BAB II
KAJIAN PUSTAKA

Bab II mengulas tentang perkembangan penelitian deteksi intrusi, posisi penelitian deteksi intrusi pada IoT dan landasan teori yang terkait dengan penelitian deteksi intrusi pada IoT. Metrik evaluasi terhadap penelitian deteksi intrusi juga akan dijelaskan pada bagian ini.

2.1 Tinjauan Pustaka

2.1.1 Penelitian Deteksi Intrusi Menggunakan Seleksi Fitur

Penelitian mengenai deteksi intrusi untuk menghadapi tantangan keamanan jaringan telah berkembang pesat. Salah satu pendekatan yang terus disempurnakan adalah penerapan DFNN (*Deep Feedforward Neural Network*) untuk membedakan antara skenario serangan dan bukan serangan (Cil et al., 2021). Upaya peningkatan kinerja DFNN berfokus pada peningkatan kualitas dataset yang digunakan sebagai masukan model. Seleksi fitur pada dataset dilakukan melalui berbagai metode untuk mempertahankan fitur yang informatif dan membuang fitur yang kurang berguna. Secara umum, terdapat tiga pendekatan utama dalam pemilihan fitur yang digunakan sebagai masukan *classifier*, sehingga model dapat belajar lebih efektif dan akurat yaitu pendekatan berbasis filter, berbasis *wrapper*, dan berbasis *embedded*.

Studi yang menggunakan metode berbasis filter memilih fitur optimal dengan mengukur hubungan statistik antara fitur dan label kelas, tanpa melibatkan model tertentu dalam proses seleksi. Salah satu penelitian relevan, (Zhang et al., 2023) mengusulkan metode FS-DL, yang menggabungkan seleksi fitur dan *deep learning* untuk meningkatkan deteksi intrusi jaringan. Seleksi fitur dalam penelitian ini memanfaatkan teknik standar deviasi dan *association rule mining*. Metode ini dirancang untuk mengidentifikasi serta memilih fitur utama yang berkontribusi pada peningkatan akurasi, pengurangan *redundansi*, dan penurunan beban komputasi. FS-DL meminimalkan jumlah fitur yang digunakan dan mengadopsi struktur neural *network* yang sederhana. Hasil eksperimen pada dataset NSL-KDD

dan UNSW-NB15 menunjukkan peningkatan kinerja deteksi dengan jumlah fitur yang lebih sedikit. FS-DL juga berhasil diimplementasikan pada lingkungan *Software-Defined Networking* (SDN), yang menunjukkan adaptabilitasnya pada berbagai skenario. Namun, meskipun FS-DL menunjukkan kinerja yang kuat pada klasifikasi biner, akurasinya pada klasifikasi multi-kelas masih rendah.

Penelitian oleh (Thakkar & Lohiya, 2023) membahas seleksi fitur pada IDS berbasis DNN dengan tujuan meningkatkan akurasi klasifikasi sekaligus mengurangi jumlah *false alarm*. Dataset yang digunakan dalam penelitian ini adalah NSL-KDD, UNSW-NB15, dan CIC-IDS-2017. Metodologi penelitian mencakup penggunaan standar deviasi, mean, dan median untuk mengidentifikasi fitur paling relevan. Fitur yang terpilih kemudian digunakan pada DFNN untuk deteksi dan klasifikasi intrusi. Hasil penelitian menunjukkan bahwa pendekatan statistik dalam seleksi fitur dapat secara efektif meningkatkan deteksi intrusi. Walau demikian, penelitian ini menekankan perlunya evaluasi lebih lanjut terhadap algoritma yang digunakan pada dataset yang lebih beragam.

Mengatasi masalah redundansi fitur dan ketidakseimbangan kelas pada deteksi intrusi IoT, (Ma et al., 2025) mengusulkan kerangka FSLLM. Metode ini memanfaatkan seleksi fitur berbasis mRMR yang diperkuat Pearson *Correlation* untuk mengurangi fitur berlebih, lalu menggunakan LightGBM dengan fungsi *Focal Loss* sebagai pengklasifikasi. Uji coba pada lima dataset IoT menunjukkan reduksi fitur lebih dari 80% tanpa menurunkan akurasi. Dengan demikian, kombinasi seleksi fitur berbasis Pearson dan LightGBM terbukti efektif dan efisien. Namun penelitian ini masih terbatas pada kebutuhan komputasi tinggi untuk penerapan nyata.

Pendekatan kedua untuk seleksi fitur adalah metode *wrapper*. Teknik ini memilih fitur dengan mengevaluasi langsung kinerja model untuk setiap kombinasi fitur, sehingga fitur dipilih berdasarkan dampaknya terhadap akurasi model. Penelitian oleh (Kasongo & Sun, 2020) menerapkan *Wrapper-Based Feature Extraction Unit* (WFEU) dengan menggunakan *Extra Trees classification* untuk meningkatkan kemampuan deteksi DFNN. Penelitian ini ditujukan untuk mengatasi

keterbatasan sistem deteksi intrusi nirkabel. Efisiensi dan akurasi deteksi intrusi dalam jaringan nirkabel sangat penting bagi keamanan jaringan di era Internet of Things (IoT). Pada pengujian dengan dataset UNSW-NB15, WFEU menghasilkan vektor fitur optimal dengan 22 atribut, mencapai akurasi 87,10% (biner) dan 77,16% (multi-kelas). Sementara pada dataset AWID, vektor fitur berkurang menjadi 26 atribut dengan akurasi 99,66% (biner) dan 99,77% (multi-kelas). Banyaknya fitur dalam proses pelatihan dapat menyebabkan waktu pemrosesan lama dan tuntutan komputasi yang besar. Karena itu, reduksi dimensi menjadi penting dalam tahap prapengolahan.

Penelitian oleh (H. S. Sharma & Singh, 2024) memperkenalkan model DFNN yang lebih maju, dengan mengintegrasikan teknik seleksi fitur untuk mengatasi tantangan keamanan yang meningkat di cloud computing. Fitur yang dimasukkan ke dalam *classifier* DFNN dipilih menggunakan *Extra Tree Classifier*, dengan tujuan meningkatkan akurasi dan efisiensi sistem deteksi intrusi. Model DFNN mencapai akurasi puncak 99,53% pada NSL-KDD, 94,45% pada UNSW-NB15, dan 99,80% pada CIC-IDS 2017. Kekurangan penelitian ini adalah beban komputasi yang tinggi jika diterapkan pada skenario *real-time*, akibat kompleksitas deep learning dan proses pengolahan data yang besar dalam seleksi fitur.

Penelitian (Swarna Priya et al., 2020) menggabungkan *Principal Component Analysis* (PCA) dan *Grey Wolf Optimization* (GWO) untuk rekayasa fitur yang digunakan sebagai masukan DFNN *classifier*. PCA digunakan untuk mengurangi dimensi data dengan menyederhanakan ruang fitur tanpa menghilangkan informasi penting. GWO, yang terinspirasi dari perilaku berburu serigala abu-abu dalam hierarki sosialnya, digunakan untuk meningkatkan seleksi fitur. Hasil penelitian ini menunjukkan peningkatan akurasi klasifikasi sebesar 15% dan pengurangan waktu pelatihan sebesar 32% dibandingkan metode machine learning lainnya seperti KNN, Naive Bayes, Random Forest, SVM, dan DFNN tanpa PCA-GWO.

Eksplorasi oleh (Tseng & Chang, 2023) bertujuan meningkatkan deteksi intrusi pada jaringan Wi-Fi dengan mengusulkan *Ensemble Binary Detection*

Models (EBDM). EBDM mengubah deteksi multi-kelas menjadi model deteksi biner dengan menggunakan *Binary Model* (BM) khusus untuk setiap kelas target. Pendekatan ini mengintegrasikan *Feature Selection* (FS), *Stack Auto Encoder* (SAE), dan *Random Sampling* (RS) untuk meningkatkan kinerja deteksi, serta memperkenalkan Ensemble FS (EFS) dan Ensemble DB untuk meningkatkan akurasi. Hasil eksperimen menunjukkan bahwa EBDM lebih akurat dalam mendeteksi tiga jenis serangan nirkabel dibandingkan pendekatan sebelumnya, khususnya pada dataset AWID, dengan akurasi 99,625% dan F1 Score 99,250%. Penelitian (Zare & Mahmoudi-Nasr, 2023) membahas pemilihan metode *Dimension Reduction* (DR) menggunakan LDA, t-SNE, dan PCA dengan DFNN sebagai classifier. Hasil penelitian menunjukkan bahwa LDA memiliki waktu pelatihan lebih cepat dibandingkan PCA, sementara t-SNE paling lambat, tetapi akurasinya lebih tinggi. Dari tiga metode seleksi fitur (*filter, wrapper, embedded*), metode *wrapper* memberikan akurasi lebih tinggi meskipun dengan biaya komputasi lebih besar.

Pendekatan ketiga adalah metode *embedded*, di mana seleksi fitur terjadi selama proses pelatihan model dengan memprioritaskan fitur yang paling relevan berdasarkan bobot atau kontribusinya terhadap hasil klasifikasi. Misalnya, penelitian (Thakkar & Lohiya, 2021b) mengevaluasi kombinasi teknik regularisasi seperti L1, L2, elastic net, dan dropout pada IDS berbasis DFNN. Hasilnya menunjukkan bahwa *dropout* lebih efektif dibandingkan teknik lainnya, dan kombinasi *dropout* dengan regularisasi lain memberikan hasil terbaik. Penelitian lain, seperti (Kunang et al., 2021), menggunakan Deep Autoencoder (DAE) untuk seleksi fitur, serta (Lu et al., 2024) yang mengusulkan algoritma Greedy Recursive Feature Elimination with Cross-Validation (GRFECV), yang terbukti lebih unggul dibandingkan RFECV meski membutuhkan sumber daya komputasi yang besar.

Dari ketiga pendekatan tersebut, metode berbasis filter menonjol karena konsumsi daya rendah, sebab tidak memerlukan keterlibatan model selama proses seleksi fitur (Albulayhi et al., 2022). Studi lanjutan menerapkan pendekatan ini hanya dengan satu metode statistik, misalnya Pearson correlation (Henry et al.,

2023), yang memilih sekitar 40 dari 77 fitur pada dataset CICIDS 2017, menghasilkan peningkatan kinerja model. Penelitian lain oleh (B. Sharma et al., 2023) mengusulkan kombinasi Pearson correlation untuk seleksi fitur dan GAN *(Generative Adversarial Networks)* untuk mengatasi ketidakseimbangan kelas pada dataset UNSW-NB15, dengan classifier DFNN. Hasilnya, akurasi deteksi meningkat dari 84% menjadi 91%. Namun, kelemahan penggunaan satu metode statistik seperti Pearson *correlation* adalah ketidakmampuannya menangkap hubungan non-linear. Oleh karena itu, terdapat peluang penelitian untuk menggabungkan beberapa metode filter secara bersamaan.

2.1.2 Posisi Penelitian Deteksi Intrusi pada IoT

Berdasarkan penelitian yang telah dipaparkan pada bab sebelumnya, maka dibuat pemetaan posisi penelitian ini sebagaimana pada Gambar 2.1. Pada gambar ditunjukkan posisi yang membedakan penelitian yang dilakukan dengan penelitian yang telah ada sebelumnya.

		Seleksi Fitur berbasis filter pada deteksi intrusi IoT		
		Pearson Correlation	Spearman Correlation	Mutual Information
Machine Learning	DT	Jing Li, 2024		
	k-NN			Al Sarem, 2021
	KELM		Gavel, 2022	
Deep Learning	DFNN	Sharma, 2023	Ikhwan, 2025	
	CNN	Henry, 2023		
	LightGBM	Huan Ma, 2025		
	SAE		Dao, 2022	

Gambar 2.1 Posisi Penelitian

Penelitian (J. Li, Chen, et al., 2024) mengevaluasi dan membandingkan berbagai teknik reduksi fitur, yaitu Feature Selection (FS) dan Feature Extraction (FE), dalam sistem Intrusion Detection System (IDS) untuk IoT dengan menggunakan dua dataset publik yang berbeda. Analisis dilakukan untuk menilai performa FS dan FE pada berbagai model machine learning, sehingga diperoleh wawasan mengenai efektivitas masing-masing metode dalam konteks IoT. Metode FS yang digunakan adalah Korelasi Pearson dan Chi-Square. (Al-Sarem et al., 2022) mengusulkan metode FS menggunakan MI dengan klasifikasi yang digunakan adalah k-NN. Sementara itu (Gavel et al., 2022) menggunakan korelasi Spearman dan MI secara berurutan pada tahap prapengolahan dengan klasifikasi KELM. Seleksi fitur menggunakan korelasi Pearson dilakukan oleh (B. Sharma et al., 2023), (Henry et al., 2023) dan (Ma et al., 2025) ketiganya menggunakan deep learning sebagai pengklasifikasi. Secara berurutan, deep learning yang digunakan adalah DFNN, CNN dan LightGBM. Penelitian berikutnya menggunakan korelasi Spearman dilakukan oleh (Dao & Lee, 2022) dengan pengklasifikasi yang digunakan adalah *Stacked Autoencoder* (SAE).

Penelitian terdahulu menunjukkan bahwa seleksi fitur berbasis filter, seperti Pearson Correlation, Spearman Correlation, dan Mutual Information (MI), dapat meningkatkan kinerja pengklasifikasi dalam membedakan serangan dan non-serangan. Berdasarkan temuan tersebut, penelitian ini mengusulkan pendekatan yang mengombinasikan ketiga metode tersebut untuk mengembangkan IDS yang lebih efektif, ringan, dan sesuai untuk penerapan praktis di lingkungan IoT. Selanjutnya, pemilihan metode DFNN digunakan berdasarkan penelitian yang dilakukan oleh peneliti sebelumnya juga bahwa DFNN memiliki keunggulan di antaranya adalah DFNN menunjukkan performa yang lebih baik daripada algoritma *machine learning* yang ada (B. Sharma et al., 2023), DFNN memiliki akurasi yang lebih baik dan penurunan kompleksitas waktu (H. S. Sharma & Singh, 2024), melakukan pembelajaran implisit pada data berdimensi tinggi dengan mudah (Swarna Priya et al., 2020), dan kemampuan generalisasi yang lebih baik (Thakkar & Lohiya, 2023).

2.2 Landasan Teori

2.2.1 Teori *Ensemble feature Selection*

Seleksi fitur diperlukan untuk mereduksi dimensi. Reduksi dimensi ini sangat dibutuhkan untuk menghindari *curse of dimensionality* (kutukan dimensi) (Ye et al., 2024), (Salman et al., 2026). Istilah *Curse of dimensionality* pertama kali diperkenalkan oleh Richard Bellman yang menggambarkan banyaknya masalah yang muncul ketika bekerja dengan data berdimensi tinggi (banyak fitur dan atribut) (Mirkes et al., 2020). Semakin tinggi dimensi suatu data, maka semakin kompleks dan sulit analisisnya. Data menjadi sangat jarang dan pola yang tampak jelas pada data dimensi rendah, menjadi kabur pada dimensi tinggi. Beberapa masalah pada data dengan dimensi yang terlalu tinggi di antaranya adalah bisa merusak performa algoritma, membuat data menjadi jarang, jarak tidak bermakna, dan model menjadi *overfit*. Oleh karena itu untuk mengurangi dimensi yang tinggi diperlukan proses seleksi fitur untuk mengurangi fitur-fitur yang kurang berkontribusi.

