DETEKSI SERANGAN SIBER PADA JARINGAN KOMPUTER MENGGUNAKAN METODE RANDOM FOREST

Laode Ikhwanul Uzlah, Rizal Adi Saputra, Isnawaty
Teknik Informatika, Universitas Halu Oleo
Jln. H.E.A Mokodompit No. 8 Kampus Baru UHO Bumi Tridharma Anduonohu
rizaladisaputra@uho.ac.id

ABSTRAK

Serangan siber merupakan ancaman yang signifikan bagi keamanan jaringan komputer. Oleh karena itu, penelitian ini bertujuan untuk memfokuskan pada pengembangan model deteksi serangan siber menggunakan metode *Random Forest* berdasarkan dataset serangan siber yang relevan. Dengan mengintegrasikan teknologi *machine learning* dan analisis dataset yang cermat, penelitian ini memberikan kontribusi signifikan untuk meningkatkan keamanan siber. Evaluasi model menunjukkan akurasi sebesar 66.13%, dengan tantangan terutama dalam mengenali serangan (kelas 1). Meskipun demikian, model berhasil memprediksi dengan baik pada data baru, menunjukkan potensi untuk deteksi proaktif. Hasil deteksi dibuktikan dengan distribusi probabilitas model terhadap kelas yang diprediksi. Keseluruhan, penelitian ini membuktikan efektivitas *Random Forest* dalam mendeteksi serangan siber, memberikan landasan untuk peningkatan lebih lanjut.

Kata kunci : *analisis dataset, deteksi proaktif, keamanan siber, Random Forest, serangan siber*

1. PENDAHULUAN

Serangan siber merupakan ancaman serius bagi organisasi dan instansi yang mengandalkan jaringan komputer dalam operasionalnya [1]. Fenomena meningkatnya ancaman serangan siber terjadi di seluruh dunia, seiring dengan pertumbuhan jumlah pengguna internet. Ancaman ini dapat merusak penyedia layanan seperti situs web, email, dan cloud melalui retas sistem atau pencurian data pengguna layanan, yang berpotensi merugikan pihak penyedia layanan dan pengguna [2].

Keberadaan internet, selain mempermudah aktivitas sehari-hari [3], juga membawa konsekuensi meningkatnya risiko keamanan siber. Pertimbangan keamanan ini semakin penting bagi penyedia layanan, seiring dengan meningkatnya pengguna internet yang dapat menjadi sasaran potensial ancaman siber [2].

Serangan siber, dilakukan oleh jaringan komputer atau telekomunikasi, mencakup target seperti website, sistem komputer, dan komputer pribadi. Kemajuan teknologi informasi dan internet memudahkan para pelaku serangan untuk beroperasi dengan lebih mudah, hemat biaya, dan efisien. Insiden serangan siber melibatkan spionase industri dan target pemerintah penting, yang dapat menimbulkan kecemasan dan ketidakamanan akibat risiko kehilangan data pribadi dan kekayaan. Serangan siber bukan hanya menjadi alat politik di dunia siber, tetapi juga dapat digunakan dalam konteks ekonomi. Berbagai jenis serangan cyber umumnya termasuk malware [4] yang mencari kelemahan perangkat lunak dan dapat menginfeksi perangkat dengan virus, worm, atau trojan horse [5]. Serangan DDoS (Distributed Denial of Service) [6] dapat melumpuhkan server dengan membanjiri lalu lintas jaringan [7] bertujuan untuk mengganggu ketersediaan layanan [1].

Selain itu, serangan phishing [3], taktik penipuan dengan mengelabui target untuk mencuri informasi sensitif, menjadi salah satu ancaman utama [8]. Penjahat dunia maya menggunakan web phishing untuk mengeksploitasi kerentanan browser, menyebarkan malware melalui URL jahat [9], dengan tujuan mendapatkan akses ke jaringan, mencuri informasi, dan diam-diam memantau sistem komputer target [1].

Oleh karena itu, diperlukan sistem deteksi serangan siber yang efektif dengan pengumpulan dataset yang relevan [10]. Dalam upaya mendeteksi ancaman siber secara dini, pendekatan yang diambil dalam penelitian ini adalah membangun sebuah model deteksi serangan siber berbasis metode Random Forest. Dengan menggabungkan kekuatan teknologi machine learning dan pengumpulan dataset yang relevan, penelitian ini berkontribusi pada upaya peningkatan keamanan siber, terutama dalam mendeteksi serangan siber secara proaktif.

2. TINJAUAN PUSTAKA

Tinjauan pustaka yang dijelaskan peneliti dalam penelitian ini menggunakan beberapa referensi seperti buku, monografi, dan jurnal. Berikut tinjauan pustaka yang mendukung teoritis dari penelitian ini.

