Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik1148911
Vol. 4, No. 11, Agustus 2024, hlm. 867-876 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 36/E/KPT/2019 e-ISSN: 2528-6579

ANALISIS KINERJA *INTRUSION DETECTION SYSTEM* BERBASIS ALGORITMA *RANDOM FOREST* MENGGUNAKAN DATASET *UNBALANCED HONEYNET* BSSN

Kuni Inayah[*1], Kalamullah Ramli[2]

[1,2]Universitas Indonesia, Depok
Email: [1]kuni.inayah@ui.ac.id, [2]kalamullah.ramli@ui.ac.id
[*]Penulis Korespondensi

(Naskah masuk: 01 April 2024, diterima untuk diterbitkan: 12 Agustus 2024)

Abstrak

Teknologi dan sistem informasi yang semakin berkembang menjadikan ancaman siber juga semakin meningkat. Pada tahun 2023, Indonesia menduduki peringkat pertama sebagai negara dengan sumber serangan tertinggi. Untuk mengatasi permasalahan tersebut, *Intrusion Detection System* (IDS) dijadikan solusi di berbagai sistem pemerintahan, bekerja sama dengan *Honeynet* BSSN. Namun, IDS ini tidak bekerja maksimal untuk mendeteksi jenis serangan baru yang belum pernah terjadi sebelumnya (*zero-day*). Untuk meningkatkan performa IDS salah satunya dengan menggunakan *machine learning*. Pada penelitian ini, diusulkan desain IDS berbasis algoritma *random forest* menggunakan dataset CIC-ToN-IoT sebagai dataset *whitelist* dan dataset *Honeynet* BSSN sebagai dataset *blacklist*. Model mengklasifikasikan 10 (sepuluh) klasifikasi yaitu *Benign, Information Leak, Malware, Trojan Activity, Information Gathering, APT, Exploit, Web Application Attack, Denial of Service (DoS)*, dan jenis serangan lainnya (*other*). Hasil analisis menunjukkan bahwa pemodelan IDS *based on machine learning* memiliki rata-rata nilai akurasi lebih dari 90%, nilai presisi 91%, nilai *recall* 90%, dan F1-*score* 90%. Untuk kelas klasifikasi dengan jumlah data *support* besar memiliki nilai presisi yang jauh lebih baik dibandingkan kelas klasifikasi dengan jumlah data *support* lebih sedikit. Dengan demikian, pemodelan *machine learning* yang dibuat dapat secara efektif dalam menganalisis berbagai serangan yang terjadi pada sistem informasi di Lingkungan Pemerintah terutama pada klasifikasi data dengan jumlah yang besar.

Kata kunci: *Intrusion Detection System, Random Forest, Dataset CIC-ToN-IoT, Dataset Honeynet BSSN*

PERFORMANCE ANALYSIS OF INTRUSION DETECTION SYSTEM BASED ON RANDOM FOREST ALGORITHM USING UNBALANCED HONEYNET BSSN DATASET

Abstract

As technology and information systems continue to develop, cyber threats also increase. In 2023, Indonesia will be ranked first as the country with the highest source of attacks. To overcome this problem, the Intrusion Detection System (IDS) is used as a solution in various government systems, in collaboration with Honeynet BSSN. However, this IDS doesn't work optimally to detect new types of attacks that have never happened before (zero-day). One way to improve IDS performance is by using machine learning. In this research, we propose an IDS design based on a random forest algorithm with the CIC-ToN-IoT dataset as a whitelist dataset and the Honeynet BSSN dataset as a blacklist dataset. The model classifies 10 (ten) classifications, namely Benign, Information Leak, Malware, Trojan Activity, Information Gathering, APT, Exploit, Web Application Attack, Denial of Service (DoS), and other types of attacks. The analysis results show that IDS modeling based on machine learning has an average accuracy value of more than 90%, a precision value of 91%, a recall value of 90%, and an F1 score of 90%. For the classification of large amounts of data, the precision value is much better than for the classification of data with smaller amounts. Thus, the machine learning modeling created can effectively analyze various attacks that occur on information systems in the government environment, especially in the classification of large amounts of data.

Keywords: *Intrusion Detection System, Random Forest, CIC-ToN-IoT Dataset, Honeynet BSSN Dataset*

1. PENDAHULUAN

Ancaman terhadap sistem informasi dan teknologi dunia maya akan meningkat seiring dengan perkembangannya, bahkan kemajuannya jauh lebih cepat. Penelitian Houssain Kettani tentang "Ancaman Teratas terhadap Sistem Siber" mengungkapkan

bahwa antara tahun 2012 dan 2018, terdapat 16 jenis ancaman siber, diantaranya *ransomware*, *phishing*, *denial of service attack*, *spam*, *botnet*, *data breaches*, *insider threat*, *physical manipulation/ damage/ theft/ loss*, *identity theft*, *cryptojacking*, *web-based attacks*, *web application attacks*, *cyber espionage*, dan *exploit kits* (Kettani dan Wainwright, 2019). Indonesia menduduki peringkat pertama dari 10 (sepuluh) negara dengan sumber serangan terbanyak, menurut Laporan Tahunan Badan Siber dan Sandi Negara - Indonesia *Honeynet Project* (BSSN-IHP) 2023.

Mekanisme keamanan jaringan komputer perlu ditingkatkan mengingat lonjakan serangan siber saat ini untuk menangani serangan yang belum terjadi atau tidak diantisipasi sebelumnya. *Cybersecurity Ventures*, organisasi riset ekonomi siber terkemuka di dunia, memperkirakan kerugian akibat kejahatan siber akan meningkat dari USD 3 triliun pada tahun 2015 menjadi lebih dari USD 10,5 triliun pada tahun 2025 (Sharif dan Mohammed, 2022). Oleh karena itu, penggunaan sistem IDS dan *Intrusion Prevention System* (IPS) merupakan salah satu cara untuk meningkatkan kapasitas kendali deteksi untuk mengidentifikasi ancaman yang canggih dan tidak umum serta meningkatkan pertahanan dan keamanan dalam teknologi dan sistem yang digunakan. IDS mengacu pada perangkat keras dan/atau perangkat lunak yang dapat secara aktif atau pasif memantau dan mengelola jaringan host untuk mengidentifikasi potensi ancaman.

