header*  *user agent* yang melakukan *reques* ke server pada trafik jaringan*.* Dalam penelitian ini *target behavior operation* adalah system dengan *behavior features* yang diekstrak lebih pada bagian *network action* dari *behavior operations*.Penelitian lainnya yang dilakukan oleh (Berlin, Slater, & Saxe, 2015) adalah deteksi malware *behavior* dengan menggunakan fasilitas bawaan pada sistem operasi Windows yaitu *Windows Audit Logs*, sama halnya dengan penelitian sebelumnya, sistem merupakan *target behavior operations* dan *behavior features* diekstrak dari *behavior operations* melalui analisis menggunakan *virtual sandbox*  yang telah dikonfigurasi sedemikian rupa, sehingga informasi yang ditemukan pada *audit logs* menurut peneliti ini cukup untuk deteksi malware*.*

Namun sayangnya penelitian berbasis *behavior approach* untuk deteksi malwareyang ada saat ini dan dari penelitan-penelitian yang disampaikan diatas tidak membahas secara spesifik tentang data user seperti file .docx, .pdf dan .jpeg sebagain *target behavior operations* padahal, beberapa tahun belakangan malware yang menjadi topik utama dalam dunia keamanan komputer adalah malware yang manargetkan data user sebagai sasaran seperti ransomware. Oleh sebab itu diperlukan sebuah penelitian yang membahas secara spesifik deteksi malware berbasi *behavior approach* dimana perilaku applikasi terhadap data dalam hal ini sebagai *target behavior operations* dapat digunakan sebagai cara deteksi malware.

Dalam penelitian ini *dynamic* analisis digunakan sebagai tahapan pertama dari dua tahapan utama, *dynamic* analisis merupakan proses mengamati aktivitas malware pada sebuah *virtual machine* sebagaimana disampaikan oleh (Yusirwan, Prayudi, & Riadi, 2015), tahapan ini juga untuk meng-ekstrak *behavior features.* Tahapan kedua yaitu merancang skema deteksi berdasarkan hasil pada tahapan pertama. Harapannya penelitian ini nantinya dapat memberikan kontribusi berupa teknik baru dalam bidang keamanan computer khususnya dalam ruang lingkup metode deteksi malware berbasis *behavior approach*.

## Rumusan Masalah

Merujuk kepada latar belakang yang telah dipaparkan sebelumnya, maka dapat diambil rumusan masalah di dalam penelitian ini, yaitu :

1. Bagaimanakah penerapan metode *behavior approach* untuk deteksi malware berdasarkan perilaku applikasi terhadap data ?
2. Bagaimanakah kinerja *behavior approach* dalam mendeteksi malware berdasarkan perilaku applikasi terhadap data ?

## Batasan Maslah

Beberapa batasan masalah yang ditetapkan di dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya mengambil satu sampel dari malware ransomware yaitu ransom.locky. Malware ransom.locky dipilih untuk dijadikan sampel karena malware ini merupakan tipe malware yang menjadikan data user berupa file dokumen sebagai terget dengan cara di enkripsi, selain itu malware ini merupakan malware dengan jumlah korban yang cukup banyak untuk tahun 2016 versi mcAfee.
2. Data yang menjadi sampel objek penelitian hanya file dokumen ber-ekstensi .DOC/X, .XLS/X, .PPT/X, .OD\*, .PDF serta file image ber-ekstensi .JPG, .JPEG, dan .PNG. File dokument dengan ekstensi tersebut dipilih karena merupakan file dokumen yang umum digunakan atau yang umumnya ada di dalam setiap komputer.

## Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dipaparkan, dapat ditentukan tujuan penelitian. Adapun tujuan dari penelitian ini adalah:

1. Menerapkan metode *behavior approach* untuk deteksi malware berdasarkan perilaku applikasi terhadap data.
2. Mengukur kinerja dari penerapan metode *behavior approach* untuk deteksi malware berdasarkan perilaku applikasi terhadap data*.*

## Manfaat Penelitian

Manfaat yang dihasilkan dari penelitian ini untuk memberikan kontribusi berupa teknik baru dalam hal pendeteksian malware berbasis *behavior detection,* dan diharapkan menjadi masukkan baru bagi peneliti yang akan melakukan penelitian dalam hal deteksi malware, dan bagi developer dibidang keamana komputer khususnya developer *tool* anti malware diharapkan menjadi sumbangan teknik baru yang dapat dimanfaatkan untuk mendeteksi dan mengatasi malware.

## *Review* Penelitian

Berikut ini akan dibahas ulasan tentang penelitian yang telah dilakukan sebelumnya berkaitan dengan pendeteksian malware, dalam tulisannya (Mujumdar, Masiwal, & Meshram, 2013) yang berjudul *Analysis of Signatue Based and Behavior Based Anti Malware Approaches* memaparkan tentang sistem anti malware yang ada saat ini, bahwa mayoritas sistem keamanan khususnya anti malware masih menerapkan sistem atau metode pendeteksian yang berbasis pada *signature detection,* hal itu menurut para peneliti ini tidak cukup untuk mencegah bahaya kemanan serta ancaman malware yang datang saat ini, penelitian ini juga membahas tentang dua metode pendeteksian malware yang ada saat ini yaitu melalui pendekatan *signature based*  dan melalui pendekatan *behavior based,* kemudian para penulis menutup dengan kesimpulan bahwa masing-masing metode baik *signature based* dan *behavior based* memiliki kelebihan dan kekurangan.

Penelitian dengan judul *From Malware Signatures to Anti Virus Assisted Attacks* adalah penelitian yang ditulis oleh (Wressnegger, Freeman, Yamaguchi, & Rieck, 2016), penelitian ini membahas tentang kelemahan yang terdapat pada metode pendeteksian malware berbasis *signature,* setelah menjelaskan tentang mekanisme pendeteksian malware menggunakan metode *signature* itu sangat simpel dan cepat tapi, jika penanganan *signature* tidak dilakukan dengan hati-hati hal itu akan menjadi masalah, berbalik dari mekanisme pertahanan menjadi instrumen serangan, para penulis juga menyajikan sebuah metode yang secara otomatis dapat menangani masalah tersebut.

Penelitian lain yang mencoba memberikan sebuah alternatif untuk pendeteksian malware berbasis *signature* ditulis oleh (Siddiqui, 2008), mengangkat tema *Data Mining Methods For Malware Detection,* peneliti ini ingin memberikan alternatif pendeteksian malware menggunakan *data mining,* alasan penulis karena sistem pendeteksian berbasis *signature* tidak dapat mendeteksi malware jenis baru, dalam mendukung penelitiannya penulis melakukan pengumpulan dan menganalisa ribuan file baik malware maupun program bersih untuk dibuatkan sebuah model yang akam mengklasifikasi program atau file berdasarkan kelasnya yaitu kelas malware atau kelas file/program bersih. Penulis mengklaim dari hasil eksperimen yang dilakukan, tingkat keberhasilan mendeteksi malware baru mencapai 98.4% dan dengan tingkat kesalahan atau *false positive* hanya 1.9% untuk pendeteksian.

Mengangkat tema *Malicious Behavior Detectinon using Windows Audit Logs,* (Berlin et al., 2015)melakukan penelitian tentang bagaimana mendeteksi malware menggunakan fasilitas bawaan pada sistem operasi Windows yaitu *audit log*, menurut para peneliti, *audit log* pada Windows memberikan informasi yang cukup untuk bisa mendeteksi adanya perilaku buruk sehingga dapat digunakan untuk mendeteksi adanya malware, dalam pengunjian yang dilakukan pada sejumlah sampel malware didapatkan hasil 83% dengan tingkat *false positive* hanya 0.1%.

Penelitian yang dilakukan oleh (Kheir, 2013) dengan judul *Behavioral Classification and detection of malware through HTTP user agent anomalies,* membahas tentang tantangan mendeteksi malware di lalu-lintas jaringan, berbagai teknik dilakukan oleh *Bootmaster* untuk menyembunyikan aktivitasnya di tengah lalu-lintas jaringan yang sangat besar, meskipun banyak aktivitas tidak normal pada bagian header *user-agent* HTTP namu hanya sedikit yang terdeteksi, olehnya itu peneliti melakukan penelitian untuk menganalisa perilaku tidak normar dalam lalu-lintas malware pada *HTTP user agent* *header.* Peneliti melakukan taksonomi dari malware untuk perilaku menyimpang pada *headerHTTP user agen*  dan mengusulkan taksonomi ini sebagai mekanisme untuk pendeteksian malware. Hasil yang didapatkan dari pengujian menggunakan mekanisme yang diusulkan yaitu berupa rendahnya angka *false positives.*

