Analisis Metode K-Nearest Neighbor Dan Metode SVM Pada Log Serangan Jaringan Untuk Rekomendasi Keamanan Jaringan

Fadil Habib, Rika Rosnelly, Bob Subhan Riza
Program Studi Magister Ilmu Komputer, Fakultas Teknik Dan Ilmu Komputer
Universitas Potensi Utama Medan
zhilan35@gmail.com, rika@potensi-utama.ac.id, bob@potensi.gmail.com

Abstrak

Penelitian ini dilakukan untuk menganalisis dan membandingkan kinerja dua algoritma machine learning, yaitu K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM), dalam proses klasifikasi log serangan jaringan menggunakan dataset CICIDS2017. Hasil dari penelitian ini diharapkan dapat memberikan gambaran mengenai metode yang memiliki performa terbaik dalam mendeteksi aktivitas serangan jaringan sehingga dapat menjadi dasar rekomendasi dalam meningkatkan keamanan jaringan komputer. Berdasarkan hasil penelitian yang telah dilakukan mengenai analisis metode K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) pada klasifikasi log serangan jaringan menggunakan dataset CICIDS2017. Metode K-Nearest Neighbor (KNN) mampu melakukan klasifikasi aktivitas jaringan menjadi kategori normal dan serangan dengan performa yang sangat baik. Berdasarkan hasil pengujian yang dilakukan, model KNN menghasilkan nilai accuracy sebesar 0,9989, precision sebesar 0,9975, recall sebesar 0,9969, dan F1-score sebesar 0,9972, yang menunjukkan bahwa metode KNN memiliki kemampuan yang sangat tinggi dalam mendeteksi aktivitas serangan jaringan pada dataset CICIDS2017. Metode Support Vector Machine (SVM) juga mampu melakukan klasifikasi aktivitas jaringan dengan performa yang cukup baik. Hasil evaluasi model menunjukkan nilai accuracy sebesar 0,9756, precision sebesar 0,9575, recall sebesar 0,9134, dan F1-score sebesar 0,9349, yang menunjukkan bahwa metode SVM masih memiliki kemampuan yang baik dalam mendeteksi aktivitas jaringan normal dan serangan, meskipun performanya masih berada di bawah metode KNN pada dataset yang digunakan. Metode K-Nearest Neighbor (KNN) lebih efektif digunakan dalam proses deteksi serangan jaringan pada dataset CICIDS2017. Dengan tingkat akurasi dan kemampuan deteksi yang tinggi, metode ini dapat dijadikan sebagai rekomendasi metode machine learning yang dapat digunakan dalam pengembangan sistem keamanan jaringan berbasis Intrusion Detection System (IDS) untuk membantu mengidentifikasi aktivitas serangan jaringan secara lebih akurat.

Kata Kunci : *Serangan Jaringan, K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM), Keamanan.*

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang semakin pesat menyebabkan peningkatan penggunaan sistem jaringan komputer dalam berbagai sektor, baik pada instansi pemerintahan, perusahaan swasta, maupun layanan berbasis internet. Ketergantungan yang tinggi terhadap sistem jaringan ini juga diikuti dengan meningkatnya potensi ancaman keamanan siber yang dapat mengganggu stabilitas sistem informasi. Berdasarkan laporan publikasi Badan Siber dan Sandi Negara (BSSN) pada bulan Maret 2025, tercatat sebanyak 27.839.199 anomali yang berkaitan dengan unauthorized access dan system misconfiguration yang terjadi pada berbagai sistem jaringan di Indonesia. Hal ini menunjukkan bahwa serangan siber masih menjadi ancaman serius terhadap keamanan infrastruktur teknologi informasi.

Salah satu jenis serangan yang sering terjadi adalah brute force attack. Serangan brute force merupakan teknik yang digunakan oleh penyerang untuk mendapatkan akses tidak sah ke dalam sistem dengan cara mencoba berbagai kombinasi username dan password secara berulang hingga menemukan kombinasi yang benar. Teknik ini biasanya dilakukan secara otomatis menggunakan skrip atau perangkat lunak tertentu sehingga mampu melakukan percobaan dalam jumlah sangat besar dalam waktu singkat. Apabila serangan ini berhasil, maka penyerang dapat memperoleh akses terhadap sistem dan berpotensi mengeksploitasi berbagai kerentanan yang ada pada aplikasi atau jaringan [1].

Dalam konteks keamanan informasi, suatu sistem harus mampu menjaga tiga prinsip utama yang dikenal sebagai Confidentiality, Integrity, dan Availability (CIA Triad).

Prinsip kerahasiaan (confidentiality) memastikan bahwa informasi hanya dapat diakses oleh pihak yang berwenang, integritas (integrity) menjamin bahwa data tidak mengalami perubahan yang tidak sah, sedangkan ketersediaan (availability) memastikan bahwa sistem dapat diakses ketika dibutuhkan [2]. Apabila sistem berhasil ditembus melalui serangan brute force atau serangan jaringan lainnya, maka ketiga prinsip tersebut dapat terganggu sehingga berdampak pada keamanan dan keandalan sistem informasi.

Untuk mengatasi permasalahan tersebut, diperlukan suatu mekanisme yang mampu mendeteksi aktivitas serangan jaringan secara otomatis sehingga dapat membantu administrator jaringan dalam mengidentifikasi pola serangan yang terjadi. Salah satu pendekatan yang banyak digunakan dalam sistem deteksi intrusi (Intrusion Detection System (IDS) adalah dengan memanfaatkan teknik machine learning. Machine learning mampu mempelajari pola dari data lalu lintas jaringan dan melakukan klasifikasi terhadap aktivitas jaringan yang tergolong normal maupun yang termasuk kategori serangan.

Salah satu metode machine learning yang dapat digunakan untuk proses klasifikasi adalah K-Nearest Neighbor (KNN). KNN merupakan algoritma klasifikasi berbasis jarak yang bekerja dengan cara menentukan kedekatan antara data uji dengan sejumlah data latih terdekat. Proses klasifikasi dilakukan dengan melihat mayoritas kelas dari sejumlah tetangga terdekat berdasarkan perhitungan jarak, seperti Euclidean Distance. Metode KNN dikenal memiliki kelebihan dalam hal kemudahan implementasi serta kemampuan menghasilkan akurasi yang tinggi pada berbagai permasalahan klasifikasi[3].

Selain KNN, metode lain yang sering digunakan dalam klasifikasi data adalah Support Vector Machine (SVM). SVM merupakan algoritma machine learning yang bekerja dengan mencari hyperplane optimal yang dapat memisahkan data ke dalam kelas yang berbeda dengan margin maksimum. Metode ini memiliki kemampuan yang baik dalam menangani data berdimensi tinggi serta mampu memodelkan hubungan linier maupun non-linier melalui penggunaan fungsi kernel[4].

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian ini, digunakan artikel-artikel penelitian yang digunakan sebagai bahan pendukung penelitian. Pada penelitian yang dilakukan oleh [5] dengan judul Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services menghasilkan kesimpulan berupa Model ExtKNNCF yang diusulkan berhasil meningkatkan akurasi dan personalisasi rekomendasi dibanding metode-metode sebelumnya.

