

DETEKSI MALWARE MENGGUNAKAN MACHINE LEARNING DENGAN METODE ENSEMBLE

Iik Muhamad Malik Matin^{1,2*}, Maria Agustin¹, Bambang Sugiarto², dan Ai Nur Asri²

¹ Teknik Informatika dan Komputer, Politeknik Negeri Jakarta
Jl. Prof. DR. G.A. Siwabessy, Kota Depok, Jawa Barat 16425, Indonesia.

² Teknik Elektro, Fakultas Teknik, Universitas Garut
Jl. Jati No 42B Garut, Jawa Barat 44151, Indonesia.

*Email: iik.muhamad.malik.matin@tik.pnj.ac.id

Abstrak

Serangan malware telah menjadi perhatian penting di era digital ini. Malware dapat diartikan sebagai perangkat lunak yang digunakan untuk melakukan kerusakan sistem, pencurian atau pengumpulan informasi, hingga mendapatkan akses terhadap suatu sistem. Machine learning merupakan sub area dari ilmu komputer yang mampu memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit. Pada penelitian ini akan mendeteksi malware windows menggunakan machine learning dengan metode ensemble. Model klasifikasi yang digunakan adalah Decision Tree, Random forest, Bagging, AdaBoost dan Hist Gradient Boosting. Penelitian ini menggunakan dataset malware berbasis windows. Hasil yang didapatkan menggunakan algoritma Hist Gradient Boosting lebih tinggi yaitu sebesar 96,9% dibandingkan dengan algoritma Decision tree sebesar 93,5% algoritma Random forest sebesar 94,9% algoritma Adaboost sebesar 87,8% dan algoritma Bagging sebesar 95,8%.

Kata kunci: Malware, Machine Learning, Metode Ensemble

1. PENDAHULUAN

Dalam era digital yang semakin pesat, *software* berbahaya atau malware telah menjadi ancaman serius bagi keamanan sistem komputer dan perangkat lunak. Setiap tahun, ancaman malware terus meningkat dengan perangkat lunak yang dirancang dengan tujuan negatif, termasuk merusak data, mencuri informasi penting, mengganggu kinerja perangkat, dan mengambil alih sistem (Ari Sandriana et al., 2022). Oleh karena itu, deteksi malware menjadi sangat penting dalam upaya melindungi sistem komputer dan jaringan dari serangan berbahaya.

Pendekatan tradisional dalam deteksi malware umumnya menggunakan *signature* atau heuristik yang diketahui dari sampel malware yang telah ada sebelumnya (Prayitno, 2022). Namun, pendekatan ini memiliki batasan karena tidak efektif dalam mendeteksi malware yang belum dikenal atau varian baru yang telah dimodifikasi. Untuk mengatasi keterbatasan ini, metode deteksi malware yang menggunakan *Machine Learning* (Pembelajaran Mesin) telah menjadi fokus penelitian yang signifikan.

Machine Learning merupakan kecerdasan buatan yang memungkinkan komputer untuk belajar dari data dan membangun model prediktif. Dalam konteks deteksi malware, *Machine Learning* mempelajari pola dan karakteristik dari sampel malware yang dikenal untuk mengklasifikasikan sampel baru sebagai malware atau bukan malware. *Machine learning* banyak diterapkan dalam pendeteksian malware baik berdasarkan pendekatan analisis statis, dinamis, maupun *hybrid* (Liu et al., 2020).

Salah satu pendekatan *Machine Learning* yang telah terbukti efektif dalam deteksi malware adalah metode *Ensemble*. *Ensemble* merupakan teknik yang menggabungkan beberapa model *Machine Learning* untuk meningkatkan performa dan akurasi deteksi. Pendekatan *Ensemble* mengumpulkan hasil dari beberapa model berbeda, seperti pohon keputusan (*decision tree*), *random forest*, atau support vector machine (SVM), dan membuat keputusan berdasarkan mayoritas suara atau bobot yang diberikan pada masing-masing model. Dengan menggunakan *Ensemble*, kelemahan dari satu model dapat dikompensasi oleh kekuatan model lainnya, sehingga meningkatkan kemampuan deteksi secara keseluruhan.

