Jurnal Pendidikan dan Teknologi Indonesia (JPTI)
Vol. 5, No. 12, Desember 2025, Hal. 3620-3629
DOI: https://doi.org/10.52436/1.jpti.1196
p-ISSN: 2775-4227
e-ISSN: 2775-4219

Perbandingan Kinerja Algoritma Naïve Bayes, Decision Tree, dan Support Vector Machine dalam Deteksi Serangan Siber Berdasarkan Log Sistem di Universitas Muhammadiyah Purwokerto

Aulya Alyana Aysha[*1], Mukhlis Prasetyo Aji[2], Ermadi Satriya Wijaya[3], Elindra Ambar Pambudi[4]

[1,2,3,4]Teknik Informatika, Fakultas Teknik dan Sains, Universitas Muhammadiyah Purwokerto, Indonesia
Email: [1]alayananaysha@gmail.com, [2]prasetyo-aji@ump.ac.id, [3]ermadi_satriya@ump.ac.id, [4]elindraambarpambudi@ump.ac.id

Abstrak

Keamanan sistem informasi merupakan aspek vital dalam era digital, terutama bagi institusi pendidikan yang sangat bergantung pada infrastruktur teknologi dan rentan terhadap serangan siber. Salah satu faktor penyebab lemahnya pertahanan siber adalah kurangnya pemanfaatan data log sistem sebagai alat deteksi dini terhadap potensi ancaman. Penelitian ini bertujuan untuk mengevaluasi efektivitas tiga algoritma klasifikasi machine learning—Naive Bayes, Decision Tree, dan Support Vector Machine—dalam mendeteksi serangan siber menggunakan data log sistem dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto. Metode penelitian meliputi preprocessing data, pemisahan data menjadi data latih dan uji, pelatihan model, serta evaluasi kinerja menggunakan metrik akurasi, precision, recall, dan f1-score. Hasil pengujian menunjukkan bahwa algoritma Decision Tree memberikan performa terbaik dengan akurasi 99,50% dan nilai evaluasi sebesar 0,9983 pada seluruh metrik. Sementara itu, Naive Bayes memperoleh akurasi terendah sebesar 67,50%, dan Support Vector Machine mencapai 77,25% dengan nilai evaluasi 0,9200. Berdasarkan temuan ini, Decision Tree direkomendasikan sebagai algoritma utama dalam pengembangan sistem deteksi dini untuk meningkatkan keamanan dan ketahanan infrastruktur teknologi informasi di lingkungan perguruan tinggi.

Kata kunci: *Deteksi Ancaman, Klasifikasi, Machine Learning, Serangan Siber, Sistem Informasi*

Comparison of the Performance of Naive Bayes, Decision Tree, and Support Vector Machine Algorithms in Detecting Cyber Attacks Based on System Logs at Muhammadiyah University of Purwokerto

Abstract

Information system security is a vital aspect in the digital era, especially for educational institutions that are highly dependent on technological infrastructure and vulnerable to cyberattacks. One factor contributing to weak cyber defense is the lack of utilization of system log data as an early detection tool for potential threats. This study aims to evaluate the effectiveness of three machine learning classification algorithms—Naive Bayes, Decision Tree, and Support Vector Machine—in detecting cyberattacks using system log data from the Information Systems Bureau of Muhammadiyah University of Purwokerto. The research methods include data preprocessing, data separation into training and test data, model training, and performance evaluation using accuracy, precision, recall, and f1-score metrics. The test results show that the Decision Tree algorithm provides the best performance with an accuracy of 99.50% and an evaluation value of 0.9983 across all metrics. Meanwhile, Naive Bayes obtained the lowest accuracy of 67.50%, and Support Vector Machine achieved 77.25% with an evaluation value of 0.9200. Based on these findings, Decision Tree is recommended as the main algorithm in the development of an early detection system to improve the security and resilience of information technology infrastructure in higher education environments.

Keywords: *Classification, Cyber Attacks, Information Systems, Machine Learning, Threat Detection*

p-ISSN: 2775-4227
e-ISSN: 2775-4219

1. PENDAHULUAN

Dalam era digital, keamanan sistem informasi menjadi aspek krusial bagi berbagai institusi, termasuk sektor pendidikan. Berdasarkan laporan Badan Siber dan Sandi Negara (BSSN) pada tahun 2020 menunjukkan bahwa sektor pendidikan di Indonesia merupakan target serangan siber terbesar dibandingkan sektor lainnya [1]. Pada 2021, University of the Highlands and Islands (UHI) mengalami serangan ransomware oleh penjahat siber dari Eropa Timur/Baltik, menggunakan *malwar*e seperti Cobalt Strike dan Ryuk, yang mengganggu layanan mahasiswa dan staf, termasuk akses jarak jauh ke jaringan, file, aplikasi, dan pencetakan [2]. Peristiwa ini menunjukkan lemahnya keamanan sistem akibat minimnya pemantauan dan analisis log yang seharusnya memberi peringatan dini.

