Deteksi Anomali pada Lalu Lintas Jaringan Lokal Area Network Menggunakan Metoda Random Forest

[1]Marisa Premitasari, [2]Rizka Milandga Milenio, [3]Mohamad Yusup Suryaman, [4]Doddy Setiyadi, [5]Choerunnisa Septiani Tri Noerdin

[1,2,3,4] Fakultas Teknologi Industri , Institut Teknologi Nasional, Jl PHH Mustofa no 23 Bandung

email: marisa@itenas.ac.id

Abstract

Anomaly detection in network traffic is crucial in maintaining the security and stability of information systems. Increasingly complex traffic network demand effective and accurate detection methods. This study aims to detect anomalies in network traffic using the Random Forest method, which is known to have high performance in data classification. In this study, network traffic data was collected and analyzed based on the IP (Internet Protocol) Source, IP Destination, Protocol, and Length features. Using the default Random Forest method, the evaluation value for the anomaly class was obtained with a precision value of 77%, a recall value of 87% and an f1-score of 81%. While the hyperparameter tuning model successfully obtained a precision value of 59%, a recall value of 99% and an f1-score of 74%. The results of this study indicate that the Random Forest method with hyperparameter tuning is very sensitive in detecting anomalies with a recall value of 99%. These findings are expected to help improve network analysis by providing early warnings of suspicious activity or changes in network patterns.

Keywords: Anomaly Detection; IP Network Traffic; Random Forest; Hyperparameter; Length

Abstrak

Deteksi anomali pada lalu lintas jaringan menjadi hal yang krusial dalam menjaga keamanan dan kestabilan sistem informasi. Lalu lintas jaringan yang semakin kompleks menuntut adanya metode deteksi yang efektif dan akurat. Penelitian ini bertujuan untuk mendeteksi anomali pada lalu lintas jaringan menggunakan metode Random Forest, yang dikenal memiliki performa tinggi dalam klasifikasi data. Dalam penelitian ini, data lalu lintas jaringan dikumpulkan dan dianalisis berdasarkan fitur IP (Internet Protokol) Source, IP Destination, Protocol, dan Length. Dengan menggunakan metode Random Forest model default didapatkan nilai evaluasi untuk kelas anomali dengan nilai presisi 77%, nilai recall 87% dan f1-score 82%, sedangkan model hyperparameter tuning berhasil mendapatkan nilai presisi 59%, nilai recall 99% dan f1-score 74%. Hasil penelitian ini menunjukkan bahwa metode Random Forest dengan hyperparameter tuning sangat sensitif dalam mendeteksi anomali dengan nilai recall 99%. Temuan ini diharapkan dapat membantu meningkatkan analisis jaringan dengan memberikan peringatan dini terhadap aktivitas mencurigakan atau perubahan pola jaringan

Keywords : Deteksi Anomali; Trafik Jaringan IP , Random Forest, Hyperparameter, Length

1. PENDAHULUAN

Lalu lintas jaringan (*network traffic*) adalah kumpulan paket data yang melintasi jaringan komputer dalam periode tertentu. Paket ini membawa informasi dari satu host ke host lain menggunakan protokol komunikasi seperti TCP, UDP, atau ICMP. Analisis lalu lintas jaringan diperlukan untuk mendeteksi pola normal dan mengidentifikasi aktivitas abnormal yang dapat menjadi indikasi serangan. Menurut Inixindo (Inixindo, 2021), pemantauan lalu lintas jaringan penting untuk menjaga stabilitas sistem, mendeteksi gangguan lebih awal, dan mencegah kebocoran data. Ketika trafik meningkat secara signifikan atau muncul paket-paket mencurigakan, ini bisa menandakan serangan seperti *flooding* atau *scanning*.

Anomali secara umum merujuk pada objek, kejadian, atau pengamatan yang menyimpang secara signifikan dari pola mayoritas data (Aggarwal, 2019). Deteksi anomali penting karena dapat mengindikasikan terjadinya kesalahan, penyalahgunaan, atau aktivitas berbahaya (Hodge, 2023). Anomali jaringan dapat terjadi karena berbagai faktor seperti serangan siber, kesalahan konfigurasi, atau lalu lintas abnormal. Deteksi anomali jaringan bertujuan mengidentifikasi pola ini untuk mencegah gangguan layanan dan kebocoran data (Raghavendra Chalapathy, 2019). Tantangan utama dalam deteksi anomali adalah ketidakseimbangan data (*class imbalance*), keberagaman pola serangan, serta dinamika lalu lintas jaringan yang terus berubah (Jan Michael Spoor, 2023)

Metode berbasis *machine learning* banyak digunakan untuk mendeteksi anomali jaringan karena mampu mempelajari pola data dan mengenali perbedaan antara trafik normal dan anomali. Beberapa penelitian sebelumnya telah menerapkan algoritma seperti *Naïve Bayes*, *Support Vector Machine* (SVM), dan *k-Nearest Neighbor* (k-NN) untuk klasifikasi anomali jaringan (Ghazi Al-Naymat, 2018); (Imam Riadi, 2019). Namun, tantangan utama masih terletak pada akurasi klasifikasi serta kemampuan algoritma untuk menangani data berdimensi tinggiparagraf menjorok adalah 0,35 cm. Tipe ukuran dan jenis huruf mengikuti ukuran yang telah dicantumkan dalam Tabel I.

