**IT-EXPLORE**
*Jurnal Penerapan Teknologi Informasi dan Komunikasi*
ISSN 2828-7940, e-ISSN 2829-1727, DOI: 10.24246/itexplore.v5i2.2026.pp119-130
Volume 05 Nomor 02 Tahun 2026
Journal homepage: https://ejournal.uksw.edu/itexplore

# Implementasi algoritma *gradient boosting machine* untuk deteksi intrusi pada jaringan komputer

**Dileando Gamaliel[1], Wiwin Sulistyo[2],**
[1,2] Faculty of Information Technology, Satya Wacana Christian University, Salatiga City, Central Java, Indonesia

**Informasi Artikel**

***Riwayat Artikel:***

Dikirim:
*20 Oktober 2025*
Direvisi:
*23 November 2025*
Diterima:
*25 November 2025*

***Kata Kunci:***

*Gradient Boosting Machine,*
*Machine Learning,*
Deteksi Intrusi,
Jaringan

***Keywords:***

*Gradient Boosting Machine,*
*Machine Learning,*
*Intrusion Detection,*
*Networks*

*This is an open access article under the CC BY-SA license*

**ABSTRAK**

Penelitian ini mengkaji implementasi algoritma *Gradient Boosting Machine* (GBM) untuk deteksi intrusi jaringan menggunakan dataset *CICIDS2017* dengan kerangka kerja *CRISP-DM*. Tahapan dimulai dari *Business Understanding*, *Data Understanding*, dan *Data Preparation* yang meliputi pembersihan data, *encoding* fitur kategorikal, normalisasi, dan pembagian data (80% pelatihan, 20% pengujian). Pada tahap *Modeling*, GBM dioptimasi melalui *Grid Search* untuk kombinasi *Hyperparameter* (*learning_rate* = 0,1; *max_depth* = 5; *n_estimators* = 150) dengan *K-Fold Cross Validation* (K = 2), dan *F1-Score* dipilih sebagai metrik utama karena distribusi data tidak seimbang. Hasil evaluasi pada data pengujian menunjukkan akurasi 99,99 %, presisi 100 %, *Recall* 99,98 %, dan *F1-Score* 99,99 %, menandakan kemampuan deteksi yang sangat baik dengan minim *false negative* dan *false positive*. Dibandingkan penelitian terdahulu, model GBM ini unggul dalam hal kinerja dan kestabilan tanpa mengalami *overfitting*. Hasil ini menegaskan bahwa GBM andal untuk sistem deteksi intrusi modern dan cocok digunakan dalam lingkungan operasional dengan sumber daya komputasi terbatas

**ABSTRACT**

*This study investigates the implementation of the Gradient Boosting Machine (GBM) algorithm for network intrusion detection using the CICIDS2017 dataset within the CRISP-DM framework. The process encompasses Business Understanding, Data Understanding, and Data Preparation including data cleaning, categorical feature encoding, normalization, and data split (80 % training, 20 % testing). In the Modeling phase, GBM Hyperparameters (learning_rate = 0.1; max_depth = 5; n_estimators = 150) were optimized via Grid Search with 2-fold Cross Validation, and F1-Score was selected as the primary metric due to class imbalance. Evaluation on the test set yielded accuracy of 99.99 %, precision of 100 %, Recall of 99.98 %, and F1-Score of 99.99 %, demonstrating exceptional detection capability with minimal false negatives and false positives. Compared to previous studies, this GBM model outperforms in accuracy and stability without overfitting. These findings confirm GBM's effectiveness for modern Intrusion Detection Systems and its suitability for Deployment in resource-constrained operational environments.*

***Corresponding Author:***

**Dileando Gamaliel**
Fakultas Teknologi Informasi, Universitas Kristen Satya Wacana
Jl. Diponegoro No.52-60, Sidorejo, Salatiga, Jawa Tengah, Indonesia
Email: 672021245@student.uksw.edu

## 1. PENDAHULUAN

Pada era revolusi industri 4.0, konektivitas jaringan menjadi elemen krusial untuk memastikan setiap individu dapat mengakses internet, sehingga berbagai lembaga maupun perusahaan membangun infrastruktur jaringan mereka sendiri demi meningkatkan efisiensi operasional [1]. Meskipun membawa manfaat besar, perkembangan ini juga menimbulkan tantangan baru berupa ancaman serangan siber [2]. Umumnya, serangan ini dilakukan oleh pihak-pihak tertentu yang berusaha mengeksploitasi jaringan guna menyerang sistem komputer, sebagaimana tercermin dari laporan Badan Siber dan Sandi Negara yang mencatat 330.527.636 serangan siber sepanjang tahun 2024, yang tidak hanya mengganggu operasional jaringan tetapi juga mencakup pencurian data dari server [1], [3]. Adapun jenis serangan yang kerap terjadi antara lain *Malware, Trojan, Information Leak*, serta *DoS*, dimana *Malware* merupakan yang paling dominan dengan sektor administrasi pemerintahan sebagai sasaran utama [4].

Seiring dengan semakin beragamnya bentuk serangan siber, upaya deteksi serangan menjadi langkah krusial dalam menjaga keamanan infrastruktur jaringan [4]. Salah satu teknologi yang berperan penting dalam hal ini adalah *Intrusion Detection System* (IDS), yang secara tradisional beroperasi dengan dua metode utama, yaitu *signature based* dan *anomaly based*, namun keduanya memiliki keterbatasan terutama dalam mengidentifikasi serangan *zero day* serta dalam menangani data dengan volume besar [5]. Untuk mengatasi kekurangan ini, pendekatan berbasis *Machine Learning* mulai dimanfaatkan karena dinilai lebih adaptif [4]. Penerapan *Machine Learning* terbukti mampu meningkatkan efektivitas IDS dalam mendeteksi dan memitigasi ancaman keamanan jaringan [6]. Berbagai algoritma telah digunakan dalam IDS berbasis *machine learning*, antara lain *Support Vector Machine, Decision Tree, Random Forest, dan Naive Bayes* [7]. Dengan demikian, pemanfaatan *Machine Learning* dalam IDS dapat mengatasi keterbatasan metode konvensional dan mampu meningkatkan keamanan jaringan secara lebih optimal [8]

