BAB 1

PENDAHULUAN

1.1 Latar Belakang

PT Edmasan Citra Telekomindo adalah penyedia layanan internet (*Internet Service Provider*) yang berlokasi di Pasir Pengaraian, Kabupaten Rokan Hulu. Sebagai salah satu penyedia layanan internet didaerah ini, perusahaan bertanggung jawab untuk menyediakan konektivitas yang andal dan aman bagi berbagai sektor, termasuk pemerintahan, bisnis, dan pengguna individu. Keamanan sistem informasi menjadi fondasi krusial bagi kelangsungan operasional organisasi seperti PT Edmasan Citra Telekomindo dan integritas privasi individu. Namun, seiring dengan pesatnya kemajuan teknologi, muncul tantangan baru dalam menjaga keamanan informasi, terutama berkembangnya jenis serangan yang mengintai jaringan komputer [51].

Meskipun PT Edmasan Citra Telekomindo telah menerapkan berbagai langkah keamanan, perusahaan masih menghadapi tantangan dalam mendeteksi dan menanggapi serangan siber yang semakin canggih dan beragam. Dengan meningkatnya volume lalu lintas data dan kompleksitas jaringan, tim keamanan perusahaan kesulitan untuk memantau dan menganalisis semua aktivitas jaringan secara manual. Hal ini menciptakan celah keamanan yang dapat dieksploitasi oleh penyerang, menempatkan data pelanggan dan integritas layanan perusahaan pada risiko yang signifikan. Oleh karena itu, diperlukan pendekatan yang lebih efektif dan

efisien dalam mendeteksi intrusi jaringan, yang dapat beradaptasi dengan cepat terhadap ancaman yang berkembang dan mengurangi beban kerja manual tim keamanan.

Pada bulan Januari tahun 2024, Indonesia menghadapi tantangan serius dengan menduduki peringkat pertama dalam jumlah anomali lalu lintas tertinggi. Total anomali mencapai 18.840.506, dengan puncak terjadi pada tanggal 29 Januari 2024 mencapai 703.019 anomali lalu lintas. Serangan tersebut melibatkan berbagai jenis serangan seperti *Malware, Trojan, Information Leak, exploit, DoS, APT*, dan lainnya, dengan jenis serangan yang paling dominan adalah *Malware*. Sektor administrasi Pemerintahan sendiri mengalami sebanyak 6 kasus peretasan [1].

Deteksi intrusi jaringan menjadi salah satu strategi pertahanan yang krusial dalam mengidentifikasi dan menanggapi serangan yang mengancam keamanan sistem informasi. Metode konvensional dalam deteksi intrusi seringkali terbatas pada penanda atau pola serangan yang sudah dikenal sebelumnya, meninggalkan celah bagi serangan yang tidak dikenali atau baru muncul [52].

Dalam konteks ini, teknik *machine learning* menawarkan pendekatan yang lebih adaptif dan efektif dalam mendeteksi intrusi jaringan. Dengan kemampuannya dalam memproses dan mempelajari pola dari data lalu lintas jaringan, algoritma *machine learning* seperti *Random Forest* mampu mengidentifikasi pola anomali yang menunjukkan potensi serangan [53], bahkan tanpa adanya pengetahuan sebelumnya tentang serangan tersebut.

Dataset KDD Cup dipilih untuk dijadikan data latih awal pada penelitian ini dikarenakan alasan keberagaman dan kompleksitasnya. *Dataset* ini terkenal karena mencakup berbagai jenis serangan yang beragam, termasuk serangan yang tidak biasa atau baru muncul. Selain itu, *dataset* ini telah banyak digunakan dalam penelitian deteksi intrusi jaringan sehingga memungkinkan hasil penelitian ini untuk dibandingkan dan dievaluasi secara lebih luas dengan penelitian sebelumnya [54]. Keterlibatan *KDD Cup* juga memungkinkan penelitian ini untuk mendapatkan wawasan yang lebih mendalam tentang permasalahan sebenarnya yang dihadapi dalam mendeteksi intrusi jaringan di dunia nyata.

Menurut penelitian oleh Mercury Fluorida Fibrianda dan Adhitya Bhawiyuga, algoritma *Naive Bayes* dan *Support Vector Machine* (*SVM*) memiliki keunggulan dalam mendeteksi serangan jaringan. *SVM,* khususnya, menunjukkan akurasi yang sangat tinggi, dengan nilai *precision* dan *recall* yang mendekati sempurna [21].

Berdasarkan penelitian oleh Tony Tan, Hendi Sama, Gautama Wijaya, dan Osei Enoch Aboagye, *SVM* dan *Artificial Neural Network* (*ANN*) efektif dalam mendeteksi serangan intrusi pada jaringan. Meskipun keduanya efektif, *SVM* terbukti lebih efisien dalam deteksi intrusi, menunjukkan akurasi yang tinggi dan waktu pelatihan yang lebih singkat [22].

Menurut penelitian oleh Farid Ridho dan Arya Aji Kusuma, *K-Means Clustering* dan *Bisect K-Means Clustering* digunakan dalam deteksi intrusi jaringan. Hasil evaluasi menunjukkan bahwa *K-Means Clustering* memiliki keunggulan

dibandingkan *Bisect K-Means Clustering* dalam hal kekuatan dan akurasi model yang dibentuk untuk mendeteksi aktivitas normal dan anomali [23].

Random Forest adalah algoritma *machine learning* yang termasuk dalam kategori *ensemble learning*, yang berarti ia menggabungkan hasil dari beberapa model *machine learning* dasar untuk meningkatkan kinerja keseluruhan. Keunggulan dari *Random Forest* antara lain adalah kemampuannya untuk menangani data yang besar dengan fitur yang banyak, toleransi terhadap *overfitting*, serta kemampuan untuk memberikan estimasi pentingnya fitur dalam model [20].

Random Forest dan metode konvensional berbasis aturan memiliki pendekatan yang berbeda dalam deteksi intrusi. *Random Forest*, yang merupakan algoritma machine learning berbasis *ensemble*, memanfaatkan banyak pohon keputusan untuk mengidentifikasi pola yang kompleks dalam data, sehingga mampu mendeteksi intrusi yang mungkin tidak terlihat oleh metode berbasis aturan. Metode berbasis aturan, di sisi lain, bergantung pada serangkaian aturan yang telah ditentukan sebelumnya untuk mengidentifikasi aktivitas mencurigakan. Hal ini membuatnya kurang adaptif terhadap ancaman baru atau yang belum dikenal, karena aturan harus secara manual diperbarui oleh para ahli keamanan. Sebaliknya, *Random Forest* dapat belajar dan beradaptasi dengan jenis serangan baru berdasarkan data pelatihan yang diperbarui, membuatnya lebih dinamis dan fleksibel dalam menghadapi ancaman keamanan yang terus berkembang.

Penelitian ini bertujuan untuk menyelidiki potensi penerapan teknik machine learning, terutama algoritma *Random Forest*, dalam meningkatkan deteksi intrusi

jaringan untuk keamanan sistem informasi. Dengan demikian, peneliti tertarik untuk mengangkat judul "Implementasi Metode *Random Forest* Dalam Deteksi Intrusi Jaringan".

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah utama dalam penelitian ini adalah bagaimana implementasi metode *random forest* dalam deteksi intrusi jaringan?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan mengimplementasikan metode *random forest* dalam deteksi intrusi jaringan dengan tujuan deteksi ancaman pada sistem informasi.