2.1. Serangan Siber

Serangan siber merupakan ancaman yang serius terhadap keamanan informasi dan infrastruktur teknologi [1]. Seiring dengan kemajuan teknologi informasi dan ketergantungan pada internet, serangan siber telah menjadi semakin canggih dan merugikan. Serangan ini dapat mencakup berbagai bentuk, termasuk *malware*, serangan DDoS, *phishing*, dan banyak lagi [2].

a. *Malware*
   Malware adalah jenis serangan siber yang mencakup berbagai program berbahaya, seperti *virus*, *worm*, dan *trojan horse* [4]. *Malware* dapat

merusak atau mencuri data, merusak fungsionalitas sistem, dan bahkan dapat memanipulasi operasi perangkat lunak [5].

b. Serangan DdoS
Serangan DDoS (*Distributed Denial of Service*) bertujuan untuk melumpuhkan server atau jaringan dengan cara membanjiri lalu lintas internet, menyebabkan penurunan ketersediaan layanan [6]. Serangan ini dapat menyebabkan *crash* sistem dan mengakibatkan ketidakmampuan sistem merespons permintaan pengguna yang sah [7].

c. *Phising*
Phishing adalah taktik penipuan yang melibatkan usaha untuk mendapatkan informasi sensitif dari korban [3] dengan menyamar sebagai entitas tepercaya. Umumnya dilakukan melalui *email* atau situs *web* palsu [8], *phishing* bertujuan untuk memancing korban agar memberikan informasi pribadi, seperti kata sandi atau data kartu kredit [9].

2.2. Deteksi Serangan Siber

Deteksi serangan siber menjadi aspek kritis dalam menjaga keamanan sistem informasi. Pendekatan yang dapat digunakan antara lain:

a. *Random Forest* dalam Deteksi Serangan
Metode *Random Forest*, sebagai algoritma pembelajaran mesin, dengan kemampuannya mengatasi kompleksitas data dan kemampuan untuk mengidentifikasi pola yang kompleks, *Random Forest* dapat menjadi pilihan efektif untuk deteksi serangan.

b. Penggunaan *Dataset* dalam Deteksi
Pengumpulan dataset yang relevan dan representatif [10] menjadi langkah kritis dalam membangun model deteksi serangan yang handal. Dataset yang baik memungkinkan pelatihan model dengan berbagai pola serangan, meningkatkan kemampuan model untuk mengenali ancaman yang beragam.

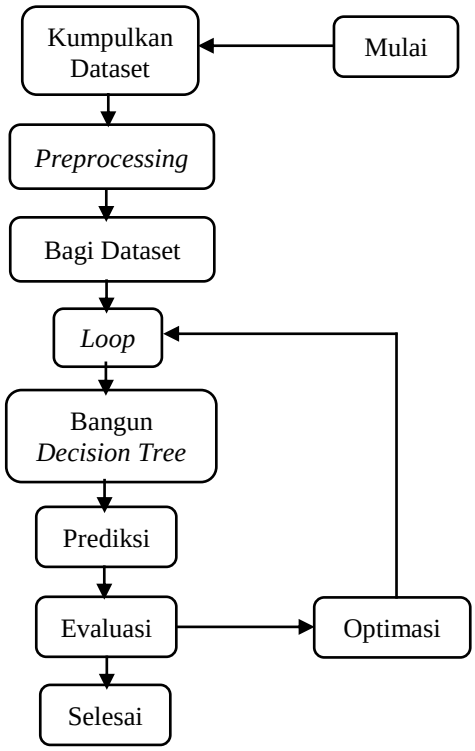
c. Teknik Analisis Data dan *Resampling*
Penggunaan teknik analisis data, seperti *resampling* data per bulan, dapat membantu dalam memahami tren serangan siber dan distribusi serangan pada rentang waktu tertentu. Ini memungkinkan analisis yang lebih mendalam untuk meningkatkan pemahaman tentang karakteristik serangan.

2.3. Metode *Random Forest*

Random Forest merupakan salah satu metode *ensemble learning* yang menggabungkan beberapa *decision tree* [1]. Metode ini dikembangkan oleh Breiman (2001) dan bekerja dengan membangun sejumlah *decision tree* dari subset data latih yang dipilih secara acak. Kemudian hasil prediksi dari semua *decision tree* dikombinasikan dengan *voting* untuk mendapatkan prediksi akhir (Liaw dan Wiener, 2002).

Kelebihan dari *Random Forest* antara lain mampu menangani data dengan banyak fitur, dan tidak mudah *overfitting* [2]. *Random Forest* juga telah terbukti efektif dalam mendeteksi *anomaly* pada jaringan komputer (Ahmed dkk., 2016). Kemampuan *Random Forest* dalam klasifikasi data tidak seimbang juga menjadi keunggulannya untuk kasus deteksi serangan siber (Moskovitch dkk., 2008).

