BAB II
TINJAUAN PUSTAKA

2.1 Kajian Relevan

Penelitian terdahulu yang relevan dijadikan sebagai rujukan untuk membandingkan serta memperkuat landasan penelitian ini. Melalui kajian tersebut, dapat diidentifikasi fokus kajian, metode yang digunakan, dan hasil yang telah dicapai oleh penelitian sebelumnya. Penelitian ini menitikberatkan pada penerapan metode *unsupervised learning* autoencoder yang dikombinasikan dengan teknik penyeimbangan data *random oversampling*, serta evaluasi berbagai metode normalisasi data untuk meningkatkan stabilitas model dalam mendeteksi anomali lalu lintas botnet pada dataset CIC-IDS-2018.

Penelitian pertama dilakukan oleh Kanimozhi & Jacob (2021) dengan judul "*Artificial Intelligence outflanks all other machine learning classifiers in Network Intrusion Detection System on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing*" yang mengkaji efektivitas berbagai metode pembelajaran mesin dalam mendeteksi serangan botnet. Penelitian ini membandingkan beberapa algoritma, antara lain K-Nearest Neighbours, Support Vector Machine, Naïve Bayes, Random Forest, serta Artificial Neural Network (ANN). Hasil penelitian menunjukkan bahwa model ANN mampu melampaui metode lainnya dengan tingkat akurasi mencapai 97%. Kesamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan dataset yang sama, yaitu CSE-CIC-IDS-2018, serta fokus pada deteksi serangan lalu lintas botnet. Adapun perbedaannya terletak pada pendekatan yang digunakan. Penelitian tersebut menggunakan algoritma *supervised learning* yang sangat bergantung pada label data, sedangkan penelitian ini menggunakan pendekatan *unsupervised learning* berbasis autoencoder serta berfokus pada pengaruh metode normalisasi data.

Penelitian kedua oleh De Bettignies (2021) dengan judul "*LSTM Autoencoders for Botnet Detection*" berfokus pada perbandingan performa model LSTM Autoencoder dan Random Forest dalam mendeteksi botnet menggunakan dataset ISOT Botnet. Hasil pengujian menggunakan metrik ROC-AUC menunjukkan bahwa model Random Forest unggul dengan nilai sebesar 0,99, mengindikasikan model tersebut lebih stabil pada

dataset yang digunakan. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pemanfaatan arsitektur autoencoder sebagai salah satu pendekatan utama untuk mendeteksi lalu lintas botnet. Adapun perbedaannya terletak pada jenis dataset yang digunakan, di mana penelitian tersebut menggunakan dataset ISOT, serta belum adanya kajian spesifik mengenai penanganan ketidakseimbangan kelas (*imbalanced data*) menggunakan *random oversampling* seperti yang diterapkan pada penelitian ini.

Penelitian ketiga dilakukan oleh Chindove & Brown (2021) dengan judul "*Adaptive Machine Learning Based Network Intrusion Detection*" mengevaluasi performa algoritma pembelajaran mesin pada dataset CIC-IDS-2017 dan CIC-IDS-2018. Beberapa model yang diuji antara lain RNN, Random Forest, SVM, KNN, MLP, dan Decision Tree. Hasil pengujian menunjukkan bahwa RNN dan Random Forest memberikan performa terbaik dengan nilai Macro F1-score masing-masing sebesar 0,73 dan 0,72. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pemanfaatan dataset CIC-IDS-2018 sebagai objek pengujian model deteksi intrusi jaringan. Adapun perbedaannya terletak pada arsitektur yang dievaluasi berfokus pada komparasi model klasifikasi konvensional dan *supervised learning*, sedangkan penelitian ini secara khusus mengeksplorasi kemampuan rekonstruksi *error* pada *unsupervised learning* autoencoder yang dipengaruhi oleh perbedaan teknik penskalaan fitur (Min-Max, Standardization, dan Robust).

Penelitian keempat oleh Park et al. (2023) dengan judul "*An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks*" mengusulkan sistem IDS berbasis kecerdasan buatan untuk mengatasi permasalahan ketidakseimbangan data yang umum terjadi pada dataset IDS. Penelitian ini mengombinasikan Generative Adversarial Networks (GAN) dengan *autoencoder-driven model* dan diuji pada dataset NSL-KDD serta UNSW-NB15. Hasil evaluasi menunjukkan bahwa model tersebut mampu mencapai akurasi sebesar 93,2% pada dataset NSL-KDD dan 87% pada UNSW-NB15. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pemanfaatan arsitektur autoencoder untuk deteksi intrusi jaringan serta adanya upaya penanganan masalah ketidakseimbangan kelas (*imbalanced data*). Adapun perbedaannya terletak pada teknik penyeimbangan data dan dataset yang digunakan.

Penelitian tersebut menggunakan GAN untuk augmentasi data pada dataset NSL-KDD dan UNSW-NB15, sedangkan penelitian ini menggunakan teknik *random oversampling* pada dataset CIC-IDS-2018.