1.4 Batasan Masalah

Penelitian ini memiliki beberapa batasan, yaitu:

1. Penggunaan *dataset KDD Cup* 1999 sebagai sumber data latih awal. *KDD Cup* 1999 adalah hasil dari kompetisi *dataset* yang dikembangkan untuk keperluan deteksi anomali jaringan komputer, berisi data jaringan yang mencakup berbagai jenis serangan dan aktivitas normal, sehingga cocok sebagai acuan dasar dalam pengembangan sistem deteksi intrusi.
2. Sebanyak 125.000 baris data dari *dataset* akan digunakan untuk membentuk *dataset training* dan *testing*. Jumlah data yang digunakan telah disesuaikan dengan kebutuhan dan kapasitas komputasi yang tersedia.

3. Fokus pada penerapan algoritma *Random Forest* sebagai Teknik *machine learning* dalam mendeteksi intrusi.
4. Hasil dari penelitian ini merupakan sistem deteksi intrusi yang mampu mendeteksi jenis serangan DDoS dan *port scan.*
5. Penelitian ini hanya berfokus pada deteksi serangan, bukan pada pengatasan atau mitigasi serangan yang terdeteksi. Sistem yang dikembangkan tidak akan melakukan tindakan pencegahan atau perbaikan terhadap serangan yang teridentifikasi, melainkan hanya memberikan informasi mengenai adanya serangan.

1.5 Manfaat Penelitian

Penelitian ini memiliki manfaat yang penting, yaitu meningkatkan pemahaman tentang penggunaan teknik *machine learning* dalam deteksi intrusi jaringan untuk meningkatkan keamanan sistem informasi, memberikan kontribusi pada pengembangan metode deteksi intrusi yang lebih efektif dan adaptif, serta menyediakan landasan untuk penelitian lebih lanjut dalam bidang keamanan sistem informasi dan teknik *machine learning*.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari lima bagian utama, yakni:

BAB 1 PENDAHULUAN

Pada bagian ini, dijelaskan latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, dan sistematika penulisan.

BAB 2 LANDASAN TEORI

Bagian ini memuat teori-teori yang relevan dengan Penerapan Teknik *Machine Learning* Dalam Deteksi Intrusi Jaringan Untuk Meningkatkan Deteksi Ancaman Pada Sistem Informasi.

BAB 3 METODOLOGI PENELITIAN

Di bagian ini, akan diuraikan tahapan-tahapan dalam pengumpulan data, pemilihan algoritma, perumusan masalah, dan analisis yang digunakan dalam penelitian.

BAB 4 ANALISIS DAN HASIL PENELITIAN

Bagian ini akan membahas hasil dari penelitian yang telah dilakukan, termasuk analisis data dan evaluasi performa model.

BAB 5 IMPLEMENTASI DAN PENGUJIAN

Bab ini berisikan proses implementasi sistem deteksi intrusi *berbasis machine learning* dan hasil pengujian sistem. Dijelaskan pula metode pengujian yang digunakan dan analisis hasil pengujian untuk menilai kinerja dan stabilitas sistem dalam mendeteksi ancaman jaringan.

BAB 6 PENUTUP

Bab ini berisikan kesimpulan dari hasil penelitian yang telah dilakukan, serta saran-saran untuk pengembangan model *machine learning* untuk deteksi intrusi jaringan atau penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 *Machine learning*

Machine Learning adalah suatu pendekatan dalam bidang kecerdasan buatan yang memungkinkan komputer untuk belajar dari data, tanpa di-program secara eksplisit [2]. *Machine learning* juga sering digunakan untuk mengemulasi atau meniru cara manusia dalam menyelesaikan masalah atau melakukan otomatisasi [24]. Teknologi ini dikembangkan berdasarkan disiplin ilmu statistika, matematika, dan data mining. Dengan kata lain, mesin tidak hanya bisa bekerja secara otomatis, tapi juga bisa belajar agar performanya menjadi lebih baik dari waktu ke waktu.

Machine Learning merupakan teknologi yang mampu mempelajari data yang ada dan melakukan tugas-tugas tertentu sesuai dengan apa yang ia pelajari [3]. Tugas yang dapat dilakukan oleh *Machine Learning* sangat beragam, tergantung dari apa yang ia pelajari.

2.2 *Random Forest*

Random Forest adalah salah satu algoritma *machine learning* yang digunakan untuk melakukan klasifikasi dan regresi pada data. Algoritma ini bekerja dengan cara menggabungkan beberapa pohon keputusan (*decision tree*) yang dibuat secara acak untuk menghasilkan prediksi yang lebih akurat [4]. Setiap pohon dalam "hutan" (*forest*) ini digunakan untuk melakukan prediksi, dan hasil prediksi dari semua pohon digabungkan untuk mendapatkan hasil akhir.

Random Forest menggunakan teknik *ensemble learning* dengan menggabungkan beberapa pohon keputusan yang dibuat secara acak untuk meningkatkan akurasi klasifikasi pada data yang kompleks. Setiap pohon keputusan dalam *Random Forest* akan memilih fitur secara acak dan hanya menggunakan sebagian data untuk membuat keputusan. Kemudian, hasil dari setiap pohon keputusan akan digabungkan untuk menghasilkan prediksi akhir [5].

Berikut adalah rumus yang digunakan dalam *Random Forest* [25]:

a. Rumus untuk satu pohon keputusan *random forest*:

$$F(x) = \sum_{i=1}^{m} W_i h_i(x)$$

$F(x)$ adalah output dari pohon keputusan.

m = jumlah pohon dalam model *random forest*.

W_i = bobot atau prediksi dari pohon keputusan ke-i.

$h_i(x)$ = fungsi keputusan dari pohon ke-i untuk input x.

b. Rumus untuk output dari keseluruhan pohon keputusan *random forest*:

$$F(x) = \frac{1}{M} \sum_{i=1}^{m} F_i(x)$$

$F(x)$ = output dari *random forest*.

M = jumlah total pohon dalam *random forest*.

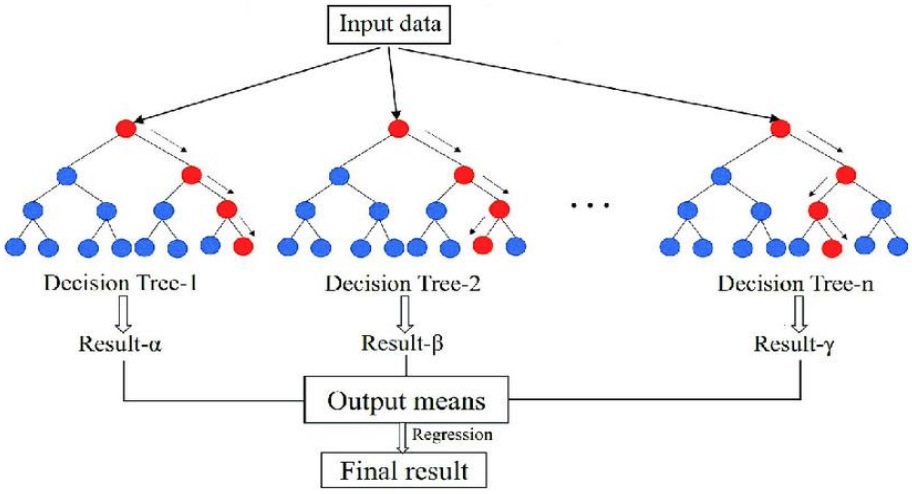
$F_i(x)$ = output dari pohon keputusan ke-i untuk input x.