Ensemble feature selection merupakan salah satu pendekatan dalam *machine learning* untuk proses seleksi fitur dengan cara mengombinasikan hasil dari berbagai metode filter yang berbeda, sehingga diperoleh sub set fitur yang lebih konsisten, kuat, dan memiliki tingkat akurasi yang lebih baik. Pendekatan *ensemble* ini berlandaskan pada prinsip "*Wisdom of the Crowd*", yang menyatakan bahwa penggabungan beberapa model independen mampu mengurangi bias dan varians secara simultan (Khaire & Dhanalakshmi, 2022). Secara konseptual, teori *ensemble* memiliki akar pada *Condorcet's Jury Theorem* yang diusulkan oleh Marquis de Condorcet pada tahun 1785 (Rokach & Maimon, 2015). Teorema ini menyatakan bahwa ketika seorang juri yang terdiri dari *T* pemilih harus mengambil keputusan melalui pemungutan suara mayoritas, jika probabilitas setiap pemilih memberikan jawaban yang benar adalah p dan probabilitas keputusan akhir benar adalah $p*$, maka berlaku ketentuan berikut:

Jika $p > 0{,}5$, maka $p* > p$

Jika jumlah suara mendekati tak terhingga, maka $p*$ akan mendekati 1,0 selama $p > 0{,}5$.

Terdapat dua prasyarat utama agar teori ini berlaku yaitu antar pemberi suara harus independen dan batasan hasil pada dua kategori luaran saja. Logika ini selaras dengan mekanisme *ensemble* independen dalam konteks deteksi intrusi. Relevansi tersebut muncul karena seluruh algoritma pendeteksi dalam sistem bekerja secara independen, dan hasil akhir dari proses identifikasi tersebut bermuara pada klasifikasi biner, yakni penentuan status suatu objek sebagai intrusi atau bukan (Zhao, 2017). Sehingga, dalam konteks *machine learning*, pendekatan *ensemble learning* memanfaatkan kekuatan kolektif dari berbagai model untuk menghasilkan sistem pembelajaran yang lebih andal (*robust*) serta mampu meningkatkan performa prediksi dibandingkan penggunaan satu model tunggal.

Penelitian (Kuncheva & Whitaker, 2003) menjelaskan bahwa keberhasilan metode *ensemble*, tidak hanya ditentukan oleh jumlah pengklasifikasi yang digunakan, tetapi juga oleh tingkat keragaman atau keberagaman antar anggota *ensemble*. Pengklasifikasi yang memiliki karakteristik terlalu mirip cenderung menghasilkan kesalahan pada objek yang sama, sehingga kontribusinya terhadap peningkatan akurasi menjadi terbatas. Sebaliknya, pengklasifikasi yang beragam dan saling melengkapi berpotensi memperbaiki hasil prediksi karena kesalahan satu pengklasifikasi dapat di kompensasi oleh pengklasifikasi lainnya.

Keberhasilan metode ensemble pada pengklasifikasi kemudian menjadi dasar untuk pengembangan metode tersebut pada pemilihan fitur (Saeys et al., 2008). Kerangka kerja *ensemble* pada seleksi fitur umumnya terdiri atas dua tahapan utama. Tahap pertama adalah penerapan pendekatan diversifikasi untuk menghasilkan beragam keluaran seleksi fitur dari beberapa metode yang berbeda. Selanjutnya, pada tahap kedua digunakan strategi agregasi untuk menggabungkan seluruh hasil seleksi tersebut sehingga diperoleh himpunan fitur yang lebih optimal dan stabil (Salman et al., 2026).

Dalam konteks penelitian ini, diversifikasi dilakukan melalui penggunaan tiga metode seleksi fitur yang memiliki karakteristik statistik berbeda. Korelasi Pearson digunakan untuk menangkap hubungan linear antar variabel, Korelasi Spearman digunakan untuk mengidentifikasi hubungan *monotonik* berbasis peringkat,

sedangkan Mutual Information digunakan untuk mengukur ketergantungan non-linear berbasis teori informasi. Kombinasi ketiganya diharapkan mampu menghasilkan himpunan fitur yang lebih representatif dibandingkan penggunaan satu metode tunggal karena setiap metode berkontribusi menangkap aspek hubungan data yang berbeda.

2.2.2 Teori Dasar Korelasi Linier

Teori korelasi linear adalah konsep atau teori statistik yang menjelaskan hubungan searah atau berlawanan arah antara dua variabel numerik dalam bentuk hubungan garis lurus. Korelasi linier digunakan untuk mengetahui apakah perubahan pada satu variabel berkaitan dengan perubahan pada variabel lain. konsep korelasi linier didukung oleh teorema matematika ketaksamaan Cauchy–Schwarz, yang menjelaskan batas nilai koefisien korelasi. Teorema Ketaksamaan Cauchy–Schwarz adalah teorema dasar dalam aljabar linear dan analisis matematika yang menyatakan bahwa nilai mutlak hasil perkalian dalam dua vektor tidak akan pernah melebihi hasil kali panjang kedua vektor tersebut. Teorema ini bermula dari bentuk jumlah berhingga yang diperkenalkan oleh Cauchy pada tahun 1821, diperluas oleh Bunyakovsky ke bentuk integral pada tahun 1859, dan diperkuat melalui pembuktian modern oleh Schwarz pada tahun 1888 (Sedrakyan & Sedrakyan, 2018). Teorema ini sangat penting karena menjadi dasar matematis bagi banyak konsep, termasuk korelasi Pearson, *cosine similarity*, regresi linier, seleksi fitur, dan analisis kemiripan data. Dalam konteks korelasi linier, teorema ini menjelaskan mengapa nilai korelasi selalu berada dalam rentang −1 sampai +1, sehingga korelasi dapat diinterpretasikan secara konsisten sebagai ukuran arah dan kekuatan hubungan linier antar variabel.

Teorema Ketaksamaan Cauchy–Schwarz ini menjelaskan bahwa hubungan antara dua vektor memiliki batas maksimum tertentu. Dalam bentuk umum:

$$|\langle x, y\rangle| \leq \|x\| \cdot \|y\| \tag{2.1}$$

$\langle x, y\rangle$ adalah inner product atau hasil kali dalam antara vektor (x) dan (y). Sedangkan $\|x\|$ adalah panjang atau norma dari vektor (x).

Misalkan terdapat dua vektor:

$$x = (x_1, x_2, \dots, x_n)$$
$$y = (y_1, y_2, \dots, y_n)$$

Maka bentuk Cauchy–Schwarz adalah:

$$\left(\sum_{i=1}^{n} x_i y_i\right)^2 \leq \left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i^2\right) \tag{2.2}$$

atau dalam bentuk nilai mutlak:

$$\left|\sum_{i=1}^{n} x_i y_i\right| \leq \sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2} \tag{2.3}$$

Teorema ini menyatakan bahwa hasil perkalian antara dua vektor selalu dibatasi oleh hasil kali panjang masing-masing vektor.

Teorema Cauchy–Schwarz sangat penting dalam teori korelasi linier, terutama untuk membuktikan bahwa nilai koefisien korelasi Pearson selalu berada pada interval:

$$-1 \leq r \leq 1$$

Koefisien korelasi Pearson ditulis sebagai:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Jika didefinisikan:

$$a_i = x_i - \bar{x}$$
$$b_i = y_i - \bar{y}$$

maka rumus korelasi Pearson menjadi:

$$r = \frac{\sum a_i b_i}{\sqrt{\sum a_i^2}\sqrt{\sum b_i^2}}$$

Berdasarkan Cauchy–Schwarz:

$$\left|\sum a_i b_i\right| \leq \sqrt{\sum a_i^2}\sqrt{\sum b_i^2}$$

Maka:

$$|r| \leq 1$$

Sehingga:

$$-1 \leq r \leq 1$$

Karenanya, Teorema Cauchy–Schwarz menjadi dasar matematis mengapa nilai korelasi Pearson tidak mungkin lebih kecil dari -1 atau lebih besar dari +1.

2.2.3 Teori Pengukuran Asosiasi Berbasis Peringkat

Dalam bidang statistika, asosiasi merujuk pada kecenderungan adanya hubungan antara dua variabel. Apabila perubahan pada variabel X diikuti oleh perubahan tertentu pada variabel Y, maka kedua variabel tersebut dianggap memiliki hubungan asosiasi (Alshahrani, 2026). Meskipun demikian, asosiasi tidak secara langsung menunjukkan adanya hubungan sebab-akibat. Pada analisis korelasi, fokus pengukuran terletak pada tingkat kekuatan hubungan serta arah pergerakan kedua variabel secara bersamaan.

Hubungan antar variabel dianalisis berdasarkan nilai asli dari data pada Korelasi Pearson, sehingga metode ini lebih sesuai digunakan ketika pola hubungan yang terbentuk bersifat linier. Di sisi lain, Korelasi Spearman mengukur tingkat asosiasi berdasarkan peringkat atau urutan data, sehingga lebih efektif untuk hubungan yang tidak linier tetapi masih menunjukkan kecenderungan pola tertentu. Konsep korelasi berbasis peringkat pertama kali diperkenalkan oleh Charles Spearman (Spearman, 1904) melalui artikelnya pada tahun 1904 yang berjudul *The Proof and Measurement of Association between Two Things*. Dalam publikasi tersebut, Spearman menjelaskan bahwa pendekatan korelasi berbasis peringkat

sangat bermanfaat ketika data sulit diukur secara absolut maupun saat jumlah sampel yang tersedia relatif terbatas.

Peringkat adalah posisi relatif suatu nilai dibandingkan nilai lainnya. Misalnya, data: $X = [10,30,20,50,40]$ dapat diubah menjadi peringkat $R(X) = [1,3,2,5,4]$ artinya, nilai 10 adalah yang paling kecil sehingga mendapat peringkat 1, nilai 20 mendapat peringkat 2, dan seterusnya. Dalam skala ordinal, angka tidak menunjukkan jarak absolut antar objek, tetapi menunjukkan urutan.

2.2.4 Statistik Nonparametrik

Statistik non parametrik adalah metode statistik yang tidak mensyaratkan bentuk distribusi tertentu, misalnya distribusi normal (Alshahrani, 2026). NIST menjelaskan bahwa prosedur nonparametrik adalah prosedur yang tidak berfokus pada parameter distribusi tertentu dan dapat digunakan ketika data berskala nominal atau ordinal. Dalam statistik parametrik, analisis biasanya melibatkan asumsi tertentu, misalnya data terdistribusi normal, hubungan linier, varians homogen, atau parameter populasi seperti rata-rata dan varians. Sebaliknya, statistik nonparametrik lebih fleksibel karena dapat digunakan ketika asumsi-asumsi tersebut tidak terpenuhi.

2.2.5 Teori Hubungan Monotonik Antar Variabel

Hubungan monotonik adalah hubungan ketika kenaikan suatu variabel cenderung diikuti oleh kenaikan atau penurunan variabel lain secara konsisten . Hubungan monotonik tidak harus membentuk garis lurus. Ada dua jenis hubungan monotonik, pertama monotonik positif jika X naik, Y cenderung naik. Kedua monotonik negatif jika X naik, Y cenderung turun (Yu & Hutson, 2024). Hubungan linier berarti hubungan antar variabel dapat digambarkan dengan garis lurus. Misalnya: $Y = 2X + 1$. Namun hubungan monotonik tidak harus lurus. Sebagai contoh $Y = X^2, X > 0$, pada domain $X > 0$, semakin besar X, semakin besar Y, tetapi bentuk hubungannya melengkung, bukan garis lurus.

2.2.6 Teori Informasi

Teori Informasi adalah cabang matematika terapan yang mempelajari bagaimana informasi diukur, dikodekan, dikirimkan, disimpan, dikompresi, dan dipulihkan dari gangguan atau ketidakpastian. Dalam konteks *data mining* dan *machine learning*, teori informasi digunakan untuk mengukur seberapa besar suatu fitur mengandung informasi terhadap target/kelas. Konsep ini sangat penting dalam seleksi fitur karena fitur yang baik adalah fitur yang dapat mengurangi ketidakpastian terhadap kelas atau label. Dasar modern teori informasi diletakkan oleh Claude E. Shannon melalui artikel klasik *A Mathematical Theory of Communication* pada tahun 1948. Secara sederhana, teori informasi menjawab pertanyaan "Seberapa banyak informasi yang diberikan oleh suatu variabel terhadap variabel lain?". Dalam seleksi fitur, pertanyaannya menjadi "Seberapa besar informasi yang diberikan oleh fitur (X_i) terhadap label kelas (Y)?".

Sebelum Shannon, gagasan tentang pengukuran informasi telah muncul melalui karya Harry Nyquist dan Ralph Hartley, terutama dalam konteks transmisi sinyal dan kapasitas komunikasi. Shannon kemudian menyatukan gagasan tersebut ke dalam kerangka matematis yang lengkap melalui konsep *entropy, mutual information,* dan *channel capacity.* Artikel Shannon tahun 1948 menjelaskan komunikasi sebagai sistem yang terdiri atas sumber informasi, *encoder*, kanal, *noise, decoder*, dan penerima. Perkembangan berikutnya diperluas dalam buku-buku klasik seperti Cover dan Thomas, *Elements of Information Theory*, yang membahas *entropy*, mutual information, *data compression, channel capacity, rate distortion, network information theory*, dan hubungan teori informasi dengan statistik. Dalam machine learning, teori informasi kemudian digunakan untuk klasifikasi, pembelajaran statistik, kompresi data, pengenalan pola, dan seleksi fitur. MacKay juga menghubungkan teori informasi dengan inferensi, *learning algorithms, Bayesian modelling*, dan neural networks.

Konsep dasar dalam teori informasi berkaitan dengan seberapa "mengejutkan" sebuah kejadian. Jika suatu kejadian sangat jarang terjadi, maka kejadian tersebut membawa informasi yang lebih besar. Misalkan jumlah informasi

dari kejadian x adalah $I(x)$, probabilitas kejadian x adalah $P(x)$ dan b adalah basis logaritma, maka dapat dirumuskan:

$$I(x) = -\log_b P\,(x) \tag{2.4}$$

Contoh sederhana dari kejadian ini misalnya jika suatu kelas serangan dalam dataset sangat jarang muncul, maka kemunculan kelas tersebut membawa informasi lebih besar dibanding kelas normal yang sering muncul.

Pada teori informasi dikenal *entropy*. *Entropy* mengukur tingkat ketidakpastian dalam suatu variabel acak. Semakin tinggi *entropy*, semakin besar ketidakpastian data. *Entropy* dirumuskan:

$$H(X) = -\sum_{x \in X} P(x) \log_2 P\,(x) \tag{2.5}$$

dimana $H(X)$ adalah *entropy* variabel X, $P(x)$ *adalah* probabilitas nilai x, dan $\log_2$ adalah logaritma basis 2 dengan satuan bit. Jika semua nilai dalam variabel memiliki peluang yang hampir sama, maka *entropy* tinggi. Namun jika satu nilai sangat dominan maka *entropy* menjadi rendah. Dalam seleksi fitur, *entropy* penting karena fitur yang informatif adalah fitur yang mampu mengurangi ketidakpastian terhadap label kelas.

