Analisis Explainable Artificial Intelligence Menggunakan SHAP dan LIME untuk Interpretasi Network Intrusion Detection System



Disusun Oleh:

N a m a : Vito Anargya

NIM : 22523117

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

Analisis Explainable Artificial Intelligence Menggunakan SHAP dan LIME untuk Interpretasi Network Intrusion Detection System

TUGAS AKHIR

Disusun Oleh:

N a m a : Vito Anargya

NIM : 22523117

Yogyakarta, 05 Juli 2026

Pembimbing 1,	Pembimbing 2,
(Fayruz Rahma, S.T., M.Eng.)	(Arrie Kurniawardhani, S.Si., M.Kom.)

HALAMAN PENGESAHAN DOSEN PENGUJI

Analisis Explainable Artificial Intelligence Menggunakan SHAP dan LIME untuk Interpretasi Network Intrusion Detection System

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Juli 2026

Tim Penguji

Fayruz Rahma, S.T., M.Eng. _____

Anggota 1

Kholid Haryono, S.T., M.Kom. _____

Anggota 2

Taufiq Hidayat, S.T., M.C.S., Ph.D. _____

Mengetahui,
Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

(Dhoma Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Vito Anargya

NIM : 22523117

Tugas akhir dengan judul:

Analisis Explainable Artificial Intelligence Menggunakan SHAP dan LIME untuk Interpretasi Network Intrusion Detection System

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 05 Juli 2026



(Vito Anargya)

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil'alamin. Dengan rasa syukur kepada Allah atas segala nikmat, rahmat dan kemudahan yang diberikan, skripsi ini saya persembahkan kepada kedua orang tua, keluarga, dosen pembimbing, sahabat, teman-teman, serta semua pihak yang telah memberikan doa, dukungan, dan semangat selama proses penyusunan skripsi. Semoga karya sederhana ini dapat memberikan manfaat dan menjadi salah satu bentuk ikhtiar dalam menuntut ilmu.

HALAMAN MOTO

"Kun fayakūn."

"*Sesungguhnya urusan-Nya apabila Dia mengendaki sesuatu, Dia hanya berkata kepadanya, 'Jadilah!' Maka jadilah sesuatu itu*." (QS. Yasin: 82)

Bagi manusia, ada mimpi yang tampak terlalu tinggi, doa yang terasa terlalu jauh, dan harapan yang mustahil untuk digapai. Namun, bagi Allah tidak ada kata mustahil. Ketika Dia berkehendak, segala yang tidak mungkin menjadi mungkin, segala yang sulit menjadi mudah, segala yang terasa terlambat akan datang pada waktu terbaik menurut ketetapan-Nya.

"Iman kepada takdir bukan hanya tentang menerima apa yang kita impikan, tetapi juga tentang mengikhlaskan apa yang tidak pernah kita inginkan." (Rukun Iman ke-6)

KATA PENGANTAR

Alhamdulillahirabbil'alamin, segala puji dan syukur penulis panjatkan ke hadirat Allah atas segala rahmat, nikmat, serta hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Analisis Explainable Artificial Intelligence Menggunakan SHAP dan LIME untuk Interpretasi Network Intrusion Detection System" sebagai salah satu syarat untuk memperoleh gelar Sarjana Informatika.

Penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, doa, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Allah atas segala rahmat, kesehatan, dan kemudahan yang diberikan selama proses penyusunan skripsi.
2. Kedua orang tua dan keluarga tercinta atas doa, kasih sayang, dukungan, serta motivasi yang tidak pernah terputus.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika Program Sarjana Universitas Islam Indonesia.
5. Ibu Fayruz Rahma, S.T., M.Eng. dan Ibu Arrie Kurniawardhani, S.Si., M.Eng. selaku Dosen Pembimbing yang telah memberikan pengarahan, bimbingan, bantuan, serta masukan kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.
6. Sahabat, teman-teman, dan semua pihak yang telah memberikan dukungan serta bantuan, baik secara langsung maupun tidak langsung.

Tentunya penulis menyadari bahwa skripsi ini masih memiliki keterbatasan dan jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran demi penyempurnaan karya ini di masa mendatang. Akhir kata, penulis berharap semoga skripsi ini dapat memberikan manfaat bagi pembaca serta menjadi salah satu referensi dalam pengembangan dan penelitian di bidang deteksi serangan jaringan dan Explainable Artificial Intelligence (XAI).

Bekasi, 05 Juli 2026



(Vito Anargya)

SARI

Perkembangan teknologi komputer yang semakin pesat diikuti dengan meningkatnya ancaman keamanan siber, sehingga diperlukan sistem yang mampu mendeteksi berbagai jenis serangan jaringan secara akurat. *Network Intrusion Detection System* (NIDS) berbasis *machine learning* telah banyak digunakan untuk meningkatkan kemampuan deteksi serangan. Namun, sebagian besar model *machine learning* memiliki karakteristik *black box* yang menyebabkan proses pengambilan keputusan sulit dipahami. Penelitian ini bertujuan menganalisis interpretasi model *machine learning* pada *Intrusion Detection System* menggunakan metode *Explainable Artificial Intelligence* (XAI) berbasis Shapley Additive exPlanations. Dataset yang digunakan adalah CIC-IDS2017 yang terdiri atas enam kelas, yaitu Benign, Bot, Brute Force, Denial of Service (DoS), PortScan, dan WebAttack. Penelitian ini menggunakan tiga model klasifikasi, yaitu Random Forest, eXtreme Gradient Boosting (XGBoost), dan Multi-Layer Perceptron (MLP). Evaluasi model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan F1-*score*, sedangkan interpretasi model dilakukan melalui analisis global dan lokal menggunakan SHAP. Selain itu, metode LIME digunakan sebagai pembanding pada interpretasi lokal. Hasil penelitian menunjukkan bahwa model XGBoost memperoleh performa terbaik dengan nilai *accuracy, precision, recall*, dan F1-*score* masing-masing sebesar 99,65%. Analisis SHAP menunjukkan bahwa fitur seperti Destination Port, Init_Win_bytes_backward, dan Bwd Packets/s memberikan kontribusi penting terhadap proses klasifikasi. Perbandingan interpretasi lokal menggunakan SHAP dan LIME menunjukkan bahwa kedua metode menghasilkan interpretasi yang konsisten terhadap fitur-fitur utama, akan tetapi SHAP memberikan informasi yang lebih komprehensif karena mampu menjelaskan model pada tingkat global maupun lokal. Hasil penelitian menunjukkan bahwa penerapan SHAP dapat meningkatkan transparansi serta pemahaman terhadap proses pengambilan keputusan model *Network Intrusion Detection System* berbasis *machine learning*.

Kata kunci: *Explainable Artificial Intelligence*, *Intrusion Detection System*, *Machine Learning*, SHAP, XGBoost.

GLOSARIUM

Balanced Dataset	Dataset yang memiliki jumlah data yang seimbang pada setiap kelas untuk mengurangi bias selama proses pelatihan model.
Bot	Jenis serangan jaringan yang memanfaatkan perangkat yang telah terinfeksi *malware* untuk melakukan aktivitas otomatis tanpa sepengetahuan pengguna
BruteForce	Metode serangan yang mencoba berbagai kombinasi kredensial secara berulang hingga memperoleh akses yang benar
CIC-IDS2017	Dataset deteksi intrusi yang dikembangkan oleh Canadian Institute for Cybersecurity dan digunakan sebagai dataset pada penelitian ini.
Destination Port	Nomor *port* tujuan pada komunikasi jaringan yang menjadi salah satu fitur penting dalam proses klasifikasi serangan jaringan.
DoS	*Denial of Service* merupakan tipe serangan yang bertujuan mengganggu ketersediaan layanan dengan membanjiri sistem menggunakan lalu lintas jaringan.
IDS	*Intrusion Detection System* digunakan untuk mendeteksi aktivitas mencurigakan atau serangan pada jaringan komputer.
LIME	*Local Interpretable Model-Agnostic Explanations* merupakan salah satu metode XAI yang memberikan penjelasan lokal terhadap hasil prediksi model menggunakan pendekatan *local surrogate model*.
MLP	*Multi Layer Perceptron* adalah algoritma jaringan saraf tiruan yang digunakan sebagai salah satu model klasifikasi pada penelitian ini.
PortScan	Serangan yang dilakukan dengan memindai *port* pada suatu sistem untuk menemukan layanan yang terbuka.
Random Forest	Algoritma *machine learning* berbasis kumpulan pohon keputusan yang menggunakan teknik *bagging*.
SHAP	Metode XAI yang menghitung kontribusi setiap fitur terhadap hasil prediksi menggunakan konsep nilai Shapley.
XGBoost	Algoritma *machine learning* berbasis *gradient boosting* yang digunakan sebagai model *machine learning* pada penelitian ini.
XAI	*Explainable Artificial Intelligence* adalah pendekatan kecerdasan buatan yang bertujuan menjelaskan proses pengambilan keputusan model sehingga hasil prediksi lebih mudah dipahami.

DAFTAR ISI

HALAMAN JUDUL i
HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI viii
GLOSARIUM ix
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Lingkup Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 3
1.6 Sistematika Penulisan 4
BAB II LANDASAN TEORI 5
2.1 *Intrusion Detection System* 5
2.1.1 *Network Intrusion Detection System* (NIDS) 6
2.2 Dataset CIC-IDS2017 7
2.3 *Machine Learning* 11
2.3.1 Random Forest 12
2.3.2 eXtreme Gradient Boosting (XGBoost) 13
2.3.3 Multi-Layer Perceptron (MLP) 14
2.4 Explainable Artificial Intelligence (XAI) 15
2.4.1 Shapley Additive exPlanations (SHAP) 16
2.4.2 Local Interpretable Model-agnostic Explanations (LIME) 18
2.5 Evaluasi Model 19
2.6 Penelitian Terkait 21
BAB III METODOLOGI PENELITIAN 25
3.1 Alur Pengerjaan Tugas Akhir 25
3.2 Uraian Pengerjaan Tugas Akhir 26
3.2.1 Kajian Pustaka 26
3.2.2 Perencanaan Eksperimen 26
3.2.3 Eksperimen 28
3.2.4 Analisis Hasil Eksperimen 33
3.2.5 Dokumentasi Eksperimen dan Penyusunan Laporan 34
BAB IV HASIL DAN PEMBAHASAN 35
4.1 Data Preparation 35
4.1.1 Cleaning Data 35
4.1.2 Penghapusan Kelas dan Label Mapping 37
4.1.3 Balancing Dataset 38
4.1.4 Persiapan Data sebelum Pelatihan Model 39
4.2 Pelatihan Model 41

4.2.1 Pelatihan Model Random Forest 42
4.2.2 Pelatihan Model XGBoost 42
4.2.3 Pelatihan Model Multi-Layer Perceptron 42
4.3 Evaluasi Model 43
4.3.1 Evaluasi Model Random Forest 43
4.3.2 Evaluasi Model XGBoost 45
4.3.3 Evaluasi Model Multi-Layer Perceptron 46
4.3.4 Perbandingan Performa Model 47
4.3.5 Pengujian Ketahanan Model pada Sisa Data 49
4.4 Interpretasi Model Menggunakan SHAP 50
4.4.1 Interpretasi Global Model Random Forest 50
4.4.2 Interpretasi Global Model XGBoost 52
4.4.3 Interpretasi Global Model Multi-Layer Perceptron 54
4.4.4 Perbandingan Interpretasi Global Antar Model 56
4.4.5 Interpretasi Lokal Menggunakan SHAP 58
4.5 Interpretasi Model Menggunakan LIME 63
4.5.1 Interpretasi Lokal Kelas DoS Menggunakan LIME 63
4.5.2 Interpretasi Lokal Kelas PortScan Menggunakan LIME 65
4.5.3 Interpretasi Lokal Kelas Bot Menggunakan LIME 67
4.5.4 Perbandingan Hasil Interpretasi SHAP dan LIME 69
4.6 Diskusi Hasil 71
4.6.1 Diskusi Performa Model 71
4.6.2 Diskusi Hasil Interpretasi SHAP dan LIME 73
BAB V KESIMPULAN DAN SARAN 75
5.1 Kesimpulan 75
5.2 Saran 76
DAFTAR PUSTAKA 77
LAMPIRAN 81

DAFTAR TABEL

Tabel 2.1 Karakteristik Dataset CIC-IDS2017 (Sharafaldin et al., 2018) 8
Tabel 2.2 Kelompok Fitur pada Dataset CIC-IDS2017 (Sharafaldin et al., 2018) 9
Tabel 2.3 Distribusi Kelas pada Dataset CIC-IDS2017 10
Tabel 2.4 Kelas yang Digunakan pada Penelitian 10
Tabel 2.5 Ilustrasi Penentuan TP, FP, TN, dan FN pada *Multi-Class Confusion Matrix* 19
Tabel 2.6 Penelitian Terdahulu 21
Tabel 3.1 Spesifikasi Perangkat Keras 28
Tabel 3.2 Lingkungan Eksperimen 28
Tabel 3.3 Karakterisik Dataset Setelah Melalui tahap Preprocessing 29
Tabel 3.4 Pemetaan Label Dataset CIC-IDS2017 30
Tabel 4.1 Distribusi Dataset setelah *Balancing* 39
Tabel 4.2 Hasil Validasi Dataset 40
Tabel 4.3 Hasil *Label Encoding* 40
Tabel 4.4 Hasil Evaluasi Model Random Forest 44
Tabel 4.5 Hasil Evaluasi Model XGBoost 45
Tabel 4.6 Hasil Evaluasi Model Multi-Layer Perceptron 47
Tabel 4.7 Perbandingan Performa Model 48
Tabel 4.8 Hasil Pengujian Model XGBoost pada Sisa Data (*Unseen Data*) 49
Tabel 4.9 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model Random Forest 51
Tabel 4.10 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model XGBoost 53
Tabel 4.11 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model MLP 55
Tabel 4.12 Perbandingan Lima Fitur dengan Nilai Mean SHAP Tertinggi pada Setiap Model 56
Tabel 4.13 Bobot Fitur Hasil Interpretasi Lokal LIME pada Sampel Kelas DoS 64
Tabel 4.14 Bobot Fitur Hasuk Interpretasi Lokal LIME pada Sampel Kelas Portscan 66
Tabel 4.15 Bobot Fitur Hasil Interpretasi Lokal LIME pada Sampel Kelas Bot 68
Tabel 4.16 Perbandingan Hasil Interpretasi SHAP dan LIME 70

DAFTAR GAMBAR

Gambar 2.1 Klasifikasi *Intrusion Detection System* 6
Gambar 2.2 Transformasi Model *Black-box* menuju *Explainable Artificial Intelligence* 15
Gambar 2.3 Ilustrasi Kontribusi Fitur pada Shapley Additive exPlanations (SHAP) 17
Gambar 2.4 Workflow Local Interpretable Model-agnostic Explanations (LIME) 18
Gambar 3.1 Alur Pengerjaan Tugas Akhir 25
Gambar 3.2 Perencanaan Eksperimen 27
Gambar 4.1 Penggabungan Dataset 36
Gambar 4.2 Pemeriksaan dan Penghapusan Missing Value dan Penghapusan Infinity Value 37
Gambar 4.3 Kode penghapusan kelas Heartbleed dan Infiltration 37
Gambar 4.4 Kode program Label Mapping 38
Gambar 4.5 Implementasi *Balancing* Dataset 39
Gambar 4.6 Persiapan Dataset Sebelum Pelatihan Model 39
Gambar 4.7 Pemisahan Fitur, Label, dan *Label Encoding* 40
Gambar 4.8 Pembagian Data Latih dan Data Uji 41
Gambar 4.9 Implementasi *Feature Scaling* 41
Gambar 4.10 Implementasi Pelatihan Model Random Forest 42
Gambar 4.11 Implementasi Pelatihan Model XGBoost 42
Gambar 4.12 Implementasi Pelatihan Model Multi-Layer Perceptron 43
Gambar 4.13 Implementasi Prediksi Model Random Forest 43
Gambar 4.14 Confusion Matrix Model Random Forest 44
Gambar 4.15 Implementasi Prediksi Model XGBoost 45
Gambar 4.16 *Confusion Matrix* Model XGBoost 45
Gambar 4.17 Implementasi Prediksi Model Multi-Layer Perceptron 46
Gambar 4.18 Confusion Matrix Model Multi-Layer Perceptron 47
Gambar 4.19 Visualisasi Global SHAP Importance Model Random Forest 51
Gambar 4.20 Hasil Global SHAP Importance pada Model XGBoost 53
Gambar 4.21 Hasil Global SHAP Importance pada Model MLP 55
Gambar 4.22 Visualisasi Waterfall SHAP pada Sampel kelas DoS 59
Gambar 4.23 Visualisasi Waterfall SHAP pada Sampel Kelas PortScan 60
Gambar 4.24 Visualisasi Waterfall SHAP pada Sampel Kelas Bot 62
Gambar 4.25 Interpretasi Lokal LIME pada Sampel Kelas DoS 64
Gambar 4.26 Interpretasi Lokal LIME pada Sampel Kelas PortScan 66

Gambar 4.27 Interpretasi Lokal LIME pada Sampel Kelas Bot 68

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah meningkatkan penggunaan jaringan komputer dalam berbagai bidang, seperti komunikasi, layanan publik, transaksi keuangan, dan operasional organisasi. Meningkatnya aktivitas jaringan juga diiringi dengan peningkatan ancaman keamanan siber yang dapat mengganggu kerahasiaan, integritas, dan ketersediaan informasi. Berbagai jenis serangan, seperti *Denial of service* (DoS), *Port Scanning*, *BruteForce*, dan *Botnet*, dapat menyebabkan kerugian baik dari sisi operasional maupun finansial. Oleh karena itu, diperlukan mekanisme keamanan yang mampu mendeteksi aktivitas mencurigakan pada lalu lintas jaringan secara efektif (Gu et al., 2024).

Salah satu mekanisme yang banyak digunakan untuk meningkatkan kemanan jaringan adalah *Network Intrusion Detection System* (NIDS) (Gu et al., 2024). NIDS berfungsi memantau lalu lintas jaringan dan mendeteksi aktivitas yang mengindikasikan adanya serangan atau anomali di bagian keamanan. Dalam perkembangannya, pendekatan *machine learning* banyak diterapkan pada NIDS karena mampu mempelajari pola serangan data lalu lintas jaringan dan menghasilkan performa deteksi yang tinggi (Hasnain et al., 2026). Berbagai algoritma *machine learning*, seperti Random Forest, eXtreme Gradient Boosting (XGBoost), dan Multi-Layer Perceptron (MLP), telah banyak digunakan untuk mengklasifikasikan lalu lintas jaringan ke dalam kategori normal maupun serangan (Kostas et al., 2025).

Meskipun mampu menghasilkan performa klasifikasi yang baik, sebagian besar model *machine learning* memiliki karakteristik *black box* sehingga proses pengambilan keputusannya sulit dipahami oleh pengguna (Wali et al., 2025). Dalam konteks keamanan jaringan, kondisi ini menjadi tantangan karena administrator jaringan tidak hanya membutuhkan hasil prediksi, tetapi juga alasan yang mendasari suatu aktivitas dikategorikan sebagai serangan. Kurangnya transparansi model dapat mengurangi tingkat kepercayaan terhadap sistem serta menyulitkan proses validasi hasil deteksi (Corea et al., 2024).