Log sistem merupakan salah satu sumber data yang sangat penting dalam mengidentifikasi ancaman siber [3]. *Log* ini mencatat aktivitas sistem seperti akses pengguna, perubahan data, dan potensi serangan, serta dapat digunakan sebagai dataset utama untuk mendeteksi pola serangan secara otomatis dengan *machine learning* [4]. Di Universitas Muhammadiyah Purwokerto, Biro Sistem Informasi sebagai unit pengelola website dan server juga menghadapi tantangan serupa. Meskipun telah menggunakan teknik responsif standar seperti antivirus, *firewall*, *spyware*, dan mekanisme otentikasi, perlindungan tersebut belum sepenuhnya efektif dalam menangkal intrusi dan ancaman keamanan lainnya [5].

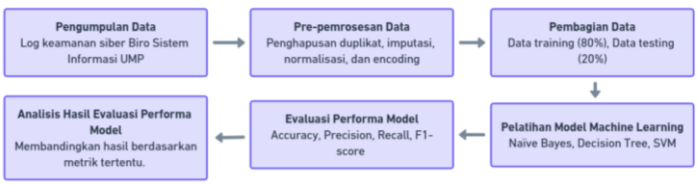
Penerapan metode *machine learning* untuk mendeteksi pola serangan secara otomatis, algoritma klasifikasi seperti Naïve Bayes, Support Vector Machine (SVM), dan Decision Tree telah terbukti efektif dalam mengidentifikasi ancaman siber yang kompleks [6]. Penelitian terkait penerapan algoritma klasifikasi dalam deteksi serangan siber telah dilakukan dalam beberapa penelitian sebelumnya, salah satunya adalah penelitian yang dilakukan oleh [7] yang berjudul "Penerapan Naïve Bayes Classifier, Support Vector Machine, dan Decision Tree untuk Meningkatkan Deteksi Ancaman Keamanan Jaringan". Hasil penelitian ini menunjukkan bahwa ketiga algoritma mampu meningkatkan efektivitas deteksi jika dibandingkan dengan metode konvensional, dengan Decision Tree menghasilkan presisi tertinggi sebesar 0,99, diikuti oleh SVM sebesar 0,98, dan Naïve Bayes sebesar 0,86 [7]. Penelitian tersebut masih terbatas pada data dan konteks tertentu, serta belum mengeksplorasi secara mendalam penerapan *multi-label classification* pada data log sistem di lingkungan pendidikan Indonesia. Selain itu, detail mengenai optimasi parameter dan evaluasi menggunakan metrik komprehensif masih kurang. Oleh karena itu, penelitian ini mengisi kekosongan tersebut dengan menguji ketiga algoritma pada dataset log sistem aktual institusi pendidikan, melakukan *hyperparameter tuning* yang lebih mendalam, serta menggunakan pendekatan *multi-label classification* untuk meningkatkan akurasi dan keandalan deteksi serangan siber.

Decision Tree unggul dalam interpretasi pola dan akurat untuk dataset kecil, namun sensitif terhadap perubahan data [8], sehingga dikembangkan metode ensemble seperti Random Forest untuk mengurangi variansi. SVM efektif dalam klasifikasi biner dengan memisahkan data menggunakan hyperplane optimal, cocok untuk data berdimensi tinggi, dan banyak digunakan dalam deteksi anomali dan pengenalan pola [9]. Sementara itu, Naïve Bayes mengandalkan probabilitas dengan asumsi independensi antar fitur, tetap efisien meski asumsi ini tidak selalu terpenuhi, serta mampu menangani data noisy dan missing value.

Berdasarkan perbandingan ketiga algoritma tersebut, penelitian ini bertujuan untuk membandingkan performa beberapa algoritma klasifikasi dalam mendeteksi intrusi pada jaringan komputer. Penelitian ini secara sistematis membandingkan performa algoritma Naive Bayes, Decision Tree, dan Support Vector Machine menggunakan data log sistem keamanan dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto. Dengan evaluasi pada dataset yang sama, penelitian ini memberikan bukti empiris mengenai algoritma mana yang paling efektif dan akurat dalam mendeteksi intrusi siber. Temuan ini menjadi dasar rekomendasi implementasi sistem keamanan jaringan yang lebih optimal dan terpercaya di institusi pendidikan.

2. METODE PENELITIAN

Penelitian ini menerapkan metodologi yang meliputi beberapa tahapan, sebagaimana ditunjukkan pada alur metode penelitian Gambar 1 berikut ini:



Gambar 1. Alur Umum Penelitian

Langkah-langkah dalam proses penelitian ini dijabarkan sebagai berikut:

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.1 Pengumpulan Data

Pengumpulan data dilakukan menggunakan log keamanan siber yang diperoleh dari Biro Sistem Informasi Universitas Muhammadiyah Purwokerto. Setiap entri dilengkapi dengan atribut *label* yang menunjukkan klasifikasi aktivitas jaringan, yakni *normal*, *suspicious*, atau *attack*, yang digunakan sebagai target pada proses klasifikasi.

