DETEKSI *MALWARE* TROJAN PADA LALU LINTAS JARINGAN *REVERSE* TCP DENGAN ALGORITMA *DECISION TREE*

SKRIPSI

Diajukan Untuk Melengkapi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer

OLEH :

RIRIN FEBRIANA
09011282126051

SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA
TAHUN 2025

HALAMAN PENGESAHAN

SKRIPSI

DETEKSI *MALWARE* TROJAN PADA LALU LINTAS JARINGAN *REVERSE* TCP DENGAN ALGORITMA *DECISION TREE*

Sebagai salah satu syarat untuk penyelesaian studi
di Program Studi S1 Sistem Komputer

OLEH:

RIRIN FEBRIANA

09011282126051

Pembimbing 1 : **Prof. Ir. Deris Stiawan, M.T., Ph.D.**
NIP. 197806172006041002

Pembimbing 2 : **Nurul Afifah, M.Kom**
NIP. 199211102023212049

Mengetahui,
Ketua Program Studi Sistem Komputer

Dr. Ir. H. Sukemi, M.T.

NIP. 196612032006041001

AUTHENTICATION PAGE

SKRIPSI

DETECTION OF TROJAN MALWARE IN REVERSE TCP NETWORK TRAFFIC USING THE DECISION TREE ALGORITHM

As one of the requirements for completing studies
in the S1 Computer System Study Program

BY:

RIRIN FEBRIANA

09011282126051

Advisor 1	**: Prof. Ir. Deris Stiawan, M.T., Ph.D.** **NIP. 197806172006041002**
Advisor 2	**: Nurul Afifah, M.Kom** **NIP. 199211102023212049**

Approved by,

Head of the Computer Systems Departement

Dr. Ir. H. Sukemi, M.T.

NIP. 196612032006041001

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada :

Hari : Jumat

Tanggal : 13 Juni 2025

Tim Penguji :

1. **Ketua : Dr. Rossi Passarella, M. Eng**
2. **Penguji : Dr. Ir. Ahmad Heryanto, M. T.**
3. **Pembimbing I : Prof. Ir. Deris Stiawan, M. T., Ph. D.**
4. **Pembimbing II : Nurul Afifah, M. Kom.**

Mengetahui, 24/6/25

Ketua Jurusan Sistem Komputer

Dr. Ir. H. Sukemi, M. T.

NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Ririn Febriana

NIM : 09011282126051

Judul : Deteksi *Malware* Trojan Pada Lalu Lintas Jaringan *Reverse* TCP Dengan Algoritma *Decision Tree*.

Hasil Pengecekan Software Turnitin : 2%

Menyatakan bahwa laporan tugas akhir saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.

ya, Juni 2025

Ririn Febriana

NIM. 09011282126051

KATA PENGANTAR

Assalamualaikum wr. wb.

Puji dan syukur kami panjatkan ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul *"Deteksi Malware Trojan pada Lalu Lintas Jaringan Reverse TCP dengan Algoritma Decision Tree"*.

Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada program Sarjana Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya. Penelitian ini bertujuan untuk menganalisis dan mengembangkan metode deteksi malware yang berjalan pada lalu lintas jaringan *Reverse TCP* dengan menggunakan algoritma *Decision Tree*. Dengan penelitian ini, diharapkan dapat memberikan kontribusi dalam bidang keamanan siber, khususnya dalam deteksi dini ancaman siber yang semakin kompleks di era digital.

Selama proses penyusunan tugas akhir ini, penulis mendapatkan banyak bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan penuh rasa hormat dan terima kasih, penulis ingin menyampaikan penghargaan kepada:

1. Allah SWT yang telah memberikan nikmat kesehatan, kekuatan, dan kesempatan dalam menyelesaikan tugas akhir ini.
2. Kedua orang tua yang selalu memberikan doa, dukungan, dan motivasi tanpa henti.
3. Bapak Prof. Dr. Erwin, S.Si., M.Si. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. H. Sukemi, M.T. selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Kepada Bapak Prof. Ir. Deris Stiawan, M. T., Ph.D. selaku Dosen Pembimbing yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses penelitian dan penyusunan tugas akhir ini.

6. Kepada Ibu Nurul Afifah, M. Kom selaku Dosen Pembimbing II yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses penelitian dan penyusunan tugas akhir ini.
7. Rekan-rekan mahasiswa Sistem Komputer angkatan 2021 yang telah berbagi ilmu, pengalaman, dan motivasi dalam perjalanan akademik ini.
8. Teman-teman yang selalu mendukung, mendengarkan keluh kesah, serta memberikan semangat dalam menyelesaikan tugas akhir ini.
9. Untuk teman satu riset Nabila Sintia dan Viona Aulia Meidy yang sudah bersedia bimbingan bersama, penulis ucapkan beribu kalimat terima kasih karena selalu men-support penulis dan sering memberi saran serta masukan dalam menyelesaikan laporan ini.
10. Semua pihak yang tidak dapat disebutkan satu per satu yang telah berkontribusi dalam berbagai bentuk, baik langsung maupun tidak langsung, dalam proses penyusunan tugas akhir ini.

Penulis menyadari bahwa tugas akhir ini masih jauh dari sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun agar penelitian ini dapat lebih bermanfaat dan terus berkembang. Semoga tugas akhir ini dapat memberikan manfaat bagi pembaca, khususnya bagi mahasiswa dan peneliti yang tertarik dalam bidang keamanan jaringan dan analisis lalu lintas jaringan.

