Penerapan Algoritma Random Forest untuk Analisis dan Deteksi Dini Serangan Siber pada Lalu Lintas Jaringan

Yohanes Brian Gudare[1], Yulianti Paula Bria[2], Frengky Tedy[*3]

[1-3]Program Studi Ilmu Komputer, Universitas Katolik Widya Mandira Kupang

E-mail: ryangudare123@gmail.com[1], yulianti.bria@unwira.ac.id[2], frengkytedy@unwira.ac.id[*3]

Abstrak. Meningkatnya frekuensi dan kompleksitas serangan siber pada jaringan komputer menuntut adanya mekanisme deteksi yang adaptif dan andal. Penelitian ini bertujuan menerapkan algoritma Random Forest untuk menganalisis lalu lintas jaringan dan melakukan deteksi dini serangan siber secara otomatis. Dataset diperoleh dari hasil pemantauan lalu lintas jaringan pada lingkungan terkontrol dan berisi fitur-fitur seperti IP sumber, IP tujuan, port, protokol, panjang paket, waktu, serta label jenis trafik (normal dan beberapa jenis serangan). Data melalui tahap pra-pemrosesan, kemudian dibagi menjadi data latih dan data uji dengan rasio 70:30. Model Random Forest dibangun menggunakan pustaka scikit-learn dan dievaluasi dengan metrik akurasi, *precision*, *recall*, dan F1-*score*, baik per kelas maupun dalam bentuk macro- dan micro-averaging. Hasil pengujian menunjukkan akurasi keseluruhan sekitar 94%, dengan nilai F1-score rata-rata di atas 88%, yang mengindikasikan kemampuan model yang baik dalam membedakan trafik normal dan berbagai jenis serangan. Model kemudian diintegrasikan ke dalam sistem pemantauan jaringan berbasis web yang telah diuji secara fungsional menggunakan *black-box testing*.

Kata kunci: Deteksi serangan siber; lalu lintas jaringan; machine learning; Random Forest; evaluasi kinerja

Abstract. *The increasing frequency and complexity of cyberattacks on computer networks demands an adaptive and reliable detection mechanism. This research aims to apply the Random Forest algorithm to analyze network traffic and automatically detect cyberattacks early. The dataset was obtained from network traffic monitoring in a controlled environment and contains features such as source IP addresses, destination IP addresses, ports, protocols, packet lengths, timestamps, and traffic type labels (normal and several types of attacks). The data underwent preprocessing and was then divided into training and testing data in a 70:30 ratio. A Random Forest model was built using the scikit learn library and evaluated using accuracy, precision, recall, and F1 score metrics, both per class and in macro and micro averaging. The test results showed an overall accuracy of approximately 94%, with an average F1 score above 88%, indicating the model's strong ability to distinguish between normal traffic and various types of attacks. The model was then integrated into a web-based network monitoring system that was functionally tested using black-box testing.*

Keywords: *Cyberattack detection; network traffic; machine learning; Random Forest; performance evaluation*

1. Pendahuluan

Di era modern, teknologi informasi dan jaringan komputer telah menjadi bagian yang sangat penting dalam berbagai sektor, seperti bisnis, pemerintahan, pendidikan, dan layanan publik [1]. Ketergantungan yang tinggi terhadap sistem digital ini membuat aspek keamanan informasi menjadi semakin krusial. Serangan siber merupakan upaya untuk merusak, mengakses, memanipulasi, atau mencuri informasi dari sistem komputer maupun jaringan digital, yang dapat muncul dalam berbagai bentuk seperti malware, phishing, maupun serangan terhadap jaringan [2]. Serangan-serangan tersebut dapat menyebabkan kehilangan data, gangguan layanan, kerugian finansial, bahkan berdampak pada stabilitas ekonomi dan politik [3].

Sejumlah kajian di Indonesia menunjukkan bahwa frekuensi dan kompleksitas serangan siber cenderung meningkat dari tahun ke tahun dan mengancam berbagai sektor strategis [4]. Perkembangan internet dan keterhubungan sistem digital memungkinkan pelaku kejahatan siber melancarkan serangan dengan lebih mudah, cepat, dan menjangkau target yang lebih luas [5]. Kondisi ini membuat instansi pemerintah dan sektor industri yang menyimpan data penting dan sensitif menjadi semakin rentan terhadap kebocoran data dan gangguan operasional [6]. Situasi tersebut menuntut adanya mekanisme deteksi serangan yang adaptif pada lalu lintas jaringan sebagai jalur utama pertukaran data.

Pendekatan keamanan tradisional seperti sistem deteksi intrusi berbasis tanda tangan (*signature-based intrusion detection systems*) memiliki keterbatasan dalam menghadapi serangan baru maupun varian serangan yang belum terdokumentasi [7]. Di sisi lain, analisis manual terhadap lalu lintas jaringan dengan volume tinggi sulit dilakukan secara konsisten dan berpotensi melewatkan pola-pola serangan yang kompleks. Oleh karena itu, berbagai penelitian mengusulkan pemanfaatan teknik *machine learning* untuk meningkatkan kemampuan deteksi serangan pada jaringan komputer [8].

Machine learning memungkinkan sistem belajar secara otomatis dari data historis dan mengekstraksi pola tanpa perlu pemrograman ulang secara eksplisit. Dalam konteks keamanan jaringan, teknik pembelajaran mesin telah dimanfaatkan untuk mengenali pola lalu lintas normal dan mendeteksi adanya anomali atau intrusi sebagai indikasi serangan siber [9]. Selain itu, *machine learning* juga digunakan untuk tugas-tugas lain yang berkaitan dengan kualitas dan kinerja jaringan, seperti prediksi kinerja jaringan komputer [10].

