JEPIN
(Jurnal Edukasi dan Penelitian Informatika)
ISSN(e): 2548-9364 / ISSN(p) : 2460-0741

Vol. 10
No. 1
April 2024

Deteksi Malware Ransomware Menggunakan Deep Neural Network

Benni Purnama[#1], Eko Arip Winanto[#2], Shairuppdin[#3], Ibnu Sani Wijaya[#4]

[#]*Departemen Ilmu Komputer, Universitas Dinamika Bangsa*
Jl. Jendral Sudirman No.1 Thehok Jambi Selatan
[1]bennipurnama@unama.ac.id
[2]*ekoaripwinanto@gmail.com*
[3]*sharipuddin@unama.ac.id*
[4]*ibnu_sw@unama.ac.id*

***Abstrak*— *Malware* pada perangkat *mobile android* menjadi sebuah tantangan yang perlu di perhatikan secara khusus. Mengingat akhir-akhir ini banyak kasus kejahatan dalam teknologi informasi dan komunikasi melalui *malware*. Sebuah *malware* ini bertujuan untuk mencuri, mengenkripsi, dan menghapus data sensitif kemudian mengubah atau membajak data dari sebuah perangkat pengguna. Oleh karena itu, pada penelitian ini bertujuan untuk mendeteksi *malware* jenis *ransomware* melalui system operasi *android* menggunakan metode *deep learning*. Metode yang diusulkan pada penelitian ini adalah *Deep Neural Network* (DNN). Dataset CIC-InvesAndMal2019 akan diujikan ke model hasil dari proses training DNN. Hasil pengujian model DNN menunjukkan bahwa DNN berhasil mendeteksi *malware ransomware* dengan tingkat akurasi mencapai 96.6 %.**

***Kata kunci*— *Malware, android, machine learning, Ransomware*, DNN**

I. PENDAHULUAN

Malware, atau malicious *software* adalah program atau file apa pun yang sengaja merusak komputer, jaringan, atau server. Jenis malware termasuk virus komputer, *worm, Trojan horse, ransomware, dan spyware*. *Malware* ini bertujuan untuk mencuri, mengenkripsi, dan menghapus data sensitive dengan cara mengubah atau membajak fungsi komputasi inti dan memantau aktivitas komputer pengguna[1]. Meskipun beragam dalam jenis dan kemampuannya, *malware* biasanya memiliki salah satu tujuan. Pertama, menyediakan remote control bagi penyerang untuk menggunakan mesin yang terinfeksi. Kedua, mengirim spam dari mesin yang terinfeksi ke target yang tidak dicurigai. Ketiga, mengeksplorasi jaringan lokal pengguna yang terinfeksi. Keempat, Mencuri data sensitif [2]. Dikarenakan dampak bahaya dari *malware* maka perlu adanya sistem deteksi yang handal untuk memonitoring prilaku dari *malware*.

Pada penelitian [3] telah mengusulkan metode *machine learning* untuk mendeteksi *malware* pada sistem operasi *android*. Pada penelitian [4-7] menunjukkan bahwa penerapan machine learning dapat diimplementasikan kedalam sistem deteksi *malware*. Terdapat beberapa penelitian sebelumnya yang sudah menggunakan *machine learning*. Pada penelitian [4] mengusulkan metode berbasis *machine learning* untuk mendeteksi *malware*. Selanjutnya [5][6][7] telah mengusulkan metode *machine learning* pada sistem deteksi *malware* pada *android* dan hasilnya menunjukkan performa yang cukup memuaskan. Selain itu pada [8] melakukan perbandingan metode untuk mendetksi dari metode *Support Vector Machine* (SVM), *Naïve Bayes, Decision Tree, Random Forest, Log Regression*, dan *K-nearest Neighbor* (KNN), hasil deteksi menunjukan bahwa hasil akurasi deteksinya masih kurang memuaskan hanya mencapai 76%. Metode *machine learning* terbaru yang memiliki potensi untuk digunakan pada sistem deteksi *malware* adalah *deep learning* [9].