Dengan mempertimbangkan kognisi pengguna dan adaptivitas K dalam algoritma KNN, sistem ini mampu memberikan rekomendasi yang lebih relevan, termasuk untuk pengguna baru (cold start). Ke depannya, model ini berpotensi dikembangkan untuk lintas domain dan dievaluasi dengan lebih banyak dataset publik.

Pada penelitian yang dilakukan oleh [6] dengan judul A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things menghasilkan kesimpulan yaitu: metode pembelajaran mesin dan pembelajaran dalam, terutama model ensemble seperti XGBoost dan CatBoost, sangat efektif dalam mendeteksi serangan siber pada perangkat IoT. Penggunaan teknik balancing seperti SMOTE juga terbukti mampu meningkatkan performa model. Penelitian ini memberikan kontribusi penting dalam pengembangan sistem keamanan siber berbasis AI untuk lingkungan IoT yang semakin kompleks.

Pada penelitian yang dilakukan oleh [7] dengan judul Cloud DDoS Attack Detection Model with Data Fusion & Machine Learning Classifiers, menghasilkan kesimpulan yaitu:pendekatan berbasis data fusion dan machine learning efektif dalam mendeteksi serangan DDoS pada lingkungan cloud. Algoritma Decision Tree dinyatakan sebagai metode paling unggul untuk klasifikasi data serangan. Penelitian juga menyarankan penggabungan beberapa classifier dan eksplorasi teknik seleksi fitur baru untuk penelitian masa depan.

Pada penelitian yang dilakukan oleh [8] dengan judul Cybersecurity Attack Detection Model Using Machine Learning Techniques, menghasilkan kesimpulan yaitu: Penelitian ini berhasil membangun sistem deteksi serangan siber berbasis machine learning yang sangat akurat dan efisien, dengan Random Forest sebagai teknik terbaik. Fokus penelitian masih terbatas pada klasifikasi lalu lintas sebagai malicious atau benign. Penelitian lanjutan disarankan untuk mengembangkan klasifikasi berdasarkan jenis serangan dan eksplorasi metode ensemble learning untuk meningkatkan akurasi lebih lanjut.

Pada penelitian yang dilakukan oleh [9] dengan judul Detection of Distributed Denial of Service Attacks for IoT-Based Healthcare Systems, menghasilkan kesimpulan yaitu: Penelitian ini berhasil mengembangkan pendekatan baru dalam mendeteksi serangan DDoS pada sistem IoT berbasis SDN yang lebih akurat dibanding metode sebelumnya. Namun, masih terdapat ruang untuk peningkatan dalam hal efisiensi pemrosesan dan kemampuan adaptasi terhadap serangan jenis baru. Di masa depan, pengembangan metode deteksi real-time yang lebih efisien sangat penting untuk menghadapi kompleksitas serangan siber di sektor kesehatan.

Pada penelitian yang dilakukan oleh [10] dengan judul A Hybrid Deep Learning Model for Intrusion Detection in Internet of Vehicles, menghasilkan kesimpulan yaitu:Kerangka kerja kriptografi hibrida ini memberikan solusi keamanan yang komprehensif untuk penyimpanan data di cloud dengan menggabungkan enkripsi berlapis, manajemen kunci adaptif, dan pembatasan akses waktu. Framework ini tidak hanya meningkatkan keamanan tetapi juga efisiensi dalam pemrosesan data. Meski demikian, tantangan seperti kompleksitas komputasi dan skalabilitas sistem perlu diperhatikan dalam implementasi lebih lanjut.

2.2 Machine Learning

Machine Learning merupakan cabang dari bidang ilmu kecerdasan buatan *(Artificial Intelligence)* yang cakupannya bertujuan untuk memungkinkan komputer berjalan tanpa diprogram secara langsung. *Machine Learning* berfokus pada objek pemanfaatan data serta algoritma dalam meniru metode belajar pada manusia dan secara bertahap dapat meningkatkan akurasinya. Metode *Machine Learning* dilatih guna memprediksi serta mengklasifikasi dalam pengembangan data. *Machine Learning* terdiri dari 3 bagian diantaranya: *Supervised Learning, Reinforcement Learning* dan *Unsupervised Learning* (Robison Manalu et al., 2022).

2.3 Network Intrusion Detection System (IDS)

Network Intrusion Detection System (IDS) merupakan sistem keamanan jaringan yang berfungsi untuk memantau lalu lintas jaringan serta mendeteksi aktivitas yang mencurigakan atau tidak sah dalam suatu sistem komputer. IDS bekerja dengan menganalisis pola aktivitas jaringan untuk mengidentifikasi potensi serangan seperti Denial of Service (DoS), Distributed Denial of Service (DDoS), port scanning, brute force attack, maupun malware yang dapat mengganggu kinerja jaringan dan membahayakan keamanan informasi. Dengan adanya IDS, administrator jaringan dapat memperoleh peringatan dini terhadap aktivitas berbahaya sehingga dapat segera melakukan tindakan pencegahan untuk melindungi sistem jaringan dari berbagai ancaman keamanan siber. [11]

Selain itu, perkembangan teknologi keamanan jaringan mendorong pemanfaatan metode **machine learning** dalam pengembangan IDS untuk meningkatkan kemampuan sistem dalam mendeteksi serangan jaringan secara lebih akurat. Model berbasis machine learning mampu mempelajari pola lalu lintas jaringan sehingga dapat membedakan aktivitas normal dan aktivitas serangan secara lebih efektif. Pendekatan ini memungkinkan sistem IDS untuk mendeteksi berbagai jenis intrusi yang terus berkembang serta meningkatkan keandalan sistem dalam menjaga keamanan jaringan komputer dari berbagai ancaman siber. [12]

24 Log Serangan Jaringan

Log serangan jaringan merupakan catatan aktivitas lalu lintas jaringan yang berisi informasi mengenai paket data, alamat IP sumber dan tujuan, port komunikasi, serta protokol yang digunakan dalam suatu komunikasi jaringan. Data log ini menjadi sumber informasi penting dalam proses analisis keamanan jaringan karena dapat digunakan untuk mengidentifikasi pola aktivitas normal maupun aktivitas mencurigakan yang mengindikasikan adanya serangan. Dalam sistem deteksi intrusi berbasis jaringan, log jaringan biasanya diperoleh melalui proses *packet capture* atau pencatatan lalu lintas jaringan yang kemudian dianalisis menggunakan berbagai metode analitik maupun teknik pembelajaran mesin untuk mendeteksi aktivitas berbahaya pada jaringan. Oleh karena itu, ketersediaan log jaringan yang akurat dan terstruktur sangat penting dalam mendukung proses deteksi dan analisis serangan jaringan secara efektif. [13]