Beberapa penelitian telah dilakukan untuk menguji efektivitas deteksi malware menggunakan machine learning berbasis PE seperti dilakukan oleh Alvian Bastian (Bastian, 2021) yang mengusulkan antivirus *signature* berdasarkan *DLL Files* dan *API Calls*. Pada penelitian ini, Alvian melakukan deteksi

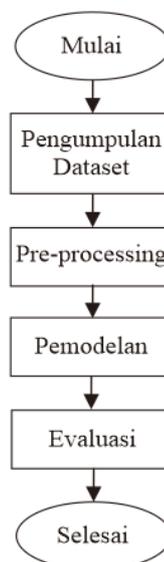
pada ransomware menggunakan model KNN, SVM, *Decision tree* dan *Random forest*. Hasilnya *Decision tree* menunjukkan precision, recall, F1-Score dan support terbaik. Namun, pada penelitian ini terbatas pada deteksi ransomware. Penelitian oleh Dieta et al (Asry et al., 2023) mengusulkan deteksi malware statis pada malware PE menggunakan machine learning. *Deep neural network* digunakan untuk menganalisis *file* portabel secara statis untuk mempelajari fitur-fitur *portable executable* guna meminimalisir kesalahan *false positive*. Model yang diusulkan mencapai AUC sebesar 99,8% dengan 98% true positive pada 1% false positive pada kurva ROC. Selain itu, penelitian oleh Paza et al (Paza, 2023) mengembangkan sistem deteksi malware menggunakan algoritma *gradient boosting classification*. Pada penelitian ini juga menggunakan sampel malware PE yang diambil dari Kaggle dan virus total dan diambil dari hasil ekstraksi *file* di windows. Sampel yang telah diekstraksi kemudian diseleksi fitur penting dari PE. Beno Ramadhan et al (Ramadhan et al., 2020) mengusulkan sistem deteksi malware menggunakan proses analisis statis dan analisis dinamis dalam mengklasifikasikan data menggunakan machine learning. Naïve bayes ,digunakan untuk mengidentifikasi malware dan *file* jinak dengan akurasi yang tinggi.

Berbeda dengan penelitian yang ada, pada penelitian ini dilakukan pengembangan model menggunakan *ensemble method*. Ensemble bekerja dengan menggabungkan output dari pengklasifikasi dasar individu untuk menghasilkan prediksi yang akurat untuk banyak masalah klasifikasi (Gupta & Rani, 2020). Kontribusi dari penelitian ini yaitu 1). melakukan eksperimen model yang terdiri dari *Decision tree*, *Random forest*, *AdaBoost*, *Bagging*, dan *Hist Gradient Boosting*. 2). mengukur performa pada setiap model yang diusulkan dengan menggunakan confusion matrix yang terdiri dari parameter akurasi, presisi, *recall* dan F1-score.

2. METODOLOGI

2.1. Metode

Deteksi malware windows menggunakan machine learning dengan metode *ensemble* dilakukan dengan tahapan yang ditunjukkan pada gambar 1.



Gambar 1. tahapan pengembangan model

2.2. Pengumpulan Dataset

Tahap pertama adalah pengumpulan dataset. Dataset ini terdiri dari kumpulan *file* eksekusi Portable Executable (PE) yang mencakup contoh *file* malware dan non-malware. Dataset ini dikumpulkan dari dataset ClaMP malware (Kumar, n.d.). Dataset terdiri dari 5.184 sampel. Sampel terdiri dari 2.683 sampel malware dan 2.501 sampel jinak yang diestraksi dari *file* windows. Selain itu, Clamp Malware

memiliki 4 fitur utama yang terdiri dari Image DOS Header, *File_Header* dan *Optional Header*. Setiap fitur utama terdiri dari fitur yang ditunjukkan pada tabel 1.