Akhir kata, penulis mengucapkan terima kasih dan berharap semoga tugas akhir ini dapat menjadi referensi yang bermanfaat dalam pengembangan penelitian lebih lanjut. Wassalamualaikum wr. wb.

Indralaya 2025

Penulis

Ririn Febriana

0901128212605

DETEKSI *MALWARE* TROJAN PADA LALU LINTAS JARINGAN *REVERSE* TCP DENGAN ALGORITMA *DECISION TREE*

RIRIN FEBRIANA (09011282126051)

Jurusan Sistem Komputer, Fakultas Ilmu Komputer

Universitas Sriwijaya

Email : 09011282126051@student.unsri.ac.id

ABSTRAK

Malware merupakan singkatan dari *malicious software*, yaitu perangkat lunak berbahaya yang dirancang untuk merusak, mencuri data, atau mengganggu sistem komputer dan jaringan. Salah satu jenis malware yang sering ditemui adalah *Trojan*, yang biasanya menyamar sebagai program aman namun memiliki tujuan merusak. Untuk mengatasi ancaman ini, dibutuhkan sistem yang mampu mendeteksi pola-pola mencurigakan dalam lalu lintas jaringan. Penelitian ini bertujuan untuk menjawab tiga pertanyaan utama, yaitu bagaimana proses ekstraksi data dari file PCAP, seberapa efektif metode *Decision Tree* dalam mendeteksi *malware Trojan*, serta bagaimana cara meningkatkan performa model deteksi. Proses ekstraksi data dilakukan dengan memanfaatkan *CICFlowMeter* untuk mengubah file PCAP menjadi format CSV yang berisi fitur-fitur aliran data jaringan. Data tersebut kemudian dianalisis menggunakan metode *Machine Learning*, khususnya algoritma *Decision Tree*, guna mengklasifikasikan antara trafik normal dan berbahaya.

Hasil penelitian menunjukkan bahwa metode *Decision Tree* efektif dalam mengidentifikasi aktivitas malware pada jaringan perangkat *mobile*. Performa terbaik diperoleh saat rasio data pelatihan, validasi dan pengujian adalah 25:25:50 untuk sebelum seleksi fitur, dengan akurasi 93,13% dan *fi-score* sebesar 92,89%, setelah seleksi fitur dengan rasio 40:40:20 dengan akurasi 97,91% dan *fi-score* sebesar 97,89%. Selain itu, penerapan sistem deteksi intrusi Snort turut memperkuat deteksi serangan dengan mengenali pola-pola lalu lintas berbahaya secara berbasis aturan. Seleksi fitur berperan penting dalam meningkatkan performa model, mengurangi *overfitting*, serta memastikan generalisasi yang lebih baik. Optimalisasi kedalaman pohon keputusan juga membantu menjaga keseimbangan antara bias dan varians dalam model.

Kata kunci: *Malware, Trojan, PCAP, Decision Tree, CICFlowMeter, Snort*, Deteksi Intrusi, Seleksi Fitur, Akurasi, *F1-Score*.

DETECTION OF TROJAN MALWARE IN REVERSE TCP NETWORK TRAFFIC USING THE DECISION TREE ALGORITHM

RIRIN FEBRIANA (09011282126051)

Department of Computer Systems, Faculty of Computer Science

Sriwijaya University

Email : 09011282126051@student.unsri.ac.id

ABSTRACT

Malware, short for malicious software, is a type of harmful software designed to damage, steal data from, or disrupt computer systems and networks. One of the most common types of malware is the Trojan, which typically disguises itself as a legitimate program but actually has malicious intent. To address this threat, a system is needed that can detect suspicious patterns in network traffic. This study aims to answer three main questions: how data is extracted from PCAP files, how effective the Decision Tree method is in detecting Trojan malware, and how to improve the performance of the detection model. The data extraction process was carried out using CICFlowMeter, which converts PCAP files into CSV format containing flow-based features of network traffic. The resulting data was then analyzed using Machine Learning methods, specifically the Decision Tree algorithm, to classify traffic as either normal or malicious.

The results show that the Decision Tree method is effective in identifying malware activity on mobile network devices. The best performance before feature selection was achieved with a 25:25:50 training+validation-to-testing ratio, reaching an accuracy 93,15% and F1-score of 92.89%. After feature selection, the highest performance was obtained with a 40:40:20 ratio, achieving an accuracy 97,89% and F1-score of 97.89%. In addition, the implementation of the Snort intrusion detection system enhanced the detection process by recognizing attack patterns in the network traffic based on predefined rules. Feature selection played a crucial role in improving model performance by reducing overfitting and ensuring better generalization. Furthermore, optimizing the depth of the decision tree helped maintain a balance between bias and variance in the model.

***Keywords**: Malware, Trojan, PCAP, Decision Tree, CICFlowMeter, Snort, Intrusion Detection, Feature Selection, Accuracy, F1-Score.*

DAFTAR ISI

HALAMAN PENGESAHAN ..... **ii**
AUTHENTICATION PAGE ..... **iii**
HALAMAN PERSETUJUAN ..... **iv**
HALAMAN PERNYATAAN ..... **v**
KATA PENGANTAR ..... **vi**
ABSTRAK ..... **viii**
ABSTRACT ..... **ix**
BAB I PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan penelitian 4
1.5 Manfaat penelitian 4
1.6 Metodologi Penelitian 5
1.7 Sistematika Penulisan 6
BAB II TINJAUAN PUSTAKA **7**
2.1 Penelitian Terkait 7
2.2 *Android* 16
2.3 *Malware* 17
2.3.1 Trojan 17
2.3.2 Deteksi *Malware* 18
2.4 *Intrusion Detection System* 19
2.4.1 *Snort* 19
2.5 Protocol TCP 20
2.5.1 *Reverse TCP* 20
2.5.2 Metasploit 21
2.6 *Cicflowmeter* 22
2.7.1 *Decision tree* 24
2.8 Metrik Evaluasi 25
2.8.1 *Recall* 25
2.8.2 *Precision* 25
2.8.3 Akurasi 26