Pemantauan lalu lintas jaringan secara real-time, analisis, dan respons defensif seperti pemblokiran serangan independen atau peringatan administrator semuanya dapat dilakukan oleh IDS (Tidjon, Frappier, dan Mammar, 2019). IDS dapat digunakan sebagai garis pertahanan awal terhadap ancaman dan/atau serangan siber. IDS berbasis *signature*, mengidentifikasi serangan dengan cara membandingkan karakteristik serangan dengan *signature* yang disimpan sebelumnya. Sehingga, teknik ini tidak mampu mengidentifikasi jenis serangan baru atau serangan *zero-day* (Amoli et al, 2016). Serangan modern yang kompleks dan terus-menerus seperti *Cobalt Strikes* tidak dapat dideteksi hanya oleh *Indicators of compromise* (IOC) karena perubahan fitur harus dipantau (Eijk dan Schuijt, 2020). Penyerang mengganti vektor serangan jaringan yang diketahui dengan cepat dan terus-menerus memodifikasi teknik serangan mereka untuk menghindari langkah-langkah keamanan. Oleh karena itu, deteksi serangan dinamis tidak dapat dihindari, dan IDS harus cukup beradaptasi dengan teknik deteksi serangan baru.

Banyak sistem deteksi intrusi jaringan (IDS) berbasis anomali yang membandingkan pola dan perilaku serangan telah dikembangkan untuk memecahkan masalah ini (Teodoro et al, 2009). Dengan mencoba mengatasi kelemahan sistem deteksi intrusi berbasis *signature*, IDS berbasis anomali memungkinkan identifikasi pola dalam lalu lintas jaringan. Untuk meningkatkan akurasi dan *detection rate* (DR) terhadap serangan *zero-day* dan ancaman baru lainnya yang tidak teridentifikasi, IDS berbasis anomali mendeteksi ancaman dengan cara membandingkan pola dan perilaku serangan dengan kondisi normal (Amoli et al, 2016).

Ada beberapa metode untuk mendeteksi anomali, salah satunya dengan *machine learning* (Teodoro et al, 2009). Kinerja IDS dalam mengidentifikasi serangan dan meneliti pola serangan dapat ditingkatkan dengan pendekatan *machine learning* (Mishra et al, 2019). Algoritma klasifikasi seperti *logistic regression*, *support vector machine*, *random forest*, dan sebagainya diperlukan dalam implementasi *machine learning*. Masing-masing algoritma berbeda memiliki kelebihan dan kekurangannya masing-masing, terutama dalam hal mengkategorikan serangan. Algoritma tertentu bagus dalam mengidentifikasi jenis serangan tertentu tetapi tidak efektif dalam mendeteksi serangan lainnya (Alvaro, Gamez, dan Garcia, 2019). Dalam penelitian yang dilakukan (Irfan et al, 2023) pada "*Machine Learning Algorithms for Intrusion Detection Performance Evaluation and Comparative Analysis*" menunjukkan bahwa algoritma *random forest* berkinerja baik dalam aplikasi *machine learning* karena *recall* yang tinggi dan tingkat *false positive* yang rendah.

Selain algoritma klasifikasi, kualitas dataset merupakan faktor utama dalam meningkatkan kinerja IDS (Zhou dan Pezaros, 2019). *Honeypot* adalah sistem salinan yang meniru fungsionalitas sistem asli. Berbagai serangan yang masuk ke dalam *log Honeypots* dikumpulkan menjadi dataset yang dapat diperiksa dan digunakan sebagai alat yang berguna untuk meneliti berbagai jenis, tren, dan taktik serangan siber. *Honeynet* atau sistem manajemen informasi dini serangan siber dibangun oleh BSSN pada tahun 2018 (Peraturan Kepala BSSN Nomor 6 Tahun 2023). *Honeynet* terdiri dari beberapa *honeypot* yang terhubung satu sama lain untuk mengumpulkan data serangan yang komprehensif dan membentuk dataset *blacklist* yang menargetkan jaringan pemerintah.

Pada dataset yang dihasilkan *honeynet* BSSN, tidak menyimpan dataset *whitelist*, sehingga diperlukan dataset yang berasal dari *Network Intrusion Detection System* dan tersedia publik untuk digunakan sebagai dataset *whitelist*. Pada penelitian yang dilakukan (Sarhan, 2020), dataset CIC-ToN-IoT memiliki nilai akurasi sebesar 99,33% dan nilai *detection rate* sebesar 99,80%.

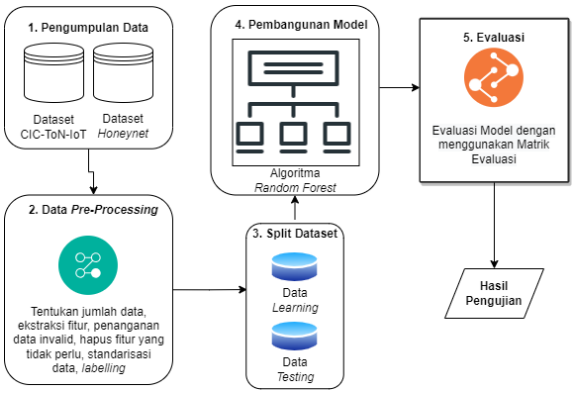
Berdasarkan informasi di atas, penelitian ini akan menggunakan dataset honeynet BSSN sebagai dataset *blacklist*, dataset CIC-ToN-IoT sebagai dataset *whitelist*, dan algoritma *random forest* sebagai algoritma klasifikasinya. Dengan kedua dataset dan algoritma klasifikasi tersebut diharapan pemodelan IDS *based on machine learning* mencapai kinerja yang baik dalam hal akurasi deteksi serangan.

Penulisan naskah pada penelitian ini disusun sebagai berikut: Bagian 1 menjelaskan Pendahuluan yang berisi latar belakang masalah dan tujuan penelitian. Bagian 2 menjelaskan Metode Penelitian yang berisi formulasi untuk menjawab permasalahan dan usulan metode penelitian yang dilakukan. Bagian 3 merupakan Pembangunan Model. Bagian 4 menjelaskan Hasil penelitian dan pembahasannya yang berisi hasil pengujian dan analisisnya, serta Bagian 5 memuat Kesimpulan.

2. METODE PENELITIAN

Metode penelitian yang diusulkan dalam penelitian ini diawali dengan tahap pengumpulan data, teknik *pre-processing*, *split* data, pemilihan algoritma klasifikasi dan pembangunan model, hingga melakukan evaluasi terhadap hasil pemodelan IDS *based on machine learning* yang telah dibuat dengan menggunakan matriks evaluasi. Langkah-langkah dalam metode penelitian ditunjukkan pada Gambar 1.