2.2 Pre-pemrosesan Data

Tahapan ini dilakukan untuk meningkatkan kualitas data melalui beberapa proses, seperti penghapusan duplikat, imputasi nilai kosong, normalisasi fitur numerik, serta encoding fitur kategorikal agar dapat dibaca oleh algoritma *machine learning*. Seluruh proses ini dilakukan menggunakan bahasa pemrograman *Python* di platform Google Colab dengan bantuan beberapa library, seperti *Pandas* untuk manipulasi data, *NumPy* untuk komputasi numerik, serta *Seaborn* dan *Matplotlib* untuk visualisasi distribusi data. Pendekatan multi-label dalam penelitian ini digunakan untuk memungkinkan setiap entri log diklasifikasikan ke dalam lebih dari satu kategori ancaman secara simultan. Hal ini penting karena dalam dunia nyata, satu aktivitas sistem dapat mengindikasikan lebih dari satu jenis potensi serangan. Dengan demikian, sistem menjadi lebih adaptif dan sensitif dalam mendeteksi pola serangan yang kompleks. Pendekatan ini merujuk pada temuan penelitian [5], yang menunjukkan bahwa klasifikasi multi-label dapat meningkatkan akurasi dalam mengenali beragam jenis ancaman secara bersamaan, dibandingkan dengan pendekatan klasifikasi biner yang hanya mengenali satu jenis serangan pada satu waktu [5].

2.3 Pembagian Data

Pada tahap pembagian data, data dibagi menjadi data *training* (80%) dan data testing (20%). Pembagian ini bertujuan untuk melatih model dalam mengenali pola baru, sehingga mengurangi risiko *overfitting* dan meningkatkan generalisasi terhadap data nyata.

2.4 Pelatihan Model *Machine learning*

Dalam penelitian ini, digunakan tiga algoritma supervised learning yang dipilih berdasarkan karakteristik data serta tujuan analisis yang ingin dicapai. Ketiga algoritma tersebut adalah Naïve Bayes, Decision Tree, dan Support Vector Machine (SVM).

1) Naïve Bayes

Naïve Bayes merupakan algoritma klasifikasi berbasis probabilistik yang dikenal sederhana serta memiliki kecepatan tinggi dalam proses komputasi [10]. Dengan menggunakan teorema Bayes, algoritma ini menghitung probabilitas kelas berdasarkan atribut yang diasumsikan independen. Naive Bayes mampu menangani *missing value* dengan imputasi dan cukup tangguh terhadap data *noisy* [11]. Namun, kinerjanya menurun jika fitur tidak independen. Algoritma ini banyak digunakan dalam klasifikasi teks, prediksi kelulusan, dan penerima beasiswa [12]. Secara umum, Naive Bayes efektif dan efisien untuk tugas klasifikasi dengan fitur yang relatif independen. Persamaan Metode Naive Bayes dapat dilihat pada Rumus 1 [13]. Naïve Bayes merupakan algoritma klasifikasi yang sederhana dan tidak memiliki banyak hyperparameter yang dapat diatur. Oleh karena itu, pada implementasinya tidak dilakukan penyesuaian parameter. Selain itu, hasil evaluasi menunjukkan bahwa Naïve Bayes tidak menunjukkan indikasi overfitting, sehingga performa algoritma tetap stabil tanpa perlu tuning tambahan.

$$P(H|X) = \frac{P(X|H).P(H)}{P(X)} \qquad (1)$$

Keterangan:

- X : Data dengan kelas yang belum diketahui (evidence).
- H : Hipotesis bahwa data X termasuk ke dalam suatu kelas tertentu.
- $P(H|X)$: Probabilitas hipotesis H berdasarkan data X (probabilitas posterior).
- $P(H)$: Probabilitas awal dari hipotesis H sebelum mempertimbangkan data X (prior probability).
- $P(X|H)$: Probabilitas data X terjadi jika hipotesis H benar (likelihood).
- $P(X)$: Probabilitas total dari data X (evidence).

Proses Klasifikasi dengan Naïve Bayes

1. Menentukan probabilitas awal (prior) untuk setiap kelas.
2. Menghitung probabilitas kemunculan fitur (likelihood) pada setiap kelas.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

3. Mengalikan semua probabilitas fitur dengan prior kelas (mengabaikan evidence).
4. Membandingkan hasil probabilitas antar kelas.
5. Mengklasifikasikan data uji ke kelas dengan probabilitas tertinggi.

2) Decision Tree

Decision Tree merupakan metode yang digunakan untuk mempelajari pola klasifikasi dan prediksi dari data dengan menggambarkan hubungan antara variabel atribut (X) dan variabel target (Y) dalam bentuk struktur pohon. [14]. Implementasinya menggunakan parameter max_depth=10 guna membatasi kedalaman pohon. Pemilihan nilai ini bertujuan untuk mengurangi risiko overfitting, mengingat Decision Tree cenderung mempelajari data secara terlalu detail jika tidak dibatasi, sehingga bisa menurunkan performa model pada data uji. Sebagai algoritma *Supervised Learning*, Decision Tree digunakan dalam klasifikasi dan regresi untuk pengenalan pola yang dapat diinterpretasikan [8]. Meskipun efektif, algoritma ini sensitif terhadap perubahan data pelatihan dan memiliki variansi tinggi. Untuk mengatasi hal ini, metode ensemble seperti *Random Forest* dan *Rotation Forest* dikembangkan untuk meningkatkan akurasi klasifikasi. [15]. Rumus *entropi* untuk suatu atribut A dapat dilihat pada Rumus 2 [16].

$$Entropy(A) = -\sum_{i=1}^{n} pi \log_2 pi \qquad (2)$$

Keterangan:

- n : Jumlah kelas pada atribut A
- pi : Proporsi data dari kelas ke-i pada atribut A

Proses Klasifikasi dengan Decision Tree

1. Memilih atribut terbaik sebagai akar pohon berdasarkan nilai informasi (gain/gini).
2. Membagi dataset berdasarkan nilai atribut tersebut.
3. Mengulangi proses pembentukan cabang untuk setiap subset hingga semua data terklasifikasi.
4. Untuk data uji, telusuri cabang pohon sesuai nilai fitur.
5. Kelas ditentukan oleh node daun terakhir yang dicapai.