2.8.4 *F1-Score* 26
2.8.5 *False Positive Rate (FPR*) 27
BAB III METODOLOGI PENELITIAN 28
3.1 Pendahuluan 28
3.2 Diagram Alur Penelitian 28
3.3 Perangkat Keras dan Lunak 29
3.3.1 Perangkat Keras 29
3.3.2 Perangkat Lunak 29
3.4 Skenario Topologi 29
3.5 *Pre-processing* 31
3.5.1 Ekstraksi dataset 31
3.6 Dataset 32
3.7 Proses membuat label *Malware* 34
3.8 *Exploratory Data Analysis* 35
3.9 Data encoding 35
3.9.1 *Encode categorical target variable* 35
3.9.2 *Encode categorical columns to numeric* 36
3.10 Pisahkan fitur (X) dan label (y) 36
3.11 Seleksi Fitur 37
3.12 Normalisasi Data 37
3.13 Resampling dengan *SMOTE* 38
3.14 *Data Testing, data Training* dan *data Validation* 39
3.15 Hyperparameter Tuning 39
3.16 Implementasi Algoritma Decision tree 42
3.17 Evaluasi Model 43
BAB IV HASIL DAN PEMBAHASAN 44
4.1 Pendahuluan 44
4.2 Proses ekstraksi PCAP ke CSV 44
4.3 Proses membuat label *Malware* 47
4.3.1 Persiapan data 47
4.3.2 Proses pelabelan 48
4.3.3 Pelabelan Data 49
4.4 *Exploratory Data Analysis* 50
4.5 Data *Encoding* 52

4.5.1 *Encode categorical target variable* 52
4.5.2 *Encode categorical columns to numeric* 55
4.6 Pisahkan fitur (X) dan label (y) 57
4.7 Seleksi fitur 58
4.8 Data Normalisasi 59
4.9 Resampling dengan *SMOTE* 60
4.10 Split data 61
4.11 *Hyperparameter tuning* 61
4.12 Implementasi metode *Decision tree* 62
4.12.1 Split 10% : 45% :45% 62
4.12.2 Split 20% : 40% :40% 63
4.12.3 Split 30% : 35% :35% 64
4.12.4 Split 40% : 30% : 30% 66
4.12.5 Split 50% : 25% :25% 67
4.13 Evaluasi model 68
4.13.1 Perbandingan akurasi dan F1-Score 68
4.13.2 Validasi dengan *confusion matrix* 70
4.13.3 Pohon Keputusan 76
BAB V PENUTUP 87
5.1 Kesimpulan 87
5.2 Saran 88
DAFTAR PUSTAKA 89

DAFTAR GAMBAR

Gambar 3.2. 1 Flowchart alur penelitian 28
Gambar 3.4 1 Gambar Topologi 30
Gambar 3.5 1 Flowchart proses ekstraksi file PCAP ke CSV 31
Gambar 3.7. 1 Flowchart proses pelabelan..... 34
Gambar 3.9.1. 1 Flowchart proses Encoding..... 36
Gambar 3.13 1 Flowchart proses SMOTE 38
Gambar 3.16 1 Flowchart proses Algoritma Decision tree 42
Gambar 4.2 1 baris serangan dari file PCAP Victim reverse TCP 44
Gambar 4.2 2 Hasil snort yang dicoba 45
Gambar 4.2 3 proses mengatur kemana file CSV akan disimpan 46
Gambar 4.2 4 proses file PCAP yang sudah diekstrak ke CSV 46
Gambar 4.2 5 beberapa baris pertama file CSV Victim reverse TCP 47
Gambar 4.3.1. 1 Beberapa baris pertama data sebelum dilabel.....48
Gambar 4.3.2. 1 Kriteria malware sesuai snort 49
Gambar 4.3.2. 2 Label berdasarkan Dst port 49
Gambar 4.3.3. 1 beberapa data awal setelah dilabel.....50
Gambar 4.3.3. 2 jumlah hasil label 50
Gambar 4.4.1. 1 Seluruh kolom..... 51
Gambar 4.5.1 1 5 baris pertama sebelum data encoding 53
Gambar 4.5.1 2 5 baris pertama setelah data encoding 54
Gambar 4.5.2.1 1 Data sebelum Encode categorical columns to numeric 55
Gambar 4.5.2.2 1 Data setelah Encode categorical columns to numeric 56
Gambar 4.6 1 Fitur x 57
Gambar 4.6 2 Label y 57
Gambar 4.7 1 Fitur setelah seleksi fitur..... 58
Gambar 4.8 1 5 data pertama setelah normalisasi 59
Gambar 4.9 1 Sebelum diterapkan SMOTE..... 60
Gambar 4.9 2 Setelah diterapkan SMOTE 60
Gambar 4.11 1 Hyperparameter tunning yang diterapkan 61
Gambar 4.13 1 Grafik F1-Score Sebelum Seleksi Fitur..... 69
Gambar 4.13 2 Grafik F1-Score Setelah Seleksi Fitur 69

