IMPLEMENTASI DAN EVALUASI KINERJA INTRUSION DETECTION SYSTEM MENGGUNAKAN ALGORITMA DECISION TREE PADA LINUX

Cesia Trisani Saragih Garingging[1], Sasmita Lumban Gaol[2], Maria Angelina Lubis[3], Lotar Mateus Sinaga[4]

[1,2,3,4]*Universitas Katolik Santo Thomas, Medan*
E-mail : [1]cesiasaragih@gmail.com, [2]sasmitalumbangaol468@gmail.com, [3]mariaangelinalubis@gmail.com, [4]lotarmateus88@gmail.com

ABSTRAK

Meningkatnya intensitas serangan siber pada jaringan komputer menuntut adanya sistem yang mampu mendeteksi aktivitas berbahaya secara cepat dan akurat. Salah satu solusi yang dapat diterapkan adalah Intrusion Detection System (IDS) berbasis Machine Learning. Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi kinerja Intrusion Detection System menggunakan algoritma Decision Tree pada lingkungan server Debian 12. Dataset yang digunakan adalah CICIDS2017 dengan dua skenario serangan, yaitu Distributed Denial of Service (DDoS) dan PortScan. Tahapan penelitian meliputi proses preprocessing data, pembagian data latih dan data uji, pelatihan model menggunakan algoritma Decision Tree, serta evaluasi performa berdasarkan nilai accuracy, precision, recall, F1-score, confusion matrix, waktu pelatihan, waktu prediksi, dan penggunaan sumber daya sistem. Hasil penelitian menunjukkan bahwa algoritma Decision Tree mampu memberikan tingkat akurasi sebesar 99,9889% pada skenario DDoS dan 99,9843% pada skenario PortScan. Waktu pelatihan yang diperoleh masing-masing sebesar 7,82 detik dan 5,66 detik, sedangkan waktu prediksi sebesar 0,081 detik dan 0,063 detik. Pengujian juga menunjukkan bahwa penggunaan CPU hanya sebesar 0,1% dengan penggunaan RAM sebesar 2072 MB pada skenario DDoS dan 2162,53 MB pada skenario PortScan. Berdasarkan hasil tersebut, algoritma Decision Tree terbukti efektif dan efisien untuk diimplementasikan sebagai Intrusion Detection System pada lingkungan server Debian 12.

Kata Kunci : *Intrusion Detection System1, Machine Learning2, Decision Tree3, CICIDS20174, Debian 125*

ABSTRACT

The increasing number of cyber attacks on computer networks requires a system capable of detecting malicious activities quickly and accurately. One of the solutions is a Machine Learning-based Intrusion Detection System (IDS). This study aims to implement and evaluate the performance of an Intrusion Detection System using the Decision Tree algorithm in a Debian 12 server environment. The dataset used is CICIDS2017 with two attack scenarios, namely Distributed Denial of Service (DDoS) and PortScan. The research stages include data preprocessing, training and testing data splitting, model training using the Decision Tree algorithm, and performance evaluation based on accuracy, precision, recall, F1-score, confusion matrix, training time, prediction time, and system resource usage. The results show that the Decision Tree algorithm achieved an accuracy of 99.9889% for the DDoS scenario and 99.9843% for the PortScan scenario. The training times were 7.82 seconds and 5.66 seconds, while the prediction times were 0.081 seconds and 0.063 seconds, respectively. The evaluation also indicates that CPU usage was only 0.1%, while RAM usage reached 2072 MB for DDoS and 2162.53 MB for PortScan. Based on these results, the Decision Tree algorithm is considered effective and efficient for implementation as an Intrusion Detection System in a Debian 12 server environment.

Keywords : *Intrusion Detection System, Machine Learning, Decision Tree, CICIDS2017, Debian 12*

PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah mendorong peningkatan pemanfaatan jaringan komputer pada berbagai bidang, seperti pendidikan, pemerintahan, bisnis, dan industri. Pemanfaatan teknologi jaringan memungkinkan proses pertukaran informasi berlangsung lebih cepat dan efisien. Namun, meningkatnya ketergantungan terhadap jaringan komputer juga diikuti oleh bertambahnya ancaman keamanan siber yang dapat memengaruhi kerahasiaan, integritas, dan ketersediaan data. Berbagai bentuk serangan, seperti Distributed Denial of Service (DDoS), Port Scanning, malware, dan aktivitas intrusi lainnya terus mengalami perkembangan baik dari sisi teknik maupun tingkat kompleksitasnya. Kondisi tersebut menuntut adanya mekanisme keamanan yang mampu mendeteksi aktivitas mencurigakan secara cepat dan akurat pada jaringan komputer (Zareen et al., 2022) (Shilpashree et al., 2019).

Salah satu teknologi yang banyak digunakan untuk mendukung keamanan jaringan adalah Intrusion Detection System (IDS). Sistem ini berfungsi untuk memantau lalu lintas jaringan serta mengidentifikasi aktivitas yang dianggap tidak normal atau berpotensi mengancam keamanan sistem. Pendekatan IDS tradisional umumnya menggunakan metode berbasis tanda tangan (*signature-based detection*), namun metode tersebut memiliki keterbatasan dalam mengenali pola serangan baru yang belum tersimpan dalam basis data aturan. Oleh karena itu, diperlukan pendekatan yang lebih adaptif agar sistem mampu mengenali berbagai pola serangan yang terus berkembang secara otomatis (Canadian Institute for Cybersecurity, n.d.) (Hidayat et al., 2023)

Perkembangan Machine Learning memberikan peluang yang besar dalam meningkatkan efektivitas sistem deteksi intrusi. Melalui kemampuan pembelajaran dari data historis, Machine Learning memungkinkan proses klasifikasi lalu lintas jaringan dilakukan secara otomatis. Berbagai algoritma telah diterapkan pada penelitian IDS, di antaranya Random Forest, Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes, dan Decision Tree. Di antara berbagai metode tersebut, Decision Tree menjadi salah satu algoritma yang banyak digunakan karena memiliki tingkat akurasi yang baik, proses pelatihan yang relatif cepat, serta hasil klasifikasi yang mudah diinterpretasikan (Jumaidi et al., 2025) (Firdaus & C, n.d.)

