BAB II
LANDASAN TEORI

Virus komputer bekerja dengan cara menempel pada suatu file komputer yang biasanya berupa file *executable*, *trojan* bekerja dengan cara melakukan *social engineering files* berbahaya dengan menampilkannya seperti files yang terlihat tidak berbahaya, *spyware* adalah perangkat lunak yang disisipi kode untuk mendapatkan informasi penting dari pengguna seperti akun bank, *password*, dan informasi lainnya yang diinginkan oleh pembuatnya, sedangkan *worm* adalah perangkat lunak jahat yang dibuat dengan memanfaatkan celah lubang keamanan pada sistem operasi untuk tujuan tertentu (Budhisantosa, 2014).

Penelitian ini adalah sebuah upaya untuk memberikan gambaran solusi dalam penanganan atas serangan *malware* dengan cara melakukan analisis terhadap *malware* yang telah berhasil melakukan serangan walaupun analisis dinamis memungkinkan sistem terinfeksi *malware*, tetapi analisis tersebut perlu dilakukan agar perilaku *malware* dapat dipantau secara langsung. Mesin Virtual dipilih sebagai *tools* bantuan yang digunakan untuk proses uji coba dengan analisis dinamis, sehingga meminimalisir adanya infeksi *malware* pada sistem yang asli. Selain itu, *Reverse Engineering* perlu digunakan untuk ekstraksi data informasi yang ada di dalam *malware* sehingga dapat diketahui bagaimana *malware* tersebut bekerja dan membuat celah dan melakukan serangan kedalam sistem.

2.1 Penelitian Terkait

Penelitian sebelumnya berfungsi untuk analisa serta membedakan dengan penelitian yang sedang dilakukan, penelitian ini berhubungan dengan identifikasi *Malware Hummingbad* dan memaparkan alur penelitian yang digambarkan dalam diagram *fishbone* dan *roadmap* penelitian.

Hasil-hasil penelitian yang dilakukan oleh peneliti lain dapat juga dimasukkan sebagai pembanding dari hasil yang akan dicobakan disini (Hasibuan, 2007).

2.2 Literatur Review

Berdasarkan topik permasalahan yang telah dirumuskan maka dibuatlah literature review dari jurnal penelitian sebelumnya yang berkaitan dengan bagaimana cara melakukan analisis malware. Berikut merupakan Literature reviews dari penelitian sebelumnya yang bersangkutan pada Malware ditunjukan pada lampiran 1 table 2.1.

Tabel 2.1 merupakan penelitian yang telah dilakukan sebelumnya mengenai analisis malware. Penelitian yang telah dilakukan sebagai dasar mengenai penelitian ini diantaranya berjudul ”Penggunaan teknik Reverse Engineering pada malware analisis untuk indentifikasi serangan malware” (Nugroho & Prayudi, 2015) ini melakukan proses Reverse Engineering pada malware Biscuit. Hal mendasar dari cara kerja malware tersebut adalah adanya auto request untuk koneksi ke ip tertentu yaitu ip pada alamat: 114.101.115.115. Selanjutnya proses

Reverse Engineering melalui penulusuran perintah: bdkzt, ckzjqk, download, exe, exit dan lists telah dapat memetakan bagaimana cara kerja dari malware Biscuit.

Penelitian yang berjudul "A Metodology of malware Analysis, tools, and Technique for windows platform – RAT analysis" dengan (Zalavadiya & Priyanka, 2017) ini melakukan analisis statis dan analisis dinamis pada malware DrakComet. Hasil dari penelitian ini adalah menguraikan metodologi yang efektif dan efisien yang dapat diterapkan untuk meningkatkan kinerja deteksi dan penghapusan malware yang dikumpulkan. Analisis dinamis cara terbaik untuk melakkan analisis sample malware.

