Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 367~375
e-ISSN: 2548-6861

Performance Comparison of Random Forest and Decision Tree Algorithms for Anomaly Detection in Networks

Rafiq Fajar Ramadhan [1]*, Wahid Miftahul Ashari [2]**
* Teknik Komputer, Universitas Amikom Yogyakarta
rafiqfjrrmdhn@students.amikom.ac.id [1], wahidashari@amikom.ac.id [2]

Article Info

Article history:

Received 2024-09-14
Revised 2024-09-26
Accepted 2024-09-30

Keyword:

Decision Tree,
Network Security,
Random Forest,
Intrusion Detection System (IDS),
UNSW-NB15.

ABSTRACT

The increase in cyber attacks has made network security a very important focus in this digital era. This research compares the performance of two machine learning algorithms, that is Random Forest and Decision Tree for detecting anomalies in networks using the UNSW-NB15 datasets, which include various types of attacks such as DoS, Backdoor, Exploits and others which will be used to train and test both models. The data collection method, pre-processing, data splitting and modelling using SMOTE method to handle data imbalanced were applied in both algorithms and then evaluated using accuracy, precision, recall and f1-score metrics. From the study result, it can be conclude that the Decision Tree algorithm performs better in detecting anomalies in binary data with an accuracy of 99,71%. However, in multi-class data, Random Forest showed slightly better performance, though it required significantly more time for training and prediction. Despite the small difference in accuracy, Decision Tree demonstrated faster prediction times, making it more efficient for time-sensitive applications. This research concludes that while Random Forest provides higher accuracy for complex datasets, Decision Tree offers a more time-efficient solution with comparable accuracy.

I. PENDAHULUAN

Peningkatan serangan siber dan potensi kerugian yang semakin besar menjadikan keamanan jaringan komputer sebagai fokus utama dalam era digital saat ini [1]. Berdasarkan data dari BSSN pada tahun 2023, tercatat ada lebih dari 4 juta anomali trafik atau serangan siber di Indonesia, dengan jenis terbanyak yaitu *Generic Trojan RAT* [2]. Data ini menegaskan betapa pentingnya upaya pengamanan jaringan komputer di Indonesia, terutama melalui deteksi anomali yang akurat dan efektif untuk mencegah gangguan layanan, pencurian data, atau kerusakan sistem yang merugikan [3]. Metode seperti *Intrusion Detection System* (IDS) dapat digunakan untuk mengatasi permasalahan pengaman jaringan komputer. Secara umum IDS diklasifikasikan menjadi dua jenis yaitu *anomaly detection* dan *misuse detection*. *Misuse detection* bekerja dengan mencocokan pola aktivitas jaringan dengan database berisi pola-pola serangan yang sudah diketahui dan jika ditemukan kecocokan maka akan dianggap terjadi intrusi.

Sementara itu, *anomaly detection* bekerja dengan cara mengindentifikasikan aktivitas jaringan yang tidak biasa atau menyimpang dari pola aktivitas normal yang sebelumnya telah dipelajari[4]. IDS berbasis anomali digunakan secara luas karena sistem ini dapat mendeteksi serangan baru yang bahkan belum diketahui [5]. Namun metode *anomaly detection* ternyata memiliki tingkat akurasi dan presisi yang masih belum efektif dalam mendeteksi intrusi pada jaringan, karena akurasinya relatif rendah [6]. IDS menggunakan *honeypot* dengan pendekatan *Reinforcement Learning* telah dilakukan pada penelitian sebelumnya dengan hasil menunjukan bahwa metode yang diusulkan dapat mendeteksi serangan DDoS dan MiTM rata-rata 99,96% lebih cepat dan juga memiliki *precision*, *recall* dan *F1-score* yang lebih baik dibandingkan dengan IDS tradisional [7].

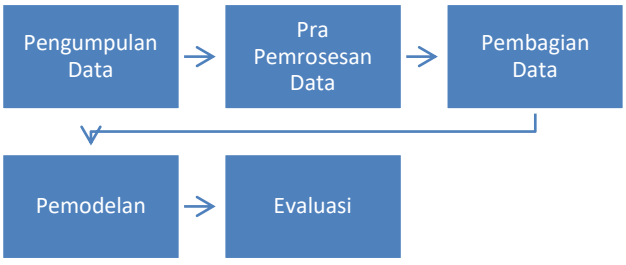
Pendekatan menggunakan teknik *machine learning* juga dapat digunakan untuk mengembangkan IDS dalam mendeteksi dan mengklasifikasikan serangan dunia maya secara efektif[8], [9], [10]. Implementasi algoritma *Decision Tree*, *Random Forest*, *Multi-layer Perceptron* dan *Stacked*

Autoencoder telah dilakukan pada penelitian dengan hasil menunjukan algoritma *Random Forest* menghasilkan akurasi yang tinggi untuk memprediksi apakah suatu jaringan sedarng diserang dan mengklasifikasikan jenis serangan secara akurat [11]. Penelitian dengan mengusulkan sebuah model deteksi intrusi yang menggabungkan *machine learning* dan *deep learning* telah dilakukan pada. Hasil dari penelitian menggunakan *dataset* NSL-KDD dan CIC-IDS2017 ini menunjukan bahwa metode yang diusulkan menunjukan akurasi yang tinggi, mencapai 85,24% pada *dataset* NSL-KDD dan 99,91% pada *dataset* CIC-IDS2017 dan juga memiliki *True Positive Rate* yang lebih baik daripada model deep learning tradisional, tertuama dalam mengidentifikasi kejadian serangan yang jarang terjadi[12]. Meskipun teknik *machine learning* dapat digunakan untuk mengembangkan IDS dalam mendeteksi dan mengklasifikaskan serangan dunia maya, pengembangan IDS menghadapi tantangan signifikan karena serangan siber terus berevolusi dan terjadi dalam skala besar, sehingga menuntuk solusi yang mampu beradaptasi dan dapat ditingkatkan [13].