Berdasarkan keunggulan tersebut, penelitian ini akan menerapkan *Random Forest* untuk membangun model deteksi serangan siber.



Gambar 1. Tahapan penelitian

Tahapan penelitian seperti pada gambar 1 memiliki 8 tahapan, yaitu:

a. Mengumpulkan dataset yang relevan untuk pelatihan model.
b. Melakukan pemrosesan data awal (*preprocessing*) seperti pembersihan data, penanganan *missing value*, dan transformasi fitur jika diperlukan.
c. Membagi dataset menjadi data latih dan data uji.
d. Membangun banyak *decision tree* (pohon keputusan) secara terpisah dengan menggunakan subset acak dari fitur dan sampel data latih.
e. Mengombinasikan prediksi dari semua *decision tree* menggunakan metode *voting* mayoritas untuk regresi atau klasifikasi.
f. Mengevaluasi performa model *Random Forest* pada data uji menggunakan metrik evaluasi yang sesuai.
g. Jika performa model belum memenuhi kriteria, akan dilakukan optimasi *hyperparameter* atau teknik lain untuk meningkatkan performa.

h. Setelah performa model memenuhi kriteria, model *Random Forest* siap digunakan untuk memprediksi data baru.

3. METODE PENELITIAN

Pada bagian ini akan dijabarkan mengenai data yang digunakan pada penelitian serta tahapan atau proses penelitian deteksi serangan siber seperti berikut:

3.1. Dataset

Dataset yang digunakan pada penelitian ini adalah data serangan siber (*cybersecurity_attacks*) ekstensi .csv yang didapat dari data *public* dengan url https://bit.ly/*CyberSecurityAttacks*. Dataset yang digunakan dalam penelitian ini menyajikan informasi seputar serangan siber pada jaringan komputer, serta memiliki 25 fitur dan sekitar 40 ribu baris data serangan yang terjadi dalam rentang waktu beberapa tahun. Beberapa fitur yang terdapat dalam dataset melibatkan:

a. *Timestamp*: Waktu terjadinya serangan.
b. *Anomaly Scores*: Skor anomali yang mencerminkan tingkat keanehan suatu kejadian.
c. *Source Port*: *Port* sumber yang terlibat dalam serangan.
d. *Packet Length*: Panjang paket data yang terlibat.
e. *Attack Type*: Jenis serangan yang terdeteksi.
f. *Action Taken*: Tindakan yang diambil sebagai respons terhadap serangan.

Tujuan penggunaan dataset ini adalah untuk melatih model *Random Forest* guna memahami dan mendeteksi pola-pola serangan siber.

3.2. *Preprocessing* Data

Proses *preprocessing* data dilakukan untuk memastikan kebersihan dan konsistensi data sebelum dilibatkan dalam pelatihan model *Random Forest*. Tahapan *preprocessing* melibatkan:

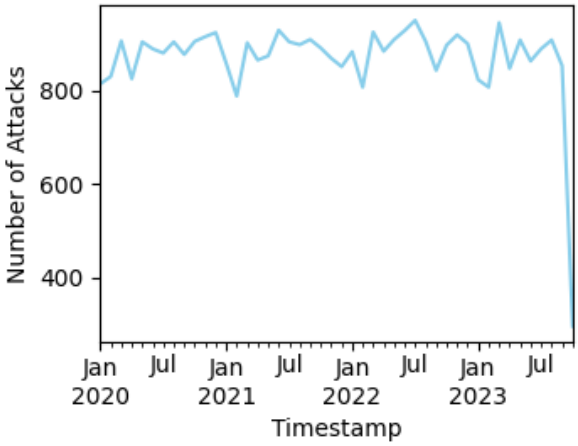
a. Penghapusan Kolom: Beberapa kolom seperti '*Malware Indicators*', '*Alerts/Warnings*', '*Proxy Information*', '*Firewall Logs*', dan 'IDS/IPS *Alerts*' dihilangkan karena kolom-kolom tersebut memiliki sejumlah besar nilai yang hilang dan dianggap kurang relevan untuk tujuan deteksi serangan.
b. Konversi *Timestamp*: Kolom '*Timestamp*' dikonversi ke objek *datetime*. Hal ini dilakukan agar informasi waktu dapat diinterpretasikan dengan benar dalam analisis.

3.3. Visualisasi Data

Visualisasi data dilakukan untuk memberikan gambaran mengenai karakteristik dataset dan distribusi serangan siber.

3.3.1. Grafik Serangan per Bulan

Grafik serangan per bulan digunakan untuk menunjukkan tren jumlah serangan siber selama periode waktu tertentu. Dengan me*resample* data per bulan, *plot* disajikan dengan tampilan berupa grafik linier yang menunjukkan tren keseluruhan serangan setiap bulan.

