BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 *Malware*

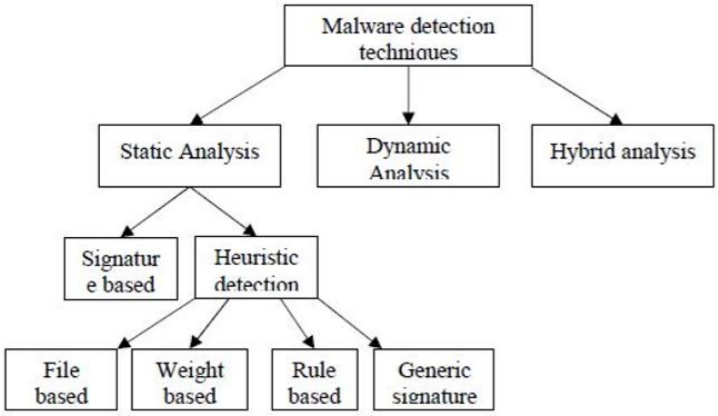
Malware (Malicious Software) merupakan program yang dirancang untuk disusupkan ke dalam sebuah sistem dengan tujuan melakukan beraneka ragam aktivitas yang bersifat merugikan pemiliknya. Merugikan dalam arti kata dampak negatif yang ditimbulkan dapat berkisar mulai dari sekedar memperlambat kinerja sistem hingga merusak bahkan menghancurkan data penting yang tersimpan dalam sistem yang dimaksud. Ada tiga jenis *malware* klasik yang paling banyak ditemui, yaitu: *Virus, Worms*, dan *Trojan Horse* (Manoppo et al., 2020).

Ransomware termasuk salah satu bagian dari *malware*, berikut merupakan beberapa perbedaan antara *malware* biasa dengan *ransomware.* Perbedaan keduanya, jika *malware* aktivitasnya cenderung bersembunyi dan tidak mau menampakkan diri, sedangkan *ransomware* lebih menampakkan aktivitasnya sebagai *virus*. Hal itu menyebabkan pengguna dapat mengetahui apabila dirinya sedang terinfeksi oleh *ransomware* (Wahidin et al., 2022).

Serangan *ransomware WannaCry* adalah serangan dunia maya global yang dimulai pada 12 Mei 2017, dan skalanya belum pernah terjadi sebelumnya dengan cepat memengaruhi lebih dari 200.000 komputer di lebih dari 150 negara. Secara umum, sindikat kejahatan *transnasional* "mengadaptasi model bisnis dengan menggunakan apa yang disebut *'ransomware'* untuk mendapatkan kendali atas jaringan komputer dan kemudian meminta pembayaran sebagai imbalan

pemulihan.” *Virus WannaCry* mengeksploitasi kerentanan di *Microsoft Windows* yang awalnya dikembangkan oleh Badan Keamanan Nasional AS dan beroperasi dengan mengenkripsi data korban dan menuntut pembayaran tebusan sebagai imbalan pemulihan data (Wahidin et al., 2022).

2.1.2 *Analysis Static*



Gambar 2.1 Deteksi *Malware*(Adenansi, 2017).

Gambar 2.1 merupakan struktur teknik deteksi *malware,* metode deteksi analisis *malware* dibagi menjadi 3 bagian, yaitu analisis statis, dinamis, dan *hybrid*. Metode analisis statis adalah analisa yang dilakukan dengan cara mengawasi secara langsung isi dari *source code* dengan menggunakan beberapa aplikasi *unpacker* tanpa melakukan eksekusi *malware* secara langsung. Melihat dan mengamati bagian *source code malware* dapat menggunakan beberapa jenis program seperti program analisa, *debugger*, dan *disassembler*. Keuntungan dalam analisis statis yaitu data menjadi aman dan analisis juga cenderung cepat. Metode analisis dinamis adalah metode yang mengamati aktivitas pada *malware* dengan cara melakukan eksekusi secara langsung, sehingga tampak cara kerja atau perilaku *malware*

sebelum dan sesudah melakukan infeksi. Metode ini menggunakan mesin *virtual*, sehingga dapat meminimalisir adanya penyebaran *malware*. Keuntungan analisis dinamis adalah mudah mendeteksi *malware* yang sedang melakukan proses infeksi tentang cara kerja *malware* tersebut secara langsung. Analisis *hybrid* merupakan metode kombinasi atau gabungan dari analisis statis dengan analisis dinamis. Metode ini menggabungkan keunggulan dari analisis dinamis dan statis yaitu dengan melakukan pengecekan setiap *signature malware* jika ditemukan adanya kode tertentu di saat pemeriksaan dan monitoring perilaku kode pada *malware* tersebut (Wahidin et al., 2022).