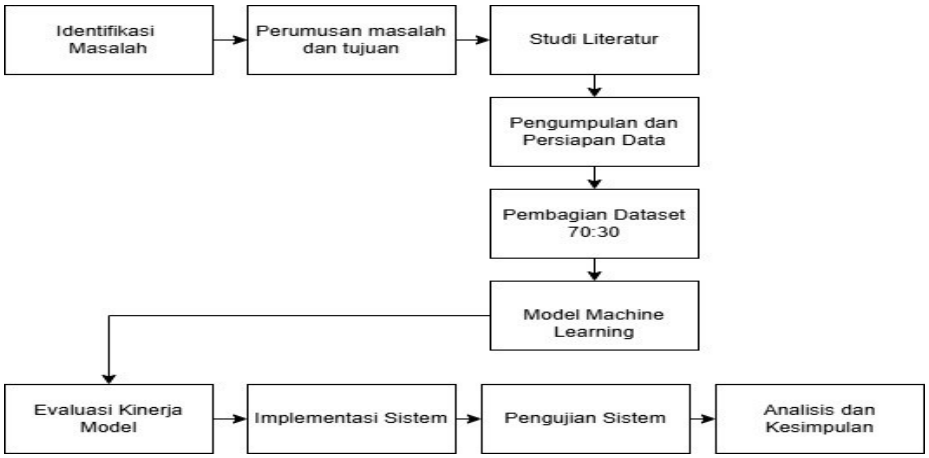
Salah satu algoritma *machine learning* yang banyak digunakan untuk tugas klasifikasi adalah Random Forest, yang menggabungkan sejumlah pohon keputusan (*decision tree*) dalam suatu kerangka *ensemble* [11]. Algoritma ini dikenal mampu menangani data berdimensi tinggi, mengurangi risiko *overfitting*, dan relatif robust terhadap *noise* pada data [12]. Berbagai kajian juga menjelaskan prinsip kerja Random Forest serta penerapannya pada beragam permasalahan klasifikasi dalam domain pembelajaran mesin [13]. Dalam ranah keamanan jaringan, Random Forest telah digunakan untuk mengklasifikasikan jenis serangan dan mendeteksi anomali lalu lintas jaringan dengan kinerja yang menjanjikan [14].

Berdasarkan uraian tersebut, permasalahan utama yang dikaji dalam penelitian ini adalah bagaimana menerapkan algoritma Random Forest untuk menganalisis lalu lintas jaringan sehingga mampu melakukan deteksi dini serangan siber dengan kinerja yang baik. Penelitian ini bertujuan untuk membangun dan mengevaluasi model Random Forest yang mengklasifikasikan lalu lintas jaringan menjadi dua kategori, yaitu normal dan serangan, dengan menggunakan dataset lalu lintas jaringan berlabel. Data yang digunakan pada penelitian ini terdiri dari 64 record yang diperoleh dari hasil uji coba sistem, yang merekam informasi lalu lintas jaringan selama proses monitoring. Setiap record memiliki sejumlah fitur utama, antara lain *destination IP address* (alamat IP tujuan paket), *source port* (port asal paket), *destination port* (port tujuan paket yang dikirim), *packet length* (ukuran paket), *packet type* (jenis paket yang diterima), *attack type* (status paket normal atau serangan), dan *time* (waktu saat paket dikirim). Dataset ini digunakan sebagai dasar analisis untuk mengidentifikasi pola aktivitas jaringan dan membangun model klasifikasi, yang selanjutnya dievaluasi menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score*. Ruang lingkup penelitian dibatasi pada deteksi serangan siber berbasis jaringan yang terekam dalam lalu lintas jaringan, tanpa membahas bentuk serangan lain di luar ranah jaringan maupun mekanisme penanggulangan dan pemulihan pasca serangan secara rinci. Hasil penelitian diharapkan memberikan kontribusi praktis berupa peningkatan kemampuan deteksi serangan pada lingkungan jaringan serta efisiensi pemantauan keamanan, sekaligus kontribusi

akademis berupa kajian empiris mengenai kinerja algoritma Random Forest untuk deteksi dini serangan siber yang dapat menjadi acuan dalam pengembangan sistem deteksi intrusi di masa mendatang.

2. Metode Penelitian

Untuk mencapai tujuan penelitian, disusun serangkaian tahapan yang terstruktur, mulai dari identifikasi masalah hingga analisis hasil dan penyusunan kesimpulan, seperti yang ditunjukkan pada Gambar 1



Gambar 1. Tahapan Penelitian

2.1. Tahapan penelitian

Tahapan penelitian pada gambar tersebut mengikuti alur sebagai berikut:

1. Identifikasi masalah
   Tahap ini dilakukan dengan mengamati meningkatnya ancaman kejahatan siber pada jaringan komputer dan keterbatasan metode deteksi tradisional. Hasil identifikasi ini menjadi dasar perumusan masalah dan kebutuhan sistem deteksi berbasis *machine learning*.
2. Perumusan masalah dan tujuan
   Berdasarkan identifikasi masalah, dirumuskan pertanyaan penelitian mengenai bagaimana menerapkan algoritma Random Forest untuk menganalisis lalu lintas jaringan dan melakukan deteksi dini serangan siber, serta ditetapkan tujuan untuk membangun dan mengevaluasi model tersebut. (Bagian ini sudah dijelaskan secara rinci di Pendahuluan.)
3. Studi literatur
   Pada tahap ini dikaji penelitian-penelitian terdahulu terkait keamanan jaringan, penerapan *machine learning* untuk deteksi intrusi, serta penggunaan algoritma Random Forest pada kasus klasifikasi serangan siber. Studi literatur digunakan sebagai landasan pemilihan metode dan perancangan eksperimen.
4. Pengumpulan dan persiapan data
   Data lalu lintas jaringan dikumpulkan dari hasil uji coba sistem pemantauan jaringan pada lingkungan terkontrol. Setiap entri (record) memuat informasi seperti IP sumber, IP tujuan, port sumber, port tujuan, protokol atau jenis paket, panjang paket, waktu kejadian, serta label jenis trafik (misalnya Normal, Exploit, Intrusion, SQL Injection).
5. Pembagian dataset 70:30
   Setelah data dibersihkan dan dipersiapkan, dataset dibagi menjadi data latih dan data uji dengan rasio 70% untuk pelatihan dan 30% untuk pengujian, menggunakan fungsi train_test_split dari pustaka scikit-learn. Data latih digunakan untuk membangun model, sedangkan data uji digunakan untuk menilai kinerja model secara objektif.
6. Pembangunan model *machine learning*

Pada tahap ini dibangun model klasifikasi berbasis algoritma Random Forest menggunakan data latih. Model dilatih untuk mengenali pola dari fitur-fitur jaringan dan mengklasifikasikan setiap entri menjadi kelas yang sesuai (normal atau jenis serangan tertentu).