Penelitian yang berjudul "Implementation of Malware Analysis using Static and Dynamic Analysis Method" dengan penulis Syarif Yusirwan S Yudi Prayudi Imam Riadi pada tahun 2015 Penelitian ini, penggabungan dari dua metode analisis malware yaitu analisis statis dan analisis dinamis mampu memberikan gambaran yang lebih lengkap tentang karakteristik dari malware TT.exe. Malware TT.exe adalah malware tipe trojan, dibuat pada hari Rabu 30 Juli 2014, menargetkan windows 7 dan windows 8. Pada awalnya ketika malware TT.exe aktif, malware akan menjalankan beberapa proses pada korban komputer seperti menyalin dirinya sendiri ke lokasi % AppData \ Roaming% dan menghapus malware asli. Selain itu, malware juga membuat beberapa registry yang membuat malware TT.exe dijalankan di sartup. Malware TT.exe sudah menginfeksi sistem komputer, malware akan menggunakan banyak memori komputer untuk menjalankan program serta menginfeksi program lain yang berjalan di komputer

korban. Malware TT.exe juga mematikan sebagian besar sistem keamanan windows, seperti windows defender, firewall, system restore, serta menghubungi server malware di alamat alhanexchange.com. Malware TT.exe juga membuat jalan bagi peretas untuk mendapatkan akses ke sistem komputer, dengan membuka port 313436. Proses infeksi dari malware TT.exe.

2.3 Tabel Literatur Review

Tabel 2.1 merupakan hasil dari *study literature* yang telah dilakukan. Tujuh belas penelitian yang telah dilakukan menjelaskan mengenai bagaimana melakukan analisis *malware*. Dua dari tujuh belas penelitian analisis *malware* pada tabel 2.1 mendekati dengan penelitian "**Analisis Malware "Hummingbad" Pada Perangkat Smartphone Menggunakan Metode Hybrid"** ditunjukkan pada tabel 2.2.

Tabel 2.2 Penelitian Terdekat

No	Penelitian / Tahun	Judul	Metode	State of The Art
1.	Alessandro Bacci, Alberto Bartoli, Fabio Martinelli, Eric Medvet, Francesco Mercaldo / 2017	*Impact of Code Obfuscation on Android Malware Detection based on Static and Dynamic Analysis*	• Metode Analisis Dinamis • Metode Analisis Statis	• Metode deteksi *malware* berbasis analisis statis pada dasarnya tidak efektif pada malware yang dikaburkan dibandingkan dengan metode dinamis.

Tabel 2.2 Penelitian Terdekat (Lanjutan)

2.	Syarif Yusirwan S Yudi Prayudi Imam Riadi (2015)	Implementation of Malware Analysis using Static and Dynamic Analysis Method	• Metode Analisis Statis • Metode Analisis Dinamis	• Penelitian ini, penggabungan dari dua metode analisis malware yaitu analisis statis dan analisis dinamis mampu memberikan gambaran yang lebih lengkap tentang karakteristik dari malware TT.exe.
3.	Heru Ari Nugroho Yudi Prayudi / 2015	Penggunaan teknik *Reverse Engineering* pada *malware* analisis untuk indentifikasi serangan *malware*	• Reverse Engineering	• Hasil yang didapat dari proses *Reverse Engineering* pada *malware* biscuit adalah gambaran bagaimana cara kerja dari *malware* tersebut.
4.	Fitri Amalia Febriadiny / 2019	Analisis *Malware "HummingBad" Pada* Perangkat *Smartphone* Menggunakan Metode Hybrid	• Metode Analisis Dinamis • Reverse Engineering	• Melakukan identifikasi perilaku dan aktivitas malware dengan menggunakan metode analisis dinamis & reverse engineering.

Tabel 2.2 merupakan tabel penelitian terdekat yang telah dilakukan sebelumnya yang berhubungan dengan analisis *malware*. Penelitian yang telah dilakukan sebagai dasar penelitian terdekat ini diantaranya berjudul "Impact of Code Obfuscation on Android Malware Detection based on Static and Dynamic Analysis" yang dilakukan oleh Bacci dkk (2018) menerangkan bahwasannya metode deteksi malware berbasis analisis statis pada dasarnya tidak efektif pada malware yang dikaburkan dibandingkan dengan metode dinamis.

