BAB 2
LANDASAN TEORI

Bab ini menyajikan landasan teori yang mendasari pengembangan sistem deteksi anomali jaringan berbasis XGBoost (*Extreme Gradient Boosting*), XAI (*Explainable Artificial Intelligence*) menggunakan SHAP (*SHapley Additive exPlanations*) dan LIME (*Local Interpretable Model-agnostic Explanations*), serta RAG (*Retrieval-Augmented Generation*) untuk penjelasan kontekstual. Setiap teori yang dibahas dipilih berdasarkan relevansinya langsung terhadap komponen sistem yang dikembangkan, sehingga landasan ini berfungsi sebagai kerangka konseptual yang menghubungkan pendekatan teknis dengan permasalahan yang diteliti.

2.1 Deteksi Anomali Pada Jaringan

Deteksi anomali jaringan berbasis *machine learning* menghadapi tantangan fundamental berupa keterbatasan transparansi model *black-box* yang dihasilkan tidak disertai penjelasan yang memadai sehingga *security analyst* kesulitan memvalidasi hasil deteksi dan membedakan *false positive* dari ancaman nyata [11, 12]. Permasalahan ini semakin kritis mengingat kompleksitas dan volume serangan jaringan yang terus meningkat, sehingga analis keamanan tidak dapat lagi mengandalkan inspeksi manual dalam skala operasional [5]. Ketidakmampuan model untuk memberikan alasan di balik setiap keputusan deteksi menyebabkan analis harus menghabiskan waktu yang signifikan untuk memverifikasi setiap *alert* secara individual, yang pada gilirannya memperlambat respons terhadap insiden yang sesungguhnya kritis [6]. Kondisi ini menghambat proses investigasi insiden secara efektif dan menurunkan tingkat kepercayaan terhadap sistem deteksi otomatis dalam lingkungan operasional, sehingga mendorong kebutuhan akan pendekatan yang tidak hanya akurat tetapi juga *interpretable* [13].

Data *network traffic* memiliki karakteristik yang secara inheren menantang untuk pemodelan, meliputi distribusi kelas yang sangat tidak seimbang antara *traffic* normal dan serangan, dimensionalitas tinggi dengan banyak fitur yang berkorelasi, serta pola serangan yang terus berevolusi mengikuti teknik-teknik baru yang dikembangkan oleh pelaku ancaman [14]. Penggunaan *machine learning* pada data *network traffic* terbukti meningkatkan akurasi deteksi secara signifikan dibandingkan metode berbasis aturan konvensional, sekaligus memberikan

kemampuan adaptasi terhadap pola serangan yang terus berkembang [15]. Berbagai penelitian juga menunjukkan bahwa sistem berbasis *machine learning* yang dilengkapi dengan kemampuan *explainability* mampu mempertahankan akurasi tinggi sekaligus memenuhi kebutuhan analis dalam memahami dasar keputusan deteksi, sehingga integrasi XAI ke dalam *pipeline* IDS menjadi arah penelitian yang semakin relevan [16].

2.2 Dataset GeNIS

Dataset GeNIS (*GECAD Network Intrusion Scenarios*) menyediakan 125 fitur *network flow-level* yang mencakup metrik paket, statistik temporal, dan informasi protokol, sehingga memungkinkan penerapan *feature engineering* untuk mengekstraksi sinyal serangan secara akurat [17]. *Dataset* ini dihasilkan dari simulasi jaringan lanjutan pada platform Airbus CyberRange oleh tim peneliti GECAD, ISEP (*Polytechnic of Porto*), dan dirilis dalam bentuk berkas PCAPNG mentah maupun *flow* CSV yang telah diagregasi pada empat interval waktu berbeda (5, 10, 30, dan 60 detik) [17]. *Dataset* ini mencakup delapan skenario serangan bertahap yang dieksekusi secara modular oleh penyerang pada jaringan Kali-*User* dan Kali-*Remote*, serta tiga skenario *traffic benign* yang merepresentasikan aktivitas pengguna, administrator, dan latar belakang jaringan perusahaan [17]. Penelitian ini menggunakan dua komponen utama dari GeNIS: *split train/test* resmi (*genis-60-sec-train.csv* dan *genis-60-sec-test.csv*) sebagai basis pelatihan dan evaluasi internal model (Bagian 4.3.2), dan arsip *3-scenarios.zip* sebagai data evaluasi eksternal tambahan di luar *split train/test* (Bagian 4.6).

Arsip *3-scenarios.zip* berisi rekaman *flow* dari kedelapan skenario tersebut, terbagi menjadi lima berkas *scenario-{1..5}-disrupt.csv* yang merepresentasikan skenario serangan bertipe gangguan operasional bisnis melalui serangan *Denial of Service* (DoS) atau penurunan ketersediaan layanan, dan tiga berkas *scenario-{6..8}-authtest.csv* yang merepresentasikan skenario percobaan pembobolan kredensial administrator menggunakan kata sandi umum [17]. Setiap berkas skenario tersedia dalam keempat granularitas agregasi *flow* yang sama dengan *dataset* utama. Penelitian ini menggunakan folder agregasi *60 detik* agar semantik fitur statistik *flow* (mis. *Rate*, *Load*, *Dur*) tetap konsisten dan dapat diperbandingkan secara *apple-to-apple* dengan model yang dilatih pada agregasi 60 detik yang sama, sehingga ketiga granularitas lainnya (5, 10, dan 30 detik) tidak digunakan untuk menghindari *confound* akibat perbedaan semantik fitur. Untuk memastikan hasil

evaluasi tidak bergantung pada satu berkas skenario tertentu, seluruh delapan berkas skenario pada granularitas 60 detik dievaluasi menggunakan model *frozen* yang identik (Lampiran 5), dan *scenario-3-disrupt.csv* dipilih sebagai representasi utama yang dibahas secara rinci pada Bagian 4.6 karena menghasilkan performa yang konsisten dengan ketujuh skenario lainnya (akurasi 98,91%–99,65% di seluruh skenario).

