BAB II

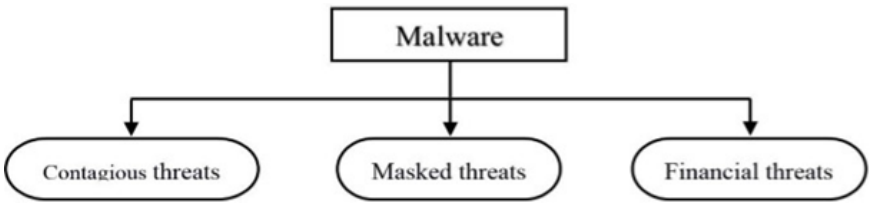
TINJAUAN PUSTAKA

2.1 *Malware*

Malware (malicious software) merupakan perangkat lunak berbahaya yang dibuat untuk merusak, mengganggu, atau mengakses sistem komputer tanpa izin. Tujuannya bisa mencuri data, merusak sistem, atau mengambil alih perangkat secara diam-diam (Anugerah dkk., 2024). *Malware* menyebar melalui email mencurigakan, situs palsu, unduhan ilegal, atau celah keamanan. Jenis-jenisnya meliputi *virus, worm, trojan, ransomware, rootkit, spyware,* dan *fileless malware,* masing-masing memiliki ciri khas karakteristik dan cara kerja yang berbeda dalam menyerang sistem (Rudd dkk., 2017).

2.2 Klasifikasi *Malware*

Malware dapat dibagi ke dalam berbagai kelas dan kategori, umumnya dikelompokkan berdasarkan cara penyebaran serta dampak yang ditimbulkan pada sistem yang terinfeksi, yang bergantung pada proses perancangan dan pengembangan program jahat tersebut dilakukan (Zalavadiya & Sharma, 2017).



Gambar 2.1 Klasifikasi *Malware (Zalavadiya & Sharma, 2017).*

Berdasarkan Gambar 2.1 *malware* dapat diklasifikasikan ke dalam tiga kategori utama berdasarkan karakteristik dan tujuan serangannya, yaitu *contagious threats, masked threats,* dan *financial threats.*

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 Klasifikasi Ancaman Menular *(Zalavadiya & Sharma, 2017)*.

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Virus	Bentuk *malware* yang mengambil kendali tanpa izin atas komputer yang terinfeksi dan menyebabkan kerusakan tanpa sepengetahuan pengguna.	Program virus tersembunyi di dalam program lain yang tampak tidak berbahaya seperti *file executable*, memiliki kemampuan mereplikasi diri ke program lain, dan menyebarkan infeksi dari satu komputer ke komputer lain.	Penurunan kinerja dan menyebabkan serangan *DoS* (*Denial of Service)*
Worm	*Worm* adalah perangkat lunak jahat mandiri yang dapat beroperasi sendiri dan tidak menempelkan diri untuk menyebar.	*Worm* mengeksploitasi kerentanan dengan memanfaatkan sumber daya komputer atau jaringan dan menyebar melalui perangkat penyimpanan seperti USB atau media komunikasi seperti email.	Mengonsumsi sangat banyak memori dan sumber daya sistem serta menimbulkan masalah kinerja jaringan.

Pada tabel 2.1 menyajikan perbandingan ringkas antara Virus dan *Worm*, berdasarkan karakteristik, cara kerja, dan dampaknya. Virus digambarkan sebagai kode berbahaya yang menyisipkan diri ke dalam program lain yang tampak normal. Virus memerlukan pengguna untuk menjalankan program yang terinfeksi, setelah itu virus dapat mereplikasi diri dan menyebabkan kerusakan pada sistem. Sedangkan *Worm* adalah jenis *malware* yang lebih otonom. Ia tidak memerlukan program inang untuk bereplikasi dan menyebar. *Worm* memanfaatkan kerentanan keamanan dalam sistem atau jaringan untuk menggandakan diri dan menyebar ke komputer lain.

2. Ancaman Tersembunyi (*Masker Threat*)

Jenis-jenis *malware* yang termasuk dalam kategori ini mencakup *Trojan*, *Backdoor, Adware,* dan *Rootkits.* Ciri utama dari *malware* ini adalah dengan melakukan infeksi pada suatu sistem, yang detailnya dapat dilihat pada tabel 2.2.