2.2.7 Algoritma korelasi Pearson

Algoritma korelasi Pearson, atau dikenal sebagai *Pearson's correlation coefficient* (PCC) disimbolkan dengan r, adalah metode statistik yang digunakan untuk mengukur kekuatan dan arah hubungan linear antara dua variabel kuantitatif. Algoritma ini sangat berguna dalam berbagai disiplin ilmu, termasuk ilmu sosial, kedokteran, dan ekonomi, untuk mengidentifikasi sejauh mana perubahan dalam satu variabel terkait dengan perubahan dalam variabel lainnya. Koefisien korelasi pearson dapat digunakan untuk mengevaluasi kemiripan dari dua fitur kemudian menentukan penghargaan pada parameter untuk mempercepat konvergensi (Zhu et al., 2019)

Algoritma Pearson dikembangkan oleh Karl Pearson (1857–1936), seorang ahli matematika dan statistika asal Inggris. Sebagai salah satu pelopor dalam statistik matematika, Pearson memainkan peran penting dalam perkembangan metode statistik modern. Ia merumuskan korelasi Pearson dalam makalahnya tahun 1895 berjudul *"Notes on Regression and Inheritance in the Case of Two Parents"*.

Pearson's correlation coefficient adalah nilai antara -1 dan 1:

- Nilai +1 menunjukkan korelasi positif sempurna, di mana kedua variabel meningkat secara bersamaan dalam garis lurus.
- Nilai -1 menunjukkan korelasi negatif sempurna, di mana satu variabel meningkat sementara yang lain menurun secara linear.
- Nilai 0 menunjukkan tidak adanya korelasi linear antara kedua variabel.

Rumus untuk menghitung Pearson's r adalah:

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \quad (2.6)$$

Di mana:

- x_i dan y_i adalah nilai individu dari dua variabel.
- $\bar{x}$ dan $\bar{y}$ adalah rata-rata dari masing-masing variabel.

Korelasi Pearson sering digunakan dalam analisis data eksploratif untuk menentukan hubungan awal antara variabel. Jika koefisien korelasi menunjukkan hubungan yang signifikan, analisis lebih lanjut biasanya dilakukan untuk memahami sifat dan kekuatan hubungan tersebut.

Keterbatasan pada algoritma Pearson di antaranya adalah:

- Pearson hanya mengukur hubungan linear dan tidak efektif untuk hubungan non-linear.
- Sensitif terhadap *outlier*, yang dapat mempengaruhi nilai korelasi secara signifikan.
- Tidak mengindikasikan sebab-akibat (kausalitas) dan hanya hubungan.

2.2.8 Korelasi Spearman

Korelasi Spearman lahir dari kebutuhan untuk mengukur hubungan antara dua variabel ketika hubungan tersebut tidak selalu berbentuk linier. Korelasi Spearman, juga dikenal sebagai *Spearman's rank correlation coefficient* atau Spearman's rho merupakan cara untuk menilai hubungan antara dua variabel yang diungkapkan dalam bentuk peringkat (Dao & Lee, 2022) (Kou et al., 2012). Metode ini diperkenalkan oleh Charles Edward Spearman dalam artikel klasik berjudul *The Proof and Measurement of Association between Two Things* yang terbit pada tahun 1904 di *The American Journal of Psychology*. Artikel ini kemudian diterbitkan ulang dalam *International Journal of Epidemiology* pada tahun 2010 karena dianggap sebagai salah satu karya penting dalam sejarah pengukuran asosiasi statistik. Korelasi Spearman mengevaluasi hubungan monotonik antara dua variabel (Gavel et al., 2022), yang berarti bahwa jika satu variabel naik, variabel lainnya cenderung naik atau turun secara konsisten. Korelasi ini cocok digunakan untuk data yang tidak terdistribusi normal atau memiliki hubungan non-linear.

Secara teoretis, korelasi Spearman menggunakan tiga dasar utama, yaitu teori korelasi, teori berbasis peringkat, dan statistik nonparametrik. Teori korelasi menjelaskan ukuran asosiasi antara dua variabel; teori peringkat mengubah nilai asli menjadi posisi relatif; sedangkan statistik nonparametrik memungkinkan analisis tanpa asumsi distribusi normal. Inilah alasan Spearman sering digunakan ketika data mengandung *outlier*, skala ordinal, distribusi tidak normal, atau hubungan yang tidak sepenuhnya linier.

Rumus yang digunakan untuk menentukan nilai koefisien korelasi Spearman rho adalah :

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.7}$$

di mana ρ adalah koefisien korelasi Spearman, d_i adalah selisih peringkat dari setiap pasangan observasi. n adalah jumlah observasi.

Nilai ρ bernilai positif satu (+1) menandakan hubungan monotonik sempurna positif (ketika satu variabel meningkat, variabel lain juga meningkat), nilai ρ -1 menandakan hubungan monotonik sempurna negatif (ketika satu variabel

meningkat, variabel lain menurun) sedangkan nilai 0 menyatakan bahwa tidak ada hubungan monotonik antara dua variabel.

2.2.9 Mutual Information

Mutual Information (MI) adalah konsep dalam teori informasi yang mengukur ketergantungan antara dua variabel acak. Dalam konteks algoritma, Mutual Information digunakan untuk menentukan seberapa banyak informasi yang diperoleh tentang satu variabel dengan mengetahui variabel lainnya (S. Gupta & Singh, 2026). Nilai mutual information menunjukkan seberapa besar ketergantungan antara kedua variabel tersebut: semakin tinggi nilainya, semakin besar ketergantungannya (Gavel et al., 2022).

Teori Informasi merupakan dasar matematis untuk seleksi fitur berbasis Mutual Information. *Entropy* digunakan untuk mengukur ketidakpastian, sedangkan Mutual Information digunakan untuk mengukur pengurangan ketidakpastian target setelah suatu fitur diketahui. Dalam seleksi fitur, fitur dengan nilai MI tinggi dianggap lebih informatif terhadap label.

Mutual Information antara dua variabel acak X dan Y didefinisikan sebagai:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) \tag{2.8}$$

di mana:

- $p(x)p(y)$ adalah distribusi bersama dari X dan Y.
- $p(x)$ dan $p(y)$ adalah distribusi marginal dari masing-masing variabel.
- log biasanya menggunakan basis 2 (untuk bit) atau basis e (untuk nat).

Mutual Information memiliki ketentuan:

- $I(X;Y) = 0$ jika dan hanya jika X dan Y independen.

Semakin besar nilai $I(X;Y)$, semakin kuat hubungan antara X dan Y.

2.2.10 Konsep Metode SPADEH

SPADEH (*Selective Predictive Attack Detection using Ensemble Hybridization*) merupakan metode *ensemble based feature selection* yang diusulkan

dalam penelitian ini untuk meningkatkan kualitas fitur masukan pada proses deteksi intrusi berbasis *Deep Feedforward Neural Network* (DFNN). Metode ini dikembangkan berdasarkan asumsi bahwa karakteristik data lalu lintas jaringan memiliki pola hubungan yang beragam, sehingga tidak dapat direpresentasikan secara optimal hanya dengan menggunakan satu metode seleksi fitur.

Pada penelitian-penelitian sebelumnya, seleksi fitur umumnya dilakukan menggunakan pendekatan tunggal, seperti *Pearson Correlation, Spearman Correlation,* atau *Mutual Information*. Meskipun masing-masing metode memiliki keunggulan, setiap metode hanya mampu merepresentasikan jenis hubungan tertentu yang terdapat dalam data. Akibatnya, terdapat kemungkinan informasi penting tidak teridentifikasi ketika proses seleksi fitur hanya bergantung pada satu pendekatan.

Korelasi Pearson digunakan untuk mengukur kekuatan hubungan linear antara fitur dan variabel target. Metode ini efektif dalam mengidentifikasi fitur-fitur yang memiliki keterkaitan linear yang kuat terhadap kelas keluaran. Namun demikian, *Pearson Correlation* memiliki keterbatasan dalam mendeteksi hubungan non-linear maupun hubungan monotonik yang tidak bersifat linear. Di sisi lain, Korelasi Spearman merupakan metode non-parametrik yang digunakan untuk mengukur hubungan monotonik antar variabel. Metode ini mampu mengidentifikasi pola hubungan yang tidak selalu linear, tetapi masih menunjukkan kecenderungan peningkatan atau penurunan secara konsisten. Dengan demikian, *Spearman Correlation* dapat melengkapi keterbatasan *Pearson Correlation* dalam menangkap karakteristik data yang lebih kompleks. Selain kedua metode tersebut, penelitian ini juga memanfaatkan Mutual Information yang berasal dari teori informasi Shannon. Mutual Information digunakan untuk mengukur tingkat ketergantungan antara fitur dan variabel target tanpa mengasumsikan bentuk hubungan tertentu. Oleh karena itu, metode ini mampu mendeteksi hubungan non-linear yang sering ditemukan pada data lalu lintas jaringan dan aktivitas serangan siber.

Berdasarkan karakteristik tersebut, dapat disimpulkan bahwa *Pearson Correlation, Spearman Correlation,* dan *Mutual Information* merepresentasikan

tiga perspektif yang berbeda dalam mengevaluasi relevansi fitur. *Pearson Correlation* merepresentasikan hubungan linear, *Spearman Correlation* merepresentasikan hubungan monotonik, sedangkan *Mutual Information* merepresentasikan hubungan non-linear berbasis teori informasi. Ketiga perspektif tersebut dipandang saling melengkapi dalam menghasilkan fitur yang lebih representatif. Untuk memanfaatkan keunggulan masing-masing metode, penelitian ini menerapkan pendekatan *ensemble feature selection* melalui mekanisme *fusion*. Pada tahap ini, setiap metode seleksi fitur dijalankan secara independen untuk menghasilkan daftar fitur terpilih berdasarkan kriteria masing-masing. Selanjutnya, hasil seleksi dari ketiga metode digabungkan dalam suatu proses *fusion* untuk membentuk himpunan fitur yang lebih komprehensif. Proses *fusion* dilakukan dengan menghilangkan fitur duplikat sehingga setiap fitur hanya muncul satu kali dalam himpunan akhir.

Pendekatan *fusion* dipilih karena mampu mengintegrasikan informasi yang diperoleh dari berbagai metode seleksi fitur tanpa kehilangan karakteristik unik yang dimiliki oleh masing-masing metode. Dengan demikian, fitur-fitur yang terpilih tidak hanya merepresentasikan hubungan linear, tetapi juga mampu menangkap pola monotonik dan non-linear yang terdapat dalam data lalu lintas jaringan. Berdasarkan proses tersebut, SPADEH dibangun sebagai suatu kerangka *ensemble feature selection* yang mengintegrasikan *Pearson Correlation, Spearman Correlation,* dan *Mutual Information* melalui mekanisme *fusion* untuk menghasilkan subset fitur yang lebih representatif. Subset fitur yang dihasilkan kemudian digunakan sebagai masukan bagi model *Deep Feedforward Neural Network* dalam proses deteksi intrusi pada jaringan kamera IP.

Secara konseptual, mekanisme kerja SPADEH dapat dijelaskan melalui empat tahapan utama, yaitu: (1) perhitungan relevansi fitur menggunakan *Pearson Correlation, Spearman Correlation,* dan *Mutual Information*; (2) pembentukan daftar fitur terpilih dari masing-masing metode; (3) proses *fusion* dan eliminasi fitur duplikat; serta (4) pembentukan subset fitur akhir yang digunakan pada tahap pelatihan dan pengujian model DFNN. Melalui pendekatan tersebut, SPADEH

diharapkan mampu mengatasi keterbatasan metode seleksi fitur tunggal dan menghasilkan fitur yang lebih optimal untuk mendukung proses deteksi intrusi.

2.2.11 *Intrusion Detection System* (IDS)

1. Definisi Intrusi dan Deteksi Intrusi

Istilah intrusi dalam keamanan siber adalah adopsi teknis dari kata bahasa Inggris, "Intrusion", yang secara harfiah berarti "penerobosan" atau "penyusupan". Lembaga standarisasi Amerika Serikat, *National Institute of Standards and Technology* (NIST) menjelaskan definisi intrusi pada dokumen CNSSI 4009-2015 yang mengutip dari IETF RFC 4949 Ver 2. Intrusi adalah peristiwa keamanan, atau kombinasi dari beberapa peristiwa keamanan, yang merupakan insiden keamanan di mana penyusup mendapatkan, atau mencoba mendapatkan, akses ke sistem atau sumber daya sistem tanpa memiliki otorisasi untuk melakukannya (Committee on National Security Systems, 2022).

Deteksi intrusi merupakan mekanisme penting untuk mendeteksi tindakan jahat yang mengeksploitasi kerentanan sistem. Intrusi mungkin saja berupa skrip pemrograman berbahaya dikendalikan oleh seorang peretas yang ingin mengambil keuntungan dari korban. Menurut (Scarfone & Mell, 2007) NIST, deteksi Intrusi adalah proses pemantauan peristiwa yang terjadi dalam sistem komputer atau jaringan dan menganalisisnya untuk pertanda kemungkinan insiden, yang merupakan pelanggaran atau ancaman pelanggaran kebijakan keamanan komputer, kebijakan penggunaan yang dapat diterima, atau praktik keamanan standar. Secara ringkas menurut (Ning & Jajodia, 2004) pada buku Handbook of Information Security (Vol. 3), intrusi adalah aktivitas yang melanggar kebijakan keamanan sistem dan deteksi intrusi adalah proses untuk mengidentifikasi intrusi tersebut.

2. Sejarah dan Definisi Sistem Deteksi Intrusi

IDS diperkenalkan pertama kali pada tahun 1980 dengan munculnya laporan dari James P. Anderson atas permintaan U.S. Air Force dengan judul "Computer Security Threat Monitoring and Surveillance". Laporan ini dianggap sebagai dokumen fondasi IDS. Anderson memperkenalkan ide untuk menganalisis

audit trail (catatan aktivitas sistem) untuk menemukan pelanggaran keamanan, penyalahgunaan oleh pengguna, dan aktivitas mencurigakan lainnya. Analisis ini membedakan antara "ancaman eksternal" (orang luar) dan "ancaman internal" (pengguna sah yang menyalahgunakan wewenang) (James & Co, 1980). Tahun 1985 Dorothy E. Denning, bekerja sama dengan Peter Neumann, mengembangkan model teoretis untuk sistem deteksi intrusi yang disebut *Intrusion Detection Expert System* (IDES). Model ini memperkenalkan dua pendekatan utama yang masih menjadi pilar hingga kini yaitu *Signature-based Detection* dan *Anomaly-based Detection* (Denning & Neuman, 1985).