Tabel 2.3 Tabel Matrik Penelitian

No	Peneliti/Tahun	Judul	Ruang Lingkup Penelitian														
			Metode / Teknik Analisis / Pendekatan												Implementasi, Dan Lain-Lain		
			Reverse Engineering	Dinamic	Static	Real Time Ponsel Bot *Miner*	Random Forest *Algorithm*	Analisis *Java* Bytecode	Teknik Physhing	Signatured based	Anomaly Based Techniques	Tasted IoT	Classifer Bernoulli Naïve Bayes	Teknik Komputer Forensik	Android	Internet	Sistem Operasi
1.	Junyang Qiu, Wei Luo, Lei Pan, Yonghang Tai, Zun Zhang, Yan Xiang / 2016	*Predicting the Impact of Android Malicious Samples via Machine Learning*	√												√		
2.	Sabam Chandra Yohanes Hutauruk, Fazmah Arif Yulianto, Gandeva Bayu Satrya / 2016	*Malware Analiysis On Windows Operating System To Detect Trojan*		√	√												√

Tabel 2.3 Tabel Matrik Penelitian (lanjutan)

3.	Triawan Adi Cahyanto, Victor Wahanggara, Darmawan Ramadana / 2017	Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Analisis Dinamis dan *Malware* Analisis Statis		√	√											√	√
4.	Iroshan Abherathe, Camila Walgampaya / 2010	*Smart Mobile Bot Detection Through behavioral Analysis*				√									√		
5.	Mansour Ahmadi, Angelo Sutgio, Giorgio Giancinto / 2017	*Towarad the Feasibility of Building Intelligent Anti- Malware on Android Devices*					√								√		

Tabel 2.3 Tabel Matrik Penelitian (lanjutan)

6.	Gerardo Canfora, Fabio Martinelli, Franscesco Mercaldo, Vitoria Nardone, Antonella Santone, Corrado Aaron Visagio / 2018	*Formal Tool For Identifying Mobile Maricious Behaviuor*						√							√		
7.	Rola Al Halaseh, Ja'far Alqatawna / 2016	*The Case of Phishing Attack*							√								√
8.	Imtithal A. Saeed, Ali Selamat, Ali M. A. Abuagoub / 2013	*A Survey On Malware and Malware Detection System*								√	√						√
9.	Kishor Krishnan Nair, Erick Dube,Samuel Lefophane / 2017	*Modelling an IoT Testbed in Context with the Security Vulnerabilities of South Africa*										√				√	

Tabel 2.3 Tabel Matrik Penelitian (lanjutan)

10.	Paolo Palumbo, Luiza Sayfullina, Dmitriy Komashinskiy, Emil Eirola, Juha Karhunen / 2017	*A Pragmatic Android Malware Detection Procedure*											√		√		
11.	Mohd Faizal Ab Razak, Fazidah Othman, Ahmad Firdaus, Firdaus Afifi, Rosli Salleh / 2017	*Feature Optimization and Malware Detection*			√												√
12.	Supraja Suresh, Fabio Di Troia, Katerina Potika, Mark Stamp / 2018	*An Analysis of Android Adware*		√	√										√		
13.	Rahmat Novrianda, Yesi Novaria Kunang, P.H. Shaksono / 2014	Analisis *Forensik Malware Pada Platform Andorid*												√	√		

Tabel 2.3 Tabel Matrik Penelitian (lanjutan)

14.	Saba Arshad, Abid Khan, Munam Ali Shah & Mansoor Ahmed / 2016	*Android Malware Detection & Protection: A Survey*		√	√										√		
15.	Heru Ari Nugroho & Yudi prayudi / 2015	Penggunaan teknik *Reverse Engineering* pada *malware* analisis untuk indentifikasi serangan *malware*	√														√
16.	Tesa Pajar Setia / 2018	Reverse Engineering Malware Ransomware Menggunakan Metode Analisis Dinamis	√	√													√
17.	Avie Triantoro / 2019	Analisis Malware hack.exe Dengan Metode Reverse Engineering dan Memory Forensic	√	√													√

Tabel 2.3 Tabel Matrik Penelitian (lanjutan)

18.	Fitri Amalia Febriandiny / 2019	Analisis *Malware "HummingBad"* Pada Perangkat *Smartphone* Menggunakan Metode Hybrid		√											√		

Tabel 2.3 menunjukkan matrik penelitian terkait mengenai *malware* analisis yang telah dilakukan sebelumnya. Keterbaruan dari penelitian dengan judul "**Analisis Malware "Hummingbad" Pada Perangkat Smartphone Menggunakan Metode Hybrid**" menggunakan metode analisis dinamis yang akan dilakukan pada perangkat *android.*