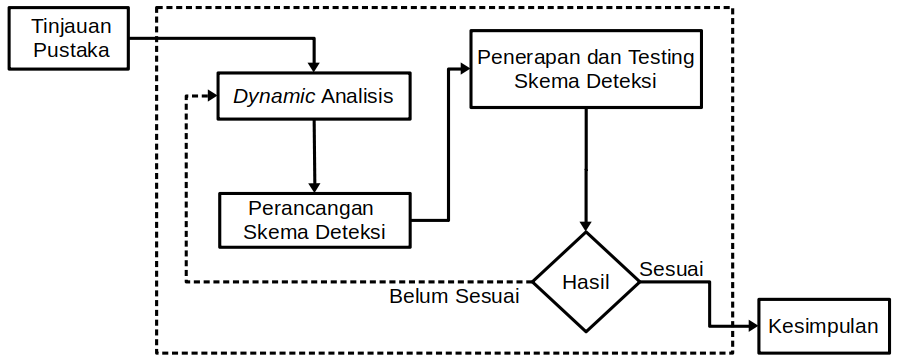
Tabel 1.2 Literatur Review Penelitian

| **No.** | **Paper** | **Masalah** | **Solusi** | ***Behavior Approach*** |
| --- | --- | --- | --- | --- |
|  | Berlin, K., Slater, D., & Saxe, J. (2015) | Sistem keamanan jaringan dan antivirus yang menggunakan metode *signature based* *detection* terbukti tidak cukup untuk mendeteksi ancaman tipe polimorfik malware pada komputer. | Melakukan pencegahan dengan pendekatan *behavior based detection* dengan memanfaatkan *audit logs* yang menjadi *standard build-in* pada sistem operasi Windows. | Melakukan eksplorasi beberapa fitur malware *behavior*  yang terekam pada log untuk semua kategori *behavior operation,* dengan target *operation behavior* adalah sistem. |
|  | Nizar Kheir, (2013) | Masalah pada deteksi anomali *botnet* pada lalu-lintas jaringan yang digunakan *bootmaster* untuk berkomunikasi dengan menyembunyikan aktivitas ditengah lalu-lintas jaringan yang sangat besar. | Menggunakan taksonomi dari anomali malware *user agent* untuk mekanisme pendeteksian malware, berbasis pada *behavior detections based* | Eksplorasi beberapa fitur *behavior operation* fokus pada *network operation* karena untuk menemukan anomali data pada jaringan, dengan sistem sebagai target *behavior operation.* |
|  | Siddiqui, (2008) | Malware baru dengan varian yang sama tidak dapat dideteksi dengan metode *signature based,* type polimorfik malware merupakan tantangan untuk metode baru untuk dapat mendeteksi tipe malware jenis virus yang dapat mengubah dirinya saat setelah menginfeksi sistem komputer untuk menghindari terdeteksi oleh antivirus. | Memperkenalkan data mining sebagai framework deteksi malware, menggunakan data mining untuk mengoleksi dan klasifikasi *behavior* malware dan *benign* program dengan pendekatan *statistical based detection*. | Mengumpulkan fitur-fitur yang dipilih dari ribuan malware dan *benign* aplikasi *behavior operation* dan diklasifikasikan*,* dengan target *operation behavior* adalah sistem. |
|  | Priyank Singhal, Nataasha Raul (2012) | Malware dapat datang dari media seperti *local network,* internet, ataupun *device* yang dipasang, pengguna komputer rumahan dapat menggunakan anti virus yang dapat selalu diupdate, tapi bagi sebuah perusahaan menjadi sangat riskan ketika virus baru dapat masuk dan menyerang semua komputer yang terhubung jaringan perusahaan. | Membangun sebuah antivirus yang dapat memeriksa file yang milintas pada jaringan dan juga dapat memriksa file yang berpotensi sebagai virus berdasarkan pada pengetahuan yang dibagnun dengan menggunakan metode *machine learning* dengan mengumpulkan *system API calls* yang dibuat oleh virus dan file yang diinfeksi oleh virus. | Mengumpulkan fitur-fitur *behavior* dari file virus dan file yang terinfeksi virus pada semua kategori  *behavior operation,* dengan target *operation behavior* adalah sistem. |
|  | Mamoun Alazab, Robert Layton, (2010) | Menawarkan lima langkah untuk bagaimana mendeteksi malware yang samar dengan melakukan invertigasi terhadap structural dan *behavioral features* dari API *calls.* | Menawarkan sistem otomatis yang membongkar dan melakukan ekstak terhadap API *calls features* secara efektif dari sebuah applikasi eksekutabel menggunakan metode analisis perhitungan *n-gram.* | Mengumpulkan fitur-fitur *behavior* dari malware pada semua kategori  *behavior operation,* dengan target *operation behavior* adalah sistem. |
|  | Penelitian yang diusulkan | Penelitian dalam hal deteksi malware dengan metode *behavior approach* yang ada saat ini tidak secara spesifik membahas perilaku applikasi terhadap data dimana malware termasuk didalamnya, hal ini yang menjadi masalah beberapa tahun terakhir seperti malware ransomware contohnya yang secara spesifik dirancang untuk menargetkan data pengguna komputer. | diperlukan sebuah penelitian yang membahas secara spesifik deteksi malware berbasi *behavior approach* dimana perilaku applikasi terhadap data dalam hal ini sebagai *target behavior operations* dapat digunakan sebagai cara deteksi malware. | Eksplorasi *behavior operation* fokus pada *file operation* dan *proses operation* karena untuk menemukan anomali pada data user, dengan user data sebagai target *behavior operation.* |

## Metodologi Penelitian

Agar penelitian ini terarah dan mendapatkan hasil yang maksimal, maka penelitian ini menggunakan beberapa tahapan metode penelitian sebagaimana dibawah ini:

1. Tinjauan Pustaka
2. Pendeteksian
3. *Dynamic a*nalisis malware
4. Perancangan skema untuk mendeteksi malware
5. Penerapan skema pendeteksian dan melakukan testing
6. Hasil, kembali ke poin 1 (analisis) jika hasil belum sesuai dan jika hasil telah sesui lanjut ke kesimpulan.
7. Kesimpulan.



Gambar 1.2 Alur Metode Penelitian.

Gambar 1.2 diatas merupakan alur dan tahapan-tahapan yang dilakukan dalam metode penelitian ini. Tahapan tinjauan pustaka dilakukan sebagai tahapan awal untuk memberikan dasar bagi arah penelitian, tahapan kedua merupakan bagian utama dari penelitian ini yang terdiri dari *dynamic* analisis, perancangan skema deteksi, penerapan dan testing skema serta evaluasi hasil, jika hasil belum sesuia maka proses kembali ke tahapan analisis, selanjutnya jika hasil sesuai maka tahapan terakhir yaitu menarik kesimpulan.

## Sistematika Penulisan

Laporan penelitian ini disusun dengan sistematika penulisan yang dapat mempermudah proses pembahasan penelitian. Adapun sistematika penulisan yang dimaksud adalah sebagai berikut:

BAB 1 PENDAHULUAN

Pendahuluan merupakan pengantar terhadap permasalahan yang akan dibahas. Bab ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, *literature review*, serta sistematika penulisan

BAB 2 LANDASAN TEORI

Bab ini memuat teori-teori penunjang yang digunakan sebagai dasar penelitian pendeteksian malware dan mitigasi data.

BAB 3 METODOLOGI PENELITIAN

Bab ini membahas tentang langlah-langkah penelitian dan gambaran umum langkah penyelesaian.

BAB 4 HASIL PENELITIAN DAN PEMBAHASAN

Hasil dan Pembahasan, berisi tentang pembahasan penyelesaian masalah yang diangkat, evaluasi dan hasil.

BAB 5 KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil penelitian serta saran dan rekomendasi untuk penelitian selanjutnya.

# BAB 2 Tinjauan Pustaka

## Malware

Malware merupakan kependekan dari *malicious software,* dalam definisi tentang malware disebutkan bahwa malware merupakan program yang sengaja dibuat untuk membahayakan dan merugikan sistem operasi atau data pada komputer (Siddiqui, 2008). Definisi ini mencakup semua program yang dapat merugikan atau membahayakan data dan sistem operasi seperti virus, worm, trojan, spyware dan yang sejenisnya.

### Sejarah malware

Sejarah kapan pertama kali malware ditemukan masih menjadi perdebatan, hasil penelitian ada yang menyebutkan pada awal tahun 1970-an malware telah ditemukan, Creeper merupakan contoh malware yang ditemukan pada tahun 1970-an awal di jaringan kompute ARPANET milik ARPA yang menyerbar hanya di jaringan milik perusahaan tersebut, malware ini dibuat oleh salah satu insiyur perusahaan teknologi *BBN Tecnology* yang bernama Robert “Bob” H. Thomas (Dalakov, 2012).

Namun (Milošević, 2013) menjelaskan bahwa sejarah munculnya malware dibagi berdasarkan kategori dan masing-masing kategori memiliki waktu kejadian munculnya malware, namun secara umum malware sudah ada sebelum tahun 1986, dan pada tahun 1986 muncul malware pertama kategori virus pada platform MsDos milik perusahaan Microsoft yang bernama Brain.A, malware ini dibuat di Pakistan oleh dua bersaudara Basit dan Amjad.

Sedangkan (Krebs, 2014) menjelaskan bahwa virus komputer merupakan satu dari sekian bayak tipe dari malware, dimana program yang dibuat dapat menggandakan diri dan didesain untuk menyebar dari satu komputer ke komputer lain. Masih menurut Krebs, sebenarnya teori tentang program yang dapat menggandakan dirinya sendiri sudah ada sejak tahun 1949 yaitu sebuah artikel berjudul “*Theory of Self-Reproducing Automata”* yang ditulis oleh John von Neumann dimana dalam artikel tersebut penulisnya menjelaskan tentang program komputer yang dapat menggandakan diri sendiri, namun artikel ini baru terkenal setelah di publis kembali pada tahun 1966.

Jika diurutkan berdasarkan waktu kejadian kapan pertama kali malware ditemukan, maka malware jenis virus merupakan malware yang pertama kali exis dari semua jenis malware, hal ini dijelaskan oleh (Krebs, 2014) dalam mengurutkan kejadian munculnya virus dari pertama kali yang diawali dengan tahun 1949 dengan terbitnya artikel berjudul “*Theory of Self-Reproducing Automata”* yang ditulis oleh John von Neumann kemudian pada tahun 1959 virus CoreWars yang dibuat oleh Victor Vysottsky, H. Douglas Mcllroy dan Robert P Morris di Bell Laboratorium, selanjutnya tahun 1960 virus Rabbit dan tahun 1971 Creeper malware jenis worm pertama kali ditemukan di jaringan milik ARPANET.

### Jenis-Jenis Malware

Terdapat banyak jenis malware, dalam penelitian ini disebutkan beberapa jenis-jenis malware tersebut dengan pengertiannya masing-masing.

1. Virus

Virus komputer merupakan program atau code yang dapat menggandakan dirinya dan meyerang file eksekusi (.exe). Virus biasanya membutuhkan canpur tangan manusia atau pengguna komputer untuk menempatkan dirinya dan menjalankannya.

1. Ransomware

Ransomware adalah malware yang dirancang untuk mencegah akses terhadap system atau data yang telah dikunci sampai dilakukannya pembayaran kepada penyerang (Microsoft, 2017). Terdapat dua jenis ransomware yaitu :

1. *Locker* Ransomware*,* atau disebut komputer *locker,* malware ini mengunci perangkat komputer milik korban dan hanya memberi akses terbatas pada keyboard seperti karakter dan nomor pada keyboard untuk korbah memasukkan kode pembayara. Hal lain yang dilakukan *locker* ransomware selain mencegah korban mengakses komputernya, malware ini juga meninggalkan pesan dan petunjuk ke korban hal-hal yang harus dilakukan korban.
2. *Crypto* Ransomware, atau disebut data *locker,* merupakan jenis ransamware yang melakukan pencarian dan enkripsi terhadap data pribadi yang menjadi korban dari malware ini. Penelitian ini akan mengambil salah satu type dari ransomware jenis *crypto* ransomware.
3. Worm

Worm mrupakan program *stand alone* yang dapat menggandakan dirinya dan menyebar di jaringan komputer. Worm biasanya tidak butuh bantuan manusia atau campur tangan pengguna komputer untuk memicunya berjalan dan menggandakan dirinya.

1. Trojan

Trojan atau dalam istilah lain disebut Trojan hourse merupakan program yang masuk dan melakukan beberapa aktivitas berbahaya dengan kedok sebagai program normal.

1. Spyware

Spyware adalah program yang diinstall secara diam-diam di komputer pribadi untuk menangkap atau mengambil informasi tentang interaksi pengguna dengan komputer tanpa sepengetahuan pengguna atau pemilik komputer.

