BAB II

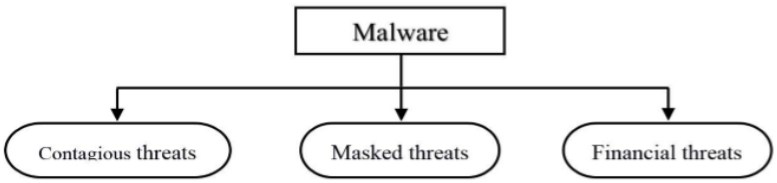
TINJAUAN PUSTAKA

2.1. *Malware*

Malware merupakan perangkat lunak yang bersifat berbahaya yang di program untuk mempengaruhi sebuah sistem baik merusak atau mengakuisisi data pada perangkat komputer tanda disadari oleh penggunanya. Jenis *malware* yang lazim digunakan adalah *Virus*, *Key Logger*, *Worm*, *Spyware*, *dan Ransomware* (Zalavadiya dan Sharma, 2017).

2.2. Klasifikasi *Malware*

Malware terbagi menjadi beberapa kategori yang dilihat dari cara kerja, efek pada suatu sistem, perkembangan *malware*, dan pengembangan lanjutan dari *malware* tersebut (Zalavadiya dan Sharma, 2017).



GAMBAR 2.1 KLASIFIKASI MALWARE (ZALAVADIYA DAN SHARMA, 2017)

1. Ancaman Menular (*Contagious Threat*)

   Virus dan *Worm* adalah ancaman yang dapat menular dikarenakan cara kerjanya dengan melakukan infeksi pada sistem yang dapat dilihat pada tabel 2.1.

TABEL 2.1 ANCAMAN MENULAR KLASIFIKASI (ZALAVADIYA DAN SHARMA, 2017)

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Virus	*Malware* bekerja dengan cara menginfeksi komputer dan mengambil alih kendali yang tidak sah yang menyebabkan kerusakan tanpa diketahui oleh pengguna	*Virus* dapat berkamuflase pada program yang nampak tidak berbahaya lainnya seperti *file* yang dapat di eksekusi dan mereplikasi dirinya ke dalam program lain serta dapat menyebarkan infeksi ke komputer lainnya	Penurunan kinerja sistem dan menyebabkan DoS (*Denial of Service)*
Worm	*Worm* adalah perangkat yang dapat beroperasi sendiri dan tidak mengaitkan dirinya untuk menyebarluaskan	*Worm* menggunakan kerentanan keamanan sistem komputer atau melalui jaringan dan dapat menyebarkan diri melalui media penyimpanan seperti USB, media komunikasi perangkat seperti *Email*	Melakukan komunikasi sejumlah besar memori sumber daya sistem dan masalah kinerja jaringan

2. Ancaman Bertopeng (*Masker Threat*)

Malware yang masuk dalam kategori ini antara lain *Trojan, Backdoor, Adware, dan Rootkits*. Karakteristik *malware* ini dengan melakukan infeksi pada sebuah sistem yang dapat ditunjukan pada tabel 2.2.

TABEL 2.2 ANCAMAN BERTOPENG KLASIFIKASI (ZALAVADIYA DAN SHARMA, 2017)

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**

Trojan	*Malware* yang tersembunyi dan berperilaku layaknya program yang sah untuk mengambil alih kendali komputer atau sistem secara tidak sah	*Trojan* tidak mereplikasi diri sebagai gantinya mengunduh atau menyalin melalui interaksi pengguna seperti *download file* dari internet atau perangkat lain	Mencuri kata sandi atau rincian *login*, pencuri uang digital, memodifikasi atau menghapus *file*, memonitor aktivitas pengguna
Backdoors	Melakukan *bypass* kendali keamanan normal dan memberikan akses kepada penyerang	Terpasang melalui program atau aktivitas berbahaya lainnya	Dapat modifikasi dan menghapus file system serta memonitor aktivitas system
Adware	Memberikan informasi yang biasa dilakukan oleh pengguna seperti riwayat penjelajahan pengguna, sehingga memungkinkan pelaku iklan untuk memberikan iklan yang ditargetkan	*Adware* menyebar melalui media seperti situs *web*	*Clickjacking*, *phising* atau membuat aktivitas jahat menggunakan *browser*
Rootkits	*Rootkits* adalah teknik *masking* untuk *malware* yang dirancang untuk maksud jahat dari program ini	Dipasang melalui eksploitasi pada perangkat lunak atau *Trojan*	Mencuri kata sandi atau melakukan mengisntal *keylogger*

3. Ancaman Keuangan (*Financial Threats*)

Malware yang termasuk dalam kategori ini antara lain *Ransomware, Spyware*, dan *Keylogger*. Karakteristik *malware* ini dengan melakukan infeksi pada sistem yang dapat dilihat pada tabel 2.3.