Istilah *Intrusion Detection System* (IDS) menurut (Scarfone & Mell, 2007) adalah perangkat lunak atau perangkat keras yang memonitor sistem komputer dan aktivitas jaringan, kemudian memproduksi laporan tentang insiden yang terdeteksi. IDS bisa memeriksa aktivitas jaringan antara satu perangkat dengan perangkat lain yang terhubung dan memberitahukan peringatan jika ada aktivitas yang mencurigakan pada jaringan (D. Li et al., 2019). Pada jaringan komputer, masukan dari IDS yang akan diperiksa diambil dari telemetri jaringan (statistik lalu lintas, informasi yang diekstrak dari paket header dan konten) dan informasi dari host (perilaku proses, penelusuran *system call*, log aplikasi, perubahan berkas sistem). Keluaran dari IDS bisa berupa sebuah label atau skor sebuah serangan untuk masing-masing masukan atau urutan masukan di mana label bisa berupa biner, normal dan serangan maupun nilai banyak yang mengindikasikan tipe dari serangan (Gamage & Samarabandu, 2020).

3. Metode Deteksi pada IDS

Metode yang digunakan dalam mendeteksi intrusi pada IDS sangat beragam, misalnya metode berbasis aturan. Contohnya dalam mendeteksi DDoS pada jaringan sensor nirkabel, terdapat tiga fase yang dilakukan. Pada tahap pertama, pemantauan informasi untuk menyaring data penting dari jaringan utama, tahap ini dilakukan dengan bantuan *node* monitor. Tahap kedua dikenal sebagai tahap aplikasi aturan, di mana aturan yang telah ditentukan diterapkan pada informasi yang telah disaring pada tahap pertama oleh *node* monitor. Pada fase

terakhir, alarm deteksi dimunculkan jika ada paket informasi yang gagal dalam pengujian aplikasi aturan (Gopi et al., 2022).

4. Klasifikasi IDS

Berdasarkan sumber informasi dan aktivitas yang diperiksa, IDS digolongkan ke dalam 2 macam. Pertama *Host Intrusion Detection System* (HIDS) dan *Network Intrusion Detection System* (NIDS) (S. K. Gupta et al., 2022). HIDS bisa memantau aktivitas internal sistem seperti penggunaan memori, aktivitas modifikasi berkas, penggunaan *cpu*, dan aktivitas lainnya secara spesifik. Sedangkan NIDS fokus pada pemantauan paket yang melewati perangkat jaringan seperti *switch* dan *router*. HIDS adalah sistem yang berorientasi pada mesin dan ditempatkan secara terdistribusi. HIDS biasanya mengonsumsi banyak sumber daya sistem, meningkatkan kompleksitas ruang dan waktu, dan relatif sulit diterapkan untuk *interoperabilitas* antar platform yang berbeda. NIDS memindai paket secara independen pada jaringan. Namun kelemahan NIDS adalah besarnya sumber daya yang digunakan saat memindai konten lalu lintas data berkecepatan tinggi. Selain itu biasanya NIDS tidak bisa memindai paket yang dienkripsi (Martins et al., 2022).

5. Strategi Penempatan IDS (Arsitektur)

IDS adalah salah satu solusi dalam memberikan perlindungan terhadap jaringan IoT beserta perangkat di dalamnya. Penempatan IDS dapat dibagi dalam 3 cara yaitu terdistribusi, tersentralisasi dan hibrid. Masing-masing penempatan IDS memiliki kelebihan dan kekurangan. IDS terdistribusi dilakukan dengan menempatkan agen IDS pada masing-masing perangkat IoT. Oleh karena itu perangkat IoT selain bekerja sebagai sensor, juga bekerja mendeteksi intrusi (Kozik, 2018) (Patil et al., 2022). Apabila terdeteksi adanya intrusi, maka *node* akan memberikan peringatan dan pesan. Penempatan IDS tersentralisasi dilakukan dengan menempatkan agen IDS pada entitas jaringan semisal *router*. Setiap komunikasi antar *node* yang dikirimkan ke jaringan akan di analisa. Komunikasi dari dan ke dalam jaringan juga akan dicatat dan di analisa oleh agen IDS. Strategi

penempatan IDS Hibrid dilakukan dengan menggabungkan penempatan terdistribusi dan tersentralisasi. Pada penempatan Hibrid, jaringan disusun membentuk klaster dan sebuah *node* IDS sentral ditugaskan pada setiap klaster. *Node* sentral ini bertugas untuk mengatur dan memonitor aktivitas dari *node* lainnya pada jaringan klaster tersebut.

2.2.12 Kelemahan keamanan pada IoT

Frasa *Internet of things* (IoT) digunakan pertama kali oleh Kevin Ashton (Shahin et al., 2024) seorang ahli teknologi berkebangsaan Inggris pada tahun 1999 saat ia bekerja di MIT Media Center. IoT didefinisikan sebagai jaringan perangkat yang saling terhubung, termasuk sensor dan objek fisik lainnya, yang berkomunikasi serta berbagi data melalui internet (Atzori et al., 2010). Konsep inti dari IoT adalah untuk menghubungkan objek yang tersebar di sekitar seperti RFID, perangkat bergerak, sensor dan aktuator ke Internet melalui kabel atau nirkabel (Nauman et al., 2020). Keterbatasan sumber daya *node* sensor di *Internet of Things*, kompleksitas jaringan, dan karakteristik komunikasi nirkabel yang bersifat terbuka membuatnya rentan terhadap serangan (D. Li et al., 2019).

Sistem IoT bisa dibagi dalam tiga lapisan : *perception layer*, *network layer* dan *application layer* (D. Li et al., 2019). *Perception Layer* adalah lapisan inti dari sistem IoT itu sendiri. Lapisan ini terdiri dari berbagai perangkat terminal dan simpul komunikasi. Perangkat terminal ini bisa jadi adalah kartu RFID , pembaca kartu, kamera dan perangkat GPS. *Network Layer* menghubungkan lapisan sensor dengan lapisan aplikasi dan *Application Layer* adalah lapisan pemrosesan informasi dan interaksi komputer ke manusia. Sebagian ahli membagi IoT menjadi empat lapisan dasar yaitu lapisan persepsi, lapisan jaringan, lapisan *middleware,* dan lapisan aplikasi. Lapisan Persepsi adalah lapisan yang bertanggung jawab untuk mengelola perangkat pintar di seluruh sistem. Lapisan jaringan adalah lapisan yang meneruskan data antara *node cloud* dan perangkat IoT. Lapisan *middleware* adalah untuk memberikan abstraksi antara detail IoT dan aplikasi pengguna. Terakhir, lapisan aplikasi bertanggung jawab atas analitik, kontrol perangkat, dan pelaporan kepada pengguna (Gaber et al., 2022).

Jaringan dan aplikasi IoT rentan terhadap serangan (Rashid et al., 2020). Hal ini dikarenakan beberapa hal. Pertama, sebagian perangkat IoT memiliki sumber daya yang terbatas misalnya daya pemrosesan dan memori kecil yang mengakibatkan kemampuan pemrosesan yang terbatas. Kedua, perangkat IoT yang terhubung ke berbagai protokol yang berbeda dan peningkatan jumlah perangkat IoT mengakibatkan adanya latensi. Ketiga, perangkat IoT terkadang tidak dijaga sehingga memungkinkan penyusup dapat mengakses secara fisik. Keempat, sebagian besar komunikasi data yang digunakan pada perangkat IoT adalah komunikasi nirkabel sehingga memudahkan kemungkinan penyadapan.

Beberapa serangan terhadap perangkat IoT menurut (Madan et al., 2022) di antaranya disebabkan oleh :

1. Autentikasi yang lemah.

   Perusahaan pembuat perangkat IoT biasanya membuat instalasi perangkat dengan prosedur yang sangat simpel untuk memudahkan pengguna produk. *Username* dan sandi yang dipasang secara *default* pada instalasi juga dibuat sederhana. Ketika pengguna tidak mengubah setingan *default* perangkat tersebut, maka perangkat menjadi sangat rentan terhadap ancaman keamanan.
2. Konektivitas

   Sebagian besar perangkat IoT terhubung sepanjang waktu ke jaringan Internet. Hal ini tentunya memungkinkan penyerang untuk bisa mencoba mengakses kerentanan dari perangkat IoT sepanjang waktu. Perangkat IoT yang memiliki *port* terbuka dan kelemahan lainnya akan menjadi rentan.
3. Komunikasi yang tidak aman

   Algoritma kriptografi yang dipasang pada perangkat IoT kebanyakan tidak memiliki keamanan yang kuat. Hal ini dikarenakan algoritma kriptografi yang kuat membutuhkan komputasi yang besar sementara perangkat IoT memiliki sumber daya yang terbatas. Akibatnya komunikasi antar perangkat IoT akan menjadi sangat rentan untuk di eksploitasi.
4. *Port* tidak biasa yang terbuka

   Port terbuka menjadi rentan untuk disusupi oleh penyerang. *Port* yang tidak biasa digunakan terkadang hanya dibutuhkan sewaktu-waktu oleh sistem. *Port*

ini misalnya hanya digunakan untuk *update firmware*. *Port* seperti ini sering kali terdapat pada perangkat tertanam.

5. Kelemahan perangkat keras

   Perangkat keras IoT biasanya adalah perangkat tertanam yang di *update* secara berkala. Penyerang bisa memanfaatkan kelemahan perangkat keras untuk mengambil alih sistem.

Setiap intrusi pada target yang diserang memberikan dampak yang berbeda terhadap layanan dan proses yang terbentuk pada perangkat IoT (Thakkar & Lohiya, 2021a). Berdasarkan target yang diserang pada perangkat dan jaringan IoT maka intrusi pada jaringan IoT dapat dikelompokkan dalam empat kategori. Kategori intrusi pada jaringan IoT itu adalah intrusi fisik, intrusi jaringan, intrusi perangkat lunak, dan intrusi enkripsi. Berikut adalah dijelaskan intrusi yang terjadi pada jaringan IoT:

1. Intrusi Fisik

   Intrusi fisik berkaitan dengan perangkat keras yang ada di jaringan IoT. Beberapa contoh dari intrusi fisik di antaranya adalah *compromised nodes*, di mana penyerang dapat merusak *node* IoT dengan mengubah konfigurasinya atau menggantinya dengan *node* jahat, mencuri informasi, dan mengubah data *routing* (Hameed et al., 2022). Selain itu, ada juga *disrupting RFID signals*, yaitu ketika penyerang mengacaukan sistem RFID dengan merusak pembaca RFID, mengirimkan sinyal kebisingan, dan menyebabkan gangguan dalam komunikasi perangkat di jaringan IoT nirkabel. Contoh lainnya termasuk *node jamming, malicious node injection, hardware component tampering, social engineering, sleep deprivation attack,* dan *malicious code injection.*

2. Intrusi Jaringan

Intrusi jaringan dilakukan dengan mencoba merusak jaringan IoT untuk mengganggu operasi jaringan atau mencuri data jaringan. Contoh dari Intrusi terhadap jaringan adalah *Packet Sniffing Attack* di mana peretas menggunakan alat *sniffing* atau penyadap paket untuk mencuri paket jaringan yang mengalir dalam jaringan (Sikos, 2020). Alat ini mengambil *header* dan *payload* paket yang kemudian dianalisis oleh peretas untuk melakukan intrusi jaringan. Contoh lainnya

adalah *Sinkhole Attack*, Serangan ini dilakukan dengan cara merusak satu *node* di jaringan dan menjalankan serangan melalui *node* yang terinfeksi tersebut. *Node* korban kemudian mengirimkan informasi *routing* palsu ke tetangganya, menunjukkan jalur palsu antara sumber dan tujuan. Hal ini menarik lalu lintas jaringan ke *node* yang terinfeksi dan kemudian mengubah atau membuang paket. Serangan *sinkhole* dapat terdeteksi dengan membandingkan rata-rata jumlah hop pada setiap *node* dengan jumlah hop rata-rata. Jika *node* tersebut memiliki jumlah hop minimum yang lebih rendah dari nilai ambang, maka *node* tersebut dianggap rentan terhadap serangan *sinkhole*. Selain yang telah disebutkan, intrusi terhadap jaringan juga di antaranya adalah *RFID Signal Attack, Man-in-the-Middle Attack, Denial of Service, Compromising Routing Table,* dan *Sybil Attack.*

3. Intrusi Perangkat Lunak

Intrusi perangkat lunak dilakukan dengan merusak data yang ada dalam sumber daya sistem dan bahkan memengaruhi pemrosesan yang berjalan di sistem. Contoh dari Intrusi terhadap perangkat lunak adalah *Phishing,* serangan ini dilakukan dengan menggunakan teknik email *spoofing* untuk menyebar program berbahaya melalui lampiran email. Tujuan dari serangan *phishing* adalah untuk mencuri informasi rahasia seperti kredensial pengguna dan rincian bank (Safi & Singh, 2023). Contoh lainnya *Malicious Software, Malicious Shell Scripts*

4. Intrusi Enkripsi

Intrusi enkripsi berkaitan menyerang kelemahan proses enkripsi dengan merusak atau mencuri kunci enkripsi publik dan pribadi. Contoh dari Intrusi Enkripsi adalah *Side-Channel Attack.* Serangan *Side-Channel* memanfaatkan data sampingan dari perangkat terenkripsi, seperti konsumsi daya, waktu komputasi, dan frekuensi kesalahan, untuk mencuri kunci enkripsi (Nath N & V Nath, 2022). Beberapa jenis serangan sampingan yang digunakan adalah *timing attack, differential power attack,* dan *fault analysis attack*. *Timing attack*, misalnya, menggunakan waktu komputasi untuk mencuri kunci rahasia dan merusak sistem kriptografi. Kedua adalah *Cryptanalysis Attack*. Serangan *Cryptanalysis* dilakukan dengan memanfaatkan teks terenkripsi atau terbuka untuk mendapatkan kunci kriptografi.

Cyber security menjadi isu yang penting bagi sebuah perusahaan atau organisasi. Sebuah perusahaan atau organisasi akan mendapatkan kepercayaan tinggi dan kesuksesan saat dinilai mampu menjaga data pribadi pelanggan dengan baik (Y. Li & Liu, 2021). *Cyber security* mencakup tindakan praktis untuk melindungi jaringan dan data dari ancaman yang bersifat internal maupun eksternal. *Cyber Security* juga memastikan bahwa hanya orang-orang yang berhak saja yang bisa mengakses informasi. Menurut sebuah studi oleh Altman Vilandrie & Company, pada bulan April 2017, hampir setengah dari perusahaan Amerika Serikat (AS) yang menggunakan IoT telah mengalami pelanggaran keamanan siber, yang dapat menelan biaya hingga 13% dari pendapatan tahunan perusahaan kecil (Omolara et al., 2022).