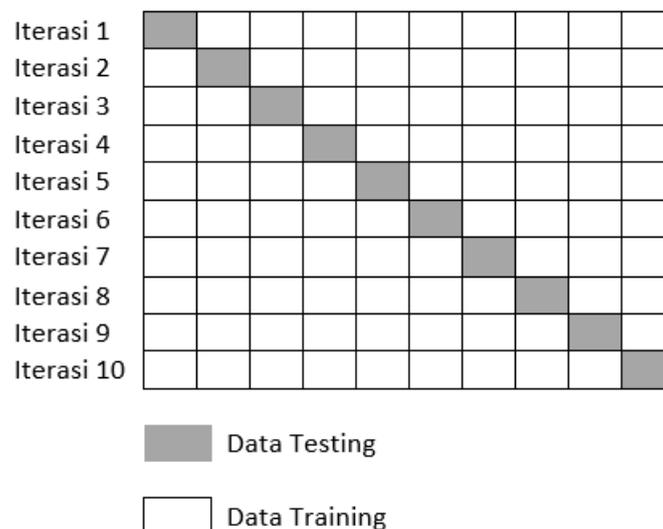
Tabel 1. Fitur Dataset

Fitur Utama	Fitur dataset
<i>Dos_Header</i>	<i>e_magic, e_cblp, e_cp, e_crlc, e_cparhdr, e_minalloc, e_maxalloc, e_ss, e_sp, e_csum, e_ip, e_cs, e_lfarlc, e_ovno, e_res, e_oemid, e_oeminfo, e_res2, e_lfanew</i>
<i>File_Header</i>	<i>Machine, NumberOfSections, CreationYear, PointerToSymbolTable, NumberOfSymbols, SizeOfOptionalHeader, Characteristics</i>
<i>Optional_Header</i>	<i>Magic, MajorLinkerVersion, MinorLinkerVersion, SizeOfCode, SizeOfInitializedData, SizeOfUninitializedData, AddressOfEntryPoint, BaseOfCode, BaseOfData, ImageBase, SectionAlignment, FileAlignment, MajorOperatingSystemVersion, MinorOperatingSystemVersion, MajorImageVersion, MinorImageVersion, MajorSubsystemVersion, MinorSubsystemVersion, SizeOfImage, SizeOfHeaders, CheckSum, Subsystem, DllCharacteristics, SizeOfStackReserve, SizeOfStackCommit, SizeOfHeapReserve, SizeOfHeapCommit, LoaderFlags, NumberOfRvaAndSizes</i>

2.3. Preprocessing

Setelah dataset dikumpulkan, tahap selanjutnya adalah preprocessing data. Pada tahap ini, *file-file* PE dalam dataset dianalisis secara mendalam untuk mengekstraksi fitur-fitur yang relevan. Fitur-fitur ini dapat mencakup *Dos_Header*, *File_Header*, dan *Optional_Header*. *Preprocessing* juga melibatkan normalisasi data menggunakan *MinMax Scaler* untuk *numeric value*, dan *Onehotencoder* untuk *categoric value*.

Dataset dibagi menjadi masing-masing data training dan data testing. Setiap data training dan data testing masing-masing memiliki rasio 80:20. Selain itu, saat training dilakukan *cross validation* dengan nilai CV=10. Skema *cross validation* ditunjukkan pada gambar 2.



Gambar 2. Cross validation dengan nilai 10

2.4. Model Klasifikasi

Setelah preprocessing, tahap selanjutnya adalah membangun model klasifikasi menggunakan metode *ensemble*. Metode *ensemble* melibatkan penggabungan beberapa algoritma machine learning untuk meningkatkan performa dan keandalan deteksi malware. Beberapa metode *ensemble* yang dapat digunakan yaitu *decision tree*, *random forest*, *adaboost*, dan *Hist Gradient Boosting*. Tabel 2 menunjukkan arsitektur setiap model yang digunakan.

Tabel 2. Model dan Arsitekturnya

Model	Arsitektur Parameter
<i>Decision tree</i>	<i>Criterion</i> : Gini
	<i>Splitter</i> : best
	<i>Min_sample_split</i> : 2
<i>Random forest</i>	<i>Criterion</i> : Entropy
	<i>Max_Depth</i> : 10
	<i>N_estimator</i> : 100
	<i>Min_Samples_split</i> : 2
<i>AdaBoost</i>	<i>N_estimator</i> : 50
	<i>Learning_rate</i> : 1.0
	<i>algorithm</i> : SAME.R
<i>Bagging Classifier</i>	<i>Max_feature</i> : 1.0
	<i>N_estimator</i> : 10
	<i>Max_samples</i> : 1.0
<i>Hist Gradient Boosting</i>	<i>Loss</i> : Log_loss
	<i>Learning_rate</i> : 0.1
	<i>Max_Iter</i> : 100
	<i>Min_samples_leaf</i> : 20

2.5. Evaluasi

Setelah model klasifikasi dibangun, tahap terakhir adalah evaluasi model. Dataset yang telah dipreprocessing dibagi menjadi subset pelatihan dan pengujian. Subset pelatihan digunakan untuk melatih model dengan menggunakan algoritma *ensemble*. *Confusion matrix* digunakan untuk menentukan hasil klasifikasi yang dilakukan pada setiap model-model *ensemble learning*. Matriks ini memudahkan untuk menemukan hubungan antara kinerja pengklasifikasi dan hasil pengujian (Koklu et al., 2021). *Confusion matrix* ditunjukkan pada gambar 3.