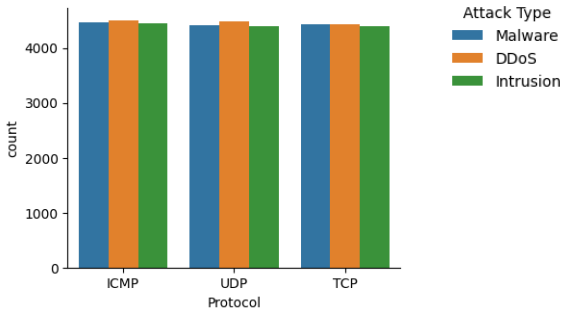


Gambar 2. Jumlah serangan siber dalam periode waktu tertentu

Dari grafik pada gambar 2, terlihat bahwa jumlah serangan mengalami fluktuasi dalam empat tahun terakhir, yaitu tahun 2020, 2021, 2022, dan 2023. Pada tahun 2020, jumlah serangan mencapai puncaknya pada bulan Juli, yaitu sekitar 800 serangan. Jumlah serangan kemudian menurun pada bulan Januari tahun 2021, tetapi kembali meningkat pada bulan Juli. Selanjutnya, jumlah serangan terus meningkat hingga mencapai sekitar 1.000 serangan pada bulan Juli tahun 2022 hingga Januari tahun 2023, dan kembali menurun pada paruh kedua tahun 2023.

3.3.2. Diagram Batang untuk Fitur Kategorikal

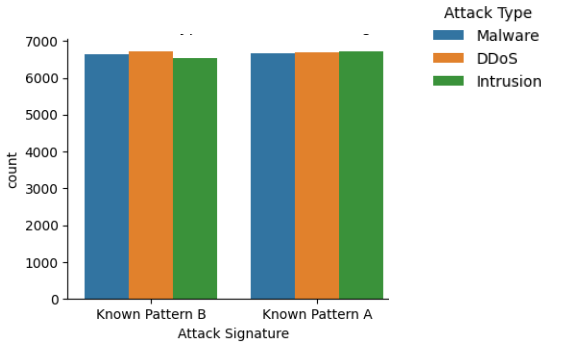
Untuk mendapatkan pemahaman lebih lanjut tentang jenis serangan dan hubungannya dengan fitur kategorikal, dilakukan visualisasi menggunakan diagram batang untuk setiap kolom seperti '*Protocol*', '*Attack Signature*', '*Action Taken*', dan '*Network Segment*'. Setiap diagram batang memberikan informasi tentang jumlah serangan untuk masing-masing kategori dalam fitur terkait, dengan warna berbeda membedakan jenis serangan



Gambar 3. Jumlah jenis serangan untuk setiap protocol

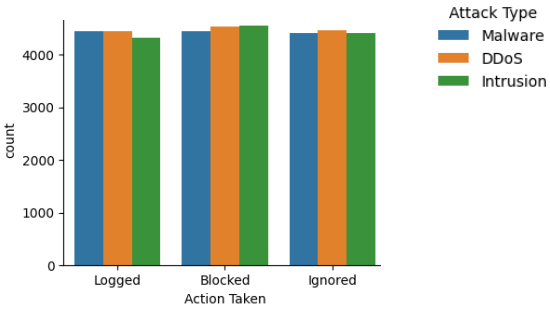
Dari diagram batang pada gambar 3, menunjukkan jumlah serangan berdasarkan jenisnya dan protokol yang digunakan. Berdasarkan

protokolnya, protokol TCP adalah protokol yang paling rentan terhadap serangan, dengan jenis serangannya yang paling umum mendominasi adalah serangan *Malware*.



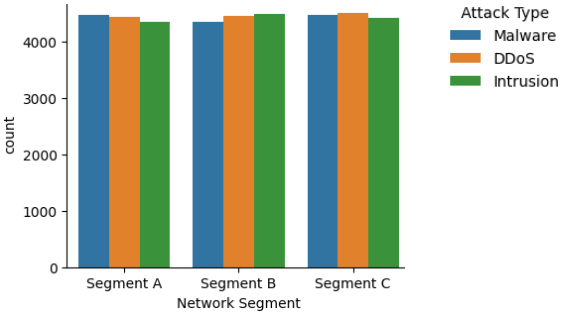
Gambar 4. Jumlah jenis serangan untuk setiap tanda serangan

Dari diagram batang pada gambar 4, menunjukkan jumlah jenis serangan untuk setiap tanda serangan. Semakin banyak jenis serangan yang dimiliki oleh suatu tanda serangan, semakin besar kemungkinan tanda serangan tersebut dapat mengidentifikasi dan mencegah serangan tersebut. Dengan demikian, diagram batang tersebut menunjukkan bahwa serangan *Malware* dan DDoS merupakan jenis serangan yang paling perlu diwaspadai oleh sistem keamanan.