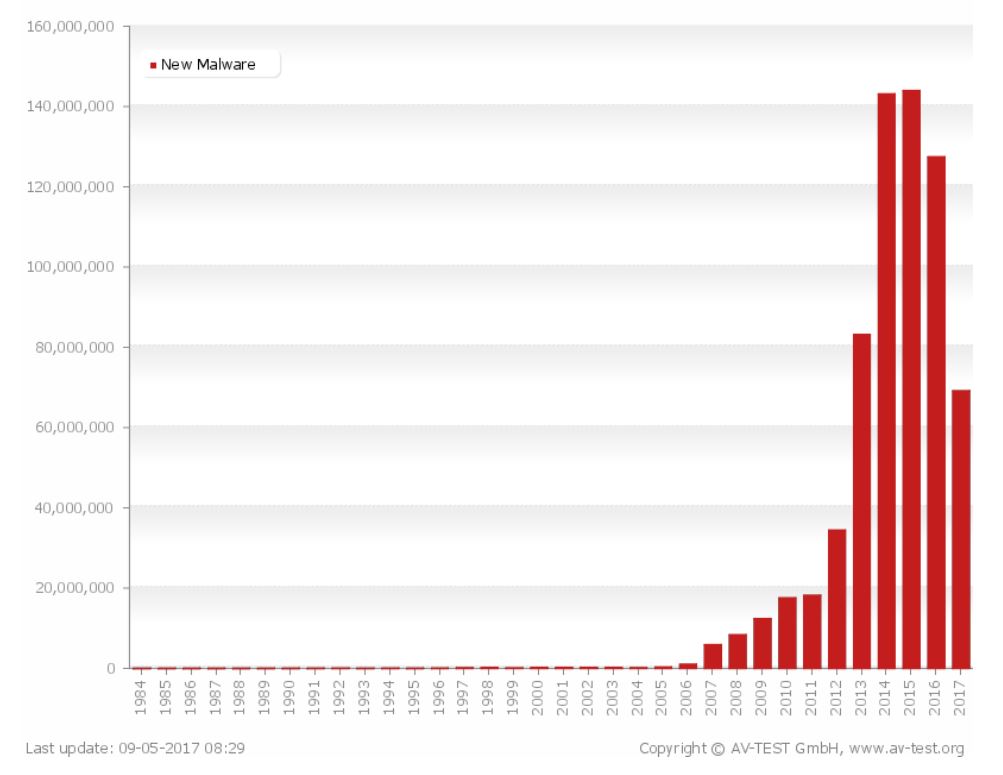
1. Rootkit

Rootkit pada awalnya merupakan program yang dipasang oleh penyerang pada sebuah sistem unix yang memungkinkan si penyerang mendapatkan hak akses sebagai administrator atau root. Saat ini rootkit digunakan lebih umum yaitu sebagai sebuah rutinitas dalam sebuah malware. Rootkit dapat mencegah agar prosesnya tidak terlihat dalam daftar proses yang berjalan pada sistem, atau mencegah agar filenya terbaca.

### Perkembangan Malware

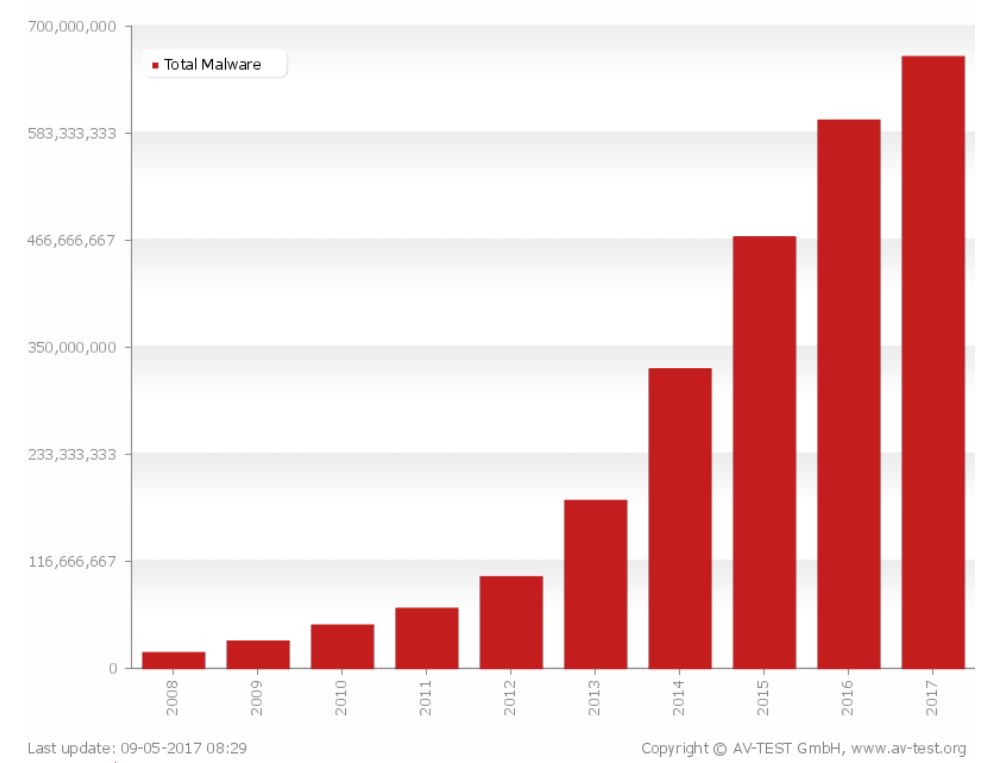
Perkembangan malware dari tahun ke tahun semakin meningkat terutama jika melihat grafik perkembangan malware untuk 10 (sepuluh) tahun terakhir, jika diawal-awal kemunculan malware jumlahnya tidak lebih dari 1 juta malware, maka untuk 5 (lima) tahun terakhir ditemukan 390.000 malware rata-rata setiap harinya, dalam laporannya (AV-TEST Institute, 2017) mulai merekam aktivitas perkembangan malware dimulai dari tahun 1984 sampain tahun 2017, peningkatan jumlah malware baru melebihi angka 1 (satu) juta dimulai dari tahun 2004 dan angka tertinggi malware baru yang ditemukan adalah di tahun 2014, 2015 dan 2016.

Tingginya jumlah peningkatan malware baru dihitung berdasarkan jumlah malware yang masuk ke database dan yang belum terekam dalam database sebelumnya, disamping metode pendeteksian malware yang masih menggunakan metode pendeteksian malware berbasis *signature based detection,* beberapa malware meskipun memiliki sekamaan *variant,* sistem, dan cara kerja namun pada kenyataannya anti virus dan program anti malware menganggapnya sebagai malware yang berbeda hal ini dikarenakan metode yang digunakan adalah metode deteksi yang masih menggunakan *signature based detection,* hal ini membuat beberapa malware yang dikembangkan menjadi malware baru meskipun dengan tingkat perubahan yang sangat sedikit akan dihitung sebagai malware baru, meskipun perubahan itu hanya berupa beberapa bari ataupun beberapa karakter hal ini cukup untuk dapat mengubah *signature* dari malware lama menjadi *signature* malware baru. Berikut merupakan grafik perkembangan malware baru berdasarkan laporan yang diambil dari situs (www.av-test.org, 2017), grafik menunjukan rekaman perkembangan malware baru sejak tahun 1984 sampai dengan tahun 2017.



Gambar 2.1 Malware baru dalam angka sejak tahun 1984 – 2017 (www.av-test.org, 2017).

Dari gambar 2.1 diatas menunjukan peningkatan jumlah malware baru yang sangat signifikan terjadi sejak tahun 2007, dan terus mengalami peningkatan jumlah setiap tahunnya dan baru mengalami penurunan ditahun 2016, dan pada tahun 2017 mengalami penurunan sangat jauh berdasarkan data yang terdapat pada grafik, penurunan dari angka 120 (seratus dua puluh) juta malware turun menjadi 70 (tujuh puluh) juta malware untuk tahun 2017. Adapun total malware yang berhasil direkam untuk semua kategori yaitu malware baru dan malware lama dalam kurun waktu 10 (sepuluh) tahun terakhir terus mengalami peningkatan dari tahun 2008 sampai dengan tahun 2017, hal ini sebagaimana yang terdapat pada grafik dibawah yang diambil dari situs yang sama (www.av-test.org, 2017), menujukan peningkatan yang sangat tinggi terjadi pada kurun waktu tahun 2013 ke tahun 2014 yaitu dari angka 200 (dua ratus) malware ke angka 300 (tiga ratus) malware, terus mengalami peningkatan lagi pada kurun waktu tahun 2014 ke tahun 2015 yaitu dari angka 300 (tiga ratus) juta malware ke 466 (empat ratus enam puluh enam) juta malware, dan masih mengalami perkembangan dari tahun 2015 ke tahun 2016 yaitu dari 466 (empat ratus enam puluh enam) juta malware menjadi 590 (lima ratus sembilan puluh) juta malware, selanjutnya perkembangan malware antara tahun 2016 ke tahun 2017 mengalami peningkatan namun tidak sebanyak tahun-tahun sebelumnya, hal ini dapat dilihat pada gambar 2.2 dibawah :



Gambar 2.2 Total Malware dalam kurun waktu 10 (sepuluh) tahun terakhir ([www.av-test.org](http://www.av-test.org), 2017)

Dari sekian banyak temuan malware yang paling banyak menarik perhatian dalam 5 (lima) tahun belakangan adalah malware jenis ransomware, bahkan tool anti virus McAfee (McAfee Labs, 2016) memberi judul laporannya dengan “Tahun 2016 akan diingat sebagai tahunnya Ransomware”, ransomware mulai diketahui sejak tahun 2006 menurut (Vysottsky, 2014), namun jenis malware yang memiliki prinsip kerja dengan melakukan proses enkripsi sudah ada sejak tahun 1989 yaitu *AIDS Trojan* dimana malware ini malakukan enkripsi terhadap nama file (Lab Kapersky, 2016) dan pada tahun 2006 muncul malware *gpcode*, *TROJ.RANSOM.A* dan beberapa malware lainnya yang memiliki kesamaan system kerja yaitu melakukan enkripsi terhadap file (Deloitte, 2016).

## Metode-Metode Deteksi

Perkembangan Jumlah malware yang terus bertambah sangat pesat dan masih akan terus bertambah membuat para peneliti keamana komputer harus melakukan terobosan untuk menemukan metode baru untuk melindungi komputer dari serangan malware. Secara umum terdapat beberapa metode deteksi malware, namun dari semua metode jika dirangkum maka akan menjadi tiga metode deteksi malware yang paling populer, ketiga metode deteksi malware tersebut yaitu *Signature Based Detection, Behavior Based Detection dan Statistical Based Detection* (Damodaran, 2015).