Tujuan penelitian ini adalah membandingkan dua algoritma yaitu *Random Forest* dan *Decision Tree* dalam mendeteksi anomali pada jaringan dengan menggunakan *dataset* UNSW-NB15 [14], [15] yang menyediakan data yang lebih relevan dan kompleks dari segi lalu lintas jaringan modern saat ini dibandingkan dengan penggunaan *dataset* pada penelitian sebelumnya.

II. METODE PENELITIAN

Penelitian dimulai dengan pengumpulan data yaitu *dataset* jaringan seperti KDD Cup, NSL-KDD, CIC-IDS2017 dan UNSW-NB15 dan dilakukan perbandingan terhadap *dataset* tersebut dengan tujuan mengetahui *dataset* mana yang paling sesuai untuk mengevaluasi kinerja kedua algoritma. Selanjutnya, data yang sudah didapat melalui tahap *pre-processing* untuk dilakukan pembersihan dan normalisasi data, kemudian data dibagi menjadi set pelatihan dan set pengujian. Berikut adalah tahapan yang akan dilakukan dalam penelitian pada gambar 1.



Gambar 1. Alur Penelitian

Selanjutnya akan dilakukan pemodelan dari kedua algoritma yaitu *Random Forest* dan *Decision Tree* untuk dilakukan pelatihan dan pengujian. Pada tahap akhir, model akan dievaluasi dengan metrik akurasi, *presisi*, *recall* dan *F1-score* untuk mengetahui algoritma mana yang lebih baik dalam melakukan deteksi anomali.

A. *Pengumpulan Data*

Tahap awal penelitian ini adalah mengumpulkan data yang dapat digunakan untuk mendeteksi anomali pada jaringan seperti log jaringan, data lalu lintas jaringan dan data keamanan jaringan. Penelitian ini menggunakan dataset UNSW-NB15 yang terdiri dari 257.673 record paket jaringan mentah yang sangat relevan dengan penelitian yang akan dilakukan karena memiliki tingkat keragaman serangan yang tinggi, mencakup berbagai jenis serangan yang kompleks dan tersedia untuk diunduh di pranala https://research.unsw.edu.au/projects/unsw-nb15-dataset.

B. *Pra-pemrosesan Data*

Setelah menemukan dataset yang relevan langkah selanjutnya adalah pra-pemrosesan data yaitu langkah yang harus dilakukan untuk mempersiapkan data mentah menjadi data yang siap digunakan untuk di analisis dan proses pemodelan. Pembersihan data dari data yang tidak relevan, data duplikat, data yang hilang, transformasi data, modifikasi format data, dan pemilihan fitur yang relevan untuk deteksi anomali diterapkan pada proses ini dengan tujuan untuk memastikan data bersih dan berkualitas sebelum memulai analisis atau proses pemodelan.

C. *Pembagian Data*

Data yang telah diproses sebelumnya kemudian dibagi ke dalam dua subset, yakni data latih (*training set*) dan data uji (*testing set*). Model machine learning dilatih menggunakan data pelatihan dan performanya pada data yang belum diuji sebelumnya diuji menggunakan data pengujian.

D. *Pemodelan*

Pada tahap ini algoritma *Random Forest* dan *Decision Tree* diterapkan sebagai bagian dari tahap pemodelan menggunakan bahasa pemograman python dan library sklearn.

TABEL I.
ISTILAH CONFUSION MATRIX

	Prediksi Positif	**Prediksi Negatif**
Sebenarnya Positif	TP (True Positive)	FN (False Negative)
Sebenarnya Negatif	FP (False Positive)	TN (True Negative)

Kedua algoritma ini dipilih karena dapat diandalkan dalam menganalisis data jaringan yang kompleks dan non-linear serta dapat mendeteksi anomali pada berbagai jenis data. *Decision Tree* memudahkan interpretasi, *Random Forest* meningkatkan akurasi dan mengurangi *overfitting*. *Hypermeter tuning* juga diterapkan pada tahapan ini sebagai cara untuk mengoptimalkan performa model. Selanjutnya kedua algoritma dievaluasi dengan menggunakan *Confusion*

Matrix sebagai salah satu metode atau representasi visual untuk mengevaluasi performa model klasifikasi yang dijelaskan pada Tabel 1.

True Positive (TP) dan *True Negative* (TN) menunjukan prediksi yang tepat, sementara *False Positive* (FP) dan *False Negative* (FN) menunjukan prediksi yang salah. TP adalah data positif yang berhasil dikenali dengan benar, sedangkan TN adalah data negatif yang juga berhasil dikenali dengan benar. FP adalah data negatif yang keliru dianggap positif, dan FN adalah data positif yang keliru dianggap negatif. Nilai TP, TN, FP dan FN ini yang akan digunakan untuk menghitung kinerja model dengan menggunakan metrik akurasi, presisi, *recall* dan *F1-score*. Berikut ini adalah penjelasan perhitungan yang digunakan pada penelitian pada Tabel 2.

TABEL II
PENJELASAN INDIKASI DAN RUMUS

Indikasi	Penjelasan	Rumus
Akurasi	Menggambarkan seberapa banyak prediksi yang benar, baik itu prediksi positif maupun negatif.TP (True Positive)	Akurasi=TP+TN/TP+TN+FP+FN
Presisi	Dari semua yang diprediksi positif, berapa banyak yang benar-benar positif.FP (False Positive)	Presisi=TP/TP+FP
Recall	Dari semua yang sebenarnya positif, berapa banyak yang berhasil diprediksi positif oleh model.	Recall=TP/TP+FN
F1-score	*Harmonic means* dari presisi dan recall	F1-score=2*Presisi*Recall/Presisi+Recall