Jadi, untuk mendapatkan F(x), kita mengambil rata-rata dari output semua pohon keputusan dalam Random Forest untuk input x. Sebagai contoh, misalkan kita memiliki Random Forest dengan 100 pohon keputusan (M=100). Untuk input data x:

- Pohon keputusan 1 memberikan output F1(x) = 0 (bukan intrusi)
- Pohon keputusan 2 memberikan output F2(x) = 1 (intrusi)
- Pohon keputusan 100 memberikan output F100(x) = 0 (bukan intrusi)

Maka, F(x) = (1/100) * (0 + 1 + ... + 0) = 0.37 (misalnya)

Nilai F(x) = 0.37 ini kemudian dapat diinterpretasikan, misalnya dengan menggunakan ambang batas tertentu, apakah input x diklasifikasikan sebagai intrusi atau bukan. Jadi, F(x) merupakan prediksi akhir gabungan dari semua pohon keputusan dalam model Random Forest untuk suatu input data x.



Gambar 2. 1 Ilustrasi algoritma *random forest* (Sun et al., 2022)

2.3 *Data Mining*

Data mining adalah proses menemukan pola-pola yang bermanfaat dan menarik dari volume data yang besar. Proses ini menggabungkan teknik-teknik dari statistik, pembelajaran mesin, dan basis data untuk mengidentifikasi informasi yang belum diketahui sebelumnya namun memiliki nilai signifikan [55]. Penggunaan *data mining* meluas di berbagai bidang seperti pemasaran, deteksi kecurangan, analisis ilmiah, dan sistem informasi bisnis [56]. Proses ini umumnya terdiri dari serangkaian tahapan, termasuk pembersihan data, integrasi data, seleksi data, transformasi data, penambangan data, dan evaluasi pola

2.4 Sistem

Secara umum, sistem merujuk pada kumpulan objek, elemen, atau komponen yang memiliki makna yang beragam dan saling berinteraksi, berkolaborasi, serta memengaruhi satu sama lain dalam lingkungan yang kompleks. Sistem juga terkait dengan rencana atau tujuan yang sama, di mana berbagai elemen atau bagian saling terhubung untuk mencapai tujuan tersebut [6]. Dengan kata lain, sistem bisa dipandang sebagai satu kesatuan yang terdiri dari dua atau lebih komponen atau subsistem yang bekerja sama untuk mencapai suatu tujuan yang telah ditentukan.

Contohnya, dalam konteks organisasi atau perusahaan, sistem bisa merujuk pada jaringan kerja yang terdiri dari berbagai prosedur dan kegiatan yang saling berhubungan dan disatukan untuk mencapai tujuan tertentu [7]. Sistem ini tidak hanya terdiri dari komponen teknis atau fisik, tetapi juga mencakup aspek-aspek

seperti prosedur, kebijakan, dan interaksi manusia yang terlibat dalam operasionalnya [26]. Dengan demikian, sistem dapat dipandang sebagai struktur yang kompleks dan dinamis yang terus beradaptasi dengan lingkungannya untuk mencapai tujuan yang telah ditetapkan.

2.5 Intrusi

Intrusi merujuk pada segala aktivitas yang tidak sah atau tidak diinginkan yang dapat mengancam kerahasiaan, integritas, atau ketersediaan informasi dalam suatu sistem [27]. Berbagai macam serangan, seperti akses ilegal, pembajakan, dan aktivitas mencurigakan lainnya, dapat terjadi dalam suatu intrusi. Tujuan utama dari intrusi adalah untuk memperoleh akses yang tidak sah ke sumber daya sistem atau jaringan, mengubah data, atau menonaktifkan layanan [8].

Tujuan ilegal ini dapat mencakup upaya untuk mengumpulkan informasi tentang sistem internal, jaringan, sistem operasi, perangkat lunak, dan komponen lainnya dalam suatu sistem. Tindakan ini dilakukan oleh pihak-pihak yang tidak bertanggung jawab, baik dari eksternal maupun internal, dengan tujuan untuk kemudian menyalahgunakan atau menyerang sistem [28].

2.6 *Intrusion Detecton System* (IDS)

Sistem Deteksi Intrusi (*Intrusion Detection System* atau IDS) adalah suatu sistem yang melakukan pemantauan terhadap lalu lintas jaringan atau sistem untuk mengidentifikasi aktivitas yang mencurigakan dan memberikan laporan dalam bentuk peringatan. IDS memiliki peran yang sangat penting sebagai elemen pertahanan dasar dalam jaringan dan memiliki kemampuan untuk menanggulangi berbagai jenis

ancaman yang berbahaya, terutama yang terkait dengan intrusi atau akses tidak sah [9].

IDS dapat diimplementasikan dalam berbagai bentuk, termasuk sebagai aplikasi perangkat lunak yang diinstal langsung pada titik akhir sistem atau sebagai perangkat keras khusus yang terhubung secara langsung kedalam infrastruktur jaringan. Selain itu, beberapa solusi IDS juga tersedia sebagai layanan yang disediakan melalui cloud, yang memungkinkan pengguna untuk mengakses dan mengelola sistem deteksi intrusi tanpa perlu mengelola infrastruktur fisik secara langsung. Pendekatan berbasis *cloud* ini dapat mengurangi biaya dan kompleksitas implementasi IDS, serta memungkinkan pemantauan dan deteksi intrusi secara efisien dalam skala yang lebih besar [30].

Perlu ditekankan bahwa IDS hanya memiliki kemampuan untuk memberikan peringatan kepada pengguna, dan tidak memiliki kemampuan untuk mengambil tindakan aktif seperti menghapus atau memblokir ancaman [29]. Segala bentuk aktivitas yang dianggap berbahaya umumnya akan dilaporkan kepada *administrator* atau diteruskan ke dalam sistem *Security Information and Event Management* (SIEM) secara sentral. Kemudian, SIEM akan mengintegrasikan keluaran dari berbagai sumber informasi dan melakukan penyaringan terhadap setiap aktivitas yang dilaporkan.

2.7 Metode deteksi pada IDS

IDS menggunakan salah satu atau kedua dari dua metode deteksi ancaman utama: deteksi berbasis tanda tangan atau berbasis anomali. Deteksi berbasis tanda tangan menganalisis paket jaringan untuk mencari tanda tangan serangan serta karakteristik atau perilaku unik yang terkait dengan ancaman tertentu. IDS berbasis tanda tangan menggunakan basis data tanda tangan serangan untuk membandingkan paket jaringan [32]. Di sisi lain, deteksi berbasis anomali menggunakan teknik *machine learning* untuk mengembangkan dan terus memperbarui model aktivitas jaringan yang normal [31]. Model ini kemudian digunakan untuk membandingkan aktivitas jaringan aktual dan menandai segala penyimpangan yang mencurigakan. Dengan demikian, kedua metode ini menyediakan pendekatan yang komplementer dalam mendeteksi ancaman dan meningkatkan keamanan jaringan secara keseluruhan [10].