### *Signature Based Detection*

S*ignature based detection*  adalah metode deteksi malware dengan cara mencocokkan dua pola atau *signature* dimana salahsatu pola menjadi model untuk pola yang akan dicocokkan, proses untuk mengukur kesamaan dua pola dilakukan dengan menggunakan salah satu atau kombinasi dari tiga bagian *signature* yaitu *file magic number* atau yang biasa disebut dengan file *signature,* kemudian file *checksum* atau biasa disebut file hash dan, *string signature* yaitu deretan *bytes* tertentu dalam tubuh file malware. Proses untuk memeriksa file *signature* dilakukan untuk menemukan setidaknya satu pola urutan byte dengan pola urutan byte yang ada di database *signature*, database ini merupakan database yang berisi daftar *signature* yang terdiri dari kombinasi pola urutan byte malware atau file hash yang telah didefinisikan sebelumnya oleh pakar keamana melalui ananlisis.

*Signature based detection* merupakan metode deteksi malware yang umum digunakan pada *tools* anti virus saat ini, pada metode ini terdapat beberapa cara yang biasa digunakan yaitu menggunakan file *checksum* atau kombinasi dari ketiga cara yaitu file *checksum, string signature* dan file *signature.* Dalam metode *signature based detection* penggunaan fungsi hash atau file *checksum* merupakan hal paling diutamakan, danalgoritma *hash* yang mula-mula digunakan adalah CRC32, terakhir algoritma ini sudah ditinggalkan karena memiliki keterbasan dan mulai beralih ke algoritma *hash* md5 atau sha1, kerena alasan keterbatasan dan mulai ditemukan kelemahan pada algoritma yang digunakan seperti md5 dan sha1 maka para penliti mengembangkan algoritma baru yang sudah mulai diterapkan di lingkungan dunia malware dan keamanan computer seperti algoritma hash sha256.

### *Behavior Based Detection*

*Behavior based detection* adalah metode deteksi malware berdasarkan pada perilaku malware. Perilaku malware di*-extract* melalui proses analisis saat malware dieksekusi menggunakan *tool* analisis, selain itu proses analisis dilakukan untuk memahami maksud dari malware. Dalam metode ini perilaku malware dan applikasi normal dipelajari dan dilakukan pemantauan dalam masa waktu tertentu, setelah itu sebuah *executable* dikategorikan sebagai malware atau sebagai applikasi normal.

Metode *behavior based detection* ini merupakan metode deteksi yang digunakan dalam penelitian ini, dalam metode ini penulis diharuskan untuk melakukan analisis terhadap sebuah malware, dipelajari dan dipantau perilaku dari malware tersebut desertai dengan beberapa applikasi normal sebagai pembanding untuk mengetahui perilaku keduanya terhadap data, dari perilaku tersebut digunakan untuk deteksi malware.

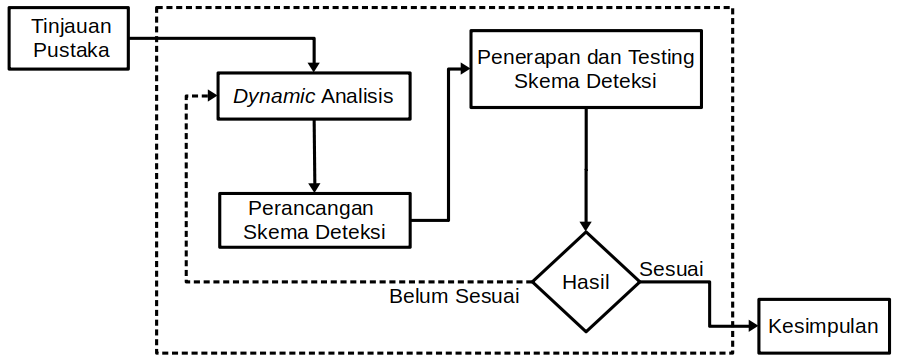
### *Statistic Based Detection*

Metode deteksi malware *Statistic Based Detection* adalah metode deteksi malware berdasarkan fitur-fitur yang tetdapat pada applikasi atau malware. *Hidden Markov Model (HMM)* adalah salah satu metode untuk deteksi malware yang berbasis *statistic detection.* Metode ini biasa digunakan di sebagian besar pada sistem *speech recognition,* digunakan juga untuk *patern recognition, artificial intelligence* serta dalam hal *malware detection.*

# BAB 3 Metodologi Penelitian

## Pendahuluan

Bab ini menjelaskan bagaimana cara penelitian dilakukan sehingga dapat diketahui rincian tentang urutan langkah-langkah yang dibuat secara sistematis dan dapat dijadikan pedoman yang jelas dalam menyelesaikan permasalahan, membuat analisis terhadap hasil penelitian, serta kesulitan-kesulitan yang dihadapi. Gambar 3.1 dibawah merupakan metode penelitian yang menjelaskan alur dan tahapan-tahapan yang dilakukan dalam penelitian ini :



Gambar 3.1 Alur metode penelitian

## Tinjauan Pustaka

Tinjauan pustaka dilakukan sebagai tahapan awal untuk memberikan dasar bagi arah penelitian yang akan dilakukan serta menjadi awal pemikiran bagi setiap peneliti sehingga penelitian yang dilakukan dapat dijadikan acuan kembali dikemudian hari. Tujuan dari tinjauan pustaka ini adalah untuk mendapatkan informasi mengenai topik dari penelitian yang dapat bersumber dari dokumen, buku, artikel, atau bahan tertulis lainnya, yang berupa teori, atau penemuan sebelumnya, baik bersifat *online source* maupun *offline source*.

Tinjauan pustaka dilakukan terhadap penelitian-penelitian sebelumnya yang berhubungan dengan deteksi malware menggunakan metode *behavior approach* untuk mendeteksi malware berdasarkan perilaku applikasi terhadap data, sehingga dapat dijadikan bahan-bahan untuk menunjang tujuan dari penelitian ini.

## Analisis

Sebelum membahas lebih jauh tahapan kedua dari metode penelitian, akan dijelaskan terlebih dahulu tentang malware analisis, dalam tulisannya (Yusirwan et al., 2015) membagi malware analisis menjadi dua yaitu *dynamic analysis* dan *static analysis*. *Static analysis* adalah metode analisis malware tanpa harus menjalankan malwarenya, hal ini membuat metode *static* analisis lebih aman dibanding dengan *dynamic analysis.* Sedangkan *dynamic analysis* adalah metode analisis malware dengan cara menjalankan malware secara langsung, untuk alasan keamanan maka malware akan dijalankan pada sebuah *virtual machine* sehingga mencegah agar malware tidak merusak sistem komputer.

Pada penelitian ini *dynamic* analisis merupakan analisis yang digunakan pada tahapan kedua dalam metode penelitian, *dynamic* analisis dilakukan untuk mendapatkan pemahaman yang mendalam tentang malware yang dijadikansampeldalam penelitian ini serta beberapa applikasi yang berhubungan dengan akses data. Tahapan kedua ini dibagi menjadi tiga sub-tahapan yaitu tahapan malware dan beberapa applikasi yang akan dianalisis, menentukan sistem operasi dan *tool* yang akan digunakan dalam penelitian ini serta terakhir adalah *extract behavior feature*, yaitu proses untuk mengeluarkan fitur perilaku dari malware yang dianalisis.

### Malware dan Applikasi yang dianalisis

Malware yang akan dilakukan analisis merupakan malware yang telah disebutkan pada batasan masalah yaitu *locky.ransomware*, malware ini di jadikan sampel untuk penelitian karena merupakan satu dari beberapa tipe *crypto* ransomware yang mengekripsi data dan merupakan ransomware dengan jumlah korban yang cukup banyak pada tahun 2016.

Selain malware, penulis melakukan analisis beberapa applikasi yang terkait dengan akses dan pengolahan data diantaranya beberapa applikasi pengolah data yaitu Ms.Office (word, excel, PowerPoint), LibreOffice, beberapa applikasi pengolah gambar yaitu ACDSee, serta beberapa applikasi bawaan sistem dan applikasi lainya yang dianggap perlu akan disertakan dalam analisis ini.

### Sistem dan *Tools* Analisis

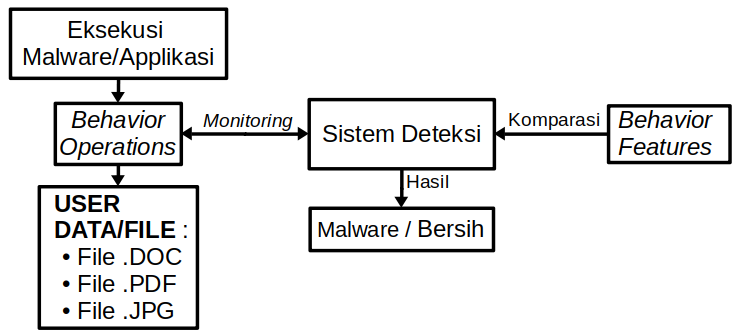
Sistem Operasi yang digunakan dalam analisis ini adalah sistem operasi *Ms.Windows 7.* Alasan dipilihnya Sistem Operasi *Windows 7* karena sistem operasi ini merupakan sistem operasi dengan jumlah pengguna terbanyak yaitu 52% dalam kurun waktu 4 (empat) tahum terakhir menurut situs *www.netmarketshare.com*. *Tool* analisis yang digunakan dalam penelitian ini adalah beberapa *tool* yang biasa digunakan untuk analisis walware yaitu *VirtualBox,* *Cuckoo Sandbox, Proces Explorer, Process Hacker, Process Monitor* dan modul python *WinAppDbg*.

### Analisis dan *Extract Behavior Features*

Tahapan ini adalah tahapan untuk mendapatkan *behavior features,* yaitu mengeluarkan perilaku dari malware dalam mengakses data yang dianggap *uniq* dari perilaku normal applikasi saat dilakukan dinamikanalisis. Fokus proses *extract behavior features* dariperilaku malware dilakukan berdasarkan pada operasi file yaitu *create, read, write, rename, delete, open* dan operasi proses yaitu *create, terminate,* oparasi lainnya akan disertakan jika diperlukan*.* Hasil dari *extracting behavior features* nantinya digunaka sebagai bagian dalam sistem deteksi untuk membandingkan perilaku yang normal dan perilaku yang tidak normal.