Bahasa pemrograman yang digunakan pada penelitian ini menggunakan *Python* dan dijalankan pada *Google Collabs* berbasis *cloud* dan dapat diintegrasikan dengan *Google Drive* sehingga dapat melakukan *training machine learning* dengan *source* yang besar tanpa memerlukan infrastruktur yang kompleks dan tidak memerlukan *space* memori komputer. *Platform* tersebut dimanfaatkan untuk pengumpulan data, *running* program, hingga visualisi data untuk selanjutnya dilakukan analisis (Ono et al, 2021). Sedangkan untuk *library* yang digunakan diantaranya *pandas*, *matplotlib*, *sklearn*, dan *numpy*.



Gambar 1. Metode Penelitian

2.1 Pengumpulan Data

Pada tahap pertama dilakukan pengumpulan data yaitu mengidentifikasi dan mengumpulkan informasi tentang dataset yang akan digunakan. Dataset yang digunakan pada penelitian ini yaitu dataset CIC-ToN-IoT sebagai dataset *whitelist* dan dataset *Honeynet* BSSN sebagai dataset *blacklist.*

Dataset CIC-ToN-IoT berasal dari hasil ekstraksi dari *file pcap* ToN-IoT menggunakan *CICFlowMeter-v4* yang berekstensi .csv (Sarnan, 2023). Dataset ini terdiri dari 5.351.760 data dengan data *benign* berjumlah 2.515.236 dan data serangan berjumlah 2.836.524 (Sarhan, Layeghy, dan Portmann, 2023). Dalam penelitian ini, dataset tersebut akan digunakan sebagai dataset *whitelist*, sehingga data yang yang diambil dari dataset tersebut hanya data yang berklasifikasi *benign.*

Sedangkan dataset *blacklist* berasal dari hasil *export* data pada kurun waktu 3 (tiga) bulan yaitu pada bulan Oktober s.d Desember 2023 dari sistem *honeynet* BSSN yang dikelola oleh Direktorat Operasi Keamanan Siber, Deputi II. Dataset *blacklist* berisi 2.579.053 data yang berekstensi .xlsx dan terdiri dari 9 (sembilan) kelas klasifikasi serangan yaitu *Information Leak* 1.037.520 data, *Malware* 270.570 data, *Trojan Activity* 131.157 data, *Information Gathering* 112.540 data, *APT* 25.978 data, *Exploit* 29.925 data, *Web Application Attack* 14.731 data, *Denial of Service (DoS)* 699 data, dan jenis serangan lainnya (*other*) 955.933 data (*Honeynet Project Team*, 2023).

2.2 Teknik *Pre-Processing*

Langkah pertama pada tahap ini yaitu menentukan jumlah data yang diperlukan untuk dilakukan pengujian. Pada dataset *blacklist* akan digunakan seluruh data yang berjumlah 2.579.053 data. Sedangkan. pada dataset *whitelist* akan diambil 1.000.000 data dari data *benign* pada dataset CIC-ToN-IoT yang berjumlah 2.515.236. Hal ini dikarenakan pada dataset *blacklist* jumlah data tertinggi berjumlah 1.037.520. Dengan demikian, jumlah data *benign* tidak terlampau tinggi dan mendekati seimbang jika dibandingkan data terbanyak pada dataset *blacklist*.

Langkah kedua pada tahap ini adalah mengekstrasi fitur-fitur dataset *whitelist* dan *blacklist.* Kemudian melakukan identifikasi fitur-fitur yang beririsan agar dataset *whitelist* dan *blacklist* dapat digabungkan menjadi dataset yang utuh. Dari kedua dataset tersebut didapatkan 6 (enam) fitur yang beririsan yaitu *Destination IP (Dst IP), Destination Port (Dst Port), Source IP (Src IP), Source Port (Src Port), Protocol*, dan Klasifikasi (*Attack*). Hasil penggabungan dataset disimpan ke dalam *file* datasets.csv.

Setelah penggabungan dataset selesai dilakukan, langkah selanjutnya melakukan identifikasi nilai-nilai pada datasets.csv, memastikan tidak ada nilai pada dataset yang hilang, semua data unik, dan tidak ada data yang kosong. Pengecekan tipe data juga diperlukan jika terdapat nilai dengan tipe data selain *number* harus diubah menjadi bentuk tipe data *number* agar memudahkan dalam proses pengujian.

Langkah terakhir pada tahap ini yaitu melakukan standarisasi data agar data memiliki skala yang sama, rentang nilai pada dataset seragam, tidak ada perbedaan nilai yang terlalu tinggi antara data minimal dan data maksimal yang dapat mempengaruhi proses *training* sehingga mungkin tidak sesuai dengan yang diharapkan (Ambarwati,

Adrian, dan Herdiyeni, 2020). Hal ini juga dilakukan agar proses pengujian berjalan lebih cepat dan tidak memerlukan komputasi yang besar. Proses standarisasi pada penelitian ini menggunakan metode *Standard Scaler* dengan menggunakan Persamaan (1) sebagai berikut.

$$f'_n = \frac{f_n - \mu}{\sigma} \quad (1)$$

Pada Persamaan (1), f'_n merupakan nilai yang telah dinormalisasi, f_n adalah nilai pada setiap fitur dataset ke n dengan n = 1, 2, …, k. Sedangkan μ yaitu *mean* dari nilai fitur pada dataset dan σ merupakan standard deviasi nilai fitur pada dataset.

2.3 *Split* Data

Dataset siap digunakan setelah data *pre-processing* selesai dilakukan. Tahap selanjutnya *split* data yaitu menentukan komposisi pembagian dataset yang akan digunakan sebagai data *learning* dan data *testing* (Bouckaert et al, 2016). Pada penelitian ini akan dilakukan *split* data dengan 3 (tiga) skenario pembagian dataset sebagai berikut:

(1) 75% data *learning* dan 25% data *testing* (75:25)
(2) 65% data *learning* dan 35% data *testing* (65:35)
(3) 50% data *learning* dan 50% data *testing* (50:50)

Data akan di *training* menggunakan data *learning*, kemudian untuk mengetahui kinerja model akan diuji dengan data *testing* (Zainudin, Shamsuddin, dan Hasan, 2019).