		Sebenarnya	
		Positif	Negatif
Prediksi	Positif	TN (True Positive)	FN (False Positive)
	Negatif	FP (False Negative)	TN (True Negative)

Gambar 3. Confusion matrix

Subset pengujian kemudian digunakan untuk menguji performa model dan mengukur kinerja model menggunakan metrik evaluasi seperti akurasi, presisi, recall, dan F1-score. Metrik evaluasi ini memberikan gambaran tentang seberapa baik model dapat mendeteksi malware PE dengan metode *ensemble* yang digunakan. Ukuran penilaian yang digunakan dalam penelitian ini adalah sebagai berikut:

- *Accuracy* adalah nilai untuk mengetahui seberapa akurat sistem mengklasifikasikan data tersebut secara benar yang dirumuskan pada formula (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

- *Precision* (presisi) adalah nilai untuk mengetahui jumlah data positif yang diklasifikasikan secara benar dibagi total data yang diklasifikasikan positif yang dirumuskan pada formula (2).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

- *Recall* adalah nilai untuk mengetahui berapa persen data kategori positif yang diklasifikasikan dengan benar oleh sistem yang dirumuskan pada formula (3).

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

- *F1-Score* adalah nilai *harmonic mean* dari *presisi* dan *recall*. Nilai terbaik *f1-score* adalah 1.0 dan terburuknya adalah 0. Jika nilai *f1-score* memiliki skor baik maka mengindikasikan bahwa metode klasifikasi yang dibangun memiliki presisi dan *recall* yang dirumuskan pada formula (4).

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4)$$

3. HASIL DAN PEMBAHASAN

Data yang dihasilkan pada proses klasifikasi model machine learning adalah sebagai berikut:

Tabel 3. hasil pengukuran

Algoritma	Confusion Matrix			
	Akurasi	Precision	Recall	F1-Score
<i>Decision Tree</i>	93,5%	93,5%	93,3%	93,4%
<i>Random forest</i>	94,9%	95,3%	94,6%	94,8%
<i>AdaBoost</i>	87,8%	87,9%	87,3%	87,5%
<i>Bagging</i>	95,8%	95,7%	95,7%	95,7%
<i>Hist Gradient Boosting</i>	96,9%	97,0%	96,7%	96,8%

Dalam tabel yang ditunjukkan pada tabel 3 terdapat beberapa algoritma pembelajaran mesin yang dievaluasi menggunakan beberapa metrik kinerja seperti akurasi, presisi, recall, dan skor F1.

Decision tree memiliki kinerja yang cukup baik dengan tingkat akurasi sebesar 93,5%. Konsistensi antara presisi dan recall juga terlihat pada angka yang hampir sama, yaitu sekitar 93%. *Random forest* memberikan kinerja yang lebih baik dibandingkan *Decision tree* dengan meningkatkan akurasi menjadi 94,9%. Presisi yang lebih tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam mengidentifikasi kelas positif, sementara recall yang tinggi menunjukkan kemampuan model dalam menemukan sebagian besar instance yang relevan dari kelas positif. *AdaBoost* memiliki akurasi yang sedikit lebih rendah dibandingkan dengan *Decision tree* dan *Random forest*. Namun, presisi, recall, dan F1-Score yang sebanding menunjukkan keseimbangan antara kemampuan model dalam mengklasifikasikan kelas positif dan menemukan instance yang relevan. *Bagging* memberikan kinerja yang sangat baik dengan akurasi sebesar 95,8%. Presisi, recall, dan F1-Score yang tinggi menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam mengklasifikasikan kelas positif dan menemukan instance yang relevan. *Hist Gradient Boosting* memberikan kinerja terbaik di antara semua algoritma yang dievaluasi. Dengan akurasi 96,9%, presisi yang sangat tinggi, dan recall yang tinggi, model ini memiliki kemampuan yang sangat baik dalam mengklasifikasikan kelas positif dan menemukan instance yang relevan.