Untuk mengatasi permasalahan tersebut, berkembang pendekatan *Explainable Artificial Intelligence* (XAI) yang bertujuan meningkatkan transparansi dan interpretabilitas model *machine learning* (Arreche, Guntur, et al., 2024). Melalui XAI, pengguna dapat memahami faktor-faktor yang memengaruhi keputusan model sehingga hasil klasifikasi menjadi lebih mudah dijelaskan (Shoukat et al., 2025). Dua metode yang banyak digunakan dalam XAI

adalah Shapley Additive exPlanations (SHAP) dan Local Interpretable Model-agnostic Explanations (LIME). SHAP dapat digunakan untuk memberikan penjelasan secara global maupun lokal, sedangkan LIME banyak digunakan untuk menjelaskan prediksi pada suatu *instance* tertentu (Arreche, Guntur, et al., 2024).

Berbagai penelitian terkait NIDS berbasis *machine learning* umumnya berfokus pada peningkatan performa klasifikasi menggunakan metrik seperti *accuracy, precision, recall,* dan, *F1-Score* (Arreche, Guntur, et al., 2024). Namun, analisis mengenai alasan di balik keputusan model masih relatif terbatas (AL & Sagiroglu, 2025). Padahal, pemahaman terhadap fitur-fitur yang memengaruhi proses deteksi sangat penting untuk meningkatkan kepercayaan pengguna terhadap sistem keamanan jaringan (Shoukat et al., 2025). Selain itu, diperlukan analisis terhadap hasil interpretasi yang diberikan oleh metode *explainability* untuk memastikan bahwa keputusan model dapat dipahami secara lebih komprehensif (Arreche, Guntur, et al., 2024).

Berdasarkan permasalahan tersebut, penelitian ini menerapkan algoritma Random Forest, XGBoost, dan MLP pada dataset CIC-IDS2017 untuk membangun model NIDS. Pemilihan dataset ini didasari oleh kemampuannya merepresentasikan topologi jaringan modern dan variasi serangan mutakhir yang tidak lagi dapat diakomodasi oleh dataset-dataset NIDS generasi terdahulu. Selanjutnya dilakukan analisis interpretabilitas menggunakan metode SHAP untuk memperoleh penjelasan global dan lokal terhadap keputusan model. Selain itu, metode LIME digunakan sebagai pembanding pada *local explanation* untuk mengevaluasi konsistensi hasil interpretasi yang dihasilkan. Penelitian ini diharapkan dapat memberikan pemahaman yang lebih baik mengenai proses pengambilan keputusan model *machine learning* pada sistem deteksi intrusi jaringan.

1.2 Rumusan Masalah

Masalah yang akan dibahas pada penelitian ini adalah:

a. Bagaimana performa algoritma Random Forest, XGBoost, dan MLP dalam mendeteksi serangan jaringan menggunakan dataset CIC-IDS2017?
b. Bagaimana metode SHAP dapat menjelaskan keputusan model *Intrusion Detection System* secara global dan lokal?
c. Bagaimana kesesuaian hasil *local explanations* yang dihasilkan SHAP dan LIME pada model terbaik?

1.3 Lingkup Masalah

Untuk membatasi ruang lingkup penelitian agar pembahasan lebih terarah, ditetapkan masalah sebagai berikut:

a. Dataset yang digunakan dalam penelitian ini adalah CIC-IDS2017 yang telah melalui proses pembersihan data dan penyeimbangan jumlah sampel pada setiap kelas.
b. Kelas yang digunakan dalam penelitian ini terdiri dari Benign, Bot, Bruteforce, DoS, PortScan, dan WebAttack
c. Algoritma *machine learning* yang digunakan untuk membangun model NIDS adalah Random Forest, XGBoost, dan MLP.
d. Metode XAI yang digunakan adalah SHAP dan LIME.
e. Analisis SHAP dilakukan pada tingkat *global explanations* dan *local explanation*.
f. Analisis LIME digunakan sebagai pembanding terhadap hasil *local explanation* yang dihasilkan oleh SHAP.
g. Evaluasi performa model dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-Score*.
h. Penelitian difokuskan pada analisis interpretabilitas model dalam mendeteksi serangan jaringan dan tidak membahas implementasi sistem NIDS secara *real-time*.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah ditetapkan, tujuan dari penelitian ini adalah sebagai berikut:

a. Menganalisis performa algoritma Random Forest, XGBoost, dan MLP dalam mendeteksi serangan jaringan menggunakan dataset CIC-IDS2017.
b. Menganalisis interpretabilitas model IDS menggunakan metode SHAP melalui *global explanation* dan *local explanation*.
c. Menganalisis kesesuaian hasil *local explanation* yang dihasilkan oleh SHAP dan LIME pada model dengan performa terbaik.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah sebagai berikut:

a. Bagi akademisi dan peneliti, penelitian ini diharapkan dapat menjadi referensi dalam pengembangan XAI pada bidang NIDS serta memberikan gambaran mengenai

penggunaan metode SHAP dan LIME untuk menjelaskan keputusan model *machine learning*.

b. Bagi praktisi keamanan jaringan, penelitan ini diharapkan dapat membantu dalam memahami faktor-faktor yang memengaruhi keputusan model NIDS sehingga hasil deteksi serangan dapat diintegrasikan dengan lebih baik.

c. Bagi pengembangan sistem keamanan jaringan, penelitian ini diharapkan dapat memberikan informasi mengenai fitur-fitur yang berpengaruh dalam proses deteksi serangan serta mendukung pengembangan model *Intrusion Detection System* yang lebih transparan dan mudah dipahami.

1.6 Sistematika Penulisan

Untuk memberikan gambaran mengenai penelitian yang dilakukan, penulisan tugas akhir ini disusun dalam lima bab yang saling berkaitan dengan sistematika sebagai berikut:

a. BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah, lingkup masalah, tujuan penelitian, manfaat penelitan, serta sistematika penulisan tugas akhir.

b. BAB II LANDASAN TEORI

Bab ini berisi teori-teori yang menjadi dasar dalam penelitan, meliputi NIDS, dataset CIC-IDS2017, *machine learning*, Random Forest, XGBoost, MLP, XAI, SHAP, LIME, metrik evaluasi, serta penelitian terkait.

c. BAB III METODOLOGI PENELITIAN

Bab ini berisi tahapan penelitian yang dilakukan, mulai dari pengumpulan dan prapemrosesan data, pembangunan model *machine learning*, evaluasi model, penerapan metode SHAP dan LIME, hingga analisis hasil penelitian.

d. BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil eskperimen yang diperoleh dari penerapan model Random Forest, XGBoost, dan MLP pada dataset CIC-IDS2017. Selain itu, bab ini membahas hasil evaluasi model, analisis *global* dan *local explanation* menggunakan SHAP, perbandingan *local explanation* antara SHAP dan LIME, serta interpretasi hasil penelitian.

e. BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh berdasarkan hasil penelitian serta saran yang dapat digunakan untuk pengembangan penelitian selanjutnya.

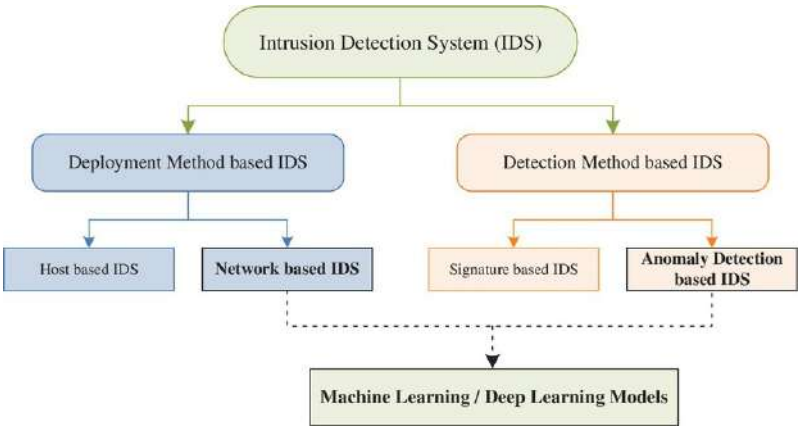
BAB II
LANDASAN TEORI

Bab ini membahas teori-teori yang menjadi dasar dalam penelitian. Teori yang dibahas meliputi NIDS, dataset CIC-IDS2017, *machine learning*, *Random Forest*, XGBoost, MLP, XAI, SHAP, LIME, metrik evaluasi model, serta penelitian terkait. Teori-teori tersebut digunakan sebagai landasan dalam membangun model NIDS dan menganalisis interpretabilitas model yang dihasilkan.

2.1 *Intrusion Detection System*

Intrusion Detection System (IDS) merupakan salah satu mekanisme keamanan jaringan yang digunakan untuk memantau aktivitas pada jaringan maupun sistem komputer guna mendeteksi adanya serangan atau pelanggaran terhadap kebijakan keamanan. IDS bekerja dengan melakukan pemantauan dan analisis terhadap lalu lintas jaringan atau aktivitas sistem untuk mengidentifikasi perilaku yang mencurigakan sehingga administrator dapat memperoleh peringatan (*alert*) dan mengambil tindakan yang diperlukan guna menjaga kerahasiaan, integritas, dan ketersediaan informasi. Seiring dengan meningkatnya kompleksitas ancaman siber, IDS menjadi salah satu komponen penting dalam menjaga keamanan infrastruktur jaringan modern (Ahmad et al., 2021).

IDS dapat diklasifikasikan berdasarkan metode penempatan (*deployment method*) dan metode deteksi (*detection method*) yang digunakan. Berdasarkan metode penempatannya, IDS dibedakan menjadi *Host-based Intrusion Detection System* (HIDS) dan *Network Intrusion Detection System* (NIDS). HIDS melakukan pemantauan terhadap aktivitas yang terjadi pada perangkat atau *host* tertentu, sedangkan NIDS melakukan pemantauan terhadap lalu lintas data yang melintasi jaringan. Sementara itu, berdasarkan metode deteksinya, IDS diklasifikasikan menjadi *Signature-based* IDS dan *Anomaly-based* IDS (Ahmad et al., 2021). Klasifikasi IDS dapat dilihat pada Gambar 2.1.



Gambar 2.1 Klasifikasi *Intrusion Detection System* (Ahmad et al., 2021)

Berdasarkan Gambar 2.1, klasifikasi IDS memberikan gambaran mengenai cara sistem ditempatkan dan mekanisme yang digunakan dalam mendeteksi ancaman keamanan. Penelitian ini berfokus pada NIDS karena menggunakan dataset CIC-IDS2017 yang merepresentasikan lalu lintas jaringan. Selain itu, penelitian ini menerapkan pendekatan klasifikasi berbasis *machine learning* terhadap data berlabel untuk mengidentifikasi beberapa jenis serangan jaringan, sehingga pembahasan selanjutnya difokuskan pada NIDS dan algoritma klasifikasi yang digunakan dalam penelitian ini.

2.1.1 *Network Intrusion Detection System* (NIDS)

Secara fundamental, IDS bekerja layaknya kamera pengawas dan pos penjagaan dalam sebuah sistem keamanan infrastruktur. Dalam konteks jaringan komputer, NIDS bertindak sebagai pengawas yang memantau "jalan raya" lalu lintas paket data secara keseluruhan, berbeda dengan *Host-based Intrusion Detection* (HIDS) yang hanya memantau dari dalam satu "kamar" atau perangkat spesifik. NIDS biasanya ditempatkan pada titik-titik strategis dalam jaringan sehingga mampu mengawasi paket data yang melewati segmen jaringan tertentu (Sharma et al., 2022).

Sistem ini bekerja dengan menganalisis karakteristik lalu lintas jaringan untuk mengidentifikasi aktivitas yang berpotensi sebagai serangan atau pelanggaran keamanan. Secara tradisional, mekanisme deteksi NIDS terbagi menjadi dua pendekatan utama: *Signature-based* (mendeteksi aktivitas berdasarkan pola serangan yang sudah diketahui sebelumnya) dan *Anomaly-based* (mendeteksi aktivitas yang menyimpang dari perilaku normal). Di era modern, pendekatan *Anomaly-based* sangat bergantung pada algoritma

machine learning untuk mengidentifikasi varian serangan baru (*zero-day attack*) yang belum pernah tercatat sebelumnya (Sharma et al., 2022).

NIDS banyak diterapkan untuk mendeteksi berbagai jenis serangan jaringan, seperti *Denial of Service* (DoS), *Port Scanning*, *BruteForce*, dan *Botnet*. Kemampuan tersebut menjadikan NIDS sebagai salah satu mekanisme yang banyak digunakan dalam menjaga keamanan jaringan modern. Pada penelitian ini, NIDS menjadi fokus pembahasan karena dataset CIC-IDS2017 yang digunakan berisi data lalu lintas jaringan yang telah diberi label ke dalam beberapa kategori serangan. Karakteristik tersebut memungkinkan penerapan model klasifikasi berbasis *machine learning* untuk mengidentifikasi aktivitas normal maupun berbagai jenis serangan jaringan (Ahmad et al., 2021).

2.2 Dataset CIC-IDS2017

Dataset merupakan salah satu komponen penting dalam pengembangan model *machine learning* karena kualitas data akan memengaruhi kemampuan model dalam mempelajari pola dan menghasilkan prediksi yang akurat. Pada penelitian di bidang NIDS, dataset digunakan sebagai representasi aktivitas lalu lintas jaringan yang terdiri atas aktivitas normal maupun berbagai jenis serangan siber. Salah satu dataset yang banyak digunakan dalam penelitian NIDS adalah Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CIC-IDS2017) (Sharafaldin et al., 2018).

CIC-IDS2017 merupakan dataset yang dikembangkan oleh Canadian Institute for Cybersecurity (CIC) dengan tujuan menyediakan data lalu lintas jaringan yang realistis dan mencerminkan kondisi jaringan modern. Dataset ini dibangun menggunakan kombinasi aktivitas normal pengguna dan berbagai skenario serangan yang dilakukan dalam lingkungan jaringan yang menyerupai kondisi nyata.

Pemilihan dataset CIC-IDS2017 dalam penelitian ini dilandasi oleh beberapa alasan ilmiah yang kuat dalam domain keamanan siber:

1. Representasi Topologi Jaringan Modern: Banyak penelitian terdahulu masih menggunakan dataset usang seperti KDD Cup '99 atau NSL-KDD yang dibangun puluhan tahun lalu. Dataset tersebut sudah tidak relevan dengan arsitektur jaringan dan serangan modern (seperti HTTPS, serangan berbasis web, dan Botnet). CIC-IDS2017 dibangun menggunakan simulasi jaringan yang menyerupai kondisi nyata saat ini (B-Profile) dengan protokol-protokol terbaru.
2. Variasi Serangan yang Lengkap dan Mutakhir: Dataset ini merangkum skenario serangan yang paling sering mengancam operasional bisnis dan infrastruktur saat ini,

seperti BruteForce, DoS, Web Attack, PortScan, dan Botnet. Hal ini membuat model yang dilatih menggunakan dataset ini memiliki nilai aplikatif yang tinggi jika diimplementasikan pada skenario industri nyata.

3. Ekstraksi Fitur Berbasis Aliran (Flow-based): Dataset ini tidak hanya menyediakan data mentah berbasis PCAP, tetapi telah mengekstraksi lebih dari 78 fitur lalu lintas jaringan menggunakan alat CICFlowMeter. Fitur-fitur statistik ini (seperti durasi aliran, ukuran paket, hingga status flag TCP) sangat ideal untuk diolah oleh algoritma klasifikasi machine learning.
4. Kesesuaian Ruang Lingkup dan Kematangan Benchmark: Meskipun saat ini telah tersedia beberapa dataset generasi yang lebih baru, seperti CSE-CIC-IDS2018 atau CICIoT2023, CIC-IDS2017 tetap dipertahankan sebagai pilihan utama karena alasan ruang lingkup dan stabilitas. Dataset terbaru seperti CICIoT2023 dirancang sangat spesifik untuk lingkungan *Internet of Things* (IoT), sehingga kurang merepresentasikan lalu lintas jaringan IT enterprise secara umum yang menjadi fokus penelitian ini. Selain itu, CIC-IDS2017 saat ini telah diakui sebagai standar acuan yang sangat matang dalam literatur keamanan siber global. Hal ini memungkinkan hasil performa dan interpretasi XAI pada penelitian ini dapat divalidasi dan dibandingkan secara *apple-to-apple* dengan berbagai literatur terdahulu.

Karakteristik Umum dataset CIC-IDS2017 disajikan pada Tabel 2.1.

Tabel 2.1 Karakteristik Dataset CIC-IDS2017 (Sharafaldin et al., 2018)

Karakteristik	**Keterangan**
Nama Dataset	CIC- IDS
Pengembangan	Canadian Institute for Cybersecurity (CIC)
Tahun Publikasi	2017
Jenis Data	*Network Traffic Flow*
Lingkungan	Simulasi jaringan yang menyerupai kondisi nyata
Label	Aktivitas normal dan berbagai jenis serangan siber
Tujuan	Evaluasi dan pengembangan *Intrusion Detection* berbasis *machine learning*

Dataset CIC-IDS2017 terdiri atas sejumlah fitur yang menggambarkan karakteristik setiap *network flow*, seperti paket, durasi komunikasi, ukuran paket, waktu antar paket, status komunikasi, serta atribut statistik lainnya. Fitur-fitur tersebut digunakan sebagai dasar dalam membedakan aktivitas normal dan aktivitas serangan sehingga sangat sesuai digunakan pada penelitian berbasis *machine learning* (Sharafaldin et al., 2018).

Secara umum, fitur-fitur pada dataset CIC-IDS2017 dapat dikelompokkan ke dalam beberapa kategori berdasarkan karakteristik informasi yang direpresentasikan. Pengelompokan fitur tersebut disajikan pada Tabel 2.2

Tabel 2.2 Kelompok Fitur pada Dataset CIC-IDS2017 (Sharafaldin et al., 2018)

Kelompok Fitur	**Contoh Fitur**	**Fungsi**
Basic Flow Features	Destination Port, Flow Duration	Informasi dasar aliran jaringan
Packet features	Packet Length Mean, Total Length of Fwd Packets	Karakteristik ukuran paket
Time Features	Flow IAT Mean, Bwd IAT Total	Karakteristik waktu antar paket
Flag Features	SYN Flag Count, Flag Count	Status komunikasi TCP
Window Features	Init_Win_bytes_forward	Informasi ukuran TCP Window
Active/Idle Features	Active mean, Idle Mean	Aktivitas dan waktu jeda koneksi

Berdasarkan Tabel 2.2, setiap kelompok fitur memiliki peran yang berbeda dalam menggambarkan karakteristik lalu lintas jaringan. Informasi tersebut menjadi dasar bagi algoritma *machine learning* untuk mempelajari pola aktivitas normal maupun aktivitas serangan. Selain itu, fitur-fitur tersebut menjadi dasar dalam proses interpretasi menggunakan metode SHAP dan LIME untuk mengetahui kontribusi masing-masing fitur terhadap hasil prediksi model.