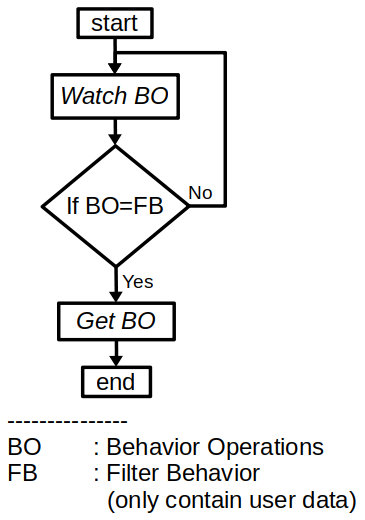
## Perancangan Skema Deteksi

Perancangan skema deteksi merupakan tahapan untuk merancang skema untuk deteksi malware, rancangan dibuat berdasarkan pada pengetahuan yang didapat dari tahapan sebelumnya yaitu tahapan analisis. Seletelah tahapan analisis dan *behavior features* telah ditetapkan maka dibuat rancangan sistem deteksi malware. Berikut merupakan gambaran skema sistem deteksi malware :



Gambar 3.2 Skema sistem deteksi malware

Pada gambar 3.2 diatas menjelaskan tentang skema sistem deteksi malware yang akan dibuat, tahapannya adalah tahapan eksekusi malware atau applikasi, pada tahapan ini malware yang telah ditetapkan sebagai sampel akan dieksekusi. Tahapan berukutnya adalah sistem deteksi melakukan monitoring detail *behavior* dari applikasi yang melakukan akses ke data. Selanjutnya adalah proses komparasi dengan *behavior features* yang telah ditetapkan serta tahapan terakhir adalah sistem deteksi memberikan hasil penilaian apakah proses applikasi yang diperiksa termasuk kategori malware ataukan applikasi bersih. Berikut merupakan gambaran tahapan-tahapan dalam proses *monitoring* dan proses komparasi yang dijelaskan dalam bentuk gambar flowchart yaitu pada gambar 3.3 dan gambar 3.4 agar memberikan gambaran detail dari proses yang ada pada gambar 3.2:

****

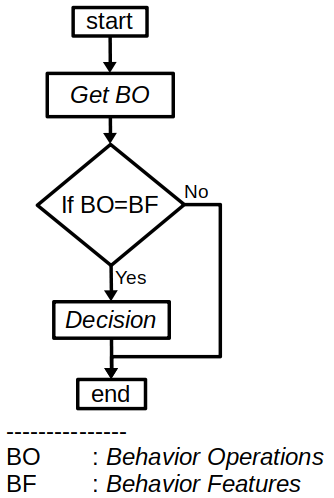
Gambar 3.3 Monitoring proses

1. ***Monitoring***

Proses *monitoring* adalah proses sistem deteksi mengawasi setiap *behavior operations.* Gambar 3.3 disamping menjelaskan tahapan langkah-langkah dalam proses *monitoring,* berikut penjelasan masing-masingtahapan :

1. *Start*, mulai proses *monitoring*
2. *Watch BO,* proses mengawasi setiap *behavior operations*
3. *If BO=FB*, proses untuk memfilter *behavior operations* yang mengandung data user,
4. *Get BO,* proses mengambil *behavior* untuk dibandingkan, dan
5. *End,* prose selesai.
6. **Komparasi**

Proses komparasi merupakan proses untuk membandingkan antara *behavior* yang diambil dari proses *monitoring* dengan *behavior features* yang telah ditetapkan. Gambar 3.4 disamping menjelaskan tahapan langkah-langkah proses komparasi, yang terdiri dari :



Gambar 3.4 Proses Komparasi *Behavior*

1. *Start*, proses komparasi,
2. *Get BO*, proses mengambil *behavior* yang diambil dari proses sebelumnya.
3. *If BO=BF*, proses komparasi *behavior* dengan *behavior features.*
4. *Decision,* proses membuat keputusan dan menampilkan hasil keputusan.
5. *End,* selesai proses.

## Implementasi dan Testing Skema Deteksi

Dalam tahapan ini dilakukan implementasi skema sistem deteksi malware berdasarkan pada apa yang telah dirancang pada bagian sebelumnya.

### Implementasi

Dalam tahapan ini skema sistem deteksi akan diimplementasikan dengan menggunakan bahasa pemrograman python. Selanjutnya setelah tahapan implementasi adalah melakukan testing, hal ini dilakukan sampai mendapatkan hasil yang diinginkan.

### Testing sistem

Tahapan testing 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. Adapaun metode pengujian yang digunakan adalah metode statistik.

## Hasil

Hasil merupakan tahapan evaluasi hasil, setelah tahapan penerapan dilanjutkan dengan tahapan testing maka tahapan selanjutnya adalah evaluasi hasil, hal ini dilakukan untuk melihat jika hasil belum sesuia maka proses akan kembali ke tahapan analisis sampai didapatkan hasil yang sesuai.

## Kesimpulan

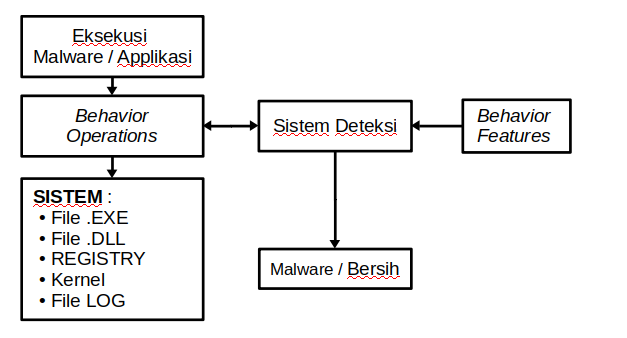
Tahapan terakhir dari penelitian ini adalah membuat kesimpulan tentang bagainamakah penerapan metode *behavior approach* untuk deteksi malware serta bagaimanakah kinerja dari metode *behavior approach* dalam mendeteksi malware berdasarkan perilaku applikasi terhadap data.

# 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 applikasi 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.

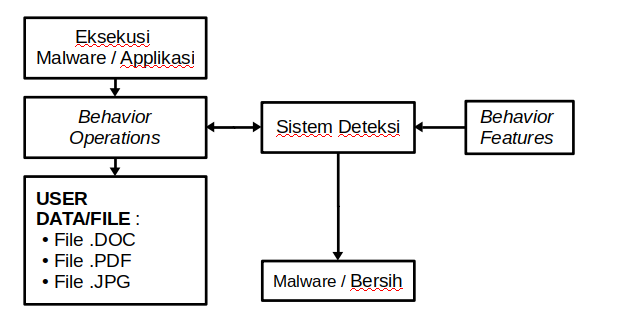
## Penerapan Metode *Behavior Approach*

Deteksi malware menggunakan metode *behavior approach* adalah metode deteksi malware berdasarkan perilaku sebuah appalikasi yang diketahui setelah melalui analisis, untuk dapat mengetahui perilaku sebuah applikasi secara tepat diperlukan analisis langsung terhadap applikasi 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 :

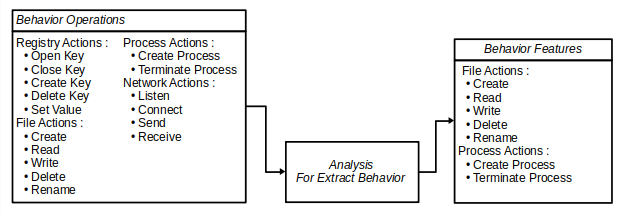


Gambar 4.2 Target *behaviour operations* pada penelitian yang diusulkan

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.

### *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 applikasi disebut sebagai *behavior operstions*. Sedangkan *behavior features* merupakan istilah yang digunakan pada penelitian ini untuk menggambarkan *behavior operations* yang telah diekstrak dari sebuah applikasi atau malware melalui proses analisis, *features* yang diekstrak merupakan *behavior* yang dianggap dapat digunakan untuk mengenali applikasi 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 applikasi ataupun malware yang dianalisis.

### User Sebagai Target *Behavior operations*

Target *behavior* merupakan target dari sebuah operasi *behavior* yang terjadi pada sebuah applikasi, 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.

## 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 |
| 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) |

## 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.

### Malware yang digunakan

Malware yang digunakan sebagai sampel adalah malware locky runsomware 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.

### 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.

### Dinamik Analisis

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 :

1. 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*.

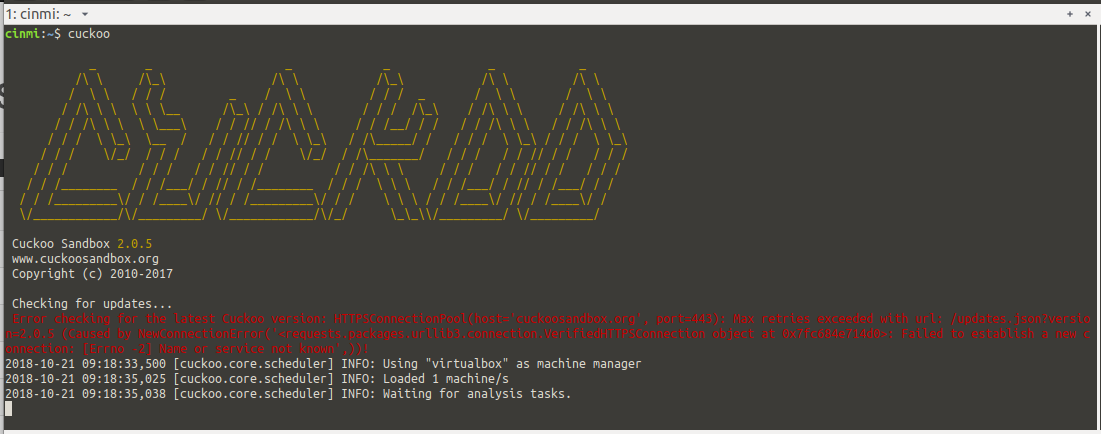
1. *Process Explorer, Process Hacker, Process Monitor*