TABEL 2.3 ANCAMAN KEUANGAN KLASIFIKASI (ZALAVADIYA DAN SHARMA, 2017)

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Ransomware	*Ransomware* adalah perangkat lunak yang	Menyebar melalui	*Malware* yang

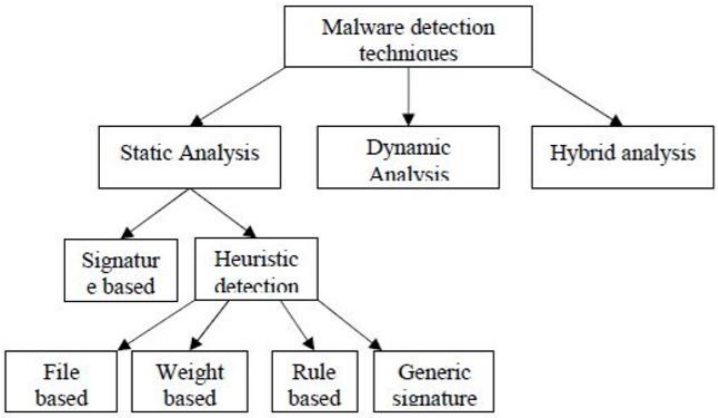
	dirancang untuk melakukan blokir akses ke sistem komputer hingga sejumlah uang dibayarkan untuk membuka akses kembali	rekayasa sosial dan interaksi pengguna, membuka lampiran email berbahaya yang melakukan klik tautan berbahaya dalam email atau di situs jejaring sosial	digunkan untuk pencurian data melakukan enkripsi data korban dan membatasi pengguna untuk melakukan askses sistem penyerang
Spyware	*Spyware* adalah perangkat lunak yang melacak aktivitas pengguna tanpa sepengetahuannya pengguna dan mengirim kembali informasi sesintif kepada penyerang	Terpasang pada perangkat lunak lain seperti *freeware* atau dijatuhkan oleh *Trojan*	*Sniffing* antarmuka jaringan sertifikasi digital, *Encrytion Key* dan informasi sensitif lainnya
Keylogger	*Keylogger* adalah perangkat lunak yang merekam *keystrokes* tanpa diketahui oleh pengguna	Terpasang program jahat lain atau ketika seorang pengguna mengunjungi	Merekam informasi sensitif seperti nama pengguna kata sandi nomor kartu kredit atau rincian perbankan *online*

2.3. *Malware AQUVAPRN.exe*

AQUVAPRN.exe merupakan *malware* yang cara kerjanya mengambil data pribadi milik pengguna baik kredensial maupun data penting lainnya dengan cara menempel pada *file* yang tidak terlihat mencurigakan. *Malware* ini menempel pada file yang memiliki ekstensi *.exe*.

2.4. Analisis *Malware* (*Malware Analysis*)

Malware Analysis merupakan langkah pertama untuk mengidentifikasi bagaimana serangan *malware* sebagai langkah penanggulangan serangan pada sistem. Analisis ini memiliki tujuan yaitu mengidentifikasi bagaimana sebuah *malware* bekerja (Adenansi dan Novarina, 2017).



GAMBAR 2.2 TEKNIK DETEKSI MALWARE (ADENANSI DAN NOVARINA, 2017)

Gambar 2.2 ada tiga teknik dalam melakukan *malware analysis* yaitu:

1. Analisis Statis (*Static Analysis*)

   Analisis ini dilakukan tanpa menjalankan perangkat lunak atau *malware*. Cara kerja analisis ini dengan memecah *malware* tersebut menggunakan alat rekayasa, lalu disusun atau dibangun kembali menjadi sebuah perangkat lunak. Teknik ini juga dipecah menjadi beberapa bagian, antara lain (Uppal dkk, 2014):

a. *Signature Detection* (Pendeteksian Ciri Khas)

Signature Detection dikenal dengan teknik pencocokan pada pola atau *string* pola yaitu sebuah urutan program yang di injeksikan kepada sebuah aplikasi oleh pembuat *malware* tersebut. Teknik ini dilakukan untuk mencari detektor ciri khas dalam kode penyusun sebuah *malware* (Uppal *dkk*, 2014).

b. *Heuristic Detection* (Pendeteksian Heuristis)