Gambar 4.13 3 Confusion matrix sebelum seleksi fitur 71
Gambar 4.13 4 Confusion matrix setelah seleksi fitur 73
Gambar 4.13 5 Pohon keputusan sebelum Seleksi Fitur 76
Gambar 4.13 6 Pohon keputusan setelah Seleksi Fitur 79

DAFTAR TABEL

Tabel 2. 1 Studi literatur 7
Tabel 3.6 1 Kolom Dataset 32
Tabel 3.15 1 Hyperparameter tunning yang dicoba 40
Tabel 4.10 1 pembagian data testing dan training yang dicoba 61
Tabel 4.12.1.1 1akurasi dan f1-score split 10:45:45 sebelum seleksi fitur 62
Tabel 4.12.1.1 2 akurasi dan f1-score split 10:45:45 sesudah seleksi fitur 62
Tabel 4.12.2.1 1 akurasi dan f1-score split 20:40:40 sebelum seleksi fitur 63
Tabel 4.12.2.1 2 akurasi dan f1-score split 20:40:40 sesudah seleksi fitur 63
Tabel 4.12.3.1 1 akurasi dan f1-score split 30:35:35 sebelum seleksi fitur 64
Tabel 4.12.3.1 2 akurasi dan f1-score split 30:35:35 setelah seleksi fitur 65
Tabel 4.12.4.1 1 akurasi dan f1-score split 40:30:30 sebelum seleksi fitur 66
Tabel 4.12.4.1 2 akurasi dan f1-score split 40:30:30 setelah seleksi fitur 66
Tabel 4.12.5.1 1 akurasi dan f1-score split 50:25:25 sebelum seleksi fitur 67
Tabel 4.12.5.1 2 akurasi dan f1-score split 50:25:25 setelah seleksi fitur 67
Tabel 4.13 1 Perbandingan akurasi dan f1-score 68

BAB I

PENDAHULUAN

1.1 Latar Belakang

Android adalah sistem operasi yang dibangun di atas *Linux* dan, berdasarkan penelitian [1] dibuat khusus untuk perangkat *mobile* seperti ponsel pintar dan tablet, sehingga pengguna dapat menggunakan berbagai aplikasi dan memanfaatkan fitur-fitur perangkat tersebut. *Android* adalah salah satu sistem operasi yang paling banyak digunakan pada *smartphone*, yang menjadikannya target utama bagi para peretas dan penyerang. Dengan menyisipkan kode berbahaya ke dalam aplikasi *Android* dengan cara yang sangat canggih, berdasarkan penelitian [2] untuk dapat mendeteksi dan mengidentifikasi aplikasi sebagai malware menjadi tantangan besar bagi penyedia keamanan. *Malware Android* telah berkembang semakin pintar dan sulit dideteksi dengan metode biasa. Saat ini metode berbasis pembelajaran mesin yang dilakukan pada penelitian [3], muncul sebagai cara yang lebih efektif untuk menghadapi ancaman *Android* yang semakin rumit. Metode ini bekerja dengan mengenali pola aktivitas *malware* yang ada dan menggunakan informasi tersebut untuk membedakan ancaman yang sudah dikenal dari ancaman baru yang perilakunya belum diketahui.

Dalam penelitian [4] *Malware (Malicious software)* dapat diartikan sebagai jenis perangkat lunak berbahaya yang dibuat untuk mengganggu, menghentikan aktivitas, mengumpulkan data pribadi tanpa izin, mengakses sistem secara ilegal, dan melakukan tindakan yang merugikan. Seiring dengan kemajuan teknologi informasi yang cepat, jumlah *malware* yang terus bertambah menjadi salah satu ancaman terbesar bagi keamanan komputer. Pada penelitian [4] mendeteksi perangkat lunak berbahaya semakin sulit karena jumlah dan jenis aplikasi dalam keamanan komputer terus bertambah. Keamanan jaringan komputer menjadi salah satu perhatian utama di dunia digital saat ini, mengingat semakin seringnya serangan *malware* yang dapat merusak sistem dan mencuri data penting. Dalam penelitian [5] yang sudah dilakukan menemukan salah satu jenis *malware* yang banyak ditemukan adalah Trojan, yang dapat menyusup ke sistem tanpa diketahui

dan memberikan akses tidak sah bagi pihak yang tidak bertanggung jawab. Mayoritas serangan siber di Indonesia berdasarkan penelitian [5] didominasi oleh aktivitas malware trojan. Trojan adalah jenis malware yang tampak seperti aplikasi atau program biasa dan berfungsi secara normal untuk menipu pengguna. Dalam penelitian [5] trojan ini bisa memberikan akses tanpa izin kepada pelaku ancaman untuk mencuri data dan merusak sistem pada perangkat yang terinfeksi. Penelitian oleh [6] menunjukkan bahwa perangkat *Android* bisa dieksploitasi dengan cara membuat pengguna menginstal APK trojan yang dibuat menggunakan alat Metasploit. Oleh karena itu, deteksi dan pencegahan terhadap *malware,* termasuk trojan, sangat penting untuk menjaga privasi dan keamanan data di semua perangkat. Dari penelitian [7] tujuan dari *malware* bisa bervariasi, mulai dari pencurian data sensitif, kerusakan sistem, hingga pemanfaatan sumber daya komputer untuk tujuan jahat, seperti penambangan *cryptocurrency* tanpa izin. *Malware* biasanya menyebar melalui email *phishing*, unduhan dari internet, atau celah keamanan dalam perangkat lunak.