Selain Terdiri atas berbagai kelompok fitur, dataset CIC-IDS2017 juga memiliki beragam kelas yang merepresentasikan aktivitas normal dan berbagai jenis serangan siber. Distribusi jumlah sampel pada setiap kelas disajikan pada Tabel 2.3. Berdasarkan distribusi

tersebut, terlihat bahwa dataset memiliki ketidakseimbangan jumlah sampel antarkelas, di mana kelas *BENIGN* memiliki jumlah data yang jauh lebih besar dibandingkan sebagian besar kelas serangan. Selain itu, beberapa jenis serangan masih direpresentasikan dalam bentuk subkelas, seperti DoS yang terdiri atas DoS Hulk, DoS GoldenEye, DoS Slowhttptest, dan DoS Slowloris, serta Web Attack yang terdiri atas WebAttack – BruteForce, Web Attack – XSS, dan Web Attack – SQL Injection. Karakteristik tersebut menjadi salah satu pertimbangan dilakukannya tahap *preprocessing* sebelum dataset digunakan pada penelitian ini

Tabel 2.3 Distribusi Kelas pada Dataset CIC-IDS2017

Kelas	Jumlah Sampel
BENIGN	2.273.097
DoS Hulk	231.073
PortScan	158.930
DdoS	128.027
DoS GoldenEye	10.923
FTP-Patator	7.938
SSH-Patator	5.897
Dos Slowloris	5.796
Dos Slowhttptest	5.499
Bot	1.966
Web Attack – BruteForce	1.507
Web Attack – XSS	652
Infiltration	36
Web Attack – SQL Injection	21
Heartbleed	11

Pada penelitian ini, tidak seluruh kelas pada dataset asli digunakan dalam proses pemodelan. Hal tersebut disebabkan adanya tahapan *preprocessing* yang meliputi pembersihan data, pengelompokan beberapa label serangan ke dalam kategori yang lebih umum, serta penyesuaian kelas agar sesuai dengan tujuan penelitian. Setelah proses tersebut dilakukan, penelitian ini menggunakan enam kelas utama yaitu, Benign, Bot, BruteForce, DoS, PortScan, dan WebAttack, sebagaimana disajikan pada Tabel 2.4

Tabel 2.4 Kelas yang Digunakan pada Penelitian

Kelas	Deskripsi
Benign	Aktivitas lalu lintas jaringan normal tanpa adanya indikasi serangan

Bot	Aktivitas perangkat yang telah terinfeksi *malware* dan dikendalikan sebagai bagian dari *botnet*.
BruteForce	Serangan yang mencoba berbagai kombinasi kredensial untuk memperoleh akses ke sistem
DoS	Serangan yang bertujuan mengganggu ketersediaan layanan dengan membanjiri target menggunakan lalu lintas jumlah yang besar
PortScan	Aktivitas pemindaian *port* untuk mengidentifikasi layanan terbuka pada suatu *host*
WebAttack	Serangan yang menargetkan aplikasi web, seperti *SQL Injection*, *Cross-site Scripting* (XSS), maupun serangan web lainnya

Berdasarkan karakteristik dataset, kelompok fitur, dan kelas serangan yang digunakan, CIC-IDS2017 dinilai mampu merepresentasikan berbagai pola lalu lintas jaringan yang diperlukan dalam penelitian ini. Dataset tersebut selanjutnya digunakan sebagai dasar dalam proses pelatihan dan pengujian model *machine learning* yang terdiri atas *Random Forest*, XGBoost, dan MLP yang akan dibahas pada subbab berikutnya.

2.3 *Machine Learning*

Machine Learning merupakan salah satu cabang dari *Artificial Intelligence* (AI) yang memungkinkan komputer mempelajari pola dari data tanpa harus diprogram secara eksplisit. Melalui proses pembelajaran tersebut, model *machine learning* mampu membangun hubungan antar data sehingga dapat digunakan untuk melakukan prediksi, klasifikasi, maupun pengambilan keputusan terhadap data baru. Kemampuan tersebut menjadikan *machine learning* banyak diterapkan pada berbagai bidang, termasuk keamanan jaringan, kesehatan, sistem rekomendasi, dan pengenalan pola (Kotsiantis et al., 2006).

Dalam bidang keamanan jaringan, *machine learning* dimanfaatkan untuk membangun model yang mampu mengklasifikasikan lalu lintas jaringan berdasarkan karakteristik data yang dipelajari. Dengan memanfaatkan data berlabel, model dapat mengenali pola yang membedakan aktivitas normal dan berbagai jenis serangan sehingga mampu melakukan proses klasifikasi terhadap data baru secara otomatis. Pendekatan ini banyak diterapkan pada NIDS untuk meningkatkan kemampuan dalam mengidentifikasi berbagai jenis serangan jaringan (Sharma et al., 2022).

Penelitian ini menerapkan pendekatan *supervised learning* karena dataset CIC-IDS2017 yang digunakan telah memiliki label kelas untuk setiap data. Pada *supervised learning*, proses pelatihan dilakukan menggunakan data yang telah diketahui kelasnya sehingga model dapat mempelajari hubungan antara atribut (*feature*) dan label untuk menghasilkan prediksi pada data baru (Kotsiantis et al., 2006). Algoritma yang digunakan dalam penelitian ini terdiri atas Random Forest, XGBoost, dan MLP. Ketiga algoritma tersebut dipilih karena memiliki karakteristik yang berbeda dalam proses pembelajaran data sehingga dapat digunakan untuk membandingkan performa klasifikasi serta hasil interpretasi menggunakan XAI. Penjelasan mengenai masing-masing algoritma disajikan pada subbab berikutnya.

2.3.1 Random Forest

Random Forest merupakan salah satu algoritma *supervised learning* yang termasuk dalam metode *ensemble learning*. Algoritma ini bekerja dengan membangun sejumlah *decision tree* dari data pelatihan, kemudian menggabungkan hasil prediksi dari setiap pohon untuk menghasilkan keputusan akhir. Pada proses klasifikasi, hasil prediksi ditentukan berdasarkan mekanisme *majority voting*, yaitu kelas yang paling banyak dipilih oleh seluruh *decision tree* akan menjadi hasil prediksi model (Breiman, 2001).

Proses pembentukan *Random Forest* diawali dengan pemilihan sampel data secara acak menggunakan teknik *bootstrap sampling*. Setiap sampel digunakan untuk membangun sebuah *decision tree* yang berbeda. Selain itu, pada setiap proses pemisahan (*split*), algoritma juga memilih sejumlah fitur secara acak sehingga setiap *decision tree* memiliki karakteristik yang berbeda. Pendekatan tersebut bertujuan untuk meningkatkan keberagaman antar pohon keputusan dan mengurangi risiko *overfitting* (Breiman, 2001).

Random Forest memiliki beberapa kelebihan, di antaranya mampu menghasilkan performa klasifikasi yang baik pada data berdimensi tinggi, lebih tahan terhadap *overfitting* dibandingkan *decision tree* tunggal, serta mampu menangani data dengan jumlah fitur yang cukup banyak. Selain itu, *Random Forest* juga dapat memberikan informasi mengenai tingkat kepentingan (*feature importance*) dari setiap fitur yang digunakan dalam proses klasifikasi (Breiman, 2001).

Di sisi lain, *Random Forest* juga memiliki beberapa keterbatasan. Semakin banyak jumlah *decision tree* yang digunakan, semakin besar pula kebutuhan komputasi dan memori selama proses pelatihan maupun prediksi. Selain itu, meskipun *Random Forest* mampu menghasilkan tingkat akurasi yang tinggi, proses pengambilan keputusannya relatif sulit

dipahami karena melibatkan banyak pohon keputusan, sehingga model ini termasuk ke dalam kategori *black-box* model (Breiman, 2001).

Pada penelitian ini, *Random Forest* digunakan sebagai salah satu model klasifikasi untuk mendeteksi aktivitas normal maupun serangan pada dataset CIC-IDS2017. Selain digunakan untuk mengevaluasi performa klasifikasi, model *Random Forest* juga dianalisis menggunakan metode SHAP untuk mengetahui kontribusi masing-masing fitur terhadap hasil prediksi model.

2.3.2 eXtreme Gradient Boosting (XGBoost)

eXtreme Gradient Boosting (XGBoost) merupakan algoritma *supervised learning* yang dikembangkan berdasarkan metode *Gradient Boosting Decision Tree* (GBDT). Algoritma ini membangun model secara bertahap dengan menambahkan *decision tree* baru untuk memperbaiki kesalahan prediksi yang dihasilkan oleh pohon sebelumnya. Pendekatan tersebut memungkinkan model meningkatkan akurasi prediksi melalui proses pembelajaran yang dilakukan secara berurutan (Chen & Guestrin, 2016).

Pada proses pelatihannya, setiap *decision tree* yang dibangun berfokus pada data yang masih memiliki tingkat kesalahan prediksi tinggi pada iterasi sebelumnya. Kesalahan tersebut dihitung menggunakan fungsi *loss*, kemudian diminimalkan melalui proses optimasi berbasis *gradient descent.* Selain itu, XGBoost menerapkan teknik regularisasi untuk mengendalikan kompleksitas model sehingga mampu mengurangi risiko *overfitting* dan meningkatkan kemampuan generalisasi terhadap data baru (Chen & Guestrin, 2016).

XGBoost memiliki berbagai kelebihan, di antaranya mampu menghasilkan performa klasifikasi yang tinggi, memiliki efisiensi komputasi yang baik, serta mampu menangani data berdimensi tinggi dan data dengan jumlah fitur yang besar. Algoritma ini juga mendukung pemanfaatan *parallel processing*, penanganan *missing value*, serta mekanisme *feature importance* yang dapat digunakan untuk mengetahui kontribusi masing-masing fitur terhadap hasil prediksi (Chen & Guestrin, 2016).

Meskipun demikian, XGBoost memiliki beberapa keterbatasan. Proses pelatihan memerlukan penentuan beberapa parameter yang cukup banyak sehingga proses *hyperparamater tuning* dapat menjadi lebih kompleks dibandingkan algoritma klasifikasi lainnya. Selain itu, model yang dihasilkan tetap termasuk ke dalam kategori *black-box model*, sehingga proses pengambilan keputusan sulit dipahami secara langsung tanpa menggunakan interpretasi tambahan (Chen & Guestrin, 2016).

Pada penelitian ini, XGBoost digunakan sebagai salah satu model klasifikasi untuk mendeteksi aktivitas normal maupun berbagai jenis serangan pada dataset CIC-IDS2017. Selain dilakukan evaluasi performa menggunakan metrik klasifikasi, hasil prediksi model XGBoost juga dianalisis menggunakan metode SHAP untuk memperoleh interpretasi terhadap kontribusi setiap fitur dalam proses pengambilan keputusan model.

2.3.3 Multi-Layer Perceptron (MLP)

Multi-Layer Perceptron (MLP) merupakan salah satu algoritma *supervised learning* yang termasuk kelompok *Artificial Neural Network* (ANN). MLP dirancang untuk mempelajari hubungan yang kompleks antara data masukan dan keluaran melalui beberapa lapisan neuron yang saling terhubung. Algoritma ini banyak digunakan pada berbagai pemasalahan klasifikasi maupun prediksi karena mampu memodelkan hubungan nonlinier yang sulit diselesaikan menggunakan metode konvensional (Alcineide et al., 2021).

Secara umum, arsitektur MLP terdiri atas tiga jenis lapisan, yaitu *input layer, hidden layer,* dan *output layer*. Ketiga lapisan tersebut bekerja secara berurutan untuk mengolah informasi hingga menghasilkan prediksi sesuai dengan jumlah kelas yang ditentukan. Selama proses pelatihan, MLP melakukan penyesuaian bobot (*weights*) pada setiap koneksi antar neuron menggunakan algoritma *backpropagation* dan metode optimasi tertentu untuk meminimalkan nilai kesalahan (*loss function*) (Alcineide et al., 2021).

MLP memiliki beberapa kelebihan, di antaranya mampu mempelajari hubungan nonlinier yang kompleks, memiliki fleksibilitas dalam menangani berbagai jenis data, serta mampu menghasilkan performa klasifikasi yang baik apabila didukung oleh jumlah data yang memadai. Selain itu, MLP dapat digunakan pada berbagai permasalahan klasifikasi multikelas sehingga banyak diterapkan dalam penelitian berbasis *machine learning* (Alcineide et al., 2021).

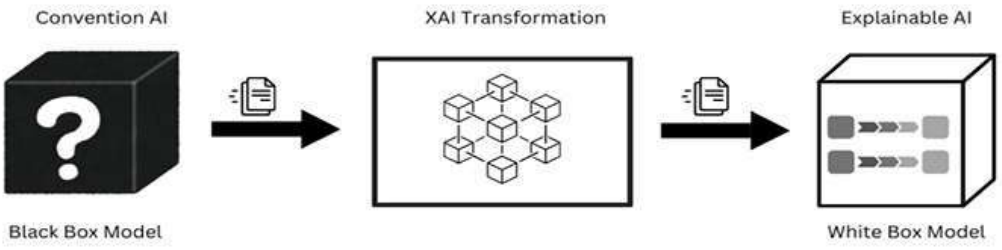
Di sisi lain, MLP juga memiliki beberapa keterbatasan. Proses pelatihan umumnya memerlukan waktu komputasi yang lebih lama dibandingkan beberapa algoritma klasifikasi lainnya, terutama ketika menggunakan jumlah neuron dan *hidden layer* yang besar. Selain itu, proses penentuan arsitektur jaringan, seperti jumlah lapisan tersembunyi, jumlah neuron, serta parameter pelatihan, memerlukan penyesuaian yang tepat agar model dapat menghasilkan performa yang optimal. Sama seperti *Random Forest* dan XGBoost, MLP juga termasuk ke dalam kategori *black-box model* sehingga proses pengambilan keputusannya sulit dipahami secara langsung (Alcineide et al., 2021).

Pada penelitian ini, MLP digunakan sebagai salah satu model klasifikasi untuk mendeteksi aktivitas normal maupun berbagai jenis serangan pada dataset CIC-IDS2017. Hasil prediksi model kemudian dianalisis menggunakan metode SHAP dan LIME untuk memberikan interpretasi terhadap kontribusi masing-masing fitur dalam proses klasifikasi.

2.4 Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI) merupakan pendekatan dalam pengembangan kecerdasan buatan yang bertujuan untuk meningkatkan transparansi dan interpretabilitas model sehingga proses pengambilan keputusan dapat dipahami oleh manusia. Berbeda dengan model *machine learning* konvensional yang hanya menghasilkan prediksi, XAI memberikan informasi mengenai alasan di balik suatu keputusan dengan menunjukkan kontribusi setiap fitur terhadap hasil prediksi. Dengan demikian, pengguna tidak hanya mengetahui hasil akhir yang diberikan oleh model, tetapi juga memahami faktor-faktor yang memengaruhi keputusan tersebut (Arreche, Guntur, et al., 2024).

Perkembangan algoritma *machine learning* telah menghasilkan model dengan tingkat akurasi yang tinggi, seperti *Random Forest*, XGBoost, dan MLP. Namun, model-model tersebut umumnya termasuk ke dalam kategori *black-box model* karena proses pengambilan keputusannya sulit dipahami secara langsung oleh pengguna. Kondisi tersebut menjadi tantangan dalam penerapan *machine learning*, khususnya pada bidang yang memerlukan tingkat kepercayaan yang tinggi, seperti keamanan siber (Arreche, Guntur, et al., 2024).



Gambar 2.2 Transformasi Model *Black-box* menuju *Explainable Artificial Intelligence* (Nada Marey et al., 2026)

Berdasarkan Gambar 2.2, pendekatan *machine learning* konvensional umumnya menghasilkan prediksi melalui mekanisme *black-box* yang sulit dipahami oleh pengguna. Melalui penerapan XAI, model tetap mempertahankan kemampuan prediksinya, dan juga

dilengkapi dengan mekanisme yang mampu memberikan penjelasan terhadap proses pengambilan keputusan. Dengan demikian, pengguna dapat mengetahui faktor-faktor yang berkontribusi terhadap hasil prediksi sehingga interpretasi model menjadi lebih transparan (Nada Marey et al., 2026).

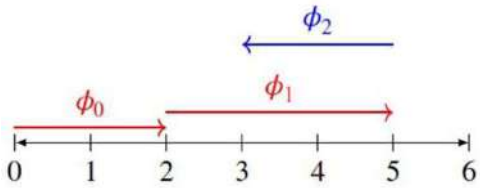
Dalam bidang NIDS, interpretabilitas model menjadi aspek yang penting karena hasil prediksi yang diberikan tidak hanya digunakan untuk mendeteksi aktivitas normal maupun serangan, tetapi juga membantu analis keamanan memahami karakteristik serangan yang teridentifikasi. Informasi mengenai fitur-fitur yang paling berpengaruh terhadap keputusan model dapat digunakan sebagai dasar dalam proses analisis, validasi, maupun pengambilan keputusan dalam menangani ancaman keamanan jaringan (Wali et al., 2025).

Pada penelitian ini, XAI digunakan untuk menginterpretasikan hasil prediksi model *Random Forest*, XGBoost, MLP. Metode yang digunakan terdiri atas *SHapley Additive exPlanations* (SHAP) sebagai metode utama untuk menghasilkan interpretasi global maupun lokal, serta *Local Interpretable Model-agnostic Explanations* (LIME) sebagai metode pembanding dalam memberikan penjelasan terhadap prediksi model. Penjelasan mengenai kedua metode tersebut dibahas pada subbab berikutnya.

2.4.1 Shapley Additive exPlanations (SHAP)

Shapley Additive exPlanations (SHAP) merupakan salah satu metode XAI yang digunakan untuk menjelaskan hasil prediksi model *machine learning*. SHAP dikembangkan berdasarkan konsep *Shapley Value* dalam teori permainan (*cooperative game theory*), yaitu pendekatan yang digunakan untuk menghitung kontribusi setiap pemain terhadap hasil akhir suatu permainan. Dalam konteks *machine learning*, setiap fitur diperlakukan sebagai pemain yang memberikan kontribusi terhadap hasil prediksi model (Lundberg & Lee, 2017).

Metode SHAP bekerja dengan menghitung besarnya kontribusi masing-masing fitur terhadap hasil prediksi model. Kontribusi tersebut dinyatakan dalam bentuk nilai SHAP (SHAP *value*) yang menunjukkan seberapa besar suatu fitur meningkatkan maupun menurunkan nilai prediksi. Nilai SHAP yang positif menunjukkan bahwa suatu fitur mendorong prediksi menuju kelas tertentu, sedangkan nilai SHAP yang negatif menunjukkan bahwa fitur tersebut mengurangi kemungkinan prediksi pada kelas tersebut. Dengan demikian setiap prediksi yang dihasilkan model dapat dijelaskan berdasarkan kontribusi masing-masing fitur (Lundberg & Lee, 2017).