Berbagai penelitian terdahulu telah menunjukkan keberhasilan penerapan algoritma Decision Tree dalam sistem deteksi intrusi. Hasil penelitian menunjukkan bahwa algoritma Decision Tree dan Extra Trees mampu memberikan performa yang baik dalam mendeteksi aktivitas serangan jaringan pada lingkungan IDS (Prihantono et al., n.d.). Penelitian lain membuktikan bahwa algoritma Decision Tree C4.5 dapat digunakan untuk mendeteksi serangan Denial of Service (DoS) dengan tingkat akurasi yang tinggi (Pramana et al., n.d.). Selain itu, penerapan Machine Learning pada IDS juga telah berhasil meningkatkan kemampuan sistem dalam mengenali berbagai pola serangan jaringan (Hidayat et al., 2023). Implementasi algoritma Decision Tree pada proses deteksi intrusi menunjukkan kemampuan klasifikasi yang baik dengan kebutuhan komputasi yang relatif rendah (Sari Sitanggang & Pribadi, 2025). Penelitian yang memanfaatkan dataset CICIDS2017 juga menunjukkan bahwa pendekatan berbasis Machine Learning mampu menghasilkan performa deteksi yang tinggi terhadap berbagai jenis serangan jaringan (Devita & Larasati, 2025).

Meskipun berbagai penelitian terdahulu telah menghasilkan performa yang baik dalam penerapan Machine Learning pada IDS, masih terdapat beberapa aspek yang dapat dikembangkan lebih lanjut. Sebagian besar penelitian lebih berfokus pada evaluasi akurasi klasifikasi tanpa mempertimbangkan penggunaan sumber daya sistem, seperti CPU dan RAM. Selain itu, beberapa

penelitian hanya menguji satu jenis serangan sehingga belum memberikan gambaran mengenai konsistensi performa algoritma pada skenario serangan yang berbeda. Implementasi IDS berbasis Decision Tree pada lingkungan server Debian 12 juga masih belum banyak dibahas dalam penelitian sebelumnya (Prihantono et al., n.d.), (Devita & Larasati, 2025). Oleh karena itu, diperlukan penelitian yang tidak hanya mengevaluasi tingkat akurasi model, tetapi juga mengukur efisiensi penggunaan sumber daya sistem selama proses deteksi intrusi.

Berdasarkan permasalahan tersebut, penelitian ini mengimplementasikan Intrusion Detection System berbasis Machine Learning menggunakan algoritma Decision Tree pada lingkungan server Debian 12. Dataset yang digunakan adalah CICIDS2017 (Puji et al., n.d.) dengan dua skenario serangan, yaitu DDoS dan PortScan. Evaluasi dilakukan menggunakan beberapa parameter pengukuran, meliputi accuracy, precision, recall, F1-score, confusion matrix, training time, prediction time, serta penggunaan CPU dan RAM selama proses klasifikasi. Penggunaan dua jenis serangan bertujuan untuk mengevaluasi konsistensi performa algoritma dalam mendeteksi aktivitas intrusi yang berbeda.

Tujuan penelitian ini adalah mengimplementasikan algoritma Decision Tree sebagai Intrusion Detection System pada lingkungan server Debian 12 serta mengevaluasi kinerjanya menggunakan dataset CICIDS2017. Hasil penelitian diharapkan dapat memberikan kontribusi dalam pengembangan sistem keamanan jaringan berbasis Machine Learning yang memiliki tingkat akurasi tinggi dan efisien dalam penggunaan sumber daya sistem. Selain itu, penelitian ini diharapkan dapat menjadi referensi bagi pengembangan penelitian selanjutnya yang berfokus pada implementasi dan evaluasi IDS berbasis Machine Learning pada lingkungan server berbasis Linux.

METODE

Tahapan Penelitian

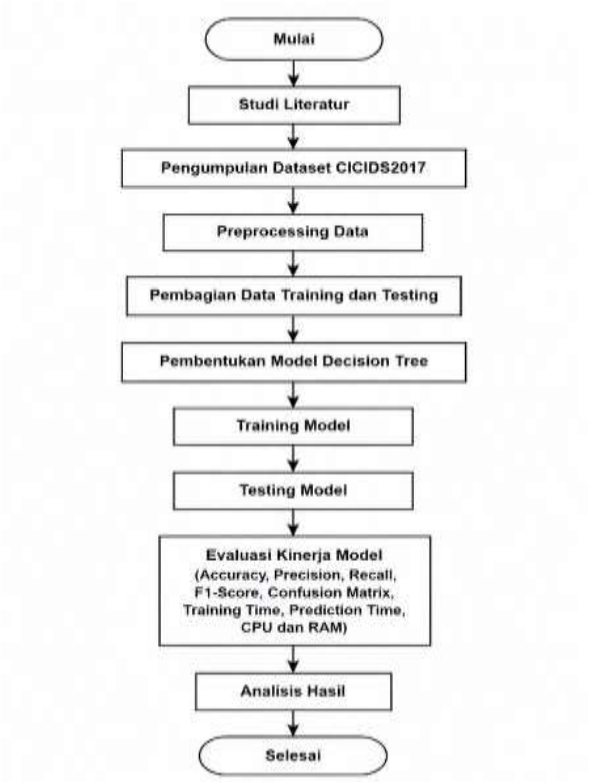
Penelitian ini menggunakan metode eksperimen untuk mengimplementasikan dan mengevaluasi kinerja Intrusion Detection System (IDS) berbasis Machine Learning menggunakan algoritma Decision Tree pada lingkungan server Debian 12. Tahapan penelitian dilakukan secara sistematis mulai dari studi literatur hingga analisis hasil pengujian.