3) Support Vector Machine (SVM)

Support Vector Machine SVM digunakan dalam berbagai bidang seperti klasifikasi, regresi, dan deteksi *anomaly* [9]. Algoritma ini efektif untuk klasifikasi biner dengan generalisasi yang baik, memisahkan data menggunakan *hyperplane* optimal [17]. SVM digunakan untuk memisahkan kelas data menggunakan hyperplane terbaik. Dalam penelitian ini, digunakan kernel RBF (Radial Basis Function) karena mampu menangani data non-linear. Penggunaan kernel ini juga bertujuan untuk menghindari overfitting dengan cara memetakan data ke ruang berdimensi lebih tinggi secara efisien. SVM diterapkan dalam pengenalan tanda tangan digital, deteksi kecurangan kartu kredit, dan analisis ekspresi gen mikro *array* [18]. Kemampuan SVM bekerja dengan data berdimensi tinggi dan menghindari *overfitting* membuatnya populer di berbagai aplikasi [19]. Algoritma ini mengoptimalkan pencarian hyperplane terbaik dengan margin maksimal, dan pemilihan fungsi kernel yang tepat penting untuk masalah non-linear. Rumus fungsi keputusan SVM dituliskan pada Rumus 3 [20].

$$f(x) = sign\left(\sum_{i=1}^{n} a_i y_i K(x_i, x) + b\right) \quad (3)$$

Keterangan:

- x : Data baru yang akan diklasifikasikan
- xi : Data latih (support vector)
- yi : Label kelas dari data latih +1 atau −1
- ai : Bobot (Lagrange multiplier) yang ditentukan saat pelatihan
- Kxi, x : Fungsi kernel yang menghitung kedekatan (similaritas) antara data latih xi dan data x
- b : Bias dari hyperplane

Proses Klasifikasi dengan SVM

1. Mengubah data ke dalam ruang berdimensi tinggi jika perlu (kernel trick).
2. Mencari hyperplane optimal yang memisahkan antar kelas dengan margin maksimum.
3. Menentukan support vectors sebagai data terdekat dengan hyperplane.
4. Menguji posisi data uji terhadap hyperplane.
5. Mengklasifikasikan data berdasarkan sisi hyperplane tempat data berada.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

2.5 Evaluasi Performa Model

Evaluasi performa model klasifikasi dalam penelitian ini dilakukan dengan menggunakan empat metrik utama, yaitu accuracy, precision, recall, dan F1-score. Keempat metrik ini sangat penting dalam mengukur seberapa baik model dalam mengklasifikasikan data ke dalam tiga label, yaitu *normal*, *suspicious*, dan *attack* berdasarkan atribut target seperti Threat Level, Action, dan Protocol. Evaluasi dilakukan dengan membandingkan hasil prediksi dari ketiga algoritma menggunakan metrik-metrik berikut [21]:

1) *Accuracy* merupakan tingkat kesesuaian prediksi dengan label sebenarnya. Persamaan *Accuracy* dapat dilihat pada Rumus 4.

$$\frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

2) *Precision* merupakan ketepatan model dalam mendeteksi ancaman. Persamaan *Precision* dapat dilihat pada Rumus 5.

$$\frac{TP}{TP+FP} \quad (5)$$

3) *Recall* merupakan kemampuan model dalam menangkap semua ancaman yang ada. Persamaan *Recall* dapat dilihat pada Rumus 6.

$$\frac{TP}{TP+FN} \quad (6)$$

4) *F1-score* merupakan kombinasi harmonis antara *precision* dan *recall* yang memberikan gambaran seimbang terhadap kinerja model. Persamaan *F1-score* l dapat dilihat pada Rumus 7.

$$F1-score = 2\,.\frac{Precision.Recall}{Precision+Recall} \quad (7)$$

Keterangaan:

- *True Positive* (TP) : Model berhasil memprediksi serangan ketika data tersebut memang attack.
- *True Negative* (TN) : Model memprediksi data sebagai normal atau suspicious dan kenyataannya memang bukan serangan.
- *False Positive* (FP) : Model salah memprediksi data sebagai *attack*, padahal data tersebut sebenarnya *normal* atau *suspicious*.
- *False Negative* (FN): Model memprediksi data sebagai *normal* atau *suspicious*, padahal data tersebut sebenarnya adalah *attack*.