Salah satu metode *deep learning* adalah *Deep Neural Network* (DNN) yang mampu belajar secara otomatis dari data yang diberikan [10]. Model DNN dapat menyesuaikan bobot dan parameter internal model sendiri untuk meningkatkan kinerja dan mengoptimalkan deteksi *malware* [11]. Ini memungkinkan model DNN untuk menjadi fleksibel dan beradaptasi dengan perubahan lingkungan dan mempelajari pola dari input yang merujuk pada *malware* tertentu, serta menghasilkan output yang sesuai [12][13]. Oleh karena itu, penelitian ini melakukan deteksi *malware* pada *android* menggunakan metode *machine learning* yaitu DNN.

Penelitian ini memiliki tujuan untuk mendeteksi akurasi dan efektivitas *malware* pada sistem operasi *Android* dengan cara meningkatkan kinerja dari sistem deteksi *malware* pada *android* menggunakan DNN. Terakhir, pada penelitian ini mengusulkan metode *machine learning* yaitu DNN sebagai metode untuk sistem

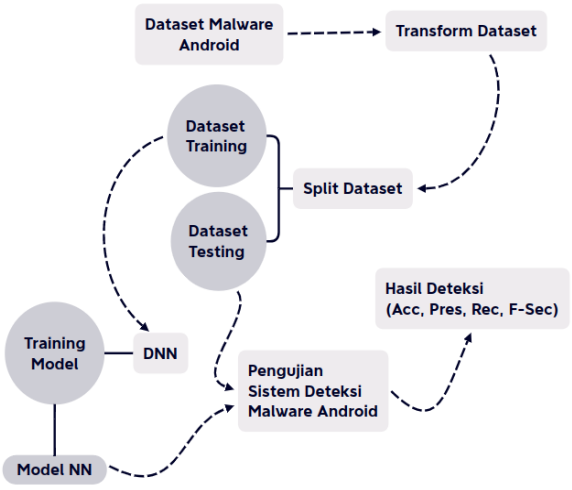
deteksi pada *android malware* jenis *ransomware*. Adapun kontribusi pada penelitian ini yaitu: mengusulkan sistem deteksi pada *malware android* jenis *ransomware* menggunakan metode DNN dengan real dataset *update*.

II. METODE PENELITIAN

Penelitian ini disusun sebagai berikut. Bagian pertama menjelaskan *Experiment setup*, kemudian bagian kedua menjelaskan tentang dataset yang digunakan, bagian tiga tentang metode deteksi yang digunakan dan bagian keempat adalah *Envorinment setup*.

A. *Experiment Setup*

Dalam upaya untuk mendeteksi dari ancaman *malware* pada *android*, penelitian ini mengusulkan metode *Deep Neural Network* (DNN). Dengan mengikuti alur penelitian yang terstruktur, yang dapat diilustrasikan dalam Gambar 1.



Gambar 1. *Experiment Setup*

Penelitian dilakukan dengan mengikuti *setup* eksperimen yang dirancang secara khusus, yang ditunjukkan dalam Gambar 1. Penelitian ini dibagi menjadi tiga tahap penting yang dirinci sebagai berikut:

a. Tahap pertama melibatkan transformasi fitur dari dataset *malware Android* dengan tujuan menyederhanakan nilai dataset. Selain itu, dataset juga dibagi menjadi data pelatihan dan data pengujian.
b. Tahap kedua adalah melatih model *Deep Neural Network* (DNN) untuk mendeteksi jenis *ransomware* dalam *malware Android*. Dalam tahap ini, model DNN dilatih sehingga menghasilkan bobot dan bias yang akan digunakan dalam proses pengujian.
c. Tahap terakhir adalah pengujian menggunakan model DNN yang telah dilatih. Model ini diuji menggunakan dataset pelatihan, dan performa model dievaluasi menggunakan metrik-metrik seperti akurasi, presisi, dan recall.