Gambar 2.3 Ilustrasi Kontribusi Fitur pada Shapley Additive exPlanations (SHAP) (Sison et al., 2023)

Berdasarkan Gambar 2.3, proses interpretasi SHAP diawali dari nilai dasar (*base value*) yang merepresentasikan nilai prediksi rata-rata model. Selanjutnya, setiap fitur memberikan kontribusi terhadap prediksi melalui nilai SHAP yang dapat bersifat positif maupun negatif. Pada ilustrasi tersebut, kontribusi positif yang ditunjukkan oleh φ0 dan φ1 menggeser nilai prediksi ke arah yang lebih tinggi, sedangkan kontribusi negatif yang ditunjukkan oleh φ2 mengurangi nilai prediksi akhir. Kombinasi seluruh kontribusi tersebut menghasilkan nilai prediksi yang diberikan oleh model (Sison et al., 2023).

SHAP dapat digunakan untuk menghasilkan dua jenis interpretasi, yaitu *global explanation* dan *local explanation*. *Global explanation* digunakan untuk mengetahui tingkat kepentingan setiap fitur berdasarkan keseluruhan data sehingga dapat menggambarkan perilaku model secara umum. Sementara itu, *local explanation* digunakan untuk menjelaskan kontribusi masing-masing fitur terhadap hasil prediksi pada satu sampel tertentu. Kombinasi kedua jenis interpretasi tersebut memungkinkan pengguna memahami perilaku model baik pada tingkat global maupun tingkat individual (Lundberg & Lee, 2017).

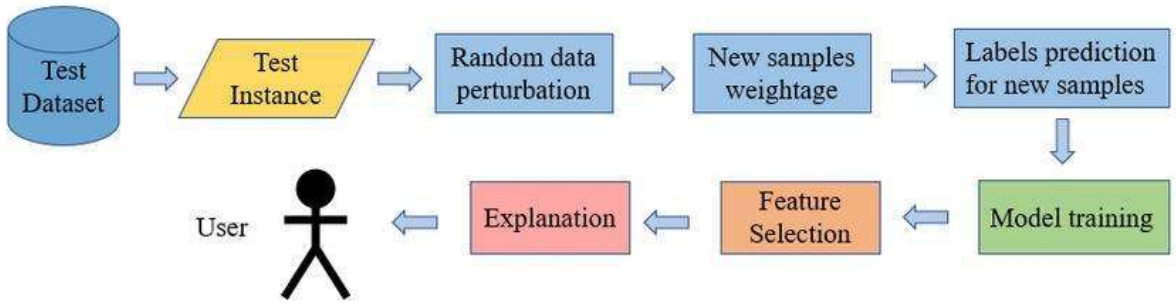
Salah satu keunggulan SHAP adalah memiliki dasar teoritis yang kuat melalui konsep *Shapley Value*, sehingga mampu memberikan interpretasi yang konsisten dan adil terhadap kontribusi setiap fitur. Selain itu, SHAP mendukung berbagai jenis model *machine learning*, termasuk model berbasis pohon keputusan maupun jaringan saraf, serta menyediakan berbagai bentuk visualisasi yang memudahkan proses interpretasi hasil prediksi (Lundberg & Lee, 2017).

Pada penelitian ini, SHAP digunakan metode utama XAI untuk menginterpretasikan hasil prediksi model *Random Forest*, XGBoost, dan MLP. Analisis dilakukan melalui *global explanation* untuk mengidentifikasi fitur-fitur yang paling berpengaruh terhadap model serta *local explanation* untuk menjelaskan kontribusi setiap fitur pada prediksi suatu sampel. Hasil interpretasi tersebut kemudian dibandingkan dengan metode *Local Interpretable Model-agnostic Explanations* (LIME) sebagai pembanding interpretasi lokal.

2.4.2 Local Interpretable Model-agnostic Explanations (LIME)

Local Interpretable Model-agnostic Explanations (LIME) merupakan salah satu metode XAI yang digunakan untuk memberikan penjelasan terhadap hasil prediksi model *machine learning*. Berbeda dengan SHAP yang memiliki dasar teoritis berdasarkan *Shapley Value*, LIME membangun model sederhana (*surrogate* model) di sekitar satu sampel tertentu untuk menjelaskan alasan di balik prediksi yang dihasilkan oleh model utama. Karena hanya berfokus pada satu data, LIME termasuk ke dalam metode *local explanation* (Ribeiro et al., 2016).

LIME bekerja dengan menghasilkan sejumlah data baru (*perturbed samples*) di sekitar data yang akan dijelaskan. Data hasil perturbasi tersebut kemudian diprediksi menggunakan model *black-box*, seperti Random Forest, XGBoost, maupun MLP. Selanjutnya, LIME membangun model lokal yang lebih sederhana berdasarkan hasil prediksi tersebut sehingga hubungan antara fitur dan hasil prediksi dapat diinterpretasikan dengan lebih mudah (Ribeiro et al., 2016).



Gambar 2.4 Workflow Local Interpretable Model-agnostic Explanations (LIME) (Nigar et al., 2022)

Berdasarkan Gambar 2.4, proses interpretasi menggunakan LIME diawali dengan memilih satu data uji (*test instance*) yang akan dijelaskan. Selanjutnya, dilakukan proses *random data perturbation* untuk menghasilkan sejumlah sampel baru di sekitar data tersebut. Sampel-sampel tersebut kemudian diberikan bobot berdasarkan tingkat kedekatannya dengan data asli dan diprediksi menggunakan model *black-box*. Berdasarkan hasil prediksi tersebut, LIME membangun model lokal yang bersifat interpretable untuk melakukan *feature selection* sehingga dapat menghasilkan penjelasan mengenai kontribusi setiap fitur terhadap prediksi pada sampel yang dianalisis (Nigar et al., 2022).

Salah satu kelebihan LIME adalah sifatnya yang *model-agnostic*, sehingga LIME dapat digunakan pada berbagai jenis algoritma *machine learning* tanpa bergantung pada struktur model yang digunakan. Selain itu, LIME mampu menghasilkan interpretasi yang mudah dipahami karena hanya berfokus pada satu prediksi tertentu. Namun, interpretasi yang dihasilkan bersifat lokal sehingga penjelasan dapat berbeda untuk setiap sampel dan tidak selalu merepresentasikan perilaku model secara keseluruhan (Ribeiro et al., 2016).

Pada penelitian ini, LIME digunakan sebagai metode pembanding SHAP dalam memberikan interpretasi lokal terhadap hasil prediksi model Random Forest, XGBoost, dan MLP. Penggunaan LIME bertujuan untuk memberikan perspektif tambahan mengenai kontribusi setiap fitur pada suatu sampel sehingga hasil interpretasi yang diperoleh dapat dibandingkan dengan metode SHAP.

2.5 Evaluasi Model

Evaluasi model merupakan tahapan yang dilakukan untuk mengukur kemampuan model *machine learning* dalam melakukan klasifikasi terhadap data yang belum pernah dipelajari sebelumnya. Hasil evaluasi digunakan untuk mengetahui tingkat performa model serta membandingkan kemampuan masing-masing algoritma dalam menghasilkan prediksi yang sesuai dengan kelas sebenarnya. Pada penelitian klasifikasi multikelas, proses evaluasi umumnya dilakukan menggunakan *Confusion Matrix* yang kemudian digunakan sebagai dasar dalam menghitung berbagai metrik evaluasi, seperti *accuracy, precision, recall,* dan *F1-score (Ahmad et al., 2021)*.

Confusion Matrix merupakan matriks yang menggambarkan hubungan antara kelas aktual dengan kelas hasil prediksi model (Ahmad et al., 2021). Pada kasus klasifikasi multikelas, matriks ini memberikan informasi mengenai jumlah prediksi yang benar maupun yang salah untuk setiap kelas sehingga dapat digunakan untuk mengevaluasi performa model secara lebih rinci, Ilustrasi *Multi-Class Confusion Matrix* ditunjukkan pada Tabel 2.5.

Tabel 2.5 Ilustrasi Penentuan TP, FP, TN, dan FN pada *Multi-Class Confusion Matrix*

		Predicted Class		
		Class 1	*Class 2*	*Class 3*
Actual Class	*Class 1*	*True Positive* (TP)	*False Negative* (FN)	*False Negative* (FN)
	Class 2	*False Positive* (FP)	*True Negative* (TN)	*True Negative* (TN)

	Class 3	*False Positive* (FP)	*True Negative* (TN)	*True Negative* (TN)

Berdasarkan Tabel 2.5, komponen evaluasi pada *Multi-Class Confusion Matrix* dapat dijelaskan menggunakan empat komponen utama, yaitu *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Keempat komponen tersebut menjadi dasar dalam perhitungan metrik evaluasi yang digunakan untuk menilai performa model klasifikasi.

Adapun uraian setiap atribut di dalam *Multi-Class Confusion Matrix* pada Tabel 2.5 sebagai berikut:

a. *True Positive* (TP) merupakan jumlah data yang berhasil diprediksi dengan benar sesuai dengan kelas sebenarnya.
b. *False Positive* (FP) merupakan jumlah data yang diprediksi sebagai suatu kelas tertentu, akan tetapi sebenarnya berasal dari kelas yang berbeda.
c. *False Negative* (FN) merupakan jumlah data yang sebenarnya berasal dari suatu kelas, tetapi diprediksi sebagai kelas lain oleh model.
d. *True Negative* (TN) merupakan jumlah data yang bukan berasal dari suatu kelas tertentu dan juga tidak diprediksi sebagai kelas tersebut oleh model

Informasi yang diperoleh dari keempat komponen tersebut selanjutnya digunakan untuk menghitung nilai *accuracy, precision, recall,* dan *f1-score* sebagai metrik evaluasi performa model klasifikasi. Penjelasan mengenai masing-masing metrik evaluasi disajikan sebagai berikut:

a. *Accuracy* merupakan metrik yang digunakan untuk mengukur proporsi jumlah prediksi yang benar terhadap keseluruhan data yang dievaluasi. Nilai *accuracy* menunjukkan tingkat ketepatan model secara keseluruhan dalam melakukan proses klasifikasi (EvidentlyAITeam, 2023).

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \quad (1)$$

b. *Precision* merupakan metrik yang digunakan untuk mengukur proporsi data yang diprediksi sebagai suatu kelas tertentu dan benar-benar termasuk ke dalam kelas tersebut. Nilai *precision* yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan *False Positive* yang rendah (EvidentlyAITeam, 2023).

$$Precision = \frac{\text{TP}}{\text{TP} + FP} \quad (2)$$

c. *Recall* merupakan metrik yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi seluruh data yang benar-benar termasuk ke dalam suatu kelas. Nilai *Recall* yang tinggi menunjukkan bahwa model mampu meminimalkan jumlah *False Negative* (EvidentlyAITeam, 2023).

$$Recall = \frac{\text{TP}}{\text{TP} + FN} \quad (3)$$

d. *F1-score* merupakan rata-rata harmonik (*harmonic mean*) dari nilai *precision* dan *recall*. Metrik ini digunakan untuk memberikan keseimbangan antara *precision* dan *recall*, sehingga sesuai digunakan pada permasalahan klasifikasi yang memiliki distribusi kelas yang tidak seimbang (Ahmad et al., 2021).

$$F1 - score = 2.\frac{Precision.Recall}{Precision + Recall} \quad (4)$$

2.6 Penelitian Terkait

Penelitian mengenai penerapan *machine learning* dan XAI pada NIDS terus mengalami perkembangan dalam beberapa tahun terakhir. Berbagai penelitian telah mengusulkan algoritma klasifikasi, dataset, serta metode interpretabilitas yang berbeda untuk meningkatkan kemampuan sistem dalam mendeteksi berbagai jenis serangan jaringan. Oleh karena itu, kajian terhadap penelitian terdahulu diperlukan untuk memberikan gambaran mengenai perkembangan penelitian yang telah dilakukan serta menjadi dasar dalam memahami pendekatan yang relevan dengan penelitian ini.

Ringkasan beberapa penelitian terdahulu yang memiliki keterkaitan dengan penelitian ini ditunjukkan pada Tabel 2.6.

Tabel 2.6 Penelitian Terdahulu

Referensi	Dataset	Algoritma Model	Metode XAI	Hasil

(Arreche, Guntur, et al., 2024)	NSL-KDD, CIC-IDS2017, RoEduNet-SIMARGL2021	Random Forest, MLP, SVM, KNN, AdaBoost, DNN, LightBGM	SHAP, LIME	Penelitian mengusulkan *framework* untuk mengevaluasi metode XAI pada sistem NIDS menggunakan berbagai model *machine learning* dan *deep learning*. Hasil menunjukkan bahwa SHAP dan LIME memiliki karakteristik interpretasi yang berbeda pada setiap model sehingga pemilihannya perlu disesuaikan dengan kebutuhan analisis.
(Bibers & Abdallah, 2025)	CIC-IDS2017	Ranfom Forest, Logistic Regression, KNN, AdaBoost	SHAP, LOCO, PFI, DALEX, ProfWeight	Penelitian mengusulkan metode seleksi fitur berbasis *frequency-based ensemble* yang menggabungkan hasil beberapa metode XAI. Hasil penelitian menunjukkan bahwa pendekatan tersebut mampu menghasilkan himpunan fitur yang lebih representatif sehingga performa klasifikasi tetap tinggi dengan jumlah fitur yang lebih sedikit
(Hooshmand et al., 2024)	NSL-KDD, UNSW-NB15	XGBoost, Random Forest, MLP, KNN, AdaBoost, LightGBM, Stacking, Voting	SHAP	Penelitian mengembangkan model NIDS berbasis XGBoost yang didukung seleksi fitur dan penanganan ketidakseimbangan data, kemudian menginterpretasikan hasil prediksi menggunakan SHAP. Hasil penelitian menunjukkan bahwa model memperoleh performa klasifikasi yang tinggi serta mampu menjelaskan kontribusi setiap fitur terhadap keputusan model.
(Wakili & Bakkali, 2025)	CiCIoT2023	Balanced Random Forest, Extra Trees, XGBoost, CatBoost, LightGBM	SHAP	Penelitian mengintegrasikan *hybrid feature selection, ensemble learning*, dan SHAP dalam suatu *framework* IDS berbasis IoT. Hasil penelitian menunjukkan bahwa *framework* mampu meningkatkan performa deteksi sekaligus memberikan interpretasi terhadap fitur-fitur yang berkontribusi pada proses klasifikasi.
(Farrukh et al., 2025)	CIC-IoT2023, CIC-IDS2017, UNSW-NB15	Heterogeneous Graph Neural Network (HGNN)	Integrated Gradients	Penelitian mengmbangkan *framework* yang mengombinasikan HGNN, IG untuk mendeteksi serta menjelaskan serangan jaringan. Hasil penelitian menunjukkan

				bahwa pendekatan tersebut mampu menghasilkan performa deteksi yang baik disertai penjelasan dalam bahasa alami yang mudah dipahami
(Sadhwani et al., 2025)	NSL-KDD, UNSW-NB15, TON-IoT	CNN, LSTM, Bi-LSTM	SHAP	Penelitian membandingkan beberapa model *deep learning* pada berbagai dataset NIDS dan menerapkan SHAP untuk mengidentifikasi fitur yang paling berpengaruh terhadap hasil klasifikasi. Hasil penelitian menunjukkan bahwa model CNN memberikan performa terbaik dengan interpretasi yang lebih transparan melalui analisis SHAP.
(Gu et al., 2024)	KDDCup99, NSL-KDD, UNSW-NB15, HIKARI-2021	Soft Prototype-based autonomous Fuzzy Inference System (SPAFIS)	Model bersifat intrinsically interpretable	Penelitian mengembangkan sistem deteksi intrusi berbasis *fuzzy inference* yang menghasilkan aturan IF-THEN sebagai dasar pengambilan keputusan. Hasil penelitian menunjukkan bahwa model mampu mencapai performa deteksi yang kompetitif sekaligus mempertahankan interpretabilitas secara intrinsik
(Arreche, Bibers, et al., 2024)	CIC-IDS2017, NSL-KDD, RoEduNet-SIMARGL2021	Random Forest, MLP, SVM, LightGBM, XGBoost, Bagging, Boosting, Stacking, Voting	XAI-based feature selection	Penelitian mengusulkan *framework ensemble* dua tingkat yang menggabungkan seleksi fitur berbasis XAI dengan berbagai algoritma *machine learning*. Hasil penelitian menunjukkan bahwa pendekatan tersebut mampu meningkatkan performa deteksi intrusi dibandingkan penggunaan model tunggal pada beberapa dataset IDS.

Berdasarkan Tabel 2.6. penelitian-penelitian yang dikaji memiliki beberapa persamaan. Sebagian besar penelitian memanfaatkan algoritma *machine learning* maupun *deep learning* untuk meningkatkan kemampuan IDS dalam mengidentifikasi berbagai jenis serangan jaringan. Selain itu, sebagian besar penelitian menggunakan dataset benchmark IDS, seperti CIC-IDS2017, NSL-KDD, dan UNSW-NB15, sehingga hasil penelitian dapat dibandingkan secara objektif. Penelitian-penelitian tersebut juga menunjukkan bahwa penerapan XAI

semakin banyak digunakan untuk meningkatkan transparansi model, dengan SHAP menjadi salah satu metoded yang paling sering diterapkan dalam menjelaskan hasil prediksi model.

Di sisi lain, penelitian-penelitian tersebut juga menunjukkan beberapa perbedaan. Perbedaan tersebut terlihat pada dataset yang digunakan, antara lain CIC-IDS2017, CICIoT2023, NSL-KDD, UNSW-NB15, TON-IoT, dan HIKARI-2021. Selain itu, algoritma yang digunakan juga bervariasi, mulai dari algoritma *machine learning* seperti Random Forest, Logistic Regression, KNN, dan XGBoost, hingga algoritma *deep learning* seperti Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bi-Long Short Term Memory (Bi-LSTM), dan Heterogeneous Graph Neural Network (HGNN). Variasi juga terlihat pada metode XAI yang diterapkan. Selain SHAP dan LIME, beberapa penelitian memanfaatkan metode interpretabilitas lain, seperti Integrated Gradients, LOCO, Permutation Feature Importance (PFI), DALEX, dan ProfWeight. Perbedaan tersebut menunjukkam bahwa setiap penelitian memili fokus yang berbeda, baik dalam pengembangan model, seleski fitur maupun teknik interpretasi yang digunakan.

Berdasarkan kajian tersebut, penelitian ini memiliki perbedaan dibandingkan penelitian-penelitian sebelumnya. Penelitian ini menkombinasikan penggunaan dataset CIC-IDS2017 yang telah melalui tahapan *preprocessing* menjadi enam kelas utama dengan tiga algoritma klasifikasi, yaitu Random Forest, XGBoost, dan MLP. Selain itu, penelitian ini menerapkan SHAP sebagai metode utama dengan menganalisis interpretasi model pada tingkat global maupun lokal, serta menggunakan LIME sebagai pembanding interpretasi lokal. Dengan demikian, penelitian ini tidak hanya berfokus pada evaluasi performa klasifikasi, tetapi juga memberikan analisis yang lebih komprehensif mengenai tranparansi proses pengambilan keputusan model melalui pendekatan XAI.