Meskipun banyak algoritma *Machine Learning* telah digunakan dalam pengembangan *Intrusion Detection System* (IDS), masih terdapat beberapa tantangan teknis yang perlu diatasi. Salah satu permasalahan utama adalah tingginya nilai *false positive*, yang dapat menyebabkan sistem sering kali salah mengidentifikasi aktivitas normal sebagai ancaman, sehingga mengurangi keandalan IDS [9]. Selain itu, beberapa algoritma memiliki keterbatasan spesifik, seperti *overfitting* pada *Decision Tree* yang membuat model kurang mampu menggeneralisasi data baru, sulitnya penentuan parameter optimal pada *Support Vector Machine* yang menyebabkan performa deteksi tidak konsisten, serta kebutuhan waktu pelatihan yang lama dan sumber daya komputasi tinggi pada *Neural Networks* meskipun mampu mengenali pola serangan yang kompleks [10]. Tantangan lain juga muncul saat implementasi pada dataset tertentu, seperti dataset *CICIDS2017*, di mana algoritma *Logistic Regression* menunjukkan performa yang kurang optimal dengan akurasi hanya 46,59% [11]. Masalah-masalah ini menunjukkan bahwa meskipun *Machine Learning* menawarkan solusi canggih, pemilihan algoritma yang tepat tetap menjadi kunci untuk meningkatkan kemampuan IDS dalam mendeteksi serangan siber. Sehingga penelitian ini difokuskan pada penerapan dan evaluasi algoritma *Gradient Boosting Machine* (GBM) dalam mendeteksi intrusi dengan memanfaatkan dataset *CICIDS2017*. Proses implementasi dan evaluasi dilakukan menggunakan kerangka kerja *CRISP-DM* agar lebih sistematis serta optimal, sekaligus memungkinkan analisis data secara mendalam sehingga mampu menghasilkan model deteksi yang lebih akurat [12]. Dengan demikian, penelitian ini dapat menunjukkan performa dari algoritma GBM dalam melakukan deteksi intrusi pada jaringan komputer.

## 2. KAJIAN PUSTAKA

### 2.1. State of The Art

Penelitian dalam bidang *Instrusion Detection System (IDS)* menunjukkan perkembangan signifikan melalui penerapan algoritma *Machine Learning* untuk meningkatkan akurasi deteksi terhadap ancaman siber yang semakin kompleks. Diantara berbagai pendekatan yang digunakan, *Gradient Boosting Machine (GBM)* menempati posisi terdepan sebagai salah satu algoritma *ensemble learning* yang mampu menggabungkan sejumlah model lemah menjadi model kuat melalui proses iterative berbasis *gradient descent*. Dimana pemilihan algoritma ini dilatarbelakangi oleh kemampuannya dalam mengombinasikan sejumlah model lemah menjadi model yang lebih kuat, dengan cara memperbaiki kesalahan pada setiap tahap pelatihan, sehingga tingkat akurasi dapat semakin ditingkatkan [13]. Dibandingkan dengan algoritma lain seperti *Random Forest maupun Logistic Regression*, GBM terbukti mampu meningkatkan akurasi hingga hampir

80% serta unggul dalam menekan tingkat *false positive* [10], [14]. Berbagai studi lainnya menunjukkan bahwa GBM memiliki performa unggul dibandingkan algoritma konvensional seperti *Decision Tree*, *Random Forest*, Maupun *Support Vector Machine (SVM).* Misalnya, Penelitian yang dilakukan oleh simanjuntak dan sijabat (2024) menunjukkan bahwa model berbasis *CNN* dan *Random Forest* membutuhkan sumber daya komputasi tinggi, sementara GBM dapat mencapai hasil akurasi tinggi dengan efisiensi yang lebih baik. Hasil penelitian Wardhana et al. (2022) juga menegaskan bahwa GBM mampu mengatasi masalah *overfitting* yang sering muncul pada algoritma *Decision Tree*, sekaligus memberikan hasil klarifikasi yang lebih stabil.

Sejalan dengan pemilihan algoritma GBM, pemilihat dataset juga menjadi aspek penting. Sehingga dataset *CICIDS2017* dipilih karena karakteristiknya yang sangat representatif dalam menggambarkan berbagai jenis serangan, menjadikannya salah satu dataset terbaik untuk pengembangan model IDS [15]. Dataset yang dikembangkan oleh *Canadian Institute for Cybersecurity* ini sering digunakan dalam penelitian karena cakupannya yang luas, yaitu lebih dari 80 juta koneksi jaringan yang menyerupai lalu lintas jaringan nyata [11], [16]. Selain itu, *CICIDS2017* memiliki lebih dari 70 fitur, meliputi *Timestamp*, alamat IP sumber dan tujuan, *port*, protokol, hingga kategori serangan seperti *Brute Force, DoS, Botnet,* dan *Web Attack* [11], [15], [16]. Dengan kelengkapan data yang mencakup beragam skenario serangan tersebut, dataset ini memungkinkan model untuk mempelajari serta mengenali pola serangan secara lebih optimal, sehingga model yang dihasilkan mampu memberikan deteksi yang lebih akurat [15].

Selain pemilihan algoritma dan dataset, proses pengembangan model juga memerlukan kerangka kerja yang sistematis. Oleh karena itu, penelitian ini menggunakan keranka kerja *CRISP-DM* untuk memastikan bahwa tahapan analisis dan pemodelan dapat dilakukan secara terstruktur dan optimal. Dengan pendekatan ini, setiap tahapan yang dilakukan dapat terstruktur dengan baik, memastikan bahwa model yang dihasilkan memiliki performa terbaik. Oleh karena itu, penelitian ini menghasilkan model deteksi intrusi yang lebih adaptif dan andal dalam berbagai kondisi, termasuk pada lingkungan dengan sumber daya komputasi terbatas, sekaligus mampu mengurangi risiko *overfitting* serta menurunkan tingkat *false positives*.

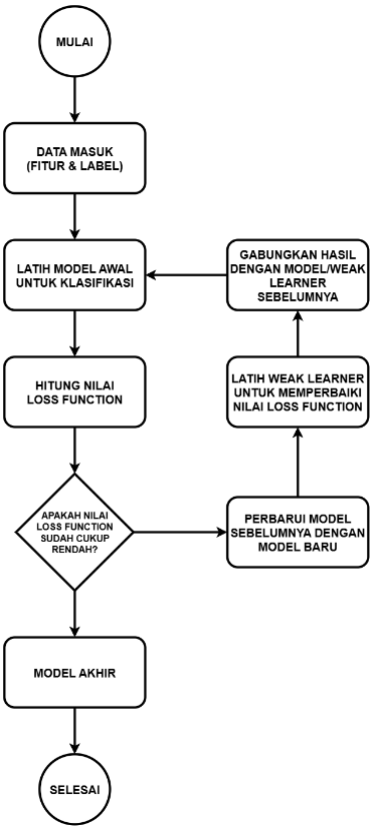
## 2.2. Landasan Teori

### 2.2.1. Sistem Deteksi Intrusi

*Intrusion Detection System* atau IDS merupakan sebuah sistem yang dibangun untuk mendeteksi aktivitas di dalam sebuah jaringan komputer [4]. Dimana nantinya administrator jaringan akan mendapat peringatan dari IDS ketika terjadi serangan seperti DDoS, maupun munculnya gangguan dalam jaringan yang disebabkan oleh peratas [1], [4]. Sehingga administrator jaringan dapat melakukan langkah pencegahan dan perbaikan yang sesuai dengan keadaan pada jaringan komputer [1]. Dalam penerapannya, IDS dibagi menjadi dua jenis, yaitu NIDS (*Network-Based Intrusion Detection System*) dan HIDS (*Host-Based Intrusion Detection System*) [17].