Gambar 5. Jumlah jenis serangan untuk setiap tindakan yang diambil

Dari diagram batang pada gambar 5, menunjukkan jumlah serangan siber berdasarkan jenis serangan dan tindakan yang diambil. Tindakan yang diambil terhadap serangan siber tersebut adalah dicatat (*Logged*), diblokir (*Blocked*), atau diabaikan (*Ignored*). Berdasarkan data tersebut, dapat disimpulkan bahwa serangan siber jenis *Malware* adalah jenis serangan siber yang paling banyak terjadi dan paling sering dicatat. Sementara itu, serangan siber jenis DDoS dan *Intrusion* adalah jenis serangan siber yang paling sering diblokir.

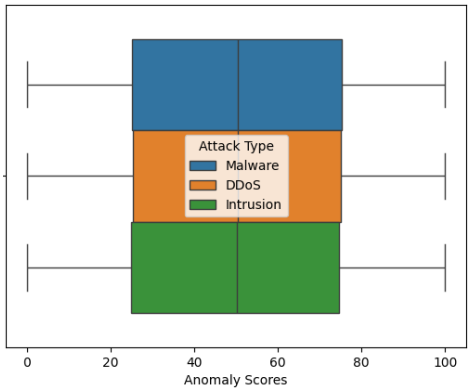


Gambar 6. Jumlah jenis serangan untuk setiap segmen jaringan

Dari diagram batang pada gambar 6, menunjukkan jumlah serangan yang terjadi di setiap segmen jaringan. Diagram ini menunjukkan bahwa segmen A mengalami serangan *Malware* terbanyak, diikuti oleh segmen B dan segmen C. Serangan DDoS terjadi paling banyak di segmen C, diikuti oleh segmen A dan segmen B. Serangan *Intrusion* terjadi paling banyak di segmen B, diikuti oleh segmen A dan segmen C.

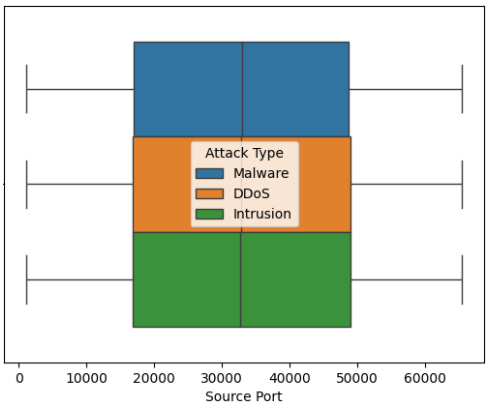
3.3.3. *Box Plot* untuk Fitur Numerik

Fitur numerik seperti '*Anomaly Scores*', '*Source Port*', dan '*Packet Length*' divisualisasikan menggunakan *box plot*. Setiap *box plot* memberikan informasi tentang distribusi nilai fitur numerik dan keberadaan *outlier* untuk masing-masing jenis serangan, dengan warna pada *box plot* membedakan jenis serangan yang berbeda.



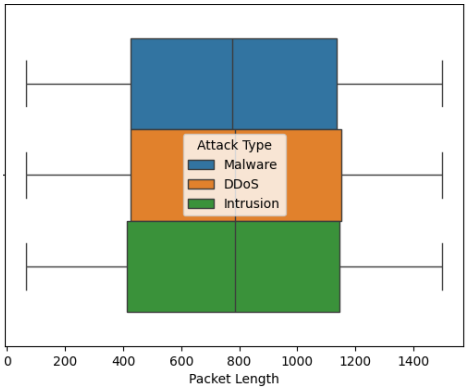
Gambar 7. Jumlah jenis serangan untuk setiap skor anomali

Box plot pada gambar 7 menunjukkan jumlah serangan berdasarkan jenis serangan. Secara umum, dapat disimpulkan bahwa jumlah serangan *Malware*, DDoS, dan *Intrusion* memiliki pola yang sama. Sebanyak 50% dari jumlah serangan ketiga jenis serangan tersebut berada pada rentang 20 hingga 60, yang berarti bahwa sebagian besar serangan memiliki tingkat keparahan sedang .



Gambar 8. Jumlah jenis serangan untuk setiap *port* sumber

Box plot pada gambar 8 menunjukkan jumlah serangan berdasarkan jenis serangan dan *port* sumber. Berdasarkan *box plot* tersebut, dapat disimpulkan bahwa, serangan *Malware* yang berasal dari *port* sumber 0-30000 lebih banyak daripada serangan *Malware* yang berasal dari *port* sumber 30000-40000. Serangan DDoS yang berasal dari *port* sumber 0-30000 lebih sedikit daripada serangan DDoS yang berasal dari *port* sumber 30000-40000. Serangan *Intrusion* yang berasal dari *port* sumber 0-30000 lebih banyak daripada serangan *Intrusion* yang berasal dari *port* sumber 30000-40000.