2.3 Pengumpulan Data PCAP dan Ekstraksi Fitur Flow dengan HERA

PCAP (*Packet Capture*) adalah format *file* standar yang digunakan oleh *capturer* untuk menyimpan paket jaringan yang ditangkap langsung dari antarmuka jaringan [18]. Format ini berfungsi sebagai sumber data primer dalam proses pembuatan *dataset* deteksi intrusi, karena merekam setiap paket beserta informasi konteks jaringannya secara lengkap. Dalam konteks *network intrusion detection*, terdapat dua pendekatan utama untuk menganalisis data PCAP, yaitu analisis berbasis paket dan analisis berbasis *flow* [18]. Data berbasis *flow* mengagregasi paket-paket ke dalam rekaman *flow* yang meringkas aktivitas komunikasi satu sesi, menjadikannya lebih efisien dalam hal penyimpanan dan komputasi dibandingkan analisis berbasis paket [18]. Pendekatan berbasis *flow* sekaligus lebih cocok untuk sistem deteksi intrusi berskala besar karena volume data yang diproses jauh lebih kecil tanpa kehilangan informasi statistik yang relevan secara domain.

Analisis dalam penelitian ini menggunakan pendekatan berbasis *flow*, karena data *flow* menawarkan pandangan yang lebih luas terhadap *traffic* jaringan dan sangat cocok untuk mendeteksi pola berbahaya, serta lebih efisien untuk sistem berskala besar dibandingkan analisis berbasis paket yang membutuhkan lebih banyak sumber daya [18]. Dari sisi lapisan jaringan, *Layer* 2 menyediakan informasi MAC *address* dan pola ARP yang relevan untuk mendeteksi *ARP poisoning*. *Layer* 3 menyediakan informasi *routing* dan pola volume *traffic* yang berkaitan dengan DDoS dan *IP spoofing*. Sementara itu, *Layer* 4 menyediakan *TCP flags* dan *port number* yang penting untuk mendeteksi *SYN flood* dan *port scanning*. Pendekatan ini sekaligus bersifat *privacy-preserving* karena tidak mengakses isi komunikasi pada lapisan aplikasi, sehingga analisis dapat dilakukan tanpa membuka *payload* yang mungkin bersifat sensitif.

5-tuple flow aggregation merupakan metode pengelompokan paket jaringan berdasarkan lima atribut identitas koneksi yang sama [18]. Kelima atribut tersebut adalah *source IP address* (*saddr*), *destination IP address* (*daddr*), *source port*

(*sport*), *destination port* (*dport*), dan *protocol* (*proto*). Seluruh paket dengan kelima atribut yang identik dikelompokkan menjadi satu rekaman *flow*, kemudian diekstraksi fitur statistiknya yang meliputi durasi, jumlah paket, jumlah *byte*, distribusi *TCP flags*, dan statistik temporal. Data *flow* beserta fitur-fitur tersebut kemudian disimpan dalam format CSV yang siap digunakan oleh algoritma *machine learning* [18]. Representasi ini memungkinkan model untuk mempelajari pola perilaku komunikasi secara holistik dari satu sesi, bukan dari paket individual yang tidak memiliki konteks sesi secara utuh.

HERA (*Holistic nEtwork featuRes Aggregator*) adalah alat *open-source* yang dikembangkan oleh kelompok riset GECAD di Porto, Portugal [18]. Alat ini dirancang untuk menghasilkan berkas *flow* dan *dataset* berlabel dari berkas PCAP dengan fitur yang dapat dikustomisasi sesuai kebutuhan penelitian. Pengembangan HERA dilatarbelakangi oleh ditemukannya berbagai ketidakakuratan pada *CICFlowMeter* yang banyak digunakan dalam penelitian sebelumnya [18, 19]. Ketidakakuratan tersebut meliputi kesalahan penandaan terminasi koneksi TCP saat mendeteksi FIN *flag* meskipun protokol belum selesai, *timing flaw* yang mengubah arah sumber dan tujuan pada *flow* yang dihasilkan, duplikasi fitur, dan kesalahan deteksi protokol. Validasi pada *dataset* UNSW-NB15 dan CIC-IDS2017 menunjukkan bahwa model yang dilatih dengan *dataset* HERA secara konsisten mengungguli model asli dalam kemampuan generalisasi, dengan peningkatan *macro-averaged F1-Score* antara 7% hingga 14% [20].

HERA mengatasi permasalahan *CICFlowMeter* dengan menggunakan *Argus* sebagai *flow exporter backend* yang andal [18]. *Argus* telah direkomendasikan oleh *European Union Agency for Cybersecurity* (ENISA) dan telah digunakan dalam pembuatan *dataset* seperti UNSW-NB15 dan BoT-IoT, yang menjadikannya pilihan yang sudah tervalidasi secara luas dalam komunitas riset keamanan jaringan [18]. *Argus* merupakan alat serbaguna yang memungkinkan penangkapan dan pelaporan data *flow* sekaligus mengintegrasikan analisis *traffic* jaringan secara *real-time* dengan berbagai alat lainnya [18]. Melalui mekanisme ini, fitur *State_INT* diinferensikan dari kondisi ketika paket SYN terdeteksi namun tidak ada SYN-ACK hingga *flow timeout*, bukan dibaca langsung dari *header* paket. Hal ini menjadikan *Argus* lebih akurat dalam merekonstruksi siklus hidup koneksi TCP dibandingkan *CICFlowMeter* yang memiliki implementasi keliru pada penanganan terminasi TCP [18, 19].