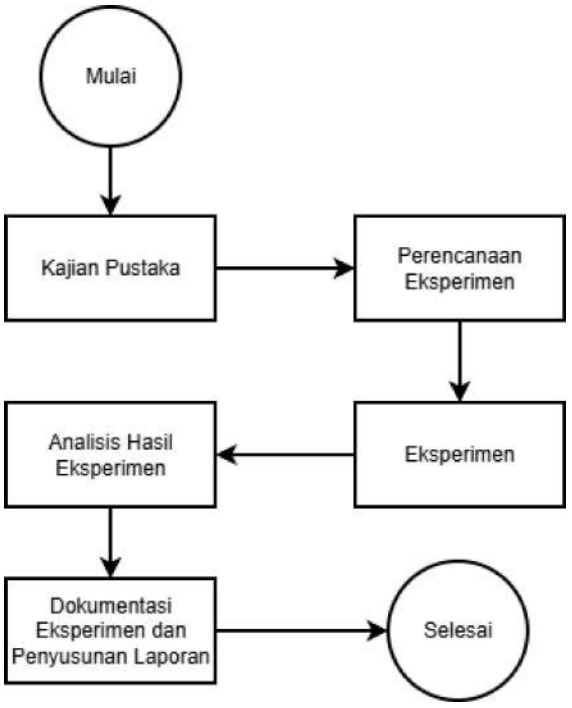
BAB III
METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan metodologi yang digunakan dalam penelitian, mulai dari alur penelitian, penggunaan dataset, proses *preprocessing*, pelatihan model *machine learning*, evaluasi model, hingga proses interpretasi hasil menggunakan metode XAI. Setiap tahapan dirancang secara sistematis untuk mendukung tujuan penelitian dan menghasilkan model IDS yang tidak hanya memiliki performa klasifikasi yang baik tetapi juga dapat diinterpretasikan.

3.1 Alur Pengerjaan Tugas Akhir

Penelitian ini dilaksanakan melalui beberapa tahapan yang disusun secara sistematis untuk mencapai tujuan penelitian. Tahapan tersebut dimulai dari kajian pustaka, dilanjutkan dengan perencanaan eksperimen, pelaksanaan eksperimen, analisis hasil eksperimen, serta dokumentasi eksperimen dan penyusunan laporan. Setiap tahapan saling berkaitan sehingga membentuk alur penelitian yang terstruktur.

Alur pengerjaan tugas akhir pada penelitian ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Alur Pengerjaan Tugas Akhir

Berdasarkan Gambar 3.1, penelitian diawali dengan tahap kajian pustaka untuk mempelajari konsep-konsep yang berkaitan dengan IDS, *machine learning*, XAI, serta dataset

CIC-IDS2017 yang digunakan dalam penelitian. Selanjutnya dilakukan perencanaan eksperimen yang meliputi penyusunan metode penelitian, penentuan dataset, algoritma *machine learning*, metode evaluasi serta metode *explainability* yang akan diterapkan.

Tahap berikutnya adalah pelaksanaan eksperimen, yang mencakup proses persiapan data, pelatihan model, evaluasi model, serta analisis menggunakan metode SHAP dan LIME. Hasil dari eksperimen kemudian dianalisis untuk mengetahui performa model dan interpretasi terhadap hasil klasifikasi yang diperoleh. Tahap terakhir adalah dokumentasi eksperimen dan penyusunan laporan sebagai bentuk penyajian hasil penelitian secara sistematis.

3.2 Uraian Pengerjaan Tugas Akhir

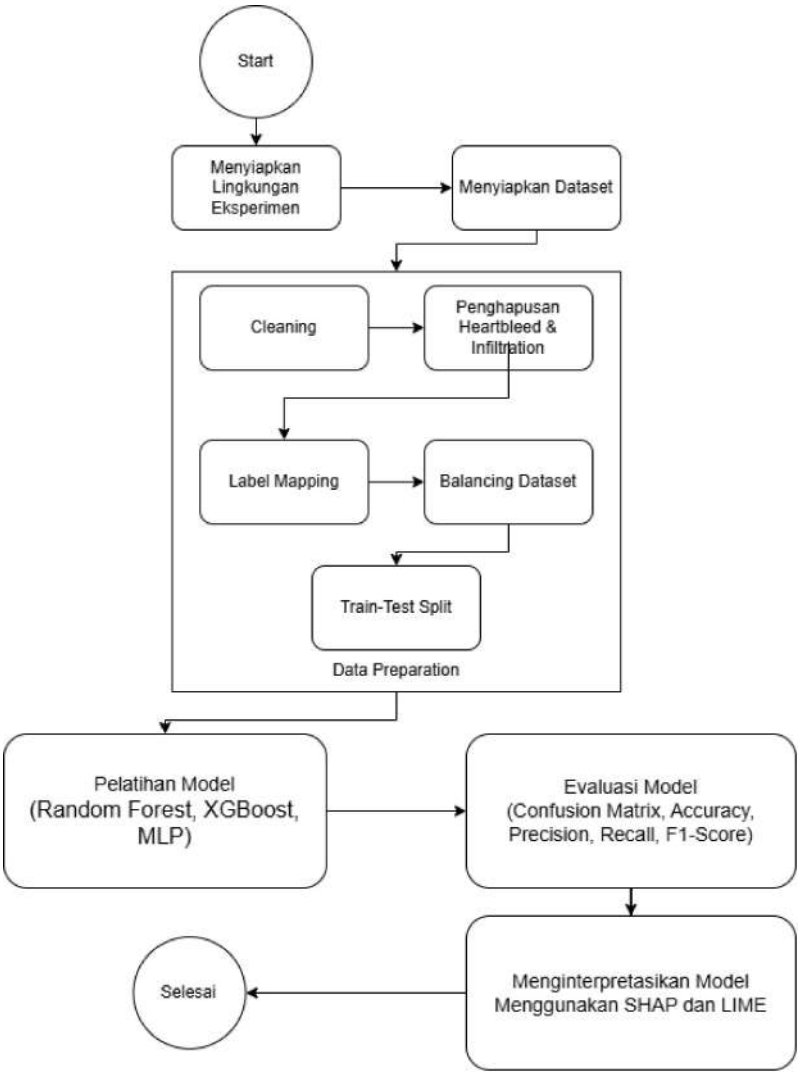
3.2.1 Kajian Pustaka

Kajian pustaka merupakan tahap awal yang dilakukan dalam penelitian sebagai dasar untuk memahami permasalahan serta menentukan pendekatan yang akan digunakan. Pada tahap ini dilakukan penelaahan terhadap berbagai sumber ilmiah, seperti jurnal, prosiding, dan artikel ilmiah yang berkaitan dengan *Intrusion Detection System* (IDS), *machine learning*, *Explainable Artificial Intelligence* (XAI), serta metode evaluasi model klasifikasi.

Selain itu, kajian pustaka juga dilakukan untuk mempelajari karakteristik dataset CIC-IDS2017, algoritma *random forest*, *eXtreme Gradient Boosting* (XGBoost), *Multi-Layer Perceptron* (MLP), serta metode interpretabilitas SHAP dan LIME yang digunakan dalam penelitian. Hasil kajian pustaka menjadi dasar dalam menyusun metodologi penelitian, menentukan tahapan eksperimen, serta melakukan analisis terhadap hasil penelitian.

3.2.2 Perencanaan Eksperimen

Pada tahap perencanaan eksperimen, dilakukan penyusunan rancangan eksperimen yang akan digunakan sebagai pedoman dalam pelaksanaan penelitian. Rancangan ini disusun berdasarkan hasil kajian pustaka yang telah dilakukan sebelumnya sehingga setiap tahapan eksperimen dapat berjalan secara sistematis dan sesuai dengan tujuan penelitian. Perencanaan eksperimen pada penelitian ini meliputi penentuan lingkungan eksperimen, dataset yang digunakan tahapan data preparation, pelatihan model *machine learning*, evaluasi model, serta penerapan XAI. Rancangan eksperimen tersebut ditunjukkan pada Gambar 3.2.



Gambar 3.2 Perencanaan Eksperimen

Berdasarkan Gambar 3.2, penelitian diawali dengan penentuan lingkungan eksperimen yang digunakan untuk menjalankan seluruh proses penelitian. Selanjutnya digunakan dataset CIC-IDS2017 yang kemudian melalui tahap *data preparation* yang meliputi *cleaning*, penghapusan kelas *Heartbleed* dan *Infiltration*, *label mapping*, *balancing dataset*, serta *train-test split.* Dataset yang telah dipersiapan kemudian digunakan pada tahap pelatihan model menggunakan algoritma *Random Forest*, XGBoost, dan MLP. Selanjutnya dilakukan evaluasi performa model menggunakan *Confusion Matrix*, *Accuracy*, *Precision*, *Recall*, dan F1-*Score*. Tahap terakhir adalah penerapan metode XAI menggunakan SHAP dan LIME untuk memberikan interpretasi terhadap hasil prediksi model.

Tahapan-tahapan tersebut kemudian diimplementasikan pada proses eksperimen yang dijelaskan secara rinci pada subbab 3.2.3.

3.2.3 Eksperimen

a. **Menyiapkan Lingkungan Eksperimen**

Tahap eksperimen pada penelitian ini dilakukan menggunakan perangkat keras berupa laptop dengan spesifikasi yang ditunjukkan pada Tabel 3.1. Perangkat keras tersebut digunakan untuk menjalankan seluruh proses penelitian, mulai dari persiapan data, pelatihan model *machine learning*, evaluasi model, hingga penerapan metode XAI.

Tabel 3.1 Spesifikasi Perangkat Keras

Spesifikasi	
Laptop	Lenovo IdeaPad Gaming 3 15ACH6
Prosesor	AMD Ryzen 5 5600h
GPU	NVIDIA GeForce RTX 3060 6GB
Penyimpanan	2x 512 GB SSD PCIe
RAM	2x 8 GB DDR4-3200
Sistem Operasi	Windows 11
Tipe Sistem	64-bit

Seluruh proses implementasi dilakukan menggunakan *Visual Studio Code* sebagai *Integrated Development Environment* (IDE) dengan bahasa pemrograman Python versi 3.13.5 dalam format *Jupyter Notebook* (.ipynb). Beberapa *library* yang digunakan dalam penelitian ini meliputi *pandas* dan *NumPy* untuk manipulasi data, *Matplotlib* dan *Seaborn* untuk visualisasi data, *scikit-learn* sebagai implementasi algoritma *machine learning* dan evaluasi model, XGBoost untuk membangun model *Extreme Gradient Boostng*, serta SHAP dan LIME sebagai metode XAI. Rincian lingkungan eksperimen yang digunakan pada penelitian ini ditunjukkan pada Tabel 3.2

Tabel 3.2 Lingkungan Eksperimen

Komponen	**Keterangan**
IDE	Visual Studio Code
Bahasa Pemrograman	Python 3.13.5
Format Notebook	Jupyter Notebook (.ipynb)
Library	Pandas (data *preprocessing* dan *random undersampling*), NumPy,

	Matplotlib, Seaborn, scikit-learn, XGBoost, SHAP, LIME

b. **Menyiapkan Dataset**

Dataset yang digunakan pada penelitian ini adalah CIC-IDS2017 yang dikembangkan oleh *Canadian Institute for Cybersecurity* (CIC) sebagai salah satu *benchmark dataset* untuk penelitian IDS. Dataset ini dipilih karena merepresentasikan lalu lintas jaringan yang menyerupai kondisi nyata serta mencakup berbagai jenis serangan jaringan yang umum digunakan dalam penelitian deteksi intrusi.

Dataset CIC-IDS2017 digunakan sebagai sumber data utama proses pelatihan dan pengujian model *machine learning*. Sebelum digunakan pada tahap eksperimen, dataset terlebih dahulu melalui serangkaian tahapan *preprocessing* untuk menyesuaikan data dengan kebutuhan penelitian. Tahapan tersebut meliputi penggabungan dataset, pembersihan data (*balancing dataset*), pembagian data latih dan data uji (*train-test-split*), serta *feature scaling*. Penjelasan mengenai setiap tahapan *preprocessing* tersebut disajikan secara rinci pada Subbab Data Preparation. Karakteristik dataset yang digunakan pada penelitian ini disajikan pada Tabel 3.3. Dataset dengan karakteristik tersebut selanjutnya digunakan sebagai masukan pada seluruh tahapan eksperimen yang meliputi proses *data preparation*, pelatihan model, evaluasi model, serta interpretasi hasil menggunakan metode XAI.

Tabel 3.3 Karakterisik Dataset Setelah Melalui tahap Preprocessing

Karakteristik	Keterangan
Nama Dataset	CIC-IDS2017
Jumlah Data	11.400
Jumlah Fitur	78
Jumlah Kelas	6
Kelas	Benign, DoS, PortScan, BruteForce, Bot, WebAttack
Dataset Hasil	CIC_IDS2017_balanced.csv

c. **Data Preparation**

Tahap *data preparation* bertujuan mempersiapkan dataset agar memiliki kualitas yang baik sebelum digunakan pada proses pelatihan model *machine learning*. Tahapan ini dilakukan secara bertahap mulai dari penggabungan dataset, pembersihan data, penyesuaian label, penyeimbangan data, validasi dataset, hingga pembagian data latih dan data uji.

Dataset CIC-IDS2017 terdiri atas beberapa berkas yang merepresentasikan aktivitas lalu lintas jaringan pada hari dan skenario serangan yang berbeda. Seluruh berkas tersebut terlebih dahulu digabungkan menjadi satu dataset menggunakan pustaka *Pandas* sehingga seluruh data dapat diproses secara terpadu. Setelah proses penggabungan selesai, dilakukan proses *cleaning* untuk meningkatkan kualitas data. Tahap ini meliputi penghapus data yang memiliki nilai kosong (*missing values*) serta penghapus kelas Heartbleed yang berjumlah 11 dan infiltration berjumlah 36, jumlah sampel tersebut sangat sedikit sehingga berpotensi memengaruhi proses pelatihan model serta menghasilkan distribusi kelas yang tidak representatif.

Setelah proses *cleaning*, dilakukan *label mapping* untuk mengelompokkan beberapa subkelas serangan ke dalam kategori serangan yang lebih umum. Proses ini bertujuan menyederhanakan proses klasifikasi sehingga modedl hanya mempelajari enam kelas utama sesuai dengan tujuan penelitian. Pemetaan label yang dilakukan pada Tabel 3.4. Berdasarkan Tabel 3.4, beberapa subkelas serangan digabungkan ke dalam kategori serangan yang lebih umum sehingga jumlah kelas berkurang dari 13 kelas menjadi 6 kelas utama, yaitu Benign, DoS, BruteForce, PortScan, Bot, dan WebAttack. Distribusi jumlah sampel setelah proses *label mapping* disajikan pada Tabel 3.4.

Tabel 3.4 Pemetaan Label Dataset CIC-IDS2017

Label Asli	**Jumlah Sampel**	**Label Baru**	**Jumlah Sampel**
BENIGN	2.273.097	Benign	2.273.097
DDoS	128.027	DoS	379.739
DoS Hulk	231.073	DoS	
DoS GoldenEye	10.293	DoS	
DoS slowloris	5.796	DoS	
DoS Slowhttptest	5.499	DoS	
FTP-Patator	7.938	BruteForce	13.835
SSH-Patator	5.897	BruteForce	
PortScan	158.930	PortScan	158.930
Bot	1.966	Bot	1.966
Web Attack BruteForce	1.507	WebAttack	2.180
Web Attack XSS	652	WebAttack	
Web Attack SQL Injection	21	WebAttack	

Setelah proses *label mapping*, dataset masih memiliki distribusi kelas yang tidak seimbang, di mana kelas Benign memiliki jumlah data yang jauh lebih besar dibandingkan

kelas serangan lainnya. Ketidakseimbangan tersebut berpotensi menyebabkan model lebih cenderung mempelajari kelas mayoritas sehingga menurunkan kemampuan kelas klasifikasi pada kelas minoritas. Oleh karena itu, dilakukan proses *balancing dataset* menggunakan *Random Undersampling*. Metode ini dipilih karena mampu menghasilkan distribusi jumlah sampel yang seimbang dengan cara mengurangi jumlah data pada kelas mayoritas secara acak hingga seluruh kelas memiliki jumlah sampel yang sama. Implementasi metode *Random Undersampling* dilakukan menggunakan fungsi *groupby*() dan *sample*() pada pustaka *Pandas*, denga mengambil sebanyak 1.900 sampel dari setiap kelas menggunakan nilai *random_state* = 42.

Setelah proses *balancing* selesai, diperoleh dataset yang terdiri dari 11.400 data, yaitu masing masing 1.900 data untuk enam kelas utama. Dataset hasil *balancing* kemudian disimpan dalam berkas CIC_IDS2017_balanced.csv dan digunakan pada seluruh tahapan eksperimen berikutnya. Dataset hasil *balancing* selanjutnya dimuat ke dalam lingkungan eksperimen untuk dilakukan proses validasi. Tahap ini meliputi pembersihan nama kolom (*clean column names*), pengubahan nilai tak hingga (*inifinity values*) menjadi nilai kosong (*missing values*), serta penghapusan data yang masih mengandung nilai kosong. Selain itu, dilakukan pemeriksaan distribusi kelas untuk memastikan bahwa setiap kelas memiliki jumlah sampel yang sama sebelum proses pelatihan model dilakukan.

Tahap berikutnya adalah memisahkan dataset menjadi fitur (*feature*) dan label (*target*). Label kategori kemudian diubah ke dalam bentuk numerik menggunakan *Label Encoding* agar dapat diproses oleh algoritma *machine learning*. Selanjutnya, dataset dibagi menjadi data latih dan data uji menggunakan metode *train-test split* dengan proporsi 80% data latih dan 20% dataa uji, menggunakan nilai *random_state* = 42 serta teknik *stratified sampling* untuk mempertahankan proporsi setiap kelas.

Tahap terakhir adalah melakukan *feature scaling* menggunakan metode *StandardScaler*. Proses ini hanya diterapkan pada model MLP karena model berbasis jaringan saraf sensitif terhadap perbedaan skala fitur. Sementara itu, model Random Forest dan XGBoost tidak memerlukan *feature scaling* karena keduanya merupakan algoritma berbasis *decision tree* yang tidak dipengaruhi oleh skala nilai fitur. Hasil dari tahapan ini berupa data latih dan data uji yang siap digunakan pada proses pelatihan model.

d. **Pelatihan Model**

Tahap pelatihan model dilakukan untuk membangun model klasifikasi yang mampu mengenali lalu lintas jaringan normal maupun berbagai jenis serangan berdasarkan dataset yang telah melalui proses *data preparation*. Pada penelitian ini, digunakan tiga algoritma *machine learning*, yaitu Random Forest, XGBoost, dan MLP. Ketiga algoritma tersebut digunakan untuk membandingkan performa klasifikasi pada dataset CIC-IDS2017 yang telah melalui proses *balancing*.