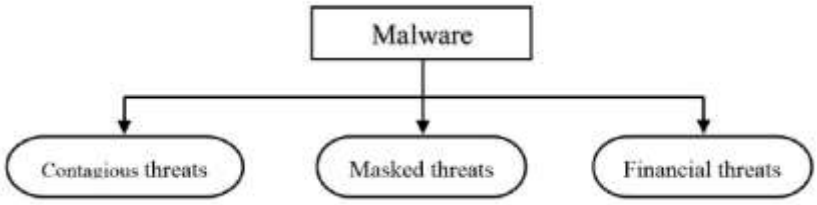
2.4 Teori Pendukung Penelitian

1.4.1 Malware

Malware adalah perangkat lunak berbahaya, yang diprogram untuk merusak atau untuk mendapatkan akses ke sistem komputer tanpa sepengetahuan pemilik sistem. *Virus*, *Worms*, Trojan, *Key logger* dan *Spyware* adalah contoh malware yang paling banyak digunakan. Istilah, seperti "*worm*", "*virus*", atau "*Trojan horse*" digunakan untuk klasifikasi *malware* yang menunjukkan perilaku jahat serupa. (Zalavadiya, 2017).

1.4.2 Klasifikasi *Malware*

Malware dapat diklasifikasikan dalam berbagai kelas dan kategori yang umumnya dikategorikan menurut proses dan reaksi yang dicapai pada sistem yang terinfeksi yang tergantung pada proses perancangan dan pengembangan program jahat. Gambar 2.4.2 berikut menunjukkan berbagai jenis *malware*. (Zalavadiya, 2017)



Gambar 2.4.2 Klasifikasi *Malware* (Zalavadiya, 2017)

a. *Contagious Threats* (Ancaman yang Menular)

Tabel 2.4 Deskripsi *Contagious Threats* (Zalavadiya, 2017)

Malware	Karakteristik	Cara Kerja	Kerusakan

Virus	*Malware* yang mengambil alih kontrol yang tidak sah dari komputer yang terinfeksi dan menyebabkan kerusakan tanpa sepengetahuan pengguna	Program virus yang tersembunyi dalam program tidak berbahaya lainnya seperti *file* yang dapat dieksekusi dan kemampuan mereplikasi dirinya ke dalam program lain dan menyebarkan infeksi dari satu komputer ke komputer lain	Penurunan kinerja dan menyebabka DOS (*Denial of Service*)
Worm	Worm adalah perangkat lunak berbahaya yang berdiri sendiri yang dapat beroprasi secara independen dan tidak mengaitkan dirinya untuk menyebarluaskan	Worm memanfaatkan kerentanan keamanan dengan menggunakan computer atau jaringan dan menyebarluaskan diri melalui perangkat penyimpanan seperti USB, media komunikasi perangkat seperti *Email*	Melakukan komunikasi sejumlah besar memori sumber daya sistem dan masalah kinerja jaringan

Tabel 2.4 menunjukan *Contagious Threats* (Ancaman yang Menular) yang didalamnya ada *virus* dan *worm. Malware virus dan worm* merupakan kategori pada ancaman yang menular, ini dilihat dari karakteristik dan cara kerja *malware* melakukan infeksi pada sistem.

b. *Masked Threats* (Ancaman Bertopeng)

Tabel 2.5 Deskripsi *Masked Threats* (Zalavadiya, 2017)

Malware	Karakteristik	Cara Kerja	Kerusakan
Trojan	*Malware* berbahaya yang tersembunyi dan berperilaku sebagai program yang sah untuk mengambil alih kendali computer atau sistem secara tidak sah	*Trojan* tidak mereplikasi diri sebagai gantinya mengunduh atau menyalin melalui interaksi pengguna seperti *download*	Mencuri kata sandi atau rincian login, pencuri uang digital, memodifikasi atau menghapus file, memonitor

		file dari internet atau perangkat lain	aktivitas pengguna
Backdoors	Melakukan bypass control keamanan normal dan memberikan peyerang ke akses yang tidak sah	Dipasang melalui program atau aktivitas berbahaya lainnya	Melakukan modifikasi dan menghapus *file* system dan memonitor aktivitas system
Adware	Memberikan informasi kepada pelaku iklan tentang kebiasaan penjelajahan pengguna, sehingga memungkinkan pelaku iklan untuk memberikan add yang ditargetkan	*Adware* menyebar melaui situs *web*	*Clickjacking*, *phising* atau membuat aktivitas jahat menggunakan *browser*