Random Forest (RF) merupakan salah satu algoritma klasifikasi berbasis *ensemble learning* yang menggabungkan banyak pohon keputusan (*decision tree*) untuk meningkatkan akurasi serta mengurangi *overfitting* (Breiman, 2001). Beberapa studi menunjukkan bahwa RF memiliki performa yang baik dalam mendeteksi anomali pada data berukuran besar maupun berdimensi tinggi (Ariyoga, 2022)

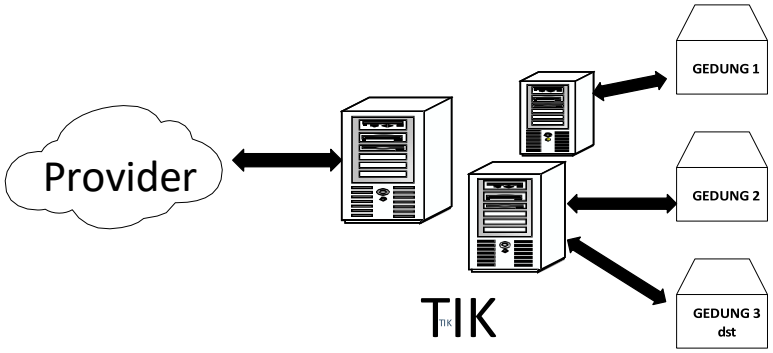
Penelitian oleh (Assiri, Al-Haidari, & Al-Ghamdi, 2020) menyoroti pentingnya optimasi parameter dengan menggunakan algoritma genetika, yang berhasil meningkatkan akurasi *Random Forest* hingga 97,2% pada dataset KDD99. Hal ini membuktikan bahwa *tuner hyperparameter* dapat secara signifikan meningkatkan kinerja model. Oleh karena itu, penelitian ini menjadi relevan untuk mengeksplorasi secara spesifik bagaimana *Random Forest* dapat dioptimalkan melalui *hyperparameter tuning* untuk meningkatkan akurasi deteksi anomali pada lalu lintas jaringan lokal.

Penelitian ini dimulai dengan melakukan sniffing paket data di sebuah jaringan lokal kampus dengan bantuan *monitoring tools* .Deteksi anomali akan diterapkan pada hasil sniffing data melalui algoritma *Random Forest* dengan model *default* dan model *hyperparameter tuning* untuk melihat performa model mana yang paling baik untuk mendeteksi anomali pada lalu lintas jaringan lokal menggunakan dataset yang didapat dengan cara *sniffing* menggunakan *software Wireshark* dan *TCPdump*.

2. KERANGKA TEORI

2.1. IP Traffic untuk Lingkungan Kampus

Jaringan telekomunikasi saat ini sudah berbasis Internet Protocol (IP) dimana pada era 4G semua jaringan telekomunikasi menggunakan IP Address sebagai protokol komunikasi . Hal itu berarti bahwa semua divais terhubung dengan internet dan trafiknya adalah trafik data. Protokol-protokol yang mengatur trafik data ini akan mengirim dan mengirimkan kembali sinyal dalam bentuk paket. Jaringan IP traffic ini dapat dikelompokkan sesuai kapasitas area penggunanya (Premitasari, 2020) . Gambar 1 menunjukkan flow traffic jaringan kampus dimana TIK (Teknologi Informasi dan Komunikasi) sebagai core jaringan telekomunikasi kampus melayani host-host yang terdiri dari gedung-gedung yang dipasang akses point dengan mengakses sebuah layanan server