B. *Dataset*

Penelitian ini menggunakan dataset *malware* yang dikenal sebagai CIC-InvesAndMal2019 [14], yang dikembangkan oleh *University of New Brunswick*. Dataset ini terdiri dari berbagai kelas *malware*, *termasuk adware*, *ransomware*, *scareware*, dan *sms*. Lebih spesifik lagi, dataset CIC-InvesAndMal2019 mencakup beberapa varian *malware android* yang signifikan, seperti *charger, jisut, koler,* dan *ransombo.* Pada penelitian ini akan fokus pada jenis *android malware* kelas *ransomware*. Informasi lebih rinci tentang jenis-jenis *malware* ini dapat ditemukan dalam Tabel 1. Dataset ini menjadi dasar analisis dan penelitian dalam upaya memahami perilaku dan karakteristik *malware* dengan lebih baik.

TABEL I
JENIS JENIS DATASET

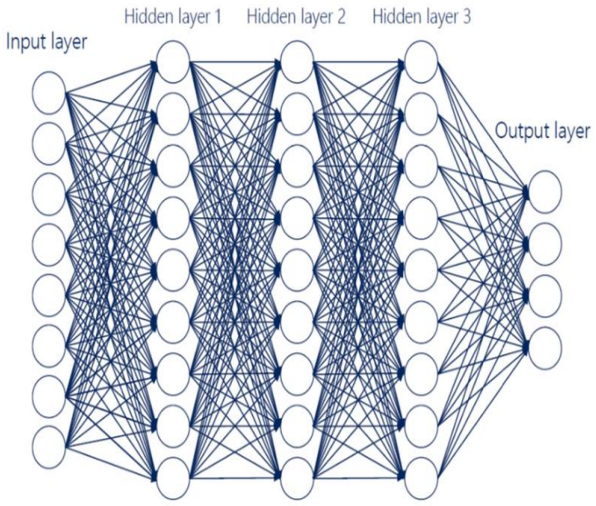
No	Tipe *Ramsomware*	Jumlah Paket
1	*Charger family*	4772
2	*Jisut family*	1861
3	*Koler family*	4214
4	*LockerPin family*	4576
5	*Simplocker family*	4346
6	*Pletor family*	505
7	*PornDroid family*	4051
8	*RansomBO family*	3579
9	*Svpeng family*	5219
10	*WannaLocker family*	2787

C. *Metode Deteksi* Android Malware

Dalam penelitian ini, kami menggunakan *Deep Neural Network* (DNN) sebagai metode untuk mendeteksi serangan dari *Android malware*. DNN merupakan salah satu teknik yang kuat dalam mempelajari pola-pola kompleks dari data. Dataset yang terdiri dari sampel *malware* dan *non-malware* yang berkaitan dengan sistem operasi *Android*. *Non-malware* atau *benign* merupakan program atau file yang tidak merusak pada system computer. Selanjutnya adalah melakukan proses pelatihan dan kemudian merancang dan melatih model DNN dengan menggunakan arsitektur yang terdiri dari beberapa lapisan untuk mempelajari pola-pola yang terkandung dalam data. Proses pelatihan model melibatkan pengoptimalan bobot dan bias melalui algoritma *backpropagation*. pada gambar 2 adalah arsitektur dari model DNN dan pada tabel 2 adalah pengaturan *environment* dari model DNN pada penelitian ini.

D. *Environment Setup*

Pada penelitian ini memperlukan perangkat hardware dan software yang digunakan untuk sistem deteksi pada *android malware*. Adapun hardware yang digunakan untuk menjalankannya berupa komputer dengan spesifikasi prosesor Intel Core i7 dengan RAM 2,60 GHz 12 GB, yang menjalankan Ubuntu 20.04.3 LTS sebagai Sistem Operasi. Untuk tujuan perangkat lunak menggunakan *python3* dan *PyCharm* IDE, Keras dan *scikit-learn* untuk proses *machine learning*.



Gambar 2. Arsitektur DNN [15]

TABEL II
ENVIRONMENT DNN

Layer Input	*100 node, Activation (relu)*
Hidden Layer	*80 node, Activation (relu)*
Layer Output	*11 node, Activation (sigmoid)*
Batch size	*16*
Epoch	*50*
Encoding	*One-hot Encoding*

III. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil percobaan yang dilakukan untuk menguji kinerja algoritma *Deep Neural Network* (DNN) dalam deteksi *Android malware* kelas *ransomware*. Pengujian kinerja dilakukan dengan menggunakan dataset *malware android ransomware* dan *non-malware*. Dalam pengujian ini, dilakukan evaluasi menggunakan metrik-metrik seperti akurasi, presisi, recall, dan F1-score dengan menggunakan fungsi-fungsi seperti berikut ini.