Tabel 2.2 Klasifikasi Ancaman Tersembunyi *(Zalavadiya & Sharma, 2017)*.

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Trojan	*Malware* berbahaya yang menyembunyikan diri dan bertingkah seperti program yang sah untuk mengambil kendali tidak sah atas komputer atau sistem.	*Trojan* tidak mereplikasi diri sendiri, melainkan diunduh atau disalin melalui interaksi pengguna seperti mengunduh *file* dari *internet* atau perangkat lain.	Mencuri kata sandi atau detail *login*, pencurian uang digital, memodifikasi/ menghapus *file*, memantau aktivitas pengguna.

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Backdoor	Melewati kontrol keamanan normal dan memberikan penyerang akses tidak sah	Diinstal melalui program atau aktivitas berbahaya lainnya	Memodifikasi dan menghapus *file* sistem serta memantau aktivitas sistem
Adware	Menyediakan informasi kepada pengiklan tentang kebiasaan *browsing* pengguna, sehingga memungkinkan pengiklan untuk memberikan iklan yang ditargetkan	*Adware* menyebar melalui situs web	*Clickjacking*, *Phishing*, atau membuat aktivitas berbahaya menggunakan *browser*
Rootkit	*Rootkit* adalah teknik *masking* untuk *malware*, pada dasarnya dirancang untuk tujuan jahat dari program tersebut	Dapat diinstal melalui eksploitasi perangkat lunak atau oleh *trojan*	Mencuri kata sandi atau menginstal *Keylogger*

Pada Tabel 2.2 menjelaskan empat jenis *malware*, cara kerja dan potensi bahaya yang dapat ditimbulkan pada sistem komputer. *Trojan* yang menyamar dan berpura – pura sebagai program sah atau berguna tujuannya untuk mencuri data dan mengendalikan sistem melalui unduhan pengguna. *Backdoor* merupakan jenis *malware* yang mampu melewati keamanan sistem untuk memberikan akses ilegal dan memanipulasi sistem. *Adware* merupakan *malware* yang dirancang untuk mengumpulkan informasi *browsing* untuk menampilkan iklan berbahaya seperti *clickjacking* dan *phishing* melalui situs web. *Rootkit* yang menyembunyikan *malware* lain, memungkinkan aktivitas berbahaya seperti pencurian kata sandi dan instalasi *keylogger* setelah masuk melalui celah keamanan atau *Trojan*.

3. Ancaman Keuangan (*Financial Threats*)

Jenis *malware* yang tergolong dalam kategori ini meliputi *Ransomware*, *Spyware,* dan *Keylogger*. Karakteristik dari *malware* ini adalah kemampuannya menginfeksi sistem, sebagaimana ditampilkan pada tabel 2.3.

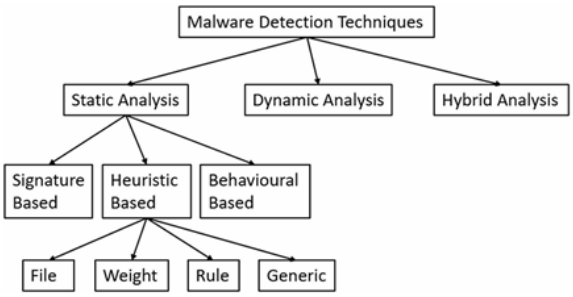
Tabel 2.3 Klasifikasi Ancaman Keuangan *(Zalavadiya & Sharma, 2017)*.

Malware	**Karakteristik**	**Cara Kerja**	**Kerusakan**
Ransomware	*Ransomware* adalah perangkat lunak yang dirancang untuk memblokir akses ke sistem komputer sampai sejumlah uang dibayarkan	*Ransomware* menyebar dan dikirimkan melalui rekayasa sosial dan interaksi pengguna, membuka lampiran *email* berbahaya, mengklik tautan berbahaya	*Ransomware* adalah *malware* untuk penculikan data, mengenkripsi data korban,dan membatasi pengguna untuk mengakses sistem mereka
Spyware	*Spyware* melacak aktivitas pengguna tanpa korban ketahui dan mengirimkan informasi sensitif kembali ke penyerang	Dapat diinstal dengan perangkat lunak lain seperti *freeware* atau dijatuhkan oleh *Trojan*	*Sniffing* antarmuka jaringan, sertifikat digital, kunci enkripsi, dan informasi sensitif lainnya
Keylogger	*Keylogger* secara diam-diam merekam ketikan tombol	Dapat diinstal oleh program jahat lain atau ketika pengguna mengunjungi situs yang terinfeksi	Menangkap informasi sensitif seperti *username*, kata sandi, nomor kartu kredit,

Pada Tabel 2.3 menjelaskan mengenai *Ransomware* yang mengenkripsi data dan meminta tebusan setelah menyebar melalui rekayasa sosial. *Spyware* yang diam-diam memantau aktivitas pengguna dan mencuri informasi sensitif. *Keylogger* yang merekam ketikan tombol untuk mencuri informasi penting seperti kata sandi dan detail perbankan.

