

BAB II

TINJAUAN PUSTAKA

Portable Document Format (PDF) telah menjadi salah satu standar utama dalam pertukaran dokumen sejak diperkenalkan pada tahun 1993. Fleksibilitasnya memungkinkan penyematan berbagai elemen, seperti teks, gambar, dan JavaScript. Namun, fitur ini juga dapat dimanfaatkan oleh pihak tidak bertanggung jawab untuk menyisipkan kode berbahaya yang mengeksploitasi kerentanan dalam perangkat lunak pembaca PDF, menjadikannya vektor utama dalam penyebaran *malware*. Ancaman *malware* PDF terus meningkat akibat celah keamanan pada perangkat lunak seperti Adobe Reader, yang telah melaporkan lebih dari 150 kerentanan sejak 2015 [9].

Untuk mendeteksi *malware* PDF, pendekatan yang digunakan umumnya terbagi menjadi dua kategori utama: analisis statis dan dinamis. Analisis statis melibatkan pemeriksaan elemen seperti *metadata*, jalur struktural, dan pola *encoding* tanpa mengeksekusi kode. Sementara itu, analisis dinamis memantau perilaku kode selama *runtime* untuk mengidentifikasi aktivitas mencurigakan, seperti modifikasi *registry* atau teknik *heap spraying* [9].

Dalam pengembangan lebih lanjut, metode *surface analysis* dan *runtime analysis* juga digunakan untuk mendeteksi perilaku mencurigakan yang dilakukan oleh *malware*, terutama saat menginfeksi sistem operasi seperti Windows. *Surface analysis* memungkinkan pemeriksaan terhadap struktur eksternal *file* tanpa perlu mengeksekusinya, misalnya dengan mengidentifikasi jenis *file* asli, ukuran sebenarnya, dan keberadaan *file* tersembunyi di dalamnya. Selain itu, metode ini juga dapat menghasilkan *hash function* sebagai identitas unik dari *file*. Sementara itu, *runtime analysis* melibatkan eksekusi *file* dalam lingkungan terkontrol untuk memantau aktivitas secara langsung, seperti proses yang berjalan, perubahan pada *registry*, komunikasi jaringan, dan peristiwa anomali lainnya selama infeksi berlangsung [10].

Teknik Obfuscation sering digunakan untuk menghindari deteksi, misalnya

dengan membagi kode berbahaya ke dalam beberapa objek atau menggunakan String terenkripsi. Untuk mengatasi tantangan ini, berbagai metode *preprocessing* telah diterapkan, termasuk normalisasi sintaksis, analisis entropi, dan seleksi fitur [9], [11].

Support Vector Machine (SVM) merupakan salah satu algoritma *machine learning* yang sering digunakan dalam deteksi *malware* PDF. Algoritma ini mampu mengenali pola anomali berdasarkan fitur statis yang diekstraksi, seperti elemen JavaScript, entropi String, dan *metadata file*. Studi sebelumnya menunjukkan bahwa SVM dapat memisahkan *file* PDF berbahaya dari *file* jinak dengan tingkat akurasi yang tinggi [12]. Selain itu, penggunaan *token* JavaScript sebagai fitur telah terbukti meningkatkan efisiensi deteksi [13].

Random Forest (RF) juga menjadi algoritma populer dalam deteksi *malware* PDF karena kemampuannya dalam menangani data berfitur banyak dan menghasilkan prediksi yang stabil. *RF* bekerja dengan membangun banyak pohon keputusan (Decision Trees) dan menggabungkan hasilnya melalui voting untuk menentukan label akhir. Pendekatan ini terbukti efektif dalam mendeteksi *malware* dengan akurasi tinggi serta mengurangi risiko *overfitting* [7], [8], [14]. Selain itu, *RF* juga memiliki keunggulan dalam hal kemudahan dalam penjelasan hasil model dan efisiensi dalam seleksi fitur, terutama ketika dikombinasikan dengan *preprocessing* yang baik.

Selain SVM dan Random Forest, pendekatan berbasis pohon keputusan (Decision Tree) tunggal juga telah dikembangkan untuk mengoptimalkan seleksi fitur, seperti struktur *metadata* dan jalur internal PDF. Kombinasi fitur ini meningkatkan efektivitas algoritma dalam mendeteksi *malware evasive* [11], [15].

Keakuratan model sangat dipengaruhi oleh kualitas *dataset* yang digunakan. CIC-Evasive-PDFMal2022, yang menggabungkan data dari Contagio dan VirusTotal, dirancang untuk mengatasi tantangan *malware evasive*. *Dataset* ini terdiri dari 18.126 sampel PDF (9.063 *file malicious* dan 9.063 *file benign*) dengan fitur yang mencakup pola *encoding* dan elemen JavaScript [6]. Penggunaan *dataset* yang lebih representatif terhadap ancaman dunia nyata telah terbukti meningkatkan ketahanan model dalam mendeteksi *malware zero-day*

[12], [16].

Evolusi *malware* terus menghadirkan tantangan bagi sistem deteksi. Pendekatan berbasis tanda tangan sering kali gagal mengenali *malware* baru, sehingga solusi berbasis *machine learning* menjadi lebih relevan. Namun, model *machine learning* juga memiliki kelemahan karena rentan terhadap serangan *malware evasive*. Untuk mengatasi hal ini, metode *clustering* telah diterapkan guna mengelompokkan *malware* berdasarkan pola serupa, yang dapat meningkatkan akurasi deteksi dengan mengidentifikasi hubungan antar sampel *malware* [13], [14].