2.1.3 *Analysis Dinamic*

Proses analisis dinamis dilakukan dengan sebuah file yang diperiksa akan diaktifkan dalam sebuah lingkungan yang *safe* baik pada sebuah mesin fisik yang telah disediakan sebagai laboratorium *malware* maupun yang berupa *virtual* (mesin *virtual*) untuk selanjutnya mampu dikumpulkan informasi mengenai dampaknya terhadap komputer ketika *file malware* menjalankan prosesnya. Hal itu dapat diketahui kegiatan apa saja yang dilakukan oleh *malware* saat berhasil menginfeksi sebuah komputer. Tahapan dalam analisis dinamis ini akan memeriksa komputer dengan secara keseluruhan seperti proses yang berjalan di komputer, perubahan *registry*, komunikasi internet dan peristiwa janggal lainnya yang memungkinkan terjadi ketika sebuah komputer telah terinfeksi oleh *malware* (Cahyanto et al., 2017)(Manoppo et al., 2020).

2.1.4 *Analysis Hybrid*

Metode *hybrid analysis* adalah sebuah penggabungan dari teknik analisa statis dan teknik analisa dinamis dengan memeriksa *source code* yang diduga sebagai *malware* kemudian melihat perilaku dari *malware* tersebut setelah menginfeksi sistem. Analisa *hybrid* ini dilakukan untuk menutupi kekurangan dari kedua teknik tersebut (Tansen & Nurdiarto, 2020).

2.1.5 *Signature-based Detection*

Signature-based detection adalah teknik deteksi yang berdasarkan pattern *matching, string, mask*, atau teknik *fingerprinting*. *Signature* adalah teknik persamaan bit yang disuntikkan dalam program aplikasi oleh *attacker*, yang secara unik mengidentifikasi jenis *malware* tertentu. Proses mendeteksi *malware* dalam kode, *detektor malware* akan melakukan *signature* yang telah ditentukan di dalam kode tersebut. Meskipun *signature-based* detection sangat efisien untuk *malware* yang telah dikenali, akan tetapi mempunyai kelemahan yaitu tidak dapat mendeteksi jenis *malware* yang tidak dikenal atau baru. Kekurangan basis data *signature* yang terbatas, sebagian besar *malware* tetap tidak akan terdeteksi. Jadi varian *malware* harus diperbaharui saat akan dideteksi untuk bisa mendapatkan *signature* dari basis datanya (Nugraha et al., 2019) (Alviana & Sumitra, 2018).

2.1.6 *Reverse Engineering*

Reverse Engineering merupakan metode melakukan *decompiled file Execute* menjadi *file source code*. Penggunaan metode ini sangat cocok jika

diterapkan dalam menganalisa file yang terindikasi *malware*. Salah satu *tools* yang digunakan dalam melakukan *Android Reverse Engineering* adalah *Jadx. Tools* tersebut mengubah file *APK Android* menjadi *Folder Source code*, *Folder Resources* dan *APK Signature*. Module Jadx ada dua macam, yaitu *Jadx,* dijalankan lewat *Command Line (CLi)* sedangkan *Jadx-gui*, berbasis *Graphical User Interface* (GUI) (Putra Wijaya & Santoso, 2021)(Agus et al., 2021)(Hazri, 2020).