2.8 Keamanan Sistem Informasi

Keamanan sistem informasi merujuk pada upaya-upaya yang dilakukan untuk melindungi informasi yang disimpan, diproses, dan ditransmisikan dalam suatu sistem informasi. Ini mencakup perlindungan terhadap berbagai ancaman seperti akses tidak sah, manipulasi data, pengungkapan informasi sensitif, dan gangguan terhadap operasi sistem [33]. Sistem informasi merupakan komponen penting bagi sebuah instansi atau organisasi dalam menyajikan informasi untuk pengambilan keputusan. Peran teknologi komputer sangat dibutuhkan untuk mendukung sistem

informasi tersebut, di mana hampir semua organisasi, baik pemerintah maupun swasta, telah beralih menggunakan komputer untuk mengelola dan menyimpan data mereka [42]. Tujuan keamanan sistem informasi adalah untuk memastikan dan menjamin integritas, ketersediaan, dan kerahasiaan dari pengolahan informasi. Manajemen keamanan sistem informasi sebaiknya dimulai sejak tahap pembangunan sistem informasi itu sendiri, bukan hanya sebagai tambahan setelah sistem informasi selesai dibangun [34].

Keamanan sistem informasi juga mencakup upaya untuk mengidentifikasi, menganalisis, dan mengelola risiko yang terkait dengan penggunaan sistem informasi. Hal ini melibatkan pelaksanaan kontrol keamanan, pemantauan aktivitas sistem, dan pelatihan pengguna untuk meningkatkan kesadaran tentang praktik keamanan yang baik. Dengan menjaga keamanan sistem informasi, organisasi dapat melindungi aset informasi mereka, menjaga reputasi mereka, dan mematuhi peraturan dan perundang-undangan yang berlaku [35].

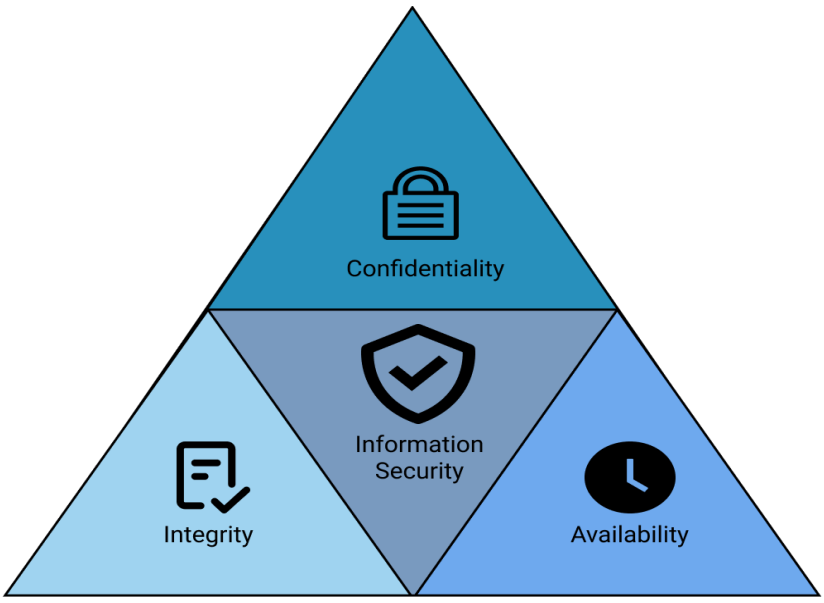
2.9 Tujuan Keamanan Sistem Informasi

Tujuan utama dari keamanan sistem informasi adalah untuk memastikan tiga aspek utama, yaitu kerahasiaan, integritas, dan ketersediaan informasi tetap aman dan terjaga [34]. Berikut merupakan pengertian dari tiga aspek utama tersebut:

1. Kerahasiaan: Menjamin bahwa informasi hanya dapat diakses oleh pihak yang berwenang, dan tidak dapat diakses oleh pihak yang tidak

berwenang. Ini melibatkan penggunaan mekanisme autentikasi dan otorisasi untuk mengontrol akses pengguna ke informasi.

2. Integritas: Memastikan bahwa informasi tidak diubah atau dimanipulasi oleh pihak yang tidak berwenang. Ini melibatkan penggunaan tanda tangan digital, enkripsi, dan mekanisme deteksi perubahan untuk memastikan bahwa data tetap utuh.
3. Ketersediaan: Memastikan bahwa informasi dapat diakses dan digunakan oleh pihak yang berwenang pada saat yang diperlukan. Ini melibatkan penerapan kebijakan pemulihan bencana, redundansi sistem, dan perlindungan terhadap serangan Denial of Service (DoS) [11].



Gambar 2. 2 Elemen-elemen keamanan informasi (Winarianto, 2022)

2.10 *Python*

Python adalah bahasa pemrograman yang diciptakan oleh Guido van Rossum. Pengembangan *Python* dimulai pada akhir tahun 1980-an dan dirilis pertama kali pada tahun 1991. Nama *Python* sendiri diambil dari serial televisi "*Monty Python's Flying Circus*", bukan dari hewan piton [12]. *Python* adalah bahasa pemrograman tingkat tinggi yang sering digunakan dalam pengembangan perangkat lunak, analisis data, kecerdasan buatan, dan berbagai aplikasi lainnya. *Python* dikenal dengan sintaksis yang bersih, mudah dipahami, dan mudah dipelajari, membuatnya menjadi pilihan populer bagi pengembang pemula dan ahli [13].

Python merupakan bahasa pemrograman yang banyak digunakan untuk berbagai keperluan, termasuk membangun situs *web*, mengembangkan perangkat lunak/aplikasi, mengotomatisasi tugas, dan melakukan analisis data. Keistimewaan *Python* terletak pada sifatnya yang merupakan bahasa tujuan umum, yang berarti dapat digunakan untuk membuat berbagai jenis program tanpa terbatas pada masalah tertentu saja. Karena fleksibilitasnya yang serba guna dan kemudahan penggunaannya.

Python juga sering digunakan untuk membangun bagian *back-end* dari sebuah situs *web* atau aplikasi, yang merupakan bagian yang tidak terlihat oleh pengguna. Selain itu, *Python* telah menjadi salah satu bahasa yang dominan dalam bidang ilmu data. Dengan *Python*, analisis data dapat dilakukan dengan melakukan perhitungan statistik yang kompleks, membuat visualisasi data, dan mengimplementasikan algoritma *machine learning* [14].

2.11 *Jupyter Notebook*

Jupyter Notebook adalah aplikasi *web* gratis dan *open-source* yang memfasilitasi pengguna dalam membuat dan berbagi dokumen yang menggabungkan kode pemrograman, visualisasi data, dan teks naratif [15]. Nama "*Jupyter*" sendiri adalah singkatan dari tiga bahasa pemrograman, yaitu *Julia* (Ju), *Python* (Py), dan R.

Jupyter Notebook sangat diminati oleh para *data scientist* karena memungkinkan mereka untuk menulis kode, melakukan perhitungan, menganalisis data, membuat visualisasi, dan menyajikan informasi secara interaktif dalam satu dokumen [36]. Selain itu, *Jupyter Notebook* juga memudahkan kerja sama antara *engineer* dan *data scientist* dengan memungkinkan mereka menulis dan berbagi teks serta kode dengan mudah. Struktur utama *Jupyter Notebook* terdiri dari tiga bagian:

1. *Front-end notebook*: Bagian ini memungkinkan pengguna untuk mengedit atau menjalankan *notebook* lain.
2. *Jupyter server*: *Server* ini adalah aplikasi yang berjalan di komputer.
3. Protokol *kernel*: Bagian ini memungkinkan *server* untuk melimpahkan tugas menjalankan kode menjadi berbagai bahasa.