Gambar 9. Jumlah jenis serangan untuk setiap panjang paket

Box plot pada gambar 9 menunjukkan distribusi panjang paket untuk tiga jenis serangan. Distribusi panjang paket yang miring ke kanan menunjukkan bahwa ada lebih banyak serangan dengan panjang paket yang lebih pendek daripada serangan dengan panjang paket yang lebih panjang. Ini dapat disebabkan oleh beberapa faktor, seperti jenis *malware* yang digunakan, tujuan serangan, atau lingkungan jaringan tempat serangan terjadi.

3.4. Pelatihan Model

Proses pelatihan model *Random Forest* melibatkan:

3.4.1. Pemilihan Fitur dan Target

Fitur yang dipilih sebagai variabel independen melibatkan '*Anomaly Scores*', '*Source Port*', dan '*Packet Length*', sementara variabel dependen adalah '*Action Taken*'.

3.4.2. Pembagian Data

Setelah pemilihan fitur, data dibagi menjadi *set* pelatihan (*training set*) dan *set* pengujian (*testing set*) dengan menggunakan fungsi *train_test_split* yang disediakan oleh pustaka *scikit-learn* (*sklearn*). Rasio pembagian antara *set* pelatihan dan *set* pengujian ditentukan sebesar 70:30, yang berarti 70% dari data akan digunakan untuk melatih model, sementara 30% akan digunakan untuk menguji kinerja model yang telah dilatih.

3.4.3. Parameter Model

Selanjutnya, parameter model *Random Forest* diatur dengan inisialisasi n_*estimators*=200, *max_depth*=10, dan *random_state*=42. Inisialisasi ini memberikan jumlah pohon keputusan sebanyak 200, batasan kedalaman setiap pohon hingga 10, dan penetapan nilai *seed* untuk *random state* agar hasil dapat direproduksi.

3.5. Evaluasi Model

Evaluasi kinerja model *Random Forest* dilakukan dengan menggunakan metrik:

3.5.1. Akurasi

Akurasi digunakan untuk menunjukkan seberapa baik model dapat memprediksi secara keseluruhan. Untuk menghitung nilai akurasi dapat menggunakan rumus persamaan berikut:

$$\text{Akurasi} = \frac{\text{Jumlah prediksi benar}}{\text{Total jumlah prediksi}} \quad (1)$$

Rumus ini memberikan persentase seberapa baik model dapat memprediksi dengan benar, dan nilai akurasi berkisar antara 0 (tidak akurat) hingga 1 (sangat akurat). Semakin tinggi nilai akurasi, semakin baik modelnya dalam melakukan prediksi yang benar.

3.5.2. *Classification Report*

Classification Report memberikan *insight* lebih lanjut mengenai *precision*, *recall*, dan f1-*score* untuk setiap kelas (aksi yang diambil).

a. *Precision*

Untuk menghitung nilai *precision* dapat menggunakan rumus persamaan berikut:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Keterangan :

True Positives (TP): Jumlah observasi positif yang benar-benar diklasifikasikan sebagai positif.

False Positives (FP): Jumlah observasi negatif yang keliru diklasifikasikan sebagai positif.
Precision mengukur seberapa banyak dari prediksi positif yang sebenarnya benar. Semakin tinggi nilai *precision*, semakin baik model dalam menghindari membuat prediksi positif palsu.

b. *Recall*
Untuk menghitung nilai *recall* dapat menggunakan rumus persamaan berikut:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Keterangan :
True Positives (TP): Jumlah observasi positif yang benar-benar diklasifikasikan sebagai positif.
False Negatives (FN): Jumlah observasi positif yang keliru diklasifikasikan sebagai negatif.
Recall mengukur seberapa banyak dari keseluruhan kelas yang berhasil diidentifikasi oleh model. Semakin tinggi nilai *recall*, semakin baik model dalam menemukan semua *instance* dari kelas yang benar.

c. F1-*Score*
Untuk menghitung nilai f1-*score* dapat menggunakan rumus persamaan berikut:

$$\text{F1 } Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

F1-*score* merupakan rata-rata harmonik antara *precision* dan *recall*. Hal ini memberikan gambaran keseluruhan tentang seberapa baik model dapat mengkombinasikan *precision* dan *recall*.