Penelitian kelima oleh Dini et al. (2023) dengan judul "*Overview on Intrusion Detection Systems Design Exploiting Machine Learning for Networking Cybersecurity*" menyajikan tinjauan komprehensif mengenai desain Intrusion Detection System (IDS) berbasis pembelajaran mesin pada berbagai dataset populer, seperti KDD99, UNSW-NB15, dan CSE-CIC-IDS-2018. Hasil kajian menunjukkan bahwa Random Forest secara konsisten memberikan performa terbaik pada ketiga dataset tersebut untuk skema klasifikasi biner maupun multi-kelas. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pemanfaatan dataset CSE-CIC-IDS-2018 sebagai salah satu tolok ukur evaluasi model deteksi intrusi. Adapun perbedaannya terletak pada jenis metodologi dan fokus model yang merupakan sebuah tinjauan komprehensif (kajian pustaka) dengan menyoroti keunggulan model *supervised learning* (Random Forest), sedangkan penelitian ini merupakan eksperimen teknis yang berfokus pada *unsupervised learning* autoencoder serta evaluasi pengaruh normalisasi data.

Penelitian keenam dilakukan oleh Nicholas Sibarani et al. (2023) dengan judul "*Intrusion Detection Systems pada Bot-IoT Dataset Menggunakan Algoritma Machine Learning*" mengkaji performa beberapa algoritma pembelajaran mesin dalam mendeteksi intrusi pada sistem *Anomaly-based Intrusion Detection System* (AIDS) menggunakan dataset BoT-IoT. Penelitian ini membandingkan tiga algoritma klasifikasi, yaitu K-Nearest Neighbor (KNN), Random Forest, dan Gaussian Naïve Bayes. Hasil penelitian menunjukkan bahwa pemilihan algoritma yang tepat berpengaruh signifikan terhadap performa deteksi intrusi pada dataset IoT berskala besar. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pendekatan sistem yang digunakan, yaitu berbasis *Anomaly-based Intrusion Detection System* (AIDS) untuk mendeteksi anomali jaringan. Adapun perbedaannya terletak pada model algoritma dan objek data yang menggunakan klasifikasi *supervised learning* pada dataset BoT-IoT, sedangkan penelitian ini menggunakan metode *unsupervised learning* autoencoder pada dataset CIC-IDS-2018.

Penelitian ketujuh oleh Alhassan et al. (2023) dengan judul "*Analyzing Autoencoder-Based Intrusion Detection System Performance: Impact of Hidden Layers*" menganalisis performa sistem IDS berbasis autoencoder dengan meninjau pengaruh jumlah *hidden layer* terhadap hasil deteksi. Evaluasi yang dilakukan pada dataset NSL-KDD dan CIC-IDS-2017 menunjukkan bahwa autoencoder dengan satu *hidden layer* justru memberikan hasil yang lebih optimal dibandingkan arsitektur multilayer. Kesamaan penelitian tersebut dengan penelitian ini terletak pada penggunaan algoritma *unsupervised learning* autoencoder sebagai model utama dalam sistem deteksi intrusi. Adapun perbedaannya terletak pada fokus optimasi model dengan berfokus pada pengujian jumlah *hidden layer* arsitektur jaringan saraf, sedangkan penelitian ini berfokus pada optimasi prapemrosesan data, khususnya pengaruh ketiga metode normalisasi yaitu Min-Max, Standardization, Robust terhadap stabilitas *error* rekonstruksi.

Pada penelitian kedelapan oleh Khan et al. (2024) dengan judul "*Balanced Multi-Class Network Intrusion Detection Using Machine Learning*" mengusulkan pendekatan untuk mengatasi ketidakseimbangan data pada dataset CIC-IDS-2017 menggunakan teknik SMOTE-Tomek. Hasil pengujian terhadap beberapa algoritma menunjukkan bahwa model Decision Tree memberikan performa terbaik dengan akurasi sebesar 96,37% untuk klasifikasi multi-kelas sekaligus menghasilkan tingkat *false alarm* yang rendah. Kesamaan penelitian tersebut dengan penelitian ini terletak pada fokus utama untuk menyelesaikan tantangan ketidakseimbangan kelas (*imbalanced data*) dalam deteksi intrusi jaringan. Adapun perbedaannya terletak pada metode penyeimbangan dan model deteksinya. Penelitian tersebut menggunakan teknik hibrida SMOTE-Tomek dan algoritma *supervised learning* (Decision Tree), sedangkan penelitian ini menggunakan teknik *random oversampling* yang dikombinasikan dengan *unsupervised learning* autoencoder.