Jupyter Notebook mendukung berbagai bahasa pemrograman, termasuk *Python*, dan menyediakan lingkungan interaktif yang memungkinkan pengguna untuk menulis dan menjalankan kode secara langsung [16].

2.12 *Skicit-Learn*

Scikit-learn adalah sebuah *library* untuk bahasa pemrograman *Python* yang umumnya digunakan dalam proyek *machine learning*. Fokus utama dari *Scikit-learn* adalah pada alat *machine learning*, termasuk algoritma matematika, statistik, dan tujuan umum yang membentuk dasar bagi banyak teknologi *machine learning* [38].

Library ini dibangun di atas beberapa data umum dan pustaka matematika *Python*, dengan penekanan yang kuat pada algoritma *machine learning* [17]. *Scikit-learn* menawarkan serangkaian algoritma yang kuat, termasuk:

1. Regresi: Memodelkan hubungan antara variabel dependen dan independen, baik secara linier maupun non-linier.
2. Klasifikasi: Mengklasifikasikan data tanpa adanya label atau supervisi eksternal.
3. *Decision Tree*: Membangun dan memangkas pohon keputusan untuk tugas klasifikasi dan regresi.
4. SVM (*Support Vector Machine*): Metode pembelajaran yang digunakan untuk mempelajari batas keputusan antara dua kelas.
5. *Naive Bayes*: Model probabilitas sederhana yang didasarkan pada teorema Bayes.
6. Metode *Ensemble*: Mencakup teknik seperti *Boosting, Bagging, Random Forest*, dan pemodelan *voting* dan *averaging* untuk meningkatkan kinerja model *machine learning* [37].

2.13 *Matplotlib*

Matplotlib merupakan sebuah *library Python* yang komprehensif dan bermanfaat untuk membuat visualisasi data, baik dalam bentuk statis maupun interaktif, dalam dimensi 2D maupun 3D [18]. Pertama kali dirilis pada tahun 2003 oleh seorang ahli saraf Amerika bernama John D. Hunter, *Matplotlib* telah menjadi salah satu alat yang sangat populer dalam analisis dan visualisasi data.

Salah satu kegunaan utama *Matplotlib* adalah untuk membuat plot grafik, baik untuk satu sumbu maupun lebih. Setiap sumbu dalam plot memiliki sumbu horizontal (x) dan sumbu vertikal (y), dan *Matplotlib* memberikan fleksibilitas bagi pengguna untuk membuat berbagai jenis plot sesuai kebutuhan, termasuk plot garis, histogram, scatter plot, dan banyak lagi. Dengan fitur-fitur yang lengkap dan kemampuan untuk menghasilkan visualisasi yang kaya dan informatif, *Matplotlib* menjadi salah satu pilihan utama bagi para pengguna *Python* dalam mengeksplorasi dan menyajikan data mereka [39].

2.14 *Pandas*

Pandas adalah *library Python* yang sering digunakan untuk analisis data, manipulasi data, dan pembersihan data [40]. *Pandas* menyediakan struktur data dan fungsi-fungsi yang kuat untuk bekerja dengan data terstruktur, terutama data tabular seperti *spreadsheet* atau *database SQL* [19]. Struktur data utama dalam *Pandas* adalah *DataFrame*, yang merupakan struktur data berbasis baris dan kolom yang sangat mirip dengan tabel dalam *database* atau *spreadsheet*. *DataFrame*

memungkinkan pengguna untuk melakukan berbagai operasi seperti pemfilteran, penggabungan, pengelompokan, dan agregasi data dengan mudah dan efisien.

Selain *DataFrame*, *pandas* juga menyediakan *Series*, yang merupakan struktur data berbasis *array* satu dimensi yang dapat digunakan untuk menyimpan data homogen seperti *array NumPy*. *Pandas* memiliki banyak fungsi bawaan untuk membantu dalam memuat data dari berbagai format file (seperti *CSV, Excel, SQL*), membersihkan dan memanipulasi data, serta membuat visualisasi sederhana dari data [41].

2.15 *KDD Cup 1999*

Dataset KDD Cup 1999 adalah *dataset* yang sangat sering digunakan sebagai acuan untuk mengevaluasi sistem deteksi intrusi. *Dataset* ini mencatat koneksi jaringan, yang terdiri dari satu jenis data normal dan 22 jenis data serangan, yang dikelompokkan ke dalam empat kategori intrusi. *Dataset* ini diperoleh dari UCI *Repository* [44]. *Dataset* ini memiliki 41 atribut/fitur yang dibagi ke dalam tiga kelompok: atribut dasar, atribut konten, dan atribut trafik [45].

Atribut dasar (atribut nomor 1 sampai 9) berasal dari *log sistem tcpdump* dalam jaringan komputer, yang menangkap rincian penting tentang koneksi tersebut. Atribut konten (atribut nomor 10 sampai 22) menangkap informasi dari aktivitas yang terjadi di dalam sistem jaringan komputer selama koneksi berlangsung. Atribut trafik dibagi lagi menjadi dua bagian: bagian pertama (atribut nomor 23 sampai 31) mencakup atribut trafik yang dihitung menggunakan jendela waktu dua detik, dan

bagian kedua (atribut nomor 32 sampai 41) mencakup atribut trafik yang dihitung menggunakan jendela waktu dua detik dari tujuan ke *host* [43].

2.16 *Unified modelling language (UML)*

Unified Modeling Language (UML) adalah bahasa visual yang digunakan untuk memodelkan dan merancang sistem perangkat lunak berorientasi objek. UML menyediakan serangkaian diagram untuk menggambarkan struktur dan perilaku sistem, serta interaksi antara komponen-komponen sistem [46]. UML memungkinkan pengembang untuk membuat representasi grafis dari sistem yang kompleks, memfasilitasi komunikasi antar anggota tim, dan membantu dalam eksplorasi dan validasi desain arsitektur perangkat lunak.

UML terdiri dari tiga kategori utama yaitu struktur diagram, *behavior diagram*, dan *interaction diagram*. *Diagram* struktur mencakup diagram kelas, objek, dan komponen, yang menggambarkan hubungan statis antar elemen sistem [47]. Diagram *behavior* mencakup diagram aktivitas dan *state*, yang menunjukkan perubahan keadaan sistem. Diagram interaksi mencakup diagram urutan dan komunikasi, yang mengilustrasikan aliran pesan antar objek dalam sistem. Dengan UML, pengembang dapat menciptakan dokumentasi yang konsisten dan standar untuk seluruh proses pengembangan perangkat lunak, dari analisis hingga implementasi [48].

2.17 *Visual Studio Code*

Visual Studio Code (VS Code) adalah editor kode sumber yang ringan namun kuat yang berjalan di desktop dan tersedia untuk Windows, macOS, dan Linux. Muncul dengan dukungan built-in untuk JavaScript, TypeScript, dan Node.js dan memiliki ekosistem ekstensi yang kaya untuk bahasa lain (seperti C++, C#, Java, Python, PHP, Go) dan runtime [50]. VS Code menawarkan berbagai fitur canggih yang mendukung proses penulisan dan debugging kode, seperti penyorotan sintaks, penyelesaian kode otomatis, snippet, restrukturisasi kode, dan integrasi Git. Fitur-fitur ini memungkinkan pengembang untuk menulis aplikasi web, mobile, dan cloud dengan lebih efisien di berbagai platform pengembangan [49].