2.6 Analisis Hasil Evaluasi Performa Model

Evaluasi kinerja masing-masing algoritma dilakukan dengan membandingkan hasil berdasarkan metrik-metrik yang telah ditentukan. Algoritma yang menunjukkan akurasi tertinggi dan performa metrik lainnya secara konsisten dipilih sebagai metode paling optimal untuk klasifikasi tingkat ancaman dalam sistem informasi Universitas Muhammadiyah Purwokerto.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan dipaparkan hasil penelitian mengenai perbandingan akurasi algoritma Naïve Bayes, Decision Tree, dan Support Vector Machine (SVM) dalam mendeteksi serangan siber pada log sistem Biro Sistem Informasi Universitas Muhammadiyah Purwokerto. Proses pengujian dilakukan dengan menerapkan ketiga algoritma tersebut pada dataset yang telah diproses, kemudian mengevaluasi performa masing-masing model berdasarkan nilai akurasi, presisi, *recall*, dan *F1-score*.

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini merupakan log sistem keamanan siber dari Sistem Informasi Universitas Muhammadiyah Purwokerto, sebagaimana ditampilkan pada Gambar 2. Dataset terdiri dari 2000 entri dan 26 atribut yang merekam aktivitas jaringan, seperti alamat IP sumber dan tujuan, port, protokol, metode akses, tingkat ancaman, serta respons sistem terhadap potensi serangan. Data ini berasal dari data packet, yaitu unit data

p-ISSN: 2775-4227
e-ISSN: 2775-4219

yang dikirim melalui jaringan dan berisi informasi penting seperti pengirim, penerima, serta isi pesan. Setiap packet yang melewati sistem diamati dan dicatat dalam bentuk log, yang kemudian digunakan untuk mendeteksi dan mengklasifikasikan aktivitas jaringan apakah termasuk aman atau berbahaya.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 26 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   No.            2000 non-null   int64
 1   Date           2000 non-null   object
 2   Type           2000 non-null   object
 3   Protocol       2000 non-null   object
 4   Method         2000 non-null   object
 5   URL/Directory  1999 non-null   object
 6   Src Zone       2000 non-null   object
 7   Src IP         2000 non-null   object
 8   Src Location   2000 non-null   object
 9   Src Port       2000 non-null   int64
 10  Xff_IP         2000 non-null   object
 11  Dst Zone       2000 non-null   object
 12  Dst IP         2000 non-null   object
 13  Dst Port       2000 non-null   int64
 14  Rule ID        2000 non-null   object
 15  State Code     2000 non-null   object
 16  Policy Name    2000 non-null   object
 17  Description    2000 non-null   object
 18  Rule Name      2000 non-null   object
 19  Solution       2000 non-null   object
 20  Reference      2000 non-null   object
 21  SANGFOR WIKI   1494 non-null   object
 22  Threat Level   2000 non-null   object
 23  Action         2000 non-null   object
 24  Impact         2000 non-null   object
 25  Data Packet    2000 non-null   object
dtypes: int64(3), object(23)
memory usage: 406.4+ KB
```

Gambar 2. Dataset Penelitian

3.2 Pre-pemrosesan Data

Data log sistem yang telah melalui proses pembersihan duplikasi, imputasi nilai kosong, normalisasi skala fitur, dan konversi data kategorikal ke dalam format numerik, sehingga siap digunakan untuk meningkatkan kompatibilitas dan kinerja model *machine learning.* Preprocessing data merupakan proses label *encoding* pada kolom target kategorikal, yaitu *Threat Level, Protocol,* dan *Action*. Penggunaan *LabelEncoder* dari pustaka *scikit-learn*, nilai-nilai kategorikal pada ketiga kolom tersebut dikonversi menjadi format numerik agar dapat diproses oleh algoritma *machine learning*. Proses ini penting untuk memastikan data bersifat numerik dan kompatibel dengan model yang akan digunakan dalam klasifikasi serangan siber.

3.3 Pembagian Data

Data dipisahkan menjadi 80% untuk pelatihan dan 20% untuk pengujian. Gambar 3 merupakan potongan kode yang menunjukkan proses pembagian data menjadi data latih dan data uji menggunakan fungsi *train_test_split* dari pustaka scikit-learn. Pada kode tersebut, data fitur (X) dan label yang telah diencoding (*y_encoded*) dibagi dengan proporsi 80% untuk data *training* dan 20% untuk data *testing*, sebagaimana ditentukan oleh parameter *test_size=0.2*. Pemisahan ini bertujuan untuk melatih model agar dapat mengenali pola dari data *training* dan kemudian menguji performanya pada data yang belum pernah dilihat sebelumnya (data *testing*), sehingga dapat meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting*. Penggunaan *random_state=42* memastikan bahwa pembagian data dilakukan secara konsisten setiap kali kode dijalankan.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y_encoded, test_size=0.2, random_state=42)
```

Gambar 3. Pembagian Data

3.4 Pelatihan Model *Machine learning* (Naive Bayes, Decision Tree, dan SVM)

Pelaksanaan pelatihan dan evaluasi model *machine learning* dilakukan untuk mengetahui performa masing-masing algoritma dalam mendeteksi serangan siber pada data log sistem. Tiga algoritma yang digunakan dalam penelitian ini adalah Naive Bayes, Decision Tree, dan Support Vector Machine (SVM).