2.4 Extreme Gradient Boosting

XGBoost (*Extreme Gradient Boosting*) adalah algoritma *ensemble* berbasis *gradient boosting* yang membangun model secara iteratif melalui *decision tree* sebagai *weak learner*, setiap pohon dilatih untuk meminimalkan kesalahan residual yang dihasilkan oleh pohon sebelumnya [21]. Berbeda dari *gradient boosting* konvensional, XGBoost secara eksplisit menyertakan term regularisasi dalam fungsi objektifnya sehingga kompleksitas pohon dapat dikontrol langsung selama proses pelatihan [22]. Kemampuan ini menjadikan XGBoost unggul dalam menangani data berdimensi tinggi dengan distribusi kelas yang tidak seimbang, sebagaimana lazim ditemukan pada *dataset* deteksi intrusi jaringan [10].

Proses pelatihan XGBoost dirancang untuk meminimalkan fungsi objektif yang terdiri dari *loss function* dan *term* regularisasi. Fungsi objektif tersebut diformulasikan sebagai berikut [21]:

$$J = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{2.1}$$

Dengan $L(y_i, \hat{y}_i)$ adalah *loss function* yang mengukur selisih antara nilai aktual y_i dan nilai prediksi $\hat{y}_i$; $\Omega(f_k)$ adalah term regularisasi yang mengontrol kompleksitas setiap pohon f_k; serta K adalah jumlah pohon dalam *ensemble*. Pada setiap iterasi t, fungsi objektif diaproksimasi menggunakan ekspansi Taylor orde kedua dari fungsi *loss*, sebagaimana diformulasikan sebagai berikut [21]:

$$J^{(t)} \approx \sum_{i=1}^{n} \left[L\left(y_i, \hat{y}_i^{t-1}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \tag{2.2}$$

Dengan g_i dan h_i adalah turunan pertama dan kedua dari fungsi *loss* terhadap prediksi sebelumnya $\hat{y}_i^{t-1}$, yang masing-masing merepresentasikan arah dan kelengkungan (*curvature*) fungsi *loss* pada iterasi tersebut [21].

Untuk mengevaluasi kandidat pemisahan pada setiap *node*, XGBoost menggunakan formula *gain* yang mengukur penurunan nilai objektif akibat suatu pemisahan. Suatu pemisahan dinilai layak apabila nilai *gain* positif, sebagaimana diformulasikan sebagai berikut [22]:

$$\text{Gain} = \frac{1}{2} \left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma \tag{2.3}$$

Dengan $G_L = \sum_{i \in I_L} g_i$ dan $G_R = \sum_{i \in I_R} g_i$ adalah jumlah *gradient* orde pertama untuk *instance* di node kiri (I_L) dan kanan (I_R); $H_L = \sum_{i \in I_L} h_i$ dan $H_R = \sum_{i \in I_R} h_i$ adalah jumlah *Hessian* orde kedua untuk masing-masing node; λ adalah koefisien regularisasi L2 yang mencegah bobot daun terlalu besar; dan γ adalah ambang batas minimum *gain* yang harus dipenuhi agar pemisahan dilakukan [22].

Pemisahan pada suatu *node* dilakukan hanya apabila nilai Gain > 0, sehingga mekanisme regularisasi yang melibatkan γ dan λ secara bersama-sama menjaga keseimbangan antara akurasi pelatihan dan kemampuan generalisasi model. Dengan demikian, XGBoost memiliki mekanisme pencegahan *overfitting* yang terintegrasi langsung dalam proses pemilihan struktur pohon, tanpa memerlukan prosedur *pruning* terpisah setelah pelatihan selesai [22].

Pemilihan XGBoost sebagai algoritma klasifikasi utama dalam penelitian ini turut mempertimbangkan dua alternatif pengklasifikasi berbasis pohon yang umum digunakan pada domain IDS, yaitu Random Forest dan LightGBM (*Light Gradient Boosting Machine*). Random Forest membangun sejumlah *decision tree* secara independen melalui *bagging* dan agregasi *voting* mayoritas, sehingga cenderung lebih stabil terhadap *noise* namun kurang mampu menangkap pola interaksi fitur yang kompleks dibandingkan pendekatan *boosting* sekuensial yang mengoreksi kesalahan residual secara iteratif. LightGBM menggunakan strategi pertumbuhan pohon *leaf-wise* dengan algoritma *histogram-based splitting* sehingga unggul dalam kecepatan pelatihan pada *dataset* berskala besar, tetapi pertumbuhan *leaf-wise* yang tidak simetris membuatnya lebih rentan terhadap *overfitting* pada *dataset* berukuran kecil hingga menengah dibandingkan pertumbuhan *level-wise* yang digunakan XGBoost.

Studi komparatif pada sistem IDS berbasis *tree ensemble* menunjukkan bahwa ketiga algoritma tersebut dapat menghasilkan performa deteksi yang relatif setara. Namun, XGBoost umumnya tetap dipilih karena kemampuannya menangani ketidakseimbangan kelas secara eksplisit melalui parameter pembobotan kelas serta mekanisme regularisasi bawaan yang mencegah *overfitting* tanpa memerlukan *tuning* tambahan yang ekstensif [23]. Pertimbangan ini selaras dengan karakteristik *dataset* GeNIS yang memiliki distribusi 13 kelas yang sangat tidak seimbang, sehingga stabilitas terhadap *class imbalance* dan kontrol *overfitting* menjadi prioritas utama dibandingkan semata-mata kecepatan komputasi.

2.5 Feature Engineering, Scaling, dan Seleksi Fitur

Proses *feature engineering* dalam penelitian ini mencakup empat tahapan yang dilakukan secara berurutan, yaitu *outlier handling*, pembentukan fitur turunan, *feature scaling*, dan seleksi fitur dua fase. Urutan tahapan ini dirancang agar setiap proses tidak menyebabkan *data leakage* dari data uji ke data latih.