Serangan yang terjadi pada jaringan IoT di antaranya adalah DDoS, *Flooding* dan *Malware*. Serangan terhadap jaringan IoT tersebut dapat melumpuhkan layanan. Berikut penjelasan berkaitan dengan serangan terhadap jaringan IoT :

1. Serangan *Distributed Denial-of-Service* (DDoS)

   Salah satu tantangan keamanan yang paling besar dalam jaringan IoT adalah DDoS, di mana banyak perangkat IoT yang terhubung ke Internet dapat disusupi oleh pihak yang tidak bertanggung jawab dan digunakan untuk menyerang satu target atau jaringan dengan lalu lintas yang sangat tinggi, membuatnya tidak dapat diakses oleh pengguna yang sah (Lawall et al., 2021).
2. Serangan *Flooding*

   Serangan *Flooding* terbagi menjadi beberapa jenis, yaitu *UDP Flood, Syn Flood, ICMP Flood,* dan *HTTP Flood. UDP Flood* dan *ICMP Flood* menyebabkan sumber daya target tidak dapat diakses, sementara *Syn Flood* memanfaatkan *spoofing* alamat IP dan mengarah ke serangan *Denial-of-Service*. *Ping Flood* mengakibatkan serangan *flooding* dengan banyak paket balasan yang dikirim karena adanya paket permintaan. Sementara itu, *HTTP Flood* dilakukan dengan membuat permintaan yang membebani sumber daya *server* dan menyebabkan layanan ditolak untuk pengguna yang sah.

3. Serangan *Malware*

   Malware atau *Malicious Sofware* adalah kode berbahaya yang dikembangkan secara khusus untuk merusak komputer dengan maksud mencuri informasi atau menghancurkan infrastruktur komputer. Sistem Operasi Windows memiliki berkas eksekusi yang nantinya dijalankan sebagai aplikasi berupa berkas PE dengan ekstensi .exe, .obj, .dll, dan lain-lain, maka pada lingkungan Linux juga memiliki berkas ELF seperti .elf, .o, .so dan lain-lain. Kebanyakan perangkat IoT dijalankan pada sistem operasi Linux yang berbasis debian. Perangkat-perangkat IoT tersebut memiliki CPU dengan arsitektur yang beragam (ARM, MIPS, x86 dan lain-lain). *Malware* kemudian dibuat oleh penyerang dengan menggunakan kode yang disesuaikan dengan lingkungan sistem operasi perangkat IoT tersebut. Berdasarkan laporan *Sonic Wall's 2019 Cyber Threat Report* (SonicWall, 2019), terjadi kenaikan serangan *malware* sekitar 22% dari tahun sebelumnya. Sekitar 217,5% kenaikan serangan *malware* terjadi pada perangkat IoT. SonicWall mencatat 32.7 juta serangan terhadap perangkat IoT terjadi pada tahun 2018. Terjadi kenaikan 217,5% dari tahun 2017 yang hanya berjumlah 10.3 juta serangan.

2.2.13 Ancaman Keamanan terhadap Kamera IP dalam Ekosistem IoT

Kamera pengawas adalah salah satu perangkat yang mulai banyak dimanfaatkan dalam menjadikan rasa aman bagi masyarakat perkotaan hari ini. Dengan adanya kamera pengawas, maka orang tidak lagi ragu untuk meninggalkan rumahnya saat akan pergi dalam waktu tertentu. Perangkat tersebut memungkinkan pengawasan terhadap rumah dan lingkungannya bisa dilakukan dari jarak jauh (Jan et al., 2021). Kamera yang digunakan saat ini ada dua jenis. Pertama adalah *CCTV (Close Circuit Television)* dan kedua adalah kamera IP. Perbedaan utama terletak dari media yang digunakan dan teknologi yang diterapkan, Kamera CCTV menggunakan sinyal analog dan ditransmisikan dengan menggunakan kabel *coaxial* ke perangkat yang dinamakan dengan DVR *(Digital Video Recorder)*. Sementara itu, kamera IP menggunakan kabel LAN yang terhubung ke perangkat NVR *(Network Video Recorder)*. Di samping itu biasanya kamera IP juga bisa diakses langsung dari jaringan tanpa harus terkoneksi ke perangkat *recorder* NVR.

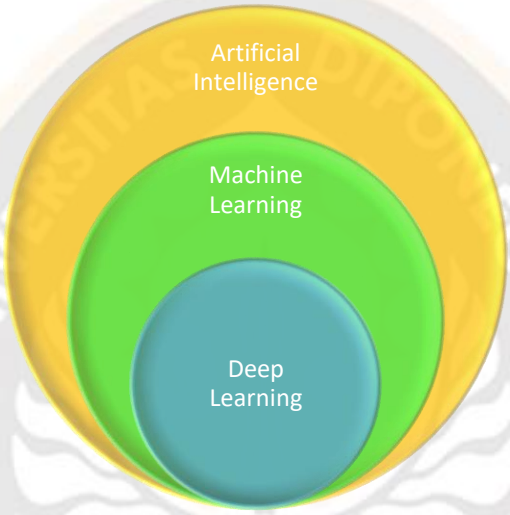
Internet yang digunakan untuk mentransmisikan video tangkapan dari kamera IP membuka celah untuk kemudian diakses oleh siapa pun. Ketika pengelolaan keamanan dari perangkat kamera IP kurang baik, maka kamera IP menjadi rentan terhadap serangan dari luar. Serangan terhadap kamera IP di antaranya adalah *Malware, Ransomware, DoS, Brute Force,* dan *Man-in-the-middle-attack* (Pao, 2021). Seorang peneliti yang menggunakan *moniker online* "*Watchful IP*" menemukan lebih dari 70 kamera Hikvision dan model NVR memiliki kerentanan. Kerentanan ini tergolong kritis karena memungkinkan peretas untuk mengendalikan perangkat dari jarak jauh tanpa interaksi pengguna apa pun. Cacat yang ditemukan dikenali sebagai CVE-2021-36260 (Kovacs, n.d.). Fakta yang dikemukakan oleh Fortinet bahwa 50% dari target eksploitasi utama selama tahun 2018 adalah perangkat IoT. Sebagian besar perangkat IoT yang di eksploitasi tersebut adalah kamera IP (FortiGuard SE Team, 2019). *Persirai Thingbot* adalah sebuah varian dari Mirai menargetkan secara khusus pada perangkat kamera IP. Sebanyak 1250 model kamera IP menjadi sasaran yang diserang. Jumlah *host* yang terinfeksi pada tahun 2017 adalah sebanyak 600.000 IP (Radanliev et al., 2018).

Kerentanan IP Kamera disebabkan berbagai faktor. Di antara faktor penyebab itu adalah terhubung secara langsung ke internet, kata sandi bawaan lemah, jarang diperbarui, keamanan perangkat yang terbatas, selalu aktif 24 jam, jumlah yang banyak tersebar di internet, dan karakter kamera IP yang jika disusupi terlihat tetap normal.

2.2.14 *Artificial Intelligence (AI)*

Artificial Intelligence (AI) adalah teknologi sains yang mempelajari dan mengembangkan teori, metode, teknik dan aplikasi yang menyimulasikan, mengembangkan dan memperluas kecerdasan manusia *(Machine Learning and Deep Learning Methods for Cybersecurity)*. AI dapat menyimulasikan proses informasi kesadaran manusia yaitu berpikir. AI bukanlah kecerdasan manusia, tetapi berpikir seperti manusia mungkin juga melebihi kecerdasan manusia. *Machine Learning (ML*) bagian dari *artificial intelligence (AI)*. ML adalah kemampuan komputer untuk belajar dan menemukan solusi tanpa harus diprogram secara spesifik (Alqudah & Yaseen, 2020). Dengan menggunakan *classifier* dan

algoritma, mesin dapat mengatasi *dataset* yang besar dan membangun model perilaku untuk membuat prediksi di masa depan. Keuntungan dari ML adalah kemampuannya untuk mendeteksi dan menganalisis serangan jaringan tanpa harus secara akurat menggambarkan serangan tersebut sebelumnya. ML dapat membantu dalam tugas-tugas umum seperti regresi, prediksi, dan klasifikasi di era jumlah data yang sangat besar dan kekurangan tenaga ahli di bidang keamanan siber.



Gambar 2.2 Hubungan antara AI, ML dan DL

Dalam perkembangan teknologi saat ini, istilah *Artificial Intelligence* (AI), *Machine Learning* (ML), dan *Deep Learning* (DL) sering digunakan secara bergantian. Namun, ketiganya memiliki makna dan lingkup yang berbeda. Untuk memahami peran dan keterkaitan masing-masing dalam dunia komputasi cerdas, penting untuk mengetahui perbedaan mendasar di antara ketiganya. Secara visualbisa digambarkan hubungan antara AI, ML dan DL sebagaimana

Gambar 2.2. AI (*Artificial Intelligence*) adalah konsep umum yang mencakup mesin yang mampu meniru kecerdasan manusia. Secara sederhana, AI adalah bidang ilmu komputer yang fokus pada pembuatan mesin yang bisa "berpikir" dan "bertindak" seperti manusia. Tujuan pembuatan AI adalah membuat program yang bisa belajar, beralasan, memecahkan masalah, memahami bahasa, dan bahkan bisa berkreasi secara mandiri. Machine Learning (ML) adalah subkategori dari AI yang mengkhususkan diri dalam membuat model statistik dan

algoritma untuk memungkinkan komputer belajar dari data, tanpa perlu diprogram secara eksplisit. Ada tiga jenis ML yang utama, *supervised learning* (pembelajaran terawasi), *unsupervised learning* (pembelajaran tak terawasi), dan *reinforcement learning* (pembelajaran penguatan). Deep Learning (DL) adalah subkategori dari ML yang menggunakan struktur jaringan saraf tiruan dengan banyak lapisan (*Deep Neural Networks*) untuk menangani data yang sangat besar dan kompleks, seperti gambar atau suara. Istilah *deep* mengacu pada banyaknya lapisan tersembunyi yang dimiliki teknik ini untuk belajar dari data dalam jumlah besar.

2.2.15 Teknik Prapengolahan Data

Prapengolahan data bertujuan untuk meningkatkan kualitas data agar lebih mudah dipahami dan diolah dengan teknik yang tepat. Tahap ini mendukung efisiensi serta menghasilkan proses pengolahan data yang lebih baik. Prapengolahan dapat dilakukan melalui pembersihan, integrasi, reduksi, penambahan, atau transformasi data, baik secara terpisah maupun dikombinasikan. Teknik prapengolahan data yang banyak digunakan pada deteksi intrusi di antaranya adalah : Data *cleaning*, *imbalanced learning*, *data convert*, seleksi fitur, ekstraksi fitur dan visualisasi (Yang et al., 2022).

1. **Pembersihan Data**

Pembersihan data dilakukan terhadap nilai yang hilang atau kosong (*null*). Nilai hilang atau kosong biasanya disebabkan karena kesalahan sistem (*error*), gangguan sinyal, ataupun karena data yang bersifat pribadi sehingga tidak didefinisikan. Data-data seperti ini harus ditangani agar tidak merusak pemodelan *dataset*. Penanganan data hilang dan kosong di antaranya adalah dengan menghapus nilai kosong atau menggantinya dengan nilai lain seperti nilai median, rata-rata (*mean*) atau modus.

2. **Normalisasi dan Standarisasi Data**

Normalisasi dan standarisasi data dibuat untuk menyamakan rentang data yang akan digunakan sebagai dataset. Jika dataset tidak dibuat dalam rentang yang sama, maka akan sulit dilakukan pemodelan terhadap dataset. Hal ini dikarenakan data memiliki nilai dan satuan yang berbeda-beda. Misalnya ukuran paket dihitung

dengan satuan *bytes* sementara waktu paket dihitung menggunakan detik. Sehingga terkadang didapatkan di dalam dataset ukuran paket data adalah 1024 bytes dan lama pengiriman 5 detik.

Normalisasi dilakukan untuk membuat data berada dalam rentang yang konsisten (J. Liu et al., 2021). Normalisasi umumnya dibuat dalam rentang 0 dan 1 (Popoola et al., 2021). Ada beberapa teknik normalisasi data yang umum digunakan, di antaranya adalah normalisasi Min-Max. Teknik normalisasi Min-Max dihitung menggunakan persamaan :

$$X_{baru} = \frac{X_{lama} - X_{min}}{X_{max} - X_{min}} \tag{2.9}$$

dimana X adalah fitur yang akan dihitung, X_{baru} adalah ukuran fitur baru yang akan dihasilkan, X_{lama} adalah ukuran fitur lama, X_{min} nilai minimal fitur dan X_{max} adalah nilai maksimal fitur.

Standarisasi mengubah data agar memiliki mean 0 dan standar deviasi 1, yang sangat berguna ketika data memiliki distribusi yang berbeda-beda atau tidak terdistribusi normal. Persamaan untuk menghitung standarisasi z-score adalah :

$$z_i = \frac{x_i - \mu}{\sigma} \tag{2.10}$$

dimana z_i adalah skor standar atau z-score dari suatu nilai data, x_i adalah nilai data mentah yang akan distandarisasi, μ adalah rata-rata populasi atau rata-rata sampel dari nilai data, dan σ adalah deviasi standar populasi atau deviasi standar sampel dari nilai data. Adapun standar deviasi dihitung menggunakan persamaan:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2} \tag{2.11}$$

dimana n adalah jumlah total data.

3. Teknik Penyeimbangan data

Ketidakseimbangan data terjadi karena jumlah sampel pada satu atau beberapa kelas jauh lebih besar dibandingkan kelas lainnya. Apabila data tidak

seimbang, maka akan terjadi beberapa masalah di antaranya adalah model yang mengalami bias terhadap kelas mayoritas (A. Binsaeed & Hafez, 2023). Model akan cenderung memprediksi kelas mayoritas karena lebih sering dilatih pada kelas mayoritas tersebut. Masalah lainnya adalah kinerja yang tampak bagus tapi menyesatkan. Hal ini dikarenakan prediksi terlihat akurat disebabkan selalu memprediksi kelas mayoritas padahal sebenarnya gagal untuk memprediksi kelas minoritas (Pramanick et al., 2025).

Beberapa teknik penyeimbangan data yang digunakan diantaranya adalah under-sampling, over-sampling, cost-sensitive learning, dan generative model. salah satu contoh dari teknik over-sampling adalah *Synthetic Minority Over-sampling Technique* (SMOTE). SMOTE penting untuk mengatasi permasalahan ketidakseimbangan kelas dalam dataset, terutama ketika kelas minoritas memiliki jumlah data yang jauh lebih sedikit dibandingkan kelas mayoritas (Osa et al., 2024). Dengan menghasilkan sampel sintetik melalui interpolasi antar data minoritas yang ada, SMOTE membantu model pembelajaran mesin untuk lebih memahami karakteristik kelas minoritas, sehingga dapat meningkatkan kinerja prediksi, memperbaiki metrik evaluasi seperti presisi, *recall*, dan *F1-score*, sekaligus mengurangi bias terhadap kelas mayoritas. Selain itu, teknik ini juga berperan dalam mengurangi risiko *overfitting* akibat keterbatasan data minoritas, sehingga model yang dihasilkan lebih seimbang, adil, dan mampu melakukan generalisasi dengan lebih baik pada berbagai kasus, termasuk deteksi penipuan, diagnosis penyakit, maupun identifikasi anomali lainnya.