7. Evaluasi kinerja model
   Model yang telah dilatih kemudian dievaluasi menggunakan data uji. Kinerja diukur dengan metrik klasifikasi seperti akurasi, *precision*, *recall*, dan *F1-score*, serta dianalisis menggunakan *confusion matrix* baik per kelas maupun secara keseluruhan.
8. Implementasi sistem
   Model Random Forest yang telah divalidasi diintegrasikan ke dalam prototipe sistem pemantauan jaringan. Sistem ini menampilkan aktivitas jaringan terkini serta visualisasi seperti distribusi jenis serangan, distribusi protokol, ukuran paket terhadap waktu, dan *timeline* serangan.
9. Pengujian sistem
   Tahap ini melakukan pengujian fungsional terhadap sistem menggunakan pendekatan *black-box testing* untuk memastikan bahwa fungsi-fungsi utama (pembacaan data, klasifikasi, dan visualisasi) berjalan dengan benar sesuai spesifikasi.
10. Analisis dan kesimpulan
    Hasil evaluasi model dan pengujian sistem dianalisis untuk menilai efektivitas Random Forest dalam mendeteksi serangan siber pada lalu lintas jaringan. Temuan-temuan utama kemudian disimpulkan dan diberikan saran pengembangan untuk penelitian selanjutnya.

2.2. Dataset dan Fitur

Dataset yang digunakan dalam penelitian ini berasal dari hasil uji coba sistem pemantauan lalu lintas jaringan pada lingkungan terkontrol. Proses monitoring menghasilkan 64 record data yang merepresentasikan aktivitas jaringan dalam periode pengamatan tertentu. Setiap record memuat informasi karakteristik paket jaringan dan label kelas trafik yang digunakan sebagai dasar pelatihan dan pengujian model.

Secara umum, fitur yang digunakan meliputi informasi alamat IP, port, protokol, ukuran paket, waktu, serta jenis trafik (normal atau jenis serangan tertentu). Deskripsi singkat masing-masing fitur ditunjukkan pada Tabel 1.

Tabel 1. Deskripsi Fitur Dataset

No	Nama Fitur	Tipe Data	Deskripsi Singkat
1	IP Sumber	String	Alamat IP asal paket
2	IP Tujuan	String	Alamat IP tujuan paket
3	Source Port	Integer	Port asal paket
4	Destination Port	Integer	Port tujuan paket
5	Protocol / Type	Kategorikal	Jenis protokol atau tipe paket (mis. TCP, UDP, ICMP)
6	Packet Length	Integer	Ukuran paket dalam satuan byte
7	Time	Waktu	Waktu saat paket dikirim atau direkam oleh sistem
8	Attack Type	Kategorikal	Label kelas trafik (mis. Normal, Exploit, Intrusion, SQL Injection)

Fitur *Attack Type* digunakan sebagai variabel target (kelas) dalam proses pelatihan model, sedangkan fitur lainnya berperan sebagai variabel input yang menggambarkan karakteristik lalu lintas jaringan.

2.3. Pra-pemrosesan dan Pembagian Data

Sebelum digunakan untuk pelatihan model, dataset melalui beberapa tahap pra-pemrosesan. Pertama, dilakukan pemeriksaan kelengkapan data untuk memastikan tidak terdapat nilai yang hilang atau inkonsistensi format pada setiap fitur. Record yang tidak lengkap atau mengandung kesalahan yang tidak dapat diperbaiki dihapus agar tidak mengganggu proses pelatihan.

Kedua, fitur yang bersifat kategorikal, seperti *Protocol* / *Type* dan *Attack Type*, dikonversi ke bentuk numerik melalui proses pengkodean (misalnya *label encoding*), sehingga seluruh fitur dapat diproses

oleh algoritma Random Forest. Jika diperlukan, fitur numerik seperti *Packet Length* dan nomor port dapat dinormalisasi atau diskalakan, meskipun secara umum Random Forest relatif tidak sensitif terhadap perbedaan skala fitur.
Secara formal, misalkan dataset berlabel dinotasikan sebagai:

$$D = \{(x_i, y_i)\}_{i=1}^{N}$$

dengan x_i adalah vektor fitur untuk record ke-i dan yi adalah label kelas (*Attack Type*). Setelah pra-pemrosesan, dataset dibagi menjadi himpunan data latih (D_{train}) dan data uji (D_{test}) menggunakan fungsi *train_test_split* dari pustaka *scikit-learn* dengan rasio 70% untuk data latih dan 30% untuk data uji, sehingga:

$$D_{\text{train}} \cup D_{\text{test}} = D, \quad D_{\text{train}} \cap D_{\text{test}} = \emptyset$$

Data latih digunakan untuk membangun model Random Forest, sedangkan data uji digunakan untuk mengevaluasi kinerja model pada data yang tidak pernah dilihat saat pelatihan.