2.3 Analisis *Malware*

Analisis *Malware* merupakan tahap awal untuk mengetahui cara kerja serangan *malware* sebagai upaya mitigasi terhadap serangan pada sistem. Tujuannya untuk memahami bagaimana suatu *malware* beroperasi (Djenna dkk., 2023).



Gambar 2.2 Teknik Deteksi *Malware* (Penmatsa dkk., 2020)

Pada Gambar 2.2 menjelaskan berbagai Teknik Deteksi *Malware*. Teknik-teknik tersebut menjadi tiga kelompok utama: *Static Analysis* (Analisis Statis), *Dynamic Analysis* (Analisis Dinamis), dan *Hybrid Analysis* (Analisis Hibrida).

1. Analisis Statis (*Static Analysis)*

   Analisis statis dilakukan tanpa mengeksekusi *malware*. Prosesnya dilakukan dengan membongkar *malware* menggunakan tools rekayasa, kemudian menyusunnya kembali menjadi sebuah perangkat lunak (Yousuf dkk., 2023).

   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 (Nigam, 2020).

b. *Heuristic Detection* (Pendeteksian Heuristis)

Heuristic Detection bekerja dengan mendeteksi perintah-perintah yang tidak terdapat dalam kode aplikasi pembentuk *malware* tersebut. Teknik ini mampu mengidentifikasi *malware* baru yang sebelumnya belum pernah terdeteksi (Eze & Chukwunonso, 2018).

Heuristic Detection dipecah menjadi beberapa cara, yaitu :

1) File Based Heuristic Analysis

Analisis dilakukan pada *file* yang dicurigai sebagai *malware*. Proses penyelidikan mencakup tujuan, isi, dan cara kerja *file*. Jika ditemukan perintah mencurigakan seperti menghapus atau merusak *file*, maka dapat dikategorikan sebagai *malware* (Eze & Chukwunonso, 2018).

2) Weight Based Heuristic Analysis

Analisis dilakukan dengan menilai potensi bahaya aplikasi. Jika skornya melewati batas tertentu maka aplikasi dicurigai mengandung kode berbahaya (Eze & Chukwunonso, 2018).

3) Rule Based Heuristic Analysis

Mengekstraksi pola atau aturan yang menjadi ciri khas sebuah aplikasi. Pola tersebut kemudian dibandingkan dengan aturan yang telah ditetapkan sebelumnya. Jika tidak ada kecocokan, aplikasi tersebut dianggap mengandung *malware* yang dapat membahayakan sistem (Eze & Chukwunonso, 2018).

4) Generic Signature Analysis

Analisis ini dilakukan dengan menggunakan definisi antivirus yang telah ditetapkan untuk mengidentifikasi varian baru dari *malware*. varian di sini adalah *malware* yang memiliki perbedaan dalam perilaku, tetapi masih memiliki keterkaitan dengan *malware* sebelumnya, seperti layaknya "kembar identik" (Eze & Chukwunonso, 2018).

c. *Behavioral Detection* (Deteksi Perilaku)

Behavioral detection adalah metode pendeteksian ancaman seperti *malware* berdasarkan perilaku atau aktivitas mencurigakan yang dilakukan program saat dijalankan, bukan hanya berdasarkan tanda tangan (*signature*) atau pola yang sudah dikenal (Aslan & Samet, 2020).

2. Analisis Dinamis (*Dynamic Analysis*)

Analisis dilakukan dengan menjalankan *malware* untuk menyelidiki cara kerjanya. Aplikasi yang menunjukkan perilaku tidak wajar dari pola normal aplikasi dapat diklasifikasikan sebagai *malware* (Damodaran dkk., 2022).