4. Seleksi Fitur pada Data dan Ekstraksi Fitur

Fitur adalah informasi tentang data yang digunakan. Seleksi fitur adalah proses dalam memilih fitur yang berkontribusi dalam prediksi variabel dan mereduksi waktu latih serta menghindari kompleksitas komputasi (Kayode Saheed et al., 2022). Teknik yang digunakan pada pemilihan fitur adalah dengan mengurangi dimensi data. dilakukan dengan menghapus beberapa atribut yang dianggap tidak relevan. Pemilihan fitur membantu dalam memahami data, mengurangi kebutuhan komputasi, mengurangi dimensi vektor karakteristik dan meningkatkan performansi prediktor (Nguyen et al., 2022). Seleksi fitur dan

ekstraksi fitur adalah dua jenis teknik pengurangan dimensi. Menurut (Cavusoglu et al., 2005) terdapat 3 hal yang sangat fundamental dalam manajemen IT yaitu pencegahan, deteksi dan respons. Pemilihan Fitur dapat dilakukan dengan beberapa cara (Yang et al., 2022):

1. Seleksi Manual, cara ini dengan menghapus fitur-fitur yang tidak relevan dengan cara manual
2. Pencarian menyeluruh *(Exhaustive Search)*, cara ini dengan mencoba seluruh kemungkinan sehingga didapatkan fitur-fitur terbaik yang memiliki kesalahan sangat kecil.
3. Metode tertanam *(Embedded method)*,
4. Pendekatan Pembungkus *(Wrapper approach)*
5. Metode Saringan *(Filtering Method)*.

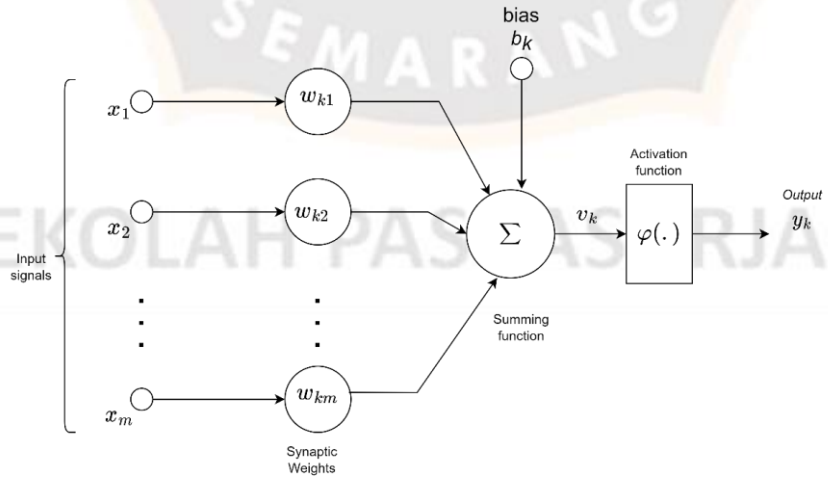
Ekstraksi Fitur berbeda dengan seleksi fitur. Ekstraksi fitur adalah pengurangan jumlah dimensi atau fitur dalam sebuah *dataset*. Tujuannya adalah untuk mengekstrak penyebaran informasi yang berharga dan relevan di antara fitur masukan dan memproyeksikannya ke dalam jumlah masukan yang berkurang sambil meminimalkan kehilangan informasi (Sarhan et al., 2024). Beberapa algoritma yang digunakan pada ekstraksi fitur adalah (Yang et al., 2022) *Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)* dan *AutoEncoder (AE)*. Dalam konteks deteksi intrusi pada perangkat IoT yang memiliki sumber daya terbatas, seleksi fitur lebih tepat untuk diterapkan dibandingkan ekstraksi fitur (J. Li, Othman, et al., 2024).

2.2.16 *Deep Feedforward Neural Network* pada Klasifikasi Data

Perkembangan *Deep Learning* (DL) dilatarbelakangi oleh beberapa keterbatasan yang ditemukan pada pendekatan *Machine Learning* konvensional, khususnya pada model *shallow learning*. Salah satu kelemahan utama dari pendekatan tersebut adalah ketergantungannya pada proses ekstraksi fitur yang dirancang secara manual. Proses pembangunan model ekstraksi fitur untuk memperoleh tingkat akurasi yang tinggi sering kali bersifat tidak praktis, kompleks, dan membutuhkan keahlian khusus pada domain tertentu (Suyanto, 2019). Selain itu, pada beberapa metode *shallow learning*, peningkatan jumlah data tidak selalu

diikuti oleh peningkatan akurasi secara signifikan, sehingga kemampuan model dalam memanfaatkan data berskala besar menjadi terbatas. *Deep Learning* hadir sebagai pendekatan yang mampu melakukan pembelajaran fitur secara otomatis melalui banyak lapisan representasi, sehingga lebih adaptif dalam mengenali pola kompleks pada data. Dalam konteks keamanan siber, teknologi DL dipandang sebagai salah satu komponen penting dalam membangun sistem pertahanan inti untuk melindungi ekosistem *Smart City* dari berbagai ancaman digital (Chen et al., 2021). Kemampuan DL dalam memodelkan pola data yang kompleks menjadikannya relevan untuk berbagai bidang kajian, termasuk deteksi intrusi, karena mampu menghasilkan performa yang kompetitif dan dalam beberapa kasus melampaui pendekatan konvensional.

Neural Network (NN) lebih dikenal dengan *Artificial Neural Network* (ANN) atau *Simulated Neural Network* (SNN). NN adalah sebuah model matematis atau sebuah model komputasi yang terinspirasi dari struktur dan fungsional dari jaringan syaraf biologi manusia. Sebuah NN terdiri dari sekelompok jaringan syaraf tiruan yang saling berhubungan dan memproses informasi menggunakan pendekatan koneksionis untuk komputasi. ANN terdiri dari lapisan *node* yang berisi satu atau lebih lapisan tersembunyi dan lapisan keluaran. Hubungan satu dengan yang lainnya diberikan nilai bobot dan ambang batas terkait. Apabila nilai keluaran individu berada di atas ambang batas yang telah ditentukan, maka *node* itu akan diaktifkan untuk mengirimkan data ke lapisan berikutnya.



Gambar 2.3 Sebuah sel saraf buatan

Ilustrasi sebuah sel saraf buatan (*perceptron*) dapat dilihat pada Gambar 2.3 (Haykin, 1999) (Suyanto, 2019) dengan masukan sinyal dinotasikan sebagai x_1, x_2, x_3, …,x_m. Bobot sinaptik dinotasikan sebagai w_{k1}, w_{k2}, …, w_{km}, di mana w_0 bernilai sama dengan bias b_k dan mendapat masukan x_0 yang bersifat tetap dengan nilai +1. Nilai bobot lainnya $(w_1, w_2, w_3, \ldots, w_n)$ mendapatkan masukan nilai acak. Fungsi yang diturunkan dari gambar tersebut adalah

$$v = \sum_{i=1}^{m} w_i x_i + b \tag{2.12}$$

di mana :

v : Keluaran dari neuron,

m : nilai maksimal dari data ke-i,

w_i : bobot neuron ke-i,

x_i : data masukan ke-i yang akan diproses oleh jaringan syaraf tiruan,

b : nilai bias biasanya berupa nilai tetap yang akan ditambahkan pada perkalian data masukan dan bobot.

Keluaran v kemudian dibatasi dengan menggunakan fungsi aktivasi $\varphi(.)$ sehingga didapatkan keluaran y_k. Keluaran y_k mendefinisikan kelas dari sebuah masukan sinyal.

$$y_k = \varphi(v_k) \tag{2.13}$$

di mana:

y_k : Keluaran neuron ke i setelah dikalikan dengan fungsi aktivasi,

φ : fungsi aktivasi.

Ada beberapa fungsi aktivasi yang digunakan pada *neural network*. Diantaranya adalah *Sigmoid*, *Hyperbolic Tangen* (tanh), *Rectified Linear Unit (ReLU)*, *Leaky Rectified Linear Unit* (Leaky ReLU), dan *Parametric Rectified Linear Unit* (PReLU).

a. Sigmoid

$$\varphi(x) = \frac{1}{1 + e^{-x}} \tag{2.14}$$

b. Tanh

$$\varphi(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.15}$$

c. ReLU

$$\varphi(x) = \max(0, x) \tag{2.16}$$

d. Leaky ReLU

$$\varphi(x) = \begin{cases} x & \text{jika } x > 0 \\ ax & \text{jika } x \leq 0 \end{cases} \tag{2.17}$$

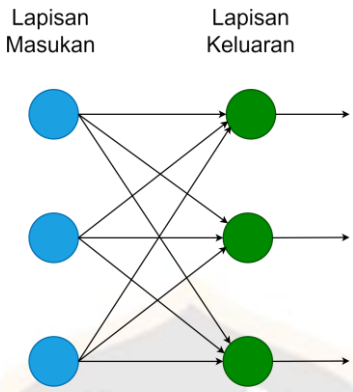
di mana a adalah nilai kecil positif yang nilainya tetap selama pelatihan.

e. PreLU

$$\varphi(x) = \begin{cases} x & \text{Jika } x > 0 \\ ax & \text{Jika } x \leq 0 \end{cases} \tag{2.18}$$

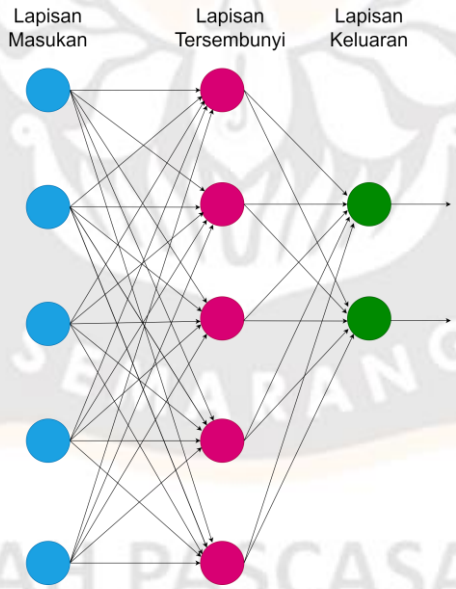
di mana a adalah parameter yang bisa berubah selama pelatihan.

Pengembangan dari *neural network* membentuk tiga arsitektur yaitu *single-layer feedforward network, multilayer feedforward network* dan *recurrent network. Single-layer feedforward network* adalah arsitektur yang paling sederhana yang hanya terdiri dari lapisan masukan dan lapisan keluaran. *Node* sumber terproyeksi langsung ke dalam lapisan keluaran sebagaimana terlihat pada Gambar 2.4. Ilustrasi pada gambar menampilkan tiga *node* pada lapisan masukan dan tiga *node* pada lapisan keluaran.



Gambar 2.4 Arsitektur *single-layer feedforward network*

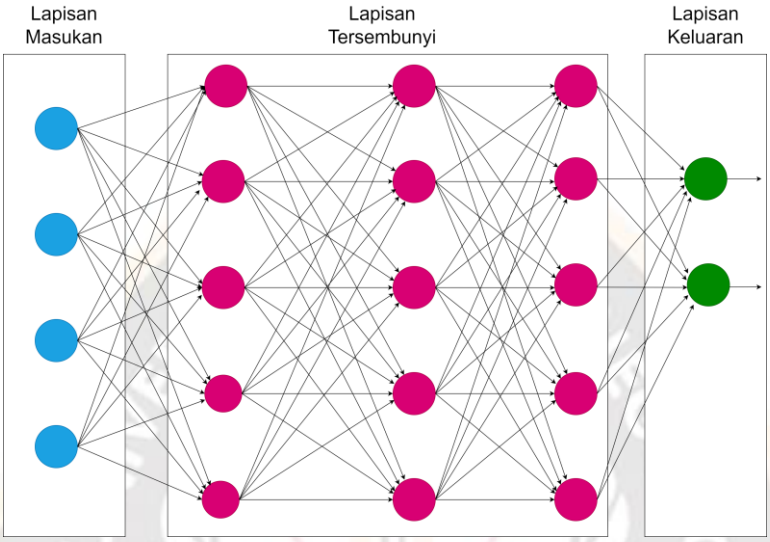
Arsitektur *multilayer feedforward network* memiliki satu atau lebih lapisan tersembunyi (*hidden layer*) dengan *node computations* yang berhubungan disebut *hidden neurons*. Gambar 2.5 memperlihatkan contoh ilustrasi *multilayer feedforward network* dengan satu lapisan masukan, satu *hidden layer* dan satu lapisan keluaran. Lapisan masukan terdiri dari lima masukan *node s, hidden layer* terdiri dari lima *neuron* dan pada lapisan keluaran terdapat dua neuron.



Gambar 2.5 Arsitektur *Multilayer Feedforward Neural Network*

DFNN dibentuk dari gabungan *feedforward neural network (FNN)* network tanpa koneksi umpan balik. Komponen utama dari DFNN adalah lapisan masukan, lapisan keluaran dan lapisan tersembunyi yang lebih dari satu lapisan (Cil et al.,

2021). Masing-masing lapisan memiliki beberapa *neuron* dan terhubung secara penuh ke *neuron* pada arah berikutnya (Narayana Rao et al., 2021). Gambar 2.6 menampilkan arsitektur Deep FNN dengan 3 lapisan tersembunyi.



Gambar 2.6 Arsitektur Deep Feedforward Network

Lapisan masukan berfungsi sebagai pintu masuk pertama data awal ke dalam jaringan. Lapisan tersembunyi adalah lapisan yang terdapat di antara lapisan masukan dan lapisan keluaran. Pada lapisan ini bisa terdapat banyak neuron yang menyebabkannya bisa disebut "dalam". Setiap neuron dalam lapisan tersembunyi ini akan mengambil hasil terbaik dari lapisan sebelumnya kemudian menerapkan fungsi aktivasi dan kemudian meneruskan hasilnya ke lapisan berikutnya. Proses ini dikenal dengan umpan maju atau *feedforward*. Sementara lapisan terakhir adalah lapisan keluaran yang merupakan hasil pengolahan dari lapisan tersembunyi. Hasil ini berupa hasil prediksi terhadap masukan yang dimasukkan pada jaringan.

Pemilihan arsitektur DFNN dalam penelitian ini didasarkan pada kesesuaiannya dengan karakteristik data yang digunakan. Dataset UNSW-NB15 maupun dataset *real-world* kamera IP direpresentasikan dalam bentuk fitur-fitur statistik hasil ekstraksi lalu lintas jaringan sehingga menghasilkan data tabular berdimensi tinggi. Berbeda dengan CNN yang dirancang untuk mempelajari pola spasial pada citra atau LSTM yang dirancang untuk mempelajari ketergantungan temporal pada data sekuensial, DFNN lebih sesuai untuk memodelkan hubungan

non-linear antar fitur dalam data tabular. Selain itu, DFNN memiliki kompleksitas yang relatif lebih rendah dibandingkan arsitektur *deep learning* yang lebih kompleks, sehingga memberikan keseimbangan yang baik antara akurasi, kemampuan generalisasi, dan efisiensi komputasi dalam sistem deteksi intrusi.