3.5.3. *Confusion Matrix*

Confusion Matrix menyajikan jumlah prediksi yang benar dan salah untuk masing-masing kelas, memberikan gambaran lebih detail tentang performa model. Untuk menentukan *confusion matrix* dapat menggunakan rumus berikut:

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \quad (5)$$

Keterangan :
True Positives (TP): Jumlah observasi positif yang benar-benar diklasifikasikan sebagai positif.
True Negatives (TN): Jumlah observasi negatif yang benar-benar diklasifikasikan sebagai negatif.
False Positives (FP): Jumlah observasi negatif yang keliru diklasifikasikan sebagai positif.
False Negatives (FN): Jumlah observasi positif yang keliru diklasifikasikan sebagai negatif.

3.5.4. Validasi Silang

Validasi silang menggunakan *StratifiedKFold* dengan 5 lipatan digunakan untuk mendapatkan estimasi akurasi yang lebih reliabel, memastikan bahwa model mampu menggeneralisasi dengan baik ke data yang tidak terlihat selama pelatihan. Untuk menghitung nilai validasi silang dapat menggunakan rumus persamaan berikut:

$$CV\ Score = \frac{1}{K} \sum_{i=1}^{K} Score_i \quad (6)$$

Keterangan:
K : jumlah lipatan dalam validasi silang
$Score_i$: nilai performa model pada lipatan ke-i.

Score ini dapat berupa akurasi, presisi, *recall*, atau metrik evaluasi lainnya, tergantung pada jenis masalah dan kebutuhan analisis.

4. HASIL DAN PEMBAHASAN

4.1. Evaluasi Model

Hasil evaluasi model *Random Forest* menunjukkan kinerja yang memuaskan pada data pengujian dan validasi silang.

Tabel 1. Hasil evaluasi model

Metric	***Precision***	***Recall***	**F1 *Score***	***Support***
Class '0'	0.66	1.00	0.80	7941
Class '1'	0.37	0.00	0.00	4059
Accuracy	0.6613			12000
Macro Avg	0.52	0.50	0.40	12000
Weighted Avg	0.56	0.66	0.53	12000
Conf. Matrix	$\begin{bmatrix} 7929 & 12 \\ 4052 & 7 \end{bmatrix}$			
CV Scores	[0.6625 0.661125 0.662 0.661375 0.661875]			
Mean CV Accuracy	0.661725			

Pada tabel 1 tersebut terlihat bahwa akurasi model mencapai 66.13%, dan hasil validasi silang memberikan estimasi akurasi yang reliabel dengan rata-rata sebesar 66.17%. Namun, saat dilihat lebih rinci melalui *Classification Report*, dapat di identifikasi tantangan lebih lanjut. Dari *report*, terindikasi bahwa model cenderung memiliki kinerja yang baik dalam mengidentifikasi kelas 0 (tidak ada serangan), namun memiliki tantangan dalam mengenali kelas 1 (serangan). Oleh karena itu, perlu investigasi lebih lanjut untuk meningkatkan kemampuan deteksi serangan.

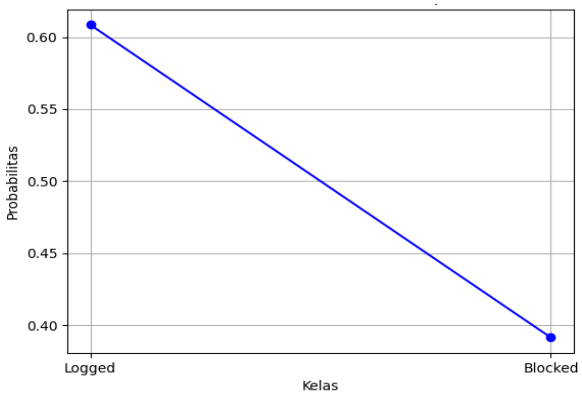
4.2. Deteksi Serangan pada Data Baru

Selain evaluasi model pada data yang telah digunakan untuk pelatihan dan pengujian, peneliti juga mengevaluasi kemampuan model dalam mendeteksi serangan pada data baru. Dalam tahap ini, peneliti menyajikan hasil deteksi untuk data baru yang diberikan pada model.

Tabel 2. Hasil deteksi serangan

Anomaly Scores	*Source Port*	*Packet Length*	*Detection Result*
15.79	12345	1500	0

Dari tabel 2 tersebut dapat dilihat bahwa hasil deteksi (*Detection Result)* yang diperoleh adalah 0, ini berarti model mendeteksi bahwa data baru yang diberikan tidak menunjukkan adanya serangan (kelas 0) atau tindakan yang diambil adalah '*Blocked*'.



Gambar 10. Distribusi probabilitas untuk setiap kelas

Pada gambar 10, menunjukkan sebuah grafik yang menggambarkan distribusi probabilitas untuk dua kelas kunci, yaitu "*Logged*" dan "*Blocked*". Dari grafik tersebut terlihat bahwa probabilitas kunci "*Logged*" lebih tinggi daripada kunci "*Blocked*". Probabilitas kunci "*Logged*" mencapai 0,6, sedangkan probabilitas kunci "*Blocked*" hanya 0,4. Hal ini menunjukkan bahwa kunci "*Logged*" lebih mungkin ditemukan daripada kunci "*Blocked*".