Selain itu, log serangan jaringan juga berperan sebagai sumber data utama dalam proses pelatihan model *machine learning* pada sistem deteksi intrusi. Data log yang dikumpulkan biasanya terdiri dari lalu lintas jaringan normal (*bonafide traffic*) dan lalu lintas yang mengandung serangan (*malicious traffic*). Kedua jenis data tersebut digunakan untuk membangun dataset yang dapat melatih model klasifikasi dalam membedakan aktivitas jaringan normal dan aktivitas serangan. Penggunaan dataset yang berkualitas sangat berpengaruh terhadap performa model deteksi intrusi, karena pola-pola serangan yang terdapat dalam log jaringan akan dipelajari oleh model untuk mendeteksi serangan serupa pada lalu lintas jaringan di masa mendatang. Dengan demikian, pengolahan log serangan jaringan menjadi tahap penting dalam pengembangan sistem keamanan jaringan berbasis pembelajaran mesin. [14]

2.5 Metode *K-Nearest Neighbor (KNN)*

Metode K-Nearest Neighbor (KNN) merupakan salah satu algoritma *supervised learning* yang digunakan untuk melakukan klasifikasi data berdasarkan tingkat kemiripan antara data uji dengan data latih. Prinsip dasar dari algoritma ini adalah menentukan kelas suatu data baru berdasarkan mayoritas kelas dari sejumlah tetangga terdekatnya yang ditentukan oleh nilai parameter k. Proses klasifikasi pada metode KNN dilakukan dengan beberapa tahapan, yaitu menentukan nilai k, menghitung jarak antara data uji dengan seluruh data latih, mengurutkan jarak untuk memperoleh tetangga terdekat, kemudian menentukan kelas berdasarkan mayoritas kelas dari tetangga tersebut.

Metode ini termasuk algoritma yang sederhana karena tidak memerlukan proses pelatihan model secara eksplisit dan seluruh data pelatihan disimpan sebagai referensi dalam proses klasifikasi data baru. [15]

Dalam proses klasifikasi menggunakan metode KNN, pengukuran kedekatan antar data biasanya dilakukan menggunakan fungsi jarak, salah satunya adalah Euclidean Distance. Rumus perhitungan jarak Euclidean yang umum digunakan dalam algoritma KNN dapat dinyatakan sebagai berikut:

$$d(x, y) = \sqrt{\sum_{n=1}^{n} (x_i - y_i)^2}$$

di mana $d(x_i, x_j)$ merupakan jarak antara dua data, x_{ik} dan x_{jk} adalah nilai atribut ke-kkk dari masing-masing data, serta nnn merupakan jumlah atribut pada data. Setelah jarak antar data dihitung, sistem akan memilih sejumlah k tetangga terdekat dan menentukan kelas data berdasarkan mayoritas kelas dari tetangga tersebut. Pendekatan ini memungkinkan sistem untuk mengklasifikasi-kan data baru berdasarkan kemiripan karakteristik dengan data yang telah diketahui sebelumnya sehingga algoritma KNN banyak digunakan dalam berbagai penelitian *machine learning*, termasuk pada sistem deteksi intrusi jaringan untuk membedakan aktivitas jaringan normal dan aktivitas serangan. [14]

2.6 *Support Vector Machine (SVM)*

Support Vector Machine (SVM) merupakan salah satu algoritma *supervised learning* yang digunakan untuk melakukan klasifikasi data dengan cara mencari *hyperplane* terbaik yang dapat memisahkan dua kelas data secara optimal. Konsep utama dari metode ini adalah menentukan batas pemisah yang memiliki *margin maksimum* antara dua kelas sehingga mampu meningkatkan kemampuan generalisasi model. Dalam proses klasifikasi, SVM memanfaatkan data pelatihan untuk membangun model yang dapat membedakan antara dua kelas, misalnya kelas data normal dan kelas data serangan dalam sistem deteksi intrusi. Metode ini dikenal memiliki kemampuan klasifikasi yang baik karena mampu bekerja secara efektif pada data berdimensi tinggi serta memiliki tingkat akurasi yang tinggi dalam berbagai aplikasi *machine learning*. [15]

Secara matematis, hyperplane pada metode SVM dapat dinyatakan dalam persamaan berikut:

$$w \cdot x + b = 0$$

di mana w merupakan vektor bobot yang tegak lurus terhadap hyperplane, x merupakan vektor fitur dari data, dan b merupakan nilai bias. Persamaan ini digunakan untuk memisahkan dua kelas data dalam ruang fitur sehingga data dengan label kelas positif dan negatif dapat dipisahkan secara optimal.

Untuk menentukan batas klasifikasi, fungsi keputusan SVM dapat dinyatakan sebagai:

$$f(x) = sign(w \cdot x + b)$$

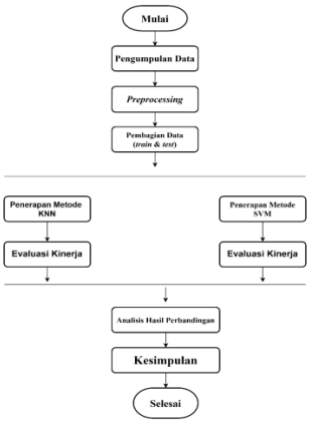
di mana fungsi **sign** digunakan untuk menentukan kelas dari suatu data berdasarkan posisi data tersebut terhadap hyperplane. Dalam penerapannya pada sistem deteksi intrusi jaringan, metode SVM digunakan untuk mengklasifikasikan lalu lintas jaringan ke dalam kategori aktivitas normal atau aktivitas serangan berdasarkan pola data yang dipelajari dari dataset pelatihan. Pendekatan ini memungkinkan sistem keamanan jaringan untuk mendeteksi berbagai jenis serangan secara lebih akurat melalui proses pembelajaran terhadap karakteristik lalu lintas jaringan. [16]

III. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian merupakan rangkaian proses yang dilakukan secara sistematis dalam pelaksanaan penelitian untuk mencapai tujuan yang telah ditetapkan. Pada penelitian ini, tahapan dimulai dari proses pengumpulan data yang digunakan sebagai bahan analisis, kemudian dilanjutkan dengan tahap *preprocessing data* untuk membersihkan dan mempersiapkan data sebelum dilakukan proses pengolahan lebih lanjut. Setelah data diproses, dilakukan pembagian data menjadi data pelatihan (*training*) dan data pengujian (*testing*) yang digunakan untuk membangun serta menguji model klasifikasi.

Selanjutnya, dilakukan penerapan metode klasifikasi yaitu K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) untuk mendeteksi pola serangan jaringan berdasarkan data yang telah diproses sebelumnya. Hasil dari masing-masing metode kemudian dievaluasi menggunakan metrik evaluasi kinerja seperti accuracy, precision, recall, dan F1-score. Tahap terakhir adalah melakukan analisis perbandingan terhadap hasil kinerja kedua metode tersebut untuk mengetahui metode yang memiliki performa terbaik dalam mendeteksi serangan jaringan, kemudian ditarik kesimpulan dari hasil penelitian yang telah dilakukan. Adapun tahapan penelitian secara keseluruhan dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

6.2 Pengumpulan Data

Pada penelitian ini, data yang digunakan merupakan dataset lalu lintas jaringan yang diperoleh dari Canadian Institute for Cybersecurity (CIC), University of New Brunswick, yang tersedia secara publik melalui situs resmi CIC. Dataset tersebut berisi data aktivitas jaringan yang mencakup lalu lintas normal maupun berbagai jenis serangan jaringan, seperti Distributed Denial of Service (DDoS), FTP-Patator, DoS Hulk, DoS Slowloris, dan DoS Slowhttptest.