Tahap pertama adalah studi literatur yang dilakukan dengan mengumpulkan referensi mengenai keamanan jaringan, Intrusion Detection System (IDS), Machine Learning, algoritma Decision Tree, serta penelitian terdahulu yang relevan. Tahap berikutnya adalah pengumpulan dataset CICIDS2017 yang digunakan sebagai data penelitian.

Setelah dataset diperoleh, dilakukan tahap preprocessing data yang meliputi pembersihan data, penghapusan nilai kosong (missing value), penghapusan nilai infinity, serta pemisahan fitur dan label. Data yang telah diproses kemudian dibagi menjadi data latih dan data uji dengan perbandingan 80% : 20%.

Selanjutnya dilakukan implementasi algoritma Decision Tree menggunakan bahasa pemrograman Python pada sistem operasi Debian 12. Model yang telah dibangun kemudian dilatih menggunakan data latih dan diuji menggunakan data uji untuk menghasilkan klasifikasi lalu lintas jaringan normal dan serangan.

Tahap terakhir adalah evaluasi kinerja model menggunakan parameter accuracy, precision, recall, F1-score, confusion matrix, training time, prediction time, penggunaan CPU, dan penggunaan RAM. Hasil pengujian kemudian dianalisis untuk mengetahui tingkat efektivitas algoritma Decision Tree dalam mendeteksi serangan jaringan.



Gambar 1. Alur Tahapan Penelitian

Berdasarkan Gambar 1, penelitian dilaksanakan melalui beberapa tahapan yang saling berhubungan. Proses penelitian diawali dengan kegiatan studi literatur untuk memperoleh pemahaman mengenai keamanan jaringan, Intrusion Detection System (IDS)[1], Machine Learning, serta algoritma Decision Tree (Shilpashree et al., 2019) yang digunakan dalam penelitian. Selain itu, kajian terhadap penelitian terdahulu juga dilakukan sebagai dasar dalam penyusunan metode penelitian.

Setelah tahap studi literatur selesai, dilakukan pengumpulan dataset CICIDS2017 (Canadian Institute for Cybersecurity, n.d.) yang digunakan sebagai sumber data utama. Dataset tersebut kemudian melalui proses preprocessing yang mencakup pembersihan data, penanganan nilai kosong (missing value), penghapusan nilai infinity, serta pemisahan atribut fitur dan label agar data siap digunakan pada tahap pemodelan.

Data yang telah diproses selanjutnya dibagi menjadi data latih dan data uji dengan perbandingan 80% : 20%. Tahap berikutnya adalah implementasi algoritma Decision Tree menggunakan bahasa pemrograman Python pada lingkungan sistem operasi Debian 12. Model yang dibangun dilatih menggunakan data latih, kemudian dilakukan pengujian menggunakan data uji untuk menghasilkan klasifikasi lalu lintas jaringan normal maupun aktivitas serangan.

Tahap akhir penelitian berupa evaluasi performa model dengan memanfaatkan beberapa parameter pengukuran, yaitu accuracy, precision, recall, F1-score, confusion matrix, training time, prediction time, serta penggunaan CPU dan RAM. Hasil evaluasi tersebut kemudian dianalisis untuk mengetahui tingkat efektivitas algoritma Decision Tree dalam mendeteksi serangan jaringan pada lingkungan server Debian 12.

Implementasi Metode Decision Tree

Metode yang digunakan dalam penelitian ini adalah algoritma Decision Tree. Algoritma ini dipilih karena memiliki kemampuan klasifikasi yang baik, mudah diinterpretasikan, dan memiliki waktu komputasi yang relatif cepat. Implementasi dilakukan menggunakan bahasa pemrograman Python dengan bantuan pustaka Scikit-Learn pada lingkungan server Debian 12.

Tabel 1. Spesifikasi Lingkungan Penelitian

Komponen	Spesifikasi
Sistem Operasi	Debian 12
Processor	Intel(R) Core(TM) i3-N305 1.80 GHz
RAM	8 GB
Storage	238 GB
Bahasa Pemrograman	Python 3
Library	Pandas, NumPy, Scikit-Learn, Matplotlib

Tabel 1 menunjukkan spesifikasi perangkat keras dan perangkat lunak yang digunakan dalam penelitian. Lingkungan penelitian tersebut digunakan untuk melakukan implementasi dan pengujian algoritma Decision Tree.

Tabel 2. Dataset Penelitian

No	Nama Dataset	Jenis Serangan	Jumlah Data	Keterangan
1	Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv	DDoS	225.745	Pengujian pertama
2	Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	PortScan	286.467	Pengujian kedua

Tabel 2 menyajikan dataset yang digunakan pada penelitian ini, yang seluruhnya berasal dari CICIDS2017. Dataset terdiri atas dua berkas, yaitu Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv dengan jumlah 225.745 data yang merepresentasikan skenario serangan Distributed Denial of Service (DDoS) serta Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv dengan jumlah 286.467 data yang merepresentasikan skenario serangan PortScan. Masing-masing dataset diproses dan diuji secara terpisah untuk mengevaluasi kemampuan algoritma Decision Tree dalam mengidentifikasi karakteristik dua jenis serangan jaringan yang berbeda.