p-ISSN: 2775-4227
e-ISSN: 2775-4219

```python
# Import model & metrics
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.multioutput import MultiOutputClassifier
from sklearn.metrics import classification_report, accuracy_score
import numpy as np
# Daftar model
models = {
    'Naive Bayes': GaussianNB(),
    'Decision Tree': DecisionTreeClassifier(max_depth=10, random_state=42),
    'SVM (Linear Kernel)': SVC(kernel='linear', C=1.0, gamma='auto', probability=True)
}
```

Gambar 4. Kode Pelatihan Model *Machine learning* (Naive Bayes, Decision Tree, dan SVM

Gambar 4 merupakan proses implementasi kode untuk pelatihan model Naive Bayes, Decision Tree, dan SVM menggunakan pendekatan *multi-label classification*. Setiap model diinisialisasi dan dibungkus dengan *MultiOutputClassifier* agar dapat menangani lebih dari satu label target secara simultan. Model dilatih dengan data *X_train* dan *y_train*, kemudian menghasilkan prediksi terhadap X_test. Hasil prediksi tersebut nantinya dievaluasi menggunakan metrik akurasi dan classification report untuk menilai kinerja masing-masing algoritma dalam mendeteksi berbagai jenis serangan siber berdasarkan data log sistem yang tersedia.

3.5 Evaluasi Performa Model

Evaluasi model dilakukan dengan membandingkan performa tiga algoritma klasifikasi, yaitu Naive Bayes, Decision Tree, dan Support Vector Machine (SVM) menggunakan beberapa metrik penting, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*.

```python
# Training & evaluasi
for name, base_model in models.items():
    print(f"\n======= {name} =======")

    model = MultiOutputClassifier(base_model)
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Evaluasi per label
    for i, col in enumerate(target_columns):
        print(f"\n-- Report for target: {col} --")
        acc = accuracy_score(y_test.iloc[:, i], y_pred[:, i])
        print(f"Akurasi: {acc:.2f}")
        print(classification_report(y_test.iloc[:, i], y_pred[:, i], zero_division=0))
        # zero_division=0 agar warning hilang
```

Gambar 5. Kode Pengujian Model *Machine learning* (Naive Bayes, Decision Tree, dan SVM)

Gambar 5 menunjukkan proses pelatihan dan evaluasi model *machine learning* menggunakan beberapa algoritma dalam pendekatan multi-label classification. Model dilatih menggunakan MultiOutputClassifier pada data pelatihan, kemudian dilakukan prediksi terhadap data uji. Setelah itu, evaluasi dilakukan untuk setiap target label (*Threat Level, Protocol, dan Action*) dengan menghitung nilai akurasi dan mencetak laporan klasifikasi (*classification_report*) yang mencakup metrik *precision*, *recall*, dan *f1-score*. Kode ini juga menghindari munculnya peringatan dengan menetapkan *zero_division=0* saat menghitung metrik.

3.6 Analisis Hasil Evaluasi Performa Model

Analisis performa model dilakukan untuk memperoleh pemahaman yang menyeluruh terhadap efektivitas masing-masing algoritma dalam melakukan klasifikasi data. Evaluasi ini mencakup pengukuran menggunakan metrik *accuracy*, *precision*, *recall*, dan *f1-score* untuk menilai kualitas prediksi dari model Naive Bayes, Decision Tree, dan Support Vector Machine secara objektif. Berdasarkan grafik perbandingan kinerja model pada Gambar 6, algoritma Decision Tree menunjukkan performa terbaik dengan *accuracy* sangat tinggi sebesar 99,50% serta nilai *precision*, *recall*, dan *f1-score* yang hampir sempurna (0,9983), mencerminkan kemampuannya dalam mengklasifikasikan data secara akurat dan seimbang di semua label. Sementara itu, Support Vector Machine (SVM) menunjukkan performa yang cukup baik dengan *accuracy* 77,25% dan nilai *precision*, *recall*, serta *f1-score* sebesar 0,9200, menandakan efektivitas model dalam mengenali pola mayoritas meskipun masih terdapat ruang untuk perbaikan. Di sisi lain, Naive Bayes mencatat *accuracy* terendah yaitu 67,50%, meskipun nilai

p-ISSN: 2775-4227
e-ISSN: 2775-4219

precision, *recall*, dan *f1-score* tetap tinggi di angka 0,8875. Hal ini menunjukkan bahwa meskipun Naive Bayes mampu menghasilkan prediksi yang konsisten pada beberapa label, model ini kurang optimal dalam menangani kompleksitas data *multi-label*. Dengan demikian, Decision Tree dapat disimpulkan sebagai model paling andal untuk klasifikasi serangan siber pada *data log system* yang digunakan.