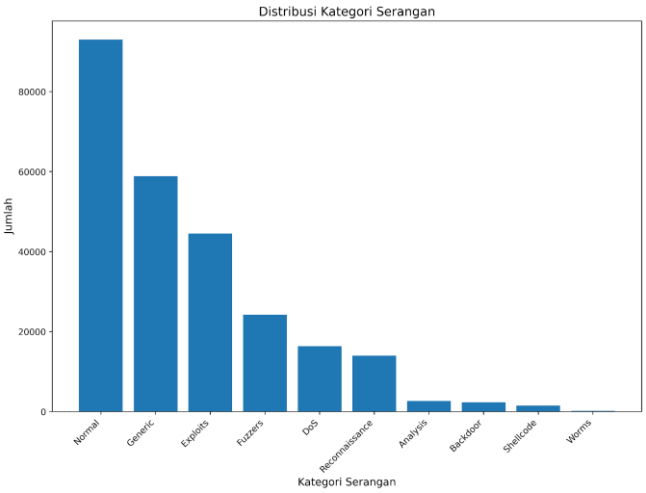
E. Evaluasi

Tahap akhir penelitian adalah mengevaluasi kinerja kedua model. Metrik evaluasi yang sesuai untuk deteksi anomali seperti akurasi, presisi, *recall*, dan *F1-score* diterapkan untuk mengukur kinerja model. Hasil evaluasi kedua model kemudian dibandingkan untuk melihat algoritma mana yang lebih baik dalam mendeteksi anomali pada jaringan dalam bentuk gambar.

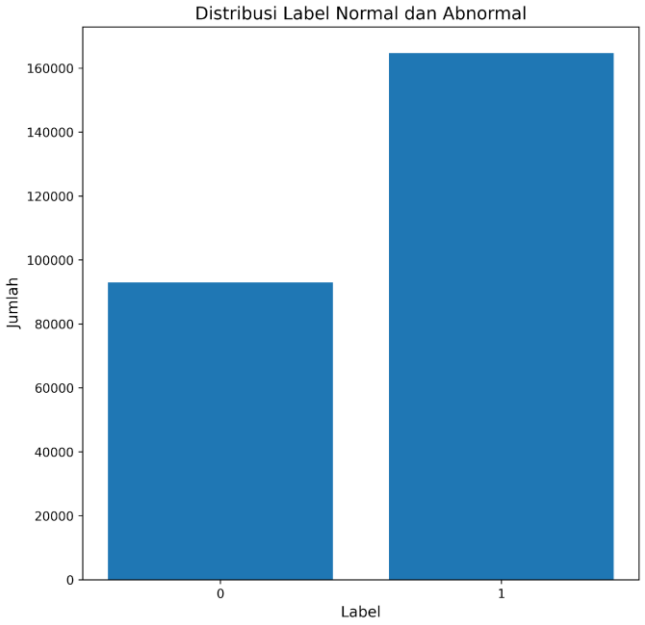
III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Tahap awal penelitian ini adalah mengumpulkan data yang dapat digunakan untuk mendeteksi anomali pada jaringan seperti log jaringan, data lalu lintas jaringan dan data keamanan jaringan. Penelitian ini menggunakan dataset UNSW-NB15 yang terdiri dari 257673 record paket jaringan mentah dengan jumlah label (164673 label 1 dan 93000 label 0) yang sangat relevan dengan penelitian yang akan dilakukan karena memiliki tingkat keragaman serangan yang tinggi, mencakup berbagai jenis serangan yang kompleks. Distribusi pada dataset dapat dilihat pada Gambar 2 dan 3.



Gambar 2. Distribusi Serangan Pada Dataset



Gambar 3. Distribusi Normal dan Abnormal Pada Dataset

B. Pra-pemrosesan Data

Proses pra-pemrosesan data (data *pre-processing*) dilakukan secara bertahap untuk memastikan data yang akan digunakan dalam pemodelan sudah bersih dan berkualitas. Berikut adalah tahap-tahap yang dilakukan dalam pra-pemrosesan data.

1) Data Cleaning. Pada *dataset* UNSW-NB15 tidak terdapat nilai yang kosong atau *null values* tetapi pada fitur *Service* terdapat nilai yang tidak diketahui atau (-) sehingga dilakukan penghapusan baris yang terdapat nilai (-) pada fitur *Service*. Setelah dilakukan pembersihan data atau data

cleaning jumlah baris pada dataset adalah 116,352 baris yang tadinya berjumlah 257,673 baris.

2) One-hot Encoding. Pada proses ini data kategorikal dikonversi kedalam format numerikal tujuannya adalah supaya algoritma *machine learning* dapat melakukan prediksi yang lebih baik. Pada dataset yang digunakan terdapat tiga fitur kategorikal yaitu '*proto*', '*service*', dan '*status*', terdapat 2 kategori pada fitur '*proto*', 12 kategori pada fitur '*service*' dan 6 kategori pada fitur '*status*'. Fitur ini diubah menggunakan fungsi *pd.get_dummies*() menjadi 20 fitur baru dan dilakukan penghapusan pada ketiga fitur kategorikal ('*proto*', '*service*' dan '*status*') yang sudah menjadi numerikal. Fitur baru ini lalu digabungkan dengan 45 fitur menjadi total 62 fitur.

3) Normalisasi Data. Pada proses ini fitur numerik pada dataset dilakukan normalisasi data yang tujuannya adalah untuk mengubah nilai pada fitur menjadi berada di rentang 0 sampai 1. Pada dataset dengan 62 fitur terdapat 39 fitur dengan jenis numerik, fitur-fitur ini yang akan dilakukan normalisasi data dengan menggunakan metode *MixMaxScaler*.

4) Binary Label Encoding. Pada dataset yang sudah dinormalisasi pada tahap sebelumnya, dilakukan perubahan nilai pada fitur label yang berisi nilai numerik (1 dan 0) menjadi nilai kategorikal (normal dan abnormal) jika nilai pada label adalah 0 maka diubah menjadi normal dan jika nilai nya 1 (*DoS , Analysis, Reconnaissance, Fuzzers, Worms Exploits, Generic, Backdoor, Normal*) maka diubah menjadi abnormal. Setelah dilakukan perubahan format dari numerik ke kategorikal, data diubah lagi ke format kategorikal dan disimpan pada dataframe bin_data. Tujuan dari pelabelan biner ini adalah untuk membuat *dataset* dengan fokus hanya pada prediksi normal dan abnormal sehingga model melakukan prediksi pada serangan secara umum tidak berdasarkan kategori serangan.