Penelitian kesembilan yang dilakukan oleh Nursiaga et al. (2025) dengan judul "*Model Jaringan Neural Untuk Deteksi Anomali Pada Sistem Keamanan (Siber): Rancangan, Implementasi, Dan Analisis*" mengembangkan model deep learning berbasis Autoencoder dan Long Short-Term Memory (LSTM) untuk mendeteksi anomali pada sistem keamanan siber. Hasil pengujian pada dataset NSL-KDD dan UNSW-NB15

menunjukkan bahwa model hibrida tersebut mampu mencapai akurasi sebesar 96,7% dengan waktu deteksi rata-rata 1,4 detik. Kesamaan penelitian tersebut dengan penelitian ini terletak pada pemanfaatan arsitektur autoencoder untuk mengenali pola lalu lintas normal dalam skema keamanan siber. Adapun perbedaannya terletak pada arsitektur dan ruang lingkup pengujian dimana penelitian tersebut menggabungkan Autoencoder dengan model sekuensial LSTM pada lingkungan simulasi kampus, sedangkan penelitian ini murni berfokus pada evaluasi performa autoencoder tunggal yang dioptimalkan melalui normalisasi fitur berskala ekstrem pada dataset CIC-IDS-2018.

Penelitian kesepuluh dan juga penelitian terbaru oleh Iwanowski et al. (2025) dengan judul "*The Choice of Training Data and the Generalizability of Machine Learning Models for Network Intrusion Detection Systems*" membahas isu generalisasi model dalam sistem IDS dengan melakukan evaluasi lintas dataset, termasuk CIC-CSE-IDS2018. Hasil eksperimen menyoroti adanya bias dataset yang menyebabkan model dengan performa tinggi pada data pelatihan (seperti Decision Tree yang mencapai akurasi 98%) mengalami penurunan performa secara drastis saat diterapkan pada lalu lintas nyata. Kesamaan penelitian tersebut dengan penelitian ini terletak pada objek evaluasi yang memanfaatkan dataset CIC-IDS-2018 serta penekanan pada tantangan bias dataset dalam melatih model IDS. Adapun perbedaannya terletak pada skenario evaluasi dimana penelitian tersebut berfokus pada masalah generalisasi melalui uji lintas dataset (*cross-dataset evaluation*) menggunakan model berbasis pohon, sedangkan penelitian ini berfokus pada penyelesaian bias akibat ketidakseimbangan kelas dalam satu dataset spesifik menggunakan autoencoder dan *random oversampling*.

Berdasarkan tinjauan pada berbagai penelitian terdahulu di atas, dapat ditarik kesimpulan bahwa sebagian besar pendekatan deteksi intrusi jaringan telah terbukti memberikan hasil yang baik ketika menggunakan algoritma *supervised learning* seperti *Random Forest*, *Decision Tree*, maupun *Artificial Neural Network* (ANN). Meskipun memiliki keunggulan dalam mendeteksi pola yang sudah dikenal, metode-metode tersebut memiliki kelemahan utama berupa ketergantungan yang tinggi terhadap kelengkapan label data dan rentan terhadap masalah generalisasi saat diterapkan pada kondisi lalu lintas jaringan dunia nyata yang dinamis.

Di sisi lain, pendekatan *unsupervised learning*, khususnya autoencoder, mulai banyak dieksplorasi dan terbukti potensial karena kemampuannya mempelajari pola tanpa label. Beberapa penelitian juga telah berupaya menangani isu ketidakseimbangan kelas (*imbalanced data*) menggunakan teknik seperti GAN maupun SMOTE-Tomek. Namun, masih terdapat celah penelitian (*research gap*) yang krusial, yaitu belum adanya kajian yang secara spesifik menyoroti bagaimana perbedaan skala fitur yang ekstrem pada dataset modern seperti CIC-IDS-2018 memengaruhi stabilitas konvergensi model autoencoder. Oleh karena itu, penelitian ini hadir untuk mengisi celah tersebut dengan mengkaji efektivitas *unsupervised learning* autoencoder yang dikombinasikan dengan teknik *random oversampling*, serta mengevaluasi secara komprehensif pengaruh berbagai metode normalisasi data terhadap performa model dalam mendeteksi anomali botnet yang samar.

2.2 Landasan Teori

2.2.1 Unsupervised Learning

Unsupervised Learning merupakan pendekatan pembelajaran mesin yang berfokus pada pemahaman pola tersembunyi dalam data tanpa adanya label kelas. Menurut (Goodfellow et al., 2017)), model ini mempelajari pola dasar dari data dengan melakukan pengelompokan atau penyederhanaan fitur agar struktur data lebih mudah dianalisis. Pendekatan ini sangat berguna ketika data dalam jumlah besar tersedia, tetapi proses pelabelan membutuhkan waktu, biaya, dan tenaga yang signifikan. Dengan demikian, model mampu mengekstraksi informasi laten yang sebelumnya tidak tampak secara eksplisit.