Gambar 6. Perbandingan Model Machine learning (Naive Bayes, Decision Tree, dan SVM)

3.7 Diskusi

Decision Tree menunjukkan performa terbaik dengan akurasi sebesar 99,50% dan skor precision, recall, serta f1-score sebesar 0,9983. Keunggulan ini diperoleh karena kemampuannya menangani data non-linear dan membentuk struktur pohon keputusan berdasarkan information gain. Decision Tree juga mampu mengelola data kategorikal maupun numerik secara langsung dan tidak terlalu sensitif terhadap outlier, sehingga cocok untuk karakteristik data log sistem keamanan. Hasil ini sejalan dengan temuan [7], yang juga menunjukkan bahwa Decision Tree mengungguli SVM dan Naive Bayes dalam klasifikasi ancaman jaringan, dengan presisi tertinggi sebesar 0,99.

Support Vector Machine (SVM) memperoleh hasil evaluasi menengah dengan akurasi 77,25% dan skor precision, recall, serta f1-score sebesar 0,9200. Meskipun SVM mampu memaksimalkan margin antar kelas, performanya sangat dipengaruhi oleh pemilihan kernel dan parameter. Dalam penelitian ini, keterbatasan tuning parameter serta kompleksitas data berdampak pada kinerjanya. SVM juga kurang ideal saat menghadapi data dengan noise tinggi atau ketika fitur memiliki korelasi kuat, karena mengasumsikan pemisahan kelas yang optimal dalam ruang berdimensi tinggi. Penelitian sebelumnya [7] juga mencatat bahwa SVM memberikan hasil yang baik namun tidak sebaik Decision Tree dalam hal stabilitas dan interpretabilitas.

Naïve Bayes mencatat akurasi terendah yaitu 67,50%, meskipun precision, recall, dan f1-score masih relatif tinggi (0,8875). Kinerja rendah ini disebabkan oleh asumsi independensi antar fitur yang menjadi dasar algoritma, padahal data log sistem keamanan cenderung memiliki fitur yang saling berkorelasi, misalnya antara port dan metode akses. Hal ini menyebabkan probabilitas gabungan tidak akurat, sehingga menurunkan efektivitas model. Temuan ini konsisten dengan studi sebelumnya yang menyebutkan bahwa Naïve Bayes kurang cocok untuk data real-world yang kompleks dan saling bergantung, namun masih berguna dalam kasus dengan distribusi data tidak seimbang karena kesederhanaannya. Untuk memperjelas perbandingan, Tabel 1 menyajikan ringkasan metrik evaluasi ketiga algoritma:

Tabel 1. Metrik Evaluasi

Algoritma	Akurasi (%)	Precision	Recall	F1-Score
Decision Tree	99,5	0,9983	0,9983	0,9983
SVM	77,25	0,92	0,92	0,92
Naive Bayes	67,5	0,8875	0,8875	0,8875

Decision Tree adalah algoritma paling efektif untuk mendeteksi serangan siber dalam konteks data log sistem Universitas Muhammadiyah Purwokerto. Namun, untuk validitas lebih luas, penelitian lanjutan perlu melibatkan dataset yang lebih beragam, penggunaan teknik validasi silang, serta eksplorasi metode ensemble seperti Random Forest atau Gradient Boosting untuk hasil yang lebih robust. Keterbatasan penelitian ini antara lain penggunaan satu dataset tanpa validasi silang yang membatasi generalisasi hasil. Oleh karena itu, pengembangan model selanjutnya sebaiknya melibatkan dataset yang lebih besar dan beragam, teknik validasi silang untuk menguji

p-ISSN: 2775-4227
e-ISSN: 2775-4219

stabilitas model, serta eksplorasi algoritma lain seperti metode ensemble yang potensial meningkatkan akurasi dan keandalan deteksi serangan siber.

4. KESIMPULAN

Penelitian ini menegaskan bahwa algoritma Decision Tree menunjukkan kinerja terbaik dalam mendeteksi serangan siber berdasarkan data log sistem di lingkungan pendidikan, dengan keunggulan dalam akurasi dan kestabilan model melalui optimasi hyperparameter. Pendekatan multi-label classification yang didukung oleh preprocessing yang tepat terbukti efektif untuk klasifikasi log sistem. Secara praktis, model Decision Tree memberikan solusi yang interpretatif, efisien, dan mudah dipelihara oleh institusi pendidikan. Penelitian selanjutnya sebaiknya menggunakan dataset yang lebih besar dan beragam, menerapkan teknik validasi silang, serta mengeksplorasi penggunaan model ensemble guna meningkatkan kemampuan generalisasi dan akurasi deteksi.

DAFTAR PUSTAKA

[1] D. Hariyadi and F. E. Nastiti, "Analisis Keamanan Sistem Informasi Menggunakan Sudomy dan OWASP ZAP di Universitas Duta Bangsa Surakarta," *J. Komtika (Komputasi dan Inform.*, vol. 5, no. 1, pp. 35–42, 2021, doi: 10.31603/komtika.v5i1.5134.

[2] H. S. Lallie, A. Thompson, E. Titis, and P. Stephens, "Analysing Cyber Attacks and Cyber Security Vulnerabilities in the University Sector," *Computers*, vol. 14, no. 2, pp. 1–28, 2025, doi: 10.3390/computers14020049.

[3] T. G. Laksana and S. Mulyani, "Pengetahuan Dasar Identifikasi Dini Deteksi Serangan Kejahatan Siber Untuk Mencegah Pembobolan Data Perusahaan," *J. Ilm. Multidisiplin*, vol. 3, no. 01, pp. 109–122, 2024, doi: 10.56127/jukim.v3i01.1143.