5) Multi-class Label Encoding. Pada proses ini fitur attack_cat pada dataset dilakukan metode one-hot encoding, tujuannya adalah membuat fitur baru berdasarkan nilai yang ada pada fitur attact_cat (Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Normal, Reconnaissance, Worms). Jadi jumlah fitur pada dataset yang tadinya berjumlah 62 fitur, bertambah 9 menjadi 71 dan dilakukan penghapusan pada fitur attack_cat menjadi 70 fitur. Data ini disimpan pada dataframe multi_data. Tujuan dari pelabelan multi-class ini adalah untuk membuat model melakukan prediksi pada dataset yang lebih spesifik dengan manambahkan 9 kategori serangan ke dalam fitur seperti yang sudah dijelaskan.

6) Binary dan Multi-class Label Feature Selection. Setelah dilakukan label encoding pada dataset, selanjutnya dilakukan pemilihan fitur yang relevan pada dataset dengan tujuan mengurangi dimensi data dari fitur yang kurang relevan dan juga meningkatkan akurasi model dalam mendeteksi anomali. Pemilihan fitur ini dilakukan dengan menggunakan metode *Pearson Correlation Coefficient* dengan memilih nilai korelasi lebih dari 0.3. Pada data biner terdapat 13 dari total 62 fitur yang akan digunakan pada tahapan pemodelan yaitu rate, sttl, sload, dload, ct_srv_src, ct_state_ttl, ct_dst_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm, ct_src_ltm, ct_srv_dst dan label. Pada data multi-class terdapat 7 dari 62 fitur yang akan digunakan di tahapan pemodelan yaitu dttl, swin, dwin, tcprtt, synack, ackdat, label.

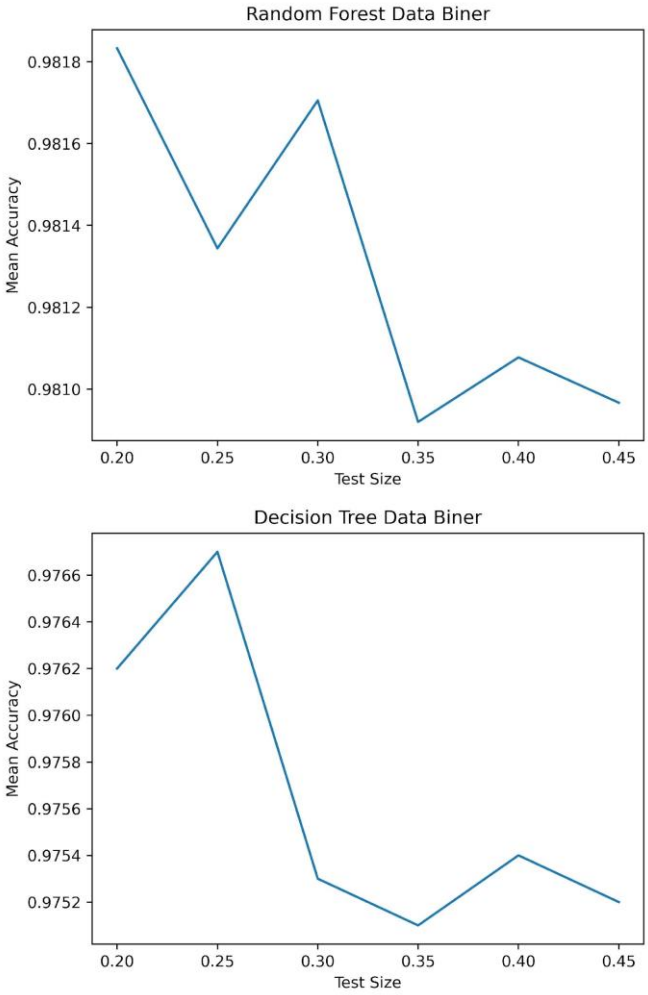
7) Hyperparameter Tuning. Pada proses ini hyperparameter tuning dilakukan dengan menggunakan metode random search. Masing-masing dataset (data biner dan data multi-class) dibagi menjadi dua bagian yaitu training dan testing. Selanjutnya kedua algoritma di terapkan pada masing-masing dataset tersebut. Hasil dari hyperparameter tuning dapat dilihat pada Tabel 3.

TABEL III
HYPERPARAMETER KEDUA MODEL

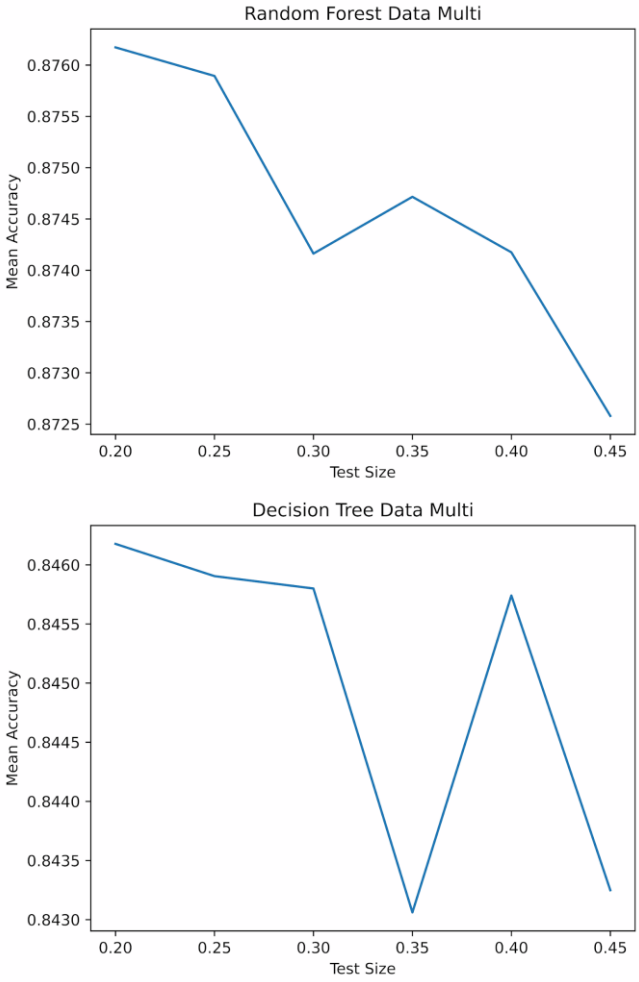
Algoritma	Dataset	Best Parameter	Accuracy
Random Forest	Biner	{'n_estimators': 300, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': 30}	98.05%
Decision Tree	Biner	{'random_state': 42, 'min_samples_split': 5, 'min_samples_leaf': 2, 'max_depth': 20, 'class_weight': None}	97.80%
Random Forest	Multi-class	{'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 4, 'max_depth': None}	88.79%
Decision Tree	Multi-class	{'random_state': 42, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 20, 'class_weight': 'balanced'}	88.86%