2.2 *State of The Art* (SOTA) Penelitian

Tabel 2.1 merupakan tabel *State of The Art* (SOTA) Penelitian yang sudah dilakukan oleh peneliti lain terkait *malware analysis.*

Tabel 2.1 State of The Art (SOTA) Penelitian

No.	Penulis	Judul	Metode	State of The Art
1	(Fatwa et al., 2022)	Analisis *Malware Ahmyth* pada *Platform* Android Menggunakan Metode *Reverse Engineering*	✓ *Reverse Engineering*	✓ *Malware ahmyth* akan menjalankan *service*nya setelah perangkat melakukan *restart* dan menunggu perintah dari C&C server untuk melakukan tindakan tertentu pada perangkat yang terinfeksi.
2	(Apriyani, 2019)	Analisis Dan *Reverse Engineering* Pada *Malware* Zeus.	✓ *Dynamic Analysis* ✓ *Reverse Engineering*	✓ Menunjukan hasil proses dari identifikasi *malware* zeus dengan metode *Dynamic Analysis* adalah mendapatkan *username* dan *password* yang diinputkan oleh korban. ✓ Percobaan yang dilakukan dengan metode *reverse engineering* menghasilkan struktur dari *malware* zeus secara detail dari sistem operasi *malware* zeus hingga interaksi antara *user* dan program.
3	(Setia et al., 2019)	*Reverse Engineering* Untuk Analisi *Malware* FLAWED AMMY RAT	✓ *Reverse Engineering* ✓ *Dynamic Analysis*	✓ Menunjukan bahwa *malware* Flawed ammyy RAT bekerja dengan bersembunyi pada aplikasi Ammyy Admin kemudian melakukan koneksi dengan *attacker* dengan ip address 103.208.86.69. net*name ip address* 103.208.86.69 adalah *zappie host.*
4	(Bavishi & Jain, 2018)	*Malware analysis*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Menjelaskan mengenai Teknik dan *tools* untuk *Dynamic Analysis* dan *Static Analysis*

No.	Penulis	Judul	Metode	State of The Art
5	Bacci, A., et.al. (2018)	*Impact of Code Obfuscation on Android Malware Detection Based on Static and Dynamic Analysis*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Membandingkan *Static Analysis* dan dinamis terhadap *malware* yang terobfuskasi ✓ Mengkombinasikan *machine learning*
6	(Qbeitah & Aldwairi, 2018)	*Dynamic Malware Analysis of Phishing Emails*	✓ *Dynamic Analysis*	✓ Menganalisis 173 *email Phishing* baru dan 45 pesan SPIM untuk mencari *malware* yang berpotensi baru, menyajikan dua sampel *malware* dan *Dynamic Analysis* yang komprehensif.
7	(Adenansi & Novarina, 2017)	*Malware Dynamic*	✓ *Dynamic Analysis*	✓ Membahas mengenai cara melakukan analisis *malware* dengan metode *Dynamic Analysis* untuk deteksi *malware*
8	(Zalavadiya & Sharma, 2017)	*A Metodology of malware Analysis, tools, and Technique for windows platform – RAT analysis*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Menguraikan metodologi yang efektif dan efisien yang dapat diterapkan untuk meningkatkan kinerja deteksi dan penghapusan *malware* yang dikumpulkan. *Dynamic Analysis* cara terbaik untuk melakukan analisis *sample malware*
9	(Cahyanto et al., 2017)	Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Dynamic Analysis dan *Malware* Static Analysis	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Tentang cara kerja *malware* (poison ivy), dapat melakukan proses *login* secara *remote* tanpa diketahui oleh pemilik komputer.