3. Analisis *Hybrid* (*Hybrid Analysis*)

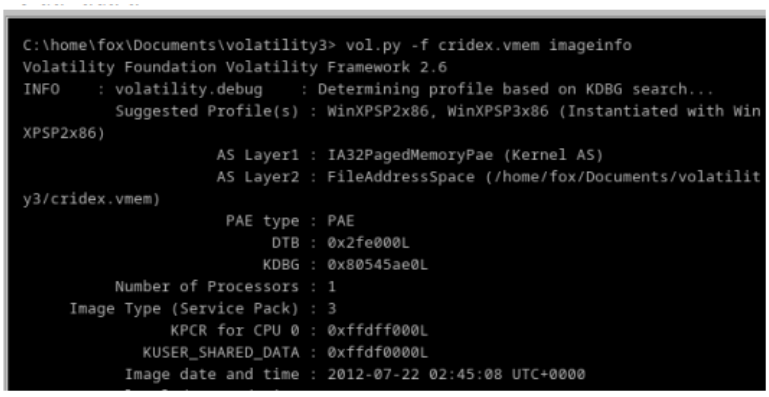
Analisis ini dilakukan dengan mengombinasikan pendekatan analisis statis dan dinamis. Metodenya bekerja dengan memadukan kedua teknik untuk menyelidiki aplikasi yang dicurigai sebagai *malware*. Keuntungan pendekatan ini adalah informasi yang diperoleh mengenai *malware* menjadi lebih akurat (Damodaran dkk., 2022).

2.4 *Memory Forensics*

Memory forensics adalah salah satu metode analisis *malware* yang canggih, praktis, dan juga dapat dimanfaatkan untuk mendeteksi kejahatan siber. Teknik ini sangat bermanfaat karena memungkinkan analisis dilakukan secara mendalam terhadap aktivitas mencurigakan yang terjadi di dalam memori sistem komputer, tanpa harus bergantung pada jenis sistem operasi, perangkat lunak, ataupun struktur sistem *file* tertentu (Triantoro dkk., 2021).

Kemampuan untuk bekerja lintas *platform* menjadikan metode ini sangat fleksibel dan relevan dalam berbagai situasi. Dalam upaya untuk mengidentifikasi dan memahami potensi serangan siber yang melibatkan *malware*, salah satu alat bantu yang umum digunakan dalam *memory forensics* adalah *Volatility*. *Tool* ini mampu mengekstraksi berbagai artefak digital dari *dump memory*, seperti proses aktif, koneksi jaringan, serta modul yang mencurigakan, sehingga membantu proses analisis forensik digital (Das, 2022).

Gambar 2.3 berikut menunjukkan hasil perintah *imageinfo* pada *file memory dump* menggunakan *Volatility Framework*.



Gambar 2.3 Informasi *File* Memori *(Kusuma, 2023)*

Berdasarkan gambar 2.3, dapat dilihat bahwa hasil analisis awal *file* memori cridex.vmem berasal dari Windows XP SP3 32-bit dengan 1 CPU, menggunakan PAE, dan diambil pada 22 Juli 2012.

2.5 *Yara Rules*

YARA rules merupakan seperangkat aturan berbasis pola yang dirancang khusus untuk mendeteksi serta mengidentifikasi keberadaan *malware* dengan cara mencocokan seperti *string, byte*, atau karakteristik spesifik yang terdapat dalam *file*, memori, maupun proses yang sedang berjalan. *YARA rules* dapat membuat para analis keamanan siber melakukan identifikasi secara lebih mendalam dan efisien terhadap potensi ancaman (Sajan dkk., 2025).