2.5.1 Penanganan Outlier dengan IQR

Penanganan *outlier* dilakukan menggunakan metode IQR (*Interquartile Range*) dengan strategi *clipping*, sehingga nilai ekstrem yang merepresentasikan sinyal serangan valid tetap dipertahankan dalam distribusi data [24]. Strategi *clipping* dipilih karena penghapusan *outlier* pada data *traffic* serangan berpotensi membuang sampel yang justru merepresentasikan perilaku serangan yang valid, seperti lonjakan paket ekstrem pada serangan *flooding* [25]. Pertimbangan ini sejalan dengan prinsip bahwa *feature scaling* pada IDS harus menjaga karakteristik statistik data yang relevan secara domain [24]. Batas *clipping* atas didefinisikan sebagai berikut [26]:

$$\text{Upper Bound} = Q_3 + 3 \times \text{IQR}, \quad \text{IQR} = Q_3 - Q_1 \tag{2.4}$$

Dengan x adalah nilai fitur asli Q_1 dan Q_3 adalah kuartil pertama (persentil ke-25) dan ketiga (persentil ke-75) yang dihitung dari data latih dan IQR adalah jangkauan interkuartil yang mengukur sebaran 50% tengah distribusi [25]. Multiplier $3\times$ dipilih lebih longgar dari standar $1{,}5\times$ Tukey karena nilai ekstrem pada data *traffic* jaringan yang mengandung serangan, seperti lonjakan paket pada DDoS *flooding*, merupakan sinyal serangan yang valid dan tidak seharusnya dihilangkan dari distribusi data latih [14, 25]. Batas *clipping* ditetapkan dari data latih dan diterapkan secara konsisten ke data uji untuk menghindari *data leakage*, sehingga evaluasi pada data uji mencerminkan kondisi operasional yang sesungguhnya [27].

2.5.2 Fitur Turunan

Fitur turunan dibentuk dari kombinasi fitur mentah HERA untuk mengekspresikan pengetahuan domain secara langsung ke dalam representasi fitur,

sehingga pola anomali yang tidak tampak pada fitur mentah individual dapat dipisahkan oleh model secara lebih efektif [17]. Dalam penelitian berbasis *flow* untuk deteksi anomali jaringan, data NetFlow digunakan secara luas sebagai standar industri untuk analisis eksploratori dan ekstraksi fitur baru yang tidak dapat diungkap oleh nilai mentah secara individual [28]. Rasio antara arah *forward* dan *backward* seperti *RateAsym* dan *FlowRatio* merupakan karakteristik yang khas untuk mendeteksi serangan seperti DoS dan *port scanning* yang menghasilkan pola *traffic* sangat tidak seimbang antara sumber dan tujuan [25].

2.5.3 Feature Scaling

Feature scaling dilakukan menggunakan StandardScaler yang mentransformasi setiap fitur sehingga memiliki *mean* nol dan standar deviasi satu, dengan tujuan memastikan seluruh fitur berada pada skala yang sebanding sehingga perbedaan magnitudo tidak mendistorsi proses pembelajaran [24]. Normalisasi ini memastikan bahwa seluruh fitur berada pada skala yang sebanding sehingga perbedaan magnitudo antar fitur tidak mendistorsi proses pembelajaran. Transformasi tersebut diformulasikan sebagai berikut [29]:

$$z = \frac{x - \mu}{\sigma} \tag{2.5}$$

Dengan x adalah nilai fitur pada suatu sampel, μ adalah rata-rata fitur yang dihitung dari data latih dan σ adalah standar deviasi fitur yang dihitung dari data latih [29]. StandardScaler diterapkan setelah SMOTE dan sebelum integrasi XAI. XGBoost sebagai model berbasis pohon tidak sensitif terhadap skala fitur, namun LIME melakukan normalisasi Z-*score* $((x_i - \mu_i)/\sigma_i)$ pada data masukan sebelum melatih model regresi linier lokal (*local linear regression model*), sehingga normalisasi fitur sebelum pemanggilan LIME diperlukan agar koefisien yang dihasilkan mencerminkan kontribusi fitur yang sebanding [30].

2.5.4 Seleksi Fitur Dua Fase (Two-Phase Feature Selection)

Seleksi fitur dilakukan melalui dua fase yang disebut strategi *two-phase feature selection*. Fase A memilih 20 fitur teratas berdasarkan skor *importance* bawaan XGBoost yang dihitung dari kontribusi setiap fitur terhadap penurunan impuritas (*gain*) secara rata-rata di seluruh *decision tree* dalam *ensemble* [22].

Pemilihan 20 fitur sebagai ambang batas *baseline* mengikuti praktik yang digunakan dalam penelitian IDS berbasis XGBoost sebelumnya, di mana 20 fitur esensial teratas diidentifikasi menggunakan algoritma XGBoost untuk seleksi fitur [31]. Penelitian menunjukkan bahwa pemilihan sejumlah kecil fitur teratas berdasarkan skor *importance* XGBoost sudah mampu memberikan performa yang kompetitif dengan penggunaan seluruh fitur, sebagaimana divalidasi pada berbagai skenario deteksi intrusi [32].

Fitur-fitur dengan potensi *data leakage* seperti *Ssaddr* dan *Sdaddr* dieksklusi dari seleksi karena nilai keduanya sangat bergantung pada topologi jaringan spesifik. Penelitian lintas-*dataset* menunjukkan bahwa *dataset-specific feature biases* secara signifikan mendegradasi performa lintas *dataset* meskipun performa dalam distribusi data latih tetap tinggi, sehingga fitur yang terlalu spesifik terhadap lingkungan pengumpulan data seperti alamat IP harus dieksklusi untuk mencegah model *overfit* ke topologi jaringan tertentu [27]. Eksklusi ini dilakukan sebelum proses seleksi dijalankan sehingga kedua fitur tersebut tidak pernah masuk ke kandidat 20 fitur teratas, terlepas dari skor *importance* yang dihasilkan model.