Gambar 2. Flow Traffic Lingkungan Kampus

2.2. Deteksi Anomali

Anomali secara umum merujuk pada objek, kejadian, atau pengamatan yang menyimpang secara signifikan dari pola mayoritas data (Aggarwal, 2019). Deteksi anomali penting karena dapat mengindikasikan terjadinya kesalahan, penyalahgunaan, atau aktivitas berbahaya (Hodge, 2023). Dalam konteks jaringan komputer dan lalu lintas jaringan , anomali mengacu pada perilaku lalu lintas jaringan yang tidak biasa, yang berbeda dari profil normal. Contohnya termasuk serangan *DDoS (Distribution Denial of Services)* yang mampu membuat sistem menjadi berjalan sangat lambat karena ada *request* tak dikenal yang datang bertubi-tubi (K. H. Purwanto, 2019). Anomali juga dapat dilihat melalui *port scanning*, penyebaran malware atau aktivitas intrusi yang memanfaatkan celah keamanan (Palindungan Tampubolon, 2024). Anomali pada jaringan salah satunya dapat diuji ketika sebuah sistem sedang tidak diberi protokol pengaman data contohnya *HTTPS (Hypertext Tranfer Protocol Secure)* dan *tools* monitoring seperti *Wireshark* dapat membaca *username* dan *password* di satu jaringan yang sama. Anomali pada *Wireshark* dapat dilihat melalui lonjakan trafik tiba-tiba, *unusual-protocol* atau *packet length* yang abnormal. Anomali pada Wireshark salah satunya ditandai oleh fitur expert info sebagai analisis diagnostic yang menampilkan peringatan, kesalahan dan membantu analisis jaringan dalam identifikasi anomali seperti retransmisi, out of order packets, handshake error hingga serangan tertentu (Wireshark Foundation, 2025). Ada empat klasifikasi expert info sesuai tingkat keparahan anomali yait: 1. ***Error*:** Menunjukkan adanya kesalahan serius dalam komunikasi (gangguan jaringan), misalnya *checksum error* atau *invalid header*. 2.**Warning**: Menunjukkan potensi masalah yang dapat memengaruhi performa

(perubahan perilaku yang mendadak), misalnya *duplicate ACK* atau *zero window.* 3. **Note** : memberikan catatan yang relevan tetapi tidak kritis seperti re-transmission atau keep-alive packet. 4. **Info** :Informasi umum yang tidak berdampak signifikan terhadap komunikasi, seperti pesan status protokol

2.3. Sniffing Wireshark vs TCPdump

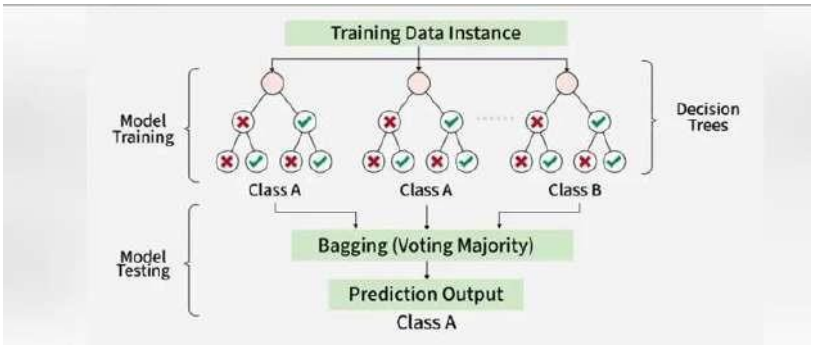
Sniffing packet adalah teknik yang digunakan untuk menangkap paket data yang melintas dalam sebuah jaringan. Paket ini mengandung informasi penting, seperti alamat sumber, alamat tujuan, protokol yang digunakan, serta payload data (Kurose, 2021). Ada dua jenis sniffing yaitu active sniffing dan passive sniffing. Active sniffing dilakukan ketika traffic IP diinterupsi dan passive sniffing dilakukan ketika user hanya mendengar traffic tanpa diinterupsi seperti yang dilakukan pada penelitian ini. Passive sniffing dilakukan dengan dua cara yaitu sniffing melalui Wireshark dan TCPdump. Perbedaan keduanya adalah Wireshark merupakan tools dengan window based dimana paket yang ditangkap dapat diidentifikasikan langsung , sementara TCPdump merupakan task scheduling dengan linux-based dimana paket yang ditangkap dilakukan analisis berdasarkan cuplikan waktu yang sama..

2.4. Machine Learning

Machine learning dapat didefinisikan sebagai aplikasi komputer dan algoritma matematika yang diadopsi dengan cara pembelajaran yang berasal dari data dan menghasilkan prediksi di masa yang akan datang. Adapun proses pembelajaran yang dimaksud adalah suatu usaha dalam memperoleh kecerdasan yang melalui dua tahap antara lain latihan (training) dan pengujian (testing). Bidang machine learning berkaitan dengan pertanyaan tentang bagaimana membangun program komputer agar meningkat secara otomatis dengan berdasar dari pengalaman (Roihan et al., 2020). Istilah Machine Learning pertama kali disebutkan oleh Arthur Samuel pada tahun 1959, pada saat itu ia menjelaskan dalam konteks menyelesaikan permainan catur dengan mesin. Secara istilah machine learningmerupakan sebuah model komputasi statistik, yang berfokus pada prediksi menggunakan komputer. Algoritma machine learning membangun model matematika dari data sampel, yang dikenal sebagai "data pelatihan atau data training", untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untukmelakukan tugas. Kemudian, secara luas algoritma machine learning dapat diklasifikasikan menjadi tiga jenis, yaitu supervised learning, unsupervised learning, dan reinforcement learning (Santoso et al., 2020). Supervised learning merupakan pembelajaran dengan data yang sudah diberi label, unsupervised learning tidak membutuhkan label, dan semi supervised merupakan gabungan keduanya