2.4 Pembangunan Model

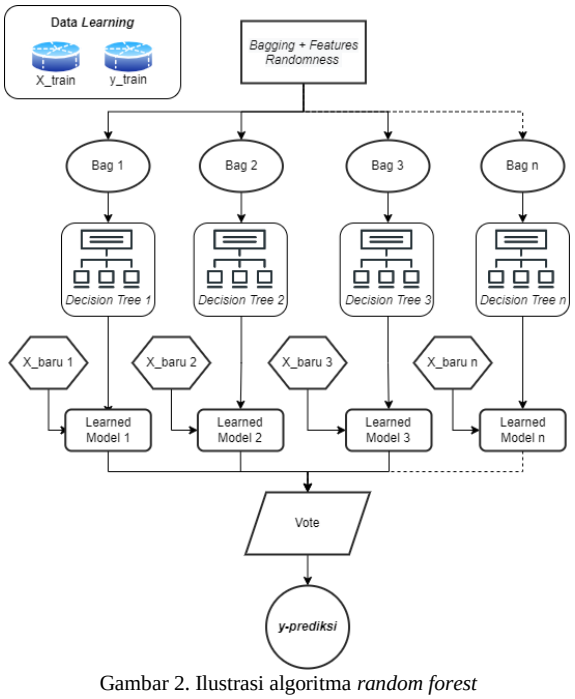
Dalam Pembangunan Model IDS *based on Machine Learning*, menentukan algoritma klasifikasi yang akan digunakan merupakan hal yang krusial.

Studi komparatif dan evaluasi kinerja dilakukan oleh (Irfan et al, 2023) terhadap dataset NSL-KDD dan KCup 1999 dengan menggunakan 5 (lima) algoritma klasifikasi yaitu *decision trees*, *random forest*, *SVM (support vector machines)*, *neural networks*, dan *models for deep learning*. Dari hasil studi diketahui bahwa penggunaan algoritma klasifikasi *deep learning models* dan *random forest* mencapai nilai akurasi yang lebih tinggi dari algoritma klasifikasi lainnya yaitu berturut-turut sebesar 90% dan 89%. Algoritma *deep learning models* memiliki nilai akurasi tertinggi, namun perlu untuk mempertimbangkan kebutuhan sumber daya menjadi lebih besar karena kompleksitas komputasi dan waktu pelatihan yang dibutuhkan yang lebih lama.

Penelitian sejenis juga dilakukan oleh (Rani dan Singh, 2023) yaitu studi komparatif dan evaluasi kinerja terhadap 2 (dua) dataset yaitu dataset UNSW-NB15 dan KCup 1999. Algoritma klasifikasi yang digunakan yaitu *decision trees*, *random forest*, *SVM (support vector machines)*, dan *logistic regression*. Hasil studi menunjukkan penggunaan algoritma klasifikasi *random forest* dan *SVM* memiliki nilai akurasi yang lebih tinggi dari algoritma klasifikasi lainnya yaitu berturut-turut sebesar 98,6% dan 99,8%. Meskipun algoritma *SVM* memiliki akurasi tertinggi, namun dalam hal *simplicity* dan proses *randomization*, algoritma klasifikasi *random forest* lebih unggul.

Penelitian terhadap dataset KDD 99, NSL-KDD, UNSW-NB15, dan CIC-IDS-2017 dilakukan oleh (Leon et al, 2022) dengan algoritma *supervised* yang digunakan yaitu *artificial neural neetwork*, *SVM*, *random forest*, *linear discriminant analysis*, dan *k-nearest neighbors*. Sedangkan *k-meean*, *mean shift*, dan DBSCAN merupakan algoritma *unsupervised* yang digunakan. Hasil penelitian menunjukkan bahwa algoritma *supervised random forest* memiliki kinerja terbaik dalam hal akurasi dan waktu yang dibutuhkan.

Penelitian lain dengan judul "*A comparative Study of Machine Learning Models for Malware Detection*" yang dilakukan (Mushtaq, Shahid, dan Zameer, 2022) melakukan evaluasi kinerja pada algoritma klasifikasi *AdaBoost*, *Gradient boosting*, *XGBoost*, *random forest*, *naive bayes* (NB), *support vector classifier* (SVC), *k-nearest neighbors* (k-NN), dan *decision tree*. Sedangkan, dataset yang digunakan berasal dari Kaggle yang terdiri dari 50.000 data *benign* dan 50.000 data *malware*. Pada penelitian ini algoritma klasifikasi *random forest* mencapai nilai akurasi tertinggi yaitu sebesar 99,96%.



Gambar 2. Ilustrasi algoritma *random forest*

Berdasarkan penelitian – penelitian tersebut, pada saat digunakan dalam berbagai model *machine learning*, algoritma *random forest* mencapai akurasi yang tinggi, cukup efisien dalam hal waktu pelatihan *machine learning* dan tidak memerlukan komputasi yang besar (Irfan et al, 2023). Sehingga, algoritma *random forest* akan digunakan dalam penelitian ini sebagai algoritma klasifikasi. Sedangkan untuk jumlah klasifikasi menggunakan *multiclass*

classification yang terdiri dari 10 (sepuluh) kelas klasifikasi yaitu *Benign*, *APT*, *Denial of Service*, *Exploit*, *Information Gathering*, *Information Leak*, *Malware*, *Trojan Activity*, *Web Application Attack*, dan *Other*. Ilustrasi algoritma *random forest* ditunjukkan pada Gambar 2.

2.5 Evaluasi

Langkah terakhir adalah mengevaluasi model yang dibuat untuk mengukur seberapa baik kinerjanya dalam melakukan klasifikasi/prediksi. Untuk menilai kinerja IDS *based on machine learning*, diperlukan matriks evaluasi karena matriks ini menggambarkan seberapa baik model yang dihasilkan beroperasi dan seberapa akurat model tersebut membuat prediksi. Perbandingan hasil klasifikasi pemodelan *machine learning* dengan hasil aktual yang diketahui disebut dengan *confusion matrix* (Gong, 2021). *Confusion Matrix* dapat dilihat pada Tabel 1 sebagai berikut.

Tabel 1. *Confusion Matrix*

		Predicted Label	
		Positve	*Negative*
True Label	*Positive*	*True Positive* (TP)	*False Negative* (FN)
	Negative	*False Positive* (FP)	*True Negative* (TN)

Berdasarkan Tabel 1, disebut *true positive* (TP) apabila pemodelan *machine learning* dapat mengklasifikasikan dengan benar (data *benign* diklasifikasikan sebagai data *benign*). Sebaliknya, apabila model *machine learning* salah mengklasifikasikan data *benign* diklasifikasikan sebagai data *malicious*, maka disebut dengan *false positif* (FP). Begitu juga dengan data *malicious*, disebut *true negative* (TN) apabila pemodelan *machine learning* dapat mengklasifikasikan data *malicious* dengan benar (data *malicious* diklasifikasikan sebagai data *malicious*). Sebaliknya, apabila pemodelan *machine learning* salah mengklasifikasikan data *malicious* diklasifikasikan sebagai data *benign*, maka disebut dengan *false negative* (FN).