Fase B menerapkan *conditional swap feature selection* sebagai *domain knowledge post-filter* untuk mengatasi *dataset-specific feature bias* yang dapat merusak generalisasi model IDS di lingkungan nyata [27, 33]. Berdasarkan prinsip ini, fitur pada zona peringkat 16–20 dapat digantikan oleh fitur domain yang bersifat interpretatif dan *protocol-agnostic* apabila skor *importance*-nya memenuhi ambang batas yang ditetapkan, sementara peringkat 1–15 dilindungi sepenuhnya untuk mencegah fitur kritis seperti *BytePerPkt* dan *RateAsym* tergantikan. Prinsip perlindungan zona atas ini diadaptasi dari pendekatan identifikasi fitur pemicu bias yang dikembangkan oleh, dengan keputusan terhadap fitur yang berpotensi bias diserahkan kepada pertimbangan pakar domain [34].

2.6 Penanganan Ketidakseimbangan Kelas dan SMOTE

Ketidakseimbangan kelas (*class imbalance*) pada *dataset* deteksi intrusi menyebabkan model *machine learning* cenderung bias terhadap kelas mayoritas dan mengabaikan kelas serangan langka namun kritis [35, 36]. Pada *dataset* GeNIS, kondisi ini terlihat nyata pada kelas *recon-dns* yang hanya memiliki 16 sampel pada data latih, sementara kelas benign mendominasi dengan lebih dari 100.000 sampel. Tanpa penanganan yang tepat, model berpotensi mencapai akurasi keseluruhan yang tinggi namun gagal mendeteksi kelas serangan langka, sehingga metrik *macro*

F1-*score* menjadi lebih representatif dibandingkan akurasi dalam mengevaluasi kinerja sistem pada kondisi ini [37].

SMOTE (*Synthetic Minority Over-sampling Technique*) merupakan teknik *oversampling* yang mengatasi ketidakseimbangan kelas dengan cara membangkitkan sampel sintetis pada kelas minoritas melalui interpolasi linier antara sampel yang ada dan tetangga terdekatnya dalam ruang fitur, bukan dengan menduplikasi sampel yang sudah ada [35]. Pendekatan interpolasi ini meningkatkan keragaman representasi kelas minoritas sekaligus memitigasi risiko *overfitting* yang dapat terjadi jika hanya menduplikasi sampel yang sudah ada [38]. Sampel sintetis baru x_{new} dihasilkan melalui formula berikut [39]:

$$x_{\text{new}} = x_i + (x_{nn} - x_i) \cdot \delta, \quad \delta \sim \mathscr{U}(0,1) \tag{2.6}$$

Dengan x_i adalah sampel minoritas yang dipilih sebagai titik awal; x_{nn} adalah salah satu dari k tetangga terdekat x_i dalam ruang fitur yang dipilih secara acak; dan δ adalah skalar acak dari distribusi uniform $\mathscr{U}(0,1)$ yang mengontrol posisi sampel baru di antara x_i dan x_{nn}, sehingga nilai δ mendekati 0 menghasilkan sampel yang mirip x_i dan nilai mendekati 1 menghasilkan sampel yang mirip x_{nn} [39].

Pada skenario *multiclass* dengan kelas yang sangat kecil seperti *recon-dns* yang hanya memiliki 16 sampel pada data latih, permasalahan *class imbalance* menjadi lebih kritis karena model *machine learning* yang dilatih pada data tidak seimbang cenderung unggul dalam mengidentifikasi *traffic* biasa (kelas mayoritas), namun kesulitan dalam mendeteksi jenis serangan langka yang berpotensi salah diklasifikasikan sebagai jenis serangan lain yang lebih umum [40]. Sistem IDS yang tidak menggunakan teknik *class imbalance learning* harus ditangani dengan sangat hati-hati karena model-model tersebut gagal mendeteksi serangan kelas minoritas secara efektif, dan teknik *oversampling* sintetis seperti SMOTE dapat menimbulkan masalah *overfitting* khususnya pada rasio *imbalance* yang lebih besar [40]. Oleh karena itu, nilai k disesuaikan secara adaptif menggunakan strategi *tiered*: untuk setiap kelas minoritas, nilai k ditetapkan sebagai $\min(3, |C_{\min}| - 1)$, dengan $|C_{\min}|$ menyatakan jumlah sampel pada kelas tersebut sebelum *oversampling* dan nilai maksimum $k = 3$ dipilih untuk mencegah interpolasi yang terlalu jauh dari distribusi asli. Pendekatan adaptif ini mengikuti prinsip dasar SMOTE bahwa jumlah tetangga yang digunakan tidak boleh melebihi ukuran kelas minoritas yang tersedia [38].

Strategi adaptif ini memastikan SMOTE dapat beroperasi secara valid

bahkan untuk kelas dengan sampel sangat sedikit seperti *recon-dns*, sekaligus mencegah *data leakage* karena penerapan SMOTE dibatasi hanya pada data latih setelah seleksi fitur selesai dilakukan [39]. Pembatasan ini penting untuk memastikan bahwa statistik *k*-tetangga yang digunakan dalam pembangkitan sampel sintetis tidak dipengaruhi oleh distribusi data uji. Dengan demikian, distribusi data uji tetap mencerminkan kondisi operasional nyata yang tidak dipengaruhi oleh proses *oversampling*. Praktik ini mengikuti desain eksperimental standar di mana proses *resampling* hanya diterapkan pada data latih, sedangkan model yang telah dilatih kemudian diuji pada data uji yang tetap dalam kondisi asli tanpa modifikasi [37], sehingga evaluasi model menghasilkan estimasi performa yang tidak bias terhadap distribusi *traffic* yang sesungguhnya [40].