Tabel 2.5 Deskripsi *Masked Threats* (Zalavadiya, 2017) (lanjutan)

Malware	Karakteristik	Cara Kerja	Kerusakan
Rootkits	*Rootkits* adalah teknik *masking* untuk *malware* yang pada dasarnya dirancang untuk maksud jahat dari program ini	Dipasang melalui eksploitasi perangkat lunak atau *Trojan*	Mencuri kata sandi atau melakukan instal *keylogger*

Tabel 2.5 menunjukan *Masked Threats* (Ancaman Bertopeng). *Malware trojan, backdoor, adware,* dan *rootkits* merupakan kategori pada ancaman bertopeng, ini dilihat dari karakteristik dan cara kerja *malware* melakukan infeksi pada sistem.

c. *Financial Threats* (Ancaman Keuangan)

Tabel 2.6 Deskripsi *Financial Threats* (Zalavadiya, 2017)

Malware	Karakteristik	Cara Kerja	Kerusakan
Ransomware	*Ransomware* adalah perangkat lunak yang dirancang untuk memblokir akses ke sistem komputer	*Ransomware* menyebar dan disalurkan melalui rekayasa social dan interaksi pengguna,	*Ransomware* adalah *malware* untuk pencurian data melakukan enkripsi data

	hingga sejumlah uang dibayarkan	membuka lampiran *email* berbahya yang melakukan klik tautan berbahaya dalam *email* atau di situs jejaring sosial	korban dan membatasi pengguna untuk melakukan askses system penyerang
Spyware	*Spyware* melacak aktivitas pengguna tanpa sepengetahuannya pengguna dan mengirim kembali informasi sesintif kepada penyerang	Dipasang dengan perangkat lunak lain seperti *freeware* atau dijatuhkan oleh *Trojan*	*Sniffing* antarmuka jaringan sertifikasi digital, kunci enkripsi dan informasi sensitive lainnya

Tabel 2.6 Deskripsi *Financial Threats* (Zalavadiya, 2017) (lanjutan)

Malware	Karakteristik	Cara Kerja	Kerusakan
Keylogger	*Keylogger* diam-diam merekam *keystrokes*	Dipasang oleh program jahat lain atau ketika seorang pengguna mengujungi.	Menangkap infromasi *sensitive* seperti nama pengguna kata sandi nomor kartu kredit atau rincian perbankan *online*.

Tabel 2.6 menunjukan *Financial Threats* (Ancaman Keuangan). *Malware ransomware*, *spyware*, dan *keyloger* merupakan kategori pada ancaman keuangan, ini dilihat dari karakteristik dan cara kerja *malware* melakukan infeksi pada system.

1.4.3 Malware HummingBad

HummingBad pertama kali ditemukan pada bulan Februari dan diduga berasal dari China, *malware* ini termasuk canggih karena mampu menghindar dengan membangun semacam rootkit permanen. *Malware* tersebut, disebutkan secara otomatis akan menginstal *software* ke perangkat *android* yang terinfeksi, kemudian digunakan

untuk menghasilkan pendapatan secara ilegal bagi kalangan pelaku kriminal di dunia maya. Mayoritas korban *malware* tersebut disebutkan terbanyak berasal dari China dan India, dengan masing-masing negara mencapai lebih dari 1 juta kasus.

HummingBad berhasil meretas perangkat *android* yang memasang aplikasi pihak ketiga di luar *Google Play Store,* sehingga dengan mudah *malware* ini mampu mengambil alih ponsel dari jarak jauh.