Evaluasi performa model dilakukan menggunakan beberapa parameter pengukuran. Salah satu parameter utama adalah *accuracy* yang dihitung menggunakan Persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Persamaan (1) digunakan untuk menghitung tingkat akurasi model klasifikasi berdasarkan jumlah data yang berhasil diprediksi dengan benar terhadap keseluruhan data pengujian. Nilai TP (*True Positive*) menunjukkan jumlah data serangan yang berhasil terdeteksi dengan benar, TN (*True Negative*) menunjukkan jumlah data normal yang berhasil dikenali dengan benar, FP (*False Positive*) menunjukkan data normal yang salah diklasifikasikan sebagai serangan, sedangkan FN (*False Negative*) merupakan data serangan yang salah diklasifikasikan sebagai data normal.

Selain accuracy, evaluasi kinerja model juga dilakukan menggunakan confusion matrix, precision, recall, dan F1-score untuk mengukur kemampuan klasifikasi model. Selain itu, waktu pelatihan (*training time*), waktu prediksi (*prediction time*), penggunaan CPU, dan penggunaan RAM turut dianalisis untuk mengetahui efisiensi algoritma Decision Tree dari sisi penggunaan sumber daya sistem.

HASIL PENELITIAN DAN PEMBAHASAN

Bagian ini menjelaskan hasil implementasi dan evaluasi kinerja *Intrusion Detection System* (IDS) berbasis *Machine Learning* menggunakan algoritma *Decision Tree* pada lingkungan server Debian 12. Pengujian dilakukan menggunakan dataset CICIDS2017 dengan dua skenario serangan, yaitu *Distributed Denial of Service* (DDoS) dan *PortScan*. Evaluasi dilakukan menggunakan parameter *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, *training time*, *prediction time*, penggunaan CPU, dan penggunaan RAM.

Implementasi Sistem

Implementasi sistem dilakukan menggunakan sistem operasi Debian 12 yang berjalan pada lingkungan virtualisasi VirtualBox. Algoritma Decision Tree diimplementasikan menggunakan bahasa pemrograman Python dengan bantuan pustaka Pandas, NumPy, Scikit-Learn, Matplotlib, Seaborn, dan Psutil.

Implementasi pada Lingkungan Debian 12

Tahap implementasi dimulai dengan pembuatan program Python untuk membaca dataset CICIDS2017, melakukan *preprocessing* data, membangun model Decision Tree, melakukan proses pelatihan (*training*), serta melakukan proses pengujian (*testing*).



Gambar 1. Implementasi dan Pengujian Algoritma Decision Tree pada Debian 12

Gambar 1 menunjukkan Implementasi algoritma Decision Tree dilakukan menggunakan bahasa pemrograman Python pada lingkungan sistem operasi Debian 12. Proses implementasi meliputi pembacaan dataset CICIDS2017, preprocessing data, pembagian data menjadi data latih dan data uji, proses pelatihan model, serta pengujian model menggunakan algoritma Decision Tree. Selain itu, sistem juga melakukan pemantauan terhadap waktu pelatihan, waktu prediksi, penggunaan CPU, dan penggunaan RAM selama proses klasifikasi berlangsung.

Proses Preprocessing Data

Tahap preprocessing dilakukan untuk mempersiapkan dataset sebelum digunakan pada proses klasifikasi. Proses ini bertujuan untuk meningkatkan kualitas data sehingga model dapat melakukan pembelajaran secara lebih optimal. Tahapan yang dilakukan meliputi pembacaan dataset, penanganan nilai infinity, penghapusan data yang mengandung nilai kosong (missing value), serta pemisahan atribut fitur dan label yang digunakan pada proses klasifikasi.

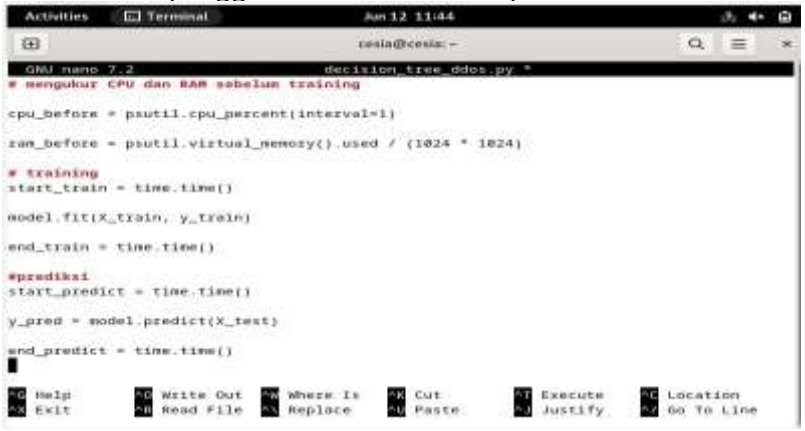


Gambar 2. Proses Pembagian Data dan Pembentukan Model Decision Tree

Gambar 2 menunjukkan proses preprocessing data dan pembentukan model Decision Tree. Tahapan yang dilakukan meliputi penggantian nilai infinity menjadi NaN, penghapusan data yang mengandung nilai kosong, pemisahan fitur dan label, pembagian dataset menjadi data latih sebesar 80% dan data uji sebesar 20%, serta inisialisasi model Decision Tree menggunakan parameter random_state=42 untuk menjaga konsistensi hasil pengujian.

Implementasi Pengukuran Kinerja Sistem

Selain melakukan klasifikasi terhadap lalu lintas jaringan, penelitian ini juga mengevaluasi penggunaan sumber daya sistem selama proses pelatihan dan pengujian model. Pengukuran dilakukan untuk mengetahui efisiensi algoritma Decision Tree dalam lingkungan server Debian 12. Parameter yang diamati meliputi waktu pelatihan (*training time*), waktu prediksi (*prediction time*), penggunaan CPU, serta penggunaan RAM selama proses klasifikasi berlangsung.