Dataset yang digunakan terdiri dari beberapa file yang merepresentasikan data trafik jaringan pada beberapa hari pengamatan. Setiap dataset memiliki jumlah fitur yang sama, yaitu sebanyak 79 fitur yang merepresentasikan berbagai karakteristik lalu lintas jaringan. Berdasarkan hasil eksplorasi data yang dilakukan pada tahap awal penelitian, total jumlah data yang digunakan dalam penelitian ini adalah sebanyak 202.535 sampel data dengan 79 fitur. Rincian jumlah data pada setiap dataset yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Jumlah Data pada Setiap Dataset

Dataset	Jumlah Data	Jumlah Fitur
Friday	35.030	79
Monday	33.709	79
Tuesday	56.131	79
Wednesday	77.665	79
Total	202.535	79

Selain itu, dataset juga memiliki beberapa jenis label yang merepresentasikan aktivitas normal maupun serangan jaringan. Distribusi label secara keseluruhan pada dataset dapat dilihat pada Tabel 2.

Tabel 2. Distribusi Label Dataset

Label	Jumlah Data
BENIGN	169.960
DDoS	12.370
FTP-Patator	6.115
DoS Slowloris	5.796
DoS Slowhttptest	5.499
DoS Hulk	2.791

3.3 *Preprocessing*

Tahap preprocessing merupakan proses persiapan data sebelum dilakukan proses pelatihan model machine learning. Tujuan dari tahap ini adalah untuk membersihkan data, memastikan format data sesuai, serta meningkatkan kualitas data sehingga dapat digunakan secara optimal dalam proses klasifikasi. Pada penelitian ini beberapa tahapan preprocessing dilakukan sebelum dataset digunakan pada model K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM).

3.4 Pembagian Data (*Train & Test*)

Setelah proses *preprocessing* data selesai dilakukan, langkah selanjutnya adalah melakukan pembagian dataset menjadi data pelatihan (*training*) dan data pengujian (*testing*). Pembagian data ini bertujuan untuk melatih model *machine learning* menggunakan sebagian data serta menguji kemampuan model dalam melakukan klasifikasi terhadap data yang belum pernah digunakan pada proses pelatihan.

Pada penelitian ini, pembagian data dilakukan menggunakan metode *train-test split* yang tersedia pada pustaka *Scikit-learn*. Dataset dibagi dengan rasio 80% untuk data training dan 20% untuk data testing. Data *training* digunakan untuk membangun model klasifikasi, sedangkan data *testing* digunakan untuk mengevaluasi kinerja model yang telah dilatih.

Selain itu, proses pembagian data menggunakan teknik *stratified sampling* dengan parameter *stratify = y*. Teknik ini bertujuan untuk menjaga proporsi distribusi label pada data *training* dan data *testing* agar tetap seimbang serta merepresentasikan distribusi kelas pada dataset asli.

Berdasarkan hasil pembagian data yang dilakukan pada dataset penelitian, diperoleh 135.500 data training dan 33.876 data testing dengan jumlah fitur sebanyak 78 fitur pada masing-masing dataset. Distribusi label pada data *training* dan data *testing* juga tetap konsisten, yaitu sekitar 80,78% data normal dan 19,22% data serangan. Rincian pembagian data *training* dan *testing* dapat dilihat pada Tabel 3, sedangkan proporsi distribusi label pada dataset penelitian dapat dilihat pada Tabel 4.

Tabel 3. Pembagian Data *Training* dan *Testing*

Dataset	Jumlah Data	Jumlah Fitur
Data Training	135.500	78
Data Testing	33.876	78
Total	169.376	78

Tabel 4. Proporsi Distribusi Label

Label	Data Training	Data Testing
Normal (0)	80,78%	80,78%
Serangan (1)	19,22%	19,22%

Pembagian data ini dilakukan sebelum proses pelatihan model *machine learning*, sehingga model K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) dapat dilatih menggunakan data *training* dan kemudian diuji menggunakan data *testing* untuk mengetahui kinerja model dalam mendeteksi serangan jaringan.

3.5 Penerapan Model *Machine Learning* (*Baseline*)

Pada tahap ini dilakukan penerapan model machine learning sebagai metode klasifikasi untuk mendeteksi aktivitas normal dan serangan jaringan pada dataset yang telah diproses sebelumnya. Model yang digunakan dalam penelitian ini adalah K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM). Kedua model tersebut diterapkan sebagai model baseline untuk melakukan proses pembelajaran terhadap data training dan kemudian diuji menggunakan data testing yang telah dipisahkan pada tahap sebelumnya. Proses ini bertujuan untuk mengetahui kemampuan masing-masing algoritma dalam melakukan klasifikasi terhadap data jaringan berdasarkan fitur yang tersedia.

3.6 Evaluasi Model

Tahap evaluasi model dilakukan untuk mengukur kinerja algoritma machine learning dalam melakukan klasifikasi data jaringan. Pada penelitian ini evaluasi dilakukan terhadap model K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) menggunakan data testing yang telah dipisahkan pada tahap sebelumnya. Hasil prediksi dari model dibandingkan dengan label sebenarnya pada data testing untuk mengetahui tingkat keberhasilan model dalam mendeteksi aktivitas normal dan serangan jaringan.

Evaluasi model pada penelitian ini menggunakan beberapa metrik evaluasi klasifikasi yang umum digunakan dalam penelitian machine learning, yaitu Accuracy, Precision, Recall, dan F1-Score. Selain itu digunakan juga Confusion Matrix, ROC Curve, serta Precision-Recall Curve untuk menganalisis performa model secara lebih komprehensif.

IV. PENGUMPULAN DATA DAN HASIL

4.1 Pengumpulan Data

Pada tahap ini dilakukan proses pengumpulan dan pemuatan dataset yang digunakan dalam penelitian. Dataset yang digunakan merupakan dataset CICIDS2017 yang diperoleh dari Canadian Institute for Cybersecurity (CIC), University of New Brunswick. Dataset ini berisi data lalu lintas jaringan yang terdiri dari aktivitas normal (*benign*) serta berbagai jenis serangan jaringan seperti DDoS, FTP-Patator, DoS Slowloris, DoS Slowhttptest, dan DoS Hulk.

Dataset yang digunakan pada penelitian ini terdiri dari beberapa file yang merepresentasikan data lalu lintas jaringan pada beberapa hari pengamatan, yaitu Monday, Tuesday, Wednesday, dan Friday. Proses awal dilakukan dengan mengecek file dataset yang tersedia pada direktori penyimpanan program untuk memastikan file dataset telah tersedia dan dapat diakses oleh sistem.