C. Pembagian Data

Pada proses ini pembagian data ditentukan menggunakan metode *K-Fold Cross Validation* dengan cara membagi data menjadi lima bagian (n_splits=5). Teknik ini membuat data uji yang digunakan untuk evaluasi model berbeda di setiap iterasi, model dilatih sebanyak lima kali, masing-masing menggunakan kombinasi yang berbeda dari empat *fold* sebagai data latih dan satu *fold* sebagai data uji. Hasil *K-Fold Cross Validation* kedua model pada data biner pada Gambar 3.



Gambar 4. Hasil Kedua Model Pada Data Biner



Gambar 5. Hasil Kedua Model Pada Multi

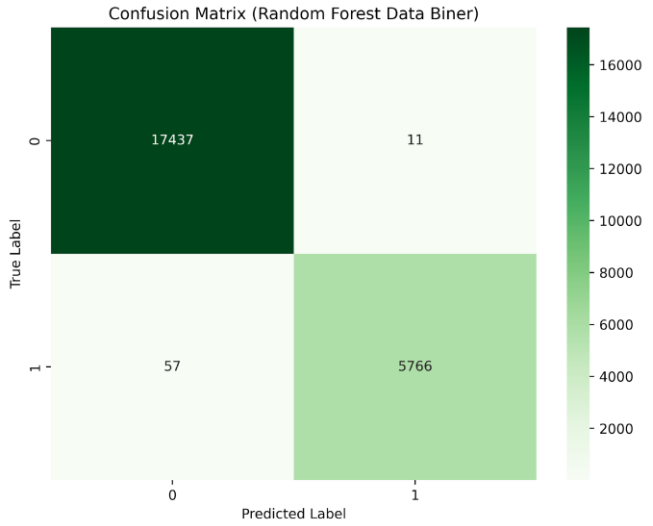
Selanjutnya hasil dari *K-Fold Cross-Validation* kedua model pada data *multi-class* pada Gambar 4. Nilai dari *test_size* yang memiliki akurasi tertinggi dalam proses ini akan digunakan sebagai parameter utama dalam tahap pemodelan untuk kedua algoritma. Pemilihan *test_size* ini penting agar model dapat diuji secara optimal dan menghasilkan performa terbaik.

D. Pemodelan

Proses selanjutnya adalah melakukan pemodelan algoritma *Decision Tree* dan *Random Forest* pada data yang sudah bersih dan berkualitas setalah dilakukan tahapan *pre-processing* dengan menggunakan library python (*DecisionTreeClassifier*) dan (*RandomForestClassifier*) untuk melatih model dengan data pelatihan, menguji model dengan data pengujian, melakukan prediksi pada dataset, menghitung akurasi prediksi dengan membandingkan hasil prediksi dengan label dan yang terakhir adalah menghasilkan akurasi model. Pada proses ini metode SMOTE diterapkan karena *dataset* tidak seimbang (*imbalanced*) yang bisa dilihat pada Gambar 2 dan 3. *Confusion Matrix* digunakan untuk memberikan visualisasi tentang kinerja model klasifikasi dengan menunjukan berapa banyak prediksi model adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Proses ini dibagi menjadi klasifikasi data biner (bin_data) dan klasifikasi data *multi-class* (multi_data) yang sudah dijelaskan pada proses sebelumnya.

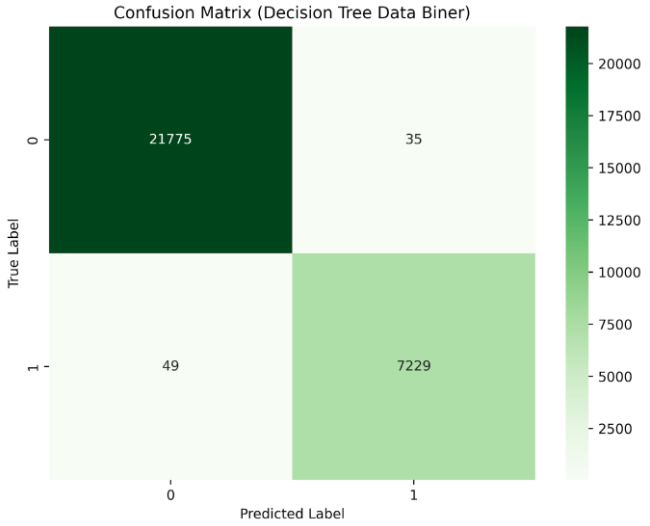
1) Klasifikasi Data Biner

Sebelum dilakukan pemodelan pada kedua algoritma, fitur label yang nantinya akan diprediksi oleh model dihapus menggunakan fungsi drop. Variable X menyimpan 13 fitur yang ada pada bin_data yang akan digunakan untuk melatih model dan variable Y menyimpan nilai dari fitur label (1 atau 0) atau target yang ingin diprediksi menggunakan fitur-fitur yang ada di variable X. Parameter dan nilai *test_size* yang sudah diketahui pada proses sebelumnya digunakan pada proses ini. Kelas 0 mewaliki kondisi normal sementara kelas 1 mewakili kondisi abnormal atau terdapat anomali. Berikut adalah hasil *Confusion Matrix* dari pemodelan *Random Forest* pada data biner pada Gambar 6.