2.5. Random Forest, Hyperparameter Tuning dan Evaluasi Metriks

Random Forest (RF) merupakan algoritma berbasis *ensemble learning* yang dikembangkan oleh Breiman (2001). RF membangun banyak *decision tree* dari subset data acak, lalu menggabungkan hasil prediksi setiap pohon untuk menghasilkan keputusan akhir. Tahapan Random Forest terdiri dari pengambilan sample data, pembangunan pohon keputusan dan proses voting yang dijelaskan oleh Gambar 3 berikut



Gambar 2. Pohon Random Forest (Ariyoga,2022)

Setiap pohon akan memberikan hasil prediksi kelas. Prediksi akhir diambil berdasarkan suara terbanyak (*majority voting*) dari seluruh pohon numerik. Persamaan (1) menjelaskan rumus dari proses voting

$$\hat{y} = mo\{h_1(x), h_2(x), \ldots, h_m(x)\} \qquad (1)$$

Dimana $\hat{y}$ adalah hasil prediksi akhir berdasarkan majority voting, m adalah jumlah pohon dalam hutan dan $h_i(x)$ adalah prediksi pohon ke-i. Hyperparameter tuning (default dan setelaah tuning) yang digunakan pada penelitian ini

dijelaskan pada Gambar 3 yang terdiri dari nilai yang sudah ditetapkan *(n_estimators; max_depth; min_samples_split, min_samples_leaf, class_weight* dan *random_state*)

```
# ==============================
# SKENARIO 1: MODEL DEFAULT
# ==============================
rf_default = RandomForestClassifier(random_state=42)
rf_default.fit(X_train, y_train)

y_pred_default = rf_default.predict(X_test)
```

```
# ==============================
# SKENARIO 2: MODEL SETELAH TUNING
# ==============================
rf_tuned = RandomForestClassifier(
    n_estimators=200,
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=2,
    class_weight='balanced',
    random_state=42
)
rf_tuned.fit(X_train, y_train)

y_pred_tuned = rf_tuned.predict(X_test)
```

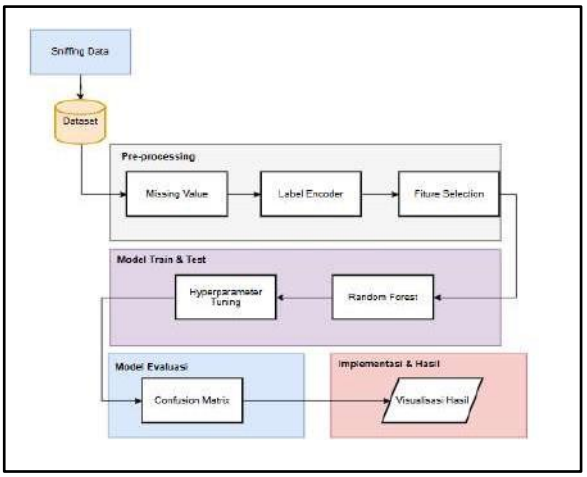
Gambar 3. Tuning model (kiri tanpa hyperparameter) dan Hyperparameter Tuning

Hasil *Random Forest* akan dievaluasi melalui nilai *accuracy* (proporsi prediksi positif yang benar dari jumlah total prediksi), *Precission* (proporsi prediksi positif yang benar dari jumlah total prediksi positif.), *Recall*(proporsi positif yang benar dari jumlah total positif yang sebenarnya) dan *F-1 Score* dengan rumus pada persamaan (2)

$$F1 - Score = 2 * Precision * Recall/(Precision + Recall) \quad (2)$$

3. METODOLOGI

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen untuk menganalisis performa algoritma *Random Forest* model *default* dan model *hyperparameter tuning* dalam mendeteksi anomali pada lalu lintas jaringan. Tahapan metodologi yang digunakan dalam penelitian ini dijelaskan pada alur penelitian pada Gambar 4



Gambar 4. Alur Penelitian

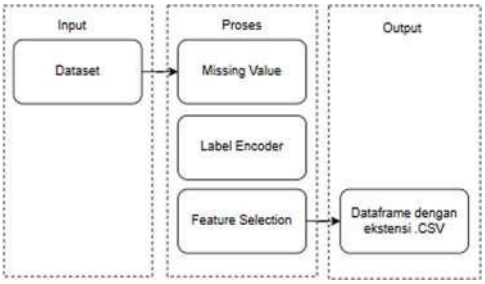
3.1 Dataset

Dataset yang digunakan dalam penelitian ini adalah *data capture* hasil *sniffing* pada server TIK (Teknologi Informasi dan Komunikasi) Institut Teknologi Nasional Bandung menggunakan *software* Wireshark dengan total 588159 paket data, yang memiliki fitur fitur data berupa No., Time, *IP source*, *IP destination*, *Protocol*, *Length, expert info, info,* dan fitur tambahan yaitu kolom label anomali yang dibuat berdasarkan nilai *expert info* pada setiap paket data. Ada empat nilai expert info yaitu *error, warning, note* dan *info*. Gambar 5 merupakan hasil monitoring *Wireshark* yang disortir untuk bagian expert info saja.