### *2.2.2. Gradient Boosting Machine*

*Gradient Boosting Machine* atau GBM adalah salah satu teknik *ensemble learning* yang bekerja dengan menggabungkan beberapa model lemah (*weak learners*) untuk membentuk sebuah model yang lebih kuat melalui optimasi pada *loss function* dengan memakai prinsip *gradient descent* [12]. Dalam prosesnya, GBM yang termasuk ke dalam algoritma boosting ini akan bekerja secara iteratif untuk memperbaiki kesalahan prediksi secara bertahap dengan memberikan bobot lebih pada data yang sulit diklasifikasikan dan menurunkannya pada data yang telah diklasifikasikan dengan benar [18]. Pendekatan seperti ini memungkinkan model untuk bisa menyesuaikan distribusi data training di setiap iterasi, sehingga model lebih efektif dalam menangani kelas yang tidak seimbang serta meningkatkan identifikasi terhadap kelas minoritas, yang akhirnya akan menghasilkan model dengan tingkat akurasi lebih tinggi [19], [20]. Algoritma ini memiliki keunggulan signifikan dibanding algoritma lain seperti *Random Forest* dan *Logistic Regression*, dimana akurasi dapat meningkah hampir 80%, dan unggul dalam kecepatan prediksi serta dapat mengatasi *outliers* [12], [13].




Gambar 1. Alur Kerja GBM

Proses kerja algoritma *Gradient Boosting Machine* (GBM) secara bertahap dan sistematis dapat dilihat pada Gambar 1. Algoritma ini dimulai dengan memasukkan data latih, yaitu data jaringan yang memuat fitur-fitur (seperti protokol, lama koneksi, ukuran paket) dan label asli berupa kelas normal atau serangan. Tahapan berikutnya adalah melatih model awal, yaitu model dasar yang digunakan untuk memberikan klasifikasi awal terhadap data dan dinyatakan sebagai $F_0(x)$. Model ini biasanya tidak dilatih dari pohon keputusan, tetapi menggunakan nilai konstan seperti rata-rata target (untuk regresi) atau *log-odds* (untuk klasifikasi biner) [21].

Setelah model awal memberikan klasifikasi awal, langkah selanjutnya adalah mengukur seberapa besar kesalahan klasifikasi yang terjadi, menggunakan rumus *loss function*. Dalam kasus klasifikasi biner, *loss function* yang umum digunakan adalah *log loss*, yang mengukur jarak antara nilai klasifikasi yang dihasilkan model dengan label sebenarnya. Jika $y_i$ adalah label asli (0 atau 1) dan $F(x_i)$ adalah nilai klasifikasi yang diberikan oleh model, maka kesalahan atau *loss* dapat dihitung dengan:

$$L\big(y_i, F(x_i)\big) = -\big[y_i \log\big(F(x_i)\big) + (1 - y_i) \log\big(1 - F(x_i)\big)\big] \qquad (1)$$

*Loss* ini dihitung untuk seluruh data, lalu digunakan untuk menentukan apakah klasifikasi sudah cukup akurat. Jika kesalahannya masih tinggi, maka sistem perlu diperbaiki.

Langkah perbaikan dilakukan dengan melatih *weak learner* pertama, yang dalam GBM biasanya berupa pohon keputusan kecil (*decision stump*). Namun, berbeda dengan model awal yang dilatih untuk memetakan fitur ke label asli, *weak learner* ini dilatih untuk memetakan fitur input (x) ke nilai residual, yaitu selisih atau koreksi dari hasil klasifikasi sebelumnya terhadap label yang sebenarnya. Secara teknis, residual ini dihitung dari turunan fungsi loss terhadap klasifikasi model saat ini, yaitu:

$$r_{mi} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right] \qquad (2)$$

Dengan kata lain, *weak learner* tidak lagi belajar dari label asli seperti 0 atau 1, melainkan dari seberapa besar kesalahan model sebelumnya, sehingga bisa memperbaiki klasifikasi ke arah yang lebih tepat [21]. Misalnya, jika model awal memberi klasifikasi 0.2 untuk data yang seharusnya bernilai 1, maka residualnya besar (sekitar +0.8), dan weak learner dilatih untuk menghasilkan koreksi sebesar itu pada data tersebut. Label targetnya bukan 1 lagi, tapi nilai residual tersebut.

Setelah *weak learner* selesai dilatih, hasil klasifikasinya akan ditambahkan ke model sebelumnya. Proses penggabungan ini mengikuti prinsip *gradient descent*, yaitu model diperbarui sedikit demi sedikit ke arah yang bisa menurunkan kesalahan klasifikasi [29]. Pembaruan model dilakukan dengan rumus:

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x) \qquad (3)$$

Dimana $F_m(x)$ adalah model pada iterasi ke-$m$. Sedangkan $F_{m-1}(x)$ adalah model atau *weak learner* sebelumnya. Untuk $h_m(x)$ adalah *weak learner* atau hasil klasifikasi setelah perbaikan. Dan $\rho_m$ adalah *learning rate* untuk mengatur seberapa besar pengaruh *weak learner* terhadap model total.

Rumus ini disebut prinsip *gradient descent* karena pada dasarnya, kita bergerak secara bertahap untuk menurunkan nilai *loss function*, dengan cara memperbaiki sedikit demi sedikit (dikontrol oleh $\rho_m$) agar tidak tergeser terlalu jauh. Prinsip ini dijelaskan sebagai proses di mana setiap weak learner melakukan langkah perbaikan terhadap model sebelumnya berdasarkan arah turunan fungsi loss, seperti seseorang yang mencari jalan menuruni lereng gunung dengan mengikuti arah kemiringan [21].