Alat keamanan untuk *malware Android* yang mampu dengan cepat mengenali dan mengelompokkan berbagai jenis *malware* telah menjadi populer dalam beberapa tahun terakhir, karena bisa membantu menciptakan strategi respons yang lebih cepat terhadap serangan. Pada [8] deteksi *malware* bisa dilakukan dengan tiga cara, yaitu metode statis, dinamis, atau gabungan keduanya (*hybrid*). Penelitian [9], yang menggunakan metode statis dengan menganalisis langsung file APK menggunakan CNN, berhasil mendeteksi *malware* pada perangkat *mobile* dengan akurasi sebesar 84,9%. Menurut [2], deteksi pada lalu lintas jaringan lebih efektif untuk mendeteksi *malware* pada perangkat *mobile*, karena sebagian besar *malware mobile* akan berkomunikasi melalui jaringan. Penelitian yang dilakukan oleh [2] menggunakan metode statis dengan menganalisis lalu lintas jaringan yang dihasilkan oleh *malware* pada perangkat mobile menggunakan *deep learning*, dan berhasil mencapai akurasi sebesar 98,9%.

Sistem yang diusulkan [10] menggunakan *AdaBoost decision tree* dengan pengaturan terbaik, yang dilatih dan diuji pada dataset terbaru, berhasil mendeteksi *malware*. Dalam penilitian [10] *decision tree* bekerja dengan mengklasifikasikan data berdasarkan fitur-fitur spesifik dalam lalu lintas jaringan, seperti alamat IP,

protokol yang digunakan, waktu komunikasi, dan ukuran paket. Dengan menggunakan fitur-fitur ini, *decision tree* dapat membedakan antara aktivitas normal dan aktivitas yang mencurigakan. Hasil pengujian [10] menunjukkan bahwa sistem deteksi ini cepat dan akurat, dengan tingkat akurasi prediksi mencapai 98,84% dan waktu prediksi yang sangat singkat, yaitu 2,174 µSec. Pada penelitian [11] Akurasi terbesar yang dihasilkan oleh metode *Decision Tree* (DT) dalam penelitian ini [11] dengan menggunakan lima fitur saja, model *Decision Tree* (DT) berhasil mencapai nilai *recall* sebesar 97% untuk kelas NotPetya dan F1 *score* serta *recall* sebesar 99% untuk kelas *Miuref*. Selain itu, meskipun menggunakan lebih sedikit fitur, DT menunjukkan performa yang lebih baik dibandingkan dengan metode lain seperti k-NN dan SVM. Namun, penelitian-penelitian tersebut menggunakan dataset yang sudah lama atau tidak terlalu fokus pada lingkungan perangkat *mobile*. Oleh karena itu, penulis terinspirasi untuk melakukan penelitian tentang deteksi mobile *malware* trojan berdasarkan penelitian sebelumnya [12], [2], [10] ,dan [11] dengan menggunakan dataset terbaru. Maka, diharapkan dari percobaan ini akan tercipta model *Decision tree* dapat memiliki akurasi yang tinggi dalam mendeteksi aktivitas *malware* trojan pada lalu lintas jaringan dengan mengolah dataset terbaru yaitu data *Reverse TCP* dari perangkat *mobile* yang terinfeksi. Sehingga penulis memilih penelitian dengan judul "**Deteksi *Malware* Trojan Pada Lalu Lintas Jaringan *Reverse* TCP Dengan Algoritma *Decision Tree***". Dengan harapan metode *decision tree* ini bisa mendeteksi malware pada lalu lintas jaringan.

1.2 Rumusan Masalah

Dari penjelasan latar belakang diatas, dapat diuraikan rumusan masalahnya sebagai berikut :

1. Bagaimana proses ekstraksi data dari file PCAP ke dalam format CSV?
2. Bagaimana metode *Decision tree* untuk mendeteksi aktivitas *malware* pada data jaringan oleh perangkat *mobile* yang terinfeksi?
3. Bagaimana cara meningkatkan performa metode *decision tree* dalam mendeteksi *Malware* ?

1.3 Batasan Masalah

Adapun batasan masalah dari penelitian ini antara lain:

1. Ekstraksi data dari file PCAP hanya mencakup fitur-fitur yang relevan untuk analisis dan klasifikasi.
2. Algoritma yang digunakan untuk mendeteksi aktivitas *malware* adalah *decision tree* tanpa melakukan perbandingan dengan metode deteksi lainnya.
3. Sistem yang dikembangkan difokuskan untuk mendeteksi aktivitas *reverse TCP* dari *malware* dan tidak mencakup deteksi serangan lainnya atau malware tipe lain.

1.4 Tujuan penelitian

Dari rumusan masalah diatas berikut tujuan dari penelitian ini:

1. Mengidentifikasi prosedur untuk mengekstrak *data network traffic* dari file PCAP ke dalam format CSV.
2. Mengidentifikasi menggunakan metode *Decision Tree* untuk mendeteksi aktivitas *malware* yang terjadi pada perangkat mobile yang terinfeksi melalui analisis data jaringan.
3. Meningkatkan performa model *Decision Tree* dalam mendeteksi *malware* Trojan, guna mencapai tingkat deteksi yang lebih tinggi dan mengurangi tingkat kesalahan.