Gambar 5. Hasil Monitoring Wireshark untuk Expert Info (Kolom Kiri)

3.2. Pre-processing Data

Data mentah yang ditangkap dari *Wireshark* akan dinormalisasi melalui tahapan *pre-processing* data. Ada tiga tahapan *pre-processing* disini yaitu *Missing Value, Label Encoder* dan *Feature Selection Missing Value* yaitu proses mengisi nilai yang kosong dengan modus atau nilai yang sering muncul. *Label Encoder* adalah proses merubah data yang bernilai kategorikal menjadi nilai numerik dan *Feature Selection* adalah proses menghapus fitur-fitur yang tidak digunakan dalam pengujian. Gambar 6 merupakan contoh data pada saat diberi label encoder

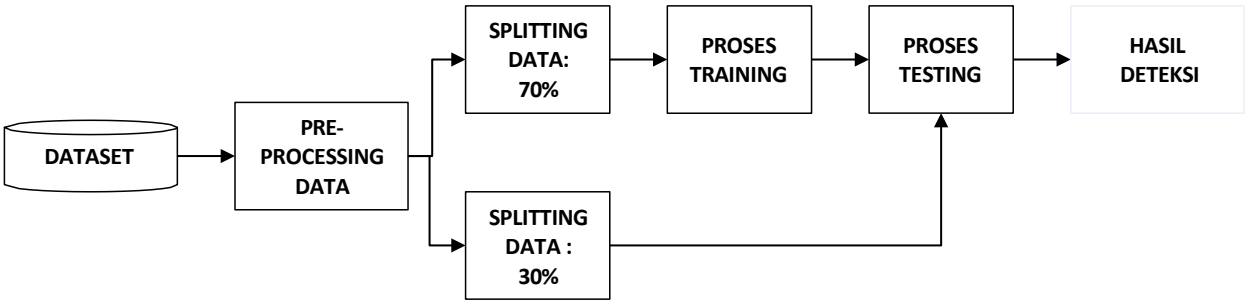


Gambar 6. Hasil Monitoring Wireshark untuk Expert Info (Kolom Kiri)

Hasil *encoder* akan diproses oleh *feature selection* sehingga sistem men-sortir fitur yang digunakan yaitu *source, destination , protocol* dan *length*

3.3. Data Splitting

Setelah dilakukan semua tahap *preprocessing*, dataset yang sudah melalui tahap *preprocessing* akan dilakukan pembagian (*Data Splitting*) dimana dari 100% total dataset dibagi menjadi 70 % data latih (*train data*) dan 30% data uji (*test data*) yang dijelaskaan oleh Gambar 7

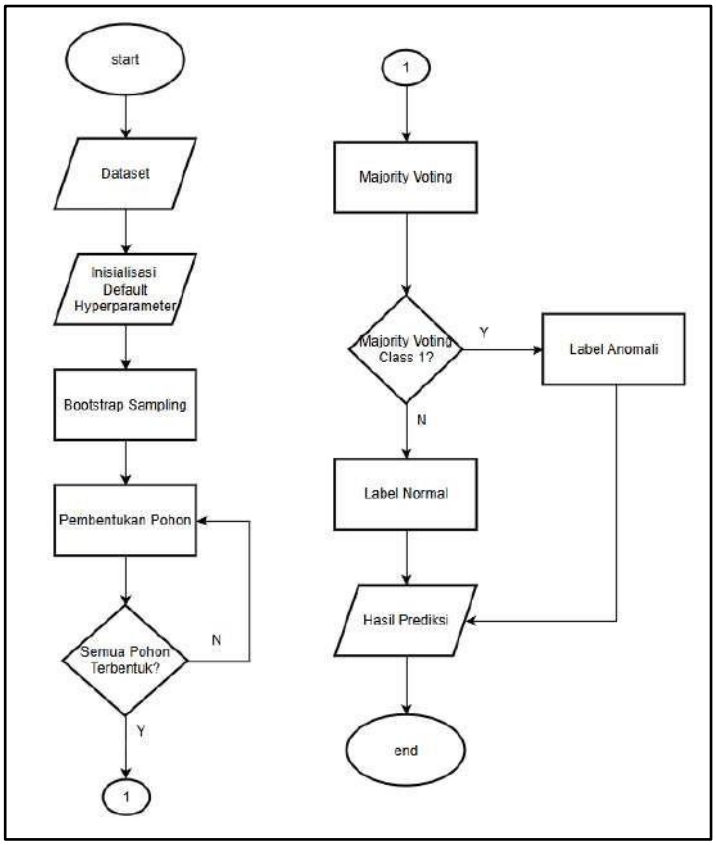


Gambar 7. Proses *Spliiting* Data

Proses *training* akan dilakukan dengan Metoda *Random Forest* dan proses testing akan diujicobakan dengan melihat parameter evaluasi metriks dengan nilai akurasi (*accuracy*) , *precission, recall, F-1 Score* sebelum dilakukan *hyperparameter tunning* dan setelah dilakukan *hyperparameter tunning*