Setelah memastikan dataset tersedia pada direktori penyimpanan, langkah selanjutnya adalah melakukan proses pemuatan dataset menggunakan pustaka Pandas dengan fungsi read_csv() untuk membaca setiap file dataset yang digunakan dalam penelitian. Proses pemuatan dataset ini dilakukan untuk mengimpor data dari file CSV ke dalam lingkungan pemrograman sehingga dapat dilakukan proses analisis lebih lanjut.

Setelah dataset berhasil dimuat, dilakukan pemeriksaan ukuran dataset untuk mengetahui jumlah data dan jumlah fitur yang terdapat pada masing-masing dataset. Informasi ini penting untuk mengetahui skala dataset yang digunakan dalam penelitian serta memastikan bahwa data telah berhasil dimuat dengan benar.

Selanjutnya dilakukan pemeriksaan jumlah fitur yang terdapat pada masing-masing dataset untuk memastikan bahwa seluruh dataset memiliki struktur fitur yang sama. Hal ini penting agar dataset dapat digabungkan dan dianalisis secara konsisten dalam proses selanjutnya.

Untuk mengetahui variasi kelas yang terdapat pada dataset, dilakukan pemeriksaan jumlah label pada masing-masing dataset. Proses ini bertujuan untuk mengidentifikasi jenis aktivitas jaringan yang terdapat pada setiap file dataset, baik aktivitas normal maupun berbagai jenis serangan jaringan.

Selain itu dilakukan juga analisis statistik deskriptif terhadap dataset untuk memperoleh gambaran umum mengenai distribusi nilai pada setiap fitur. Statistik deskriptif memberikan informasi mengenai nilai mean, standar deviasi, nilai minimum, nilai maksimum, serta nilai kuartil dari setiap fitur dataset.

Tahap terakhir pada proses pengumpulan data adalah melakukan pemeriksaan distribusi label secara keseluruhan pada dataset yang digunakan dalam penelitian. Pemeriksaan ini bertujuan untuk mengetahui jumlah data normal dan data serangan yang terdapat pada dataset.

4.2 Hasil Preprocessing

Tahap preprocessing merupakan proses persiapan data sebelum digunakan pada proses pelatihan model machine learning. Pada tahap ini dilakukan beberapa proses pembersihan dan transformasi data untuk memastikan dataset berada dalam kondisi yang baik dan dapat digunakan secara optimal dalam proses klasifikasi. Proses preprocessing yang dilakukan pada penelitian ini meliputi pembersihan dan seleksi fitur, transformasi label, pemisahan fitur dan target, penanganan data tidak valid, serta normalisasi data. Tahapan ini bertujuan untuk meningkatkan kualitas dataset sehingga dapat mendukung proses pembelajaran model K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) secara lebih efektif.

4.3 Normalisasi Data

Setelah proses pembersihan data dan penanganan nilai tidak valid selesai dilakukan, tahap selanjutnya adalah melakukan normalisasi data. Normalisasi data bertujuan untuk menyamakan skala nilai pada setiap fitur sehingga tidak terdapat perbedaan rentang nilai yang terlalu besar antar fitur. Proses ini sangat penting terutama pada algoritma K-Nearest Neighbor (KNN) yang berbasis perhitungan jarak serta Support Vector Machine (SVM) yang sensitif terhadap skala data.

Pada penelitian ini normalisasi data dilakukan menggunakan metode StandardScaler dari pustaka *Scikit-learn*. Metode ini bekerja dengan melakukan standarisasi nilai fitur sehingga memiliki nilai rata-rata (*mean*) sebesar 0 dan standar deviasi sebesar 1. Sebelum proses normalisasi dilakukan, terlebih dahulu dilakukan pembatasan nilai ekstrem (clipping) untuk menghindari nilai yang terlalu besar atau terlalu kecil yang dapat mempengaruhi proses standarisasi.

Setelah proses pembatasan nilai ekstrem selesai dilakukan, tahap berikutnya adalah melakukan normalisasi fitur menggunakan StandardScaler. Proses ini menghasilkan dataset dengan skala fitur yang lebih seragam sehingga dapat meningkatkan performa model machine learning dalam proses pelatihan dan klasifikasi. Selain itu juga dilakukan pengecekan untuk memastikan tidak terdapat nilai NaN maupun Infinite pada dataset setelah proses normalisasi.

Setelah model diinisialisasi, tahap berikutnya adalah melakukan pelatihan model (training) menggunakan data training yang telah diperoleh dari proses pembagian dataset sebelumnya. Pada tahap ini model mempelajari pola hubungan antara fitur jaringan dan label kelas dari data training. Berdasarkan hasil proses pelatihan model, waktu komputasi yang dibutuhkan untuk melatih model KNN adalah sekitar 0,03 detik.

Setelah proses pelatihan selesai, model kemudian digunakan untuk melakukan prediksi terhadap data testing yang sebelumnya tidak digunakan dalam proses pelatihan model. Proses ini bertujuan untuk menguji kemampuan model dalam melakukan klasifikasi terhadap data baru. Berdasarkan hasil pengujian tersebut, waktu komputasi yang dibutuhkan untuk melakukan prediksi adalah sekitar 61,71 detik.

Untuk mengetahui performa model KNN dalam melakukan klasifikasi, dilakukan proses evaluasi menggunakan beberapa metrik evaluasi yaitu accuracy, precision, recall, dan F1-score. Berdasarkan hasil evaluasi yang diperoleh, model KNN menghasilkan accuracy sebesar 0,9989, precision sebesar 0,9975, recall sebesar 0,9969, serta F1-score sebesar 0,9972.

Hasil evaluasi ini menunjukkan bahwa model KNN mampu melakukan klasifikasi aktivitas jaringan dengan tingkat akurasi yang sangat tinggi.

Selain menggunakan metrik evaluasi, performa model juga dianalisis menggunakan confusion matrix untuk mengetahui jumlah prediksi yang benar dan salah pada setiap kelas. Berdasarkan hasil confusion matrix diperoleh bahwa model berhasil mengklasifikasikan 27.348 data normal dengan benar, serta 6.492 data serangan dengan benar. Sementara itu terdapat 16 data normal yang salah diklasifikasikan sebagai serangan, dan 20 data serangan yang salah diklasifikasikan sebagai normal.

Selanjutnya dilakukan analisis menggunakan ROC Curve (Receiver Operating Characteristic) untuk melihat kemampuan model dalam membedakan antara kelas normal dan kelas serangan. Berdasarkan hasil pengujian diperoleh nilai Area Under Curve (AUC) sebesar 0,9992, yang menunjukkan bahwa model KNN memiliki kemampuan klasifikasi yang sangat baik dalam membedakan kedua kelas tersebut.

Selain itu dilakukan juga analisis menggunakan Precision-Recall Curve untuk melihat hubungan antara nilai precision dan recall dalam proses klasifikasi. Kurva ini digunakan untuk mengevaluasi performa model terutama pada dataset yang memiliki distribusi kelas yang tidak seimbang.

Untuk mengetahui pengaruh parameter K terhadap performa model, dilakukan pengujian dengan beberapa nilai K yaitu 3, 5, 7, 9, dan 11. Proses ini bertujuan untuk melihat perubahan performa model berdasarkan jumlah tetangga yang digunakan dalam proses klasifikasi.