Pendekatan terbaru menggunakan Ensemble Learning, seperti *stacking*, untuk menggabungkan hasil dari berbagai model guna meningkatkan performa deteksi, terutama dalam menghadapi *dataset* yang kompleks [6]. Dengan mengintegrasikan teknik *preprocessing*, algoritma seperti SVM dan Random Forest, serta *dataset* yang komprehensif, pendekatan ini bertujuan untuk menciptakan sistem deteksi *malware* PDF yang lebih efektif dan efisien, yang tidak hanya mampu mendeteksi ancaman saat ini tetapi juga mengantisipasi ancaman pada masa depan.

Tabel 2.1 Tabel Perbandingan Literatur

No	Penulis	Judul	Metode yang Digunakan	Jenis Analisis	Dataset	Fitur Utama	Teknik Preprocessing	Keterbatasan	Metode Evaluasi
1	Elingiusti et al [9]	PDF-malware detection: a survey and taxonomy of current techniques	<i>Taxonomy development, comparative analysis</i>	Statis dan dinamis	Tidak Disebutkan	<i>Metadata, elemen JavaScript, structural paths</i>	Tidak spesifik (bervariasi tergantung teknik)	Fokus pada kategori, tidak implementasi langsung	Analisis komparatif teknik deteksi
2	Siddiq et al [10]	Analisis Perilaku Malware Dengan Metode Surface Analysis Dan Runtime Analysis	<i>Surface analysis dan runtime analysis</i>	Dinamis	Data lokal malware Windows	<i>Hash file (MD5, SHA-1), aktivitas runtime</i>	Pemindaian <i>metadata</i> , logging aktivitas	Tidak mencakup metode berbasis <i>machine learning</i>	Studi kasus berbasis perilaku <i>malware</i>
3	Wiharja et al [12]	Comparative Study of the Effect of Datasets and Machine learning Algorithms for PDF Malware Detection	<i>Machine learning (SVM, Decision Tree, Random Forest)</i>	Statis	CIC-Evasive-PDFMal2022	<i>String entropy, metadata, embedded keywords</i>	Seleksi fitur, normalisasi	Bergantung pada kualitas <i>dataset</i>	<i>Accuracy, Precision, Recall</i>
4	Lu et al [11]	A Universal Malicious Documents Static Detection Framework	<i>Generalized static detection</i>	Statis	<i>Diverse datasets</i>	<i>Structural paths, metafeatures</i>	<i>Path consolidation, clustering</i>	Tidak mencakup serangan <i>evasive</i> secara rinci	<i>Clustering effectiveness, Akurasi</i>

		Based on Feature Generalization							
5	Issakhani et al [6]	PDF Malware Detection based on Stacking Learning	Ensemble learning (<i>stacking</i>)	Statis dan dinamis	VirusTotal, Contagio, dataset publik	<i>Opcode sequences, JS tokens</i>	Normalisasi sintaks, analisis entropi	Konsumsi waktu tinggi pada <i>preprocessing</i>	<i>Ensemble performance (Precision, Recall)</i>
6	Kendrick Chandra	Deteksi Malware pada File PDF dengan Algoritma SVM	SVM dan Random Forest	Statis	CIC-Evasive-PDFMal2022	<i>Metadata, Javascript, Keywords</i>	Seleksi Fitur, Normalisasi, <i>Encoding</i>	Tidak implementasi langsung	<i>Accuracy, Precision, Recall, F1-Score, AUC</i>

Tabel 2.1 merangkum sejumlah penelitian terdahulu berdasarkan metode yang digunakan, jenis analisis, *dataset*, fitur utama, teknik *preprocessing*, keterbatasan, dan metode evaluasi. Dari rangkuman tersebut, terlihat bahwa studi oleh Elingiusti et al. [9] dan Siddiq et al. [10] lebih berfokus pada pengembangan taksonomi dan analisis perilaku tanpa menerapkan metode *machine learning* secara langsung. Sebaliknya, penelitian seperti yang dilakukan oleh Wiharja et al. [12], Lu et al. [11], dan Issakhani et al. [6] menggunakan pendekatan *machine learning*, baik secara individu maupun melalui teknik Ensemble Learning seperti *stacking*.

Beberapa studi memanfaatkan *dataset* publik seperti *Evasive-PDFMal2022* dan *VirusTotal*, yang menyediakan data seimbang dan representatif terhadap kondisi ancaman di dunia nyata. Fitur utama yang dianalisis meliputi *metadata*, elemen JavaScript, entropi String, dan *structural paths*, yang kemudian diproses melalui teknik *preprocessing* seperti seleksi fitur, normalisasi sintaksis, dan analisis entropi.

Meskipun demikian, masih terdapat beberapa keterbatasan, seperti ketergantungan pada kualitas *dataset* dan tingginya konsumsi waktu dalam proses *preprocessing*. Evaluasi performa model dilakukan dengan berbagai metrik, termasuk *Accuracy*, *Precision*, *Recall*, *F1-score*, dan AUC. Rangkuman ini memperkuat pemahaman terhadap pendekatan yang telah dilakukan dalam studi sebelumnya, sekaligus menegaskan posisi dan kontribusi penelitian ini dalam mengembangkan sistem deteksi *malware* PDF yang lebih efektif dan efisien.