Dalam konteks deteksi intrusi jaringan, *unsupervised learning* menjadi relevan karena sebagian besar lalu lintas jaringan bersifat tidak berlabel dan sangat dinamis. Data jaringan terus berkembang setiap detik dengan karakteristik yang variatif, sehingga pendekatan supervised yang sangat bergantung pada pelabelan dianggap kurang adaptif terhadap variasi lalu lintas baru. Penelitian yang dilakukan oleh (Shone et al., 2018) menunjukkan bahwa metode *unsupervised* mampu mengidentifikasi serangan baru seperti *zero day attacks* yang sampai saat ini belum ada dalam dataset pelatihan. Hal ini

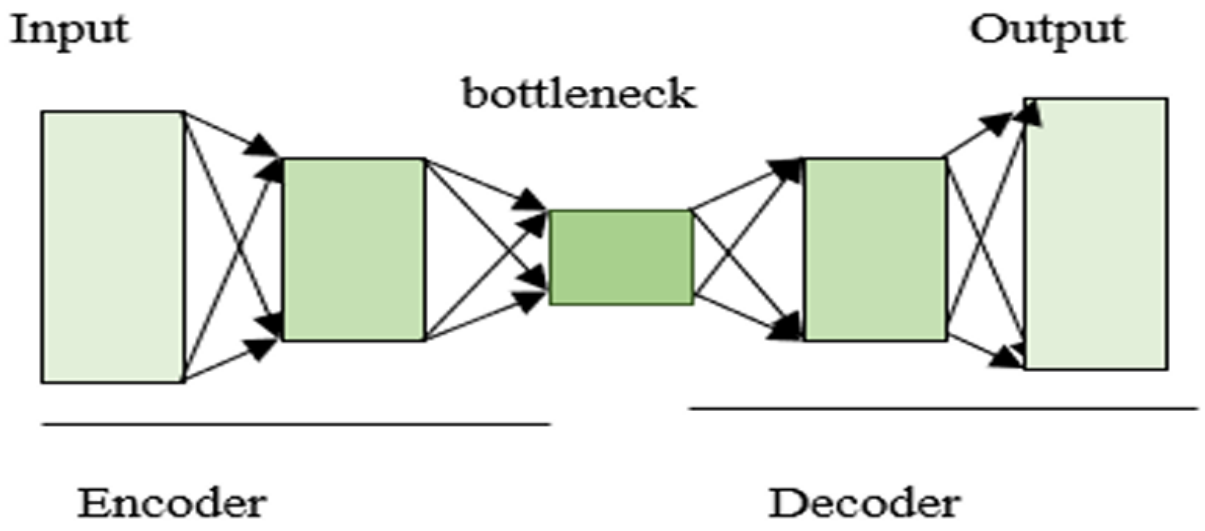
menunjukkan bahwa metode *unsupervised* lebih fleksibel dalam menghadapi ancaman yang terus berevolusi.

2.2.2 Autoencoder

Autoencoder adalah salah satu arsitektur *neural network* dalam *unsupervised learning* yang berfungsi melakukan proses kompresi (*encoding*) dan rekonstruksi (*decoding*) sebuah data. Konsep autoencoder dijelaskan oleh (Hinton & Salakhutdinov, 2006) dalam laporan berjudul "*Reducing the Dimensionality of Data with Neural Networks*". Arsitektur ini bekerja dengan meminimalkan kesalahan dalam rekonstruksi atau *reconstruction error*, yaitu selisih antara input dan juga output sehingga dapat mempelajari representasi data secara efisien. Dalam konteks deteksi anomali, autoencoder dilatih dengan mempelajari pola lalu lintas normal. Ketika model menghadapi lalu lintas botnet, nilai rekonstruksi yang didapatkan akan bernilai tinggi karena ketidaksesuaian pola (Sakurada & Yairi, 2014). Dengan kata lain, autoencoder berfungsi dalam mekanisme penyaring yang dapat mengidentifikasi perbedaan antara perilaku normal dan serangan secara signifikan. Komponen utama dalam autoencoder mencakup:

1. Encoder yaitu proses transformasi input ke representasi laten berbentuk vektor dengan dimensi rendah.
2. Latent Representation atau representasi laten yaitu representasi data secara vektor yang terkompresi dengan hanya menetapkan fitur penting ke dalam struktur data.
3. Decoder yaitu proses rekonstruksi ulang data dari representasi laten ke bentuk asli.

4. Reconstruction Error yaitu metode pengukuran untuk mengetahui perbedaan antara data masukan (input) dan juga keluaran (output) dari autoencoder.