Hasil pengujian tersebut kemudian divisualisasikan dalam bentuk grafik untuk melihat hubungan antara nilai K dengan accuracy yang dihasilkan oleh model. Grafik ini menunjukkan bahwa nilai K yang lebih kecil cenderung menghasilkan akurasi yang sedikit lebih tinggi dibandingkan nilai K yang lebih besar.

Selain accuracy, dilakukan juga analisis hubungan antara nilai K dengan F1-score untuk melihat keseimbangan antara precision dan recall yang dihasilkan oleh model. Grafik ini menunjukkan bahwa performa model relatif stabil pada beberapa nilai K yang diuji.

Terakhir dilakukan pengukuran waktu komputasi model untuk mengetahui efisiensi model dalam proses pelatihan dan prediksi. Berdasarkan hasil pengujian diperoleh bahwa waktu pelatihan model KNN adalah sekitar 0,03 detik, sedangkan waktu yang dibutuhkan untuk melakukan proses prediksi adalah sekitar 61,71 detik.

V. PEMBAHASAN

5.1 *Dataset*

Pada penelitian ini dataset yang digunakan adalah CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System Dataset) yang dikembangkan oleh *Canadian Institute for Cybersecurity (CIC), University of New Brunswick*. Dataset ini merupakan salah satu dataset yang banyak digunakan dalam penelitian Intrusion Detection System (IDS) karena memiliki karakteristik data lalu lintas jaringan yang realistis dan mencerminkan kondisi jaringan pada lingkungan nyata.

Dataset CICIDS2017 berisi berbagai jenis aktivitas jaringan yang terdiri dari aktivitas normal (benign) serta beberapa jenis serangan jaringan seperti DDoS, FTP-Patator, DoS Slowloris, DoS Slowhttptest, dan DoS Hulk. Keberagaman jenis serangan pada dataset ini memungkinkan model machine learning untuk mempelajari pola aktivitas jaringan yang berbeda antara lalu lintas normal dan aktivitas serangan, sehingga model dapat melakukan klasifikasi secara lebih akurat.

Pada penelitian ini dataset yang digunakan terdiri dari beberapa file data yang merepresentasikan lalu lintas jaringan pada beberapa hari pengamatan, yaitu Monday, Tuesday, Wednesday, dan Friday. Setiap file dataset memiliki sejumlah fitur yang menggambarkan karakteristik koneksi jaringan seperti durasi aliran data (*flow duration*), jumlah paket yang dikirim, ukuran paket, serta berbagai statistik lalu lintas jaringan lainnya.

Berdasarkan hasil penggabungan dataset yang dilakukan pada tahap pengumpulan data, diperoleh total 202.535 data sampel dengan 79 fitur yang digunakan sebagai atribut dalam proses analisis. Jumlah data yang relatif besar ini memberikan keuntungan dalam proses pelatihan model machine learning karena model dapat mempelajari pola data secara lebih baik dan menghasilkan performa klasifikasi yang lebih stabil.

5.2 *Preprocessing*

Tahap preprocessing dilakukan untuk memastikan dataset berada dalam kondisi yang siap digunakan pada proses pembelajaran model machine learning. Dataset mentah yang diperoleh dari CICIDS2017 masih mengandung beberapa atribut yang tidak relevan, perbedaan format data, serta kemungkinan nilai yang tidak valid. Oleh karena itu dilakukan beberapa tahapan preprocessing yang meliputi pembersihan dan seleksi fitur, transformasi label, pemisahan fitur dan target, penanganan data tidak valid, serta normalisasi data. Tahapan ini bertujuan untuk meningkatkan kualitas data sehingga model dapat mempelajari pola data secara lebih efektif.

5.3 *Pembagian Data*

Setelah proses preprocessing selesai dilakukan, tahap berikutnya adalah melakukan pembagian dataset menjadi data pelatihan (training set) dan data pengujian (testing set). Pembagian data ini bertujuan untuk melatih model machine learning menggunakan sebagian data dan mengevaluasi kemampuan model dalam melakukan klasifikasi terhadap data yang tidak digunakan selama proses pelatihan. Dengan demikian, performa model dapat diukur secara lebih objektif dan mampu menggambarkan kemampuan generalisasi model terhadap data baru.

Pada penelitian ini proses pembagian data dilakukan menggunakan metode train-test split dengan rasio 80% data training dan 20% data testing. Rasio ini dipilih karena memberikan keseimbangan antara jumlah data yang digunakan untuk melatih model dan jumlah data yang digunakan untuk menguji performa model. Jika jumlah data training terlalu sedikit, model tidak dapat mempelajari pola data dengan baik. Sebaliknya, jika jumlah data testing terlalu sedikit, hasil evaluasi model menjadi kurang representatif. Oleh karena itu rasio 80:20 banyak digunakan dalam penelitian machine learning karena mampu memberikan hasil evaluasi yang stabil.

Selain itu, proses pembagian data menggunakan teknik stratified sampling dengan parameter stratify=y. Teknik ini bertujuan untuk mempertahankan proporsi distribusi label antara data training dan data testing agar tetap sama dengan distribusi label pada dataset asli. Hal ini penting karena dataset yang digunakan memiliki distribusi kelas yang tidak seimbang, dimana jumlah data normal lebih besar dibandingkan data serangan. Tanpa stratified sampling, pembagian data secara acak dapat menyebabkan ketidakseimbangan distribusi kelas pada data training maupun data testing, yang pada akhirnya dapat mempengaruhi performa model dalam mendeteksi serangan jaringan. Hasil pembagian dataset pada penelitian ini dapat dilihat pada Tabel 5.

Tabel 5. Pembagian Dataset Training dan Testing

Jenis Data	Jumlah Data	Jumlah Fitur
Data Training	135.500	78
Data Testing	33.876	78
Total	169.376	78

Berdasarkan tabel tersebut dapat dilihat bahwa sebagian besar data digunakan sebagai data training untuk melatih model machine learning dalam mempelajari pola aktivitas jaringan. Sementara itu sebagian data lainnya digunakan sebagai data testing untuk mengevaluasi kemampuan model dalam melakukan klasifikasi terhadap data yang belum pernah dilihat sebelumnya.

Selain jumlah data, distribusi label pada data training dan data testing juga dianalisis untuk memastikan bahwa proporsi kelas tetap terjaga setelah proses pembagian data. Hasil distribusi label pada data training dan data testing dapat dilihat pada Tabel 6.

Tabel 6. Distribusi Label pada Data Training dan Testing

Label	Data Training	Data Testing
Normal (0)	80,78%	80,78%
Serangan (1)	19,22%	19,22%

Berdasarkan tabel tersebut dapat dilihat bahwa distribusi label pada data training dan data testing memiliki proporsi yang sama. Hal ini menunjukkan bahwa penggunaan metode stratified sampling berhasil mempertahankan distribusi kelas pada dataset sehingga model machine learning dapat dilatih dan diuji menggunakan data yang memiliki karakteristik distribusi yang serupa.