2.7 Explainable Artificial Intelligence (XAI)

XAI menyediakan interpretasi terhadap keputusan model *black-box* dalam konteks keamanan siber, sehingga memungkinkan *analyst* memahami mengapa suatu *traffic* diklasifikasikan sebagai anomali serta memvalidasi keputusan tersebut dalam proses investigasi insiden. Penelitian ini menggunakan dua metode XAI yang saling melengkapi: SHAP untuk interpretasi global dan lokal berbasis teori permainan kooperatif, serta LIME untuk aproksimasi model linier lokal berbasis perturbasi yang bersifat *model-agnostic*.

2.7.1 SHapley Additive exPlanations (SHAP)

SHAP mengukur kontribusi setiap fitur terhadap prediksi model berdasarkan nilai Shapley dari teori permainan kooperatif, yang didefinisikan sebagai rata-rata kontribusi marginal fitur *i* di seluruh kemungkinan koalisi yang dapat dibentuk dari himpunan fitur [41]. Nilai Shapley untuk fitur i diformulasikan sebagai berikut [41, 42]:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)] \tag{2.7}$$

Dengan F adalah himpunan semua fitur dalam model, S adalah subset fitur yang sudah ada dalam koalisi, mencakup semua kemungkinan subset $S \subseteq F \setminus \{i\}$; $\frac{|S|!\,(|F|-|S|-1)!}{|F|!}$ adalah bobot yang memastikan rata-rata marginal dihitung secara adil untuk semua urutan koalisi yang mungkin, $f(S \cup \{i\})$ adalah prediksi model ketika

fitur i ditambahkan ke koalisi S, dan selisih $f(S \cup \{i\}) - f(S)$ mengukur kontribusi marginal fitur i dalam konteks koalisi tersebut [42].

Pendekatan ini memenuhi empat aksioma fundamental, yaitu *efficiency* (jumlah seluruh ϕ_i sama dengan selisih prediksi model dari ekspektasi *baseline*), *symmetry* (fitur identik memperoleh nilai yang sama), *dummy* (fitur yang tidak berpengaruh mendapatkan $\phi = 0$), dan *additivity*, yang secara bersama menjamin bahwa atribusi fitur yang dihasilkan bersifat adil, konsisten, dan dapat diperbandingkan antar sampel [41]. Pemenuhan keempat aksioma ini merupakan keunggulan teoretis SHAP dibandingkan metode atribusi fitur lainnya yang tidak memiliki jaminan properti matematis yang setara.

Untuk efisiensi komputasi, digunakan algoritma TreeSHAP yang dirancang khusus untuk model berbasis pohon seperti *decision tree*, *random forest*, dan *gradient-boosting tree* [43]. Model-model tersebut memanfaatkan struktur hierarkis pohon untuk menghitung nilai SHAP secara *bottom-up* dan memanfaatkan kondisi pemisahan fitur untuk mereduksi kompleksitas komputasi ke tingkat yang dapat diterima [43]. SHAP menghasilkan interpretasi pada tingkat global (fitur paling berpengaruh pada model secara agregat) maupun lokal (penjelasan kontribusi fitur untuk setiap prediksi individual), sehingga memungkinkan *analyst* memahami pola ancaman secara menyeluruh sekaligus menginvestigasi insiden secara individual.

2.7.2 Local Interpretable Model-agnostic Explanations (LIME)

LIME membangun penjelasan yang dapat diinterpretasikan atas prediksi model dengan memanfaatkan representasi data yang dapat dipahami manusia [44]. LIME bersifat *model-agnostic* sehingga dapat diterapkan pada sembarang model *black-box* tanpa memerlukan akses ke struktur internal model, dan terbukti efektif untuk investigasi forensik pada kasus individual [10]. Setiap *instance* $x \in \mathbb{R}^d$ direpresentasikan sebagai vektor biner $x' \in \{0,1\}^{d'}$ yang menunjukkan ada atau tidaknya komponen yang dapat diinterpretasikan [44]. LIME kemudian menghasilkan N sampel perturbasi $z' \in \{0,1\}^{d'}$ di sekitar representasi x', mengkueri setiap sampel ke model *black-box* untuk mendapatkan label prediksi, dan membobotkan setiap sampel berdasarkan kedekatannya ke *instance* asal menggunakan fungsi *kernel* eksponensial [44]. Fungsi *kernel* tersebut diformulasikan sebagai berikut [44]:

$$\pi_x(z) = \exp\left(-\frac{D(x,z)^2}{\sigma^2}\right) \tag{2.8}$$

Dengan z adalah sampel dalam ruang fitur asli yang diperoleh dari sampel perturbasi z'; $D(x,z)$ adalah fungsi jarak antara *instance* asal x dan sampel z, sehingga sampel yang lebih dekat ke x mendapatkan bobot yang lebih besar; dan σ adalah lebar *kernel* yang menentukan radius lokalitas [44]. Data masukan dinormalisasi menggunakan Z-*score* sebelum model regresi linier lokal dilatih [30]. Berdasarkan bobot tersebut, LIME mengoptimasi loss berikut untuk mendapatkan koefisien model linier lokal [44]:

$$\mathscr{L}(f,g,\pi_x) = \sum_{z,z' \in \mathscr{Z}} \pi_x(z)\big(f(z) - g(z')\big)^2 \tag{2.9}$$

Dengan $f(z)$ adalah prediksi model *black-box* untuk sampel z, $g(z')$ adalah prediksi model linier lokal untuk representasi biner z', dan $\pi_x(z)$ adalah bobot dari fungsi *kernel* pada Persamaan 2.8 [44]. Koefisien positif $w_i > 0$ berarti fitur mendorong prediksi ke kelas target, sedangkan $w_i < 0$ berarti fitur menahan prediksi.