3.4. Pelatihan dan Pengujian Model

Proses selanjutnya yaitu training dan testing model dilakukan bertahap , Pertama adalah melakukan skenario latih dan uji model. Skenario ini dilakukan pelatihan dan pengujian model untuk mendapatkan hasil yang beragam dan menganalisis performa model (*metrics evaluation*). Pada pelatihan dan pengujian model akan dilakukan pada model *default* dan *hyperparameter tuning* terhadap label anomali termasuk label *erro*r dan *warning* yang termasuk kategori anomali. Selanjutnya dilakukan pelatihan Model *Ramdom Forest*, yaitu model *default* dan model *hyperparameter tuning* dilakukan untuk melatih model dalam menentukan pembentukan pohon berdasarkan nilai *hyperparameter* yang diterapkan dari kedua model tersebut. Terakhir adalah Evaluasi yang dilakukan menggunakan metrik *Accuracy*, *Precision*, *Recall*, dan *F1-Score* untuk mengukur kinerja model dalam mendeteksi ano nantinya akan mengeluarkan hasil majority voting kelas normal dan kelas anomali.Gambar 8 menjelaskan flowchart dari algoritma Random Forest mulai dari *raw dataset*, sampai keluar keputudan deteksi normal atau anomali



Gambar 8. *Flowchart Random Forest*

Bagian ini memuat langkah-langkah peneliti dalam melakukan penelitian, disajikan secara lengkap namun padat. Kalau melakukan pendataan harus dijelaskan mulai dari metoda pengambilan sampel sampai dengan teknik analisis.

4. HASIL DAN PEMBAHASAN

Bab ini menyajikan hasil dan pembahasan dari penelitian yang telah dilakukan. Secara spesifik, bab ini menguraikan implementasi dan pengujian sistem yang dibuat, serta menganalisis performa model dalam mendeteksi anomali pada lalu lintas jaringan.

4.1 Lingkungan Pengembangan

Penelitian ini menggunakan perangkat keras berupa personal computer dengan spesifikasi:

- Processor: AMD Ryzen 3 3250U
- Memory: 8192MB RAM

Sementara itu, perangkat lunak yang digunakan meliputi:

- Sistem operasi: Windows 11

- Code editor: Visual Studio Code
- Bahasa pemrograman: Python

4.2 Dataset dan Pre-processing

Dataset yang digunakan berasal dari data lalu lintas jaringan yang diperoleh melalui *sniffing* menggunakan Wireshark. Sebelum digunakan, data melewati tahap *preprocessing* yang terdiri dari:

- **Penyandian Label (*Label Encoding*)**: Mengubah data kategorikal menjadi bentuk numerik, seperti yang ditunjukkan pada Tabel 1.

Tabel 1. Contoh Hasil Label Encoder

No	Time	Source	Destination	Protocol	Expert	Anomali	Length	Info
1	0.000	1539	484	4	2	0	60	96637
2	0.022	1539	484	4	2	0	60	96922
3	0.052	1761	484	4	2	0	60	93005
4	0.096	1539	484	4	2	0	60	97233

- **Pemilihan Fitur (*Feature Selection*)**: Menghilangkan fitur-fitur yang tidak relevan untuk mendeteksi anomali, seperti yang ditunjukkan pada Tabel 2.

Tabel 2. Hasil Feature Selection

Source	Destination	Protocol	Anomali	Length
1539	484	4	0	60
1539	484	4	0	60
1761	484	4	0	60
1539	484	4	0	60

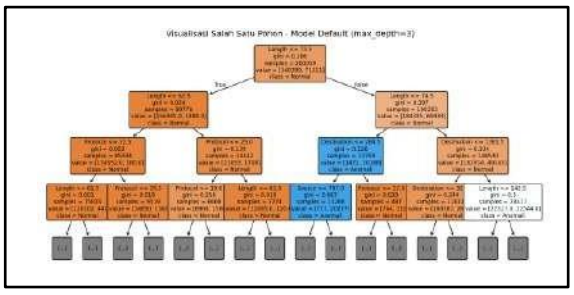
Setelah pra-pemrosesan, data dibagi menjadi data latih dan data uji dengan rasio 70:30.

4.3 Penerapan Model dan Proses Pelatihan

Penelitian ini membandingkan dua model *Random Forest*: Model *default* (menggunakan pengaturan bawaan) dan Model *tuning* (dengan mengoptimalkan *hyperparameter*). *Tuning hyperparameter* dilakukan pada n_estimators, max_depth, min_samples_split, min_samples_leaf, dan max_features untuk meningkatkan performa model.