Berikut ini merupakan matriks evaluasi yang digunakan pada penelitian ini:

(1) Akurasi

Salah satu metrik penilaian, yaitu akurasi yang ditentukan dengan menghitung frekuensi jumlah prediksi yang akurat oleh suatu model. Dalam hal klasifikasi, akurasi menunjukkan frekuensi model mengklasifikasikan kelompok positif dan negatif secara akurat (Bhattacharyya dan Khalita, 2014). Perhitungan nilai akurasi dapat menggunakan Persamaan (2).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

(2) Presisi

Presisi menggambarkan seberapa baik model mengklasifikasikan kelas positif (*true positive*) dengan benar dibandingkan dengan seluruh data pada kelas positif (*true positive* dan *false positive*) (Bhattacharyya dan Khalita, 2014). Untuk menghitung nilai presisi dapat menggunakan Persamaan (3).

$$Precission = \frac{TP}{TP+FP} \quad (3)$$

(3) *Recall*

Recall digunakan untuk mengevaluasi kinerja model dalam mengklasifikasikan kelas positif (*true positive*) dengan benar dibandingkan dengan jumlah data pada dataset yang diklasifikasikan dalam kelas positif (*true positive* dan *false negative*) (Bhattacharyya dan Khalita, 2014). Perhitungan *recall* menggunakan Persamaan (4) sebagai berikut.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

(4) F1-*score*

F1-score menunjukkan seberapa seimbang antara *Precision* dan *Recall* yang dihasilkan pemodelan *machine learning*. Nilai *F1-score* maksimum apabila nilai *precision* dan *recall* mencapai 100%. Semakin mendekati 100%, semakin baik pula kinerja pemodelan yang dibuat (Bhattacharyya dan Khalita, 2014). Untuk mengitung nilai F1-*score* dapat menggunakan Persamaan (5) berikut ini.

$$\text{F1} - score = 2\text{x}\frac{Precision * Recall}{Precision+Recall} \quad (5)$$

3. PEMBANGUNAN MODEL

3.1 Persiapan Dataset

Model yang dibangun merupakan Pemodelan IDS berbasis *machine learning* khususnya *supervised machine learning*, sehingga sangat bergantung pada dataset yang digunakan. Fitur-fitur dari dataset *whitelist* dan *blacklist* diekstraksi untuk mendapatkan fitur yang beririsan. Tabel 2 dan Tabel 3 menunjukkan kutipan fitur dari kedua dataset.

Tabel 2 Kutipan fitu-fitur dataset *whitelistt*

No	Fitur	Type Data
1	Attack	Object
2	Src IP	Object
3	Src Port	Int64
4	Dst IP	Object
5	Dst Port	Int64
6	Protocol	Int64
7	Flow Duration	Int64
8	Tot Fwd Pkts	Int64
9	Tot Bwd Pkts	Int64
10	TotLen Fwd Pkts	Float64
11	TotLen Bwd Pkts	Float64

Dataset yang digunakan dalam Pembangunan model diperoleh dari penggabungan data dari dataset *whitelist* dan *blacklist* yang memiliki persamaan fitur. Pada dataset *whitelist*, hanya menggunakan data berklasifikasi *benign*. Sedangkan pada dataset *blacklist*, seluruh klasifikasi data digunakan. Jumlah

data final dataset sebanyak 3.579.053 data dengan 1.000.000 dataset *whitelist* dan 2.579.053 dataset *blacklist*.

Tabel 3 Kutipan fitu-fitur dataset *blacklist*

No	Fitur	Type Data
1	Timestamp	Datetime64[ns]
2	Subdomain IOC	Object
3	Domain	Object
4	Threat Name	Object
5	Attack Results	Object
6	Destination IP	Object
7	Destination Port	Object
8	Source IP	Object
9	Source Port	Object
10	Klasifikasi	Object
11	Protocol	Object

3.2 Data *Pre-Processing*

Dataset baru harus dilakukan pembersihan agar seluruh data memiliki tipe data dan skala yang sama serta tidak ada data kosong atau *invalid*. Jika ada data *invalid* akan dihapus dan seluruh tipe data dilakukan pelabelan menjadi numerik atau *integer* menggunakan *labelencoder*. Sedangkan untuk standarisasi data menggunakan *StandardScaler*. Pengklasifikasian model menggunakan *multiclass classfification*. Hasil pelabelan klasifikasi data ditunjukkan pada Tabel 4.

Tabel 4. Label Klasifikasi Dataset

Label	Klasifikasi
0	*Benign*
1	*Information Leak*
2	*Malware*
3	*APT*
4	*Other*
5	*Web Application Attack*
6	*Exploit*
7	*Trojan Activity*
8	*Denial of Service*
9	*Information Gathering*

3.3 Pengujian dan Evaluasi

Pembersihan dataset menghasilkan dataset baru dengan fitur yang sama. Tabel 5 merupakan fitur-fitur pada dataset baru.

Tabel 5. Fitur Dataset Baru

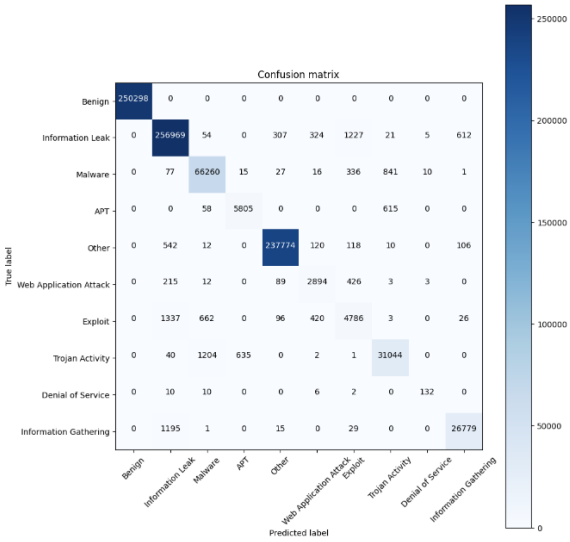
No	Fitur
1	*Source IP*
2	*Source Port*
3	*Destination IP*
4	*Destination Port*
5	*Protocol*
6	*Klasifikasi*

Setelah dataset dibersihkan, dataset dibagi menjadi data *learning* dan data *testing*. Pada penelitian ini, proporsi pembagian data *learning* dan data *testing* menggunakan proporsi 75:25, 65:35, dan 50:50. Proporsi ini digunakan untuk mengetahui pengaruh besar kecilnya data *learning* terhadap hasil klasifikasi model. Algoritma *random forest* diterapkan sebagai algoritma klasifikasi pada penelitian ini.