		Predicted Class	
		P	***N***
Actual Class	***P***	**TP** $\lambda_{PP}m_P$	**FN** $(1-\lambda_{PP})m_P$
	N	**FP** $(1-\lambda_{NN})m_N$	**TN** $\lambda_{NN}m_N$

Gambar 3. *Confusion Matrix* [16]

Adapun persamaan *Confusion matrix* yang digungakan adalah sebagai berikut :

$$Akurasi = \frac{TP+TN}{TP+FN+TN+FP}$$

$$Presisi = \frac{TP}{TP+FP}$$

$$Recall = \frac{TN}{TN+FP}$$

$$F1\ Score = 2\ x\ \frac{presisi\ x\ Recall}{presisi\ +\ Recall}$$

. Hasil pengujian menunjukkan bahwa algoritma DNN mampu mengenali dan membedakan dengan baik antara malware ransomware dan non-malware.

E. Hasil Pengujian Deteksi

Proses deteksi yang melibatkan penggunaan bobot dan bias yang telah didapatkan pada proses pembelajaran. Data input yang berupa fitur-fitur dari *malware android* akan diproses melalui jaringan DNN untuk menghasilkan keluaran yang menunjukkan kemungkinan adanya *malware*. Nilai keluaran ini kemudian dapat diinterpretasikan sebagai prediksi deteksi *malware* atau *non-malware*.

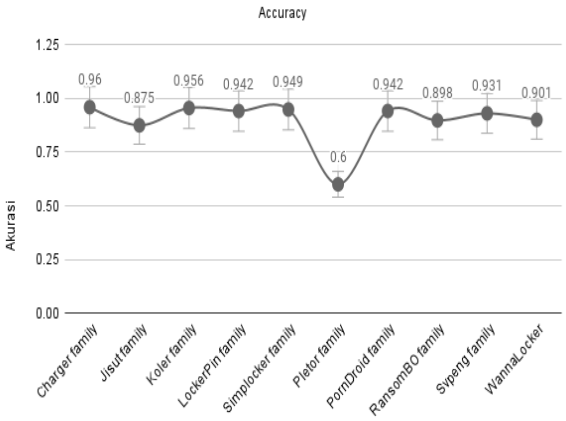
Dalam penelitian ini, dilakukan dengan menggunakan metode DNN. Melalui tahap pembelajaran dan proses deteksi, DNN dapat mempelajari pola-pola yang terkandung dalam data dan mampu mengidentifikasi serangan malware dengan akurat dan cepat. Pada tabel 3 adalah hasil dari pengujian deteksi *android malware* menggunakan metode DNN dengan 10 jenis *malware android* dan program *non-malware*.

TABEL III
HASIL PENGUJIAN DARI JENIS DATASET

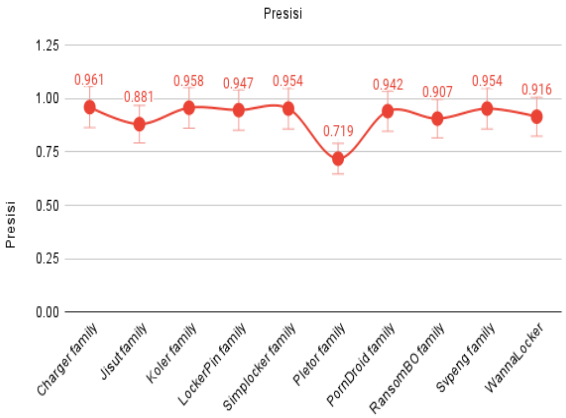
No	*Tipe Ramsomware*	Akurasi	Presisi	Recall	F-Score
1	*Charger family*	0.960	0.961	0.962	0.962
2	*Jisut family*	0.875	0.881	0.875	0.878
3	*Koler family*	0.956	0.958	0.956	0.957
4	*LockerPin family*	0.942	0.947	0.952	0.949
5	*Simplocker family*	0.949	0.954	0.949	0.952
6	*Pletor family*	0.600	0.719	0.882	0.792
7	*PornDroid family*	0.942	0.942	0.944	0.943
8	*RansomBO family*	0.898	0.907	0.939	0.923
9	*Svpeng family*	0.931	0.954	0.932	0.943
10	*WannaLocker family*	0.901	0.916	0.906	0.911