1.5 Manfaat penelitian

Manfaat yang dapat menjawab rumusan masalah sebagai berikut:

1. Dapat memberikan metode yang efektif untuk mengekstraksi data dari file PCAP ke dalam format CSV.
2. Dapat membantu dalam implementasi deteksi *malware* yang berbasis pada metode *Decision Tree.*
3. Dapat meningkatkan performa model *Decision Tree* dalam mendeteksi *malware* dan dapat memperkuat sistem deteksi

1.6 Metodologi Penelitian

Berikut tahapan tahapan dilakukan penelitian ini:

1. Tahap pertama adalah melakukan studi literatur.
   Di tahap ini, penulis mengumpulkan dan mempelajari berbagai penelitian yang relevan dengan topik yang sedang diteliti. Sumber-sumbernya bisa berupa jurnal, paper dari konferensi, dan dokumen ilmiah lainnya. Tujuan dari tahap ini adalah untuk memahami lebih dalam tentang topik tersebut dan menemukan informasi penting yang akan mendukung penelitian.
2. Tahap kedua adalah pengolahan data. Pada tahap ini, data dikumpulkan, kemudian diolah dan diekstrak untuk mendapatkan fitur-fitur penting yang nantinya akan digunakan dalam learning model. Data yang berbentuk file PCAP akan diubah ke dalam bentuk CSV menggunakan tools *CiFlow*. *CiFlow* akan memproses paket-paket dalam file tersebut dan mengekstrak informasi yang dibutuhkan. *CiFlow* secara otomatis akan mengidentifikasi dan mengekstrak berbagai fitur yang ada dalam traffic jaringan, seperti alamat IP, port, protokol, durasi koneksi, jumlah paket, dan lainnya.
3. Tahap ketiga adalah pelabelan, di mana data yang sudah memiliki format CSV akan diberi label pada kolom label untuk menentukan apakah data tersebut merupakan malware atau non-malware. Proses pelabelan ini sangat penting karena menentukan target yang akan diprediksi oleh model. Dengan pelabelan yang akurat, model dapat belajar dengan baik dan menghasilkan prediksi yang lebih tepat saat diterapkan pada data baru.
4. Tahap keempat adalah perancangan model deteksi. Pada tahap ini, penulis akan membangun model *Decision tree* (DT) yang akan dilatih menggunakan data *traffic malware trojan* dari Metasploit.
5. Tahap kelima adalah pengujian model deteksi. Di tahap ini, model yang sudah dibuat sebelumnya akan diuji menggunakan test dataset untuk mengevaluasi performa dan kemampuannya. Selain itu, dilakukan juga validasi untuk memastikan akurasi model.
6. Tahap keenam adalah hasil dan analisis. Di tahap ini, penulis mengevaluasi dan menganalisis hasil pengujian yang telah dilakukan untuk menilai

akurasi model, serta mengidentifikasi kekurangan dan kelebihannya yang dapat mempengaruhi kinerja model yang dibuat.

7. Tahap ketujuh adalah kesimpulan dan saran. Di tahap terakhir ini, penulis menarik kesimpulan berdasarkan perumusan masalah, studi literatur, hasil perancangan model, dan analisis dari pengujian model. Penulis juga akan memberikan saran yang bisa digunakan sebagai acuan untuk penelitian berikutnya.

1.7 Sistematika Penulisan

BAB I PENDAHULUAN

Bab pertama berisi penjelasan tentang latar belakang penelitian, perumusan masalah, tujuan dari penelitian ini, manfaat penelitian, metode penelitian serta susunan sistematika penulisan yang digunakan.

BAB II TINJAUAN PUSTAKA

Bab kedua akan membahas teori-teori dasar dan istilah-istilah penting yang menjadi dasar dalam penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ketiga akan memaparkan tahapan dan langkah-langkah yang dilakukan dalam penelitian ini. Mulai dari studi literatur dan tahap pengolahan dataset, pengumpulan dan pengolahan data, ekstraksi fitur, pembuatan model, hingga analisis dan pengujian model.

BAB IV HASIL DAN ANALISIS

Bab keempat akan menjelaskan hasil pengujian dan analisis terhadap hasil yang didapat, serta proses meningkatkan model.

BAB V KESIMPULAN

Bab kelima akan berisi kesimpulan dan saran dari penulis berdasarkan hasil dan analisis yang telah dilakukan dalam penelitian ini.

DAFTAR PUSTAKA

[1] A. Banik dan J. P. Singh, "Android Malware Detection by Correlated Real Permission Couples Using FP Growth Algorithm and Neural Networks," *IEEE Access*, vol. 11, no. November, hal. 124996–125010, 2023, doi: 10.1109/ACCESS.2023.3323845.

[2] M. Gohari, S. Hashemi, dan L. Abdi, "Android Malware Detection and Classification Based on Network Traffic Using Deep Learning," *2021 7th Int. Conf. Web Res. ICWR 2021*, hal. 71–77, 2021, doi: 10.1109/ICWR51868.2021.9443025.

[3] B. Urooj, M. A. Shah, C. Maple, M. K. Abbasi, dan S. Riasat, "Malware Detection: A Framework for Reverse Engineered Android Applications Through Machine Learning Algorithms," *IEEE Access*, vol. 10, no. February, hal. 89031–89050, 2022, doi: 10.1109/ACCESS.2022.3149053.

[4] D. O. Won, Y. N. Jang, dan S. W. Lee, "PlausMal-GAN: Plausible Malware Training Based on Generative Adversarial Networks for Analogous Zero-Day Malware Detection," *IEEE Trans. Emerg. Top. Comput.*, vol. 11, no. 1, hal. 82–94, 2023, doi: 10.1109/TETC.2022.3170544.

[5] I. Riadi, D. Aprilliansyah, dan S. Sunardi, "Mobile Device Security Evaluation using Reverse TCP Method," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, 2022, doi: 10.22219/kinetik.v7i3.1433.