2.8 Retrieval Augmented Generation (RAG)

SHAP dan LIME menghasilkan atribusi fitur yang akurat secara teknis, namun *output*-nya berupa nilai numerik dan visualisasi yang kurang *actionable* bagi analis operasional yang memerlukan panduan investigasi berbasis bahasa alami [6]. Oleh karena itu, RAG (*Retrieval-Augmented Generation*) diterapkan untuk mengatasi kesenjangan tersebut dengan menggabungkan *information retrieval* berbasis *LLM* dengan *natural language generation* untuk menghasilkan penjelasan kontekstual yang dapat ditindaklanjuti [45].

Secara arsitektural, sistem RAG yang diterapkan dalam penelitian ini terdiri atas dua fase operasional, yaitu fase *indexing* yang dilakukan sekali secara *offline* dan fase *retrieval-generation* yang dijalankan setiap kali permintaan penjelasan diajukan [45]. Pada fase *indexing*, seluruh dokumen *knowledge base* yang berisi deskripsi teknis, dampak, dan langkah mitigasi untuk setiap kelas serangan pada *dataset* GeNIS diproses melalui model *encoder all-MiniLM-L6-v2* untuk menghasilkan representasi vektor *embedding* [46]. Vektor *embedding* hasil proses tersebut kemudian disimpan ke dalam *Document Store* berbasis Haystack sebagai basis data vektor yang menjadi rujukan pencarian pada fase berikutnya [46].

Pada fase *retrieval-generation*, kueri disusun dari *output* XAI yang mencakup fitur dominan hasil SHAP dan kontribusi lokal hasil LIME untuk

suatu prediksi. Kueri tersebut kemudian diubah menjadi vektor *embedding* menggunakan *encoder* yang sama agar berada pada ruang representasi vektor yang konsisten dengan dokumen yang telah diindeks sebelumnya. *Retriever* selanjutnya menghitung kemiripan *cosine* antara vektor kueri dan seluruh vektor dokumen dalam *Document Store*, kemudian mengambil $k = 5$ dokumen dengan skor kemiripan tertinggi sebagai konteks [46, 47].

Dokumen-dokumen yang diambil tersebut kemudian digabungkan dengan *output* XAI ke dalam satu *prompt* terstruktur, yang selanjutnya diteruskan ke komponen *Generator* berupa *LLM* untuk menghasilkan narasi penjelasan berbahasa alami yang membumikan atribusi fitur numerik ke dalam konteks operasional yang dapat ditindaklanjuti oleh *security analyst* [45, 46]. Alur dua fase ini menjamin bahwa penjelasan yang dihasilkan tetap berbasis pada bukti yang di-*retrieve* dari *knowledge base* dan bukan murni hasil generasi bebas dari *LLM*, sehingga risiko *hallucination* yang lazim ditemukan pada sistem generatif tanpa *grounding* dapat diminimalkan [45].

Relevansi semantik antara kueri dan dokumen dalam *knowledge base* diukur menggunakan kemiripan kosinus (*cosine similarity*), yang berfungsi sebagai dasar mekanisme *retrieval* dalam sistem RAG [47]. Dokumen dengan nilai kemiripan tertinggi selanjutnya digunakan sebagai konteks untuk menghasilkan penjelasan berbasis bahasa alami, sebagaimana diformulasikan sebagai berikut [47]:

$$\text{sim}(\mathbf{q}, \mathbf{v}) = \frac{\mathbf{q} \cdot \mathbf{v}}{\|\mathbf{q}\| \cdot \|\mathbf{v}\|} \tag{2.10}$$

Dengan $\mathbf{q}$ adalah vektor *embedding* dari kueri yang dihasilkan dari *output* XAI, mencakup fitur dominan SHAP dan kontribusi lokal LIME, $\mathbf{v}$ adalah vektor *embedding* dari dokumen ke-*i* dalam *knowledge base*, dan pembagi $\|\mathbf{q}\| \cdot \|\mathbf{v}\|$ menormalisasi hasil menjadi rentang $[-1, 1]$, dengan nilai mendekati 1 menunjukkan kemiripan semantik tinggi antara kueri dan dokumen yang diambil [47]. Implementasi dalam penelitian ini menggunakan *framework* Haystack yang dipilih berdasarkan hasil *benchmark* pada studi yang disitasi dan arsitektur *pipeline* modular. Haystack menyediakan tiga komponen inti, yaitu *Document Store* sebagai basis data vektor, *Retriever* berbasis *cosine similarity*, dan *Generator* berupa *LLM* untuk menghasilkan penjelasan berbasis bahasa alami bagi *security analyst* [46].

2.9 Metrik Evaluasi Model

Evaluasi performa sistem deteksi anomali dalam penelitian ini menggunakan serangkaian metrik yang dihitung dari *confusion matrix* berukuran 13×13 untuk skenario 13-kelas, dengan empat besaran dasar: TP (*True Positive*), TN (*True Negative*), FP (*False Positive*), dan FN (*False Negative*) [6, 48]. Keempat besaran tersebut menjadi fondasi perhitungan metrik-metrik berikut [48]:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.12}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.13}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.14}$$