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.*

1. *Cuckoo Sandbox*

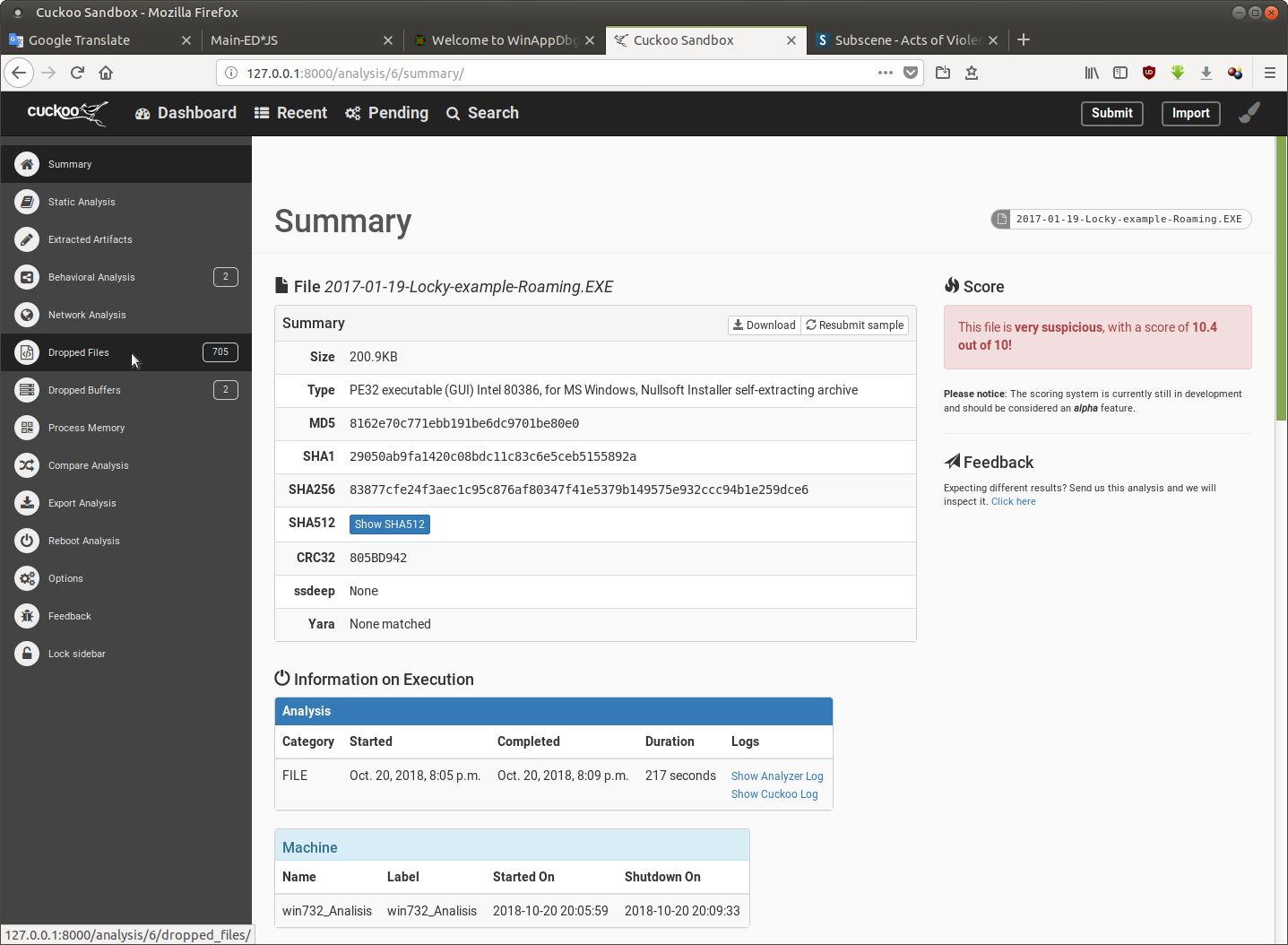
Tahapan analisis menggunakan *Cuckoo sandbox* adalah proses untuk mendapatkan informasi dari sebuah file yang di-eksekusi pada sebuah mesih virtual virtualbox. Proses analisis menggunakan cuckoo diawali dengan meng-install *cuckoo* *sandbox* dilanjutkan dengan *setting* konfigurasi sampai *cuckoo* siap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

1. Menjalankan *cuckoo web interface* dengan perintah “*cuckoo web runserver*” yang bertujuan agar informasi hasil analisis dapat diakses melalui browser pada mesin lokal atau *localhost*.
2. 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](http://127.0.0.1:8000) selanjutnya dapat memilih tombol submit yang disediakan pada halaman tersebut.
3. *Cuckoo* akan menapilkan pesan ketika proses analisis telah selesai, pesan pemberitahuan akan tampil pada terminal ketika prose berhasil ataupun proses gagal.
4. 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*](http://127.0.0.1:8000)*.* Berikut screenshot hasil analisis malware locky dengan nama file “*2017-01-19-Locky-example-Roaming.EXE*” :



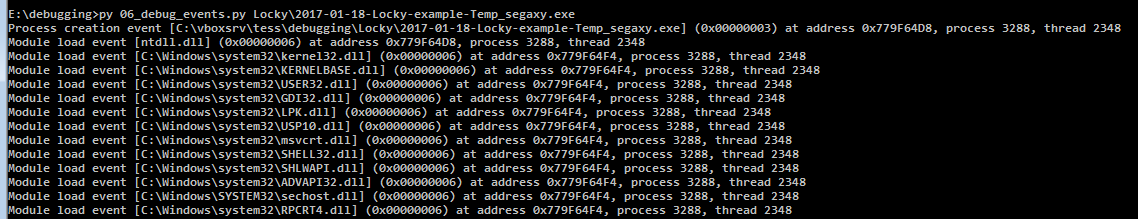
Gambar 4.5 Screenshot hasil analisis *cuckoo* pada malware

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 applikasi dijalankan, berapa jumlah file yang di-drop, serta memberikan score pada malware, nilai score yang tinggi mengindikasikan file yang dianalisis adalah malware.

1. 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 events,* 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, events* 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 events, proses ini bertujuan untuk melihat lebih detail event-event saat file applikasi 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.*

### *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 mengambilperilaku malware berdasarkan pada operasi file yaitu *create, read, write, rename, delete, open* dan operasi proses yaitu *create process, terminate process.*

## Implementasi Sistem Deteksi Malware

Tahapan implementasi sistem deteksi adalah tahapan untuk meng-implementasikan sistem deteksi malware berdasarkan perilaku applikasi 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.

### 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/>, proses instalasi python di windows sama halnya dengan proses instalasi *software* pada umumnya di windows, diawalai dengan memilih user yang akan menggunakan python dan dilanjutkan memilih lokasi instalasi, selanjutnya mengikuti arahan installer sambil menunggu proses instalasi selesai.

### Instalasi Modul Python

Setelah python terinstall, selanjutnya adalah meng-install modul-modul yang diperlukan untuk membangun sistem deteksi. Modul python merupakan file pustaka yang berisi kumpulan class dan fungsi yang dibuat dan disediakan untuk memudahkan programmer atau developer dalam membuat script, umumnya modul python disediakan secara gratis.

1. Instalasi Modul Python *psutil*

Nama modul *psutil* merupakan kependekan dari *process and system utilities,* merupakan pustaka lintas *platform* untuk mengambil informasi tentang proses yang berjalan dan penggunaan sistem meliputi cpu, memori, disk, jaringan, dan sensor menggunakan python. Instalasi modul psutil dilakukan dengan printah “*pip install psutil”* menggunakan *command line* (cmd.exe) dengan asumsi komputer harus terhubung dengan jaringan internet.

1. Instalasi Modul Python *pywin32*

*Pywin32* merupakan modul python untuk *platform Microsoft Windows* yang menyediakan banyak akses ke *API Win32*. Instalasi modul *pywin32* dilakukan dengan perintah “*pip install pywin32*” menggunakan *command line* (cmd.exe).

1. Install Modul Python *winappdbg.*

Deskripsi tentang modul *winappdbg* ini telah diulas pada poin sebelumnya sehingga tidak perlu diulas kembali. Instalasi modul *winappdbg* dilakukan dengan perintah “*pip install winappdbg”* menggunakan *command line* (cmd.exe).

### 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.

1. File Script Pendukung

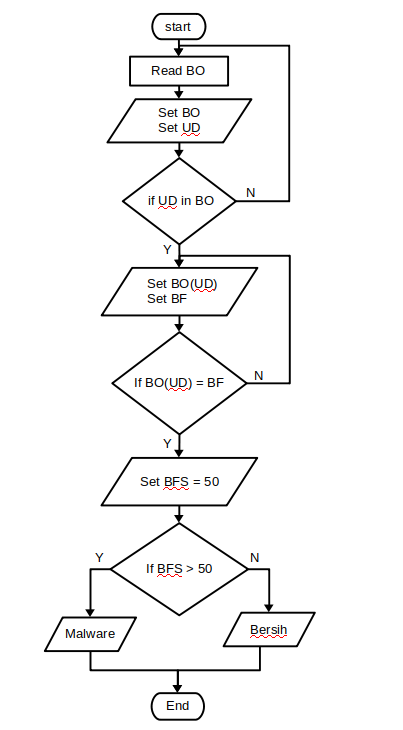
Persiapan untuk membuat sistem deteksi malware diawali dengan mempersiapkan informasi yang dikumpulkan dari tahapan sebelumnya yaitu tahapan analisis, informasi ini dijadikan sebagai data yang akan diimplementasikan kedalam script dalam bentuk pendeklarasian variabel-variabel dalam file script python. 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, variabel ini bertipe list dan digunakan untuk menampung jenis file yang akan diproses seperti .doc, docx dan jenis file yang lainnya, fungsi variabel ini ketika digunakan adalah sebagai filter sehingga, hanya jenis file yang ekstensinya sama dengan salah satu jenis file yang ada dalam variabel ini yang akan diproses oleh proses komparasi. Terdapat variabel lainnya dalam file *myvar.py* namun tidak semua akan dijelaskan disini dikarenakan penjelasan tersebut akan memerlukan banyak tempat sehingga hal yang dijelaskan disini adalah yang dianggap penting dan perlu dari variabel-variabel yang terdapat pada file *myvar.py*.
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. Beberapa fungsi penting yang telah disebutkan perlu dijelaskan seperti fungsi manipulasi proses, terdapat beberapa fungsi untuk manipulasi proses dan umumnya fungsi manipulasi proses yang dibuat dalam file *mymodul.py* menggunakan pustaka python yang bernama *psutil*, diantara fungsi tersebut adalah *getexe(pid),* fungsi ini berguna untuk mengambil sebuah parameter bernama pid dan mengembalikan nilai berupa nama dari applikasi yang menggunakan pid tersebut, berikutnya adalah fungsi *getpidexe(name),* fungsi ini merupakan kebalikan dari fungsi sebelumnya yaitu fungsi ini akan mengembalikan nilai berupa pid dari nama applikasi yang diberikan pada parameter, selanjutnya adalah fungsi *getuserps()* merupakan fungsi untuk mengambil semua nama proses yang dimiki oleh user saja dan mengabaikan proses yang dimiliki oleh sistem dan *service,* fungsi ini hanya mengembalikan nilai berupa nama applikasi dari proses yang berjalan sedangkan untuk mengambil pid dari proses yang berjalan digunakan fungsi *getuserps\_id()*, fungsi lain untuk manipulasi proses adalah fungsi *pspause(pid)* dan fungsi *psresume(pid),* kedua fungsi ini merupakan pasangan dimana *pspause(pid)* digunakan untuk memaksa suatu proses berhenti sementara waktu berdasarkan pid yang ada di parameter, sedangkan fungsi *psresume(pid)* digunakan untuk melanjutkan proses yang dihentikan sementara waktu berdasarkan pid yang di masukkan di parameter. Fungsi-fungsi yang telah dijelaskan diatas merupakan beberapa fungsi utama yang berguna untuk mendukung script utama.
3. 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 kembali ke point nomor 2
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 kembali ke point nomor 7
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, 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 *events*, yaitu proses untuk memastikan hanya *events* 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 *events* 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 melakukanpenggabungan data, memilah data berdasarkan *behavior,* menghitung data berdasarkan *behavior,* menghitung total file yang diakses, proses menghitug setiap *events* 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.