Gambar 6. Confusion Matrix Random Forest Pada Data Biner

Pada Gambar 6 kelas 0 mewakili kondisi normal, sementara kelas 1 mewakili kondisi abnormal atau anomali. Model *Random Forest* memunjukan bahwa sebanyak 17436 sampel diprediksi normal ketika data memang normal (*True Negative*). Sebanyak 11 sampel data diprediksi abnormal ketika data sebenarnya adalah normal (*False Positive*). Sebanyak 57 sampel data gagal di prediksi sebagai abnormal, dimana sampel diprediksi sebagai normal tetapi sebenarnya menunjukan adanya anomali (*False Negative*). Sementara itu, 5766 sampel data berhasil dideteksi terdapat anomali ketika data memang menunjukan adanya anomali (*True Positive*). Berikut adalah hasil *Confusion Matrix* dari pemodelan *Decision Tree* pada data biner pada Gambar 7.

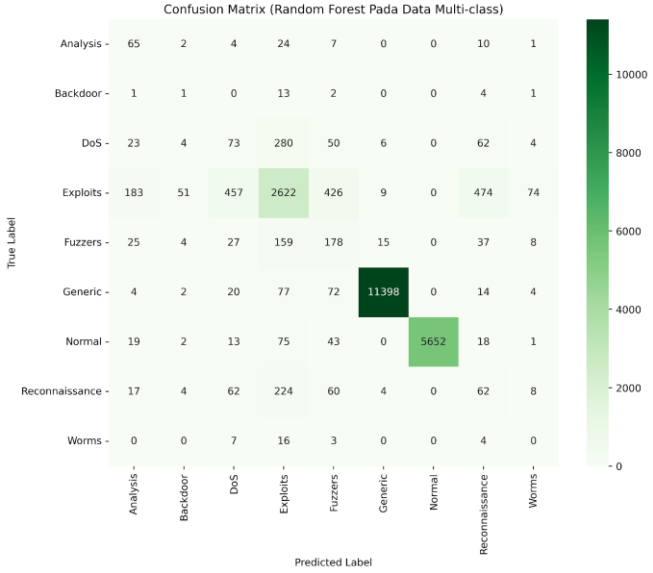


Gambar 7. Confusion Matrix Decision Tree Pada Data Biner

Pada model *Decision Tree* sebanyak 21775 sampel data diprediksi normal ketika data memang normal (*True Negative*) dan 35 sampel data diprediksi abnormal ketika data sebenarnya adalah normal (*False Positive*). Sebanyak 49 sampel data gagal di prediksi sebagai abnormal, dimana sampel diprediksi sebagai normal pada sampel yang sebenarnya abnormal (*False Negative*). Sementara itu 7229 sampel data berhasil dideteksi terdapat anomali ketika data memang menunjukan adanya anomali (*True Positive*).

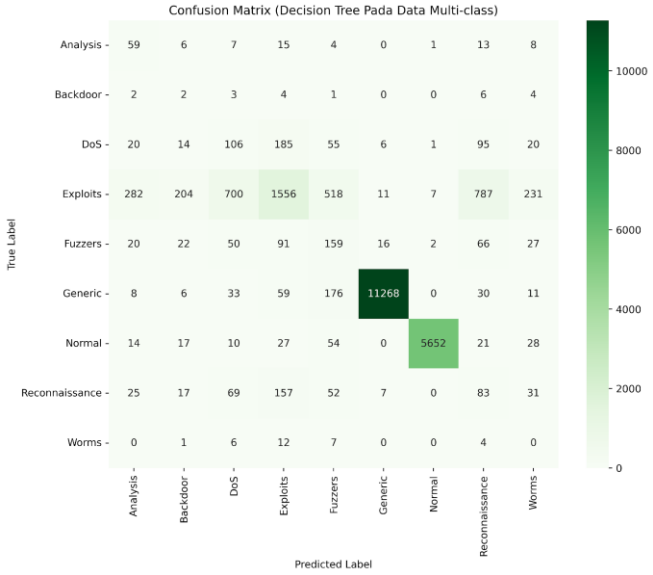
2) Klasifikasi Data Multi-class

Pada klasifikasi menggunakan data *multi-class*, model *Decision Tree* dan *Random Forest* akan mendeteksi berbagai jenis serangan. Proses yang sama dengan klasifikasi data biner diterapkan pada klasifikasi data *multi-class* sebelum dilakukan pemodelan pada Algoritma *Decision Tree* dan *Random Forest*. Model *Random Forest* pada klasifikasi data *multi-class* memiliki performa paling baik dengan 11398 prediksi benar dari total 11591 sampel pada kelas serangan *Generic*. Model *Random Forest* juga memiliki performa yang baik pada kelas *Normal* dengan 5652 prediksi benar dari total 5823 sampel. Pada kelas seperti *Exploits* model memiliki banyak kesalahan prediksi, dimana dari total 6296 sampel data 2622 yang berhasil di prediksi dengan benar sebagai *Exploits* dan kesulitan model dalam memprediksi kelas lainnya dengan benar dapat dilihat pada Gambar 8.



Gambar 8. Confusion Matrix Random Forest Pada Data Multi-class

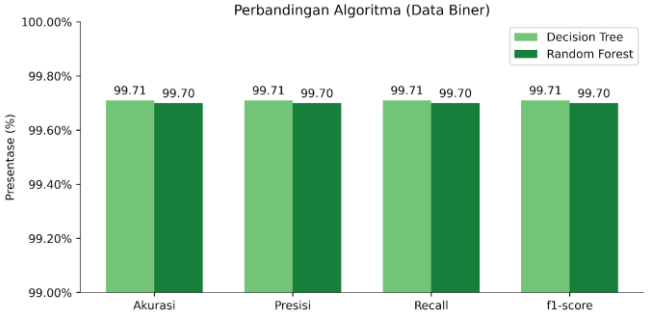
Pada model *Decision Tree* dapat dilihat pada Gambar 9. Model berhasil memprediksi kelas *Generic* dengan benar sebanyak 11268 dari total 11591 sampel data. Pada kelas normal model berhasil memprediksi kelas secara akurat sebanyak 5652 dari total 5823 sampel data dan 1556 prediksi benar pada kelas *Exploits* dengan total data sampel sebanyak 6296 dan menunjukan performa yang kurang baik dalam memprediksi kelas lainnya.