2.4. Model Random Forest

Algoritma utama yang digunakan dalam penelitian ini adalah Random Forest, yaitu metode *ensemble learning* yang membangun sejumlah *decision tree* pada berbagai subset data latih dan menggabungkan hasil prediksinya melalui mekanisme *majority voting* untuk tugas klasifikasi [15]. Pendekatan ini bertujuan mengurangi variansi prediksi dan mengatasi kelemahan *overfitting* yang sering muncul pada *single decision tree* [16].
Model Random Forest pada penelitian ini diimplementasikan menggunakan pustaka *scikit-learn* [17]. Beberapa parameter utama yang diatur antara lain jumlah pohon (*n_estimators*), kedalaman maksimum pohon (*max_depth*), dan jumlah fitur yang dipertimbangkan pada setiap pemisahan (*max_features*) [16]. Nilai awal parameter ditentukan berdasarkan rekomendasi umum pada literatur dan kemudian disesuaikan melalui uji coba sederhana untuk memperoleh keseimbangan antara akurasi dan kompleksitas model. Proses pelatihan dilakukan dengan memberikan data latih sebagai input, di mana model mempelajari hubungan antara fitur-fitur jaringan dan label kelas (*Attack Type*), sehingga mampu mengklasifikasikan trafik menjadi kategori Normal dan beberapa jenis serangan.
Secara matematis, misalkan $h_b(x)$ adalah prediksi *decision tree* ke-b pada sampel x, dengan $b = 1, 2, \ldots, B$ adalah jumlah pohon dalam *Random Forest*. Untuk kasus klasifikasi multi-kelas, prediksi akhir $\hat{y}$ diperoleh melalui *majority voting*:

$$\hat{y} = \arg\max_{k} \sum_{b=1}^{B} 1\{h_b(x) = k\}$$

dengan:

a. k menyatakan salah satu kelas yang mungkin (misalnya Normal, Exploit, Intrusion, SQL Injection),
b. 1 {.} adalah fungsi indikator yang bernilai 1 jika kondisi di dalam kurung terpenuhi, dan 0 jika tidak.

Artinya, kelas yang dipilih sebagai prediksi akhir adalah kelas yang paling sering diprediksi oleh seluruh pohon dalam *Random Forest*.

2.5. Evaluasi Kinerja Model

Kinerja model Random Forest dievaluasi menggunakan data uji D_{test}. Hasil prediksi model dibandingkan dengan label sebenarnya untuk menghitung berbagai metrik evaluasi klasifikasi. Evaluasi dilakukan berdasarkan *confusion matrix* dan beberapa metrik turunan, yaitu *accuracy*, *precision*, *recall*, dan F1-*score*, baik per kelas maupun dalam bentuk rata-rata (*macro* dan *micro*).

2.5.1. ***Confusion Matrix*** **dan Definisi TP, TN, FP, FN**

Untuk setiap kelas *k*, hasil klasifikasi dapat dirangkum dalam *confusion matrix* yang memuat empat komponen utama:

1. *True Positive* (TP_k): jumlah *instance* kelas *k* yang diprediksi benar sebagai kelas *k*,
2. *False Positive* (FP_k): jumlah *instance* dari kelas lain yang salah diprediksi sebagai kelas *k*,
3. *False Negative* (FN_k): jumlah *instance* kelas *k* yang salah diprediksi sebagai kelas lain,
4. *True Negative* (TN_k): jumlah *instance* dari kelas lain yang diprediksi dengan benar bukan sebagai kelas *k*.

Nilai-nilai ini menjadi dasar perhitungan metrik evaluasi per kelas.

2.5.2 Rumus Metrik Per Kelas

Berdasarkan nilai TP, FP, FN, dan TN untuk suatu kelas *k*, metrik evaluasi dihitung dengan rumus sebagai berikut:

1. *Akurasi (Accuracy):*

$$\text{Accuracy}_k = \frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k}$$

2. *Precision:*

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k}$$

3. *Recall:*

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k}$$

4. *F1-score:*

$$\text{F1}_k = 2 \cdot \frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}$$

Dalam praktik evaluasi multi-kelas, *accuracy* biasanya dihitung secara global (berdasarkan total TP dan TN seluruh kelas), sedangkan *precision*, *recall*, dan F1-*score* dihitung per kelas.

2.5.3. ***Macro-Averaging dan Micro-Averaging***

Untuk memperoleh gambaran kinerja model secara keseluruhan pada kasus multi-kelas dengan *k* kelas, digunakan dua pendekatan perata-rataan, yaitu:

1. *Macro-averaging*, yaitu rata-rata sederhana nilai metrik per kelas:

$$\text{Precision}_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K} \text{Precision}_k$$

$$\text{Recall}_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K} \text{Recall}_k$$

$$\text{F1}_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K} \text{F1}_k$$

2. *Micro-averaging*, yaitu perhitungan metrik berdasarkan agregasi global TP, FP, dan FN semua kelas:

$$\text{Precision}_{\text{micro}} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} (TP_k + FP_k)}$$

$$\text{Recall}_{\text{micro}} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} (TP_k + FN_k)}$$

$$\text{F1}_{\text{micro}} = 2 \cdot \frac{\text{Precision}_{\text{micro}} \cdot \text{Recall}_{\text{micro}}}{\text{Precision}_{\text{micro}} + \text{Recall}_{\text{micro}}}$$

Pendekatan macro memberikan bobot yang sama untuk setiap kelas, sedangkan micro lebih mencerminkan kontribusi kelas dengan jumlah *instance* yang besar. Sebagai ringkasan, Tabel 2 memuat rumus dasar metrik evaluasi yang digunakan.

Tabel 2. Rumus Metrik Evaluasi

Metrik	Rumus Utama
Accuracy	(TP+TN)/(TP+TN+FP+FN)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1-score	$2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

2.6. Pengujian Fungsional Sistem

Selain evaluasi model, dilakukan juga pengujian fungsional sistem untuk memastikan bahwa implementasi model Random Forest di dalam aplikasi web berjalan sesuai dengan kebutuhan. Pengujian dilakukan dengan pendekatan *black-box testing*, di mana penguji hanya memperhatikan masukan dan keluaran sistem tanpa melihat proses internal.

Skenario pengujian mencakup antara lain:

1. Pengujian form login admin dengan kombinasi username dan password yang benar maupun salah,
2. Pengujian halaman konfigurasi pemantauan (pengisian alamat IP, port, dan pengaktifan pemantauan),
3. Pengamatan tampilan dashboard ketika serangan tertentu (misalnya DDoS, SQL *Injection*, ICMP *Flood*, DNS *Amplification*) dijalankan pada lingkungan uji, serta pengujian fungsi logout.