2.18 Penelitian Terkait

Berikut adalah beberapa penelitian terkait dengan tugas akhir yang membahas tentang Penerapan Teknik *Machine Learning* dalam Deteksi Intrusi Jaringan untuk Meningkatkan Deteksi Ancaman pada Keamanan Sistem Informasi.

Tabel 2. 1 Penelitian Terkait

No	Nama	Tahun	Judul	Hasil
1.	Julian Dewanto	2024	Analisis Perbandingan *Dataset* KDD'99, UNSW-Nb15, dan CICICDS2017 Untuk Sistem Deteksi Intrusi	Berdasarkan pembahasan yang telah dilakukan, maka dapat disimpulkan penelitian ini menunjukkan bahwa penggunaan metode PCA dalam *preprocessing* dan pengurangan dimensi dapat memberikan hasil yang baik dalam konteks deteksi intrusi jaringan. Meskipun terdapat

				perbedaan dalam karakteristik dataset, evaluasi model dengan *Logistic Regression* memberikan gambaran tentang keefektifan deteksi intrusi pada masing-masing dataset. Dengan demikian, hasil ini dapat membantu peneliti dan praktisi dalam memilih pendekatan yang sesuai untuk meningkatkan keamanan jaringan berbasis deteksi intrusi. Hasil evaluasi model menunjukkan akurasi yang baik untuk KDD'99 (94,68%) dan UNSW-NB15 (98,87%), sementara CICIDS2017 memiliki akurasi yang lebih rendah sebesar 46,59%. Hal ini mengindikasikan perbedaan dalam karakteristik dataset dan kompleksitas deteksi intrusi.
2.	Sista Naelly Adzimi, Hafiz Agi Alfasih, Fauzan Naufal Gibran Ramadhan, Shelvie Nidya Neyman, Aep Setiawan	2024	*Implementasi* Konfigurasi *Firewall* dan *Sistem* Deteksi Intrusi Menggunakan Debian	Kesimpulan dari penelitian ini menunjukkan bahwa *implementasi firewall* dan sistem deteksi intrusi (IDS) menggunakan Debian memberikan peningkatan signifikan dalam keamanan jaringan. Pengujian yang komprehensif mengungkapkan bahwa firewall yang dikonfigurasi dengan tepat mampu memblokir upaya akses tidak sah dan mengidentifikasi pola

				serangan yang mencurigakan secara *efektif. Sistem* ini menunjukkan kinerja yang optimal dalam menghadapi berbagai skenario serangan, baik dari dalam maupun luar jaringan, termasuk serangan DDoS, brute force, dan serangan berlapis yang lebih kompleks. *Implementasi firewall* dan Wazuh IDS secara signifikan meningkatkan keamanan jaringan. *Firewall* berfungsi sebagai penghalang pertama terhadap ancaman, sementara Wazuh IDS menambahkan lapisan tambahan dengan mendeteksi aktivitas mencurigakan secara real-time. Implementasi ini dapat berfungsi sebagai model bagi lembaga pendidikan lain yang menghadapi tantangan serupa.
3.	Isma Elan Maulani, Aldo Faisal Umam	2023	*Evaluasi Efektivitas Sistem* Deteksi *Intrusi* Dalam Menjamin Keamanan Jaringan	Penelitian ini menyimpulkan bahwa *sistem* deteksi intrusi (IDS) merupakan komponen penting dalam menjaga keamanan jaringan. IDS memiliki kemampuan untuk mendeteksi aktivitas mencurigakan atau tidak diinginkan dalam jaringan dan berperan sebagai garis pertahanan pertama dalam menghadapi serangan siber. Meskipun IDS umumnya

				mampu mendeteksi serangan dengan tingkat yang cukup baik, masih ada tantangan yang perlu diatasi seperti keakuratan deteksi, respons yang cepat, dan pengelolaan kinerja jaringan yang optimal. Dalam rangka meningkatkan efektivitas IDS, diperlukan perbaikan dalam deteksi palsu guna mengurangi notifikasi serangan palsu yang dapat mengganggu kepercayaan pada IDS. Pengembangan metode deteksi yang lebih canggih dan kebijakan deteksi yang lebih cermat dapat membantu meningkatkan akurasi IDS. Respons yang cepat terhadap serangan yang terdeteksi juga sangat penting dalam meminimalkan dampak serangan, yang dapat dicapai melalui pengembangan metode deteksi yang lebih *efisien* dan pemantauan serangan yang kontinu. Penggunaan IDS dapat berdampak pada kinerja jaringan, oleh karena itu, perlu dipertimbangkan konfigurasi yang tepat dan pemilihan solusi IDS yang sesuai dengan kapasitas jaringan untuk menjaga keseimbangan antara keamanan dan kinerja. Rekomendasi praktis yang

				dihasilkan dari penelitian ini mencakup pembaruan dan peningkatan IDS secara berkala, pengurangan deteksi palsu, optimalisasi konfigurasi IDS, dan peningkatan pelatihan serta kesadaran pengguna. Secara keseluruhan, penelitian ini memberikan pemahaman yang lebih baik tentang efektivitas IDS dalam menjaga keamanan jaringan. Dengan memperhatikan tantangan yang ada dan mengimplementasikan rekomendasi yang dihasilkan, organisasi dapat meningkatkan keefektifan IDS dalam mendeteksi dan melindungi jaringan mereka dari serangan siber.
4.	Tony han, Hendi Sama, Gautama Wijaya, Osei Enoch Aboagye	2023	Studi Perbandingan Deteksi Intrusi Jaringan Menggunakan *Machine Learning*: (Metode SVM dan ANN)	Dalam penelitian ini, *algoritma machine learning* diimplementasikan untuk mendeteksi intrusi pada Dataset KDD Cup 99 digunakan. Dataset KDD Cup 99 dikumpulkan di MIT *Lincoln Laboratory*, yang di bawah *Defense Advanced Research Project Agency* (DARPA). Support *Vector Machines* (SVM) dan Artificial Neural Network (ANN) adalah dua algoritma *supervised machine learning* yang digunakan dalam proyek ini. Parameter yang berbeda digunakan untuk

				mengukur kinerja algoritma. Parameter meliputi; akurasi pelatihan, akurasi pengujian, waktu pelatihan, waktu penguujian, AUC dan kecepatan jaringan. AUC dilakukan dalam metode *multiclass* menggunakan ovr (one-versus-rest). Status probabilitas diatur ke true untuk algoritma karena probabilitas false secara default. Ada lima (5) kelas dan AUC selesai mengambil satu kelas melawan empat lainnya. Hasil dari eksperimen (pelatihan, pengujian dan AUC) pada Dataset KDD CUP 99 yang diunduh dari Kaggle menunjukkan bahwa SVM dan ANN dapat secara efektif mendeteksi serangan intrusi. SVM memiliki akurasi pelatihan dan pengujian masing-masing 99,87% dan 99,81%. SVM dilatih dan diuji masing-masing dalam waktu 2 menit dan 1 menit. Pada AUC, SVM memiliki 1,00 untuk semua kelas (normal, DOS, U2R, R2L dan Probe). ANN memiliki akurasi pelatihan dan pengujian masingmasing 99,86% dan 99,85%. ANN dilatih dan diuji dalam waktu masing-masing 11 menit dan 10 detik. Pada AUC, ANN memiliki 1,00