Gambar 3. Implementasi Pengukuran Kinerja Sistem

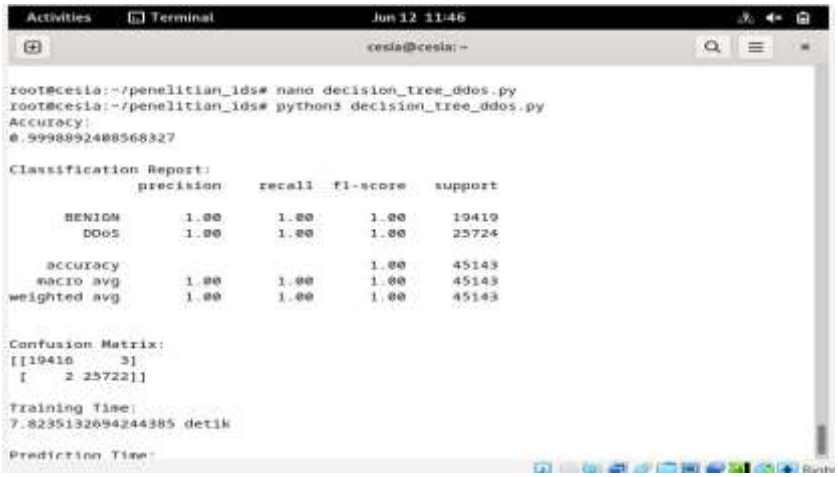
Berdasarkan Gambar 3 proses pengukuran kinerja sistem diawali dengan pengambilan nilai penggunaan CPU dan RAM sebelum proses pelatihan model menggunakan pustaka psutil. Selanjutnya dilakukan pencatatan waktu awal dan waktu akhir pelatihan untuk memperoleh nilai *training time*. Setelah model selesai dilatih, dilakukan proses prediksi terhadap data uji dengan mencatat waktu awal dan waktu akhir prediksi sehingga diperoleh nilai *prediction time*. Parameter-parameter tersebut digunakan untuk mengevaluasi efisiensi algoritma Decision Tree tidak hanya dari sisi akurasi klasifikasi, tetapi juga dari aspek penggunaan sumber daya sistem selama proses deteksi intrusi.

Hasil Pengujian Serangan DdoS

Proses Pelatihan dan Pengujian Model

Pengujian terhadap serangan DDoS dilakukan menggunakan dataset *Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv* yang berasal dari CICIDS2017. Dataset yang telah melalui tahap *preprocessing* kemudian dibagi menjadi data latih sebesar 80% dan data uji sebesar 20% menggunakan fungsi train_test_split() dengan nilai random_state=42. Selanjutnya model Decision Tree dilatih menggunakan data latih dan dilakukan pengujian terhadap data uji untuk menghasilkan klasifikasi lalu lintas normal (*BENIGN*) dan lalu lintas serangan DDoS.

Hasil Klasifikasi DdoS

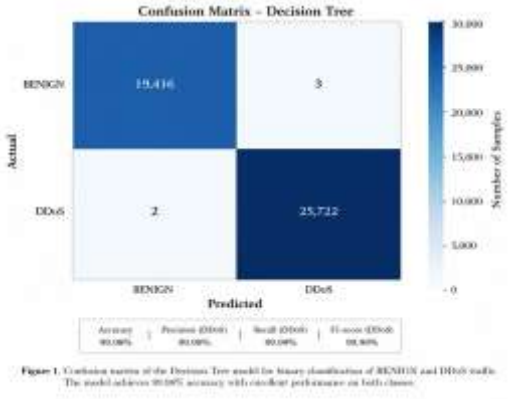


Gambar 4. Hasil Pengujian Decision Tree pada Dataset DdoS

Berdasarkan Gambar 4, model Decision Tree menghasilkan nilai accuracy sebesar 99,9889%. Nilai tersebut menunjukkan bahwa sebagian besar data berhasil diklasifikasikan dengan benar. Selain itu, nilai precision, recall, dan F1-score pada kedua kelas menunjukkan hasil yang sangat tinggi, yaitu mendekati 100%. Hal ini mengindikasikan bahwa model memiliki kemampuan yang sangat baik dalam membedakan lalu lintas normal (BENIGN) dan aktivitas serangan DDoS.

Proses pelatihan model memerlukan waktu sebesar 7,82 detik, sedangkan waktu prediksi yang dibutuhkan relatif singkat sehingga menunjukkan bahwa algoritma Decision Tree memiliki efisiensi komputasi yang baik dalam proses klasifikasi. Hasil tersebut membuktikan bahwa model mampu memberikan performa deteksi yang tinggi dengan waktu pemrosesan yang relatif cepat.

Confusion Matrix DDoS



Gambar 5. Confusion Matrix DdoS

Gambar 5 menunjukkan hasil *confusion matrix* yang diperoleh dari pengujian model Decision Tree terhadap dataset serangan DDoS. Berdasarkan hasil tersebut, sebanyak **19.416** data lalu lintas normal (*BENIGN*) berhasil diklasifikasikan dengan benar sebagai kelas BENIGN (*True Negative*), sedangkan **25.722** data serangan DDoS berhasil terdeteksi dengan benar sebagai kelas DDoS (*True Positive*).