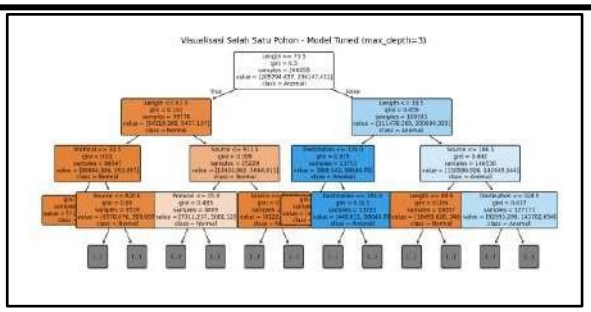
4.4 Analisis Visualisasi Pohon Keputusan

Visualisasi pohon keputusan (decision tree) dari setiap skenario menunjukkan bagaimana model membuat keputusan.(Gambar 9)



Gambar 9. Visualisasi pohon model *default*

Gambar 10 menunjukkan visualisasi pohon model *hyperparameter tuning*



Gambar 10. Visualisasi pohon model *hyperparameter tuning*

- Pada model default untuk label anomali, pohon keputusan menunjukkan Gini impurity yang rendah, menandakan distribusi kelas yang tidak seimbang (mayoritas normal).
- Sebaliknya, pada model tuning, penggunaan hyperparameter class_weight='balanced' berhasil membuat distribusi kelas lebih seimbang, yang terlihat dari nilai Gini impurity sebesar 0.5 pada root node.

4.5 Pengujian Data dan Perbandingan Hasil

Model yang telah dilatih kemudian diuji menggunakan 30% data uji. Hasil evaluasi menggunakan metrik *Confusion Matrix, Accuracy, Precision, Recall*, dan *F1-Score.*(Gambar 11)

Perbandingan Hasil Evaluasi:

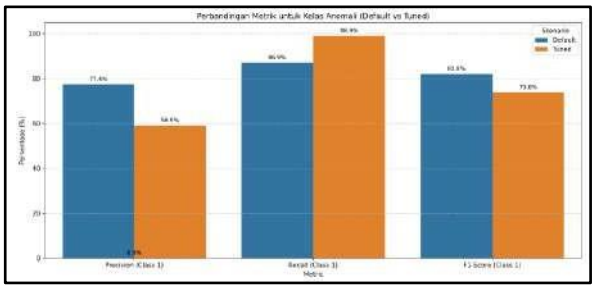
	Skenario	Accuracy	Precision	Recall	F1-Score
0	Default	0.933312	0.872736	0.907709	0.888771
1	Tuned	0.878786	0.793231	0.922350	0.829764

Perbandingan Metrik untuk Kelas Anomali (1):

	Skenario	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
0	Default	0.773689	0.868562	0.818385
1	Tuned	0.589153	0.988960	0.738411

Gambar 11. Hasil Evaluasi global dan kelas anomali

Gambar 12 menunjukkan grafik perbandingan model terhadap kelas anomali



Gambar 12. Grafik perbandingan model terhadap kelas anomali

Ringkasan Hasil Pengujian:

- Model Default: Mencapai Akurasi 93.3% dan Recall Anomali 86.9%.
- Model Tuning: Mencapai Akurasi 86.3% dan Recall Anomali 98.9%.

Hasil dan pembahasan diungkapkan dengan padat dan jelas kerangka keilmuan yang diperoleh, bukan merupakan barisan tabel data atau gambar.

5. KESIMPULAN

Perbandingan hasil menunjukkan bahwa model *default* memiliki akurasi dan presisi global yang lebih tinggi. Namun, untuk tujuan utama penelitian ini, yaitu deteksi anomali, model *tuning* lebih unggul karena memiliki nilai Recall yang jauh lebih tinggi (98.9%).

Meskipun presisi model *tuning* lebih rendah (58.9%), yang dapat meningkatkan *false positive* (alarm palsu), nilai *recall* yang signifikan menunjukkan kemampuan model yang superior dalam mengidentifikasi sebagian besar anomali yang ada. Dengan demikian, dapat disimpulkan bahwa model Random Forest dengan *tuning hyperparameter* lebih efektif untuk mendeteksi anomali lalu lintas jaringan pada penelitian ini.

Penelitian ini terbatas pada jaringan IP kampus ITENAS yang tertangkap oleh Wireshark

DAFTAR PUSTAKA (Time New Roman, 10 Bold)

Aggarwal, C. C. (2019). Outlier Analysis. *Springer Science+Business Media, New York*.

Ainurrohmah. (2021). Akurasi Algoritma Klasifikasi pada Software Rapidminer dan Weka. *Prisma, vol. 4*, 493–499.

Ariyoga, D. (2022). PERBANDINGAN METODE SELEKSI FITUR FILTER, WRAPPER, DAN EMBEDDED PADA KLASIFIKASI DATA NIRS MANGGA MENGGUNAKAN RANDOM FOREST DAN SUPPORT VECTOR MACHINE (SVM). *UII Journal*, 1-119.

Assiri, A. F., Al-Haidari, F. Y., & Al-Ghamdi, H. (2020). A Genetic Algorithm Based-Random Forest Model for Network Intrusion Detection. *Journal of King Saud University - Computer and Information Sciences, Vol. 32, No. 8*.