				untuk normal, DOS, Probe dan R2L tetapi 0,99 untuk U2R. Melihat hasil di atas, dapat disimpulkan bahwa kedua algoritma tersebut baik untuk deteksi intrusi tetapi SVM lebih efektif daripada ANN. Oleh karena itu saya merekomendasikan SVM untuk deteksi intrusi. Proyek ini dapat diimplementasikan di perusahaan untuk menganalisis paket jaringan, mendeteksi dan melaporkan serangan di sistem mereka. Proyek ini sangat berguna bagi semua orang. Mahasiswa yang mempelajari keamanan siber dan administrator server dapat memperoleh banyak wawasan tentang deteksi intrusi dari proyek ini. Metode hibrida bekerja lebih efektif karena menggabungkan dua atau lebih algoritma. Pada proyek selanjutnya, penulis akan menggunakan SVM dan ANN dalam *hybrid intrusion detection.* Keterbatasan penelitian ini.
5.	Faizal Riza	2023	Sistem Deteksi Intrusi Pada Server Secara Realtime Menggunakan Seleksi Fitur dan *Firebase Cloud Messaging*	Studi ini mengembangkan algoritma efektif yang dapat diterapkan pada IDS praktis dan memainkan peran penting dalam mendeteksi aktivitas ilegal melalui jaringan komputer. Berkembangnya jaringan serta teknologi informasi

				akan mengakibatkan berkembangnya jenis serangan pada masa yang akan datang sehingga sehingga perlu dilakukan penelitian lebih lanjut
6.	Irma Elan Maulani, Dwi Rayhan Sunandar Putra, Komarudin	2023	Sistem Intrusi Cerdas: Studi Perbandingan Algoritma Pembelajaran Mesin Untuk Keamanan Siber	Dalam penelitian ini, kami menjelajahi dan membandingkan kinerja berbagai algoritma pembelajaran mesin dalam konteks Sistem Deteksi Intrusi (IDS) untuk meningkatkan keamanan siber. Temuan penelitian memberikan wawasan penting yang dapat membimbing keputusan implementasi dan pengembangan solusi keamanan siber yang efektif. Perbandingan kinerja algoritma menunjukkan variasi yang signifikan. Algoritma B muncul sebagai pilihan yang menjanjikan dengan keseimbangan akurasi dan tingkat kebocoran yang baik, sementara Algoritma C menonjol dalam mendeteksi serangan spesifik. Algoritma A, sementara memiliki akurasi tinggi, juga menunjukkan tingkat kebocoran yang tinggi. Pengaruh skalabilitas menjadi faktor kritis dalam pemilihan algoritma. Algoritma C menunjukkan respons yang

				baik pada skala yang lebih besar, sementara Algoritma A mengalami penurunan kinerja
7.	Isma Elan Maulani, Dwi Rayhan Sunandar Putra, Komarudin	2023	Sistem Deteksi intrusi Cerdas: Studi kasus Perbandingan Algoritma Pembelajaran Mesin Untuk Keamanan Siber	Dalam penelitian ini, kami menjelajahi dan membandingkan kinerja berbagai algoritma pembelajaran mesin dalam konteks Sistem Deteksi Intrusi (IDS) untuk meningkatkan keamanan siber. Temuan penelitian memberikan wawasan penting yang dapat membimbing keputusan *implementasi* dan pengembangan solusi keamanan siber yang efektif. Perbandingan kinerja algoritma menunjukkan variasi yang signifikan. Algoritma B muncul sebagai pilihan yang menjanjikan dengan keseimbangan akurasi dan tingkat kebocoran yang baik, sementara Algoritma C menonjol dalam mendeteksi serangan spesifik. Algoritma A, sementara memiliki akurasi tinggi, juga menunjukkan tingkat kebocoran yang tinggi. Pengaruh skalabilitas menjadi faktor kritis dalam pemilihan algoritma. Algoritma C menunjukkan respons yang baik pada skala yang lebih besar, sementara

				Algoritma A mengalami penurunan kinerja yang signifikan.Keberhasilan Sistem Deteksi Intrusi tergantung pada keberlanjutan dan adaptabilitas algoritma. Algoritma yang mampu belajar secara dinamis dan mengidentifikasi tren baru dalam serangan siber terbukti lebih efektif. Peran analis keamanan manusia tetap krusial. Integrasi manusia dalam interpretasi hasil IDS dan pengambilan keputusan memperkuat ketahanan sistem. Pertimbangan biaya dan pemeliharaan harus seimbang dengan keputusan implementasi. Algoritma dengan kinerja tinggi tidak boleh mengabaikan keterbatasan sumber daya yang mungkin diperlukan untuk pemeliharaan dan operasionalitas sehari-hari.
8.	Fajar Wibowo	2023	Analisis Algoritma Kecerdasan Buatan Pada Pengembangan Sistem Deteksi Intrusi Pada Jaringan Komputer	Analisis algoritma kecerdasan buatan dalam pengembangan sistem deteksi intrusi pada jaringan komputer adalah suatu langkah penting untuk memahami dan meningkatkan efektivitas sistem yang dibangun [201]–[210]. Hasil dari analisis ini mencakup evaluasi performa berbagai algoritma yang digunakan dalam mendeteksi

				intrusi, seperti *Decision Trees, Support Vector Machines, Neural Networks,* dan algoritma genetika. Salah satu hasil yang dapat ditemukan dalam analisis ini adalah efisiensi dan akurasi masing-masing algoritma [211]–[220]. Penelitian ini dapat membandingkan kinerja algoritma-algoritma tersebut dalam mendeteksi ancaman keamanan pada jaringan, termasuk sejauh mana mereka dapat mengidentifikasi serangan yang berbeda dan seberapa cepat mereka merespons terhadap situasi yang berkembang
9.	Isma Elan Maulani, Aldo Faisal Umam	2023	*Evaluasi Efektivitas Sistem* Deteksi Intrusi Dalam Menjamin Keamanan Jaringan	Penelitian ini menyimpulkan bahwa sistem deteksi intrusi (IDS) merupakan komponen penting dalam menjaga keamanan jaringan. IDS memiliki kemampuan untuk mendeteksi aktivitas mencurigakan atau tidak diinginkan dalam jaringan dan berperan sebagai garis pertahanan pertama dalam menghadapi serangan siber. Meskipun IDS umumnya mampu mendeteksi serangan dengan tingkat yang cukup baik, masih ada tantangan yang perlu diatasi seperti keakuratan deteksi, respons yang cepat, dan pengelolaan kinerja jaringan yang optimal. Dalam rangka

				meningkatkan efektivitas IDS, diperlukan perbaikan dalam deteksi palsu guna mengurangi notifikasi serangan palsu yang dapat mengganggu kepercayaan pada IDS. Pengembangan metode deteksi yang lebih canggih dan kebijakan deteksi yang lebih cermat dapat membantu meningkatkan akurasi IDS. Respons yang cepat terhadap serangan yang terdeteksi juga sangat penting dalam meminimalkan dampak serangan, yang dapat dicapai melalui pengembangan metode deteksi yang lebih efisien dan pemantauan serangan yang kontinu.
10.	Andi Muhammad Nur Hidayat	2021	Sistem Deteksi Intrusi Dan Prevensi Berbasis *Open Source*	Berdasarkan hasil pengujian dengan skenario serangan yang dibuat aplikasi WAIDPS mampu mendeteksi dan mengklasifikasikan serangan yang terjadi. Bahkan proses authentikasi dan deauthentikasi yang dilakukan oleh pengguna jaringan semua bisa dimonitoring oleh aplikasi. Setelah itu administrator jaringan bisa segera melakukan pencegahan agar tidak terjadi kerugian pada pengguna. Aplikasi WAIDPS mampu dikembangkan untuk selanjutnya diterapkan pada jaringan komputer yang

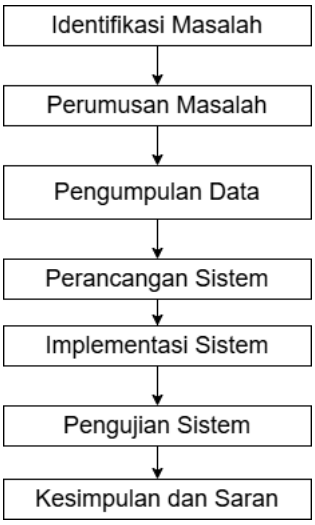
				sesuai dengan kebijakan masing-masing. Proses monitoring dari WIDPS ini bersifat real-time sehingga dibutuhkan proses pengawasan oleh administrasi jaringan.