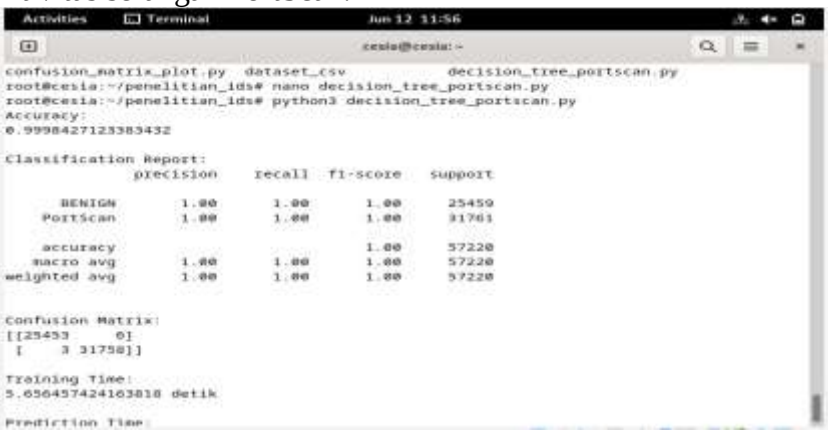
Kesalahan klasifikasi yang terjadi relatif sangat kecil, yaitu terdapat **3 data BENIGN** yang diprediksi sebagai DDoS (*False Positive*) dan **2 data DDoS** yang diprediksi sebagai BENIGN (*False Negative*). Jumlah kesalahan yang sangat sedikit tersebut menunjukkan bahwa model memiliki kemampuan deteksi yang sangat baik dengan tingkat akurasi yang tinggi.

Selain itu, nilai *precision*, *recall*, dan *F1-score* untuk kelas DDoS masing-masing mencapai **99,99%**, sehingga dapat disimpulkan bahwa algoritma Decision Tree mampu mengidentifikasi serangan DDoS secara efektif dan memiliki tingkat kesalahan klasifikasi yang sangat rendah.

Hasil Pengujian Serangan PortScan

Hasil Klasifikasi PortScan

Pengujian terhadap serangan PortScan dilakukan menggunakan dataset *Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv*. Model Decision Tree yang telah dibangun kemudian diuji menggunakan data uji untuk menghasilkan klasifikasi antara lalu lintas normal (*BENIGN*) dan aktivitas serangan PortScan.

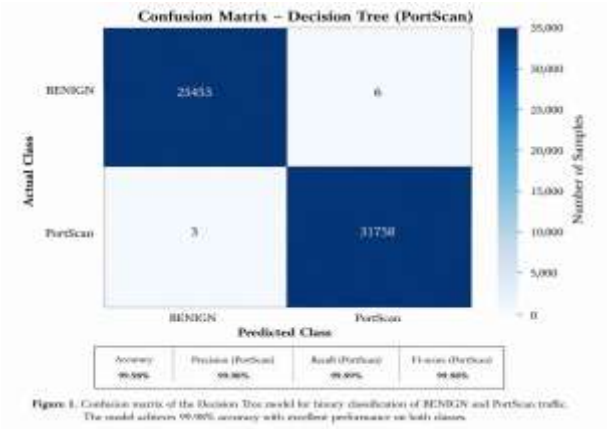


Gambar 6. Hasil Pengujian Decision Tree pada Dataset PortScan

Berdasarkan Gambar 3.6, model Decision Tree menghasilkan nilai **accuracy sebesar 99,9843%**. Nilai tersebut menunjukkan bahwa sebagian besar data dapat diklasifikasikan dengan benar. Selain itu, nilai **precision**, **recall**, dan **F1-score** pada kedua kelas menunjukkan performa yang sangat tinggi, yaitu mendekati 100%, sehingga model mampu membedakan lalu lintas normal dan aktivitas serangan PortScan dengan sangat baik.

Proses pelatihan model membutuhkan waktu sebesar **5,66 detik**, sedangkan waktu prediksi berlangsung relatif cepat. Hasil tersebut menunjukkan bahwa algoritma Decision Tree memiliki efisiensi komputasi yang baik dalam mendeteksi serangan PortScan.

Confusion Matrix PortScan



Gambar 7. Confusion Matrix PortScan

Gambar 7 menunjukkan hasil *confusion matrix* dari model Decision Tree pada pengujian serangan PortScan. Berdasarkan hasil tersebut, sebanyak **25.453** data lalu lintas normal (*BENIGN*) berhasil diklasifikasikan dengan benar sebagai kelas BENIGN (*True Negative*), sedangkan **31.758** data serangan PortScan berhasil terdeteksi dengan benar sebagai kelas PortScan (*True Positive*).

Kesalahan klasifikasi yang terjadi relatif sangat kecil, yaitu terdapat **6 data BENIGN** yang diprediksi sebagai PortScan (*False Positive*) dan **3 data PortScan** yang diprediksi sebagai BENIGN (*False Negative*). Jumlah kesalahan tersebut menunjukkan bahwa model memiliki kemampuan deteksi yang sangat baik dengan tingkat kesalahan yang sangat rendah.

Selain itu, nilai *precision*, *recall*, dan *F1-score* untuk kelas PortScan masing-masing mencapai sekitar **99,98%**, sehingga algoritma Decision Tree mampu mengidentifikasi aktivitas PortScan secara efektif dan menghasilkan performa klasifikasi yang sangat baik.

Perbandingan Hasil Pengujian DDoS dan PortScan

Tabel 3. Perbandingan Hasil Pengujian Decision Tree pada Serangan DDoS dan

Parameter	DDoS	PortScan
Dataset	Friday-WorkingHours-Afternoon-DDoS.pcap_ISCX.csv	Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv
Algoritma	Decision Tree	Decision Tree
Accuracy	**99,9889%**	**99,9843%**
Precision	100%	100%
Recall	100%	100%
F1-Score	100%	100%
Jumlah Data Uji	45.143	57.220

True Positive (TP)	25.722	31.758
True Negative (TN)	19.416	25.453
False Positive (FP)	3	6
False Negative (FN)	2	3
Training Time	7,82 detik	5,66 detik
Prediction Time	0,081 detik	0,063 detik
CPU Usage	0,1%	0,1%
RAM Usage	2072 MB	2162,53 MB

Analisis Hasil Pengujian

Berdasarkan Tabel 3.3, algoritma Decision Tree menunjukkan performa yang sangat baik pada kedua jenis serangan yang diuji. Pada skenario serangan DDoS, model menghasilkan nilai accuracy sebesar 99,9889%, sedangkan pada pengujian serangan PortScan diperoleh accuracy sebesar 99,9843%. Nilai precision, recall, dan F1-score pada kedua skenario pengujian mencapai 100%, yang menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik dalam membedakan lalu lintas normal dan lalu lintas serangan.