Setiap skenario diuji dengan langkah masukan tertentu dan dibandingkan dengan hasil yang diharapkan (misalnya penolakan akses, pesan galat, atau tampilan jenis serangan pada dashboard). Sistem dinyatakan lolos pengujian fungsional apabila seluruh skenario menghasilkan keluaran sesuai spesifikasi dan tidak ditemukan kesalahan yang mengganggu proses deteksi maupun penyajian informasi kepada pengguna.

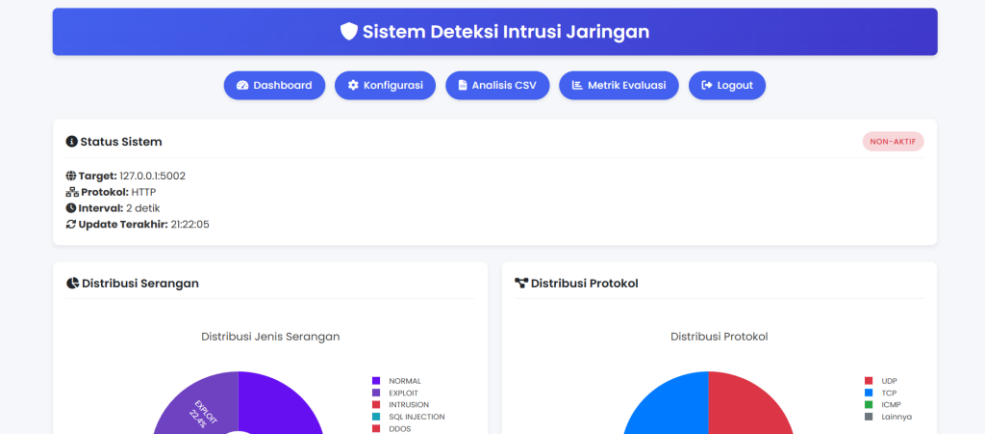
3. Hasil dan Pembahasan

3.1. Implementasi Sistem

Sistem deteksi serangan yang dibangun diimplementasikan dengan menggunakan aplikasi berbasis web. Aplikasi ini terdiri dari beberapa komponen antarmuka utama, yaitu halaman *login admin*, *dashboard* pemantauan, halaman konfigurasi, dan halaman evaluasi (*perhitungan*). Halaman login admin berfungsi sebagai gerbang autentikasi sebelum pengguna dapat mengakses fitur pemantauan dan evaluasi. Admin harus memasukkan kombinasi *username* dan *password* yang benar agar dapat diarahkan ke halaman *dashboard*, seperti yang ditunjukkan pada Gambar 2.

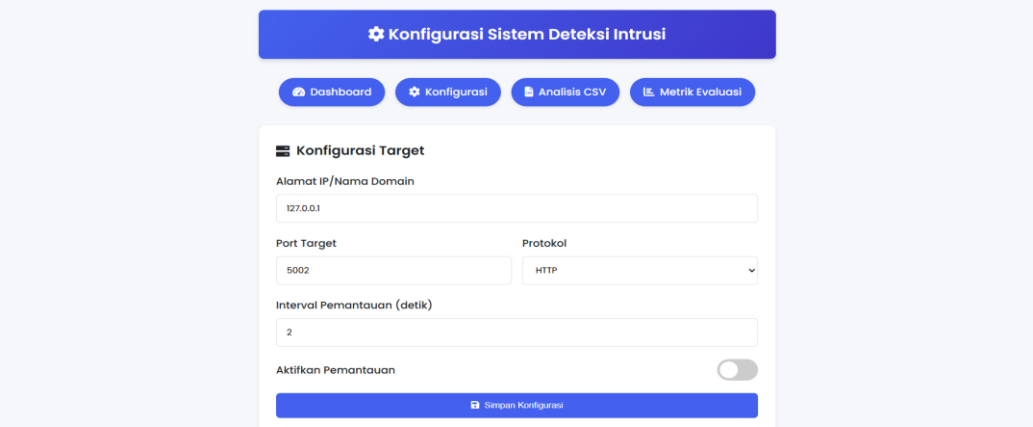
Gambar 2. Halaman Login Admin

Halaman *dashboard* menampilkan aktivitas jaringan terkini dalam bentuk tabel yang memuat informasi waktu, IP sumber, IP tujuan, protokol, ukuran paket, port, serta jenis serangan yang terdeteksi. Pada halaman ini juga tersedia menu untuk mengakses konfigurasi pemantauan, halaman perhitungan, *log* aktivitas, dan pengaturan sistem, seperti yang ditunjukkan pada Gambar 3.