Gambar 2. 1 Struktur Autoencoder

Pada Gambar 2.1 diambil dari penelitian oleh Alhassan et al. (2023), struktur dasar autoencoder terdiri atas tiga bagian yaitu bagian *encoder*, bagian *bottleneck* (*hidden layer*), dan bagian *decoder*. Adapun rumus dasar yang berlaku dalam autoencoder adalah sebagai berikut :

1. Encoder

$$z = f_\theta(x) = \sigma(W_e x + b_e) \quad \textbf{(2.1)}$$

Di mana :

a. x merupakan input data
b. W_e, b_e merupakan parameter encoder (weight dan bias)
c. σ merupakan fungsi aktivasi (ReLU/sigmoid)
d. z merupakan representasi laten (bottleneck)

2. Decoder

$$\hat{x} = g_\phi(z) = \sigma(W_d z + b_d) \quad \textbf{(2.2)}$$

Di mana :

a. W_d, b_d merupakan parameter decoder
b. $\hat{x}$ merupakan output rekonstruksi
c. z merupakan hasil perhitungan encoder

3. Reconstruction Error

$$\mathcal{L}_{\mathcal{MSE}}(x, \hat{x}) = \frac{1}{n}\sum_{i=1}^{n} | x_i - \hat{x_i}|^2 \quad \textbf{(2.3)}$$

Di mana :

a. x_i merupakan data input ke-i
b. $\hat{x_i}$ merupakan hasil rekonstruksi dari decoder
c. n merupakan jumlah data dalam satu batch
d. $|x_i - \hat{x_i}|^2$ merupakan jarak kuadrat (error) antara input asli dan output rekonstruksi

4. Latent Space Representation

$$\mathrm{Z} = \{\mathrm{z} \in \mathrm{R}^{\mathrm{k}} \mid \mathrm{z} = \mathrm{f}_{\theta}(\mathrm{x}), \mathrm{x} \in \mathrm{X}\} \quad \textbf{(2.4)}$$

Di mana :

a. $z \in R^k$ merupakan representasi dari latent space
b. $k < d$ yang dimana d adalah dimensi input sedangkan k adalah dimensi laten
c. z merupakan himpunan yang dihasilkan oleh encoder

Penelitian yang dilakukan oleh (Mirsky et al., 2018) juga membuktikan efektivitas autoencoder dalam mendeteksi serangan botnet yang berbasis lalu lintas jaringan dengan nilai *false positive* rendah. Selain itu, autoencoder memiliki kemampuan untuk beroperasi secara daring serta dapat beradaptasi dengan lalu lintas baru tanpa label.

2.2.3 Botnet Traffic Detection

Botnet adalah suatu jaringan perangkat yang saling terhubung oleh pengendali bot (*botmaster*) untuk aktivitas berbahaya seperti *Distributed Denial of Service*, spam, dan pencurian data. (Silva et al., 2013) dalam karya tulis berjudul "*Botnets: A survey*" menyatakan bahwa lalu lintas botnet ditandai dengan pola komunikasi secara periodik, *command-and control* (C&C), serta perilaku sinkronisasi antar host. Dengan karakteristik ini, botnet menjadi sulit terdeteksi karena perilakunya yang menyerupai lalu lintas normal.

Metode deteksi botnet berbasis pembelajaran mesin berfokus pada pemisahan antara lalu lintas normal dan lalu lintas botnet. Model pembelajaran mesin digunakan dalam mengenali perbedaan halus dalam pola komunikasi jaringan, seperti variasi frekuensi, ukuran paket data, maupun durasi aliran data. Penelitian yang dikemukakan oleh (Nadeem et al., 2023) menunjukkan bahwa model pembelajaran mendalam, termasuk autoencoder, memiliki akurasi yang tinggi dalam mengklasifikasikan lalu lintas botnet bahkan saat lalu lintas yang sangat mirip dengan lalu lintas normal.

2.2.4 Dataset CIC-IDS-2018

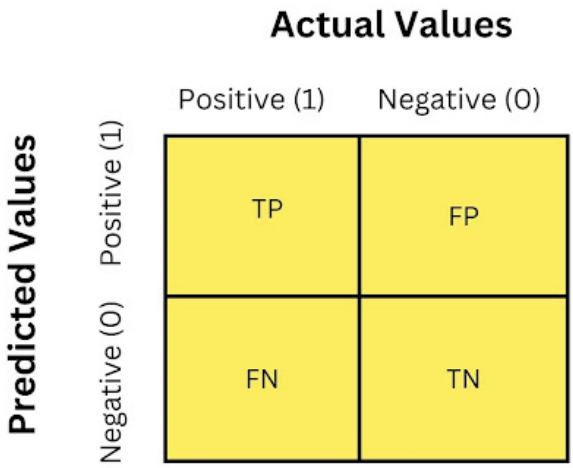
Dataset CIC-IDS-2018 atau CSE-CIC-IDS-2018 merupakan dataset yang dikembangkan secara kolaboratif oleh *Communications Security Establishment* (CSE) dan *Canadian Institute for Cybersecurity* (CIC) sebagai tolok ukur penelitian deteksi intrusi pada jaringan modern. Dataset ini mencakup berbagai jenis serangan, termasuk botnet, DDos, bruteforce, dan infiltration. (Sharafaldin et al., 2018) dalam penelitian yang berjudul "*Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization*" menjadi rujukan dasar dalam perancangan CIC-IDS-2018 yang merupakan representasi lalu lintas jaringan dunia nyata dengan *packet-level capture* dan *flow features*.