Jumlah kesalahan klasifikasi yang terjadi juga relatif sangat kecil. Pada pengujian DDoS terdapat 3 data false positive dan 2 data false negative, sedangkan pada pengujian PortScan terdapat 6 data false positive dan 3 data false negative. Hal ini menunjukkan bahwa algoritma Decision Tree mampu mendeteksi kedua jenis serangan dengan tingkat kesalahan yang sangat rendah.

Dari sisi efisiensi waktu, proses pelatihan model pada dataset PortScan memerlukan waktu lebih singkat, yaitu 5,66 detik, sedangkan pada dataset DDoS membutuhkan waktu 7,82 detik. Waktu prediksi pada kedua skenario juga relatif cepat dengan nilai kurang dari 0,1 detik sehingga memungkinkan sistem melakukan proses deteksi secara efisien.

Penggunaan sumber daya sistem juga tergolong rendah. Penggunaan CPU pada kedua pengujian hanya sebesar 0,1%, sedangkan penggunaan RAM berada pada kisaran 2 GB. Hasil tersebut menunjukkan bahwa implementasi algoritma Decision Tree pada lingkungan server Debian 12 dapat dijalankan dengan baik pada perangkat dengan spesifikasi menengah tanpa membutuhkan sumber daya komputasi yang tinggi.

Pembahasan

Hasil penelitian menunjukkan bahwa algoritma Decision Tree memiliki kemampuan yang sangat baik dalam mendeteksi serangan DDoS maupun PortScan menggunakan dataset CICIDS2017. Tingginya nilai accuracy, precision, recall, dan F1-score menunjukkan bahwa model mampu mengidentifikasi pola serangan secara efektif. Selain itu, waktu komputasi yang relatif singkat serta penggunaan CPU dan RAM yang rendah menunjukkan bahwa metode ini tidak hanya unggul dari sisi akurasi, tetapi juga efisien dari sisi penggunaan sumber daya sistem.

Dengan demikian, algoritma Decision Tree dapat dijadikan sebagai salah satu alternatif yang efektif dalam implementasi Intrusion Detection System berbasis Machine Learning pada lingkungan server Debian 12. Hasil penelitian ini juga menunjukkan bahwa metode tersebut memiliki kemampuan yang konsisten dalam mendeteksi dua jenis serangan yang berbeda, yaitu DDoS dan PortScan.

KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa algoritma Decision Tree berhasil diterapkan sebagai Intrusion Detection System (IDS) berbasis Machine Learning pada lingkungan server Debian 12. Penelitian ini memanfaatkan dataset CICIDS2017 dengan dua skenario serangan, yaitu Distributed Denial of Service (DDoS) dan PortScan, untuk mengevaluasi kemampuan model dalam membedakan lalu lintas normal dan aktivitas serangan pada jaringan komputer.

Hasil pengujian menunjukkan bahwa algoritma Decision Tree memiliki kemampuan klasifikasi yang sangat baik. Pada pengujian menggunakan dataset DDoS diperoleh nilai accuracy sebesar 99,9889%, sedangkan pada dataset PortScan diperoleh nilai accuracy sebesar 99,9843%. Selain itu, nilai precision, recall, dan F1-score pada kedua skenario pengujian berada pada kisaran 100%, yang menunjukkan bahwa model mampu mengidentifikasi pola serangan dengan tingkat ketepatan yang sangat tinggi. Hasil confusion matrix juga memperlihatkan bahwa jumlah kesalahan klasifikasi yang terjadi relatif sangat kecil sehingga performa deteksi yang dihasilkan dapat dikatakan sangat baik.

Dari aspek kinerja sistem, algoritma Decision Tree menunjukkan efisiensi komputasi yang baik. Waktu pelatihan maupun waktu prediksi yang dibutuhkan relatif singkat, sedangkan penggunaan CPU dan RAM selama proses klasifikasi berada pada tingkat yang rendah. Kondisi tersebut menunjukkan bahwa metode Decision Tree tidak hanya memberikan tingkat akurasi yang tinggi, tetapi juga mampu bekerja secara efisien tanpa memerlukan sumber daya komputasi yang besar, sehingga sesuai untuk diterapkan pada lingkungan server dengan spesifikasi menengah.

Secara keseluruhan, hasil penelitian ini menunjukkan bahwa algoritma Decision Tree dapat dijadikan sebagai salah satu alternatif yang efektif dalam membangun sistem deteksi intrusi berbasis Machine Learning pada lingkungan server Debian 12. Untuk pengembangan penelitian selanjutnya, pengujian dapat diperluas dengan melibatkan jenis serangan lain pada dataset CICIDS2017 serta membandingkan performa Decision Tree dengan algoritma Machine Learning lainnya, seperti Random Forest, Support Vector Machine (SVM), maupun K-Nearest Neighbor (KNN), sehingga diperoleh model deteksi intrusi yang memiliki tingkat akurasi dan efisiensi yang lebih optimal.