Heuristic Detection mirip seperti *Signature Detection*. Cara kerjanya dengan mendeteksi perintah yang tidak ada dalam kode aplikasi penyusun *malware* tersebut. Teknik ini dapat mendeteksi *malware* baru yang belum ditemukan sebelumnya. *Heuristic Detection* dipecah menjadi beberapa cara, yaitu (Uppal dkk, 2014):

1) *File Based Heuristic Analysis*

Analisa ini dilakukan berdasarkan *file* yang dicurigai sebagai *malware*. Cara kerja analisis ini dengan menginvestigasi tujuan dari tujuan, isi, pengerjaaan *file* tersebut. Jika ditemukan perintah yang mencurigakan seperti menghapus atau merusak sebuah *file*, maka *file* tersebut dapat dikatakan sebagai *malware* (Uppal dkk, 2014).

2) *Weight Based Heuristic Analysis*

Analisa ini dilakukan dengan melihat bobot bahaya. Aplikasi yang dicurigai sebagai *malware* diberi bobot bahaya yang mungkin dimilikinya. Jika nilai yang ditunjukan melebihi batas yang ditemukan, maka bisa dikatakan aplikasi tersebut berisi kode yang dapat merusak sistem (Uppal dkk, 2014).

3) *Rule Based Heuristic Analysis*

Analisa ini dilakukan dengan mengekstraksi aturan yang menjadi identifikasi sebuah aplikasi. Aturan ini dicocokkan dengan aturan yang sudah ditetapkan sebelumnya. Apabila tidak menemui kecocokan, aplikasi tersebut berisi *malware* yang dapat merusak sistem (Uppal dkk, 2014).

4) *Generic Signature Analysis*

Analisa ini dilakukan dengan menggunakan definisi antivirus yang ditetapkan untuk mendapatkan varian baru sebuah *malware*. Arti varian disini yaitu *malware* tersebut memiliki perbedaan dalam perilakunya namun masih berkaitan dengan *malware* sebelumnya seperti “kembar identik” (Uppal dkk, 2014).

Analisis statis memiliki keuntungan yaitu cepat dan aman serta dapat mengumpulkan struktur kode program dalam pelaksanaannya. Analisis ini juga dapat melihat perilaku dan cara kerja sebuah *malware* yang berguna sebagai antisipasi keamanan dimasa depan (Uppal dkk, 2014).

2. Analisis Dinamis (*Dynamic Analysis*)

Analisis ini dilakukan dengan mengeksekusi *malware* tersebut sehingga dapat menginsvetigasi cara kerja *malware* tersebut melihat dari fungsi panggilan, pelacakan informasi, analisa parameter, dan instruksi saat *malware* tersebut berjalan. Analisis ini biasanya digunakan dalam ruang lingkup *virtual* dalam penggunaanya. Aplikasi yang berjalan tidak normal atau sesuai dengan bentuk aplikasi pada umumnya dapat dikategorikan sebagai *malware* (Uppal dkk, 2014).

3. Analisi Hybrid (*Hybrid Analysis*)

   Analisis ini dilakukan dengan mengkombinasikan analisis statis dan analisis dinamis. Cara kerja analisis ini dengan menggabungkan dua metode dalam menginvestigasi sebuah aplikasi yang diduga sebagai *malware*. Keutungan dari analisis ini adalah informasi yang didapat tentang *malware* tersebut menjadi lebih lengkap dan akurat (Uppal dkk, 2014).

2.5. *Reverse Engineering*

Reverse Engineering adalah pengekstraksian sebuah pengetahuan atau cetak biru dari apapun yang dibuat oleh manusia. Konsep ini sudah ada sebelum adanya komputer, yang dimana digunakan dalam bidang industri untuk mengetahui informasi sebuah produk atau ciptaan agar mengetahui informasi yang lebih banyak terkait produk tersebut untuk tujuan pengembangan lebih lanjut (Nugroho dan Prayudi, 2014). *Reverse Engineering* dapat dilakukan dengan langkah-langkah berikut:

1. *Assembly*

   Bahasa *assembly* digunakan pada mikro prosesor yang telah diprogram menggunakan bahasa pemrograman tingkat rendah (Megira dkk, 2018).

2. *Disassembly*

   Disassembly adalah proses pembongkaran yang bertujuan untuk mengubah bahasa *assembly* menjadi kode mesin (Yusirwan dkk, 2015).

3. *Debugging*

Sebuah metode yang digunakan untuk mengurangi *bug*, mencari *bug*, dan mengisolasi masalah. Cara ini digunakan untuk pengujian dari *malware* (Almarri dan Sant, 2014).