Model pertama yang dilatih adalah Random Forest menggunakan data latih (*training set*) hasil proses *train-test split*. Pelatihan dilakukan menggunakan parameter *random_state* sebesar 42 untuk menjaga konsistensi hasil eksperimen. Selanjutnya dilakukan pelatihan menggunakan algoritma XGBoost dengan data latih yang sama. Pada penelitian ini digunakan parameter *random_state* sebesar 42 dan *evaluation metric* berupa mlogloss untuk mendukung proses pelatihan model klasifikasi multikelas.

Model terakhir yang dilatih adalah MLP. Berbeda dengan dua model sebelumnya, MLP menggunakan data yang telah melalui proses *feature scaling* menggunakan StandardScaler. Model dilatih menggunakan satu *hidden layer* yang terdiri atas 100 neuron, nilai max_iter sebesar 500, serta *random_state* sebesar 42 untuk memperoleh hasil pelatihan yang konsisten. Setelah seluruh model selesai dilatih, masing-masing model digunakan untuk melakukan prediksi terhadap data uji (*testing set*) yang selanjutnya dievaluasi menggunakan beberapa metrik evaluasi guna membandingkan performa klasifikasi setiap model.

e. **Evaluasi Model**

Tahap evaluasi model dilakukan untuk mengukur performa klasifikasi dari setiap algoritma *machine learning* yang digunakan pada penelitian ini. Setelah proses pelatihan selesai, masing-masing model digunakan untuk melakukan prediksi terhadap data uji (*testing set*). Hasil prediksi tersebut kemudian dibandingkan dengan label sebenarnya untuk mengetahui kemampuan model dalam mengklasifikasikan setiap kelas serangan jaringan.

Evaluasi performa model dilakukan menggunakan *Confusion Matrix*, *Accuracy*, *Precision*, *Recall*, F1-*Score*. *Confusion Matrix* digunakan untuk menggambarkan hasil klasifikasi setiap kelas secara rinci, sedangkan nilai *Accuracy*, *Precision*, *Recall*, dan F1-*Score* digunakan sebagai indikator kuantitatif untuk membandingkan performa ketiga model yang digunakan pada penelitian ini. Hasil evaluasi dari setiap model kemudian digunakan sebagai dasar untuk membandingkan performa klasifikasi antara algoritma Random Forest, XGBoost,

dan MLP. Model dengan performa terbaik kemudian dianalisis lebih lanjut menggunakan metode XAI.

Selain pengujian menggunakan *testing set* sebesar 20%, penelitian ini juga melakukan pengujian *robustness* model terhadap sisa data CIC-IDS2017 yang tidak diikutsertakan dalam proses pelatihan maupun *balancing*. Dataset sisa ini berjumlah sekitar 2,8 juta baris data dengan distribusi kelas yang aslinya sangat tidak seimbang (didominasi oleh kelas *Benign*), yang secara akurat merepresentasikan kondisi lalu lintas jaringan di dunia nyata (*real-world scenario*). Pengujian tambahan ini diterapkan pada model dengan performa terbaik (XGBoost) untuk memvalidasi seberapa baik model dapat mempertahankan nilai *recall* dalam mendeteksi serangan, serta mengamati dampaknya terhadap *false positive* pada lingkungan operasional yang sebenarnya.

f. **Menginterpretasikan Model Menggunakan SHAP dan LIME**

Tahap terakhir pada penelitian ini adalah penerapan metode XAI untuk memberikan interpretasi terhadap hasil prediksi model *machine learning*. Metode XAI digunakan untuk menjelaskan kontribusi setiap fitur terhadap keputusan yang dihasilkan oleh model sehingga proses klasifikasi dapat dipahami dengan lebih baik.

Pada penelitian ini digunakan dua metode XAI, yaitu SHAP dan LIME. SHAP dipilih sebagai metode utama untuk menganalisis kontribusi setiap fitur baik secara global maupun lokal terhadap hasil prediksi model. Analisis dilakukan menggunakan nilai SHAP untuk mengidentifikasi fitur-fitur yang memiliki pengaruh terbesar terhadap proses klasifikasi pada masing-masing kelas serangan.

Selain SHAP, penelitian ini juga menerapkan LIME sebagai metode pembanding untuk memberikan interpretasi lokal terhadap hasil prediksi model. LIME digunakan untuk menjelaskan keputusan model terhadap suatu sampel tertentu sehingga hasil interpretasi yang diperoleh dapat dibandingkan dengan penjelasan yang dihasilkan oleh SHAP. Hasil interpretasi menggunakan SHAP dan LIME kemudian dianalisis untuk mengetahui karakteristik penjelasan yang dihasilkan oleh masing-masing metode serta mendukung proses interpretasi terhadap model *machine learning* yang digunakan pada penelitian ini.

3.2.4 Analisis Hasil Eksperimen

Tahap analisis hasil eksperimen dilakukan untuk menginterpretasikan hasil evaluasi dari setiap model *machine learning* yang telah dilatih. Analisis dilakukan dengan membandingkan

performa klasifikasi dari algoritma Random Forest, XGBoost, dan MLP berdasarkan metrik evaluasi yang telah ditentukan. Selain membandingkan performa klasifikasi, analisis juga dilakukan terhadap hasil interpretasi menggunakan metode XAI. Pada tahap ini, metode SHAP digunakan untuk mengidentifikasi fitur-fitur yang memberikan kontribusi terbesar terhadap proses klasifikasi baik secara global maupun lokal. Selanjutnya, hasil interpretasi menggunakan SHAP dibandingkan dengan hasil interpretasi menggunakan LIME untuk mengetahui karakteristik penjelasan yang dihasilkan oleh kedua metode tersebut. Hasil analisis tersebut digunakan untuk mengevaluasi kemampuan masing-masing model dalam mendeteksi serangan jaringan serta memberikan pemahaman mengenai faktor-faktor yang memengaruhi proses pengambilan keputusan model.

3.2.5 Dokumentasi Eksperimen dan Penyusunan Laporan

Tahap terakhir penelitian ini adalah dokumentasi eksperimen dan penyusunan laporan penelitian. Seluruh proses yang telah dilakukan, mulai dari persiapan data, pelatihan model, evaluasi, hingga interpretasi menggunakan metode XAI, didokumentasikan secara sistematis sebagai bentuk pelaporan hasil penelitian. Selanjutnya, hasil eksperimen, analisis, dan pembahasan disusun ke dalam laporan skripsi sesuai dengan pedoman penulisan tugas akhir Program Studi Informatika Universitas Islam Indonesia. Penyusunan laporan dilakukan untuk mendokumentasikan seluruh proses penelitian serta menyajikan hasil penelitian secara terstruktur sehingga dapat dipahami dan dipertanggungjawabkan secara ilmiah.

BAB IV
HASIL DAN PEMBAHASAN

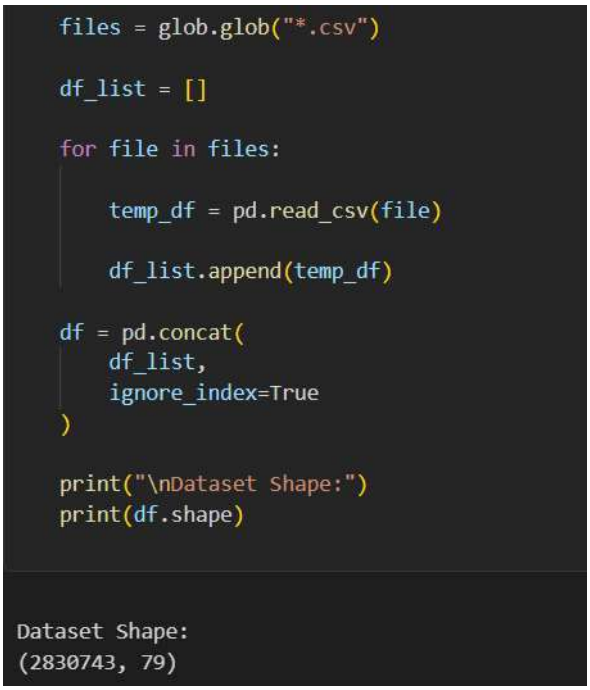
Bab ini membahas hasil penelitian yang diperoleh berdasarkan tahapan metodologi yang telah dijelaskan pada bab sebelumnya. Pembahasan diawali dengan proses *data preparation* sebagai tahap persiapan dataset sebelum proses pelatihan model. Selanjutnya disajikan hasil pemodelan menggunakan algoritma Random Forest, XGBoost, dan MLP, beserta evaluasi kinerja masing-masing model. Selain itu, dilakukan analisis menggunakan metode SHAP untuk memberikan interpretasi terhadap hasil prediksi model dan mengidentifikasi fitur-fitur yang berkontribusi dalam proses klasifikasi serangan jaringan.

4.1 Data Preparation

Tahap *data preparation* dilakukan untuk mempersiapkan dataset sebelum digunakan pada proses pelatihan model *machine learning*. Pada tahap ini dilakukan proses, yaitu *cleaning data*, penghapusan kelas dengan jumlah data yang sangat sedikit, *label mapping*, *balancing dataset*, serta persiapan data sebelum pelatihan model yang meliputi pemisahan fitur dan label, *label encoding*, pembagian data latih dan data uji, serta *feature scaling*. Tahapan-tahapan tersebut bertujuan untuk menghasilkan dataset yang bersih, seimbang, dan siap digunakan pada proses pembangunan model.

4.1.1 Cleaning Data

Proses *cleaning* diawali dengan menggabungkan delapan file CSV penyusun dataset CIC-IDS2017 menggunakan pustaka *pandas* sehingga seluruh data berada dalam satu dataset. Hasil penggabungan tersebut menghasilkan dataset awal yang terdiri dari 2.830.743 data dengan 79 atribut, seperti yang tertera pada Gambar 4.1.



Gambar 4.1 Penggabungan Dataset

Setelah proses penggabungan selesai, dilakukan pemeriksaan terhadap keberadaan *missing value* pada dataset. Pemeriksaan ini bertujuan untuk memastikan tidak terdapat data yang tidak lengkap yang dapat memengaruhi proses pelatihan model. Berdasarkan hasil pemeriksaan, ditemukan sebanyak 1.358 *missing value* pada dataset.

Selanjutnya dilakukan proses cleaning menggunakan fungsi dropna() untuk menghapus seluruh baris yang mengandung missing value. Setelah proses tersebut dilakukan, jumlah data berkurang menjadi 2.829.385 data dengan jumlah atribut tetap sebanyak 79 atribut. Dengan demikian, dataset telah terbebas dari missing value dan siap digunakan pada tahapan berikutnya. Berdasarkan hasil cleaning, seluruh data yang mengandung missing value berhasil dihapus sehingga dataset menjadi lebih bersih dan siap digunakan pada proses data preparation berikutnya. Setelah itu juga dilakukan penghapus Infinity Value yang berjumlah 1.509 data, sehingga sekarang siap dilakukan penghapus kelas dan label mapping. Proses ini bisa dilihat pada Gambar 4.2



Gambar 4.2 Pemeriksaan dan Penghapusan Missing Value dan Penghapusan Infinity Value

4.1.2 Penghapusan Kelas dan Label Mapping

Setelah proses *cleaning* selesai dilakukan, tahap berikutnya adalah menghapus kelas Heartbleed dan Infiltration dari dataset. Penghapusan kedua kelas tersebut dilakukan karena jumlah data yang sangat sedikit, yaitu masing-masing sebanyak 11 dan 36 data. Jumlah data yang sangat terbatas berpotensi menyebabkan distribusi kelas menjadi tidak representatif sehingga dapat memengaruhi proses pelatihan model klasifikasi.

```
df = df[
    ~df[" Label"].isin([
        "Heartbleed",
        "Infiltration"
    ])
```

Gambar 4.3 Kode penghapusan kelas Heartbleed dan Infiltration

Setelah proses penghapusan dilakukan, dataset tidak lagi mengandung kelas Heartbleed dan Infiltration sehingga hanya menyisakan kelas-kelas serangan yang digunakan pada penelitian ini. Selanjutnya dilakukan proses *label mapping* untuk mengelompokkan beberapa jenis serangan yang memiliki karakteristik serupa ke dalam enam kelas utama, yaitu Benign, DoS, BruteForce, PortScan, Bot, dan WebAttack. Proses ini bertujuan untuk menyederhanakan label serangan sehingga sesuai dengan skenario klasifikasi multikelas yang digunakan pada

penelitian. Hasil *label mapping* menghasilkan enam kelas utama yang selanjutnya digunakan pada proses *balancing dataset* dan pembangunan model klasifikasi pada tahap berikutnya.

```
    attack_mapping = {

    "BENIGN": "Benign",

    # DoS
    "DDoS": "DoS",
    "DoS Hulk": "DoS",
    "DoS GoldenEye": "DoS",
    "DoS slowloris": "DoS",
    "DoS Slowhttptest": "DoS",

    # Brute Force
    "FTP-Patator": "BruteForce",
    "SSH-Patator": "BruteForce",

    # Port Scan
    "PortScan": "PortScan",

    # Bot
    "Bot": "Bot",

    # Web Attack
    "Web Attack - Brute Force": "WebAttack",
    "Web Attack - XSS": "WebAttack",
    "Web Attack - Sql Injection": "WebAttack"
}

    df[" Label"] = df[" Label"].map(attack_mapping)
```

Gambar 4.4 Kode program Label Mapping

4.1.3 Balancing Dataset

Setelah proses *label mapping* selesai dilakukan, tahap berikutnya adalah melakukan *balancing dataset*. Proses ini bertujuan untuk menghasilkan distribusi jumlah data yang seimbang pada setiap kelas sehingga dapat mengurangi potensi bias model terhadap kelas yang memiliki jumlah data lebih banyak. Pada penelitian ini, proses *balancing* dilakukan menggunakan metode *random undersampling* dengan mengambil sebanyak 1.900 data dari setiap kelas menggunakan fungsi *sample()* pada pustaka *pandas*.

```
     balanced_df = (
         df.groupby(" Label", group_keys=False)
           .apply(
               lambda x: x.sample(
                   n=1900,
                   random_state=42
               )
           )
           .reset_index(drop=True)
```

```
)
```

Gambar 4.5 Implementasi *Balancing* Dataset

Hasil proses *balancing* menunjukkan bahwa setiap kelas memiliki jumlah data yang sama yaitu sebanyak 1.900 data. Dengan demikian, dataset hasil *balancing* terdiri atas 11.400 data yang terbagi secara merata ke dalam enam kelas, yaitu Benign, Bot, BruteForce, DoS, PortScan, dan WebAttack. Seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Distribusi Dataset setelah *Balancing*

Kelas	Jumlah Data
Benign	1.900
Bot	1.900
BruteForce	1.900
DoS	1.900
PortScan	1.900
WebAttack	1.900
Total	11.400

Dataset hasil *balancing* kemudian disimpan menjadi file CIC_IDS2017_balanced.csv dan digunakan sebagai dataset utama pada proses persiapan data sebelum pelatihan model *machine learning*. Dataset tersebut terdiri atas 79 atribut, dengan 78 atribut fitur dan 1 atribut label. Seluruh data pada dataset hasil *balancing* memiliki nilai yang lengkap (*non-null*) sehingga siap digunakan pada tahap persiapan data sebelum pelatihan model.

4.1.4 Persiapan Data sebelum Pelatihan Model

Dataset hasil *balancing* selanjutnya dimuat kembali untuk proses persiapan data sebelum pelatihan model. Pada tahap ini dilakukan pembersihan nama atribut menggunakan fungsi str.strip() untuk menghilangkan spasi yang tidak diperlukan pada nama kolom.

```
df = pd.read_csv("CIC_IDS2017_balanced.csv")

df.columns = df.columns.str.strip()
```

Gambar 4.6 Persiapan Dataset Sebelum Pelatihan Model

Selanjutnya dilakukan proses validasi dataset untuk memastikan bahwa data yang digunakan telah memenuhi kebutuhan penelitian. Hasil validasi menunjukkan bahwa dataset

tidak lagi memiliki *missing value* dan terdiri atas enam kelas utama dengan distribusi data yang relatif seimbang.

Tabel 4.2 Hasil Validasi Dataset

Parameter	Nilai
Jumlah data	11.400
Jumlah Atribut	79
Missing Value	0
Jumlah Kelas	6

Setelah proses validasi selesai, dataset dipisahkan menjadi atribut fitur (*feature*) dan atribut target (*target*). Dataset memiliki 78 atribut fitur dan 1 atribut target. Selanjutnya dilakukan proses *label encoding* seperti pada Tabel 4.3 untuk mengubah label kategorikal menjadi representasi numerik agar dapat diproses oleh model *machine learning*.

```
X = df.drop("Label", axis=1)
y = df["Label"]

label_encoder = LabelEncoder()

y_encoded = label_encoder.fit_transform(y)

label_mapping = {
    label: int(code)
    for label, code in zip(
        label_encoder.classes_,
        label_encoder.transform(label_encoder.classes_)
    )
}
```

Gambar 4.7 Pemisahan Fitur, Label, dan *Label Encoding*

Tabel 4.3 Hasil *Label Encoding*

Label	Kode
Benign	0
Bot	1
BruteForce	2
DoS	3
PortScan	4
WebAttack	5

Dataset kemudian dibagi menjadi data latih dan data uji menggunakan perbandingan 80:20 dengan parameter random_state = 42 dan *stratify* untuk mempertahankan proporsi setiap kelas. Hasil pembagian menghasilkan 9.120 data latih dan 2.280 data uji.

```
X_train, X_test, y_train, y_test = train_test_split(
    X,
    y_encoded,
    test_size=0.20,
    random_state=42,
    stratify=y_encoded
)
```

Gambar 4.8 Pembagian Data Latih dan Data Uji

Tahap terakhir pada proses persiapan data adalah melakukan *feature scaling* menggunakan metode StandardScaler, khususnya agar proses pelatihan model MLP dapat berlangsung lebih optimal. Proses ini diterapkan untuk menyeragamkan skala setiap fitur sehingga memiliki rata-rata nol dan simpangan baku satu. Hasil *feature scaling* menghasilkan data latih berukuran (9120, 78) dan data uji berukuran (2280, 78). Dataset tersebut selanjutnya digunakan pada proses pelatihan model *machine learning*.

```
scaler = StandardScaler()

X_train_scaled = pd.DataFrame(
    scaler.fit_transform(X_train),
    columns=X_train.columns
)

X_test_scaled = pd.DataFrame(
    scaler.transform(X_test),
    columns=X_test.columns
)
```

Gambar 4.9 Implementasi *Feature Scaling*

4.2 Pelatihan Model

Setelah proses *data preparation* selesai dilakukan, tahap berikutnya adalah pelatihan model *machine learning*. Pada penelitian ini digunakan tiga algoritma klasifikasi yaitu Random Forest, XGBoost, dan MLP. Setiap model dilatih menggunakan data latih yang telah dipersiapkan pada tahap sebelumnya untuk menghasilkan model klasifikasi yang selanjutnya digunakan pada proses evaluasi dan interpretasi menggunakan metode XAI.