Sehingga hasil deteksi menunjukkan bahwa model dengan sukses mengidentifikasi kelas untuk data baru. Dengan kata lain, model *Random Forest* berhasil memprediksi bahwa data baru tersebut tidak mencerminkan serangan yang perlu diantisipasi atau direspons lebih lanjut.

5. KESIMPULAN DAN SARAN

Dalam konteks ancaman siber yang terus berkembang, model *Random Forest* menawarkan pendekatan yang kuat untuk deteksi serangan. Meskipun perlu pemahaman lebih lanjut terhadap kelas serangan, hasil penelitian ini menggaris bawahi potensi model dalam meningkatkan keamanan siber. Penggunaan dataset yang relevan dan proses *preprocessing* yang cermat menjadi kunci keberhasilan dalam membangun model yang handal. Secara keseluruhan, *Random Forest* dapat menjadi alat efektif dalam pertahanan siber, dengan upaya lebih lanjut untuk mengatasi tantangan spesifik pada kelas serangan tertentu. Pada penelitian mendatang, disarankan untuk memperluas dataset dengan mencakup variasi serangan yang lebih luas, khususnya pada kelas serangan yang menantang. Penelitian lebih lanjut juga dapat mempertimbangkan peningkatan parameter model dan eksplorasi metode *ensemble learning* lainnya untuk meningkatkan akurasi. Selain itu, kolaborasi antara peneliti dan praktisi keamanan siber diperlukan untuk memvalidasi model dalam konteks penggunaan dunia nyata, memastikan implementasi yang sukses dalam meningkatkan keamanan sistem informasi.

DAFTAR PUSTAKA

[1] S. Rabbani and P. Korespondensi, "Prediksi Kategori Serangan Siber dengan Algoritma Klasifikasi Random Forest Menggunakan Rapidminer," *SMATIKA STIKI Inform. J.*, vol. 13, no. 2, pp. 284–293, 2023.

[2] K. G. Fiqri, A. T. Hanuranto, and C. Setianingsih, "Analisa Perbandingan Klasifier Decision Tree, Random Forest, Dan Adaboost Dalam Mendeteksi Serangan Comparative Analysis Decision Tree, Random Forest, And Adaboost Classifier On Detecting Attack," *e-Proceeding Eng.*, vol. 7, no. 1, p. 403, 2020.

[3] A. Ferdita Nugraha, R. Faticha, A. Aziza, and Y. Pristyanto, "Penerapan metode Stacking dan Random Forest untuk Meningkatkan Kinerja Klasifikasi pada Proses Deteksi Web Phishing," *J. Infomedia Tek. Inform. Multimed. Jar.*, vol. 7, no. 1, pp. 39–44, 2022.

[4] D. S. Bhayangkara, H. Dwi Putranto, F. Toriq, and F. Wijayanto, "Analisis Static Malware Menggunakan Algoritma Random Forest Machine Learning," *J. Teknol. Inf.*, vol. 9, no. 2, pp. 172–176, 2023.

[5] Y. Wanli Sitorus, P. Sukarno, S. Mandala, F. Informatika, and U. Telkom, "Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest," *e-Proceeding Eng.*, vol. 8, no. 6, p. 12504, 2021.

[6] I. Maulana and Alamsyah, "Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer," *Indones. J. Math. Nat. Sci.*, vol. 46, no. 2, pp. 83–92, 2023.

[7] S. Komputer and S. D. Bangsa, "Seleksi Fitur dengan Information Gain untuk Meningkatkan Deteksi Serangan DDoS Menggunakan Random Forest," *Techno.COM*, vol. 19, no. 1, pp. 56–66, 2020.

[8] S. Diantika, "Penerapan Teknik Random Oversampling Untuk Mengatasi Imbalance Class Dalam Klasifikasi Website Phishing Menggunakan Algoritma Lightgbm," *J. Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 19–25, 2023.

[9] G. D. Setyawan, A. Yuswanto, A. M. Ridwan, B. Wibowo, and M. Firmansyah, "Implementasi Metode Adasyn Dalam Deteksi Url Berbahaya Menggunakan Machine Learning: Demi Meningkatkan Keamanan Siber Di Era Digital," *TEKNOKOM*, vol. 6, no. 2, pp. 123–126, Aug. 2023.

[10] O. Adiputra and E. Setiawan, "Klasifikasi Malicious URL Menggunakan Algoritma Improved Random Forest dan Random Forest Berbasis Web," *J. SAINS DAN Inform.*, vol. 09, no. 01, pp. 8–14, 2023.