Chappell, L. (2012). *WIRESHARK NETWORK ANALYSIS The Official Wireshark Certified Network Analyst™ Study Guide 2™ Edition (Version 2.1c)* . Reno, AS: Protocol Analysis Institute.

F. Al-Hajri, B. A.-M.-B.-Q. (2023). Performance Evaluation of Random Forest Algorithm for Anomaly Detection in IoT Networks. *Sensors, Vol. 23, No. 1 MDPI*, 456.

F. Li, S. W. (2021). An Ensemble Learning-Based Approach for Anomaly Detection in Industrial IoT. *IEEE Access, Vol. 9*, 12345-12355.

G.Prashanth, V. P. (2020). Using Random Forests for Network-based Anomaly detection at Active routers. *IEEE-International Conference on Signal processing, Communications and Networking Madras Institute of Technology, Anna University Chennai India*, 93-96.

Ghazi Al-Naymat, M. A.-K.-H. (2018). Using machine learning methods for detecting network anomalies within SNMP-MIB dataset. *Int. J. Wireless and Mobile Computing*, 67-76.

Hodge, V. J. (2023). Survey of outlier detection methodologies. *Artif. Intell. Rev. 22,*, 85-126.

Hsiao, C., Lee, R., & Chen, C. (2017). A novel approach to detect ARP spoofing attacks using SNMP-MIB. *Proceedings of the 2017 IEEE International Conference on Applied System Innovation*.

Ige, O. A., Ogunfayo, O. A., & Ayeni, B. A. (2023). Performance Evaluation of Naïve Bayes Classifiers for Network Intrusion Detection System. *International Journal of Computer and Information Technology, Vol. 12, No. 1*.

Imam Riadi, R. U. (2019). ANALISIS PERBANDINGAN DETECTION TRAFFIC ANOMALY DENGAN METODE NAIVE BAYES DAN SUPPORT VECTOR MACHINE (SVM). *ILKOM Journal Ilmiah*, 17-24.

Inixindo. (2021, Maret 29). *Inixindo jogja*. Retrieved from Kenapa Monitoring Jaringan Sangat Penting?: https://inixindojogja.co.id/kenapa-monitoring-jaringan-sangat-penting-ini-penjelasan-lengkapnya/

Jan Michael Spoor, J. W. (2023). A Proposal for Formalization and Definition of Anomalies in Dynamical Systems. *P. Brito et al. (eds.), Classification and Data Science in the Digital Age, Studies in Classification, Data Analysis, and Knowledge Organization*, 373-381.

K. H. Purwanto, Y. a. (2019). Traffic anomaly detection in ddos flooding. *International Conference on Telecommunication Systems Services and Applications (TSSA),*.

Kurose, J. F. (2021). *Computer Networking: A Top-Down Approach.* Boston: Pearson.

M. A. Shauma, Y. P. (2020). Deteksi Anomali Trafik Menggunakan Algoritma Birch Dan Dbscan Pada Streaming Traffic. *eProceedings Eng., vol. 3, no. 3*, 5004–5012.

M. S. Uddin, M. A. (2021). A Random Forest-Based Intrusion Detection System with Feature Selection Using Gain Ratio for IoT Networks. *Sensors, Vol. 21, No. 10 MDPI*, 3345.

Palindungan Tampubolon, E. L. (2024). Identifikasi Malware Pada Wireshark. *JURNAL KAJIAN TEKNIK ELEKTRO*, 64-68.

Raghavendra Chalapathy, S. C. (2019). Deep Learning For Anomaly Detection : A Survey. *arXiv*, 1-50.

Rakhmadi Rahman, E. F. (2024). IMPLEMENTASI NETWORK TRAFFIC ANALISIS UNTUK MENDETEKSI ANOMALI JARINGAN PADA TWITTER/X DAN INSTAGRAM. *Digibe: Digital Business and Entrepreneurship Journal*, 88-96.

Smith, J. (2018). Advances in Network Monitoring. *Journal of Network and Systems Management, 26(4)*, 835-850.

Triya Agustina, M. I. (2024). Performance Analysis of Random Forest Algorithm for Network Anomaly Detection using Feature Selection. *Sinkron : Jurnal dan Penelitian Teknik Informatika*, 1116-1124.

Xiaoling Tao, Y. P. (2018). A Parallel Algorithm For Network Traffic Anomaly Detection Based on Isolation Forest. *International Journal of Distributed Sensor Networks*, 1 - 11.

Wireshark Foundation. (2025). *Expert Information*. Retrieved from Wireshark User's Guide: https://www.wireshark.org/docs/wsug_html_chunked/ChAdvExpert.html.

YA-NAN WANG, J. W. (2020). Network Traffic Anomaly Detection Algorithm Based on Intuitionistic Fuzzy Time Series Graph Mining. *IEEE Access*, 63381-63389.