DAFTAR PUSTAKA

[1] Sufia Zareen, Kaosar Hossain, Mohd Abdullah Al Mamun, and Samia Hasan Suha, “Machine Learning-Based Intrusion Detection Systems (IDS) for real-time cyber threat monitoring,” World Journal of Advanced Research and Reviews, vol. 15, no. 2, pp. 863–872, Aug. 2022, doi: 10.30574/wjarr.2022.15.2.0706.

[2] S. Shilpashree, S. C. Lingareddy, N. G. Bhat, and G. Sunil Kumar, “Decision tree: A machine learning for intrusion detection,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, no. 6 Special Issue 4, pp. 1126–1130, Apr. 2019, doi: 10.35940/ijitee.F1234.0486S419.

[3] Canadian Institute for Cybersecurity, “CICIDS2017 Dataset,” University of New Brunswick. Accessed: Jun. 17, 2026. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[4] I. Hidayat, M. Z. Ali, and A. Arshad, "Machine Learning-Based Intrusion Detection System: An Experimental Comparison," Journal of Computational and Cognitive Engineering, vol. 2, no. 2, pp. 88–97, May 2023, doi: 10.47852/bonviewJCCE2202270.

[5] K. Jumaidi, A. Adhyaksa Waskita, and A. Musyafa, "Analisis Performa Sistem Deteksi Intrusi pada Jaringan IoT Menggunakan Ensemble Learning dan Decision Tree," Integrative Perspectives of Social and Science Journal, vol. 2, no. 2, p. 2025, [Online]. Available: https://www.yorku.ca/research/bccc/ucs-technical/cybersecurity-datasets-cds/

[6] M. Firdaus and T. A. C, "IMPLEMENTASI ALGORITMA DECISION TREE UNTUK KLASIFIKASI POLA SERANGAN PADA LOG FILE."

[7] Y. Prihantono et al., "ANALISA PERBANDINGAN ALGORITMA MACHINE LEARNING DECISION TREES DAN EXTRA TREES TERHADAP KINERJA INTRUSION DETECTION SYSTEM DALAM MEMPREDIKSI SERANGAN PADA JARINGAN".

[8] K. Rai, M. Syamala Devi Professor, and A. Guleria, "Decision Tree Based Algorithm for Intrusion Detection".

[9] M. C. Belavagi, J. Veer Singh, and G. Attigeri, "Comparative Analysis of Anomaly Based Intrusion Detection Techniques."

[10] M. Pramana, E. Setyati, and F. X. Ferdinandus, "Wahana : Tridarma Perguruan Tinggi Identifikasi Serangan Denial Of Service (Dos) Di Jaringan Dengan Algoritma Decision Tree C4.5", [Online]. Available: http://jurnal.unipasby.ac.id/index.php/whn

[11] Y. Yuliana, D. H. Supriyadi, M. R. Fahlevi, and M. R. Arisagas, "Analysis of NSL-KDD for the Implementation of Machine Learning in Network Intrusion Detection System," Journal of Informatics Information System Software Engineering and Applications (INISTA), vol. 6, no. 2, pp. 80–89, Feb. 2024, doi: 10.20895/inista.v6i2.1389.

[12] M. Politeknik, N. J. Devita, and A. Larasati, "PENERAPAN ALGORITMA RANDOM FOREST DAN K-NEAREST NEIGHBOR UNTUK DETEKSI INTRUSI PADA DATASET CICIDS2017," Jurnal Multidisiplin Ilmu Akademik, vol. 2, no. 6, pp. 632–640, 2025, doi: 10.61722/jmia.v2i6.7213.

[13] I. Puji, S. Pusat, T. Informasi, D. Komunikasi, E. Utami, and A. H. Muhammad, "Comparison of Anomaly Based and Signature Based Methods in Detection of Scanning Vulnerability."

[14] M. Fadhlurrohman, A. Muliawati, and B. Hananto, "Analisis Kinerja Intrusion Detection System pada Deteksi Anomali dengan Metode Decision Tree Terhadap Serangan Siber Analysis of Intrusion Detection System Performance on Anomaly Detection with Decision Tree Method Against Cyber Attacks", [Online]. Available: http://journal.ipb.ac.id/index.

[15] N. Ziems, G. Liu, J. Flanagan, and M. Jiang, "Explaining Tree Model Decisions in Natural Language for Network Intrusion Detection," Oct. 2023, [Online]. Available: http://arxiv.org/abs/2310.19658

[16] S. S. Tripathy and B. Behera, "PERFORMANCE EVALUATION OF MACHINE LEARNING ALGORITHMS FOR INTRUSION DETECTION SYSTEM," 2023, doi: 10.17605/OSF.IO/WX6CS.

[17] M. Albarka Umar, C. Zhanfang, and Y. Liu, "A Hybrid Intrusion Detection with Decision Tree for Feature Selection."

[18] N. M. Sari A Sitanggang and O. Pribadi, "Journal of Artificial Intelligence and Engineering Applications Implementation of the Decision Tree Method for Detecting Attacks in Networks," 2025. [Online]. Available: https://ioinformatic.org/

[19] E. Hariyanti, D. Pramana Hostiadi, Y. Priyo Atmojo, I. Made Darma Susila, and I. Tangkawarow, "Analisis Perbandingan Metode Seleksi Fitur pada Model Klasifikasi Decission Tree untuk Deteksi Serangan di Jaringan Komputer".

[20] T. Yuliswar, I. Elfitri, and O. W. Purbo, "OPTIMIZATION OF INTRUSION DETECTION SYSTEM WITH MACHINE LEARNING FOR DETECTING DISTRIBUTED ATTACKS ON SERVER OPTIMALISASI SISTEM DETEKSI INTRUSI MENGGUNAKAN MACHINE LEARNING UNTUK DETEKSI SERANGAN TERDISTRIBUSI PADA SERVER," vol. 10, no. 1, p. 2025.