4. X86 *Architecture*

Sebuah desain kompleks dari set komputer. X86 *Architecture* ini kebanyakan menggunakan X86 *Architecture Von Neumann* (Nugroho dan Prayudi, 2014).

5. *Instruction*

Instruction adalah sebuah konstruksi yang dibuat oleh sebuah program. Kontruksi x86 itu sendiri terdiri dari *nemonic* dan nol atau lebih *operands* (Megira dkk, 2018).

6. *Hashing*

Metode *Hashing* digunakan sebagai bentuk verifikasi sebelum ataupun sesudah dilakukannya proses *malware analysis*. Verifikasi ini digunakan sebagai indikator adanya perubahan pada *malware* yang telah diinvestigasi (Megira dkk, 2018).

7. *String Analysis*

String adalah nilai yang dimuat dari *malware* yang sedang diinvestigasi dalam sebuah program. *Reverse Engineering* harus menggunakan metode ini untuk mendapatkan bukti yang akurat dari sampel *malware* yang diinvestigasi (Megira dkk, 2018).

2.6. *Memory Forensics*

Memory Forensics merupakan salah satu metode *malware analysis* canggih, akarkit, dan dapat juga digunakan untuk mendeteksi kejahatan dunia maya. Metode ini berguna dalam menganalisis *malware* karena mudah digunakan terlepas dari bentuk sistem operasi, perangkat lunak, dan *file* sistem yang ada (Rathnayaka dan Jamdagni, 2017). Mengidentifikasi kemungkinan serangan *cyber* menggunakan *malware* dapat menggunakan *tools* volatilitas (Bahtiar dkk, 2018).

2.7. *State Of The Art*

State of the art akan menjawab pertanyaan yang berhubungan pada *malware analysis*. Penelitian mengenai *malware analysis*, *reverse engineering*, dan *memory engineering* disajikan pada tabel 2.4 *state of the art*.

Tabel 2.4 *State Of The Art*

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
1.	Investigasi Serangan Malware Njrat Pada PC	Devi Rizky Septani Nur Widiyasono Husni Mubarok (2016)	Metode Analisis Dinamis	Mengetahui cara kerja *malware* Njrat.
2.	Metode Klasifikasi dan Analisis Karakteristik Malware Menggunakan Konsep Ontologi	Abdul Haris Muhammad Bambang Sugiantoro Ahmad Lutfi (2017)	Metode Analisis Statis dan Analisis Dinamis	Penerapan ontologi sebagai *knowledge base* dasar dalam melakukan analisis karakteristik *malware* sebagai *knowledge base* sangat dibutuhkan dalam melakukan analisis karakteristik *malware*.
3.	*Malware Analysis* Pada *Windows* Operating System Untuk Mendeteksi Trojan	Sabam Chandra Yohanes Hutauruk Fazmah Arif Yulianto Gandeva Bayu Satrya (2016)	Metode Analisis Statis dan Analisis Dinamis	Data karakteristik *trojan*, yang dapat digunakan sebagai indikator untuk menganalisa *trojan* berdasarkan *behavior*-nya.
4.	*Malware Dynamic*	Retno Adenansi Lia A. Novarina (2017)	Metode Analisis Dinamis	Membahas mengenai cara melakukan analisis malware dengan metode analisis dinamis untuk deteksi *malware*.

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
5.	Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Analisis Dinamis dan Malware Analisis Statis	Triawan Adi Cahyanto Victor Wahanggara Darmawan Ramadana (2017)	Metode Analisis Statis dan Analisis Dinamis	Tentang cara kerja *malware* (*poison ivy*), dapat melakukan proses *login* secara *remote* tanpa diketahui oleh pemilik komputer.
6.	Penggunaan Teknik *Reverse Engineering* pada *Malware* Analisis untuk Indentifikasi Serangan *Malware*	Heru Ari Nugroho Yudi prayudi (2015)	*Reverse Engineering*	Hasil yang didapat dari proses *Reverse Engineering* pada *malware biscuit* adalah gambaran bagaimana cara kerja dari *malware* tersebut.
7.	*Malware Analysis*	Ujaliben Kalpesh Bavishi Bhavesh Madnlal jain (2017)	Metode analisis statis dan analisis dinamis	Menjelaskan mengenai teknik dan tools untuk analisis dinamis dan analisis statis.
8.	*A Metodology of malware Analysis, tools, and Technique for Windows platform – RAT analysis*	Nayan zalavadiya Priyanka Sharman (2017)	Metode analisis statis dan analisis dinamis	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*.