Gambar 2.4 menunjukkan struktur umum dari *YARA rule*, yang terdiri atas bagian metadata, *string signature*, serta kondisi logika pencocokan yang harus dipenuhi agar *rule* dinyatakan cocok dengan objek yang dianalisis.

```
rule RuleName
{
meta:
 description = "descriptions of rule"
 author = "name"
 date = "dd/mm/yyyy"
 reference = "url"

strings:
 $text_string1 = "text1 you wish to find in malware"
 $text_string2 = "text2 you wish to find in malware"

 $hex_string1 = {hex1 you wish to find in malware}
 $hex_string2 = {hex2 you wish to find in malware}

 $reg_exp_string1 = /regular expressions1 you wish to find in malware/
 $reg_exp_string2 = /regular expressions2 you wish to find in malware/

condition:
 $text_string1 or $text_string2 or
 $hex_string1 or $hex_string2 or
 $reg_exp_string1 or $reg_exp_string2
}
```

```
rule WannaCry
{
meta:
 description = "Generic Signature of WannaCry"
 author = "Nitin Naik"
 date = "01/06/2018"
 reference = "www.mydomain.com"

strings:
 $text_string1 = "encrypt"
 $text_string2 = "bitcoin"

 $hex_string1 = {B6 D3 56 A5 78 43}
 $hex_string2 = {E8 27 F9 83 C4 82}

 $reg_exp_string1 = /md5: [0-9a-fA-F]{32}/
 $reg_exp_string2 = /state: (on|off)/

condition:
 $text_string1 or $text_string2 or
 $hex_string1 or $hex_string2 or
 $reg_exp_string1 or $reg_exp_string2
}
```

Gambar 2.4 *YARA rules: syntax and example (Naik dkk., 2021)*

Berdasarkan gambar 2.4, dapat diketahui bahwa setiap rule *YARA* terdiri dari tiga komponen utama, yaitu meta, *strings*, dan *condition*. Komponen meta berisi informasi deskriptif seperti nama rule dan penulisnya, *strings* mendefinisikan pola-pola yang akan dicocokkan, sedangkan *condition* menentukan logika pencocokan yang harus terpenuhi agar *rule* dianggap valid. Struktur ini menjadikan *YARA* sangat fleksibel dan efektif dalam melakukan deteksi pola pada berbagai jenis data.

Salah satu keunggulan utama dari *YARA rules* yaitu dapat mengenali pola atau jejak digital yang menjadi indikator adanya *malware*, meskipun *malware* tersebut telah mengalami modifikasi oleh pelaku kejahatan siber. Hal ini menjadikan *YARA* sebagai alat yang sangat berguna dalam kegiatan analisis forensik. *YARA* sering digunakan bersama dengan *tools* lain dalam ekosistem keamanan informasi untuk memperkuat sistem pertahanan. (Jadhav & Jadhav, 2024).

2.6 Penelitian Terkait

Penelitian ini dimulai dengan melakukan studi literatur untuk merangkum berbagai penelitian terdahulu yang relevan. Tujuan dari studi literatur ini adalah untuk memperoleh pemahaman mendalam mengenai perkembangan terbaru di bidang yang diteliti serta mengidentifikasi celah penelitian yang masih jarang dibahas. Melalui pendekatan ini, *state of the art* dari penelitian ini dapat disajikan secara lebih terstruktur, yang kemudian dirangkum dalam Tabel 2.4 berisi ringkasan temuan-temuan dari penelitian sebelumnya.

Tabel 2.4 *State Of The Art*

No.	Judul	Penulis	Metode	Hasil Penelitian
1.	*Advance Malware Analysis Using Static and Dynamic Methodology*	(Saurabh, 2018)	Metode analisis statis dan analisis dinamis	Analisis *malware QQQ.exe* mampu mematikan sistem keamanan *Windows*, menginfeksi program lain, dan berkomunikasi dengan server jarak jauh.
2.	*A Comparison of Static, Dynamic, and Hybrid Analysis for Malware Detection*	(Damodaran dkk., 2022)	Metode analisis statis, analisis dinamis dan *hybrid* analisis	Penelitian ini membandingkan tiga pendekatan analisis *malware*. Hasil menunjukkan bahwa analisis dinamis memberikan tingkat deteksi terbaik, sedangkan pendekatan *hybrid* tidak selalu lebih unggul dibanding metode statis.
3.	Analisis Dan Deteksi *Malware* Poison Ivy Dengan Metode *Malware* Analisis Dinamis Dan *Malware* Analisis Statis	(Amdani dkk., 2021)	Metode analisis statis dan analisis dinamis	Poison Ivy memanfaatkan enkripsi *password* untuk autentikasi serta bekerja tanpa mengonsumsi banyak sumber daya.