Gambar 9. Confusion Matrix Decision Tree Pada Data Multi-class

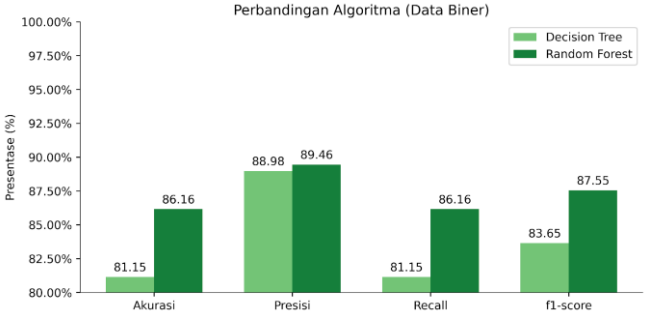
E. Evaluasi

Berikutnya, evaluasi kedua model akan dilakukan dengan memanfaatkan metrik akurasi, presisi, *recall* dan *F1-score*. Metrik perhitungan ini digunakan dengan tujuan untuk memahami seberapa baik model bekerja dalam memprediksi setiap kelas secara individual baik pada klasifikasi data biner maupun pada data *multi-class*. Perhitungan metrik dihitung menggunakan rumus yang sudah dijelaskan pada Tabel 2. Pada klasifikasi data biner model *Random Forest*, akurasi yang didapat sebesar 99.70%, presisi 99.70%, *recall* 99.70% dan *F1-score* 99.70%. Pada model *Decision Tree*, akurasi yang didapat sebesar 99.71%, presisi 99.71%, recall 99.71% dan F1-score 99.71%. Perbandingan hasil kedua algoritma pada data biner ditunjukan dalam Gambar 10.



Gambar 10. Perbandingan Performa Algoritma Pada Data Biner

Pada klasifikasi data *multi-class* model *Random Forest* mendapat angka akurasi sebesar 86.16%, presisi 89.46%, *recall* 86.16% dan *F1-score* sebesar 87.55%. Sementara model *Decision Tree* mendapat nilai akurasi sebesar 81.15%, presisi 88.98%, *recall* 81.15% dan *F1-score* sebesar 83.65%. Perbandingan hasil performa kedua algoritma terlihat pada Gambar 11.



Gambar 11. Perbandingan Performa Algoritma Pada Data Multi

Selanjutnya dilakukan perbandingan waktu yang dibutuhkan oleh kedua model algoritma untuk melakukan pelatihan dan prediksi menggunakan satuan detik pada Tabel 4.

TABEL IV
PERBANDINGAN WAKTU KEDUA MODEL

Algoritma	Dataset	Waktu Pelatihan	Waktu Prediksi	Total Waktu Pemodelan
Random Forest	Data Biner	27.5125	0.3268	28.0840
Decision Tree	Data Biner	0.3861	0.0050	0.6145
Random Forest	Data Multi-class	147.4151	0.4157	147.8308
Decision Tree	Data Multi-class	5.7600	0.0060	5.7660

Pada *dataset* data biner dan data *multi-class* kedua model menunjukan perbedaan yang signifikan dalam hal efisiensi waktu pelatihan dan prediksi. Pada *dataset* data biner. Model *Random Forest* membutuhkan waktu pelatihan selama 27.5125 detik dan waktu prediksi sebesar 0.3268 detik, dengan waktu pemodelan mencapai 28.0840 detik. Sebaliknya, model *Decision Tree* menunjukan waktu yang jauh lebih efisien pada *dataset* yang sama, yaitu 0.3861 detik untuk pelathina dan 0.0050 detik untuk prediksi, dengan total waktu pemodelan 0.6145 detik. Perbedaan waktu kedua model semakin terlihat perbedaannya pada dataset multi-class, model *Random Forest* membutuhkan waktu pelatihan yang lebih lama, yaitu 147.4151 detik dan waktu prediksi sebesar 0.4157 detik, sehingga total pemodelan mencapai 147.8308 detik. Sementara itu, model *Decision Tree* pada dataset multi-class menunjukan waktu yang lebih cepat dengan waktu pelatihan hanya 5.7600 detik dan waktu prediksi 0.0060 detik, menghasilkan total waktu pemodelan 5.7660 detik.

TABEL V

PERBANDINGAN AKURASI KEDUA ALGORITMA PADA TIAP KELAS PADA DATASET MULTI-CLASS

Kelas	Random Forest	Decision Tree
Analysis	57.52%	52.21%
Backdoor	04.55%	09.09%
DoS	14.54%	21.12%
Exploits	61.03%	36.22%
Fuzzers	39.29%	35.10%
Generic	98.33%	97.21%
Normal	97.06%	97.06%
Reconnaissance	14.06%	18.82%
Worms	0.0%	0.0%

Pada kelas "*Generic*" dan "*Normal*", kedua model menunjukan performa akurasi yang baik, dengan *Random Forest* unggul sedikit pada kelas "*Generic*" (98.33%) dibandingkan *Decision Tree* (87.21%), sementara akurasi kelas "*Normal*" sama-sama tinggi di 97.06%. Pada kelas "*Exploits*", model *Random Forest* lebih unggul (61.03%) dibandingkan *Decision Tree* (36.22%) dan kedua model memiliki hasil akurasi yang tidak jauh berbeda pada kelas "*Fuzzers*" (39.29% *Random Forest* dan 35.10% *Decision Tree*). Sementara itu pada kelas "*DoS*" dan "*Reconnaissance*", model *Decision Tree* lebih unggul dengan nilai akurasi sebesar 21.12% dan 18.82% sedangkan *Random Forest* hanya mencapai 14.54% dan 14.06%. Untuk kelas "*Analysis*", model *Random Forest* sedikit lebih baik (57.52%) dibandingkan *Decision Tree* (52.21%), tetapi pada kelas "*Backdoor*", model *Decision Tree* unggul (9.09%) dibandingkan model *Random Forest* (4.55%) dan kedua model gagal memprediksi kelas "*Worms*" dengan akurasi 0%.