## 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.

### Testing Untuk Applikasi

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

1. Testing untuk *MsOffice Word*

Testing pertama yang dilakukan adalah testing terhadpa applikasi pengolah kata *MsOffice Word,*  testing dilakukan dengan cara membuka data user berupa file document MsWord yang berekstensi .docx untuk melihat *behavior* yang terrekam oleh sistem deteksi, berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat applikasi *MsWord* dijalankan untuk mengakses file dokument 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** |
|  | *Total Event* | 58 | 57 | 24 | 46.33 |
|  | *Open Events* | 0 | 0 | 0 | 0.00 |
|  | *Created Events* | 20 | 22 | 4 | 15.33 |
|  | *Changed Events* | 30 | 24 | 10 | 21.33 |
|  | *Deleted Events* | 8 | 11 | 10 | 9.67 |
|  | *Renamed Events* | 0 | 0 | 0 | 0.00 |
|  | *Total Files* | 11 | 14 | 11 | 12.00 |
|  | *Files Exist* | 10 | 8 | 2 | 6.67 |
|  | *Files Missing* | 1 | 6 | 9 | 5.33 |
|  | *Signature Know* | 1 | 1 | 0 | 0.67 |
|  | *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 applikasi *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 *events* dan infomasi file, *total events* merupakan jumlah total *events* yang merangkum beberapa *events* dibawahnya yaitu *open events, created events, changed events, deleted events,* dan *renamed events,*  dari tiga kali testing *total events* menunjukan angka rata-rata 46.33, data testing total *events* menunjukan data yang konsisten kecuali pada testing ketiga yang hanya setengah dari dua data testing sebelumnya, dari lima *events* hanya tiga *events* yang menunjukan angka diatas 1 yaitu *created, changed,* dan *deleted* sedangkan dua *events* lainya bernilai 0. Total file adalah total file yang diakses, data diatas menunjukan 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 diperiksaumumnya merupakan file *temporary.*

1. Testing untuk *MsOffice Excel*

Testing kedua dilakukan terhadpa applikasi pengolah data yaitu *MsOffice Excel,* testing dilakukan dengan membuka documen file berekstensi .xlsx, berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat applikasi *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*** |
|  | *Total Event* | 37 | 22 | 24 | 27.67 |
|  | *Open Events* | 0 | 0 | 0 | 0.00 |
|  | *Created Events* | 10 | 8 | 8 | 8.67 |
|  | *Changed Events* | 12 | 10 | 10 | 10.67 |
|  | *Deleted Events* | 6 | 4 | 6 | 5.33 |
|  | *Renamed Events* | 2 | 0 | 0 | 0.67 |
|  | *Total Files* | 4 | 3 | 3 | 3.33 |
|  | *Files Exist* | 3 | 3 | 3 | 3.00 |
|  | *Files Missing* | 1 | 0 | 0 | 0.33 |
|  | *Signature Known* | 1 | 0 | 0 | 0.33 |
|  | *Signature Unknown* | 3 | 3 | 3 | 3.00 |

Data pada tabel 4.3 merupakan hasil testing tiga kali untuk applikasi MsExcel, data menunjukkan pola data yang hampir sama dengan pola data pada hasil testing sebelumnya pada applikasi MsWord, data rata-rata *events* menunjukan penurunan hampir setengah jika dibandingkan data total *events* pada tabel 4.2 dengan total *events* 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* menunjukan konsistensi pada angka 3 disetiap testing sama halnya dengan data *signature unknown* yang juga menunjukan angka yang sama, sehingga secara umum data pada tabel 4.2 dan tabel 4.3 tidak menunjukan berbedaan angka yang cukup jauh.

1. Testing untuk *MsOffice PowerPoint*

Testing applikasi ketiga adalah testing terhadpat applikasi *MsOffice PowerPoint,* testing dilakukan dengan membuka document .pptx menggunakan applikasi *MsOffice PowerPoint,*  berikut merupakan hasil dari beberapa kali melakukan pengujian sistem deteksi malware saat applikasi *MsPowerPoint* dijalankan, dan berikut merupakan hasil yang dirangkum dalam tabel :

Tabel 4.4 Data hasil testing *application* *behavior MsPowerPoint*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO.** | **Daftar *Behavior*** | **Data *Behavior***  **Per Testing** | | | **Jumlah**  **Rata-Rata** |
| ***Test 1*** | ***Test 2*** | ***Test 3*** |
|  | *Total Event* | 14 | 13 | 37 | 21.33 |
|  | Open Events | 0 | 11 | 22 | 11.00 |
|  | *Created Events* | 4 | 1 | 3 | 2.67 |
|  | *Changed Events* | 9 | 1 | 11 | 7.00 |
|  | *Deleted Events* | 1 | 2 | 1 | 1.33 |
|  | *Renamed Events* | 0 | 0 | 0 | 0.00 |
|  | *Total Files* | 3 | 3 | 3 | 3.00 |
|  | *Files Exist* | 3 | 3 | 3 | 3.00 |
|  | *Files Missing* | 0 | 0 | 0 | 0.00 |
|  | *Signature Known* | 1 | 1 | 2 | 1.33 |
|  | *Signature Unknown* | 2 | 2 | 1 | 1.67 |

Data pada tabel 4.4 merupakan hasil testing tiga kali untuk applikasi *MsPowerPoint*, data menunjukkan pola data yang juga hampir sama dengan pola data pada hasil testing sebelumnya pada applikasi *MsWord*, data rata-rata *events* menunjukan penurunan jika dibandingkan data total *events* pada tabel 4.3 yaitu 27.67 dibanding data pada table 4.4 dengan total *events* 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 menunjukan berbedaan data yang cukup jauh. Data-data yang dihasilkan dari testing terhadap applikasi *MsOffice* pada tabel 4.2, dan tabel 4.3 serta tabel 4.4 menunjukan perbedaan data yang tidak terlalu jauh.

1. Testing untuk *Adobe Acrobat Reader*

Testing applikasi keempat adalah testing terhadpat applikasi *Adobe Acrobat Reader,* testing dilakukan terhadpa applikasi pembaca file pdf dengan membuka file berekstensi .pdf*,*  berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat applikasi *Adobe Acrobat Reader* dijalankan, berikut merupakan hasil yang dirangkum dalam tabel :

Tabel 4.5 Data hasil testing application *behaviour* *Adobe Acrobat Reader*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO.** | **Daftar *Behavior*** | **Data *Behavior***  **Per Testing** | | | **Jumlah**  **Rata-Rata** |
| ***Test 1*** | ***Test 2*** | ***Test 3*** |
|  | *Total Event* | 2 | 1 | 2 | 1.67 |
|  | Open Events | 0 | 0 | 0 | 0.00 |
|  | *Created Events* | 1 | 1 | 1 | 1.00 |
|  | *Changed Events* | 1 | 0 | 1 | 0.67 |
|  | *Deleted Events* | 0 | 0 | 0 | 0.00 |
|  | *Renamed Events* | 0 | 0 | 0 | 0.00 |
|  | *Total Files* | 1 | 1 | 1 | 1.00 |
|  | *Files Exist* | 1 | 1 | 1 | 1.00 |
|  | *Files Missing* | 0 | 0 | 0 | 0.00 |
|  | *Signature Known* | 0 | 0 | 0 | 0.00 |
|  | *Signature Unknown* | 1 | 1 | 1 | 1.00 |

Data pada tabel 4.5 diatas menunjukkan perbedaan yang jauh dibandingankan dengan data yang ada pada beberapa tabel sebelumnya, jika data pada ketiga tabel sebelumnya jumlah total *events* rata-rata diatas angka 21.33 maka data total *events* yang dihasilkan dari applikasi *Adobe Acrobat Reader* hanya 1.67 *events,* data total file hanya 1 file dengan file yang tidak dikenali masing-masing 1 file setiap testing.

1. Testing untuk *Xn View Image*

Testing applikasi kelima adalah testing terhadpat applikasi *Xn View Image,* testing dilakukan terhadpa applikasi pembuka atau pengelola gambar, testing dilakukan dengan membuka file gambar berekstensi .jpg*,*  berikut merupakan hasil dari beberapa kali pengujian sistem deteksi malware saat applikasi *Xn View Image* dijalankan, berikut merupakan hasil yang dirangkum dalam tabel :

Tabel 4.6 Data hasil testing application *behaviour Xn View Image*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO.** | **Daftar *Behavior*** | **Data *Behavior***  **Per Testing** | | | **Jumlah**  **Rata-Rata** |
| ***Test 1*** | ***Test 2*** | ***Test 3*** |
|  | *Total Event* | 2 | 1 | 1 | 1.33 |
|  | Open Events | 0 | 0 | 0 | 0.00 |
|  | *Created Events* | 1 | 1 | 1 | 1.00 |
|  | *Changed Events* | 1 | 0 | 0 | 0.33 |
|  | *Deleted Events* | 0 | 0 | 0 | 0.00 |
|  | *Renamed Events* | 0 | 0 | 0 | 0.00 |
|  | *Total Files* | 1 | 1 | 1 | 1.00 |
|  | *Files Exist* | 1 | 1 | 1 | 1.00 |
|  | *Files Missing* | 0 | 0 | 0 | 0.00 |
|  | *Signature Known* | 0 | 0 | 0 | 0.00 |
|  | *Signature Unknown* | 1 | 1 | 1 | 1.00 |

Data pada tabel 4.6 diatas menunjukkan perbedaan yang tipis dengan data pada tabel 4.5, data total *events* yang dihasilkan dari applikasi *Adobe Acrobat Reader* hanya 1.67 *events* sedangkan pada applikasi *Xn View Image* diatas menunjukkan 1.33 total *events* 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.

### 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 4.7 Data hasil testing malware *Locky Ransomware* *behaviour*

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

Data pada table 4.7 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 applikasi sebelumnya, tingginya perbedaan data tidak hanya pada satu poin melainkan semua poin pada tabel kecuali poin *created events*, dari poin pertama yaitu total *events*  jika rata-rata data tertinggi pada tabel 4.2 sampai tabel 4.6 hanya mencapai angka 46.66, maka angka rata-rata total *events* dari hasil testing malware pada tabel 4.7 menunjukkan peningkatan 23 kali lipat dari data tertinggi total *events* hasil sebelumnya yaitu pada tabel 4.2, peningkatan data lebih dari 10 kali lipat juga terjadi pada poin lain yaitu poin *changed events, renamed events, files exist, file missing,* dan poin *signature unknow.* Perbedaan data angka rata-rata yang sangat tinggi antara malware dengan applikasi pada setiap poinnya dapat dilihat dengan lebih jelas pada penjelasn berikutnya yaitu tabel 4.8.