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
4.	Analisis Deteksi *Malicious Activity* Menggunakan Metode Analisis *Malware* Dinamis Berbasis Anomali	(Daniswara dkk., 2019)	Metode analisis dinamis	Dengan menganalisis pola anomali yang tidak sesuai dengan pola normal program, penelitian ini menguji 10 sampel *malware* secara acak menggunakan tiga alat pendeteksi. Hasilnya menunjukkan bahwa 40% dari sampel yang diuji teridentifikasi sebagai *malware* aktif.
5.	*Malware Detection Approach Based on Artifacts in Memory Image and Dynamic Analysis*	(Sihwail dkk., 2019)	Metode analisis dinamis dan *memory forensics*	Deteksi *malware* yang mengintegrasikan analisis memori dan analisis dinamis, dengan menggunakan *memory forensics* untuk mengekstrak artefak berbahaya, pendekatan ini meningkatkan akurasi deteksi hingga 98,5% dan menurunkan *positif false* menjadi 1,7%.
6.	Analisis *Malware Hummingbad* dan *Copycat* Pada *Android* Menggunakan Metode *Hybrid*	(Qomariah dkk., 2023)	Metode analisis *Hybrid*	*HummingBad* memiliki tingkat keamanan 27/100, *CopyCat* memiliki tingkat 38/100. *HummingBad* dianggap lebih berbahaya karena memiliki lebih banyak pelacak perangkat yang ditemukan melalui *framework* MobSF.

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
7.	*Hack.exe Malware Analysis and Investigation Using Memory Forensics*	(Triantoro dkk., 2021)	Analisis dinamis, *Memory Forensics*	*Malware Hack.exe* diidentifikasi sebagai *trojan* yang mencuri data pengguna dan menyinkronkannya ke IP tertentu (24.146.133.195). Pencegahan mencakup kewaspadaan terhadap *email* mencurigakan, pemulihan melibatkan penggunaan antivirus, merubah kredensial akun yang terkena.
8.	*Analisis Malware Aquvaprn.exe Untuk Investigasi Sistem Operasi Dengan Metode Memory Forensics*	(Aditya dkk., 2024)	*Memory Forensics*	*Malware* AQUVAPRN.exe dapat membocorkan data pribadi dengan memanfaatkan hak RAT. *Memory Forensics* berhasil mengungkap proses, koneksi jaringan, dan metode pengamanan *malware*.
9.	*MalDicom: A Memory Forensic Framework for Detecting Malicious Payload in DICOM Files*	(Mishra & Bagade, 2023)	*Memory Forensics*	*Framework* MalDicom mendeteksi *malware* yang disisipkan ke *file* DICOM menggunakan analisis memori forensik berbasis pembelajaran mesin. algoritma *Random Forest* mencapai akurasi 75% diidentifikasi sebagai indikator penting menggunakan *Shapley Values.*

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
10.	*Live Memory Forensic: Capture and Analyzing Volatile Data*	(Rabia Mehmood, 2024)	*Memory Forensics*	*Live Memory Forensics* memungkinkan analisis data volatil di *RAM* untuk mendeteksi ancaman siber seperti *malware*. Dengan alat seperti *Volatility* dan *FTK Imager*, metode ini mengungkap bukti penting yang sering kali tidak terdeteksi oleh metode tradisional.
11.	*Malware Analysis Using Memory Forensics*	(Das, 2022)	*Memory Forensics,* Analisis Statis dan dinamis	Mengidentifikasi fitur penting dari *file* PE untuk klasifikasi *malware*, mencapai akurasi tinggi dalam deteksi ancaman. Teknik ini meningkatkan keamanan siber dengan pendekatan berbasis pembelajaran mesin.
12.	*AUMFOR: Automated Memory Forensics for Malware Analysis*	(Rughani & Rughani, 2017)	*Memory Forensics*	AUMFOR adalah alat forensik memori otomatis berbasis GUI yang menyederhanakan analisis *dump* memori dengan integrasi *Volatility* dan *VirusTotal*. Memberikan laporan akurat untuk membantu investigasi digital

No.	Judul	Penulis dan Tahun	Metode	Hasil Penelitian
13.	*Memory forensic: Acquisition and analysis mechanism for operating systems*	(Shree dkk., 2021)	*Memory Forensics*	Pentingnya *Memory Forensics* dalam investigasi digital untuk mendeteksi ancaman yang tersembunyi di memori sistem. Teknik ini memungkinkan analisis data volatil di *RAM* untuk mengidentifikasi *malware*.
14.	*Embedding Fuzzy Rules with YARA Rules for Performance Optimisation of Malware Analysis*	(Naik dkk., 2020)	*Yara Rules*	Dengan menggabungkan logika fuzzy *dan yara*, metode ini mampu menangani data yang tidak pasti, meningkatkan akurasi deteksi hingga 73,5%, dan memberikan fleksibilitas dalam analisis *ransomware*.
15.	*Living off the Analyst: Harvesting Features from Yara Rules for Malware Detection*	(Gupta dkk., 2024)	*Yara Rules*	Dengan menggabungkan *sub-signatures YARA* dan fitur dataset EMBER 2018, metode ini meningkatkan akurasi deteksi hingga 1,8% pada tingkat *false positive* yang rendah (0,01%), memperkaya model deteksi *malware* dengan fitur yang lebih efisien dan spesifik.