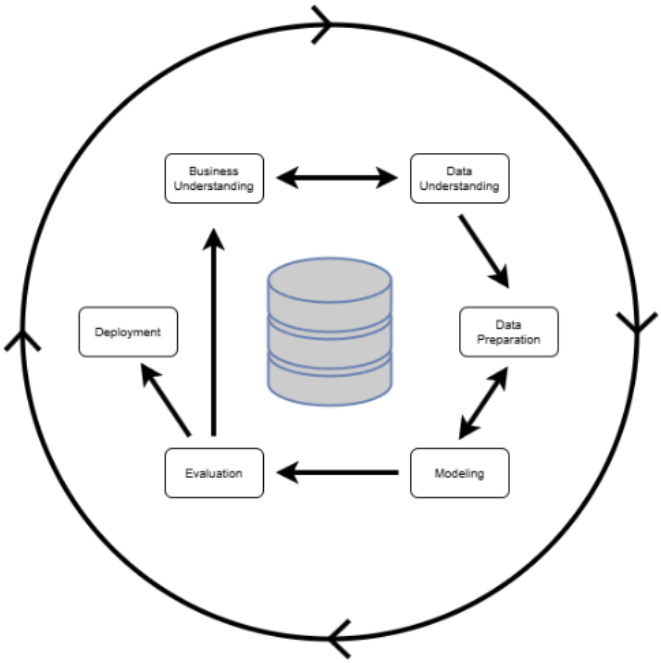
Setelah model diperbarui, kesalahan klasifikasi dihitung kembali. Jika masih belum cukup rendah, proses dilanjutkan ke *weak learner* berikutnya, dan seterusnya secara berulang. Proses ini diulang sampai hasil klasifikasi sudah cukup baik, atau hingga jumlah maksimum iterasi tercapai. Dengan kata lain, model akhir merupakan hasil dari penjumlahan bertahap semua *weak learner* sebelumnya, bukan hanya model awal saja. Sehingga proses GBM dapat diibaratkan seperti tim analis keamanan jaringan dalam sistem deteksi intrusi (IDS). Pada awalnya, seorang analis utama membuat klasifikasi awal apakah suatu trafik jaringan adalah normal atau mencurigakan, berdasarkan informasi dasar. Namun, jika klasifikasi awal ini masih banyak salahnya, maka data tersebut diteruskan ke analis berikutnya, yang bertugas memberikan koreksi berdasarkan kesalahan sebelumnya. Setiap analis baru akan fokus pada bagian yang keliru dari klasifikasi sebelumnya dan memberikan saran perbaikan. Setelah itu, hasil semua analis digabung untuk mendapatkan klasifikasi akhir yang lebih tepat. Proses ini berulang sampai sistem secara keseluruhan mampu membedakan dengan lebih akurat mana trafik normal dan mana yang merupakan potensi serangan. Sehingga, setiap weak learner bisa diibaratkan sebagai analis tambahan yang memberikan kontribusi korektif secara bertahap terhadap keputusan awal.

### 2.2.3. Metrik Evaluasi

Metrik evaluasi adalah metode yang digunakan untuk mengukur kinerja model *Machine Learning* dalam mendeteksi serangan, salah satunya adalah *Confusion Matrix* [22]. Terdapat empat kategori hasil prediksi dalam *confusion matrix*, yaitu TP (*True Positive*), TN (*True Negative*), FP (*False positive*) dan FN (*False Negative*) [13]. Dimana TP dan TN menunjukkan hasil prediksi yang benar sesuai dengan nilai aktualnya pada data, sedangkan FP dan FN menunjukkan kesalahan prediksi [16].

## 3. METODE PENELITIAN

Penelitian ini menggunakan metode *CRISP-DM* (*Cross Industry Standard Process for Data Mining*) seperti pada Gambar 2. *CRISP-DM* merupakan sebuah kerangka kerja standar yang awalnya dikembangkan oleh *Daimler-Chrysler*, SPSS, dan NCR yang terdiri dari enam tahap utama, yaitu *Business Understanding* (pemahaman terhadap masalah), Data Understanding (pengumpulan dan eksplorasi data), Data *Preparation* (persiapan data), *Modeling* (pembangunan model), *Evaluation* (evaluasi performa model), dan *Deployment* [23]. Pada tahap awal atau *Business Understanding*, penelitian difokuskan pada identifikasi masalah keamanan siber dan solusi deteksi intrusi yang diusulkan, hingga batasan-batasan dalam penelitian ini [24]. Selanjutnya, pada tahap Data *Understanding*, data penelitian dikumpulkan dan dianalisis secara mendalam, seperti memastikan jumlah *instances*, jenis atribut, serta karakteristik dataset untuk memastikan kesesuaiannya dengan tujuan penelitian [23], [24].




Gambar 2. *CRISP-DM*

Setelah data dipahami secara komprehensif, tahap Data *Preparation* dilakukan untuk mempersiapkan data sebelum pemodelan, seperti membersihkan data dari missing values atau nilai kosong, mentransformasi format data, serta membagi data menjadi *subset training* dan *testing* [23], [24]. Tahap selanjutnya adalah *Modeling*, yang melibatkan pembangunan model deteksi intrusi menggunakan algoritma GBM berdasarkan data *training* yang telah disiapkan. Model ini kemudian dievaluasi pada tahap *Evaluation* dengan mengukur performanya menggunakan metrik akurasi, presisi, recall, dan *F1-Score* untuk memastikan keandalannya dalam mendeteksi intrusi. Tahap terakhir adalah *Deployment*, namun karena keterbatasan yang ada, maka penelitian ini hanya mencakup 5 tahap dari *Business Understanding* hingga tahap *Evaluation* saja.

## 4. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini telah dipaparkan dan dianalisis sesuai dengan tahapan dalam metode *CRISP-DM*, yang mencakup *Business Understanding*, Data Understanding dan Data *Preparation* serta *Modeling* dan *Evaluation*.