Pengujian dengan skenario 1 (75:25) menghasilkan nilai akurasi, presisi, *recall*, dan F1-*score* yang ditunjukkan pada Tabel 6. Sedangkan *confusion matrix* dapat dilihat pada Gambar 3.

Tabel 6. Hasil pengujian skenario 1 (75:25)

	presisi	*recall*	F1-*score*	*Support*
Benign	1,00	1,00	1,00	250.298
Information Leak	0,99	0,99	0,99	259.519
Malware	0,97	0,98	0,98	67.583
APT	0,90	0,90	0,90	6.478
Other	1,00	1,00	1,00	238.682
Web Application Attack	0,77	0,79	0,78	3.642
Exploit	0,69	0,65	0,67	7.330
Trojan Activity	0,95	0,94	0,95	32.926
Denial of Service	0,88	0,82	0,85	160
Information Gathering	0,97	0,96	0,96	28.019
akurasi			0,99	894.637
Macro avg	0,91	0,90	0,91	894.637



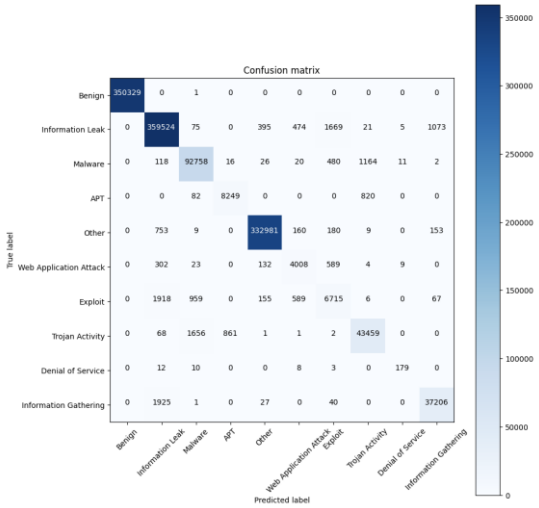
Gambar 3. *Confusion Matrix* Skenario 1 (75:25)

Untuk pengujian dengan skenario 2 (65:35) menghasilkan nilai akurasi, presisi, *recall*, dan F1-*score* yang disajikan pada Tabel 7 dan *confusion matrix* skenario 2 disajikan pada Gambar 4.

Tabel 7. Hasil pengujian skenario 2 (65:35)

	presisi	*recall*	F1-*score*	*Support*
Benign	1,00	1,00	1,00	350.330
Information Leak	0,99	0,99	0,99	363.236
Malware	0,97	0,98	0,98	94.595
APT	0,90	0,90	0,90	9.151
Other	1,00	1,00	1,00	334.245
Web Application Attack	0,76	0,79	0,78	5.067
Exploit	0,69	0,65	0,67	10.409
Trojan Activity	0,96	0,94	0,95	46.048
Denial of Service	0,88	0,84	0,86	212

	presisi	*recall*	F1-*score*	*Support*
Information Gathering	0,97	0,95	0,96	39.199
akurasi			0,99	1.252.492
Macro avg	0,91	0,90	0,91	1.252.492



Gambar 4. *Confusion Matrix* Skenario 2 (65:35)

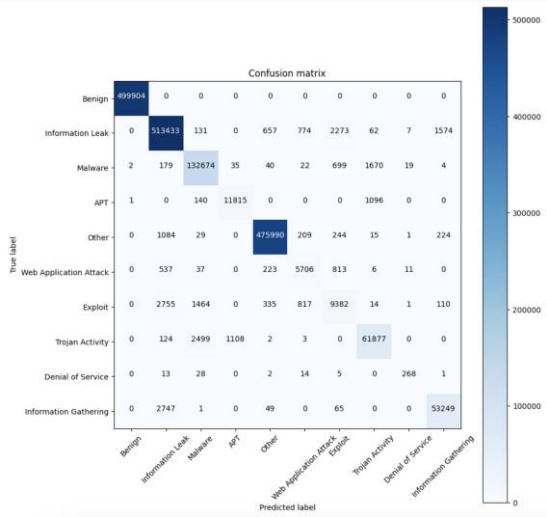
Tabel 8. Hasil pengujian skenario 3 (50:50)

	presisi	*recall*	F1-*score*	*Support*
Benign	1,00	1,00	1,00	499.904
Information Leak	0,99	0,99	0,99	518.911
Malware	0,97	0,98	0,97	135.344
APT	0,91	0,91	0,91	13.052
Other	1,00	1,00	1,00	477.796
Web Application Attack	0,76	0,78	0,77	7.333
Exploit	0,70	0,63	0,66	14.878
Trojan Activity	0,96	0,94	0,95	65.613
Denial of Service	0,87	0,81	0,84	331
Information Gathering	0,97	0,95	0,96	56.111
akurasi			0,99	1.789.273
Macro avg	0,91	0,90	0,90	1.789.273

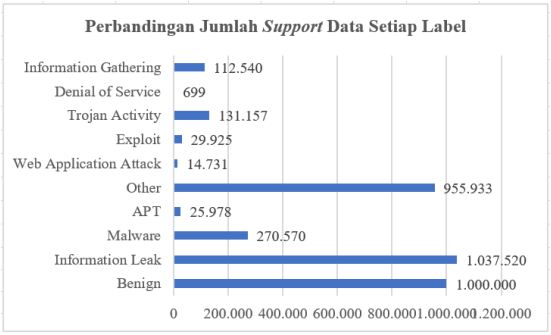
Sedangkan, pengujian dengan skenario 3 (50:50) menghasilkan nilai akurasi, presisi, *recall*, dan F1-*score* yang disajikan pada Tabel 8 dan *confusion matrix* skenario 3 disajikan pada Gambar 5.