$$\text{Macro F1} = \frac{1}{C} \sum_{c=1}^{C} \text{F1-score}_c \tag{2.15}$$

Recall diprioritaskan dalam konteks keamanan jaringan karena *false negative* yang tidak terdeteksi berpotensi menyebabkan insiden keamanan yang serius, sementara Macro F1-*score* dipilih sebagai metrik agregat utama (*headline metric*) dalam penelitian ini karena *accuracy* tidak dapat dijadikan acuan utama akibat ketidakseimbangan kelas yang signifikan pada *dataset* GeNIS 13-kelas, di mana kelas seperti *recon-dns* hanya memiliki 16 sampel pada data latih sementara kelas *benign* memiliki lebih dari 100.000 sampel [10, 48]. Pada kondisi distribusi kelas yang timpang seperti ini, model yang gagal total mendeteksi kelas minoritas tetap dapat mencapai nilai *accuracy* yang tinggi karena dominasi kelas mayoritas dalam perhitungan, sehingga *accuracy* dalam penelitian ini hanya diposisikan sebagai metrik pelengkap (*secondary metric*) untuk memberikan gambaran umum, bukan sebagai dasar utama evaluasi kinerja sistem [37]. Macro F1-*score*, *recall*,

dan *precision* per kelas saling melengkapi dalam memberikan gambaran yang komprehensif tentang kemampuan sistem mendeteksi seluruh kategori serangan secara seimbang sekaligus meminimalkan gangguan terhadap *traffic* yang sah.

2.10 Penelitian Terkait

Tinjauan literatur pada Tabel 2.1 mengidentifikasi tiga kluster penelitian yang menjadi landasan sekaligus celah yang diisi oleh penelitian ini. Analisis komparatif dilakukan untuk mengidentifikasi aspek yang belum ditangani secara memadai oleh penelitian terdahulu, sehingga dapat dirumuskan kontribusi keterbaruan yang spesifik dan terukur.

1. IDS berbasis XAI: Hosain & Cakmak [48], Hermosilla et al. [10], Barnard et al. [49], Muhammad et al. [50], serta Gaspar et al. [51] telah membuktikan bahwa XAI secara konsisten meningkatkan interpretabilitas dan ketangguhan IDS. Namun, penelitian-penelitian tersebut belum mengintegrasikan narasi mitigasi berbasis bahasa alami yang dapat langsung ditindaklanjuti oleh praktisi keamanan jaringan.

2. Seleksi fitur dan penanganan distribusi: Thakkar & Lohiya [52, 36] mengidentifikasi perubahan infrastruktur jaringan sebagai tantangan utama IDS, sementara Eren et al. [27] menunjukkan melalui pengujian lintas dataset bahwa *dataset-specific feature bias* mendegradasi performa lintas *dataset*. Omae & Mori [33] dan Openja et al. [34] meletakkan dasar konseptual *swap feature selection* berbasis pengetahuan domain, namun prinsip tersebut belum pernah diterapkan secara konkret dalam konteks IDS berbasis XAI.

3. RAG untuk narasi kontekstual: Blefari et al. [53] dan Alquliti et al. [54] memvalidasi bahwa RAG mampu menghasilkan rekomendasi mitigasi yang *actionable* dan mengurangi *false positive*, namun implementasinya belum dikombinasikan dengan XAI multi-metode (SHAP + LIME).

Berdasarkan analisis ketiga kluster di atas, penelitian ini mengisi *gap* yang teridentifikasi melalui integrasi XGBoost + SHAP + LIME + RAG dalam satu *pipeline*, dilengkapi dengan SMOTE *tiered* adaptif dan *conditional swap feature selection* [27, 33, 34].

Tabel 2.1. Tinjauan literatur

Penulis	Judul	Tahun	Metode	Keterbaruan Penelitian Ini
Barnard et al.	Robust NIDS Through XAI	2022	XAI, NIDS	Menambahkan RAG untuk penjelasan kontekstual [49].
Gaspar et al.	XAI for IDS: LIME & SHAP on MLP	2024	MLP, LIME, SHAP	Menambahkan RAG untuk narasi *actionable* [51].
Muhammad et al.	L-XAIDS: LIME-based XAI for IDS	2025	LIME, ELI5	Menggabungkan interpretasi global-lokal SHAP dengan RAG [50].
Hosain & Cakmak	XAI-XGBoost for IoMT IDS	2025	XGBoost, SHAP, LIME	Akurasi 99,22%; menambahkan RAG untuk interpretasi kontekstual [48].
Khan et al.	XAI-Based IDS for Industry 5.0	2025	XAI, IDS	Memperluas ke narasi kontekstual *actionable* [55].
Mohale & Obagbuwa	Evaluating ML-Based IDS with XAI	2025	XGBoost, SHAP, LIME	Menerapkan pada GeNIS 13 kelas dengan evaluasi lintas distribusi [56].
Hermosilla et al.	XAI for Forensic Analysis: SHAP & LIME	2025	XGBoost, SHAP, LIME	Mengintegrasikan SHAP+LIME ke *pipeline* RAG [10].
Thakkar & Lohiya	Attack Classification of Imbalanced IDS	2023	DNN, Bagging	Mengganti Bagging dengan SMOTE *tiered* adaptif [36].
Thakkar & Lohiya	A Review on Challenges for ML-IDS	2023	ML-IDS Survey	Mengatasi tantangan infrastruktur dengan *conditional swap* seleksi fitur [52].
Eren et al.	Distributional Drift in IoT IDS	2026	RF, FFNN	Mengatasi *feature bias* via *conditional swap feature selection* [27].
Omae & Mori	Bayesian Swap Feature Selection	2023	Bayesian, swap FS	Mengadaptasi prinsip *swap* untuk integrasi *domain knowledge* IDS [33].

Lanjut di halaman berikutnya

Tabel 2.1. Tinjauan Literatur (Lanjutan)

Penulis	**Judul**	**Tahun**	**Metode**	**Keterbaruan Penelitian Ini**
Openja et al.	Bias-inducing Features in ML	2023	Feature swapping	Mengadaptasi prinsip *domain expert decides* untuk Fase B seleksi fitur [34].