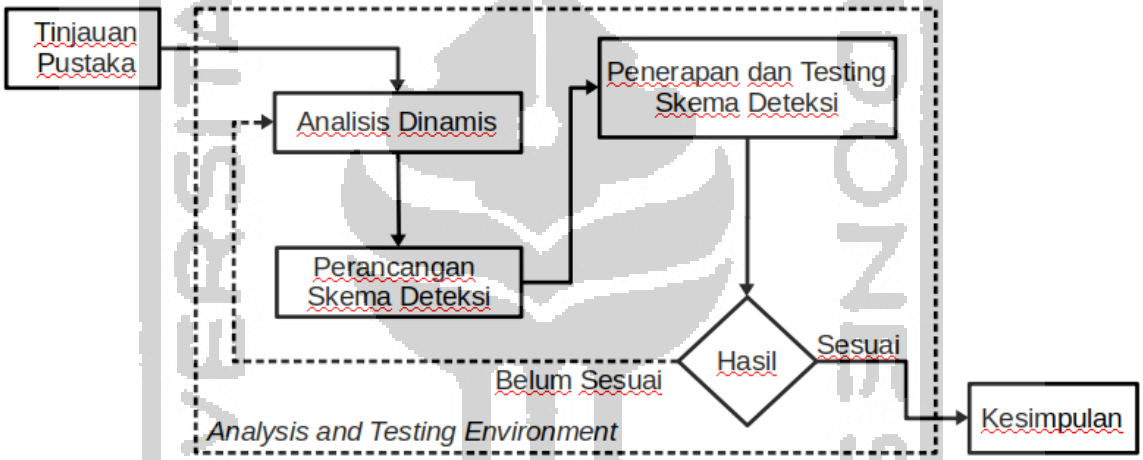
BAB 3
Metodologi Penelitian

3.1 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

3.2 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 aplikasi terhadap data, sehingga dapat dijadikan bahan-bahan untuk menunjang tujuan dari penelitian ini.

3.3 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 dijadikan sampel dalam penelitian ini serta beberapa aplikasi yang berhubungan dengan akses data. Tahapan kedua ini dibagi menjadi tiga sub-tahapan yaitu tahapan malware dan beberapa aplikasi 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.

3.3.1 Malware dan Aplikasi 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 aplikasi yang terkait dengan akses dan pengolahan data diantaranya beberapa aplikasi pengolah data yaitu Ms.Office (word, excel, PowerPoint), LibreOffice, beberapa aplikasi pengolah gambar yaitu ACDSee, serta beberapa aplikasi bawaan sistem dan aplikasi lainya yang dianggap perlu akan disertakan dalam analisis ini.

3.3.2 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

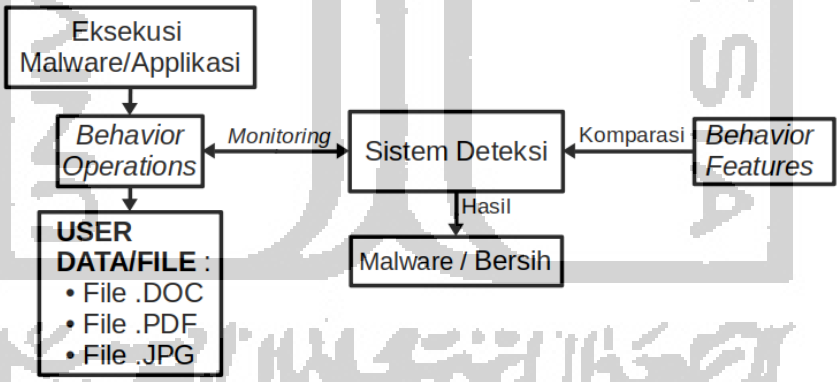
VirtualBox, Cuckoo Sandbox, Proces Explorer, Process Hacker, Process Monitor dan modul python *WinAppDbg*.

3.3.3 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 aplikasi saat dilakukan dinamik analisis. Fokus proses *extract behavior features* dari perilaku 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.

3.4 Perancangan Skema Deteksi

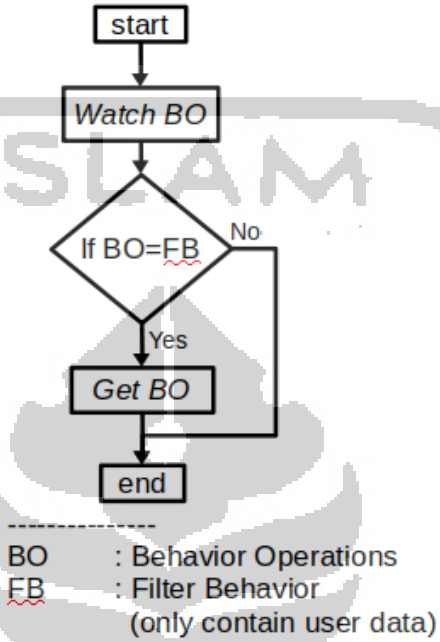
Perancangan skema deteksi merupakan tahapan untuk merancang skema untuk deteksi malware, rancangan dibuat berdasarkan pada pengetahuan yang didapat dari tahapan sebelumnya yaitu tahapan analisis. Seletelah tahapan analisis dan *behavior features* telah ditetapkan maka dibuat rancangan sistem deteksi 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 aplikasi, pada tahapan ini malware yang telah ditetapkan sebagai sampel akan dieksekusi. Tahapan berukutnya adalah sistem deteksi melakukan monitoring detail *behavior* dari aplikasi 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 aplikasi yang diperiksa termasuk kategori malware ataukan aplikasi 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

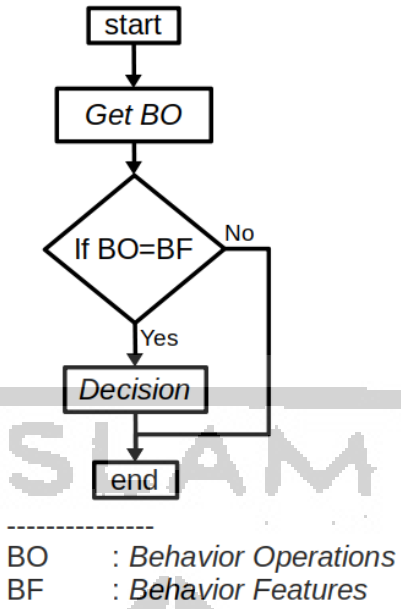
a. *Monitoring*

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

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.

b. 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.

3.5 Implementasi dan Testing Skema Deteksi

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

3.5.1 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.

3.5.2 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.

3.6 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.

Beberapa parameter keberhasilan juga diukur dari data yang didapatkan yaitu data *actions* berupa poin-poin yang ekstrak dan dijadikan sebagai standar behaviour pembanding. Hal lain yang menjadi ukuran untuk menlihat keberhasilan adalah melihat data persentase dari *true positive* (angka keberhasilan mendeteksi malware), *false positive* (angka kesalahan mendeteksi malware), *true negative* (angka keberhasilan mendeteksi aplikasi normal), dan *false negative* (angka kesalahan mendeteksi aplikasi normal).

3.7 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 aplikasi terhadap data.