Dengan pembagian data yang seimbang tersebut, model machine learning yang digunakan dalam penelitian ini dapat mempelajari pola aktivitas jaringan dengan lebih baik dan menghasilkan evaluasi performa yang lebih akurat. Tahap pembagian data ini menjadi sangat penting karena kualitas pembagian dataset secara langsung mempengaruhi kemampuan model dalam melakukan klasifikasi aktivitas jaringan pada tahap pengujian selanjutnya.

5.4 *Model KNN*

Pada penelitian ini metode K-Nearest Neighbor (KNN) digunakan sebagai salah satu algoritma klasifikasi untuk mendeteksi aktivitas jaringan normal dan serangan jaringan. KNN merupakan algoritma berbasis jarak yang bekerja dengan menentukan kelas suatu data uji berdasarkan mayoritas kelas dari K tetangga terdekat pada data latih. Prinsip dasar KNN adalah bahwa data yang memiliki karakteristik yang mirip cenderung berada pada kelas yang sama. Oleh karena itu kualitas hasil klasifikasi KNN sangat dipengaruhi oleh kedekatan jarak antar data, distribusi data pada ruang fitur, serta jumlah tetangga yang digunakan.

Pada penelitian ini nilai parameter K ditetapkan sebesar 5 pada proses klasifikasi awal. Nilai ini dipilih karena mampu memberikan keseimbangan antara sensitivitas model terhadap pola data dan stabilitas hasil klasifikasi. Jika nilai K terlalu kecil, model cenderung terlalu sensitif terhadap variasi data sehingga dapat menyebabkan overfitting. Sebaliknya jika nilai K terlalu besar, model akan mempertimbangkan terlalu banyak tetangga sehingga batas klasifikasi menjadi kurang jelas. Oleh karena itu pemilihan nilai K yang tepat sangat mempengaruhi performa model KNN dalam melakukan klasifikasi aktivitas jaringan.

Berdasarkan hasil pengujian yang telah dilakukan, model KNN menghasilkan performa klasifikasi yang sangat tinggi dengan nilai accuracy sebesar 0,9989, precision sebesar 0,9975, recall sebesar 0,9969, dan F1-score sebesar 0,9972. Hasil evaluasi tersebut dapat dilihat pada Tabel 7.

Tabel 7. Hasil Evaluasi Model KNN

Metrik Evaluasi	Nilai
Accuracy	0,9989
Precision	0,9975
Recall	0,9969
F1-score	0,9972
AUC	0,9992

Selain metrik evaluasi, performa model juga dapat dianalisis menggunakan confusion matrix untuk melihat jumlah prediksi yang benar dan salah pada setiap kelas. Hasil confusion matrix model KNN dapat dilihat pada Tabel 8.

Tabel 8. Confusion Matrix Model KNN

Actual / Predicted	Normal (0)	Serangan (1)
Normal (0)	27.348	16
Serangan (1)	20	6.492

Berdasarkan tabel tersebut dapat dilihat bahwa model KNN mampu mengklasifikasikan sebagian besar data dengan benar. Jumlah kesalahan klasifikasi pada data normal maupun data serangan relatif sangat kecil dibandingkan jumlah data yang berhasil diklasifikasikan dengan benar. Kesalahan klasifikasi yang rendah ini menunjukkan bahwa model KNN memiliki kemampuan yang sangat baik dalam membedakan aktivitas jaringan normal dan aktivitas serangan.

5.5 *Model SVM*

Pada penelitian ini metode Support Vector Machine (SVM) digunakan sebagai algoritma klasifikasi untuk membedakan aktivitas jaringan normal dan aktivitas serangan. SVM merupakan algoritma machine learning yang bekerja dengan mencari hyperplane optimal yang mampu memisahkan dua kelas data secara maksimal pada ruang fitur. Hyperplane tersebut ditentukan berdasarkan titik-titik data yang berada paling dekat dengan batas pemisah yang disebut sebagai support vectors. Keunggulan metode SVM terletak pada kemampuannya dalam membangun batas klasifikasi yang kuat pada data berdimensi tinggi.

Pada penelitian ini digunakan algoritma Linear Support Vector Classifier (LinearSVC) dengan parameter C = 1.0 dan max_iter = 5000. Parameter C berfungsi sebagai parameter regularisasi yang mengontrol keseimbangan antara memaksimalkan margin pemisah dan meminimalkan kesalahan klasifikasi pada data pelatihan.

Nilai parameter C yang lebih besar cenderung membuat model berusaha mengklasifikasikan data pelatihan secara lebih akurat, namun dapat meningkatkan risiko overfitting. Sebaliknya nilai C yang lebih kecil menghasilkan margin yang lebih besar tetapi memungkinkan lebih banyak kesalahan klasifikasi pada data pelatihan.

Berdasarkan hasil pengujian yang dilakukan, model SVM menghasilkan performa klasifikasi yang cukup baik dengan nilai accuracy sebesar 0,9756, precision sebesar 0,9575, recall sebesar 0,9134, serta F1-score sebesar 0,9349. Hasil evaluasi model SVM dapat dilihat pada Tabel 9.

Tabel 9. Hasil Evaluasi Model SVM

Metrik Evaluasi	Nilai
Accuracy	0,9756
Precision	0,9575
Recall	0,9134
F1-score	0,9349
AUC	0,9975

5.6 Perbandingan Model

Pada penelitian ini dilakukan perbandingan kinerja antara dua metode klasifikasi yang digunakan, yaitu K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM). Tujuan dari perbandingan ini adalah untuk mengetahui metode yang memiliki performa terbaik dalam mendeteksi aktivitas jaringan normal dan serangan jaringan berdasarkan dataset CICIDS2017 yang digunakan dalam penelitian. Perbandingan dilakukan menggunakan beberapa metrik evaluasi yaitu accuracy, precision, recall, F1-score, serta Area Under Curve (AUC).

Berdasarkan hasil pengujian yang telah dilakukan pada kedua model, diperoleh hasil evaluasi yang dapat dilihat pada Tabel 10.

.

Tabel 10. Perbandingan Kinerja Model KNN dan SVM

Model	Accuracy	Precision	Recall	F1-score	AUC
KNN	0,9989	0,9975	0,9969	0,9972	0,9992
SVM	0,9756	0,9575	0,9134	0,9349	0,9975

Berdasarkan tabel tersebut dapat dilihat bahwa metode KNN memiliki performa yang lebih baik dibandingkan metode SVM pada seluruh metrik evaluasi yang digunakan. Perbedaan performa yang cukup signifikan terlihat pada nilai recall dan F1-score, yang menunjukkan bahwa model KNN memiliki kemampuan yang lebih baik dalam mendeteksi aktivitas serangan jaringan.