### 4.1. *Business Understanding*

Pada tahap *Business Understanding*, penelitian difokuskan untuk mengidentifikasi permasalahan mendasar terkait urgensi deteksi intrusi dalam jaringan komputer. Hal ini disebabkan oleh semakin meningkatnya jumlah serta kompleksitas serangan siber seiring perkembangan teknologi, khususnya pada sektor pemerintahan dan bisnis, sehingga diperlukan solusi yang mampu mendeteksi serangan secara lebih cepat dan akurat. Metode deteksi konvensional pada IDS terbukti memiliki keterbatasan, terutama dalam menangani volume data yang sangat besar dan dalam mengidentifikasi serangan baru seperti zero-day [5]. Sementara itu, pendekatan berbasis *Machine Learning* sudah banyak diterapkan, namun masih menghadapi kendala seperti tingginya tingkat *false positive*, risiko *overfitting*, serta kebutuhan sumber daya komputasi yang signifikan dalam proses pelatihan model [9], [10]. Oleh karena itu, penelitian ini mengusulkan pemanfaatan algoritma *Gradient Boosting Machine* (GBM) sebagai salah satu pendekatan *Machine Learning* yang dinilai lebih andal dan akurat dalam membangun model deteksi intrusi. Model ini akan dilatih dengan menggunakan dataset *CICIDS2017*, kemudian dievaluasi dan dibandingkan kinerjanya dengan model *Machine Learning* dari penelitian terdahulu dalam konteks deteksi intrusi.

## 4.2. *Data Understanding & Data Preparation*

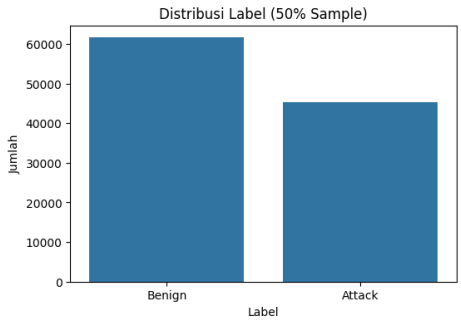
Tahap berikutnya adalah *Data Understanding*, yang bertujuan untuk mengenali serta memahami karakteristik dari dataset *CICIDS2017* yang digunakan dalam penelitian ini. Dataset *CICIDS2017* dikembangkan oleh *Canadian Institute for Cybersecurity* dan mencakup lalu lintas jaringan normal maupun berbagai jenis serangan, seperti *Brute Force, Heartbleed, Botnet, DoS, DDoS, Web Attack* , dan *Infiltration* [11]. Dataset ini terdiri atas delapan *file csv* yang masing-masing dikumpulkan pada waktu berbeda, sehingga menghasilkan variasi kondisi lalu lintas jaringan. Selanjutnya, pada tahap *Data Preparation*, kedelapan file tersebut digabungkan menjadi satu *file csv* dengan ukuran akhir sebesar 80 *megabyte*, yang memuat 79 kolom serta 2.830.743 *instances*. Gambaran sebagian isi dataset dapat dilihat pada Gambar 3.

| | Destination Port | Flow Duration | Total Fwd Packets | Total Backward Packets | Total Length of Fwd Packets | Total Length of Bwd Packets | Fwd Packet Length Max | Fwd Packet Length Min | Fwd Packet Length Mean | Fwd Packet Length Std | ... | min_seg_size_forward | Active Mean | Active Std | Active Max | Active Min | Idle Mean | Idle Std | Idle Max | Idle Min | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 22 | 1266342 | 41 | 44 | 2664 | 6954 | 456 | 0 | 64.975610 | 109.864573 | ... | 32 | 0.0 | 0.0 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | BENIGN |
| 1 | 22 | 1319353 | 41 | 44 | 2664 | 6954 | 456 | 0 | 64.975610 | 109.864573 | ... | 32 | 0.0 | 0.0 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | BENIGN |
| 2 | 22 | 160 | 1 | 1 | 0 | 0 | 0 | 0 | 0.000000 | 0.000000 | ... | 32 | 0.0 | 0.0 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | BENIGN |
| 3 | 22 | 1303488 | 41 | 42 | 2728 | 6634 | 456 | 0 | 66.536585 | 110.129945 | ... | 32 | 0.0 | 0.0 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | BENIGN |
| 4 | 35396 | 77 | 1 | 2 | 0 | 0 | 0 | 0 | 0.000000 | 0.000000 | ... | 32 | 0.0 | 0.0 | 0 | 0 | 0.0 | 0.0 | 0 | 0 | BENIGN |

5 rows × 79 columns

Gambar 3. Dataset

Selanjutnya, dengan memanfaatkan file dataset yang telah digabungkan, dilakukan proses *Data Cleaning* yang merupakan bagian dari tahap Data *Preparation*. Pada tahap ini, *instances* yang terdeteksi sebagai duplikat serta *instances* dengan nilai *null* atau *NaN* akan dihapus. Kemudian, kolom yang dianggap kurang relevan seperti *Flow ID, Source IP, Destination IP*, dan *Timestamp* juga dihilangkan karena tidak memberikan kontribusi signifikan terhadap proses pembelajaran model. Langkah berikutnya dalam tahap *Data Preparation* adalah pemberian label, yaitu 0 untuk *instances* yang tergolong *Benign* (bukan serangan), dan label 1 untuk *instances* yang dikategorikan sebagai serangan, baik berupa DoS, *Botnet*, maupun jenis serangan lainnya.




Gambar 4. Distribusi Data

Distribusi dari kedua label data ditampilkan pada Gambar 4. Terlihat bahwa jumlah data berlabel *Benign* lebih dominan dibandingkan data berlabel serangan, yang menandakan bahwa dataset ini termasuk dalam kategori tidak seimbang. Suatu data disebut tidak seimbang apabila salah satu kelas memiliki jumlah *instance* yang jauh lebih besar dibandingkan kelas lainnya, sedangkan data seimbang ditandai dengan distribusi yang relatif merata antar kelas. Meskipun demikian, pada dataset ini perbedaan antara kelas mayoritas dan minoritas tidak terlalu signifikan, sehingga memberikan peluang yang tepat untuk menguji secara langsung kinerja GBM dalam menangani kondisi data tidak seimbang tanpa perlu menerapkan metode penyesuaian khusus seperti *oversampling*. Selain itu, pendekatan *boosting* seperti GBM dianggap mampu menyesuaikan proses pembelajaran terhadap distribusi data pada setiap iterasi, sehingga lebih efektif dalam mengenali kelas minoritas sekaligus menjaga keseimbangan performa klasifikasi [17], [20].

Pada tahap berikutnya, kolom dengan tipe data kategorikal terlebih dahulu dilakukan *encoding* agar nilainya dapat diubah menjadi numerik sehingga bisa diproses oleh algoritma GBM. Sebelum melangkah lebih jauh,

hanya 50% dari keseluruhan data yang digunakan dalam proses pelatihan model. Hal ini dilakukan karena keterbatasan sumber daya komputasi sekaligus untuk menjaga agar model tetap berjalan secara optimal. Selanjutnya, data yang telah dipilih akan melalui proses normalisasi menggunakan *StandardScaler* dengan tujuan menyetarakan skala pada setiap kolom, sehingga seluruh fitur memiliki pengaruh yang seimbang terhadap algoritma dan tidak ada satu fitur pun yang mendominasi secara tidak proporsional. Tahap terakhir dalam *Data Preparation* adalah membagi data menjadi dua bagian, yaitu 80% untuk proses pelatihan dan 20% sisanya untuk pengujian.