2.2.17 Dataset UNSW-NB15

UNSW-NB15 merupakan singkatan dari University of New South Wales Network Behavior-15, yaitu dataset lalu lintas jaringan yang dikembangkan oleh Universitas New South Wales (UNSW) untuk mendukung penelitian dalam bidang deteksi intrusi jaringan. Dataset ini dibuat sebagai respons terhadap berbagai keterbatasan pada dataset sebelumnya, seperti KDD Cup 1999 dan NSL-KDD, yang dianggap kurang merepresentasikan karakteristik serangan jaringan modern. UNSW-NB15 menyediakan data yang lebih realistis, beragam, dan relevan dengan kondisi lalu lintas jaringan kontemporer, sehingga banyak digunakan sebagai bahan evaluasi dalam pengembangan model keamanan siber, khususnya sistem deteksi intrusi berbasis ML dan DL. Dataset ini terdiri atas lebih dari dua juta instance dan mencakup 49 fitur yang merepresentasikan berbagai aspek lalu lintas jaringan, seperti informasi aliran paket, protokol, layanan, durasi koneksi, serta karakteristik statistik jaringan. Selain kelas Normal, UNSW-NB15 memuat sembilan kategori serangan, yaitu Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode, dan Worms. Keberagaman fitur dan jenis serangan tersebut menjadikan UNSW-NB15 sebagai salah satu dataset yang penting untuk menguji kemampuan model dalam mengenali pola serangan jaringan secara lebih akurat dan adaptif. (Moustafa & Slay, 2015)

Dataset UNSW-NB15 dibuat menggunakan *IXIA PerfectStorm tool* yang menghasilkan lalu lintas jaringan normal dan serangan terkini berdasarkan basis data CVE. Fitur-fitur yang dibuat pada dataset ini terdiri dari fitur aliran (*flow features*), fitur dasar (*basic features*), fitur isi (*content features*), fitur waktu (*time features*), fitur tambahan (*additional generated features*) dan fitur label (*labelled features*). Adapun tipe data dari masing-masing fitur ditandai dengan kode N (*Nominal*), I (*Integer*), F (*Float*), T (*Timestamp*) dan B (*Binary*).

Tabel 2.1 Fitur aliran pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Deskripsi Singkat
1	srcip	N	Alamat IP sumber
2	sport	I	*Port* sumber
3	dstip	N	Alamat IP tujuan
4	dsport	I	*Port* tujuan
5	proto	N	Protokol (TCP/UDP/ICMP)

Fitur aliran adalah kelompok fitur awal yang digunakan untuk mengidentifikasi karakteristik dasar suatu aliran koneksi jaringan (*network flow*). Fitur ini berfungsi sebagai penanda utama agar setiap transaksi jaringan dapat dikenali berdasarkan sumber, tujuan, *port*, dan protokol yang digunakan. Sebagaimana yang ditampilkan pada Tabel 2.1, fitur aliran menjawab pertanyaan "siapa berkomunikasi dengan siapa, melalui port apa, dan menggunakan protokol apa." Fitur ini penting karena menjadi dasar pencocokan antara hasil ekstraksi dari Argus dan Bro-IDS.

Tabel 2.2 Fitur dasar pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Deskripsi Singkat
1	state	N	Status koneksi (ACC, CLO, dll.)
2	dur	F	Durasi total koneksi
3	sbytes	I	Byte dari sumber ke tujuan
4	dbytes	I	Byte dari tujuan ke sumber
5	sttl	I	TTL dari sumber ke tujuan
6	dttl	I	TTL dari tujuan ke sumber
7	sloss	I	Paket hilang/retransmit dari sumber
8	dloss	I	Paket hilang/retransmit dari tujuan
9	service	N	Jenis layanan (http, ftp, dns, dll.)
10	sload	F	Bit rate dari sumber
11	dload	F	Bit rate dari tujuan
12	spkts	I	Jumlah paket sumber → tujuan
13	dpkts	I	Jumlah paket tujuan → sumber

Fitur dasar adalah kelompok fitur yang menggambarkan karakteristik umum dari suatu koneksi jaringan, terutama informasi terkait durasi, ukuran data, jumlah paket, arah lalu lintas, kondisi koneksi, dan jenis layanan yang digunakan (Tabel 2.2). Secara umum, fitur dasar berfungsi untuk menangkap pola dasar lalu lintas jaringan. Misalnya, serangan tertentu dapat terlihat dari durasi koneksi yang tidak wajar, jumlah paket yang sangat besar, *byte* yang tidak seimbang antara sumber dan

tujuan, atau penggunaan layanan tertentu terlihat mencurigakan. Karena itu, fitur ini menjadi bagian penting dalam proses klasifikasi trafik normal dan trafik serangan pada sistem deteksi intrusi jaringan

Tabel 2.3 Fitur isi pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Deskripsi Singkat
1	swin	I	Source TCP window size
2	dwin	I	Destination TCP window size
3	stcpb	I	Source TCP sequence number
4	dtcpb	I	Destination TCP sequence number
5	smeansz	I	Rata-rata ukuran paket dari sumber
6	dmeansz	I	Rata-rata ukuran paket dari tujuan

Tabel 2.3 Fitur isi pada dataset UNSW-NB15 (lanjutan)

No	Nama Fitur	Tipe	Deskripsi Singkat
7	trans_depth	I	Kedalaman transaksi HTTP
8	res_bdy_len	I	Panjang body respon HTTP

Fitur isi adalah kelompok fitur yang menggambarkan informasi isi atau karakteristik internal dari koneksi jaringan, terutama yang berkaitan dengan perilaku protokol TCP dan aktivitas layanan tertentu seperti HTTP. Fitur ini digunakan untuk melihat pola komunikasi yang lebih mendalam, bukan hanya alamat IP, port, atau durasi koneksi. Fitur ini penting karena beberapa serangan dapat terlihat dari pola isi koneksi, misalnya ukuran paket yang tidak wajar, transaksi HTTP yang mencurigakan, atau pola komunikasi TCP yang berbeda dari trafik normal. Tabel 2.3 menampilkan rincian masing-masing fitur dan definisinya.

Tabel 2.4 Fitur waktu pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Deskripsi Singkat
1	sjit	F	Jitter (ms) dari sumber
2	djit	F	Jitter (ms) dari tujuan
3	stime	T	Waktu mulai koneksi
4	ltime	T	Waktu selesai koneksi
5	sintpkt	F	Waktu antar paket sumber (ms)
6	dintpkt	F	Waktu antar paket tujuan (ms)
7	tcprtt	F	RTT TCP (synack + ackdat)
8	synack	F	Delay SYN → SYN-ACK
9	ackdat	F	Delay SYN-ACK → ACK

Fitur waktu sebagaimana ditampilkan pada Tabel 2.4 menggambarkan karakteristik waktu dari aliran lalu lintas jaringan, terutama terkait kapan koneksi dimulai, kapan berakhir, jeda antar paket, serta waktu respons pada koneksi TCP. fitur ini digunakan untuk melihat pola temporal komunikasi jaringan. Pada trafik normal, jeda antar paket dan waktu respons biasanya memiliki pola yang relatif wajar. Sebaliknya, pada serangan seperti DoS, *scanning*, atau *exploit*, pola waktunya dapat berubah, misalnya koneksi terjadi sangat cepat, jeda paket tidak stabil, atau waktu respons TCP menjadi tidak normal. Atas dasar tersebut, maka fitur waktu penting karena membantu model deteksi intrusi mengenali serangan tidak hanya dari isi atau jumlah paket, tetapi juga dari perilaku waktu lalu lintas jaringan.

Tabel 2.5 Fitur tambahan pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Kategori	Deskripsi Singkat
1	is_sm_ips_ports	B	General	1 jika src=dst IP & *port* sama
2	ct_state_ttl	I	General	Jumlah state berdasarkan TTL
3	ct_flw_http_mthd	I	General	Jumlah flow HTTP (GET/POST)
4	is_ftp_login	B	General	1 jika login FTP sukses
5	ct_ftp_cmd	I	General	Jumlah perintah FTP
6	ct_srv_src	I	Connection	Koneksi dengan layanan + sumber sama
7	ct_srv_dst	I	Connection	Koneksi dengan layanan + tujuan sama
8	ct_dst_ltm	I	Connection	Jumlah koneksi dengan dst sama
9	ct_src_ltm	I	Connection	Jumlah koneksi dengan src sama
10	ct_src_dport_ltm	I	Connection	Koneksi dengan src + dstport sama
11	ct_dst_sport_ltm	I	Connection	Koneksi dengan dst + sport sama
12	ct_dst_src_ltm	I	Connection	Koneksi dengan src + dst sama

Fitur tambahan (Tabel 2.5) adalah fitur yang tidak langsung diambil dari paket mentah, tetapi dibentuk melalui proses analisis lanjutan. Fitur-fitur ini dibangkitkan (*generated*) dari hasil pengolahan fitur sebelumnya (*flow, basic, content,* dan *time*). Tujuannya adalah menangkap pola koneksi yang lebih kompleks, terutama pola yang sering muncul pada aktivitas serangan seperti

scanning, eksploitasi, login FTP, atau koneksi berulang dalam rentang tertentu. Fitur tambahan terdiri dari 12 fitur yang dibagi menjadi dua kelompok:

1. Fitur tujuan umum

Fitur ini digunakan untuk menangkap informasi umum yang penting dari sisi keamanan jaringan, misalnya kesamaan IP dan port, kombinasi state dengan TTL, metode HTTP, serta aktivitas login dan perintah FTP. Contohnya: *is_sm_ips_ports, ct_state_ttl, ct_flw_http_mthd, is_ftp_login,* dan *ct_ftp_cmd*.

2. Fitur koneksi

Fitur ini digunakan untuk menghitung pola koneksi yang berulang dalam 100 koneksi terakhir berdasarkan waktu akhir koneksi. Fitur ini penting untuk mendeteksi perilaku serangan yang dilakukan secara bertahap atau tersembunyi, misalnya *scanning* pelan, koneksi berulang ke *host* tertentu, atau percobaan akses ke *port* tertentu. Contohnya: *ct_srv_src, ct_srv_dst, ct_dst_ltm, ct_src_ltm, ct_src_dport_ltm, ct_dst_sport_ltm,* dan *ct_dst_src_ltm.*

Tabel 2.6 Fitur label pada dataset UNSW-NB15

No	Nama Fitur	Tipe	Deskripsi Singkat
1	attack_cat	N	Kategori serangan (9 jenis)
2	Label	B	0 = normal, 1 = attack

Fitur label sebagaimana yang diperlihatkan pada Tabel 2.6 adalah kelompok fitur yang berfungsi sebagai penanda kelas data. Fitur ini tidak menjelaskan karakteristik teknis lalu lintas jaringan seperti durasi, *byte*, paket, atau waktu koneksi, tetapi digunakan untuk menunjukkan apakah suatu *record* termasuk lalu lintas normal atau serangan. Oleh karenanya, *attack_cat* digunakan untuk klasifikasi multi kelas, yaitu membedakan jenis serangan, sedangkan label digunakan untuk klasifikasi biner, yaitu membedakan data normal dan data serangan.

Tabel distribusi UNSW-NB15 (Tabel 2.7) menunjukkan proporsi *record* normal dan sembilan kategori serangan yang digunakan untuk mengevaluasi *Network Intrusion Detection System* (NIDS). Dari total sekitar 2,54 juta *record*,

mayoritas berasal dari trafik normal (lebih dari 2,2 juta *record*), sehingga dataset ini tetap realistis karena merepresentasikan kondisi jaringan sehari-hari. Dari distribusi ini, terlihat bahwa kategori *Generic* (215 ribu) dan Exploits (44 ribu) merupakan serangan dengan jumlah terbesar, sedangkan Worms (174) dan Shellcode (1,511) hanya memiliki sedikit data. Kondisi ini menimbulkan tantangan *class imbalance*, di mana algoritma pembelajaran mesin cenderung bias terhadap kelas dengan jumlah data besar.

Tabel 2.7 Distribusi kategori dataset UNSW-NB15

Tipe / Kategori	Baris	Deskripsi Singkat
Normal	2,218,761	Transaksi jaringan normal (trafik alami).
Fuzzers	24,246	Serangan dengan data acak untuk menyebabkan crash atau suspend program/jaringan.
Analysis	2,677	*Port* scan, spam, dan penetrasi berkas HTML.
Backdoors	2,329	Akses tersembunyi untuk melewati mekanisme keamanan.
DoS	16,353	*Denial of Service*, membuat server/jaringan tidak tersedia.
Exploits	44,525	Memanfaatkan kerentanan software/OS untuk menyerang.
Generic	215,481	Serangan *cipher* generik yang bekerja terhadap semua blok *cipher*.
Reconnaissance	13,987	Aktivitas pengintaian untuk mengumpulkan informasi target.
Shellcode	1,511	Kode berukuran kecil yang digunakan sebagai *payload* eksploitasi.
Worms	174	Malware yang mereplikasi diri dan menyebar ke komputer lain.

Dataset UNSW-NB15 memiliki versi yang telah dipilih untuk digunakan melatih dan menguji model deteksi intrusi (Das et al., 2022). Dataset terdiri dari dua berkas yang berbeda. Salah satu sebagai data latih dan lainnya sebagai data uji. Masing-masing dataset terdiri dari 1 kategori normal dan lainnya sebagai serangan. Sebaran kelompok dataset ini ditunjukkan pada Tabel 2.8.

Tabel 2.8 Kategori sampel dataset pelatihan dan pengujian UNSW-NB15

No	Kategori	Data Pelatihan	Data Pengujian
1	Normal	56000	37000
2	Generic	40000	18871
3	Exploits	33393	11132

Tabel 2.8 Kategori sampel dataset pelatihan dan pengujian UNSW-NB15 (lanjutan)

No	Kategori	Data Pelatihan	Data Pengujian
4	Fuzzers	18184	6062
5	DoS	12264	4089
6	Reconnaissance	10491	3496
7	Analysis	2000	677
8	Backdoor	1746	583
9	Shellcode	1133	378
10	Worms	130	44

Kajian terhadap dataset UNSW-NB15 menunjukkan bahwa terdapat sejumlah kelebihan yang dapat dianalisis melalui beberapa parameter yang telah dibahas dalam beragam penelitian sebelumnya. Kelebihan dataset UNSW-NB15 di antaranya adalah :

3. Fitur yang lengkap dan menyeluruh

   Dataset mencakup 49 fitur merepresentasikan beragam perilaku jaringan, sehingga menjadikannya cocok untuk melatih berbagai model *machine learning*. Dataset ini menyediakan lebih dari 2,5 juta *instance* lalu lintas jaringan, bersifat normal maupun intrusi, bermanfaat untuk mengembangkan sistem deteksi intrusi yang lebih tangguh (B. Sharma et al., 2023).

4. Beragam jenis serangan

   Dataset ini mencakup berbagai jenis serangan siber, termasuk Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, dan Worms, sehingga memungkinkan dilakukannya evaluasi yang komprehensif terhadap NIDS (Kayode Saheed et al., 2022).

5. Keterkaitan dengan Kondisi Nyata

   UNSW-NB15 diakui karena representasinya yang realistis terhadap lalu lintas jaringan modern, yang mencakup aktivitas normal maupun berbahaya, sehingga sangat relevan untuk tantangan keamanan siber saat ini (Sehrawat & Singh, 2023).