No.	Penulis	Judul	Metode	State of The Art
10	(Haris Muhammad et al., 2017)	Metode Klasifikasi dan Analisis Karakteristik *Malware* Menggunakan Konsep Ontologi	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Penerapan ontologi sebagai *knowledge base* dasar dalam Melakukan analisis karakteristik *malware* sebagai *knowledge base* sangat dibutuhkan dalam melakukan analisis karakteristik *malware*
11	(Chandra et al., 2016)	*Malware* Analysis Pada *Windows Operating System* Untuk Mendeteksi *Trojan*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Data karakteristik trojan, yang dapat digunakan sebagai indikator untuk menganalisa trojan berdasarkan behaviornya
12	(Septiani et al., 2016)	Investigasi Serangan *Malware Njrat* Pada PC	✓ *Dynamic Analysis*	✓ Mengetahui cara kerja *malware Njrat*
13	Berlin, K., Slater, D., & Saxe, J. (2015)	*Malicious Behavior Detection using Windows Audit Logs*	✓ *Dynamic Analysis* ✓ *Behavior Based*	✓ Melakukan pencegahan dengan pendekatan *behavior based detection* dengan memanfaatkan *audit logs* yang menjadi standard *build-in* pada sistem operasi Windows.
14	(Yusirwan et al., 2015)	*Implementation of Malware Analysis using Static and Dynamic Analysis Method*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Berdasarkan penelitian ini, penggabungan dari dua metode analisis *malware* yaitu *Static Analysis* dan *Dynamic Analysis* mampu memberikan gambaran yang lebih lengkap tentang karakteristik dari *malware* TT.exe.
15	(Nugroho & Prayudi, 2015)	Penggunaan teknik *Reverse Engineering* pada *malware* analisis untuk identifikasi serangan *malware*	✓ *Reverse Engineering*	✓ Hasil yang didapat dari proses *Reverse Engineering* pada *malware* biscuit adalah gambaran bagaimana cara kerja dari *malware* tersebut.
16	(Uppal et al., 2014)	*Basic on Malware Analysis, tools and Technique*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Berfokus pada studi dasar *malware* dan berbagai deteksi teknik yang dapat digunakan untuk deteksi *malware*

No.	Penulis	Judul	Metode	State of The Art
17	Gadhiya et.al. (2013)	*Techniques of Malware Analysis*	✓ *Static Analysis* ✓ *Dynamic Analysis*	✓ Komparasi *Static Analysis* dan dinamis dari segi keunggulan, kelemahan dan *tools* yang digunakan
18	(Liu et al., 2011)	*Behavior-Based Malware Analysis and Detection*	✓ *Dynamic Analysis* ✓ *Behavior Based*	✓ Melakukan investigasi terhadap teknik untuk melakukan ekstrak *malware behavior feature* dengan melakukan klasifikasi dari formal *behavior* dan merancang sistem untuk mendeteksi *malware* jenis baru.
19	(Binsalleeh et al., 2010)	*On the Analysis of the Zeus Botnet Crimeware Toolkit*	✓ *Reverse Engineering*	✓ Menyajikan analisis rekayasa balik terperinci dari perangkat *crimeware* Zeus untuk mengungkap arsitektur yang mendasarinya dan memungkinkan mitigasinya. Merancang alat untuk melakukan pemulihan kunci enkripsi dan ekstraksi informasi konfigurasi dari *executable bot biner*.

2.3 Matriks Penelitian

Tabel 2.2 merupakan tabel matriks penelitian terkait *malware analysis* yang bersumber dari jurnal nasional, jurnal internasional dan sumber lain.

Tabel 2.2 Matriks peneltian

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
1	(Yusirwan et al., 2015)	*Implementation of Malware Analysis using Static and Dynamic Analysis Method*	✓	✓	✓										✓	
2	(Uppal et al., 2014)	*Basic on Malware Analysis, tools and Technique*	✓	✓	✓										✓	
3	(Bavishi & Jain, 2018)	*Malware analysis*	✓	✓	✓										✓	
4	(Zalavadiya & Sharma, 2017)	*A Methodology of malware Analysis, tools, and Technique for windows platform – RAT analysis*	✓	✓											✓	
5	(Nugroho & Prayudi, 2015)	Penggunaan teknik *Reverse Engineering* pada *malware* analisis		✓		✓									✓	

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
		Untuk identifikasi serangan *malware*														
6	(Cahyanto et al., 2017)	Analisis dan Deteksi *Malware* Menggunakan Metode *Malware* Dynamic Analysis dan *Malware* Static Analysis	✓	✓	✓										✓	
7	(Adenansi & Novarina, 2017)	*Malware Dynamic*		✓											✓	
8	(Chandra et al., 2016)	*Malware* Analysis Pada *Windows Operating System* Untuk Mendeteksi *Trojan*	✓	✓	✓											
9	(Haris Muhammad et al., 2017)	Metode Klasifikasi dan Analisis Karakteristik *Malware* Menggunakan Konsep Ontologi	✓	✓											✓	
10	(Fatwa et al., 2022)	Analisis *Malware Ahmyth* pada *Platform*Android Menggunakan Metode *Reverse Engineering*				✓									✓	
11	(Septiani et al., 2016)	Investigasi Serangan *Malware Njrat* Pada PC		✓											✓	