VI. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan mengenai analisis metode K-Nearest Neighbor (KNN) dan Support Vector Machine (SVM) pada klasifikasi log serangan jaringan menggunakan dataset CICIDS2017, maka dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Metode K-Nearest Neighbor (KNN) mampu melakukan klasifikasi aktivitas jaringan menjadi kategori normal dan serangan dengan performa yang sangat baik. Berdasarkan hasil pengujian yang dilakukan, model KNN menghasilkan nilai accuracy sebesar 0,9989, precision sebesar 0,9975, recall sebesar 0,9969, dan F1-score sebesar 0,9972, yang menunjukkan bahwa metode KNN memiliki kemampuan yang sangat tinggi dalam mendeteksi aktivitas serangan jaringan pada dataset CICIDS2017.
2. Metode Support Vector Machine (SVM) juga mampu melakukan klasifikasi aktivitas jaringan dengan performa yang cukup baik. Hasil evaluasi model menunjukkan nilai accuracy sebesar 0,9756, precision sebesar 0,9575, recall sebesar 0,9134, dan F1-score sebesar 0,9349, yang menunjukkan bahwa metode SVM masih memiliki kemampuan yang baik dalam mendeteksi aktivitas jaringan normal dan serangan, meskipun performanya masih berada di bawah metode KNN pada dataset yang digunakan.
3. Berdasarkan hasil perbandingan performa kedua metode menggunakan metrik evaluasi accuracy, precision, recall, dan F1-score, metode K-Nearest Neighbor (KNN) menunjukkan performa yang lebih baik dibandingkan metode Support Vector Machine (SVM). Hal ini terlihat dari nilai evaluasi yang lebih tinggi serta jumlah kesalahan klasifikasi yang lebih kecil pada model KNN.
4. Hasil analisis klasifikasi yang diperoleh menunjukkan bahwa metode K-Nearest Neighbor (KNN) lebih efektif digunakan dalam proses deteksi serangan jaringan pada dataset CICIDS2017. Dengan tingkat akurasi dan kemampuan deteksi yang tinggi, metode ini dapat dijadikan sebagai rekomendasi metode machine learning yang dapat digunakan dalam pengembangan sistem keamanan jaringan berbasis Intrusion Detection System (IDS) untuk membantu mengidentifikasi aktivitas serangan jaringan secara lebih akurat.

6.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya maupun penerapan hasil penelitian dalam bidang keamanan jaringan, yaitu sebagai berikut:

1. Penelitian selanjutnya disarankan untuk menggunakan dataset yang lebih beragam atau menggabungkan beberapa dataset intrusion detection lainnya

seperti CICIDS2018, UNSW-NB15, atau NSL-KDD, sehingga model yang dihasilkan dapat diuji pada berbagai jenis serangan jaringan dan memiliki kemampuan generalisasi yang lebih baik.

2.

3. Penelitian berikutnya dapat mengembangkan metode klasifikasi dengan menggunakan algoritma machine learning lain atau metode ensemble learning, seperti Random Forest, Gradient Boosting, atau Deep Learning, sehingga dapat diperoleh perbandingan performa yang lebih luas dalam mendeteksi serangan jaringan.

Penelitian selanjutnya juga dapat melakukan optimasi parameter dan seleksi fitur yang lebih mendalam, misalnya menggunakan teknik Genetic Algorithm, Particle Swarm Optimization, atau metode feature selection lainnya, sehingga dapat meningkatkan performa model serta mengurangi kompleksitas komputasi dalam proses klasifikasi serangan jaringan.

DAFTAR PUSTAKA

[1] F. Fachri, "Optimasi Keamanan Web Server Terhadap Serangan Brute-Force Menggunakan Penetration Testing," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 1, pp. 51–58, 2023, doi: 10.25126/jtiik.20231015872.

[2] A. Hermawan, T. Hartati, and Y. A. Wijaya, "Analisa Keamanan Data Melalui Website Zahra Software Menggunakan Metode Keamanan Informasi CIA Triad," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 7, no. 3, pp. 125–130, 2022, doi: 10.30591/jpit.v7i3.3428.

[3] S. Rahayu, Y. MZ, J. E. Bororing, and R. Hadiyat, "Implementasi Metode K-Nearest Neighbor (K-NN) untuk Analisis Sentimen Kepuasan Pengguna Aplikasi Teknologi Finansial FLIP," *Edumatic: Jurnal Pendidikan Informatika*, vol. 6, no. 1, pp. 98–106, 2022, doi: 10.29408/edumatic.v6i1.5433.

[4] T. Setiyorini and R. T. Asmono, "Penerapan Metode K-Nearest Neighbor Dan Gini Index Pada Klasifikasi Kinerja Siswa," *Jurnal Techno Nusa Mandiri*, vol. 16, no. 2, pp. 121–126, 2019, doi: 10.33480/techno.v16i2.747.

[5] L. V. Nguyen, Q. T. Vo, and T. H. Nguyen, "Adaptive KNN-Based Extended Collaborative Filtering Recommendation Services," *Big Data and Cognitive Computing*, vol. 7, no. 2, 2023, doi: 10.3390/bdcc7020106.

[6] O. A. Alkhudaydi, M. Krichen, and A. D. Alghamdi, "A Deep Learning Methodology for Predicting Cybersecurity Attacks on the Internet of Things," *Information (Switzerland)*, vol. 14, no. 10, 2023, doi: 10.3390/info14100550.

[7] L. M. Pattnaik, P. K. Swain, S. Satpathy, and A. N. Panda, "Cloud DDoS Attack Detection Model with Data Fusion & Machine Learning Classifiers," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 10, no. 6, pp. 1–9, 2023, doi: 10.4108/eetsis.3936.

[8] İ. Avcı and M. Koca, "Cybersecurity Attack Detection Model , Using Machine Learning Techniques," vol. 20, no. 7, pp. 29–44, 2023.

[9] G. Kaur and P. Gupta, "Detection of Distributed Denial of Service Attacks for IoT-Based Healthcare Systems," vol. 30, no. 2, pp. 167–186, 2023, doi: 10.24423/cames.450.

[10] D. Shivaramakrishna and M. Nagaratna, "A novel hybrid cryptographic framework for secure data storage in cloud computing : Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control," *Alexandria Engineering Journal*, vol. 84, no. September, pp. 275–284, 2023, doi: 10.1016/j.aej.2023.10.054.

[11] A. Hossain and S. Islam, "Ensuring network security with a robust intrusion detection system using," *Array*, vol. 19, no. May, p. 100306, 2023, doi: 10.1016/j.array.2023.100306.

[12] K. F. Hasan, A. Akhter, M. A. Yousuf, F. Alharbi, and M. A. Moni, "A Dependable Hybrid Machine Learning Model for Network Intrusion Detection".

[13] G. D. E. C. Bertoli *et al.*, "An End-to-End Framework for Machine Learning-Based Network Intrusion Detection System," vol. 9, 2021, doi: 10.1109/ACCESS.2021.3101188.

[14] N. Ali, "Comparative study between (SVM) and (KNN) classifiers by using (PCA) to improve of intrusion detection system," vol. 1, no. June, pp. 22–33, 2022, doi: 10.52940/ijici.v1i1.4.

[15] O. Alghushairy, R. Alsini, Z. Alhassan, A. Yafoz, and X. Ma, "An Efficient Support Vector Machine Algorithm Based Network Outlier Detection System," vol. 12, no. February, 2024.