4.2.1 Pelatihan Model Random Forest

Model Random Forest dibangun menggunakan kelas RandomForestClassifier dari pustaka *scikit-learn*. Pada penelitian ini digunakan parameter *random_state* = 42 seperti pada Gambar 4.10 untuk memastikan proses pelatihan dapat direproduksi sehingga menghasilkan keluaran yang konsisten pada setiap percobaan. Setelah proses pelatihan selesai dilakukan, model Random Forest siap digunakan untuk melakukan prediksi terhadap data uji pada tahap evaluasi model.

```
rf_model = RandomForestClassifier(
    random_state=42
)

rf_model.fit(X_train, y_train)
```

Gambar 4.10 Implementasi Pelatihan Model Random Forest

4.2.2 Pelatihan Model XGBoost

Model XGBoost dibangun menggunakan XGBClassifier. Pada penelitian ini, digunakan parameter *random_state* = 42 dan *eval_metric* = "mlogloss" selama proses pelatihan model seperti pada Gambar 4.11. Parameter *eval_metric* digunakan sebagai fungsi evaluasi untuk mengukur performa model pada kasus klasifikasi multikelas. Setelah pelatihan dilakukan, model XGBoost digunakan untuk menghasilkan prediksi terhadap data uji sehingga performanya dapat dibandingkan dengan model lainnya.

```
xgb_model = XGBClassifier(
    random_state=42,
    eval_metric="mlogloss"
)

xgb_model.fit(X_train, y_train)
```

Gambar 4.11 Implementasi Pelatihan Model XGBoost

4.2.3 Pelatihan Model Multi-Layer Perceptron

Model MLP dibangun menggunakan kelas MLPClassifier dengan parameter *hidden_layer_sizes* = (100,), *max_iter* = 500, dan *random_state* = 42. Berbeda dengan Random Forest dan XGBoost, proses pelatihan MLP menggunakan data hasil *feature scaling* karena model MLP lebih sensitif terhadap perbedaan skala antar fitur dibandingkan algoritma berbasis

decision tree seperti Random Forest dan XGBoost. Setelah proses pelatihan selesai dilakukan, model MLP selanjutnya digunakan untuk melakukan prediksi terhadap data uji dan dibandingkan dengan model Random Forest serta XGBoost pada tahap evaluasi.

```
mlp_model = MLPClassifier(
    hidden_layer_sizes=(100,),
    max_iter=500,
    random_state=42
)

mlp_model.fit(
    X_train_scaled,
    y_train
)
```

Gambar 4.12 Implementasi Pelatihan Model Multi-Layer Perceptron

4.3 Evaluasi Model

Setelah proses pelatihan model selesai dilakukan, tahap berikutnya adalah mengevaluasi performa masing-masing model klasifikasi. Evaluasi dilakukan dengan menghasilkan prediksi terhadap data uji, kemudian membandingkannya dengan label sebenarnya (*ground truth*). Pada penelitian ini, evaluasi dilakukan menggunakan *confusion matrix*, serta metrik *accuracy*, *precision*, *recall*, dan F1-*score* untuk mengukur kemampuan model dalam mengklasifikasikan enam kelas serangan pada dataset CIC-IDS2017.

4.3.1 Evaluasi Model Random Forest

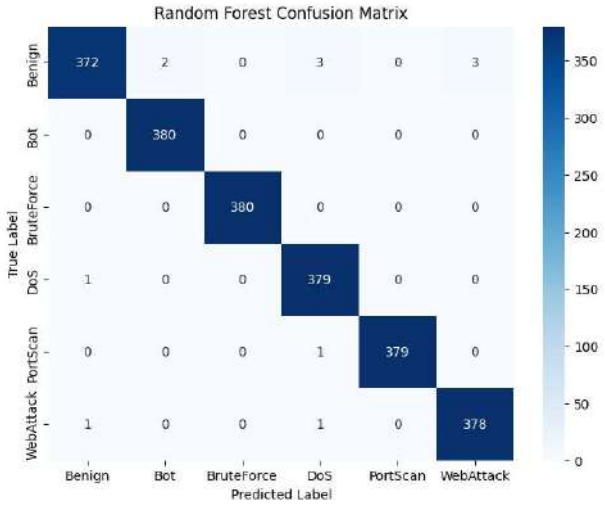
Model Random Forest dievaluasi menggunakan data uji yang telah dipersiapkan pada tahap sebelumnya. Proses evaluasi diawali dengan menghasilkan prediksi terhadap data uji menggunakan fungsi predict() seperti pada gambar Gambar 4.13. Hasil prediksi tersebut kemudian dibandingkan dengan *ground truth* untuk mengukur performa model.

```
y_pred_rf = rf_model.predict(X_test)
```

Gambar 4.13 Implementasi Prediksi Model Random Forest

Hasil prediksi model Random Forest selanjutnya divisualisasikan menggunakan *confusion matrix* seperti yang ditunjukkan pada Gambar 4.14. Berdasarkan *confusion matrix,* sebagian besar data berhasil diklasifikasikan ke kelas yang sesuai. Kesalahan klasifikasi hanya terjadi pada beberapa sampel, yaitu dua data Benign yang diprediksi sebagai Bot, tiga data Benign yang diprediksi sebagai WebAttack, tiga data Benign yang diprediksi sebagai DoS, satu

data DoS yang diprediksi sebagai Benign, serta satu data WebAttack yang diprediksi sebagai Benign, dan yang terakhir 1 PortScan dan WebAttack yang diprediksi sebagai DoS. Sementara itu, seluruh data pada kelas BruteForce dan Bot berhasil diklasifikasikan dengan benar tanpa kesalahan.



Gambar 4.14 Confusion Matrix Model Random Forest

Tabel 4.4 Hasil Evaluasi Model Random Forest

Metrik	Nilai
Accuracy	0,9947
Precision	0,9948
Recall	0,9947
F1-Score	0,9947

Berdasarkan Hasil Evaluasi pada Tabel 4.4, model Random Forest memperoleh nilai *accuracy* sebesar 0,9947 dengan nilai *precision* dengan nilai 0,9948, *recall* dan F1-*Score*, memiliki nilai yang sama yaitu 0,9947. Nilai tersebut menunjukkan bahwa model memiliki kemampuan klasifikasi yang sangat baik dalam membedakan enam kelas serangan pada dataset CIC-IDS2017. Hasil ini juga didukung oleh *confusion matrix* yang memperlihatkan bahwa sebagian besar data uji berhasil diklasifikasikan dengan benar dan hanya terdapat sedikit kesalahan klasifikasi pada beberapa kelas.

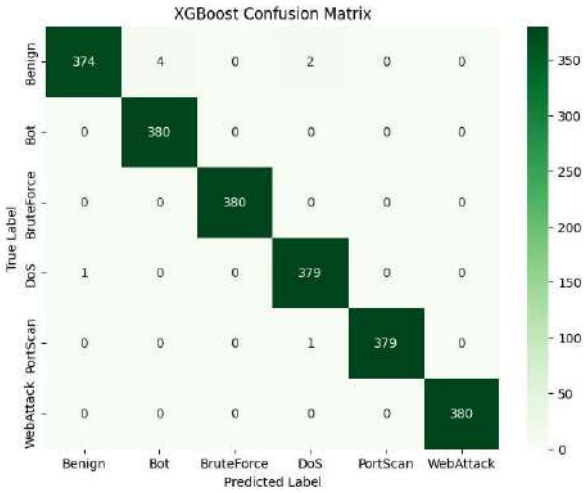
4.3.2 Evaluasi Model XGBoost

Untuk mengetahui kemampuan model XGBoost dalam mengklasifikasikan data uji, dilakukan proses prediksi menggunakan fungsi predict() seperti pada gambar Gambar 4.15. Prediksi yang dihasilkan kemudian dibandingkan dengan label sebenarnya untuk memperoleh nilai performa model berdasarkan *confusion matrix*, *accuracy*, *precision*, *recall*, dan F1-*Score*.

```
y_pred_xgb = xgb_model.predict(X_test)
```

Gambar 4.15 Implementasi Prediksi Model XGBoost

Gambar 4.16 menunjukkan hasil *confusion matrix* dari model XGBoost. Secara umum, model mampu mengklasfikasikan hampir seluruh data uji dengan benar. Kesalahan prediksi hanya terjadi pada beberapa sampel, yaitu empat data Benign yang diprediksi sebagai Bot, dua data Benign yang diprediksi sebagai DoS, satu data DoS yang diprediksi sebagai Benign, satu data PortScan yang diprediksi sebagai DoS. Sementara itu, seluruh data pada kelas Bot, BruteForce dan WebAttack berhasil dikenali tanpa kesalahan klasifikasi.



Gambar 4.16 *Confusion Matrix* Model XGBoost

Tabel 4.5 Hasil Evaluasi Model XGBoost

Metrik	Nilai
Accuracy	0,9965
Precision	0,9965
Recall	0,9965

F1-Score	0,9965

Berdasarkan hasil evaluasi pada Tabel 4.5, model XGBoost memperoleh nilai *accuracy*, *precision*, *recall*, dan F1-*Score* masing-masing sebesar 0,9965. Nilai tersebut menunjukkan bahwa model memiliki tingkat ketepatan klasifikasi yang sangat tinggi pada dataset CIC-IDS2017. Jika dibandingkan dengan Random Forest, XGBoost memberikan performa yang sedikit lebih baik, ditunjukkan oleh peningkatan nilai pada seluruh metrik evaluasi. Hasil ini mengindikasikan bahwa XGBoost lebih efektif dalam mempelajari pola dari data sehingga mampu mengurangi jumlah kesalahan klasifikasi pada beberapa kelas.

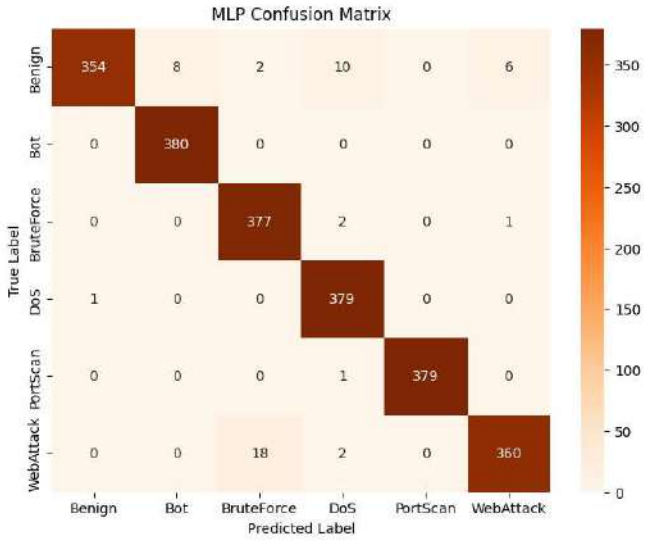
4.3.3 Evaluasi Model Multi-Layer Perceptron

Berbeda dengan Random Forest dan XGBoost, evaluasi model MLP dilakukan menggunakan data uji yang telah melalui proses *feature scaling*. Langkah ini diperlukan karena MLP merupakan model berbasis jaringan saraf yang sensitif terhadap perbedaan skala antar fitur. Selanjutnya, model menghasilkan prediksi terhadap data uji menggunakan fungsi predict() seperti yang ada di Gambar 4.17, kemudian hasil prediksi dibandingkan dengan label sebenarnya untuk memperoleh nilai *accuracy, precision, recall*, F1-*Score*, serta menyusun *confusion matrix*.

```
y_pred_mlp = mlp_model.predict(
    X_test_scaled
)
```

Gambar 4.17 Implementasi Prediksi Model Multi-Layer Perceptron

Hasil klasifikasi model MLP ditunjukkan pada Gambar 4.18. Berdasarkan *confusion matrix*, sebagian besar data uji berhasil diklasifikasikan dengan benar. Namun, dibandingkan dua model sebelumnya, MLP menghasilkan jumlah kesalahan klasifikasi yang lebih banyak. Kesalahan tersebut terutama terjadi pada kelas Benign yang diprediksi sebagai BruteForce, DoS, dan WebAttack, serta pada kelas WebAttack yang cukup sering diprediksi sebagai BruteForce. Meskipun demikian, model masih mampu mengklasifikasikan seluruh data pada kelas Bot dengan benar.



Gambar 4.18 Confusion Matrix Model Multi-Layer Perceptron

Tabel 4.6 Hasil Evaluasi Model Multi-Layer Perceptron

Metrik	Nilai
Accuracy	0,9776
Precision	0,9782
Recall	0,9776
F1-Score	0,9775

Berdasarkan hasil evaluasi pada Tabel 4.6, model MLP memperoleh nilai *accuracy* sebesar 0,9776, *precision* sebesar 0,9782, *recall* sebesar 0,9776, dan F1-*Score* sebesar 0,9775. Meskipun seluruh metrik menunjukkan performa yang masih tergolong sangat baik, nilainya lebih rendah dibandingkan Random Forest maupun XGBoost. Hasil ini menunjukkan bahwa MLP belum mampu mempelajari karakteristik data CIC-IDS2017 seefektif dua model berbasis *ensemble tree*. Hal tersebut juga terlihat pada *confusion matrix*, di mana jumlah kesalahan klasifikasi lebih banyak, terutama pada kelas Benign dan WebAttack.

4.3.4 Perbandingan Performa Model

Setelah dilakukan evaluasi terhadap masing-masing model, langkah selanjutnya adalah membandingkan performa Random Forest, XGBoost, dan MLP berdasarkan metrik *accuracy, precision, recall,* dan F1-*Score*. Perbandingan ini bertujuan untuk mengetahui model yang

memberikan performa terbaik dalam mengklasifikasikan enam kelas serangan pada dataset CIC-IDS2017.

Tabel 4.7 Perbandingan Performa Model

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0,9947	0,9948	0,9947	0,9947
XGBoost	**0,9965**	**0,9965**	**0,9965**	**0,9965**
Multi-Layer Perceptron	0,9776	0,9782	0,9776	0,9775

Berdasarkan Tabel 4.7, ketiga model menunjukkan performa klasifikasi yang sangat baik dengan nilai seluruh metrik di atas 97%. Hal ini menunjukkan bahwa ketiga algoritma mampu mengenali pola pada dataset CIC-IDS2017 dengan tingkat akurasi yang tinggi. Namun demikian, terdapat perbedaan performa di antara ketiga model yang digunakan.

Model XGBoost memperoleh nilai tertinggi pada seluruh metrik evaluasi, yaitu *accuracy, precision, recall,* dan F1-*Score* sebesar 0,9965. Hasil tersebut menunjukkan bahwa XGBoost memiliki kemampuan terbaik dalam mengklasifikasikan enam kelas serangan pada penelitian ini. Selain itu, hasil *confusion matrix* juga menunjukkan bahwa jumlah kesalahan klasifikasi pada XGBoost lebih sedikit dibandingkan model lainnya.

Model Random Forest juga menghasilkan performa yang sangat baik dengan nilai *accuracy, precision, recall*, dan F1-*Score* terbaik setelah XGBoost. Selisih performanya terhadap XGBoost relatif kecil, sehingga kedua model sama-sama mampu memberikan hasil klasifikasi yang sangat akurat. Di sisi lain, model MLP memperoleh nilai evaluasi yang lebih rendah dibandingkan kedua model berbasis *ensemble tree*. Meskipun demikian, nilai evaluasi MLP masih berada di atas 97%, sehingga model ini tetap menunjukkan kemampuan klasifikasi yang baik pada dataset yang digunakan.

Berdasarkan hasil perbandingan tersebut, dapat disimpulkan bahwa XGBoost merupakan model dengan performa terbaik pada penelitian ini, diikuti oleh Random Forest dan MLP. Selanjutnya, hasil prediksi dari masing-masing model akan diinterpretasikan menggunakan metode XAI, yaitu SHAP dan LIME. Untuk mengetahui fitur-fitur yang paling berpengaruh dalam proses pengambilan keputusan model.

4.3.5 Pengujian Ketahanan Model pada Sisa Data

Setelah melakukan evaluasi performa pada data uji yang telah melalui proses *balancing*, penelitian ini melakukan pengujian tambahan menggunakan keseluruhan sisa data CIC-IDS2017 yang belum pernah dilihat model. Pengujian ini bertujuan untuk memvalidasi *robustness* model XGBoost sebagai model terbaik ketika dihadapkan pada skenario dunia nyata. Hasil prediksi model XGBoost pada jutaan *unseen data* tersebut dievaluasi dan disajikan pada Tabel 4.8 berikut.

Tabel 4.8 Hasil Pengujian Model XGBoost pada Sisa Data (*Unseen Data*)

Kelas	Precision	Recall	F1-Score
Benign	1,00	0,99	0,99
Bot	0,12	1,00	0,22
BruteForce	0,82	1,00	0,90
DoS	0,99	1,00	0,99
PortScan	0,99	1,00	0,99
WebAttack	0,59	1,00	0,75
Accuracy	0,9907		

Berdasarkan Tabel 4.8, model XGBoost terbukti sangat tangguh dengan mempertahankan tingkat akurasi keseleruhan sebesar 99,07% pada skala data yang besar. Hal yang paling menonjol dari pengujian ini adalah nilai *recall* yang menyentuh angka 1,00 (100%) untuk seluruh kelas serangan. Nilai *recall* yang sempurna ini mengindikasikan bahwa model tidak membiarkan satu pun aktivitas serangan jaringan yang sebenarnya lolos tanpa terdeteksi.

Meskipun demikian, terlihat adanya anomali pada metrik *precision*, khususnya pada kelas serangan minoritas seperti Bot (0,12) dan WebAttack (0,59). Penurunan nilai *precison* ini disebabkan oleh tingginya *false positive* pada kelas-kelas tersebut. Kejadian ini merupakan efek samping dari penggunaan metode *random undersampling* saat proses pelatihan model. Dengan menyeimbangkan seluruh kelas menjadi masing masing 1.900 sampel pada tahap *training*, model dilatih untuk menjadi sangat sensitif terhadap karakteristik setiap jenis serangan agar tidak ada kelas minoritas yang diabaikan.

Ketika model yang sangat sensitif (*hypersensitive*) ini diimplementasikan pada kondisi dunia nyata yang didominasi oleh jutaan pake data *benign*, model cenderung mencurigai lalu lintas data normal yang memiliki kemiripan pola dengan kelas serangan, dan langsung

melabelinya sebagai ancaman. Hal ini menyebabkan banyak data *benign* yang salah diklasifikasikan sebagai Bot atau WebAttack.