Hasil dari pengujian menunjukkan akurasi tertinggi deteksi yang cukup memuaskan dengan tingkat akurasi mencapai 96 %. Akan tetapi ada satu malware Pletor family hanya mendapat akurasi 60 %. Ini terjadi dikarena

sebaran data *malware Pletor family* kemungkinan mirip dengan data *non-malware*. Hasil pengujian terdiri dari 4 parameter yaitu akurasi, presisi, recall dan F-*score*. Hasil rata-rata akurasi pada pengujian deteksi menggunakan DNN pada jaringan kompleks *Internet of Things* (IoT) mencapai 89,54 %. Selain itu pada parameter presisi dan *recall* memperoleh hasil yang cukup memuaskan, yaitu 96,1% untuk presisi dan 96,2% untu *recall*.

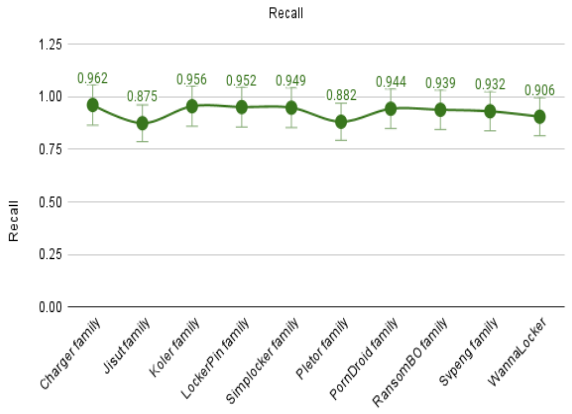


Gambar 3. *Experiment Setup Accuracy*



Gambar 4. *Experiment Setup* Presisi

Hasil pengujian yang diperoleh pada setiap parameter pengukuran dapat dilihat pada gambar 3 untuk akurasi, gambar 4 untuk presisi, gambar 5 untuk *recall* dan gambar 6 untuk f-*score*. Dari setiap pengukuran menunjukkan bahwa hasil terbaik diperoleh pada jenis deteksi *Charger family* yang mecapai 96.6 %. Sedangkan untuk hasil yang kurang memuaskan ditunjukkan pada deteksi malware pada kelas *Pletor family* yang hanya mencapai 60 %. Kesimpulan akhir dari penelitian ini adalah metode DNN menunjukkan bahwa dapat mendeteksi dengan cukup baik.



Gambar 5. *Experiment Setup Recall*

Gambar 6. Experiment Setup F-Score

IV. KESIMPULAN

Keamanan perangkat menggunakan system operasi *android* menjadi sebuah tantangan tersendiri dan menjadi perhatiaan khusus baik industri maupun akademisi. Pada penelitian ini berfokus untuk mendeteksi *malware android* jenis *ransomware* menggunakan metode *deep learning*. Metode yang diusulkan pada penelitian ini adalah *deep neural network* (DNN). Model hasil proses traning DNN diujikan pada dataset CIC-InvesAndMal2019. Hasil pengujian menunjukkan bahwa metode DNN dapat mendeteksi *malware Charger family* dengan tingkat akurasi mencapai 96.6 %. Penelitian kedepan akan mencoba menerapkan metode pemilihan fitur dan variasi jenis *malware* yang lebih banyak.

REFERENSI

[1] M. S. Babak Bashari Rad, Mohammad Kazem Hassan Nejad, "Malware Classification And Detection Using Artificial Neural Network A Literature Review," J. Eng. Sci. Technol., no. 7, pp. 14–23, 2018.

[2] Y. Du, C. Liu, and Z. Su, "Detection and Suppression of Malware Based on Consortium Blockchain," IOP Conf. Ser. Mater. Sci.

Eng., vol. 490, no. 4, 2019, doi: 10.1088/1757-899X/490/4/042031.