Keunggulan dataset CIC-IDS-2018 terdapat pada kelengkapan fitur (80 fitur flow), keseimbangan antara lalu lintas normal dan serangan, serta skenario realistis. Dataset ini mencakup berbagai jenis protokol komunikasi jaringan seperti HOPOPT, TCP, dan UDP serta disimulasikan dengan pengguna nyata, sehingga dapat memberikan gambaran yang lebih akurat dibandingkan dataset lama seperti KDD99 (KDD Cup) atau NSL-KDD. Dataset ini seringkali digunakan sebagai standar penelitian deteksi botnet (Ferrag et al., 2020). Indikator utama dataset CIC-IDS-2018 antara lain:

1. Fitur lalu lintas jaringan yang memberikan dasar representasi untuk model pembelajaran mesin.
2. Label ground truth yang dapat digunakan sebagai acuan dalam evaluasi meskipun model dilatih secara unsupervised.
3. Ketersediaan skenario serangan yang beragam dapat memungkinkan pengujian model terhadap berbagai pola serangan dengan tingkat kompleksitas berbeda.

2.2.5 Confusion Matrix

Confusion matrix adalah matriks berbentuk persegi dimana baris menunjukkan kelas sebenarnya dari instance yang ada dan kolom berfungsi untuk memprediksi kelas (Sathyanarayanan, 2024). Ketika menangani sebuah tugas deteksi lalu lintas botnet maka confusion matrix berupa matriks yang menunjukkan angka true positive (TP), true negative (TN), false positives (FP), dan false negatives (FN) sebagaimana terlihat sebagai berikut:



Gambar 2. 2 Confusion Matrix

Pada Gambar 2.2 merupakan matriks yang mengandung semua baris informasi mengenai prediksi yang telah diperoleh dari proses pelatihan model autoencoder kepada dataset untuk mengevaluasi akurasi secara general dari sebuah model.

Penilaian kinerja sebuah model klasifikasi seringkali menggunakan parameter seperti accuracy, precision, f1-score, dan accuracy. Berikut adalah langkah-langkah untuk mengevaluasi kinerja model klasifikasi:

1. Accuracy

   Accuracy adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar dari total prediksi yang dilakukan. Dalam konteks deteksi lalu lintas botnet, accuracy memberikan gambaran mengenai seberapa sering model memprediksi lalu lintas yang benar, baik itu lalu lintas normal maupun botnet. Berikut persamaannya:

$$\frac{Jumlah\ TP}{Total\ Sample}\ x\ 100\ \% \quad \textbf{(2.5)}$$

2. Precision

   Precision adalah metrik evaluasi yang mengukur seberapa baik model membuat prediksi yang benar untuk kelas positif dari total prediksi positif yang dilakukan. Dalam konteks deteksi lalu lintas botnet, precision memberikan gambaran mengenai seberapa sering model memprediksi kelas bot dengan benar, dari keseluruhan label normal yang ada dalam dataset. Berikut persamaannya:

$$\frac{TP}{TP\ +\ FP} \quad \textbf{(2.6)}$$

3. F1-Score

   F1-score merupakan metrik evaluasi yang mencerminkan keseimbangan antara Presisi (Precision) dan Sensitivitas (Recall). Berikut persamaannya:

$$2\ x\ \frac{Precision\ +\ Recall}{Precision\ -\ Recall} \quad \textbf{(2.7)}$$

4. Recall

   Recall adalah metrik evaluasi yang menggambarkan seberapa baik suatu model dalam mengidentifikasi lalu lintas botnet dengan benar. Berikut persamaannya:

$$\frac{TP}{TP\ +\ FN} \quad \textbf{(2.8)}$$

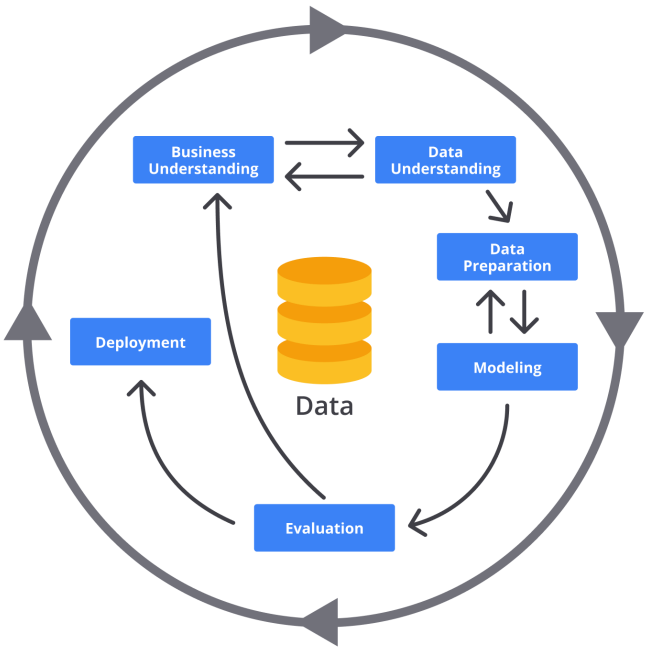
2.2.6 Data Oversampling

Oversampling merupakan teknik prapemrosesan data yang digunakan untuk menangani permasalahan ketidakseimbangan kelas dalam tugas klasifikasi. Secara teoretis, ketidakseimbangan kelas dapat menyebabkan model pembelajaran mesin menjadi bias terhadap kelas mayoritas karena algoritma cenderung meminimalkan kesalahan global tanpa mempertimbangkan representasi kelas minoritas secara proporsional (Mujahid et al., 2024). Oversampling bertujuan meningkatkan jumlah sampel pada kelas minoritas agar distribusi data menjadi lebih seimbang, sehingga estimasi distribusi probabilitas kelas minoritas menjadi lebih representatif dan model terdorong untuk mempelajari pola yang relevan dari kelas tersebut.

Pendekatan oversampling dapat dilakukan melalui duplikasi data maupun pembangkitan sampel sintetis seperti pada metode SMOTE. Menurut (Wongvorachan et al., 2023), metode sintetis dapat dikatakan lebih unggul dibandingkan metode tradisional karena menghasilkan variasi data baru melalui interpolasi pada ruang fitur, sehingga mengurangi risiko *overfitting* dibandingkan replikasi langsung. Dengan distribusi kelas yang lebih seimbang, kontribusi kesalahan kelas minoritas terhadap fungsi kerugian meningkat dan batas keputusan model menjadi lebih adil. Meskipun demikian, efektivitas oversampling sangat bergantung pada karakteristik data dan asumsi bahwa sampel tambahan merepresentasikan distribusi asli, sehingga penerapannya perlu disertai strategi evaluasi dan validasi yang tepat.

2.2.7 CRISP-DM (*Cross-Industry Standard Process for Data Mining*)

CRISP-DM (*Cross-Industry Standard Process for Data Mining*) merupakan suatu kerangka kerja metodologis yang dikembangkan untuk mengarahkan proses analisis data dan pengembangan model pembelajaran mesin secara sistematis, terstruktur, dan berorientasi pada tujuan (Saltz, 2021). Metodologi ini terdiri atas enam tahapan utama, yaitu *business understanding*, *data understanding*, *data preparation*, modeling, *evaluation*, dan *deployment*, yang membentuk suatu siklus iteratif dalam proses penambangan data. Bentuk diagram CRISP-DM ditampilkan sebagai berikut:



Gambar 2. 3 Bentuk CRISP-DM

Tahap *business understanding* berfungsi untuk merumuskan permasalahan penelitian serta menetapkan tujuan analisis yang ingin dicapai, sehingga menjadi dasar konseptual bagi seluruh proses selanjutnya. Tahap *data understanding* kemudian berfokus pada eksplorasi karakteristik data, identifikasi pola awal, serta evaluasi kualitas data. Hasil dari tahap ini menjadi landasan bagi tahap *data preparation*, di mana data dibersihkan, ditransformasi, dan direpresentasikan dalam bentuk yang sesuai untuk kebutuhan pemodelan.

Tahap modeling dalam CRISP-DM melibatkan penerapan algoritma pembelajaran mesin untuk membangun model yang mampu merepresentasikan pola atau struktur yang terkandung dalam data, sedangkan tahap *evaluation* bertujuan untuk menilai kinerja model serta kesesuaiannya dengan tujuan yang telah ditetapkan pada tahap awal. Evaluasi ini tidak hanya mempertimbangkan metrik performa, tetapi juga relevansi hasil terhadap konteks permasalahan yang dikaji. Tahap *deployment* merupakan tahap akhir dari siklus CRISP-DM, yang dalam konteks penelitian ilmiah tidak selalu diwujudkan dalam bentuk aplikasi, melainkan dapat berupa penyajian hasil analisis, interpretasi model, serta penyusunan laporan atau publikasi ilmiah. Menurut Shimaoka et al. (2024) dan Schröer

et al. (2021), dengan karakteristik yang iteratif dan berorientasi pada pemecahan masalah, CRISP-DM menjadi salah satu metodologi yang paling banyak digunakan dalam penelitian *data mining* dan *machine learning* karena mampu mengintegrasikan aspek teknis pemodelan dengan tujuan analitis secara komprehensif.