### 4.3. *Modelling & Evaluation*

Setelah proses *Data Preparation* selesai, tahap selanjutnya adalah *Modeling*. Pada tahap ini, algoritma *Gradient Boosting Machine* (GBM) diterapkan untuk membangun model deteksi intrusi. Prinsip kerja GBM adalah dengan menggabungkan sejumlah model lemah menjadi satu model yang lebih kuat melalui proses perbaikan kesalahan prediksi secara iteratif dan bertahap dari model sebelumnya [13][18]. *Pseudocode* dari proses kerja algoritma GBM yang digunakan dalam penelitian ini dapat dilihat pada Kode Program 1.

Kode Program 1. *Pseudocode*

```
# 1. Load dataset
data = load_csv("CICIDS2017_combined.csv")

# 2. Data cleaning
data = remove_duplicates(data)
data = drop_null_values(data)
data = drop_columns(data, ['Flow ID', 'Source IP', 'Destination IP', 'Timestamp'])

# 3. Labeling & encoding
data['Label'] = data['Attack Type'].apply(lambda x: 0 if x == 'Benign' else 1)
data = encode_categorical_columns(data)

# 4. Feature scaling
data_scaled = StandardScaler().fit_transform(data.drop('Label', axis=1))

# 5. Train-test split
X_train, X_test, y_train, y_test = train_test_split(data_scaled, data['Label'],
test_size=0.2)

# 6. Train GBM model
model = GradientBoostingClassifier().fit(X_train, y_train)

# 7. Evaluation
y_pred = model.predict(X_test)
evaluate_model(y_test, y_pred)
```

Pada percobaan awal, GBM dijalankan menggunakan parameter bawaan, yaitu *learning_rate* = 0.1, *n_estimators* = 100, serta *max_depth* = 3, dan hasilnya sudah memberikan performa yang sangat tinggi dengan nilai *F1-Score* mencapai 99,98%. Sebagai langkah validasi tambahan, dilakukan penyesuaian parameter melalui *Grid Search* untuk memastikan konfigurasi yang digunakan benar-benar optimal. *Grid Search* merupakan metode pencarian *Hyperparameter* terbaik dengan cara menguji semua kemungkinan kombinasi yang tersedia. Dalam penelitian ini, terdapat tiga *Hyperparameter* yang dioptimalkan, yaitu *n_estimators* (jumlah total pohon dalam *ensemble*), *learning_rate* (tingkat kontribusi dari setiap pohon baru), dan *max_depth* (kedalaman maksimum pada masing-masing pohon keputusan). Seluruh kombinasi *Hyperparameter* tersebut dievaluasi menggunakan teknik *K-Fold Cross Validation* dengan nilai K = 2, yang berarti data pelatihan dibagi menjadi dua bagian secara bergantian untuk proses pelatihan dan validasi. Dengan pengaturan ini, total 8 kombinasi parameter diuji dan divalidasi sebanyak 16 kali untuk memperoleh hasil terbaik. Adapun Gambar 5 menampilkan potongan kode *Python* menggunakan *scikit-learn* untuk implementasi *Grid Search* dalam menentukan kombinasi *Hyperparameter* optimal pada model *Gradient Boosting Machine*.

```
param_grid = {
    'n_estimators': [100, 150],
    'learning_rate': [0.05, 0.1],
    'max_depth': [3, 5]
}

model = GradientBoostingClassifier(random_state=42)
grid_search = GridSearchCV(model, param_grid, scoring='f1', cv=2, n_jobs=-1, verbose=1)
grid_search.fit(X_train, y_train)

best_model = grid_search.best_estimator_
```

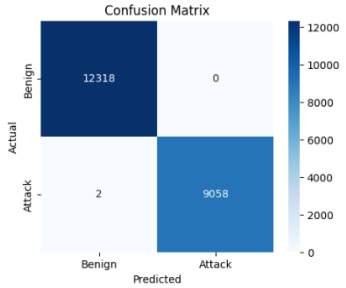
Gambar 5. Kode *Grid Search*

Karena dataset *CICIDS2017* memiliki distribusi yang tidak seimbang, sebagaimana terlihat pada Gambar 4, maka metrik evaluasi utama yang digunakan untuk menentukan *Hyperparameter* terbaik adalah *F1-Score* . Hal ini dikarenakan *F1-Score* merupakan gabungan dari nilai precision (kemampuan model untuk menghindari *false positive*) dan recall (kemampuan model dalam mendeteksi seluruh serangan atau mengurangi false negative) menjadi satu ukuran. Penggunaan metrik ini sangat penting agar performa model tidak hanya terlihat baik karena mampu menebak kelas mayoritas, tetapi juga benar-benar efektif dalam mengenali serangan. Hasil dari *Grid Search* menunjukkan bahwa kombinasi *Hyperparameter* terbaik adalah *learning_rate* = 0.1, *max_depth* = 5, dan n_estimators = 150. Konfigurasi ini menghasilkan nilai *F1-Score* sebesar 99,98%, yang membuktikan kemampuan model dalam mendeteksi serangan dengan performa yang sangat optimal.

**Tabel 1.** Hasil Evaluasi Model

| **Hasil Evaluasi** | | |
|---|---|---|
| **Akurasi** | : | 99.99% |
| **Presisi** | : | 100% |
| ***Recall*** | : | 99.98% |
| ***F1-Score*** | : | 99.99% |

Tahap terakhir adalah *Evaluation*, yaitu pengujian model menggunakan data uji sebesar 20% yang telah dipisahkan sejak awal. Hasil evaluasi yang ditampilkan pada Tabel 1 memperlihatkan bahwa model memiliki kinerja yang sangat baik. Nilai accuracy mencapai 99,99%, yang berarti hampir seluruh prediksi yang dihasilkan model sesuai, baik dalam mengidentifikasi serangan maupun lalu lintas normal. Sementara itu, nilai precision sebesar 100% menunjukkan bahwa setiap data yang diprediksi sebagai serangan benar-benar merupakan serangan, sehingga tidak terdapat *false positive*. Dengan demikian, sistem mampu mencegah munculnya peringatan palsu secara berulang. Selanjutnya, nilai recall sebesar 99,98% menandakan bahwa hampir semua serangan berhasil terdeteksi oleh model, dengan jumlah *false negative* yang sangat kecil. Adapun nilai *F1-Score* mencapai 99,99%, yang mencerminkan keseimbangan yang sangat tinggi antara recall dan precision. Hal ini menunjukkan bahwa model sangat andal dalam mendeteksi serangan pada konteks intrusion detection, terutama ketika dihadapkan pada kondisi dataset yang tidak sepenuhnya seimbang.