Dijuluki *"HummingBad"* oleh para peneliti di perusahaan keamanan *Check Point,* itu adalah salah satu serangan terbesar hingga saat ini di *android* - sistem operasi seluler paling populer di dunia, yang berjalan di lebih dari 80% dari semua ponsel cerdas dan juga tablet. *Malware HummingBad* menginfeksi 10m perangkat *android,* Meskipun serangan ini bukan merupakan bencana besar, serangan ini membuka pintu bagi serangan di masa depan, kata para pakar keamanan. belum dapat mengatakan handset *android* mana yang paling rentan, tetapi mengatakan bahwa sebanyak 85m perangkat *android* dunia rentan.

1.4.4 Laboratorium

a. Virtual Machine

Menurut Gerard J. Popek dan Robert P. Goldberg pada tahun 1974 *virtual machine* adalah sebuah duplikat yang efisien dan terisolasi dari suatu mesin asli. *Virtual Machine* merupakan software yang digunakan untuk mensimulasikan lingkungan kerja suatu perangkat komputer secara *virtual*. *Virtual machine* dapat

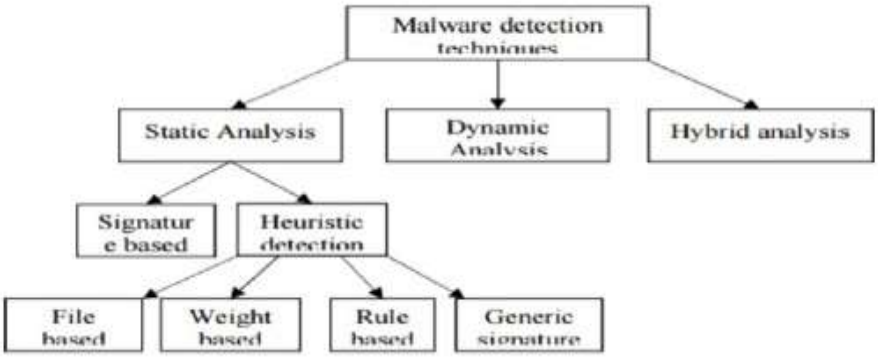
menghemat biaya pembelian *hardware*, karena dengan menggunakan satu komputer saja. Kelebihan lainnya adalah adanya fungsi ***snapshot*** dan konfigurasi jaringan ***host-only***. Komputer yang digunakan untuk melakuan Analisa malware sebaiknya tidak terhubung ke jaringan. Kekurangan penggunaan virtual mesin adalah kami membutuhkan RAM yang besar. Tahap selanjutnya membutuhkan system operasi (OS). Proses analisa minimal membutuhkan 2 sistem operasi. Satu sebagai smartphone korban, dan satu lagi sebagai *server* yang menyediakan beberapa layanan jaringan. Sistem operasi yang digunakan tergantung dari *malware* yang akan dianalisa. *Malware* dibuat untuk OS *Windows* maka gunakan OS *Windows* sebagai komputer korban, contohnya *Windows 7*.

b. Android

Android adalah sistem operasi yang berbasis *Linux* untuk telepon seluler seperti telepon pintar dan komputer tablet. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *Android Inc.*, pendatang baru yang membuat peranti lunak untuk ponsel, kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia* (Saudi, Mohd, & Basir,2017).

1.4.5 Malware Analisis

Malware analysis adalah kumpulan dari proses penentuan tujuan dan fungsionalitas dari *sample malware* yang diberikan seperti *virus, worm, trojan,* dan sebagainya untuk melakukan deteksi *malicious code.* Baik dengan mengeksekusi *malware* tersebut (*dynamic analysis)* ataupun dengan pemeriksaan kode program saat sebelum dieksekusi. (Hutauruk, 2016). Tujuan dari *malware analysis* yang dilakukan pada *malware* jenis *trojan* ini adalah untuk menghasilkan data karakteristik *trojan* dan menganalisis siklus hidup masing – masing jenis *trojan.* Melakukan *malware analysis* dilakukan dengan metode penelitian berupa studi literature berupa pendalaman materi yang berhubungan dengan *malware analysis.* (Hutauruk, 2016)



Gambar 2.4 Metode Malware Analysis

a. Analisis Malware Static

Malware statis analisis adalah proses menganalisis program dengan memeriksanya. Dikenal sebagai teknik analisis kode. Sebelum program benar-benar dijalankan, informasi statis ditemukan dalam *file* yang dapat dieksekusi termasuk data header dan urutan *byte* yang digunakan untuk menentukan

apakah *file* tertentu adalah *file* berbahaya atau tidak. (Jerlin, 2015).