4. HASIL DAN PEMBAHASAN

Klasifikasi *multiclass* diterapkan pada pengujian model *based on machine learning* dengan data *unbalanced*. Tiga skenario pembagian dataset yang digunakan untuk pengujian ini yaitu 75:25, 65:35, dan 50:50. Gambar 6 menunjukkan perbandingan jumlah data *support* untuk setiap kelas atau label, dan Tabel 9 memberikan ringkasan hasil pengujian.



Gambar 5. *Confusion Matrix* Skenario 3 (50:50)



Gambar 6. Perbandingan jumlah *support* data setiap kelas

Dari Tabel 9 di atas ditunjukkan bahwa hasil pengujian sangat dipengaruhi oleh jumlah data yang digunakan. Kelas dengan lebih banyak data *support* memiliki nilai akurasi, presisi, *recall*, dan F1-*score* yang lebih tinggi dibandingkan kelas dengan data *support* yang jauh lebih sedikit. Misalnya, kelas "*Information Leak*", yang secara keseluruhan memiliki 1.037.520 data, menghasilkan presisi, *recall*, dan F1-*score* sekitar 99% pada seluruh skenario. Sebaliknya, kelas "*Exploit*", yang memiliki 29.925 data secara keseluruhan, memiliki nilai presisi, *recall*, dan F1-*score* sekitar 60%. Namun, kelas dengan data *support* yang jauh lebih sedikit (*Denial of Service* dengan 699 data) memiliki nilai presisi, *recall*, dan F1-*score* yang cukup tinggi yaitu berkisar 80%. Hal ini dikarenakan data *learning* yang tersedia sangat terbatas, sehingga model tidak dapat mengklasifikasikan dengan baik.

Dengan memeriksa nilai akurasi pada Tabel 9, dapat dilihat bahwa setiap skenario pada semua kelas menghasilkan tingkat akurasi yang berkisar 90% hingga 100%. Namun, dengan data *unbalancd* terdapat variasi jumlah data *support* yang besar sehingga nilai akurasi tidak mencerminkan nilai akurasi sebenarnya secara akurat. Sehingga, nilai akurasi tidak dapat dijadikan tolak ukur untuk mengevaluasi seberapa baik model yang dibangun.

Tabel 9. Ringkasan hasil pengujian

Klasifikasi (Support)	Skenario	Akurasi	Presisi	*Recall*	F1-*Score*	Prosentase Klasifikasi Benar
Benign (1.000.000)	75:25	100%	100%	100%	100%	100%
	65:35	99,99%	100%	100%	100%	99,99%
	50:50	99,99%	100%	100%	100%	100%
Information Leak (1.037.520)	75:25	99,33%	99%	99%	99%	99,02%
	65:35	99,30%	99%	99%	99%	99,09%
	50:50	99,28%	99%	99%	99%	98,94%
Malware (270.570)	75:25	99,63%	97%	98%	98%	98,04%
	65:35	99,45%	97%	98%	98%	98,06%
	50:50	99,61%	97%	98%	97%	98,03%
APT (25.978)	75:25	99,83%	90%	90%	90%	89,61%
	65:35	99,86%	90%	90%	90%	90,14%
	50:50	99,87%	91%	91%	91%	90,52%
Other (955.933)	75:25	99,84%	100%	100%	100%	99,62%
	65:35	99,76%	100%	100%	100%	99,62%
	50:50	99,83%	100%	100%	100%	99,62%
Web Application Attack (14.731)	75:25	99,82%	79%	79%	78%	79,46%
	65:35	99,81%	76%	79%	78%	79,10%
	50:50	99,81%	76%	78%	77%	77,81%
Exploit (29.925)	75:25	99,48%	69%	65%	67%	65,29%
	65:35	99,47%	69%	65%	67%	64,51%
	50:50	99,46%	70%	63%	66%	63,06%
Trojan Activity (131.157)	75:25	99,62%	95%	94%	95%	94,28%
	65:35	99,63%	96%	94%	95%	94,38%
	50:50	99,63%	96%	94%	95%	94,31%
Denial of Service (699)	75:25	99,99%	88%	82%	85%	82,5%
	65:35	99,99%	88%	84%	86%	84,43%
	50:50	99,99%	87%	81%	84%	80,97%
Information Gathering (112.540)	75:25	99,78%	97%	96%	96%	95,57%
	65:35	99,74%	97%	95%	96%	94,94%
	50:50	99,73%	97%	95%	96%	94,90%

Pada penelitian ini mengklasifikasikan ke dalam 10 (sepuluh) klasifikasi, dengan 9 (sembilan) kelas dari klasifikasi tersebut adalah "*malicious*" dan 1 (satu) klasifikasi "*benign*". Oleh karena itu, titik berat penelitian ini meminimalkan *false positive* yaitu berupaya untuk mencegah "salah mengklasifikasikan" kelas "*malicious*" sebagai kelas "*benign*". Sehingga, untuk mengetahui seberapa baik model dalam melakukan klasifikasi, matriks evaluasi yang harus diperhatikan adalah nilai presisi. Presisi membandingkan frekuensi model memprediksi dengan tepat kelas positif (*true positive*) dengan jumlah total kelas positif (*true positive* dan *false positive*). Nilai presisi yang semakin besar menunjukkan jumlah *false positive* yang dihasilkan semakin kecil.

Tabel 9 juga menunjukkan bahwa nilai presisi dipengaruhi oleh data *unbalanced*. Berbeda dengan kelas yang memiliki data *support* yang jauh lebih sedikit, yang memberikan nilai presisi lebih kecil, kelas dengan jumlah data *support* yang signifikan besar menghasilkan nilai presisi yang besar. Dengan total data *support* sebanyak 270.570, kelas "*Malware*" memiliki nilai presisi sebesar 97%, sedangkan kelas "*Web Application Attack*" memiliki total data *support* sebanyak 14.731 menghasilkan nilai presisi sebesar 76% s.d 79%. Namun, kelas dengan data *support* terlampau sedikit juga dapat menghasilkan presisi yang cukup besar sebesar 87%-88%. Data yang sangat sedikit mengakibatkan model tidak dapat melakukan *learning* dengan baik, sehingga berpeluang model salah mengklasifikasikan. Berdasarkan komposisi pembagian dataset, dari ketiga skenario dapat disimpulkan bahwa nilai presisi akhir tidak dipengaruhi secara signifikan oleh komposisi tersebut. Nilai presisi yang hampir sama dihasilkan oleh setiap kelas di tiga skenario pembagian dataset.