Gambar 6. *Confussion Matrix*

Selain itu, hasil evaluasi juga diperkuat melalui *Confusion Matrix* yang ditampilkan pada Gambar 6, yang memperlihatkan distribusi klasifikasi model terhadap data uji. Dari total 12.318 data *benign* (normal),

seluruhnya berhasil diklasifikasikan dengan tepat tanpa menghasilkan *false positive*. Sementara itu, dari 9.060 data serangan (*attack*), sebanyak 9.058 data berhasil teridentifikasi dengan benar, dan hanya terdapat 2 *instance* yang salah dikategorikan sebagai benign (*false negative* = 2). Hasil ini semakin menegaskan bahwa model tidak hanya akurat dalam mengenali lalu lintas normal, tetapi juga sangat andal dalam mendeteksi serangan nyata dengan tingkat kesalahan yang sangat rendah. *Confusion Matrix* ini konsisten dengan hasil metrik evaluasi sebelumnya dan memperlihatkan bahwa model GBM memiliki tingkat sensitivitas dan spesifisitas yang sangat tinggi dalam penerapan pada sistem deteksi intrusi.

**Tabel 2.** Perbandingan dengan Penelitian Terdahulu

| Algoritma | Akurasi | Presisi | *Recall* | *F1-Score* |
|---|---|---|---|---|
| **GBM** | 99.99% | 100% | 99.98% | 99.99% |
| ***CNN* [5]** | 98.5% | 96.2% | 97.4% | 99.68% |
| ***Random Forest* [8]** | 96.8% | 95.%% | 94.7% | 95.1% |
| ***Decision Tree* [17]** | 99.9% | 99.9% | 99.99% | 99.99% |
| ***Random Forest* [18]** | 99.99% | 99.98% | 100% | - |
| ***Random Forest + PCA* [22]** | 99.2% | >63% | >63% | - |
| ***DNN* [23]** | 100% | - | - | - |

Jika dibandingkan dengan pendekatan lain yang banyak digunakan pada penelitian sebelumnya, sebagaimana ditunjukkan pada Tabel 2, sistem deteksi intrusi sebelum menerapkan algoritma GBM umumnya masih mengandalkan metode seperti *Random Forest*, *Decision Tree*, maupun pendekatan *deep learning* seperti CNN. Walaupun beberapa metode tersebut sempat memberikan tingkat *accuracy* yang cukup tinggi, model berbasis *Random Forest* maupun *Decision Tree* masih sering menghadapi masalah *overfitting* serta keterbatasan dalam mengidentifikasi serangan yang lebih kompleks. Di sisi lain, metode berbasis CNN membutuhkan kapasitas komputasi yang besar dan waktu pelatihan yang relatif lama, sehingga kurang efisien bagi sistem IDS yang membutuhkan respons cepat. Dalam penelitian ini, penerapan GBM mampu memberikan peningkatan performa yang signifikan, dengan nilai *accuracy*, *precision*, dan *recall* yang sangat tinggi serta konsisten, bahkan ketika diuji pada dataset yang tidak seimbang. Nilai *precision* yang sempurna mengindikasikan tidak adanya *false positive*, sementara nilai *recall* yang mendekati sempurna menunjukkan kemampuan model dalam mengenali hampir semua serangan yang ada. Hal ini membuktikan bahwa penggunaan GBM tidak hanya meningkatkan kinerja sistem deteksi intrusi dibandingkan metode sebelumnya, tetapi juga menjadikannya lebih andal, efisien, serta mampu beradaptasi dengan distribusi data nyata yang umumnya tidak seimbang. Dengan demikian, GBM layak dipertimbangkan sebagai pendekatan unggulan dalam membangun IDS modern yang presisi dan stabil. Dan kedepannya model ini dapat terus dikembangkan untuk dievaluasi secara *real-time* pada skenario serangan yang lebih kompleks, sehingga mampu meningkatkan reliabilitas sistem secara menyeluruh.

## V. SIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan algoritma *Gradient Boosting Machine* atau GBM dengan menggunakan metode *CRISP-DM* untuk melakukan deteksi intrusi pada jaringan dengan dataset *CICIDS2017*. Dimana hasil evaluasi menunjukkan nilai yang sangat baik sebesar 99.99% untuk akurasi, 100% untuk presisi, 99.98% untuk *Recall* dan 99.99% untuk *F1-Score* . Terbukti dengan kombinasi *Hyperparameter* terbaik yang didapatkan dengan *Grid Search* sangat membantu memaksimalkan kinerja model dalam melakukan deteksi intrusi, mengurangi nilai *false positive* dan *false negative* serta mejaga generalisasi tanpa mengalami *overfitting*. Sehingga model GBM ini menunjukkan keandalan yang tinggi dan dapat bersaing dengan model lain dari penelitian terdahulu.

Untuk penelitian berikutnya disarankan melakukan implementasi model GBM secara real-time pada IDS yang beroprasi untuk menguji performa model dibawah skenario serangan yang lebih kompleks dan dinamis. Serta dapat melakukan eksplorasi terhadap teknik lainnya seperti *Hybrid*, yang mengkombinasikan beberapa algoritma. Selain itu bisa ditambahkan dengan metode seleksi fitur dan teknik-teknik tambahan untuk lebih menyeimbangkan data sebelum pelatihan, sehingga dapat mengurangi beban komputasi.

## DAFTAR PUSTAKA

[1] A. Khaliq and S. Novida Sari, "Pemanfaatan Kerangka Kerja Investigasi Forensik Jaringan Untuk Identifikasi Serangan Jaringan Menggunakan Sistem Deteksi Intrusi (Ids)," *Jurnal Nasional Teknologi Komputer*, vol. 2, no. 3, pp. 150–158, 2022, doi: 10.61306/jnastek.v2i3.52.

[2] R. Ahmad, "Survei Teknik Pemilihan Fitur Untuk Sistem Deteksi Intrusi Berbasis Machine Learning," *Infotek: Jurnal Informatika dan Teknologi*, vol. 8, no. 1, pp. 317–323, 2025, doi: 10.29408/jit.v8i1.28657.