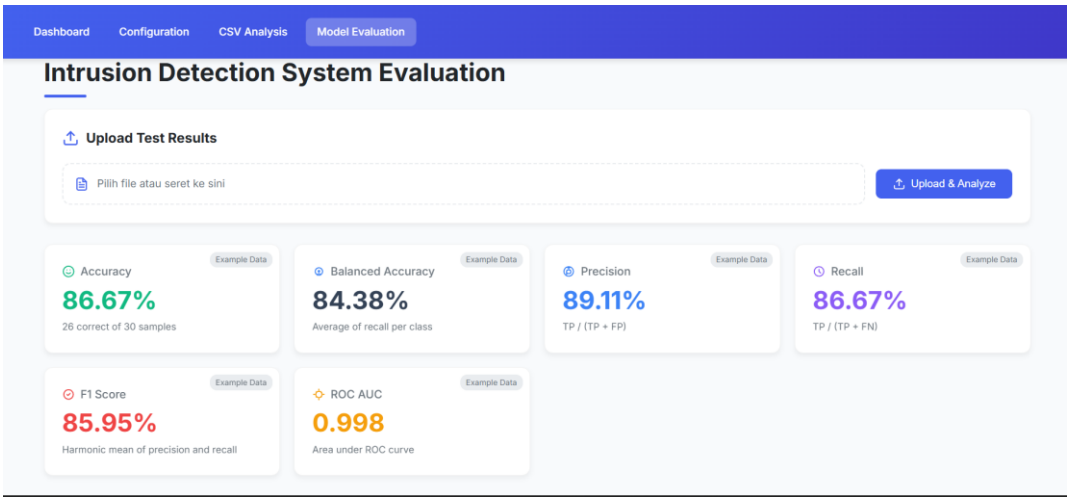
***Gambar 3. Halaman* Dashboard**

Halaman konfigurasi memungkinkan admin mengatur parameter pemantauan, seperti alamat IP atau domain target, port tujuan, jenis protokol (misalnya HTTP), serta interval pemantauan dalam satuan detik. Admin juga dapat mengaktifkan atau menonaktifkan proses pemantauan melalui saklar (*toggle*). Setelah pengaturan disimpan, sistem akan menggunakan konfigurasi ini untuk melakukan *sniffing* lalu lintas jaringan dan mengirimkannya ke model deteksi, seperti yang ditunjukkan pada Gambar 4.



Gambar 4. Halaman Konfigurasi

Halaman perhitungan digunakan untuk mengevaluasi performa model deteksi intrusi terhadap dataset uji yang diunggah dalam format CSV. Pada halaman ini ditampilkan berbagai metrik evaluasi seperti akurasi, precision, recall, F1-score, *balanced accuracy*, dan ROC AUC, sehingga admin dapat menilai sejauh mana model mampu membedakan aktivitas normal dan berbagai jenis serangan, seperti yang ditunjukkan pada Gambar 5.



Gambar 5. Halaman Perhitungan

3.2. Hasil Klasifikasi Model Random Forest

Model Random Forest dilatih dan diuji menggunakan dataset hasil monitoring lalu lintas jaringan yang telah melalui tahap pembersihan dan pemrosesan awal. Dataset berisi 64 entri dengan fitur-fitur utama seperti IP sumber, IP tujuan, port sumber, port tujuan, protokol, panjang paket, waktu, dan label jenis trafik (Normal, *Exploit*, *Intrusion*, SQL *Injection*, dan seterusnya.).

Berdasarkan *confusion matrix* yang diperoleh dari data uji, model berhasil mengklasifikasikan lalu lintas jaringan dengan akurasi keseluruhan sebesar 94,53%. Nilai precision, recall, dan F1-score rata-rata (macro-averaging) masing-masing adalah 88,03%, 88,72%, dan 88,27%, sedangkan nilai *micro-averaging* menunjukkan precision, recall, dan F1-*score* sebesar 89,06%. Hasil ringkasan metrik per kelas ditunjukkan pada Tabel 3.

Tabel 3. Ringkasan Metrik Per Kelas

Jenis Serangan	*Precision*	*Recall*	*F1 Score*	*Accuracy*
SQL Injection	82.35%	87.5%	84.85%	92.19%

Exploit	92.31%	85.71%	88.89%	95.31%
Intrusion	81.82%	90%	85.71%	95.31%
Normal Traffic	95.65%	91.67%	93.62%	95.31%

Untuk melihat performa model secara keseluruhan pada kasus multi-kelas, dilakukan pula perhitungan rata-rata *macro* dan *micro* dari metrik-metrik tersebut. Metrik rata-rata macro dan micro dirangkum pada Tabel 4.

Tabel 4. Ringkasan Nilai Macro dan Micro

Metrik	Macro-Averaging	Micro-Averaging
Precision	88,03%	89,06%
Recall	88,72%	89,06%
F1-Score	88,27%	89,06%
Accuracy	–	94,53%

Dari sudut pandang per kelas, trafik Normal memiliki nilai precision dan F1-score tertinggi (sekitar 95,65% dan 93,62%), yang menunjukkan bahwa model sangat jarang salah mengklasifikasikan aktivitas normal sebagai serangan. Untuk jenis serangan, kategori Exploit dan SQL Injection memiliki nilai F1-score di atas 84%, sedangkan Intrusion sedikit lebih rendah namun masih berada dalam kisaran yang dapat diterima. Hal ini menunjukkan bahwa model Random Forest cukup baik dalam membedakan beberapa tipe serangan, meskipun masih terdapat sejumlah kesalahan klasifikasi antarjenis serangan yang memiliki pola mirip.

Secara umum, kombinasi nilai akurasi di atas 94% dan nilai F1-score rata-rata di atas 88% menunjukkan bahwa Random Forest mampu mempelajari pola dari fitur-fitur jaringan yang digunakan dan memberikan performa yang kompetitif untuk tugas deteksi serangan multi-kelas pada dataset ini.

Jika Anda ingin memasukkan juga hasil dari halaman CSV-metrics (86,67% akurasi, ROC AUC 0,998), bisa ditulis sebagai uji tambahan: Pada pengujian dengan dataset uji terpisah yang diunggah melalui halaman evaluasi, model mencapai akurasi 86,67% dengan ROC AUC 0,998, yang mengindikasikan kemampuan pemisahan kelas yang sangat baik.

3.3. Analisis Visualisasi Lalu Lintas Jaringan

Selain metrik numerik, sistem juga menyajikan beberapa visualisasi untuk membantu analisis pola serangan. Grafik distribusi jenis serangan menunjukkan bahwa trafik normal masih mendominasi, namun proporsi serangan bertipe Exploit dan Intrusion cukup signifikan, sedangkan serangan SQL Injection dan jenis lain seperti DDoS atau Malware muncul dengan frekuensi yang lebih rendah.

Diagram distribusi protokol memperlihatkan bahwa protokol UDP lebih sering digunakan dibandingkan TCP dalam lalu lintas yang dipantau, sehingga dapat menjadi fokus pemantauan lebih lanjut. Grafik ukuran paket terhadap waktu dan *timeline* serangan menggambarkan variasi intensitas dan jenis serangan pada periode pengamatan tertentu, yang dapat dimanfaatkan untuk mengidentifikasi pola serangan berulang atau lonjakan aktivitas tidak wajar.