BAB 3

METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang terhubung. Tahapan tersebut dijelaskan dalam metode penelitian dengan skema yang sistematis, terstruktur, dan jelas. Tahapan-tahapan penelitian ini ditunjukkan dalam Gambar 3.1:



Gambar 3. 1 Tahapan Penelitian

Penjelasan dari tahapan penelitian pada gambar 3.1 dapat dilihat pada penjelasan dibawah ini:

3.2 Identifikasi Masalah

Identifikasi masalah merupakan tahapan awal yang dilakukan dalam penelitian ini dengan tujuan untuk mengamati dan memahami permasalahan yang terjadi. Adapun permasalahan yang dapat diidentifikasi untuk pelaksanaan tugas akhir ini adalah sebagai berikut:

1. Meningkatnya ancaman terhadap keamanan jaringan yang memerlukan deteksi intrusi.
2. Perlunya sistem deteksi intrusi yang dapat diintegrasikan dengan mudah ke dalam lingkungan jaringan yang ada.

3.3 Perumusan Masalah

Rumusan masalah utama dalam penelitian ini adalah bagaimana implementasi metode *random forest* untuk sistem deteksi intrusi?

3.4 Pengumpulan Data

Tahap pengumpulan data melibatkan beberapa sub-tahap sebagai berikut:

3.4.1 *Dataset*

Penelitian ini menggunakan *dataset KDDCup* sebagai data latih awal. *KDDCup* terdiri dari serangkaian transaksi jaringan yang merekam aktivitas pada sistem jaringan komputer, termasuk akses pengguna, pengiriman paket data, dan tindakan lainnya. Dataset ini mencakup berbagai jenis serangan dan aktivitas normal, sehingga cocok digunakan untuk melatih dan menguji model deteksi intrusi.

3.4.2 *Pre-Processing*

Tahap *pre-processing* mencakup beberapa langkah penting untuk memastikan kualitas dan konsistensi data sebelum digunakan untuk pelatihan model. Langkah-langkah ini meliputi:

1. Menghapus atau memperbaiki data yang tidak lengkap, duplikat, atau tidak valid untuk memastikan integritas data.

2. Mengubah data ke skala yang sama untuk memastikan bahwa setiap fitur memiliki kontribusi yang seimbang dalam pelatihan model. Teknik normalisasi seperti *Min-Max Scaling* digunakan dalam tahap ini.

3.4.3 *Train Test Split*

Setelah data di proses, *dataset* dibagi menjadi dua bagian, yaitu :

1. *Training Dataset:* 80% dari data digunakan untuk melatih model. Bagian ini digunakan untuk mengidentifikasi pola dan hubungan dalam data.
2. *Testing Dataset*: 20% dari data digunakan untuk menguji performa model. Bagian ini memastikan bahwa model dapat membuat prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

3.5 Perancangan Sistem

Tahap perancangan sistem melibatkan pembuatan model dan arsitektur sistem yang akan digunakan dalam penelitian ini. Langkah-langkah ini meliputi:

3.5.1 *Training Model*

Pada tahap ini, model *machine learning* dilatih menggunakan *training dataset*. Proses pelatihan melibatkan langkah-langkah berikut:

1. Pemilihan Algoritma: Memilih algoritma *machine learning* untuk tugas deteksi intrusi. Algoritma yang digunakan pada penelitian ini adalah *random forest*.

2. Pelatihan Model: Model dilatih menggunakan *training dataset* untuk mengenali pola dan hubungan dalam data. Proses ini melibatkan iterasi berulang hingga model mencapai performa yang optimal

3.5.2 *Parameter Tuning*

Setelah model dilatih, langkah selanjutnya adalah mengoptimalkan kinerja model melalui proses *parameter tuning.* Langkah-langkah ini meliputi:

1. Penyesuaian *Parameter*: Mengatur *parameter* model seperti *learning rate*, jumlah pohon dalam *random forest*, untuk meningkatkan akurasi dan performa model.
2. *Cross-Validation*: Menggunakan teknik *cross-validation* untuk memastikan bahwa model tidak *overfitting* dan dapat generalisasi dengan baik pada data baru.

3.6 Implementasi Sistem

Setelah sistem dirancang, langkah berikutnya adalah implementasi sistem ke dalam lingkungan nyata. Tahap ini melibatkan implementasi model ke dalam kode *Python* dan menguji fungsionalitasnya. Langkah-langkah ini meliputi:

1. Implementasi Model: Mengimplementasikan model *machine learning* yang telah dilatih ke dalam kode *Python*.
2. Integrasi dengan Sistem: Mengintegrasikan model dengan sistem yang ada untuk memastikan bahwa model dapat bekerja dalam lingkungan nyata.

3. Uji Coba Sistem: Menguji sistem untuk memastikan bahwa model dapat mendeteksi intrusi.

3.7 Pengujian Sistem

Tahap pengujian sistem melibatkan pengujian model dan sistem secara keseluruhan untuk memastikan bahwa mereka bekerja sesuai dengan yang diharapkan. Langkah-langkah ini meliputi:

1. Pengujian *Black Box*: Menguji sistem dari perspektif pengguna tanpa melihat ke dalam kode untuk memastikan bahwa sistem berfungsi sesuai dengan yang diharapkan.
2. Evaluasi Performa Model: Menggunakan *testing dataset* untuk mengukur seberapa baik model dapat membuat prediksi. Evaluasi ini melibatkan metrik kinerja seperti akurasi, presisi, *recall*, dan *F1-score*.

3.8 Kesimpulan dan Saran

Tahap terakhir dalam penelitian ini adalah menarik kesimpulan dan memberikan saran berdasarkan hasil yang diperoleh. Langkah-langkah ini meliputi:

1. Visualisasi Hasil: Menggunakan grafik dan diagram untuk menyajikan hasil evaluasi model, seperti *confusion matrix, ROC curve*, dan *precision-recall curve.*
2. Analisis Temuan: Menginterpretasikan hasil visualisasi untuk mengidentifikasi kekuatan dan kelemahan model, serta memberikan rekomendasi untuk perbaikan lebih lanjut.

3. Kesimpulan: Menyimpulkan temuan utama dari penelitian dan memberikan gambaran umum tentang efektivitas model dalam mendeteksi intrusi jaringan.
4. Saran: Memberikan saran untuk penelitian lebih lanjut dan peningkatan model di masa depan.