Disassembly, menjadi salah satu teknik analisis statis, analisis statis dalam proses ini dilakukan dengan pembongkaran menggunakan alat-alat *disassemble* yang digunakan untuk mendapatkan *file* program bahasa *assembly*. *Opcode* diekstrak sebagai fitur untuk menganalisis perilaku aplikasi secara statis untuk mendeteksi *malware*. (Jerlin, 2015)

1) *Signature-based Detection*

Signature-based Detection merupakan deteksi berbasis *signature* mempertahankan catatan tanda tangan dan mengidentifikasi *malware* dengan membandingkan atau menyejajarkan pola terhadap database. Sebagian besar alat *antivirus* didasarkan pada teknik pendeteksian ini. Tanda tangan dibuat untuk memeriksa kode yang dibongkar dari biner perangkat lunak perusak. Kode ini kemudian dianalisis dan fitur-fiturnya diekstraksi. Fitur yang diekstraksi digunakan dalam membuat tanda tangan dari keluarga *malware* tertentu. (Jerlin, 2015).

Keuntungan utama dari teknik pendeteksian berbasis *signature* adalah bahwa dapat mendeteksi *malware* yang dikenal secara akurat sedangkan sumber daya yang lebih sedikit diperlukan untuk mengungkap *malware* dan terutama berfokus pada tanda tangan serangan sementara

kelemahan utamanya adalah karena tidak ada tanda tangan yang tersedia, tidak dapat mendeteksi kejadian baru yang tidak dikenal dari *malware*. (Jerlin, 2015)

2) *Heuristic-based Detection*

Heuristic-based Detection merupakan deteksi berbasis perilaku atau anomali. Tujuan utama dari teknik deteksi ini adalah untuk menganalisis perilaku *malware* selain diketahui atau tidak diketahui. Parameter perilaku mencakup berbagai faktor seperti jenis sumber atau tujuan lampiran, alamat *malware*, dan fitur statistik lainnya yang dapat dihitung. (Jerlin, 2015).

Biasanya terjadi pada fase pelatihan dan fase deteksi. Selama fase pelatihan perilaku sistem diamati dengan tidak adanya serangan dan teknik pembelajaran mesin digunakan untuk membuat profil perilaku normal tersebut. (Jerlin, 2015)

a) *File based heuristic analysis*

Analisis heuristik berdasarkan file juga dikenal sebagai analisis *file*. Teknik ini *file* dianalisis secara mendalam seperti isi, tujuan, pengerjaan *file*, jika *file* berisi perintah untuk menghapus atau merusak *file* lain, maka dianggap sebagai *file* berbahaya. (Uppal, 2014).

b) *Weight based heuristic analysis*

Analisis heuristik berdasarkan *weight* adalah teknik kuno. Aplikasi diberi bobot sesuai dengan bahaya yang mungkin dimilikinya, jika nilai tertimbang melebihi nilai ambang batas yang ditemukan, maka aplikasi tersebut berisi kode bahaya. (Uppal, 2014).

c) *Rule based heuristic analysis*

Analisis disini melakukan ekstrasi aturan yang mengidentifikasikan aplikasi. Aturan-aturan ini kemudian dicocokkan dengan aturan yang ditetapkan sebelumnya, jika aturan tidak cocok, maka aplikasi tersebut terdapat *malware.* (Uppal, 2014)

d) *Generic signature analysis*

Varian *malware* berarti, *malware* itu berbeda dalam perilakunya tetapi memiliki keluarga yang sama seperti "kembar identik". Teknik ini menggunakan definisi antivirus yang ditetapkan sebelumnya, untuk menemukan varian baru *malware.* (Uppal, 2014)

b. Analisis Malware Dinamis

Analisis *malware* dinamis dikenal sebagai analisis *file* yang terinfeksi selama pelaksanaannya. *File* yang terinfeksi dianalisis dalam lingkungan simulasi, sesuatu seperti mesin *virtual*. Menggunakan alat-alat tertentu seperti *SysAnalyzer*, *Process Explorer*, dan lain - lain. Mengidentifikasi perilaku umum *file* tertentu. Prosesnya, *file* terdeteksi setelah

mengeksekusinya di lingkungan yang sebenarnya dan selama pelaksanaan *file* interaksi sistemnya, perilaku dan efeknya pada sistem diamati (Jerlin, 2015).