[3] H. Soni, P. Arora, and D. Rajeswari, "Malicious Application Detection in Android using Machine Learning," Proc. 2020 IEEE Int. Conf. Commun. Signal Process. ICCSP 2020, pp. 846–848, 2020, doi: 10.1109/ICCSP48568.2020.9182170.

[4] A. H. El Fiky, A. Elshenawy, and M. A. Madkour, "Detection of Android Malware using Machine Learning," 2021 Int. Mobile, Intelligent, Ubiquitous Comput. Conf. MIUCC 2021, pp. 9–16, 2021, doi: 10.1109/MIUCC52538.2021.9447661.

[5] H. Haidros Rahima Manzil and S. Manohar Naik, "DynaMalDroid: Dynamic Analysis-Based Detection Framework for Android Malware Using Machine Learning Techniques," IEEE Int. Conf. Knowl. Eng. Commun. Syst. ICKES 2022, pp. 1–6, 2022, doi: 10.1109/ICKECS56523.2022.10060106.

[6] W. Wang, M. Zhao, and J. Wang, "Effective android malware detection with a hybrid model based on deep autoencoder and convolutional neural network," J. Ambient Intell. Humaniz. Comput., vol. 0, no. 0, pp. 1–9, 2018, doi: 10.1007/s12652-018-0803-6.

[7] M. Anshori, F. Mar'i, and F. A. Bachtiar, "Comparison of Machine Learning Methods for Android Malicious Software Classification based on System Call," Proc. 2019 4th Int. Conf. Sustain. Inf. Eng. Technol. SIET 2019, pp. 343–348, 2019, doi: 10.1109/SIET48054.2019.8985998.

[8] A. K. T. Lee Yam, J. M. R. Ballesta, J. A. H. Lanceta, M. K. T. Mogol, and R. Labanan, "Hybrid Android Malware Detection Model using Machine learning Algorithms," Proc. - 2022 2nd Int. Conf. Inf. Comput. Res. iCORE 2022, pp. 66–71, 2022, doi: 10.1109/iCORE58172.2022.00032.

[9] D. Arivudainambi, V. K. K. A, S. C. S, and P. Visu, "Malware traffic classification using principal component analysis and artificial neural network for extreme surveillance," Comput. Commun., vol. 147, no. June, pp. 50–57, 2019, doi: 10.1016/j.comcom.2019.08.003.

[10] R. Wang, J. Zheng, Z. Shi, and Y. Tan, "Detecting Malware Using Graph Embedding and DNN," Proc. - 2022 Int. Conf. Blockchain Technol. Inf. Secur. ICBCTIS 2022, pp. 28–31, 2022, doi: 10.1109/ICBCTIS55569.2022.00018.

[11] M. Gullu and N. Barisci, "Android Malware Classification with Gray Wolf Optimization Algorithm and Deep Neural Network Hybrid Approach," 2022 30th Signal Process. Commun. Appl. Conf. SIU 2022, pp. 13–16, 2022, doi: 10.1109/SIU55565.2022.9864822.

[12] B. Vasu and N. Pari, "Combining Multimodal DNN and SigPid technique for detecting Malicious Android Apps," Proc. 11th Int. Conf. Adv. Comput. ICoAC 2019, pp. 289–294, 2019, doi: 10.1109/ICoAC48765.2019.247134.

[13] M. K. Alzaylaee, S. Y. Yerima, and S. Sezer, "DL-Droid: Deep learning based android malware detection using real devices," Comput. Secur., vol. 89, p. 101663, 2020, doi: 10.1016/j.cose.2019.101663.

[14] L. Taheri, A. F. A. Kadir, and A. H. Lashkari, "Extensible android malware detection and family classification using network-flows and API-calls," Proc. - Int. Carnahan Conf. Secur. Technol., vol. 2019-October, no. Cic, 2019, doi: 10.1109/CCST.2019.8888430.

[15] M. Merenda, C. Porcaro, and D. Iero, "Edge Machine Learning for AI-Enabled IoT Devices: A Review," Sensors, vol. 20, p. 2533, 2020, doi: 10.3390/s20092533.