2.2.8 Normalisasi Data dalam Machine Learning

Normalisasi data dalam machine learning merupakan proses transformasi skala nilai fitur ke dalam rentang tertentu agar setiap variabel memiliki kontribusi yang lebih seimbang dalam proses pembelajaran model. Secara teoretis, banyak algoritma pembelajaran mesin bekerja berdasarkan perhitungan jarak atau optimasi numerik yang sensitif terhadap perbedaan skala antar fitur. Menurut Cabello-Solorzano et al. (2023), apabila suatu fitur memiliki rentang nilai yang jauh lebih besar dibandingkan fitur lainnya, maka fitur tersebut akan mendominasi proses pelatihan dan berpotensi mengaburkan pola penting dari fitur lain. Oleh karena itu, normalisasi bertujuan untuk menyelaraskan skala data sehingga proses komputasi menjadi lebih stabil, efisien, dan tidak bias terhadap fitur tertentu.

Dari sudut pandang matematis, normalisasi dilakukan dengan mengubah distribusi nilai asli ke bentuk baru yang lebih terstandarisasi, misalnya melalui metode seperti *min-max scaling* atau *z-score standardization*. Transformasi ini tidak mengubah makna informasi dalam data, tetapi hanya menyesuaikan representasi numeriknya agar lebih kompatibel dengan kebutuhan algoritma (Ahmed et al., 2022). Normalisasi juga berperan penting dalam mempercepat proses konvergensi pada algoritma berbasis optimasi seperti regresi logistik, neural network, maupun metode berbasis jarak seperti K-Nearest Neighbor dan clustering. Dengan demikian, secara teoretis normalisasi merupakan langkah fundamental dalam prapemrosesan data yang berkontribusi langsung terhadap peningkatan performa, stabilitas, serta kemampuan generalisasi model machine learning. Berikut rumus pendekatan normalisasi yang umum digunakan :

1. Min-Max Scaler

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.9}$$

Di mana :

a. x merupakan nilai asli
b. x_{min} merupakan nilai minimum fitur
c. x_{max} merupakan nilai maksimum fitur
d. x' merupakan nilai hasil normalisasi dengan rentang [0,1]

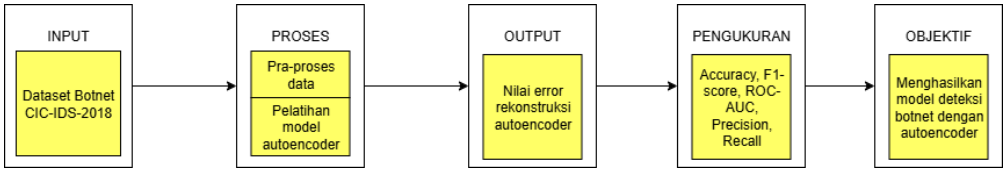
2. Z-score Standardization

$$z = \frac{x - \mu}{\sigma} \quad \textbf{(2.10)}$$

Di mana :

a. μ merupakan rata-rata (mean) fitur
b. σ merupakan standar deviasi
c. z merupakan nilai hasil standardisasi

2.3 Kerangka Pemikiran



Gambar 2. 4 Kerangka Pemikiran

Gambar 2.4 merupakan alur konseptual dari penelitian ini yang menjelaskan hubungan antar komponen utama dalam proses pengolahan data hingga deteksi. Alurnya dijelaskan sebagai berikut:

1. Input (Masukan)

   Dataset yang digunakan adalah dataset CIC-IDS-2018 edisi 03-02-2018 yang berisi lalu lintas jaringan normal dan juga lalu lintas botnet. Data yang digunakan merupakan data mentah (*raw data*) yang memerlukan pemrosesan lebih lanjut dengan fitur seperti Flow Duration, Tot Fwd Pkts, Tot Bwd Pkts, FwdPkt Len Std, BwdPkt Len Std, dan fitur-fitur lainnya.

2. Proses

   Terdapat dua tahapan yang dilakukan yaitu tahapan pra-pemrosesan (*preprocessing*) dan pelatihan model autoencoder. Pada tahapan pra-pemrosesan, dipilih atribut atau fitur penting dari lalu lintas serta dilakukan normalisasi

menggunakan scaling. Selanjutnya pada tahapan pelatihan model, autoencoder akan dilatih untuk merekonstruksi lalu lintas jaringan normal dan model akan diuji efektivitasnya menggunakan lalu lintas campuran untuk mendapatkan nilai rekonstruksi terbaru.

3. Output

   Jika nilai rekonstruksinya rendah, maka dapat dipastikan bahwa autoencoder berhasil merekonstruksi ulang lalu lintas normal. Namun, jika nilai rekonstruksinya tinggi, terindikasi lalu lintas botnet.

4. Evaluasi

   Model autoencoder akan dievaluasi kinerjanya menggunakan confusion matrix seperti akurasi, precision, recall, F1-score, dan ROC-AUC (*Area Under the Receiver Operating Characteristic Curve*).

5. Objektif akhir

   Menghasilkan model deteksi botnet berbasis autoencoder yang dapat membedakan lalu lintas normal dan lalu lintas botnet secara efektif, memberikan kontribusi dalam keamanan sistem jaringan, dan membantu pencegahan serangan siber dengan pendekatan pembelajaran mesin.