### Hasil Testing Rata-Rata

Hasil testing terhadap beberapa applikasi menunjukan data-data yang telah disampaikan pada beberapa table diatas, maka data-data tersebut dapat diambil bagian rata-rata dari setiap hasil dan dikumpulkan menjadi satu tabel, yaitu dari hasil testing applikasi sampai hasil testing malware maka berikut merupakan hasil dari rata-rata setiap tabel-tabel diatas yang dikumpulkan menjadi satu tabel :

Tabel 4.8 Data rangkuman hasil testing rata-rata per applikasi dan malware

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **NO.** | **Daftar *Behavior*** | **Data *Behavior***  **Rata-Rata** | | | | | |
| **MsWord** | **MsExcel** | **MsPower Point** | **Adobe Reader** | **XnView Image** | **Locky Malware** |
|  | *Total Event* | 46.33 | 27.67 | 8.67 | 1.67 | 1.33 | 1065.33 |
|  | Open Events | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|  | *Created Events* | 15.33 | 8.67 | 2.67 | 1.00 | 1.00 | 3.67 |
|  | *Changed Events* | 21.33 | 10.67 | 4.33 | 0.67 | 0.33 | 511.00 |
|  | *Deleted Events* | 9.67 | 5.33 | 1.67 | 0.00 | 0.00 | 0.00 |
|  | *Renamed Events* | 0.00 | 0.67 | 0.00 | 0.00 | 0.00 | 329.33 |
|  | *Total Files* | 12.00 | 3.33 | 2.00 | 1.00 | 1.00 | 364.33 |
|  | *Files Exist* | 6.67 | 3.00 | 1.67 | 1.00 | 1.00 | 261.00 |
|  | *Files Missing* | 5.33 | 0.33 | 0.33 | 0.00 | 0.00 | 103.33 |
|  | *Signature Known* | 0.67 | 0.33 | 0.33 | 0.00 | 0.00 | 0.00 |
|  | *Signature Unknown* | 11.33 | 3.00 | 1.67 | 1.00 | 1.00 | 364.33 |

Laporan data pada tabel 4.8 diatas merupakan data rata-rata hasil pengujian sistem deteksi malware terhadap aktivitas applikasi dan malware locky ransomware dalam melakukan akses terhadap data user. Tabel 4.8 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 applikasi dan malware menggunakan sistem deteksi malware. Hasil rangkuman data rata-rata menunjukan perbedaan sangat tinggi, data rata-rata tertinggi untuk poin *total events* selain malware adalah data dari applikasi *MsWord* yaitu 46.33 dan terendah adalah data milik applikasi *XnView Image* yaitu 1.33, sedangkan data total file yang diakses tertinggi selain malware adalah milik applikasi *MsWord* yaitu 12 file dan yang terendah yaitu 1 file, dengan demikian data rata-rata dari beberapa applikasi tersebut pada point total *events* menunjukkan angka antara 1.33 sampai dengan 46.33 *events* 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 *events* adalah berjumlah 643 *events* sedangkan tertinggi adalah berjumlah 1769 *events,* adapun pada poin jumlah total file data terendah total file yang diakses adalah berjumlah 153 file dan tertingi adalah sebanyak 755 file yang diakses.

## Hasil dan Kesimpulan

### Kesimpulan Hasil Testing

Tingginya berbedaan jumlah data *events,* jumlah file yang diakses, serta perbedaan data lainya yang dihasilkan dari pengujian antara applikasi malware dengan beberapa applikasi yang umum digunakan oleh pengguna komputer untuk membuka data mereka seperti file dokument, file pdf serta file gambar sebagaimana applikasi tersebut diatas, menunjukkan perilaku applikasi terhadap data berdasarkan hasil diatas dapat dijadikan sebagai cara untuk deteksi malware dengan menggunakan metode *behavior based detection approach,* tingginya perbedaan data antara malware dan applikasi 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 applikasi berhubungan dengan data user maka sistem apapun yang dibangun untuk deteksi malware menggunakan metode pendekatan *behavior* dapat mendeteksi perilakuyang tidak normal tersebut sebagai sebuah malware.

### 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 peneliain 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.

### Kesimpulan

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 menunjukan poin keberhasil yang dimaksud.

# BAB 5 Kesimpulan dan Saran

## Kesimpulan

Setelah melalui semua tahapan dalam penelitian dari persiapan, analisis, perancangan, dan implementasi serta pengujian dan hasil maka sampai pada bagian kesimpulan, kesimpulan yang dapat diambil adalah :

1. Penerapan metode *behavior approach* untuk deteksi malware berdasarkan perilaku applikasi terhadap data dilakukan dengan melewati beberapa tahapan yaitu, tahapan persiapan analisi yaitu mempersiapkan semua hal yang dibutuhkan untuk analisis mencakup komputerdan *tools* yang akan digunakan, tahapan kedua dinamik analisis yaitu proses mengamati malware dengan cara menjalankan malware dan applikasi secara langsung pada mesin virtual untuk mendapatkan informasi *behavior*, selanjutnya membuat rancangan sistem deteksi malware berdasarkan informasi *behavior* yang didapat dari analisis, kemudian tahapan implementasi rancangan sistem deteksi kedalam bahasa pemrograman python.
2. Berdasarkan hasil pengujian terhadap sistem deteksi malware berdasarkan perilaku applikasi terhadap data meggunakan metode *behavior approach* menunjukan bahwa penggunaan metode *behavior approach* sebagai metode deteksi malware memperlihatkan indikasi keberhasilan, dengan indikasi keberhasilan berupa tingginya jumlah perbedaan angka *behavior* antara *behavior* normal dan *behavior* yang tidak normal setelah dilakukan pengujian berulang kali terhadap sistem deteksi malware, tingginya 2 (dua) nilai *behavior* tersebut dapat digunakan untuk menentukan sebuah applikasi itu malware atau bukan malware.

## Saran

Berdasarkan hasil penelitian ini terdapat beberapa hal yang dapat dikembangkan, hal ini menjadi masukan saran untuk pengembangan penelitian selanjutnya, yaitu :

1. Memperluas dengan menambah objek pembahasan khususnya pada *behavior operations,* yaitu pembahasan pada topik registry *operation* dan *network operation* dalam lingkungan metode *behavior based detection*.
2. Menggunakan metode *signature based detection*  sebagai teknik tambahan untuk deteksi malware dalam salah satu tahapan untuk deteksi malware berbasi *behavior.*

# Daftar Pustaka

Alqurashi, S., & Batarfi, O. (2016). A Comparison of Malware Detection Techniques Based on Hidden Markov Model. *Journal of Information Security*, *7*(April), 215–223. https://doi.org/10.4236/jis.2016.73017

AV-TEST Institute, W. av-test. or. (2017). Malware Statistics and Trends Report | AV-TEST. Retrieved September 13, 2017, from https://www.av-test.org/en/statistics/malware/

Berlin, K., Slater, D., & Saxe, J. (2015). Malicious Behavior Detection using Windows Audit Logs. In *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security - AISec ’15* (pp. 35–44). https://doi.org/10.1145/2808769.2808773

Dalakov, G. (2012). First computer virus of Bob Thomas. Retrieved from http://history-computer.com/Internet/Maturing/Thomas.html

Damodaran, A. (2015). Combining Dynamic and Static Analysis for Malware Detection.

Deloitte. (2016). Ransomware Holding Your Data. *Deloitte Threat Intelligence and Analytics*. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/us/Documents/risk/us-aers-ransomware.pdf

Deylami, H. M., Muniyandi, R. C., Ardekani, I. T., & Sarrafzadeh, A. (2016). Taxonomy of Malware Detection Techniques: A Systematic Literature Review. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)* (pp. 629–636). https://doi.org/10.1109/PST.2016.7906998

Kheir, N. (2013). Behavioral classification and detection of malware through HTTP user agent anomalies. *Journal of Information Security and Applications*, *18*(1), 2–13. https://doi.org/10.1016/j.jisa.2013.07.006

Krebs, B. (2014). A Short History of Computer Viruses and Attacks. *Washington Post*, *2014*-*02*–*14*. Retrieved from http://www.securityfocus.com/news/2445

Lab Kapersky, W. kaspersky. co. (2016). KSN Report: Mobile Ransomware in 2014 - 2016. *Securelist*, (June). Retrieved from https://securelist.com/analysis/publications/75183/ksn-report-mobile-ransomware-in-2014-2016/

Liu, W., Ren, P., Liu, K., & Duan, H. (2011). Behavior-Based Malware Analysis and Detection. *2011 First International Workshop on Complexity and Data Mining*, (60203044), 39–42. https://doi.org/10.1109/IWCDM.2011.17

McAfee Labs. (2016). McAfee Labs Threats Report, (December).

Microsoft. (2017). Microsoft Security Intelligence Report, *21*, 7–8. Retrieved from https://www.microsoft.com/security/sir/default.aspx

Milošević, N. (2013). History of malware. *arXiv Preprint arXiv:1302.5392*, 1–11. Retrieved from http://arxiv.org/ftp/arxiv/papers/1302/1302.5392.pdf%5Cnhttp://adsabs.harvard.edu/abs/2013arXiv1302.5392M

Mujumdar, A., Masiwal, G., & Meshram, B. B. (2013). Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches. *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, *2*(6), 6–8.

Preda, M. D., Christodorescu, M., Jha, S., & Debray, S. (2008). A semantics-based approach to malware detection. *ACM Transactions on Programming Languages and Systems*, *30*(5), 1–54. https://doi.org/10.1145/1387673.1387674

Siddiqui, M. A. (2008). Data Mining Methods For Malware Detection.

Souri, A., & Hosseini, R. (2018). A state-of-the-art survey of malware detection approaches using data mining techniques. *Human-Centric Computing and Information Sciences*, *8*(1), 3. https://doi.org/10.1186/s13673-018-0125-x

Veeramani, R., & Rai, N. (2012). Windows API based Malware Detection and Framework Analysis. *… Conference on Networks and Cyber Security*, *3*(3), 1–6. Retrieved from http://www.academia.edu/download/30183099/Conference\_Proceedings\_book\_with\_links.pdf#page=45

Vysottsky, V. (2014). Virus timeline, 114–118.

Wressnegger, C., Freeman, K., Yamaguchi, F., & Rieck, K. (2016). From Malware Signatures to Anti-Virus Assisted Attacks. *CoRR*, *abs/1610.0*.

Yusirwan, S., Prayudi, Y., & Riadi, I. (2015). Implementation of Malware Analysis using Static and Dynamic Analysis Method. *International Journal Of Computer Applications*, *117*(6), 11–15. https://doi.org/10.5120/20557-2943