Keuntungan dari analisis dinamis adalah bahwa secara akurat menganalisis *malware* yang dikenal maupun yang tidak dikenal, teknik analisis ini lebih memakan waktu. Ini membutuhkan waktu sebanyak untuk mempersiapkan lingkungan untuk analisis *malware* seperti lingkungan mesin *virtual* (Jerlin, 2015).

c. *Reverse Engineering*

Reverse Engineering atau dalam bahasa Indonesia Rekayasa Balik merupakan proses mencari dan menemukan prinsip kerja dari suatu produk teknologi yang ada pada sebuah sistem, perangkat atau objek, dengan menganalisis mendalam pada fungsi dan cara kerja dari sistem, perangkat atau objek yang diteliti (Utomo, Ismail, & Zani, 2018).

d. Analisis *Hybrid*

Teknik ini adalah kombinasi dari analisis statis dan analisis dinamis. Prosedur yang mengikutinya pertama kali memeriksa signature *malware*, jika ada dalam kode dan kemudian memonitor perilaku kode. Teknik ini menggabungkan keuntungan dari kedua teknik diatas. (Uppal, 2014).

2.4.6 Metodologi

a. Kuantitatif

Menurut Sugiyono, metode penelitian kuantitatif dapat diartikan sebagai metode penelitian yang berlandaskan pada filsafat positivisme, digunakan untuk meneliti pada populasi atau sampel tertentu. Teknik pengambilan sampel pada umumnya dilakukan secara random, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif atau statistik dengan tujuan untuk menguji hipotesis yang telah ditetapkan (Sugiyono, 2012: 7).

Metode penelitian kuantitatif salah satu jenis penelitian yang spesifikasinya adalah sistematis, terencana, dan terstruktur dengan jelas sejak awal hingga pembuatan desain penelitiannya. Definisi lain menyebutkan penelitian kuantitatif adalah penelitian yang banyak menuntut penggunaan angka, mulai dari pengumpulan data, penafsiran terhadap data tersebut, serta penampilan dari hasilnya. Tahap kesimpulan penelitian akan lebih baik bila disertai dengan gambar, tabel, grafik, atau tampilan lainnya. (Burhanuddin, 2013).

b. Kualitatif

Menurut Bogdan dan Taylor (1975) dalam buku Moleong (2004:3) mengemukakan metode kualitatif sebagai prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata atau lisan dari orang-orang dan perilaku yang dapat diamati. Metode penelitian kualitatif juga merupakan metode penelitian yang lebih menekankan pada aspek pemahaman secara

mendalam terhadap suatu masalah dari pada melihat permasalahan untuk penelitian generalisasi. Metode penelitian ini lebih suka menggunakan teknik analisis mendalam (*in-depth analysis*), yaitu mengkaji masalah secara kasus perkasus karena metodologi kualitatif yakin bahwa sifat suatu masalah satu akan berbeda dengan sifat dari masalah lainnya. (Burhanuddin, 2013).

2.4.7 Sistem Operasi

Sistem operasi adalah seperangkat program yang mengelola sumber daya perangkat keras komputer, dan menyediakan layanan umum untuk aplikasi perangkat lunak. Sistem operasi adalah jenis yang paling penting dari perangkat lunak sistem dalam sistem komputer, tanpa sistem operasi, pengguna tidak dapat menjalankan program aplikasi pada komputer yang dimiliki, kecuali program aplikasi *booting*. (Hariyanto, 2007).

Beberapa sistem operasi berukuran sangat besar dan kompleks, serta inputnya tergantung kepada input pengguna, sedangkan sistem operasi lainnya sangat kecil dan dibuat dengan asumsi bekerja tanpa intervensi manusia sama sekali. Tipe yang pertama sering disebut sebagai *Desktop OS*, sedangkan tipe kedua adalah *Real-Time OS*, contohnya adalah Windows, Linux, *Free BSD, Solaris, palm, symbian,* dan sebagainya. (Hariyanto, 2007).