5. KESIMPULAN ATURAN LAIN

Pada penelitian ini telah berhasil dibangun model IDS *based on machine learning* dengan data *unbalanced.* Kinerja model sangat dipengaruhi oleh besarnya data *support.* Kelas dengan data *support* yang besar menghasilkan nilai presisi, *recall*, dan F1-*score* yang lebih besar dibandingkan dengan kelas dengan jumlah data *support* yang kecil. Nilai akurasi tidak dapat dijadikan acuan untuk menilai seberapa baik model karena data *unbalanced* sehingga nilai yang dihasilkan tidak merepresentasikan nilai yang sebenarnya. Pada penelitian ini, memfokuskan meminimalisir nilai *false positive*, sehingga nilai presisi sangat penting untuk menilai kinerja model. Nilai presisi yang semakin besar menunjukkan kinerja yang semakin baik karena hal ini menunjukkan nilai *false positive* yang dihasilkan semakin kecil sesuai dengan yang diharapkan.

Untuk kinerja model yang lebih baik, gunakan jumlah data *support* yang besar dan *balanced* pada setiap kelas klasifikasi. Selain itu, pemilihan dataset dengan irisan fitur yang lebih banyak dapat diimplementasikan untuk menghasilkan pemodelan IDS *based on machine learning* dengan kinerja yang lebih baik.

6. UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Hibah Badan Penelitian dan Pengembangan Sumber Daya Manusia (Balitbang SDM), Kementerian Komunikasi dan Informatika Republik Indonesia.

DAFTAR PUSTAKA

ALVARO, E., GAMEZ, M., dan GARCIA, N., 2019. Ensemble Classification Methods with Applications in R. Vol. 7. John Wiley.

AMBARWATI, A., ADRIAN, Q. J., dan HERDIYENI Y., 2020. Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman. Jurnal RESTI, 4(1) p. 117-122.

AMOLI, P. V., HAMALAINEN, T., DAVID, G., ZOLOTUKHIN, M., dan MIRZAMOHAMMAD, M., 2016. Unsupervised network intrusion detection systems for zero-day fast-spreading attacks and botnets. JDCTA (International Journal of Digital Content Technology and its Applications,10 (2), p. 1–13.

BHATTACHARYYA, D.K. dan KHALITA, J. K., 2014. Network Anomaly Detection: A Machine Learning Perspective. Florida: CRC Press.

BOUCKAERT, R.R., FRANK, E., HALL, M., KIRKBY, R., REUTEMANN, P., SEEWALD, A., dan SCUSE, D., 2016. WEKA Manual Version 3-8-1. New Zealand: University of Waikato.

EIJK, V.V.D., SCHUIJT, C., 2020. Detecting cobalt strike beacons in netflow data.

IRFAN, B. M., POORNIMA, V., KUMAR, S.M., ASWAL, U.S., KRISHNAMOORTHY, N., dan MARANAN, R., 2023. Machine Learning Algorithms for Intrusion Detection Performance Evaluation and Comparative Analysis. 4th International Conference on Smart Electronics and Communication (ICOSEC). India: Trichy.

KETTANI, H. dan WAINWRIGHT, P., 2019. On the Top Threats to Cyber Systems. IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, pp. 175-179.

BSSN (Badan Siber dan Sandi Negara), 2023. Laporan Tahunan Honeynet Project BSSN Tahun 2023. Jakarta: Badan Siber dan Sandi Negara.

LEON, M., MARKOVIC, T. L, dan PUNNEKKAT, S., 2022. Comparative Evaluation of Machine Learning Algorithms for Network Intrusion Detection and Attack Classification. International Joint Conference on Neural Networks (IJCNN). Italia: Padua.

MISHRA, P., VARADHARAJAN, V., TUPAKULA, U., dan PILLI, E. S., 2019. A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection. IEEE Communication Surveys and Tutorials, 21(1).

MUSHTAQ, E., SHAHID, F., dan ZAMEER, A., 2022. A comparative study of machine learning models for malware detection. 19th International Bhurban Conference on Applied Sciences and Technology (IBCAST). Pakistan: Islamabad.

GONG, M. 2021. A novel performance measure for machine learning classification. International Journal of Managing Information Technology (IJMIT), 13. DOI:10.5121/ijmit.2021.13101.

ONO, J.P., FREIRE, J. AND SILVA, C.T., 2021. Interactive Data Visualization in Jupyter Notebooks. Computing in Science & Engineering 23(2), p.99-106. https://doi.org/10.1109/MCSE.2021.3052619.

Peraturan Kepala Badan Siber Dan Sandi Negara Nomor 6 Tahun 2023 tentang Penyelenggaraan Layanan Honeynet Badan Siber dan Sandi Negara. Jakarta: Badan Siber dan Sandi Negara.

RANI, M.J. dan SINGH, D., 2023. Machine Learning Algorithm for Intrusion Detection: Performance Evaluation and Comparative Analysis. 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). Nepal: Kirtipur.

SARHAN, M., 2020. Netflow datasets. Available at: http://staff.itee.uq.edu.au/marius/NIDS_datasets/.

SARHAN, M., LAYEGHY, S., PORTMANN, M. 2023. Dataset CIC-ToN-IoT. The University of Queensland, Australia. https://rdm.uq.edu.au/files/127784c0-ef9d-11ed-a964-b70596e96ad5.

SARNAN, M. 2023. The Detection of Network Cyber Attacks Using Machine Learning. Australia: The University of Queensland.

SHARIF, M.H.U dan MOHAMMED, A., 2022. A literature review of financial losses statistics for cyber security and future trend. World Journal of Advanced Research and Reviews 15, pp. 138-156.

TEODORO, P.G., VERDEJO, J.D., FERNANEZ, G.M., dan VAZQUEZ, E., 2009. Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), p.18–28.

TIDJON, L. N., FRAPPIER, M., dan MAMMAR, A., 2019. Intrusion Detection Systems: A CrossDomain Overview. IEEE Communications Surveys & Tutorials, 21(4).

TIM HONEYNET PROJECT BSSN, 2023. Jakarta: BADAN SIBER DAN SANDI NEGARA.

ZAINUDIN, Z., SHAMSUDDIN, S.M., dan HASAN, S., 2019. Deep learning for image processing in WEKA environment. International Journal of Advances in Soft Computing and its Applications, 11(1), p. 1-21.

ZHOU, Q. dan PEZAROS D., 2019. Evaluation of Machine Learning Classifiers for Zero-Daya Intrusion Detection-An Analysis on CIC-AWS-2018 Dataset. in ArXiv, abs/1905.03685.