[6] R. Satrio Hadikusuma, L. Lukas, dan E. M. Rizaludin, "Methods of Stealing Personal Data on Android using a Remote Administration Tool with Social Engineering Techniques," *Ultim. J. Tek. Inform.*, vol. 15, no. 1, hal. 44–49, 2023, doi: 10.31937/ti.v15i1.3122.

[7] F. B. Khan, M. H. Durad, A. Khan, F. A. Khan, M. Rizwan, dan A. Ali, "Design and Performance Analysis of an Anti-Malware System Based on Generative Adversarial Network Framework," *IEEE Access*, vol. 12, no.

February, hal. 27683–27708, 2024, doi: 10.1109/ACCESS.2024.3358454.

[8] F. A. Almarshad, M. Zakariah, G. A. Gashgari, E. A. Aldakheel, dan A. I. A. Alzahrani, "Detection of Android Malware Using Machine Learning and Siamese Shot Learning Technique for Security," *IEEE Access*, vol. 11, no. October, hal. 127697–127714, 2023, doi: 10.1109/ACCESS.2023.3331739.

[9] A. Mohanraj dan K. Sivasankari, "Android traffic malware analysis and detection using ensemble classifier," *Ain Shams Eng. J.*, vol. 15, no. 12, hal. 103134, 2024, doi: 10.1016/j.asej.2024.103134.

[10] Q. Abu Al-Haija, A. Odeh, dan H. Qattous, "PDF Malware Detection Based on Optimizable Decision Trees," *Electron.*, vol. 11, no. 19, hal. 1–18, 2022, doi: 10.3390/electronics11193142.

[11] J. Velasco-Mata, V. Gonzalez-Castro, E. F. Fernandez, dan E. Alegre, "Efficient Detection of Botnet Traffic by Features Selection and Decision Trees," *IEEE Access*, vol. 9, hal. 120567–120579, 2021, doi: 10.1109/ACCESS.2021.3108222.

[12] H. Bragança, V. Rocha, L. Barcellos, E. Souto, D. Kreutz, dan E. Feitosa, "Android malware detection with MH-100K: An innovative dataset for advanced research," *Data Br.*, vol. 51, 2023, doi: 10.1016/j.dib.2023.109750.

[13] R. Islam, M. I. Sayed, S. Saha, M. J. Hossain, dan M. A. Masud, "Android malware classification using optimum feature selection and ensemble machine learning," *Internet Things Cyber-Physical Syst.*, vol. 3, no. January, hal. 100–111, 2023, doi: 10.1016/j.iotcps.2023.03.001.

[14] J. Feng, L. Shen, Z. Chen, Y. Lei, dan H. Li, "HGDetector: A hybrid Android malware detection method using network traffic and Function call graph," *Alexandria Eng. J.*, vol. 114, no. July 2024, hal. 30–45, 2025, doi: 10.1016/j.aej.2024.11.068.

[15] L. Shen, M. Fang, dan J. Xu, "GHGDroid: Global heterogeneous graph-based android malware detection," *Comput. Secur.*, vol. 141, no. April, hal.

103846, 2024, doi: 10.1016/j.cose.2024.103846.

[16] S. Li, Z. Tang, H. Li, J. Zhang, H. Wang, dan J. Wang, “GMADV: An android malware variant generation and classification adversarial training framework,” *J. Inf. Secur. Appl.*, vol. 84, no. June, hal. 103800, 2024, doi: 10.1016/j.jisa.2024.103800.

[17] W. Guo *et al.*, “MalOSDF: An Opcode Slice-Based Malware Detection Framework Using Active and Ensemble Learning,” *Electron.*, vol. 13, no. 2, hal. 1–19, 2024, doi: 10.3390/electronics13020359.

[18] Y. D. Puji Rahayu dan Nanang Trianto, “Analisis Malware Menggunakan Metode Analisis Statis dan Dinamis untuk Pembuatan IOC Berdasarkan STIX Versi 2.1,” *Info Kripto*, vol. 15, no. 3, hal. 105–111, 2021, doi: 10.56706/ik.v15i3.30.

[19] S. Sharma, Prachi, R. Chhikara, dan K. Khanna, “A novel feature selection technique: Detection and classification of Android malware,” *Egypt. Informatics J.*, vol. 29, no. December 2024, hal. 100618, 2025, doi: 10.1016/j.eij.2025.100618.

[20] D. Suryono, “Analisis Keamanan Jaringan Hardware Trojan Pada IoT,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 4, hal. 3529–3537, 2022, doi: 10.35957/jatisi.v9i4.2845.

[21] C. Dong *et al.*, “A Cost-Driven Method for Deep-Learning-Based Hardware Trojan Detection,” *Sensors*, vol. 23, no. 12, hal. 1–29, 2023, doi: 10.3390/s23125503.

[22] V. Das, B. B. Nair, dan R. Thiruvengadathan, “A Novel Feature Encoding Scheme for Machine Learning Based Malware Detection Systems,” *IEEE Access*, vol. 12, no. May, hal. 91187–91216, 2024, doi: 10.1109/ACCESS.2024.3420080.

[23] P. Maniriho, A. N. Mahmood, dan M. J. M. Chowdhury, “MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations,” *Comput.*

Secur., vol. 142, no. December 2023, hal. 103864, 2024, doi: 10.1016/j.cose.2024.103864.