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
12	(Binsalleeh et al., 2010)	*On the Analysis of the Zeus Botnet Crimeware Toolkit*				✓						✓				
13	(Qbeitah & Aldwairi, 2018)	*Dynamic Malware Analysis of Phishing Emails*	✓	✓	✓										✓	
14	(Setia et al., 2019)	*Reverse Engineering* Untuk Analisi *Malware* FLAWED AMMY RAT	✓	✓	✓	✓									✓	
15	(Apriyani, 2019)	Analisis Dan *Reverse Engineering* Pada *Malware* Zeus.	✓	✓	✓	✓									✓	
16	(Liu et al., 2011)	*Behavior-Based Malware Analysis and Detection*		✓					✓						✓	
17	Bacci, A., et.al. (2018)	*Impact of Code Obfuscation on Android Malware Detection Based on Static and Dynamic Analysis*	✓	✓	✓						✓					
18	Berlin, K., Slater, D., & Saxe, J. (2015)	*Malicious Behavior Detection using Windows Audit Logs*		✓					✓						✓	

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
19	Gadhiya et.al. (2013)	*Techniques of Malware Analysis*	✓	✓	✓										✓	
20	Alazab, Layton, Venkatara man, & Watters, (2010)	*Malware Detection Based on Structural and Behavioural Features of API Calls*							✓						✓	
21	Veeramani & Rai, (2012)	*Windows API based Malware Detection and Framework Analysis*							✓						✓	
22	Nizar Kheir, (2013)	*Behavioral classification and detection of malware through HTTP user agent anomalies*		✓					✓						✓	
23	Siddiqui, (2008)	*Data Mining Methods For Malware Detection*							✓						✓	
24	Almarri (2014)	*Optimized Malware Detection in Digital Forensics*	✓	✓	✓										✓	
25	Singhal & Raul, (2012)	*Malware Detection Module using Machine Learning Algorithms to Assist in Centralized Security in Enterprise Networks*		✓			✓		✓						✓	

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	*Dynamic*	*Hybrid*	*Reverse Engineering*	*Machine Learning*	*Signature-based*	*Behavior Based*	*Heuristic Based*	*Android*	*IoT*	*ML*	*Cloud*	*Windows*	*Linux*
26	Lindorfer (2015)	*MARVIN: Efficient and Comprehensive Mobile App Classification Through Static and Dynamic Analysis*	✓	✓	✓						✓					
27	MK Alzaylaee (2017)	*EMULATOR vs REAL PHONE: Android Malware Detection Using Machine Learning*		✓			✓				✓					
28	Junyang Qiu et.al. (2016)	*Predicting the Impact of Android Malicious Samples via Machine Learning*					✓				✓					
29	Mansour Ahmadi et.al. (2017)	*Toward the Feasibility of Building Intelligent Anti-Malware on Android Devices*		✓							✓					
30	Mohd Faizal Ab Razak et.al. (2017)	*Bio-inspired for Features Optimization and Malware Detection*		✓											✓	
31	Supraja Suresh et.al. (2018)	*An Analysis of Android Malware*	✓	✓	✓						✓					

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
32	Avie Triantoro (2020)	*Hack.exe Malware Analysis and Investigation Using Memory Forensics*	✓	✓	✓	✓									✓	
33	(Shatnawi et al., 2022)	*An Android Malware Detection Approach Based on Static Feature Analysis Using Machine Learning Algorithms*	✓			✓					✓					
34	Raman Dugyala et.al. (2022)	*Analysis of Malware Detection and Signature Generation Using a Novel Hybrid Approach*			✓			✓							✓	
35	Fahd Alhaidari et.al (2022)	*ZeVigilante: Detecting Zero-Day Malware Using Machine Learning and Sandboxing Analysis Techniques*		✓			✓								✓	
36	(Widiyasono et al., 2021)	Analisis Pola Dan Dampak Serangan *Cryptojacking* Dengan Menggunakan Pendekatan *Dynamic Analysis*		✓											✓	