Visualisasi ini melengkapi hasil klasifikasi model dengan memberikan konteks yang lebih intuitif bagi admin jaringan dalam memahami tren dan sebaran serangan pada sistem yang dipantau.

3.4. Evaluasi Fungsional Sistem

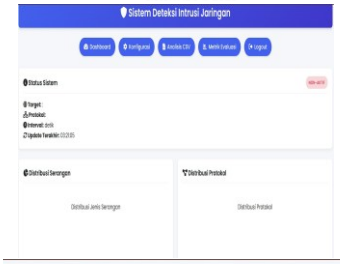
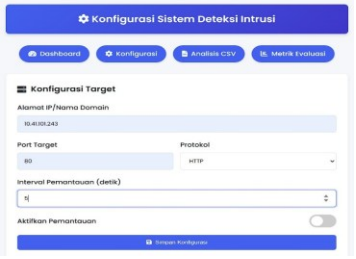
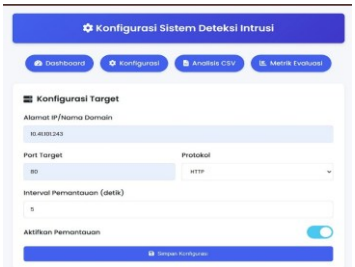
Pengujian fungsional sistem dilakukan dengan pendekatan *black-box testing* untuk memastikan bahwa seluruh fitur utama berjalan sesuai dengan kebutuhan [18]. Skenario pengujian meliputi:

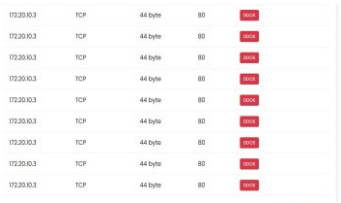
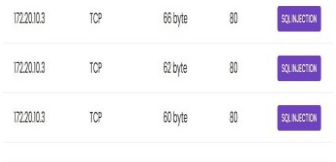
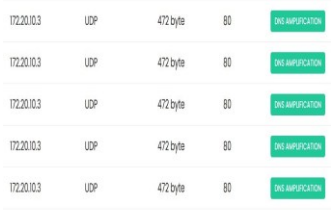
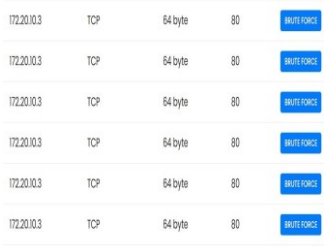
1. Validasi input pada halaman login (kombinasi username/password benar dan salah),
2. Validasi isian konfigurasi alamat IP dan port target,
3. Pemantauan tampilan dashboard saat serangan DDoS, SQL Injection, ICMP Flood, dan DNS Amplification dijalankan pada halaman uji coba,

4. Fungsi logout untuk mengakhiri sesi admin.

Rangkuman skenario pengujian fungsional sistem dan hasil aktualnya disajikan pada Tabel 5.

Tabel 5. Hasil Pengujian

No	Uji Fitur	Langkah Uji	Hasil	Tampilan
1	Admin hanya Mengisi *username* tanpa mengisi *password*	Admin hanya Mengisi *username* tanpa mengisi *password*	Sistem akan menolak dan muncul notifikasi untuk mengisi *password*	
2	*Admin* hanya mengisi *password*	*Admin* hanya mengisi *password* tanpa mengisi *username*	Sistem akan menolak dan muncul notifikasi untuk mengisi *username*	
3	Admin mengisi username dan password salah	*Admin* mengisi *username* dan *password* salah lalu klik tombol masuk	Sistem akan menolak akses dan muncul tulisan *username* atau *password* salah	
4	*Admin* mengisi *username* dan *password* benar	*Admin* mengisi *username* dan *password* benar	*Admin* mengisi *username* dan *password* benar lalu klik tombol masuk akan langsung diarahkan ke halaman *dashboard*	
5	*Admin* mengisi alamat IP dan Port Target tapi tidak aktif	*Admin* mengisi alamat IP dan Port Target namun mengaktifkan pemantauan	Sistem tidak akan menampilkan dashboard yang melakukan sniffing	
6	*Admin* mengisi alamat IP dan Port Target tapi aktif	*Admin* mengisi alamat IP dan Port Target namun mengaktifkan pemantauan	Sistem akan menampilkan dashboard yang sudah melakukan sniffing	

7	*Admin* melakukan serangan DDOS	*Admin* Melakukan serangan DDOS pada halaman uji coba	Sistem akan mendeteksi serangan tersebut dan menampilkan nya pada halaman *dashboard*	
8	Admin melakukan serangan SQL INJECTION	*Admin* Melakukan serangan *SQL INJECTION* pada halaman uji coba	Sistem akan mendeteksi serangan tersebut dan menampilkan nya pada halaman *dashboard*	
9	*Admin* melakukan serangan *ICMP Flood*	*Admin* Melakukan serangan *ICMP Flood* pada halaman uji coba	Sistem akan mendeteksi serangan tersebut dan menampilkan nya pada halaman *dashboard*	
10	Admin melakukan serangan DNS Amplification	*Admin* Melakukan serangan *DNS Amplification* pada halaman uji coba	Sistem akan mendeteksi serangan tersebut dan menampilkan nya pada halaman *dashboard*	
11	Admin melakukan serangan DNS Amplification	*Admin* Melakukan serangan *DNS Amplification* pada halaman uji coba	Sistem akan mendeteksi serangan tersebut dan menampilkan nya pada halaman *dashboard*	
12	Logout	*Admin* mengklik tombol *logout*	Kembali ke tampilan awal untuk melakukan *login*	

Dari hasil pengujian yang dirangkum pada Tabel 5 menunjukkan bahwa seluruh kasus uji menghasilkan keluaran yang sesuai dengan harapan, yaitu sistem menolak kredensial yang salah, menampilkan pesan kesalahan ketika konfigurasi pemantauan tidak valid, dan mampu menampilkan jenis serangan yang dilakukan pada dashboard secara real-time. Hal ini menunjukkan bahwa integrasi antara model Random Forest dan antarmuka sistem telah berjalan dengan baik dari sisi fungsionalitas.