[24] P. Bhat, S. Behal, dan K. Dutta, "A system call-based android malware detection approach with homogeneous & heterogeneous ensemble machine learning," *Comput. Secur.*, vol. 130, hal. 103277, 2023, doi: 10.1016/j.cose.2023.103277.

[25] F. Wijayanto, H. D. Putranto, F. Toriq, dan D. S. Bhayangkara, "Analisis Static Malware Menggunakan Algoritma Random Forest Machine Learning," *J. Teknol. Inf.*, vol. 9, no. 2, hal. 172–176, 2023, doi: 10.52643/jti.v9i2.2850.

[26] H. S. Sharma dan K. J. Singh, "Intrusion detection system: a deep neural network-based concatenated approach," *J. Supercomput.*, vol. 80, no. 10, hal. 13918–13948, 2024, doi: 10.1007/s11227-024-05994-1.

[27] Z. Azam, M. M. Islam, dan M. N. Huda, "Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through Decision Tree," *IEEE Access*, vol. 11, no. July, hal. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[28] O. H. Abdulganiyu, T. Ait Tchakoucht, dan Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *Int. J. Inf. Secur.*, vol. 22, no. 5, hal. 1125–1162, 2023, doi: 10.1007/s10207-023-00682-2.

[29] T. Bajtoš, P. Sokol, dan F. Kurimský, "Processing of IDS alerts in multi-step attacks[Formula presented]," *Softw. Impacts*, vol. 19, no. January, hal. 100622, 2024, doi: 10.1016/j.simpa.2024.100622.

[30] A. Wahab *et al.*, "SEDERHANA MENGGUNAKAN CISCO PACKET TRACER," vol. 8, no. 6, hal. 12100–12107, 2024.

[31] S. S. Vladimirov, A. Vybornova, A. Muthanna, A. Koucheryavy, dan A. A. A. El-Latif, "Network Coding Datagram Protocol for TCP/IP Networks," *IEEE Access*, vol. 11, no. April, hal. 43485–43498, 2023, doi:

10.1109/ACCESS.2023.3266289.

[32] A. I. M. P. E-journal, “Vidhyayana - ISSN 2454-8596,” vol. 8, no. 7, hal. 317–332.

[33] S. Rani dan R. Nagpal, “Penetration Testing Using Metasploit Framework: An Ethical Approach,” *Int. Res. J. Eng. Technol.*, vol. 06, no. 08, hal. 538–542, 2019, [Daring]. Tersedia pada: https://www.academia.edu/40379823/IRJET-_PENETRATION_TESTING_USING_METASPLOIT_FRAMEWORK_AN_ETHICAL_APPROACH

[34] M. A. S. Sardar, H. Saha, M. N. Sultan, dan M. F. Rabbi, “Intrusion Detection in Electric Vehicles using Machine Learning with Model Explainability,” *J. Inf. Hiding Multimed. Signal Process.*, vol. 14, no. 3, hal. 81–89, 2023.

[35] M. M. Taye, “Understanding of Machine Learning with Deep Learning :,” *Comput. MDPI*, vol. 12, no. 91, hal. 1–26, 2023.

[36] G. Hasan, A. Masud, R. I. Shanto, I. Sakin, dan M. R. Kabir, “Jou rna lP,” *Array*, hal. 100375, 2025, doi: 10.1016/j.array.2025.100375.

[37] N. Al Sarah, F. Y. Rifat, M. S. Hossain, dan H. S. Narman, “An Efficient Android Malware Prediction Using Ensemble machine learning algorithms,” *Procedia Comput. Sci.*, vol. 191, no. 2019, hal. 184–191, 2021, doi: 10.1016/j.procs.2021.07.023.

[38] D. Loreti dan G. Visani, “Parallel approaches for a decision tree-based explainability algorithm,” *Futur. Gener. Comput. Syst.*, vol. 158, no. September 2023, hal. 308–322, 2024, doi: 10.1016/j.future.2024.04.044.

[39] M. Fadli dan R. A. Saputra, “Klasifikasi Dan Evaluasi Performa Model Random Forest Untuk Prediksi Stroke,” *JT J. Tek.*, vol. 12, no. 02, hal. 72–80, 2023, [Daring]. Tersedia pada: http://jurnal.umt.ac.id/index.php/jt/index

[40] N. Omer, A. H. Samak, A. I. Taloba, dan R. M. Abd El-Aziz, “A novel optimized probabilistic neural network approach for intrusion detection and

categorization," *Alexandria Eng. J.*, vol. 72, hal. 351–361, 2023, doi: 10.1016/j.aej.2023.03.093.

[41] H. Setiawan, "Visualisasi Serangan Trojan Metasploit pada Android dengan Metode K-Means," Skripsi S1, Program Studi Sistem Komputer, Universitas Sriwijaya, Palembang, 2024. [Online]. Tersedia: https://repository.unsri.ac.id/146546/1/RAMA_56201_09011182025004_0003047905_01_front_ref.pdf

[42] M. A. Ildiansyah, "Deteksi Serangan Trojan Metasploit pada Android dengan Metode Support Vector Machine (SVM)," Skripsi S1, Program Studi Sistem Komputer, Universitas Sriwijaya, Palembang, 2024. [Online]. Tersedia: https://repository.unsri.ac.id/155713/

[43] H. A. Yanti, H. Sukoco, dan S. N. Neyman, "Pemodelan Identifikasi Trafik Bittorrent Dengan Pendekatan Correlation Based Feature Selection (CFS) Menggunakan Algoritme Decision Tree (C4.5)," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 6, no. 1, hal. 1, 2021, doi: 10.24114/cess.v6i1.20855.