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
37	(Rusdi et al., 2019)	Analisis Infeksi *Malware* Pada Perangkat Android Dengan Metode *Hybrid Analysis*			✓						✓					
38	Jashanpreet Singh Srawa, Keshav Kumar (2021)	*Using Static and Dynamic Malware features to perform Malware Ascription*	✓	✓	✓										✓	
39	Arwa Abdulkarim Al Alsadi et.al. (2022)	*No Spring Chicken: Quantifying the Lifespan of Exploits in IoT Malware Using Static and Dynamic Analysis*	✓	✓	✓										✓	
40	Sun Qirui et.al. (2022)	*MLxPack: Investigating the Effects of Packers on ML-based Malware Detection Systems Using Static and Dynamic Traits*	✓	✓	✓										✓	
41	(Bazrafshan et al., 2013)	*A Survey On Heuristic Malware Detection Techniques*						✓	✓	✓						

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
42	(Yu et al., 2018)	*A Survey Of Malware Behavior Description And Analysis*							✓							
43	(Gupta et al., 2016)	*Malware Characterization Using Windows API Call Sequences*							✓							
44	(Mujumdar et al., 2013)	*Analysis of Signature-Based and Behavior-Based Anti-Malware Approaches*						✓	✓							
45	(Tahir, 2018)	*A Study on Malware and Malware Detection Techniques*						✓	✓							
46	(Zolkipli & Jantan, 2010)	*Malware Behavior Analysis: Learning and Understanding Current Malware Threats*						✓	✓							
47	(M.P.M et al., 2018)	Analisis Perbandingan Deteksi *Trojan* Menggunakan Metode *Static-Dynamic* dengan Metode *Behaviour* Pada Sistem Operasi *Windows*	✓	✓	✓				✓							

No.	Penulis	Judul	Ruang Lingkup Penelitian													
			Metode								Implementasi				OS	
			Static	***Dynamic***	***Hybrid***	***Reverse Engineering***	***Machine Learning***	***Signature-based***	***Behavior Based***	***Heuristic Based***	***Android***	***IoT***	***ML***	***Cloud***	***Windows***	***Linux***
48	Zeni Zaenil Waro (Usulan Penelitian 2023)	Analisis *Malware WannaCry*Menggunakan Metode,*Reverse Engineering, dan Signature-based Detection*	✓	✓	✓	✓		✓							✓	

2.4 *Gap Research*

Tabel 2.3 Gap Research/Terdekat

No.	Penulis	Judul	Metode	*State of The Art / Gap Analysis*
1	(Julian Dwi Nugraha, 2019)	Analisis *Malware* berdasarkan *api call memory* dengan metode *deteksi signature-based*	✓ *Static Analysis* ✓ *Signature-based*	Penentuan identifikasi *malware* dilakukan dengan mencari API call memory dan hasil *signature*. Berdasarkan data yang diperoleh, hasil API call memory dengan hasil *signature* saling terkait. Hasil *signature* dihasilkan dari beberapa API call memory yang dijalankan oleh *malware*. Hasil API call memory dan hasil *signature* didapatkan dengan menggunakan tools static analysis yang digunakan selama pengujian berlangsung.

Berdasarkan *Gap Research* 2.4 hasil analisis mengenai *signature* masih memperoleh gap terkait kode atu pola terkait *malware*nya *WannaCry,* analisis yang ingin diselesaikan dalam penelitian ini adalah menerapkan analisis *malicious malware WannaCry* menggunakan metode *hybrid* dengan Teknik *reverse engineering* dan *signature-based detection* agar menghasilkan sebuah *signature* pola *byte* dan kode instruksi analisis yang berguna dimasa mendatang.

Model yang akan dikembangkan ditargetkan dapat menterjemahkan kebutuhan untuk menangkap ragam tantangan global, serta dapat mengetahui suatu *virus* yang terdeteksi dengan model, metode serta teknik yang digunakan. Keterbaruan dan orisinalitas yang diharapkan dari penelitian ini yaitu analisis yang dilakukan dapat mengikuti *trend* sebuah teknologi dalam menyajikan informasi setiap tahunnya, agar keterbaruan sistem selalu terjaga dalam mengikuti *trend* penyajian informasi.