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
9.	*Basic on Malware Analysis, tools and Technique*	Dolly Uppal Vishakha Mehra Vinod verma (2014)	Metode analisis statis dan analisis dinamis	Berfokus pada studi dasar malware dan berbagai deteksi teknik yang dapat digunakan untuk deteksi *malware*
10.	*Implementation of Malware Analysis using Static and Dynamic Analysis Method*	Syarif Yusirwan S Yudi Prayudi Imam Riadi	Metode analisis statis dan analisis dinamis	Berdasarkan 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*.
11.	Penggunaan teknik Reverse Engineering pada malware analisis untuk indentifikasi serangan malware Flawed Ammyy RAT	Tesa Pajar Setia1 Nur Widiyasono Aldy Putra Aldya (2018)	*Reverse Enginering, Malware Ransomware*, Menggunakan Analisis Dinamis	Hasil dari proses identifikasi *malware* Flawed Ammyy RAT menggunakan metode dinamis dan *reverse engineering* memberikan gambaran cara kerja *malware* Flawed Ammy RAT.
12.	Penggunaan Teknik *Reverse Engineering* Pada *Malware* Analisis Untuk Indentifikasi Serangan *Malware* Hack.*exe*	Avie Triantoro Nur Widiyasono Rohmat Gunawan (2018)	Reverse Enginering, Malware, Ransomware,	Hasil dari proses identifikasi *malware* Hack.*exe* dengan metode dinamis dan *reverse engineering* memberikan gambaran cara kerja *malware* Hack.*exe*.

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
			Menggunakan Analisis Dinamis	
13.	Memory Volatile Forensik Untuk Deteksi Malware Menggunakan Algoritma Machine Learning	Fikri Bahtiar Nur Widiyasono Aldy Putra Aldya (2018)	Metode Analisis Statis dan Analisis Dinamis	Hasil dari penelitian ini didapatkan bahwa dari ke-5 algoritma tersebut, algoritma Random Forest yang unggul dalam pengklasifikasian malware menggunakan dataset. Memberikan manfaat dan bantuan untuk penyelidik forensik dalam menganalisis memori *volatile* dan mendeteksi malware secara *offline* (tidak terhubung ke internet) yang mungkin ada pada memory *volatile*.

2.8. Matriks Penelitian

TABEL 2.5 MATRIKS PENELITIAN

No.	Judul	Ruang ligkup penelitian					Penulis
		Analisis			*Reverse Enginering*	*Memory Forensic*	
		Statis	Dinamis	*Hybrid*			
1.	Investigasi Serangan *Malware* Njrat Pada PC		✔				Devi Rizky Septani
2.	Metode Klasifikasi dan Analisis Karakteristik *Malware* Menggunakan Konsep Ontologi			✔			Abdul Haris Muhammad
3.	*Malware* Analysis Pada *Windows Operating System* Untuk Mendeteksi *Trojan*		✔				Sabam Chandra Yohanes
4.	*Malware Dynamic*			✔	✔		Retno Adenansi
5.	Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Analisis Dinamis dan *Malware*			✔			Triawan Adi Cahyanto
6.	Analisis Statis Penggunaan Teknik *Reverse Engineering* pada *Malware* Analisis Untuk Indentifikasi Serangan *Malware*		✔				Heru Ari Nugroho
7.	*A Metodology of malware Analysis, tools, and Technique for Windows platform – RAT analysis*			✔			Nayan zalavadiya

No.	Judul	Ruang ligkup penelitian					Penulis
		Analisis			*Reverse Enginering*	*Memory Forensic*	
		Statis	Dinamis	*Hybrid*			
8.	*Malware analysis*			✔			Ujaliben Kalpesh Bavishi
9.	*Basic on Malware Analysis, tools and Technique*			✔			Dolly Uppal
10.	*Implementation of Malware Analysis using Static and Dynamic Analysis Method*			✔			Syarif Yusirwan S
11.	Analisis Malware Flawed Ammyy RAT dengan Metode *Reverse Engineering*		✔		✔		Tesa Pajar Setia
12.	Analisis *Malware* hack.*exe* dengan Metode *Reverse Engineering* dan *Memory Forensic*			✔	✔	✔	Avie Triantoro
13.	**Penelitian Usulan**			✔	✔	✔	Hafish Naufal Aditya

Berdasarkan tabel 2.5 pada penelitian ini terdapat persamaan metode seperti *dynamic method* dan *reverse engineering*. Keterbaharuan yang diambil dalam penelitian ini adalah dengan metode *memory forensics* untuk investigasi *malware AQUVAPRN.exe*.