Tabel 2.5 Matriks Penelitian

No.	Judul	Ruang ligkup penelitian					Penulis
		Analisis					
		Statis	**Dinamis**	***Hybrid***	***Memory Forensic***	***Yara Rules***	
1.	*Advance Malware Analysis Using Static and Dynamic Methodology*	✔	✔				(Saurabh, 2018)
2.	*A Comparison of Static, Dynamic, and Hybrid Analysis for Malware Detection*	✔	✔	✔			(Damodaran dkk., 2022)
3.	Analisis Dan Deteksi *Malware Poison Ivy* Dengan Metode *Malware* Analisis Dinamis Dan *Malware* Analisis Statis	✔	✔				(Amdani dkk., 2021)
4.	Analisis Deteksi *Malicious Activity* Menggunakan Metode Analisis *Malware* Dinamis Berbasis Anomali		✔				(Daniswara dkk., 2019)
5.	*Malware Detection Approach Based on Artifacts in Memory Image and Dynamic Analysis*		✔		✔		(Sihwail dkk., 2019)
6.	Analisis *Malware Hummingbad* Dan *Copycat* Pada *Android* Menggunakan Metode *Hybrid*			✔			(Qomariah dkk., 2023)
7.	*Hack.exe Malware Analysis and Investigation Using Memory Forensics*		✔		✔		(Triantoro dkk., 2021)

No.	Judul	Ruang ligkup penelitian					Penulis
		Analisis			*Memory Forensic*	*Yara Rules*	
		Statis	**Dinamis**	***Hybrid***			
8.	Analisis *Malware Aquvaprn.exe* untuk Investigasi Sistem Operasi dengan Metode *Memory Forensics*				✔		(Aditya dkk., 2024)
9.	*MalDicom: A Memory Forensic Framework for Detecting Malicious Payload in DICOM Files*				✔		(Mishra & Bagade, 2023)
10.	*Live Memory Forensic: Capture and Analyzing Volatile Data*				✔		(Rabia Mehmood, 2024)
11.	*Malware Analysis Using Memory Forensics*	✔	✔		✔		(Das, 2022)
12.	*AUMFOR: Automated Memory Forensics for Malware Analysis*				✔		(Rughani & Rughani, 2017)
13.	*Memory forensic: Acquisition and analysis mechanism for operating systems*				✔		(Shree dkk., 2021)
14.	*Embedding Fuzzy Rules with YARA Rules for Performance Optimisation of Malware Analysis*					✔	(Naik dkk., 2020)
15.	*Living off the Analyst: Harvesting Features from Yara Rules for Malware Detection*					✔	(Gupta dkk., 2024)
	Penelitian Usulan	✔	✔	✔	✔	✔	

Berdasarkan Tabel 2.5 dapat ditemukan adanya kesamaan dalam pendekatan yang digunakan oleh berbagai penelitian sebelumnya, khususnya dalam pemanfaatan metode *dynamic analysis* dan teknik *memory forensics* untuk melakukan investigasi terhadap aktivitas *malware*. Meskipun pendekatan-pendekatan tersebut telah banyak digunakan, penelitian ini menghadirkan suatu pembaruan yaitu penggunaan *YARA Rules* sebagai metode tambahan dalam proses investigasi. Inovasi ini menjadi pembeda utama dari penelitian-penelitian terdahulu, karena *YARA Rules* memungkinkan identifikasi karakteristik *malware* secara lebih spesifik dan efisien, terutama dalam kasus analisis terhadap *malware* bernama *ramez.exe*. Pendekatan ini tidak hanya memperkuat hasil investigasi, tetapi juga memperluas cakupan deteksi terhadap pola-pola berbahaya yang mungkin tidak terdeteksi melalui cara yang biasa digunakan.