Model deteksi anomali pada jaringan, khususnya yang menggunakan algoritma *machine learning* seperti *Random Forest* dan *Decision Tree* memiliki potensi besar dalam meningkatkan keamanan jaringa. Dengan kemampuannya untuk mengidentifikasi pola lalu lintas yang menyimpang dari normal, model ini dapat mendeteksi serangan siber yang belum diketahui (*novel attacks*) secara real-time. Penerapannya dalam sistem keamanan jaringan memungkinkan deteksi dini ancaman, sehingga tindakan mitigasi dapat dilakukan dengan lebih cepat. Namun, implementasi model ini juga dihadapkan pada beberapa tantangan. Salah satunya adalah kebutuhan akan infrastruktur yang memadai untuk mengumpulkan, menyimpan, dan memproses data jaringan dalam jumlah yang besar. Selain itu, diperlukan mekanisme pemantauan yang kuat untuk memastikan model tetap berfungsi dengan baik dan dapat beradaptasi dengan perubahan pola serangan.

IV. KESIMPULAN

Berdasarkan data yang diperoleh dari penelitian yang dilakukan dengan cara membandingkan dua algoritma, yaitu *Random Forest* dan *Decision Tree* dalam mendeteksi anomali pada jaringan menggunakan *dataset* UNSW-NB15, hasil evaluasi menggunakan metrik akurasi, presisi, *recall* dan *F1-score* dapat disimpulkan bahwa algoritma *Decision Tree* memiliki performa yang lebih baik dalam mendeteksi anomali pada data biner dengan akurasi mencapai 99.71%. Hal ini menunjukan keunggulan algoritma tersebut dalam menangani data yang kompleks dan tidak seimbang. Pada data *multi-class*, kedua algoritma menunjukan performa yang sebanding, meskipun algoritma *Random Forest* lebih unggul dalam mendeteksi jenis serangan seperti *Exploits*, sementara algoritma *Decision Tree* lebih baik dalam memprediksi serangan *DoS* dan *Reconnaissance*.

Selain itu, dari segi efisiensi waktu, algoritma *Decision tree* menunjukan hasil lebih cepat dalam proses pelatihan dan prediksi, terutama pada dataset *multi-class*. Namun, algoritma *Random Forest* menawarkan akurasi yang lebih tinggi, meskipun membutuhkan waktu yang lebih lama untuk proses pemodelan. Berdasarkan hasil yang diperoleh dari proses penelitian, penelitian ini merekomendasikan penggunaan algoritma *Random Forest* untuk sistem deteksi anomali pada jaringan, terutama jika akurasi menjadi prioritas utama, sementara algoritma *Decision Tree* dapat dipilih jika efisiensi waktu lebih diutamakan.

DAFTAR PUSTAKA

[1] T. Vimy *et al.*, "Ancaman Serangan Siber Pada Keamanan Nasional Indonesia," *Jurnal Kewarganegaraan*, vol. 6, no. 1, 2022.

[2] Badan Siber dan Sandi Nasional, "Lanskap Keamanan Siber Indonesia," 2023. Accessed: Jul. 13, 2024. [Online]. Available: https://www.bssn.go.id/wp-content/uploads/2024/03/Lanskap-Keamanan-Siber-Indonesia-2023.pdf

[3] G. Saputra and S. Informasi, "Pengamanan Jaringan Komputer: Tantangan Dan Strategi Terkini," 2023.

[4] S. M. Lestari, "Penerapan Snort Intrusion Detection System Menggunakan Machine Learning," 2019.

[5] T. Widodo and A. Sekti Aji, "Pemanfaatan Network Forensic Investigation Framework untuk Mengidentifikasi Serangan Jaringan Melalui Intrusion Detection System (IDS)," 2022.

[6] R. Budiarto, Y. Dwi Kuntjoro, and P. Korespondensi, "Analisis Perilaku Entitas Untuk Pendeteksian Serangan Internal Menggunakan Kombinasi Model Prediksi Memori dan Metode PCA," vol. 10, no. 6, pp. 1223–1232, 2023, doi: 10.25126/jtiik.2023107123.

[7] A. Pashaei, M. E. Akbari, M. Zolfy Lighvan, and A. Charmin, "Early Intrusion Detection System using honeypot for industrial control networks," Results in Engineering, vol. 16, Dec. 2022, doi: 10.1016/j.rineng.2022.100576.

[8] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems," IEEE Access, vol. 8, pp. 73127–73141, 2020, doi: 10.1109/ACCESS.2020.2988359.

[9] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a deep learning model for intrusion detection on apache spark platform," IEEE Access, vol. 8, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[10] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," IEEE Access, vol. 9, pp. 123448–123464, 2021, doi: 10.1109/ACCESS.2021.3109081.

[11] A. Devarakonda, N. Sharma, P. Saha, and S. Ramya, "Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets," in Journal of Physics: Conference Series, IOP Publishing Ltd, Jan. 2022. doi: 10.1088/1742-6596/2161/1/012043.

[12] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," IEEE Access, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[13] W. Wijayanto, N. H. Sardini, and G. N. Elsitra, "Menciptakan Ruang Siber yang Kondusif bagi Pegiat Anti-Korupsi," Integritas : Jurnal Antikorupsi, vol. 7, no. 1, pp. 179–196, Jun. 2021, doi: 10.32697/integritas.v7i1.732.

[14] J. Slay and N. Moustafa, "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Article in Information Security Journal A Global Perspective, 2016, doi: 10.1080/19393555.2015.1125974.

[15] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Dec. 2015. doi: 10.1109/MilCIS.2015.7348942.

[16] D. E. Kurniawan, H. Arif, N. Nelmiawati, A. H. Tohari, and M. Fani, "Implementation and analysis ipsec-vpn on cisco asa firewall using gns3 network simulator," in Journal of Physics: Conference Series, IOP Publishing, 2019, p. 012031.