[4] C. Tarigan, V. Jeremias, L. Engel, and D. Angela, "Sistem Pengawasan Kinerja Jaringan Server Web Apache dengan Log Management System ELK (Elasticsearch , Logstash , Kibana)," pp. 7–14, 2018.

[5] M. P. Aji, "Klasifikasi Tingkat Ancaman Siber menggunakan Pembelajaran Mesin pada Web Application Firewall (WAF) Cyber," *J. Media Pratama*, vol. 17, no. 1, pp. 61–73, 2023.

[6] A. Purnomo, A. Kurniasih, A. Nuraminah, and S. Hartati, "Peran Artificial Intelligence dalam Deteksi Dini Ancaman Keamanan Jaringan," vol. 13, pp. 2044–2048, 2024.

[7] A. T. Zy, A. T. Sasongko, and A. Z. Kamalia, "Penerapan Naïve Bayes Classifier, Support Vector Machine, dan Decision Tree untuk Meningkatkan Deteksi Ancaman Keamanan Jaringan," *Media Online)*, vol. 4, no. 1, pp. 610–617, 2023, doi: 10.30865/klik.v4i1.1134.

[8] B. Siswoyo, "MultiClass Decision Forest Machine Learning Artificial Intelligence," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 1–7, 2020, doi: 10.30871/jaic.v4i1.1155.

[9] H. Kibriya, R. Amin, J. Kim, M. Nawaz, and R. Gantassi, "A Novel Approach for Brain Tumor Classification Using an Ensemble of Deep and Hand-Crafted Features," *Sensors*, vol. 23, no. 10, 2023, doi: 10.3390/s23104693.

[10] K. S. Arlandy *et al.*, "Mengoptimalkan Kinerja Naïve Bayes Pada Ancaman Modern Dengan Menggunakan PCA Pada Data Intrusion Detection System (IDS)," vol. 8, no. 1, 2025.

[11] A. I. S. Azis, Budy Santoso, and Serwin, "Local Learning K-Nearest Neighbor in Absolute Correlation Weighted Naïve Bayes for Numerical Data Classification," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 1, pp. 28–36, 2020, doi: 10.29207/resti.v4i1.1348.

[12] A. C. Fauzan and K. Hikmah, "Implementasi Algoritma Naive Bayes Dalam Analisis Polarisasi Opini Masyarakat Terkait Vaksin Covid-19," *Rabit J. Teknol. dan Sist. Inf. Univrab*, vol. 7, no. 2, pp. 122–128, 2022, doi: 10.36341/rabit.v7i2.2403.

[13] Rayuwati, Husna Gemasih, and Irma Nizar, "Implementasi Algoritma Naive Bayes untuk Memprediksi Tingkat Penyebaran Covid-19 di Indonesia," *Jural Ris. Rumpun Ilmu Tek.*, vol. 1, no. 1, pp. 38–46, 2022, doi: 10.55606/jurritek.v1i1.127.

[14] I. Sutoyo, "Implementasi Algoritma Decision Tree Untuk Klasifikasi Data Peserta Didik," *J. Pilar Nusa Mandiri*, vol. 14, no. 2, p. 217, 2018, doi: 10.33480/pilar.v14i2.926.

[15] Y. Purwananto, D. Purwitasari, and Y. Nugroho, "Pengkategorian Isi Berita Berbahasa Indonesia Menggunakan Algoritma Symbolic Rule Induction Berbasis Decision Tree," *JUTI J. Ilm. Teknol. Inf.*, vol. 3, no. 1, p. 55, 2004, doi: 10.12962/j24068535.v3i1.a131.

p-ISSN: 2775-4227
e-ISSN: 2775-4219

[16] T. Wiratama Putra, A. Triayudi, and A. Andrianingsih, "Analisis Sentimen Pembelajaran Daring Menggunakan Metode NaÃ¯ve Bayes, KNN, dan Decision Tree," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 6, no. 1, pp. 20–26, 2022, doi: 10.35870/jtik.v6i1.368.

[17] W. S. Noble, "What is a support vector machine?," *Nat. Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006, doi: 10.1038/nbt1206-1565.

[18] F. Fahmi, "Model Support Vector Regression (SVR) Berdimensi Tinggi dengan Pendekatan Fungsi Kernel Berbeda untuk Peramalan Harga Saham TLKM: Sebuah Pemodelan Deret Waktu Selama Masa Pandemi Covid-19," *J. Infomedia*, vol. 5, no. 2, p. 44, 2021, doi: 10.30811/jim.v5i2.2033.

[19] C. Xia *et al.*, *Classification research on syndromes of TCM based on SVM*. 2009. doi: 10.1109/BMEI.2009.5305418.

[20] R. R. S. Putri Kumala Sari, "Komparasi Algoritma Support Vector Machine dan Random Forest untuk Analisis Sentimen Metaverse," vol. 7, no. 1, pp. 31–39, 2024.

[21] S. Dalal *et al.*, "Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree," *J. Cloud Comput.*, vol. 12, no. 1, 2023, doi: 10.1186/s13677-023-00517-4.