Secara umum, algoritma *Random forest*, *Bagging*, dan *Hist Gradient Boosting* menunjukkan kinerja yang lebih baik dibandingkan *Decision tree* dan *AdaBoost* dalam hal akurasi, presisi, recall, dan F1-Score. Dari ketiga algoritma tersebut, *Hist Gradient Boosting* memberikan kinerja terbaik dengan akurasi tertinggi dan presisi yang sangat tinggi. Namun, penting untuk mempertimbangkan faktor-

faktor lain seperti waktu eksekusi, kompleksitas model, dan ukuran dataset saat memilih algoritma yang paling sesuai untuk suatu tugas klasifikasi.

4. KESIMPULAN

Dalam penelitian ini, kami mengusulkan metode klasifikasi ensemble sampel malware berbasis windows. Sampel malware yang digunakan merupakan sampel malware PE yang terdiri dari 3 fitur utama yaitu *DOS_Header*, *Image_Header* dan *Optional_Header*. Klasifikasi dilakukan dengan menggunakan model-model ensemble yang terdiri dari *decision tree*, *random forest*, *Ada Boost*, *Bagging*, dan *Histogram-Based Gradient Boosting*. Dengan metode *ensemble*, algoritma *Histogram-based Gradient Boosting* memiliki capaian akurasi tertinggi sebesar 96,9%. Kedepan dapat dikembangkan penelitian lanjutan yang dapat dikaji kedalam beberapa fokus seperti pada seleksi fitur, variasi model maupun hyperparameter.

KESIMPULAN

Penulis mengucapkan terima kasih atas dukungan keuangan berikut untuk penelitian, kepenulisan, dan publikasi artikel ini yang didukung oleh Politeknik Negeri Jakarta dengan nomor hibah B.264/PL3.B/PN.003/2023.

DAFTAR PUSTAKA

- Ari Sandriana, Rianto, & Firmansyah Maulana. (2022). Klasifikasi serangan Malware terhadap Lalu Lintas Jaringan Internet of Things menggunakan Algoritma K-Nearest Neighbour (K-NN). *E-JOINT (Electronica and Electrical Journal Of Innovation Technology)*, 3(1), 12–22. <https://doi.org/10.35970/e-joint.v3i1.1559>
- Asry, D. W., Siswanto, E., Kurniawan, D., & ... (2023). Deteksi Malware Statis Menggunakan Deep Neural Networks Pada Portable Executable. *Teknik: Jurnal Ilmu ...*, 3(1), 19–34.
- Bastian, A. (2021). Improving Antivirus Signature For Detection Ransomware Attacks With Machine Learning. *Smart Comp :Jurnalnya Orang Pintar Komputer*, 10(1), 30–34. <https://doi.org/10.30591/smartcomp.v10i1.2190>
- Gupta, D., & Rani, R. (2020). Improving malware detection using big data and ensemble learning. *Computers and Electrical Engineering*, 86, 106729. <https://doi.org/10.1016/j.compeleceng.2020.106729>
- Koklu, M., Cinar, I., & Taspinar, Y. S. (2021). Classification of rice varieties with deep learning methods. *Computers and Electronics in Agriculture*, 187(June), 106285. <https://doi.org/10.1016/j.compag.2021.106285>
- Kumar, A. (n.d.). *ClAMP (Classification of Malware with PE headers)*.
- Liu, K., Xu, S., Xu, G., Zhang, M., Sun, D., & Liu, H. (2020). A Review of Android Malware Detection Approaches Based on Machine Learning. *IEEE Access*, 8, 124579–124607. <https://doi.org/10.1109/ACCESS.2020.3006143>
- Paza, L. S. (2023). *Malware Detection using Portable Executable Header and Gradient Boosting Classification Algorithm*. 916–924. <https://doi.org/10.46254/ap03.20220174>
- Prayitno, D. (2022). Systematic Literature Review: Implementasi Metode Statis Dan Dinamis Pada Analisa Malware. *Simetris*, 16(2), 53–57.
- Ramadhan, B., Purwanto, Y., & Ruriawan, M. F. (2020). Forensic malware identification using naive bayes method. *2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings*, 1–7. <https://doi.org/10.1109/ICITSI50517.2020.9264959>