6. Mendukung Penerapan Teknik lanjutan

   Dataset ini telah digunakan secara efektif dengan teknik *machine learning* dan deep learning tingkat lanjut, seperti BiLSTM, XGBoost, dan *ensemble*

learning, yang menunjukkan fleksibilitas serta ketangguhannya dalam berbagai skenario eksperimen (Alsharaiah et al., 2024).

Walaupun memiliki berbagai kelebihan sebagaimana yang telah disebutkan, dataset UNSW-NB15 juga memiliki beberapa kekurangan dalam penggunaannya sebagai dataset deteksi intrusi:

1. Distribusi Data yang Tidak Seimbang

   Salah satu tantangan signifikan pada dataset UNSW-NB15 adalah ketidakseimbangan antara *instance* normal dan intrusi, yang dapat mempengaruhi kinerja model *machine learning* (Soflaei et al., 2024).
2. Kompleksitas Komputasi

   Jumlah fitur dan *instance* yang besar dapat menyebabkan biaya komputasi yang tinggi untuk proses pelatihan dan pengujian model. Teknik seleksi fitur dan reduksi dimensi sering kali diperlukan agar dataset lebih mudah dikelola (Das et al., 2022).
3. Kendala dalam Skalabilitas dan Interpretabilitas

   Model *representation learning* dan *deep learning* yang diterapkan pada dataset dapat mengalami keterbatasan dalam hal *skalabilitas* dan *interpretabilitas*, sehingga menyulitkan penerapan model-model tersebut pada skenario dunia nyata.

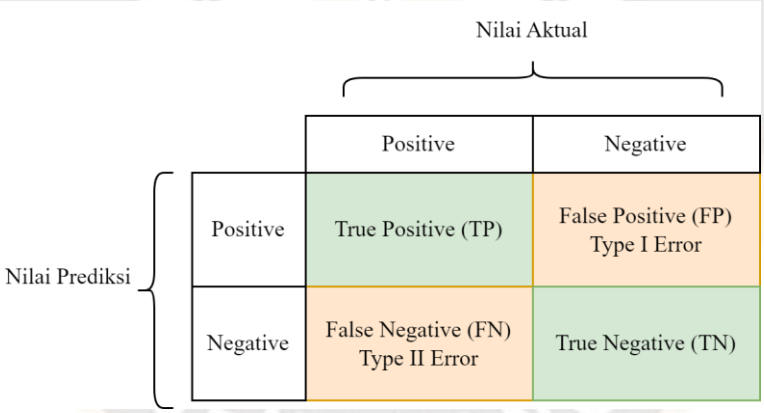
2.2.18 Metrik Evaluasi

Metrik yang digunakan untuk mengevaluasi model deteksi intrusi adalah akurasi, presisi, *recall*, F1 *score*, dan *false-alarm rate* (FAR). *Recall* dan akurasi adalah metrik evaluasi yang paling banyak digunakan dalam berbagai artikel (Yang et al., 2022). Metrik evaluasi didapatkan dari data pada *confusion matrix* yang dihasilkan pada tahap evaluasi model. Pada *confusion matrix* data dibagi dalam empat kategori yaitu *true positive (*TP*), false positive (*FP*), true negative (*TN*),* dan *false negative (*FN*)*. TP adalah ketika model mengklasifikasikan serangan sebagai sebuah serangan. TN adalah ketika model mengklasifikasikan lalu lintas normal sebagai normal. FP adalah ketika model mengklasifikasikan lalu lintas normal sebagai serangan sedangkan FN adalah saat model mengklasifikasikan sebuah

serangan sebagai lalu lintas normal. *Confusion matrix* diperlihatkan pada Gambar 2.7.

1. Akurasi

Akurasi didefinisikan sebagai jumlah data yang diprediksi benar terhadap seluruh *dataset* uji. Semakin besar nilai dari akurasi, maka model *machine learning* dihasilkan dinilai semakin baik. Akurasi berfungsi sebagai ukuran yang baik bagi *dataset* latih yang berisi kelas data seimbang (Yousuf & Mir, 2022). Pada deteksi intrusi, akurasi dapat digunakan untuk memberikan gambaran umum sejauh mana model berhasil dalam membedakan antara lalu lintas normal dengan lalu lintas mencurigakan. Hal ini menjadikan akurasi bisa digunakan sebagai indikator awal dalam menentukan kinerja model. Namun pada data yang tidak seimbang untuk membuat model deteksi intrusi, akan menyebabkan bias atau keliru pada model. Oleh karena itu dibutuhkan indikator lain dalam membuat model deteksi yang tepat.



Gambar 2.7 *Confusion matrix*

Berikut ini dijelaskan matriks evaluasi yang digunakan pada penelitian :

2. Akurasi

Akurasi didefinisikan sebagai jumlah data yang diprediksi benar terhadap seluruh *dataset* uji. Semakin besar nilai dari akurasi, maka model *machine learning* dihasilkan dinilai semakin baik. Akurasi berfungsi sebagai ukuran yang baik bagi *dataset* latih yang berisi kelas data seimbang (Yousuf & Mir, 2022). Pada deteksi

intrusi, akurasi dapat digunakan untuk memberikan gambaran umum sejauh mana model berhasil dalam membedakan antara lalu lintas normal dengan lalu lintas mencurigakan. Hal ini menjadikan akurasi bisa digunakan sebagai indikator awal dalam menentukan kinerja model. Namun pada data yang tidak seimbang untuk membuat model deteksi intrusi, akan menyebabkan bias atau keliru pada model. Oleh karena itu dibutuhkan indikator lain dalam membuat model deteksi yang tepat. Akurasi dirumuskan mengikuti

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.19}$$

3. Presisi

Presisi merupakan rasio antara jumlah serangan yang berhasil diprediksi dengan benar terhadap seluruh data yang diprediksi sebagai serangan. Nilai presisi yang tinggi menunjukkan bahwa model memiliki kemampuan yang baik dalam meminimalkan kesalahan prediksi positif *(false positive*). Dalam konteks deteksi intrusi, hal ini mengindikasikan bahwa alarm yang dihasilkan cenderung merepresentasikan serangan yang sebenarnya. Sebaliknya, nilai presisi yang rendah menunjukkan tingginya jumlah *false* alarm. Oleh karena itu, presisi yang tinggi penting untuk meningkatkan keandalan sistem dalam membedakan antara intrusi yang valid dan kesalahan deteksi, meskipun tetap perlu dipertimbangkan bersama metrik lain seperti *recall* untuk memperoleh evaluasi kinerja yang komprehensif. Presisi dirumuskan sebagai

$$Precision = \frac{TP}{TP + FP} \tag{2.20}$$

4. Recall

Recall juga dikenal sebagai *True Positive Rate* (TPR), merupakan rasio antara jumlah serangan yang berhasil diprediksi dengan benar terhadap seluruh kejadian serangan yang sebenarnya. Dengan kata lain, *recall* mengukur kemampuan model dalam mendeteksi seluruh serangan yang ada. Nilai *recall* yang

tinggi menunjukkan bahwa model mampu mengidentifikasi sebagian besar serangan, sehingga risiko terlewatnya intrusi (*false negative*) menjadi lebih kecil. Oleh karena itu, semakin tinggi nilai *recall*, semakin baik kemampuan model dalam mendeteksi serangan, meskipun tetap perlu dipertimbangkan bersama metrik lain untuk memperoleh evaluasi kinerja yang menyeluruh. *Recall* dirumuskan

$$Recall = \frac{TP}{TP + FN} \tag{2.21}$$

5. *F1-Score*

F1-score merupakan metrik evaluasi yang digunakan untuk mengukur kinerja model klasifikasi dengan mempertimbangkan keseimbangan antara presisi dan *recall*. F merupakan singkatan dari F-*measure* atau fungsi parameter matematika yang diperkenalkan oleh ilmuwan komputer C. J. van Rijsbergen pada tahun 1979 (Christen et al., 2023). Sedangkan angka 1 menunjukkan bobot kepentingan yang seimbang (1:1) antara Presisi dan *Recall*. Nilai *F1-score* diperoleh dari rata-rata harmonik antara kedua metrik tersebut, sehingga memberikan penilaian yang lebih representatif ketika terdapat ketidakseimbangan antara kemampuan model dalam mengidentifikasi prediksi positif yang benar dan kemampuannya dalam mendeteksi seluruh kejadian positif. Secara konseptual, *F1-score* akan bernilai tinggi apabila model mampu menghasilkan nilai presisi dan *recall* yang sama-sama tinggi. Sebaliknya, apabila salah satu nilai tersebut rendah, maka *F1-score* akan ikut menurun secara signifikan. Hal ini menunjukkan bahwa metrik ini sensitif terhadap ketidakseimbangan performa model. *F1-Score* didefinisikan dengan rumus

$$F - measure = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.22}$$

Dalam konteks deteksi intrusi, *F1-score* menjadi penting karena mampu merepresentasikan kompromi antara kesalahan prediksi positif (*false positive*) dan kegagalan mendeteksi serangan (*false negative*). Dalam konteks ini, penggunaan *F1-score* memberikan gambaran yang lebih komprehensif terhadap kinerja model

dibandingkan dengan penggunaan satu metrik tunggal seperti presisi atau *recall* saja.

6. *False Positive Rate (FPR)*

FPR adalah perbandingan antara jumlah data normal yang dianggap sebagai serangan terhadap jumlah semua data normal. Jika nilai FPR semakin kecil maka model *machine learning* yang dihasilkan dianggap semakin baik. FPR dirumuskan sebagai

$$FPR = \frac{FP}{FP + TN} \tag{2.23}$$

7. Kurva *Receiving Operating Characteristics (ROC)*

Kurva ROC adalah kurva yang dibuat dengan *recall* pada sumbu y dan FPR pada sumbu x melintasi ambang batas yang berbeda. *Area Under the ROC Curve* (AUC) didefinisikan sebagai area yang berada di bawah kurva. Model akan dikatakan lebih baik jika nilainya semakin mendekati 1.

$$AUC = \int_0^1 \frac{TP}{TP + FN} d\frac{FP}{TN + FP} \tag{2.24}$$

2.2.19 Hipotesis Penelitian

Berdasarkan hasil kajian teori, analisis penelitian terdahulu, serta kerangka konseptual yang telah disusun, dapat dipahami bahwa permasalahan deteksi intrusi pada jaringan multimedia *Internet of Things* (IoT) tidak hanya berkaitan dengan kemampuan model klasifikasi dalam mengenali pola serangan, tetapi juga dipengaruhi oleh kualitas data, relevansi fitur, distribusi kelas, serta stabilitas proses pembelajaran model. Kompleksitas lalu lintas jaringan kamera IP yang bersifat dinamis, heterogen, dan berdimensi tinggi menyebabkan pendekatan deteksi intrusi konvensional sering mengalami keterbatasan dalam hal akurasi, generalisasi, maupun tingkat kesalahan deteksi (*false alarm rate*).

Sejumlah penelitian sebelumnya menunjukkan bahwa pendekatan berbasis *machine learning* dan *deep learning* memiliki kemampuan yang lebih baik dalam mempelajari pola lalu lintas jaringan dibandingkan metode berbasis aturan (*rule-based detection*) (Wang et al., 2021). Namun demikian, performa model pembelajaran mendalam sangat dipengaruhi oleh kualitas fitur masukan, keseimbangan distribusi data, serta konfigurasi parameter pelatihan yang digunakan. Oleh karena itu, peningkatan performa sistem deteksi intrusi tidak dapat dicapai hanya melalui penggunaan model klasifikasi yang kompleks, tetapi juga memerlukan strategi optimasi data dan parameter pembelajaran yang tepat (Vigneswaran et al., 2018).

Dalam penelitian ini, *Deep Feedforward Neural Network* (DFNN) dipilih sebagai model utama karena memiliki kemampuan untuk memodelkan hubungan non linier dan pola kompleks pada data lalu lintas jaringan. DFNN bekerja melalui mekanisme propagasi maju (*forward propagation*) pada beberapa lapisan tersembunyi (*hidden layer*) sehingga mampu melakukan proses klasifikasi secara adaptif terhadap pola trafik normal maupun serangan (Gamage & Samarabandu, 2020). Berdasarkan karakteristik tersebut, penelitian ini merumuskan hipotesis pertama,

H1: Penerapan DFNN mampu melakukan klasifikasi lalu lintas jaringan kamera IP secara efektif sehingga dapat meningkatkan performa deteksi intrusi dibandingkan pendekatan konvensional.

Selain model klasifikasi, penelitian ini juga mempertimbangkan pentingnya kualitas fitur masukan dalam proses pembelajaran model (Thakkar & Lohiya, 2023). Penggunaan fitur yang *redundan* atau tidak relevan berpotensi menurunkan kemampuan generalisasi model dan meningkatkan kompleksitas komputasi (Zhang et al., 2023). Oleh karena itu, penelitian ini menerapkan kombinasi Pearson Correlation, Spearman Correlation, dan Mutual Information (PCSMI) sebagai pendekatan seleksi fitur hibrida. Kombinasi ketiga metode tersebut diharapkan mampu mengevaluasi hubungan linier, hubungan berbasis peringkat, serta

ketergantungan informasi antar fitur secara lebih komprehensif. Berdasarkan pertimbangan tersebut, dirumuskan hipotesis kedua,

H2: Penerapan seleksi fitur berbasis PCSMI mampu meningkatkan kualitas fitur masukan sehingga berdampak pada peningkatan akurasi, presisi, *recall*, dan *F1-score* pada model DFNN.

Penelitian ini juga mempertimbangkan permasalahan ketidakseimbangan kelas (*class imbalance*) yang umum terjadi pada dataset deteksi intrusi. Dominasi kelas tertentu dapat menyebabkan model cenderung bias terhadap kelas mayoritas dan mengalami penurunan kemampuan dalam mendeteksi serangan minoritas (Pramanick et al., 2025). Untuk mengatasi permasalahan tersebut, digunakan metode *Synthetic Minority Oversampling Technique* (SMOTE) sebagai teknik penyeimbangan data (Osa et al., 2024). Selain itu, optimasi proses pelatihan dilakukan melalui penggunaan optimizer AdamW, penyesuaian *learning rate*, dan pengaturan *batch size* guna meningkatkan stabilitas proses pembelajaran dan kemampuan generalisasi model. Berdasarkan pendekatan tersebut, dirumuskan hipotesis ketiga,

H3: Optimasi model melalui SMOTE, penggunaan AdamW, penyesuaian *learning rate*, dan *batch size* mampu meningkatkan kemampuan generalisasi model serta menurunkan *false positive rate* (FPR) pada sistem deteksi intrusi jaringan kamera IP.

Mengacu pada uraian di atas, maka ketiga hipotesis tersebut menjadi dasar konseptual dalam penelitian ini untuk menguji hubungan antara optimasi data, seleksi fitur, dan konfigurasi model terhadap peningkatan performa sistem deteksi intrusi berbasis DFNN pada lingkungan jaringan kamera IP. Selanjutnya, pengujian terhadap hipotesis tersebut dilakukan melalui tahapan perancangan sistem, implementasi model, dan evaluasi performa sebagaimana dijelaskan pada bab berikutnya.