4. Kesimpulan

Penelitian ini telah menerapkan algoritma Random Forest untuk menganalisis lalu lintas jaringan dan mendeteksi serangan siber secara otomatis melalui sebuah sistem pemantauan berbasis web. Model Random Forest dilatih menggunakan dataset lalu lintas jaringan yang berisi fitur-fitur seperti alamat IP, port, protokol, panjang paket, waktu, dan label jenis trafik. Berdasarkan hasil pengujian

menggunakan data uji, model mampu mengklasifikasikan trafik ke dalam beberapa kategori, termasuk trafik normal dan beberapa jenis serangan (misalnya Exploit, *Intrusion*, dan SQL *Injection*), dengan akurasi keseluruhan sekitar 94,53% serta nilai F1-score rata-rata sekitar 88%.

Hasil ini menunjukkan bahwa *Random Forest* efektif dalam menangkap pola serangan pada data jaringan yang digunakan dan layak dipertimbangkan sebagai komponen deteksi dini serangan siber. Integrasi model ke dalam sistem pemantauan jaringan memungkinkan admin untuk melihat aktivitas jaringan secara real-time, termasuk jenis serangan yang terdeteksi dan distribusinya berdasarkan tipe serangan maupun protokol yang digunakan. Pengujian fungsional dengan pendekatan *black-box testing* juga menunjukkan bahwa fitur-fitur utama sistem, seperti login, konfigurasi pemantauan, tampilan dashboard, dan evaluasi kinerja model, telah berjalan sesuai dengan kebutuhan.

Meskipun demikian, penelitian ini masih memiliki beberapa keterbatasan, di antaranya ukuran dan keragaman dataset yang relatif terbatas dan penggunaan satu algoritma utama. Oleh karena itu, pengembangan selanjutnya dapat diarahkan pada penggunaan dataset yang lebih besar dan beragam, perbandingan dengan algoritma lain, serta integrasi dengan mekanisme peringatan otomatis dan penyimpanan data ke basis data untuk analisis jangka panjang.

5. Ucapan Terima Kasih

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada Dosen Pembimbing pada Program Studi Ilmu Komputer yang telah memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses penyusunan penelitian ini. Ucapan terima kasih juga penulis sampaikan kepada orang tua yang senantiasa memberikan dukungan moral, doa, dan motivasi, serta kepada teman-teman seangkatan yang telah banyak membantu dan memberikan semangat dalam menyelesaikan penelitian ini.

6. Referensi

[1] Y. Sutisnawinata, Keamanan Siber di Indonesia: Ancaman dan Solusi. Jakarta, Indonesia: Pustaka Teknologi, 2023.
[2] B. Simanullang, R. Syuhada, M. B. Lewa, and D. A. Martin, “Penerapan machine learning untuk deteksi ancaman siber,” Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 8, no. 2, pp. 120–130, 2021.
[3] R. N. Uzlah, R. Damanik, Y. Ramadhan, and M. Faizal, “Analisis dampak serangan siber terhadap stabilitas ekonomi dan politik di Indonesia,” Jurnal Keamanan Siber Nasional, vol. 6, no. 1, pp. 25–35, 2024.
[4] I. Setyawan, T. R. Putra, and R. A. Lestari, “Klasifikasi tingkat keparahan serangan jaringan komputer dengan metode machine learning,” Jurnal Teknik Informatika dan Sistem Informasi, vol. 8, no. 2, pp. 101–108, 2021.
[5] A. Situmorang and M. Yafhizam, “Analisis kinerja algoritma machine learning dalam deteksi anomali jaringan,” Jurnal Teknologi dan Sistem Komputer, vol. 11, no. 1, pp. 45–52, 2023.
[6] R. Bororing, “Pengembangan algoritma machine learning untuk mendeteksi anomali jaringan pada komputer,” Jurnal Ilmu Komputer dan Informatika, vol. 6, no. 1, pp. 33–40, 2024.
[7] P. Sari, H. Kurniawan, and L. Mulyani, “Implementasi machine learning untuk deteksi intrusi pada jaringan komputer,” Jurnal Teknologi Informasi dan Keamanan Siber, vol. 5, no. 2, pp. 58–65, 2024.
[8] A. H. Rohman, E. Prasetyo, and R. Setiawan, “Klasifikasi serangan jaringan menggunakan algoritma Random Forest,” Jurnal Ilmiah Teknik Komputer, vol. 4, no. 3, pp. 145–150, 2018.
[9] U. Laode, “Deteksi serangan siber pada jaringan komputer menggunakan metode Random Forest,” JATI (Jurnal Aplikasi Teknologi Informasi), vol. 8, no. 2, pp. 4–5, 2024.
[10] D. A. Putra, “Prediksi kinerja jaringan komputer menggunakan model machine learning,” Jurnal Informatika dan Sistem Komputer, vol. 9, no. 1, pp. 22–29, 2024.
[11] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[12] A. Liaw and M. Wiener, “Classification and regression by randomForest,” R News, vol. 2, no. 3, pp. 18–22, 2002.

[13] A. Kantinit, "Pengantar algoritma Random Forest dan aplikasinya dalam machine learning," Jurnal Teknologi Informasi, vol. 5, no. 2, pp. 45–52, 2022.
[14] R. Moskovitch and Y. Shahar, "Detection of anomalies in network traffic using random forests," in Proc. IEEE Int. Conf. Data Mining, 2008, pp. 270–279.
[15] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[16] A. Liaw and M. Wiener, "Classification and regression by randomForest," R News, vol. 2, no. 3, pp. 18–22, 2002. [Online]. Available: https://cran.r-project.org/doc/Rnews/
[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html
[18] M. Nurudin, "Pengujian perangkat lunak: Pendekatan black box testing," Jurnal Teknologi Informasi, vol. 6, no. 1, pp. 23–29, 2020.