[3] H. F. Fajar, A. Fadhila, and M. K. Yades, "Redesain Kebijakan Hukum Keamanan dan Ketahanan Siber: Studi Kasus Serangan Siber pada Pusat Data Nasional Tahun 2024," *Rechts Vinding: Media Pembinaan Hukum Nasional*, vol. 14, no. 2, 2025. doi: 10.33331/rechtsvinding.v14i2.2155.

[4] C. L. Mindara, A. Zulianto, H. P. Utomo, T. Hatati, and G. P. Mindara, "Deteksi Intrusi Untuk Klasifikasi Serangan Jaringan Dengan Penerapan Algoritma Convolutional Neural Network," *Jurnal ICT : Information Communication & Technology*, vol. 23, pp. 517–522, 2023.

[5] R. P. Simanjuntak and R. R. M. Sijabat, "Meningkatkan Keamanan Siber dalam Lingkungan Internet of Things (IoT) dengan Menggunakan Sistem Deteksi Intrusi Berbasis Pembelajaran Mesin," *DIKE : Jurnal Ilmu Multidisiplin*, vol. 2, no. 2, pp. 62–68, 2024, doi: 10.69688/dike.v2i2.106.

[6] B. Ramadhan, D. Firdaus, and A. R. Rafi, "MIND (Multimedia Artificial Intelligent Networking Database Teknik SMOTE Sebagai Solusi Imbalance Class dalam Model Deteksi Intrusi DDoS dengan Metode PCA-Random Forest," *Journal MIND Journal | ISSN*, vol. 8, no. 1, pp. 52–64, 2023, doi: 10.26760/mindjournal.v8i1.52-64.

[7] F. Wibowo, "Analisis Algoritma Kecerdasan Buatan pada Pengembangan Sistem Deteksi Intrusi pada Jaringan Komputer," *Duniailmu.org*, vol. 3, no.9, 2023.

[8] A. Sulandri, A. Basuki, and F. A. Bachtiar, "Metode Deteksi Intrusi Menggunakan Algoritma Extreme Learning Machine dengan Correlation-Based Feature Selection," *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, vol. 8, no. 1.

[9] D. P. Sari, Z. Halim, I. Irlon, B. Waseso, and S. Saromah, "Implementasi Machine Learning untuk Deteksi Intrusi pada Jaringan Komputer," *Jurnal Minfo Polgan*, vol. 13, no. 2, pp. 1389–1394, 2024, doi: 10.33395/jmp.v13i2.14074.

[10] N. F. Wardhani, "Penggunaan Machine Learning dalam Deteksi Intrusi pada Jaringan Komputer," *Duniadata*, vol. 1, no. 4, 2024.

[11] J. Dewanto, "Analisis Perbandingan Dataset KDD'99, UNSW-Nb15, dan CICIDS2017 untuk Sistem Deteksi Intrusi," *Kohesi: Jurnal Multidisiplin Saintek*, vol. 2, no. 9, 2024.

[12] R. Rianti and R. Andarsyah, "Memprediksi Tingkat Atrisi Karyawan Menggunakan Machine Learning," *Jurnal Tekno Insentif*, vol. 18, no. 1, pp. 39–52, 2024, doi: 10.36787/jti.v18i1.1263.

[13] M. I. Rizaldi, D. R. Chandranegara, and D. R. Akbi, "Comparison of Machine Learning Techniques for Classification of Distributed Denial of Service Attacks based on Feature Engineering in SDN-Based Networks," *JIPI Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*, vol. 9, no. 3, 2024.

[14] I. Wardhana, M. Ariawijaya, and V. A. Isnaini, "Gradient Boosting Machine, Random Forest dan Light GBM untuk Klasifikasi Kacang Kering," *JURNAL RESTI*, vol. 6, no. 1, 2022.

[15] A. Oyelakin, “Overview and Exploratory Analyses of CICIDS 2017 Intrusion Detection Dataset,” *J Syst Eng Inf Technol*, vol. 2, no. 2, pp. 45–52, 2023.

[16] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, “A Case Study with CICIDS2017 on the Robustness of Machine Learning against Adversarial Attacks in Intrusion Detection,” *ACM International Conference Proceeding Series*, 2023, doi: 10.1145/3600160.3605031.

[17] P. A. Khairunnisa, N. Annisa, and J. Parhusip, “Perancangan Sistem Keamanan Jaringan Berbasis Cybersecurity untuk Mitigasi Ancaman Siber pada Infrastruktur TI: Studi Kasus di Indonesia,” *Jurnal Ilmu Teknik dan Informatika*, vol. 4, pp. 9–16, 2024.

[18] F. Septian, “Optimasi Klusterisasi pada Lama Tempo Pekerjaan Berbasis *Gradient Boost Algorithm*,” *Indonesian Journal of Information Technology*, vol. 20, no. 20, 2023.

[19] A. R. Arrahimi, M. K. Ihsan, D. Kartini, et al., “Teknik *Bagging* dan *Boosting* pada Algoritma *CART* untuk Klasifikasi Masa Studi Mahasiswa,” *Jurnal Sains dan Informatika*, vol. 5, no. 1, Jun. 2019.

[20] N. H. Setyawan and N. Wakhidah, “Analisis Perbandingan Metode *Logistic Regression*, *Random Forest*, *Gradient Boosting* untuk Prediksi Diabetes,” *JIPI* (*Jurnal Ilmiah Penelitian dan Pembelajaran Informatika*), vol. 10, no. 1, Mar. 2025, doi: https://doi.org/10.29100/jipi.v10i1.5743.

[21] L. W. Rizkallah, "*Enhancing the performance of gradient boosting trees on regression problems*," *Journal of Big Data*, vol. 12, no. 35, 2025, doi: https://doi.org/10.1186/s40537-025-01071-3.

[22] B. Ramadhan, D. Firdaus, and A. R. Rafi, “Teknik SMOTE Sebagai Solusi Imbalance Class dalam Model Deteksi Intrusi DDoS dengan Metode PCA-Random Forest,” *Journal MIND Journal | ISSN*, vol. 8, no. 1, pp. 52–64, 2023.

[23] R. Mubarok, A. A. Syahputra, A. T. Permana, L. Sholiah, and Tarwoto, “Implementasi Data Mining untuk Clustering Lowongan Pekerjaan Menggunakan Metode Algoritma K-Means,” *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 9, no. 2, 2025.

[24] R. R. Aria, “Implementasi Algoritma K-Means untuk Pengelompokan Data Imunisasi Balita dengan Metode CRISP-DM,” *Remik*, vol. 9, no. 1, pp. 189–197, 2025, doi: 10.33395/remik.v9i1.14391