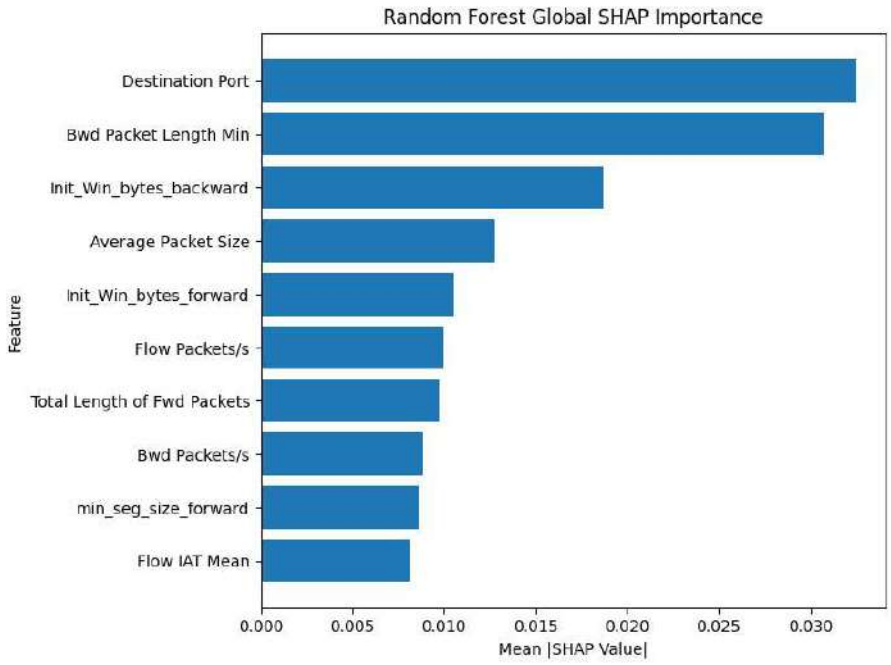
Namun, dari perspektif operasional keamanan siber, *trade off* antara *precision* dan *recall* ini merupakan hasil yang sangat ideal. Dalam penerapan NIDS, sistem jauh lebih baik menghasilkan peringatan palsu yang dapat divalidasi lebih lanjut oleh analis keamanan, daripada memiliki sistem yang melewatkan serangan nyata yang dapat berakibat fatal pada kerusakan atau pencurian data organisasi. Hasil ini menegaskan bahwa model yang dibangun sudah memenuhi standar keandalan untuk diimplementasikan sebagai sistem deteksi dini.

4.4 Interpretasi Model Menggunakan SHAP

Setelah dilakukan evaluasi terhadap performa model, tahap selanjutnya adalah menginterpretasikan proses pengambilan keputusan model menggunakan metode SHAP. SHAP digunakan untuk mengukur kontribusi setiap fitur terhadap prediksi yang dihasilkan model sehingga dapat memberikan penjelasan mengenai fitur-fitur yang paling berpengaruh dalam proses klasifikasi. Pada penelitian ini, interpretasi dilakukan terhadap tiga model, yaitu Random Forest, XGBoost, dan MLP, baik secara global maupun lokal.

4.4.1 Interpretasi Global Model Random Forest

Interpretasi global pada model Random Forest dilakukan untuk mengetahui tingkat kepentingan masing-masing fitur terhadap keseluruhan proses klasifikasi. Nilai kepentingan fitur dihitung berdasarkan rata-rata nilai absolut SHAP (*mean absolute SHAP value*), sehingga semakin besar nilai SHAP suatu fitur maka semakin besar pula kontribusinya terhadap keputusan model. Hasil interpretasi global ditunjukkan pada Gambar 4.19.



Gambar 4.19 Visualisasi Global SHAP Importance Model Random Forest

Berdasarkan Gambar 4.19, visualisasi global SHAP menunjukkan bahwa model Random Forest memanfaatkan sejumlah fitur utama dalam proses klasifikasi. Panjang batang pada grafik merepresentasikan nilai rata-rata absolut SHAP (*Mean SHAP Value*), sehingga semakin panjang batang suatu fitur menunjukkan semakin besar kontribusinya terhadap keputusan model. Secara visual terlihat bahwa hanya beberapa fitur memiliki kontribusi yang jauh lebih dominan dibandingkan fitur lainnya, sedangkan sebagian besar fitur memberikan pengaruh yang relatif lebih kecil. Kondisi ini menunjukkan bahwa model Random Forest mampu memusatkan proses pengambilan keputusan pada fitur-fitur yang paling informatif tanpa bergantung pada seluruh fitur secara merata.

Tabel 4.9 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model Random Forest

Peringkat	Fitur	Mean_SHAP
1	Destination Port	0,032511
2	Bwd Packet Length Min	0,030731
3	Init_Win_bytes_backward	0,018677
4	Average Packet Size	0,012781
5	Init_Win_bytes_forward	0,010509
6	Flow Packets/s	0,010006
7	Total Length of Fwd Packets	0,009741
8	Bwd Packets/s	0,008856

9	min_seg_size_forward	0,008589
10	Flow IAT Mean	0,008102

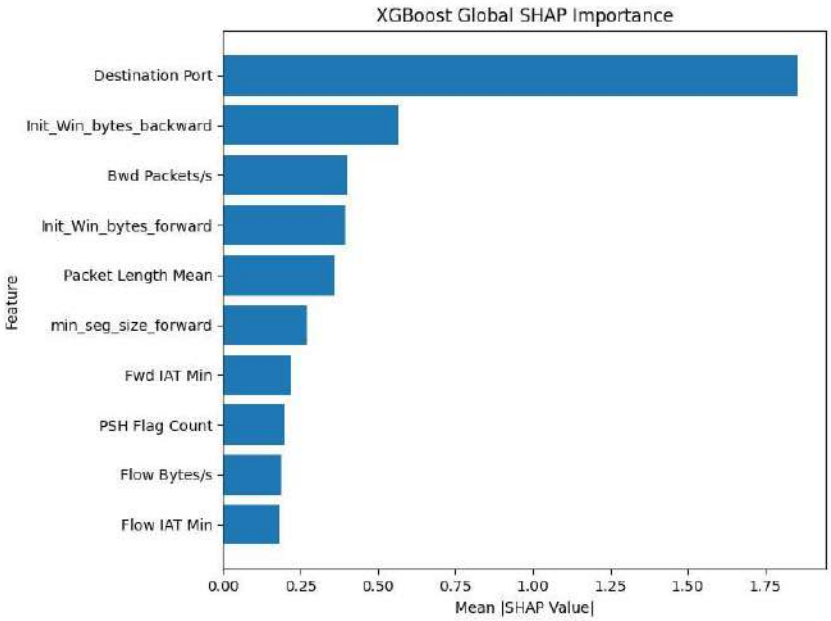
Berdasarkan Tabel 4.9, fitur Destination Port memiliki nilai *Mean SHAP Value* tertinggi sebesar 0,032511, diikuti oleh Bwd Packet Length Min sebesar 0,030731. Kedua fitur tersebut memiliki nilai yang relatif lebih tinggi dibandingkan fitur lainnya, menunjukkan bahwa model Random Forest sangat mengandalkan informasi mengenai port tujuan dan karakteristik ukuran paket pada arah komunikasi balik dalam membedakan aktivitas normal maupun aktivitas serangan.

Selain itu, fitur Init_Win_bytes_backward, Average Packet Size, dan Init_Win_bytes_forward juga memberikan kontribusi yang cukup besar terhadap proses klasifikasi. Ketiga fitur tersebut berkaitan dengan karakteristik komunikasi TCP dan ukuran paket, yang mampu menggambarkan perilaku lalu lintas jaringan pada berbagai jenis serangan. Sementara itu, fitur Flow Packets/s, Total Length of Fwd Packets, Bwd Packets/s, min_seg_size_forward, dan Flow IAT Mean menunjukkan bahwa model juga mempertimbangkan laju pengiriman paket, ukuran paket, serta pola waktu komunikasi sebagai informasi pendukung dalam proses pengambilan keputusan.

Secara keseluruhan, hasil interpretasi global menunjukkan bahwa model Random Forest tidak hanya mengandalkan satu fitur dominan, tetapi memanfaatkan kombinasi beberapa fitur yang saling melengkapi. Hal tersebut menunjukkan bahwa proses klasifikasi dibentuk berdasarkan berbagai karakteristik lalu lintas jaringan sehingga model mampu membedakan enam kelas pada penelitian ini secara lebih komprehensif.

4.4.2 Interpretasi Global Model XGBoost

Interpretasi global pada model XGBoost, dilakukan untuk mengidentifikasi fitur-fitur yang memberikan kontribusi terbesar terhadap proses klasifikasi pada seluruh data uji. Pada penelitian ini, interpretasi global dilakukan menggunakan metode SHAP dengan menghitung nilai rata-rata absolut SHAP (*Mean SHAP Value*) dari setiap fitur. Nilai tersebut menunjukkan tingkat kepentingan masing-masing fitur terhadap keputusan model secara keseluruhan, sehingga semakin besar nilai *Mean SHAP Value* maka semakin besar pula pengaruh fitur tersebut dalam proses klasifikasi.



Gambar 4.20 Hasil Global SHAP Importance pada Model XGBoost

Berdasarkan Gambar 4.20, visualisasi global SHAP menunjukkan bahwa model XGBoost memanfaatkan sejumlah fitur utama dalam proses klasifikasi. Panjang batang pada grafik merepresentasikan nilai rata-rata absolut SHAP (*Mean SHAP Value*), sehingga fitur dengan batang yang lebih panjang memiliki kontribusi yang lebih besar terhadap keputusan model. Terlihat bahwa Destination Port memiliki kontribusi yang jauh lebih dominan dibandingkan fitur lainnya, sedangkan sembilan fitur berikutnya memiliki kontribusi yang relatif lebih kecil. Hal tersebut menunjukkan bahwa XGBoost memberikan perhatian yang sangat besar terhadap fitur tertentu dalam membentuk proses klasifikasi, namun tetap mempertimbangkan kombinasi beberapa fitur lain sebagai informasi pendukung.

Tabel 4.10 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model XGBoost

Peringkat	Fitur	Mean_SHAP
1	Destination Port	1,853921
2	Init_Win_bytes_backward	0,566692
3	Bwd Packets/s	0,402260
4	Init_Win_bytes_forward	0,394701
5	Packet Length Mean	0,360132
6	min_seg_size_forward	0,269145
7	Fwd IAT Min	0,217613
8	PSH Flag Count	0,198364
9	Flow Bytes/s	0,189196

10	Fwd IAT Min	0,180552

Berdasarkan Tabel 4.10, fitur Destination Port memiliki nilai *Mean SHAP Value* tertinggi sebesar 1,853921, sehingga menjadi fitur yang paling berpengaruh terhadap proses klasifikasi pada model XGBoost. Nilai tersebut jauh lebih tinggi dibandingkan fitur lainnya, menunjukkan bahwa informasi mengenai *port* tujuan memiliki peran yang sangat dominan dalam membedakan aktivitas normal dan berbagai jenis serangan jaringan. Karakteristik ini sesuai dengan sifat beberapa serangan, seperti PortScan, DoS, dan BruteForce, yang umumnya menunjukkan pola komunikasi pada port tertentu sehingga dapat dimanfaatkan sebagai indikator penting dalam proses klasifikasi.

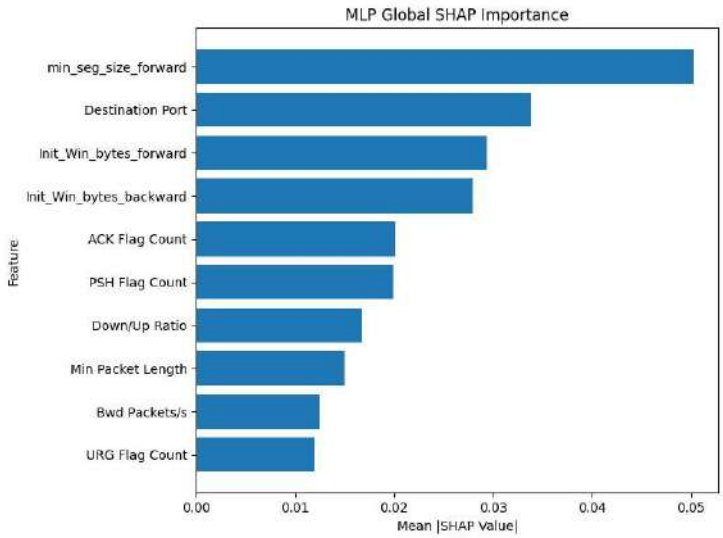
Selain Destination Port, fitur Init_Win_bytes_backward, Bwd Packets/s, Init_Win_bytes_forward, dan Packet Length Mean juga memberikan kontribusi yang cukup besar terhadap keputusan model. Fitur-fitur tersebut berkaitan dengan karakteristik komunikasi TCP, laju pengiriman paket, serta ukuran paket yang mencerminkan pola lalu lintas jaringan selama proses komunikasi berlangsung. Sementara itu, fitur min_seg_size_forward, Fwd IAT Min, PSH Flag Count, Flow Bytes/s, dan Flow IAT Min berperan sebagai informasi pendukung yang membantu model mengenali perbedaan pola komunikasi pada setiap kelas serangan.

Secara keseluruhan, hasil interpretasi global menunjukkan bahwa model XGBoost memanfaatkan kombinasi fitur yang berkaitan dengan informasi port, karakteristik paket, laju lalu lintas, serta atribut komunikasi TCP dalam membangun proses klasifikasi. Dibandingkan dengan model Random Forest, XGBoost menunjukkan ketergantungan yang lebih besar terhadap fitur Destination Port, yang ditunjukkan oleh nilai *Mean SHAP Value* yang jauh lebih tinggi dibandingkan fitur lainnya. Hal ini mengindikasikan bahwa XGBoost mampu mengidentifikasi fitur yang paling diskriminatif untuk membedakan aktivitas normal dan aktivitas serangan, sekaligus tetap memanfaatkan fitur-fitur pendukung untuk meningkatkan akurasi klasifikasi.

4.4.3 Interpretasi Global Model Multi-Layer Perceptron

Interpretasi global pada model MLP, dilakukan untuk mengidentifikasi fitur-fitur yang memberikan kontribusi terbesar terhadap proses klasifikasi pada seluruh data uji. Pada penelitian ini, interpretasi global menggunakan metode SHAP dengan menghitung *Mean SHAP Value* dari setiap fitur. Nilai tersebut menunjukkan tingkat kepentingan masing-masing

fitur terhadap keputusan model secara keseluruhan, sehingga semakin besar nilai *Mean SHAP Value* maka semakin besar pula pengaruh fitur tersebut dalam proses klasifikasi.



Gambar 4.21 Hasil Global SHAP Importance pada Model MLP

Berdasarkan Gambar 4.21, visualisasi global SHAP menunjukkan bahwa model MLP memanfaatkan beberapa fitur utama dalam proses klasifikasi lalu lintas jaringan. Panjang batang pada grafik merepresentasikan nilai *Mean SHAP Value*, sehingga fitur dengan batang yang lebih panjang memiliki kontribusi yang lebih besar terhadap keputusan model. Berbeda dengan model Random Forest dan XGBoost, pada model MLP fitur min_seg_size_forward memiliki kontribusi terbesar, diikuti oleh Destination Port, Init_Win_bytes_forward, dan Init_Win_bytes_backward. Hal ini menunjukkan bahwa MLP memberikan perhatian yang lebih besar terhadap karakteristik ukuran segmen paket serta atribut komunikasi TCP dalam membangun proses klasifikasi, meskipun tetap mempertimbangkan informasi mengenai port tujuan dan karakteristik lalu lintas jaringan lainnya.

Tabel 4.11 Sepuluh Fitur dengan Nilai Mean SHAP Tertinggi pada Model MLP

Peringkat	Fitur	Mean_SHAP
1	Min_seg_size_forward	0,050220
2	Destination Port	0,033811
3	Init_win_bytes_forward	0,029325
4	Init_win_bytes_backward	0,027918
5	ACK Flag Count	0,020133
6	PSH Flag Count	0,019903

7	Down/Up Ratio	0,016706
8	Min Packet Length	0,014967
9	Bwd Packets/s	0,012438
10	URG Flag Count	0,011937

Berdasarkan Tabel 4.11, fitur min_seg_size_forward memiliki nilai *Mean SHAP Value* tertinggi sebesar 0,050220, sehingga menjadi fitur yang paling berpengaruh terhadap proses klasifikasi pada model MLP. Fitur ini merepresentasikan ukuran minimum segmen pada arah komunikasi maju (*forward*), yang dapat menggambarkan karakteristik pengiriman paket selama proses komunikasi jaringan. Tingginya kontribusi fitur tersebut menunjukkan bahwa MLP memanfaatkan informasi mengenai pola ukuran segmen paket sebagai salah satu indikator utama dalam membedakan aktivitas normal dan berbagai jenis serangan jaringan.

Secara keseluruhan, hasil interpretasi global menunjukkan bahwa model MLP memanfaatkan kombinasi fitur yang berkaitan dengan karakteristik paket, atribut komunikasi TCP, serta pola lalu lintas jaringan dalam melakukan proses klasifikasi. Berbeda dengan model Random Forest dan XGBoost yang menempatkan Destination Port sebagai fitur dengan kontribusi terbesar, model MLP lebih menitikberatkan pada fitur min_seg_size_forward sebagai indikator utama dalam proses pengambilan keputusan. Perbedaan tersebut menunjukkan bahwa setiap algoritma *machine learning* dapat mempelajari representasi data yang berbeda meskipun menggunakan dataset yang sama, sehingga interpretasi menggunakan SHAP memberikan informasi yang penting untuk memahami karakteristik keputusan masing-masing model.

4.4.4 Perbandingan Interpretasi Global Antar Model

Perbandingan interpretasi global dilakukan untuk mengetahui persamaan dan perbedaan fitur-fitur yang dianggap penting oleh setiap model *machine learning* dalam proses klasifikasi. Meskipun ketiga model dilatih menggunakan dataset yang sama, mekanisme pembelajaran yang berbeda memungkinkan setiap model memberikan tingkat kepentingan yang berbeda terhadap suatu fitur. Hasil perbandingan sepuluh lima dengan nilai *Mean SHAP Value* tertinggi pada model Random Forest, XGBoost, dan MLP disajikan pada Tabel 4.12.

Tabel 4.12 Perbandingan Lima Fitur dengan Nilai Mean SHAP Tertinggi pada Setiap Model

Peringkat	Random Forest	XGBoost	MLP
1	**Destination Port**	**Destination Port**	Min_seg_size_forward

2	Bwd Packet Length Min	**Init_Win_bytes_backward**	**Destination Port**
3	**Init_Win_bytes_backward**	Bwd Packets/s	**Init_Win_bytes_forward**
4	Average Packet Size	**Init_Win_bytes_forward**	**Init_Win_bytes_backward**
5	**Init_Win_bytes_forward**	Packet Length Mean	ACK Flag Count

Berdasarkan Tabel 4.12, terlihat bahwa terdapat beberapa fitur yang secara konsisten muncul pada ketiga model, yaitu Destination Port, Init_Win_bytes_forward, Init_Win_bytes_backward. Kemunculan fitur-fitur tersebut pada seluruh model menunjukkan bahwa karakteristik yang berkaitan dengan informasi *port*, komunikasi TCP, serta pola pengiriman paket merupakan faktor penting dalam membedakan aktivitas normal dan aktivitas serangan pada dataset CIC-IDS2017.

Meskipun memiliki beberapa fitur yang sama, urutan tingkat kepentingan setiap fitur berbeda pada masing-masing model. Pada model Random Forest dan XGBoost, fitur Destination Port menempati urutan pertama sebagai fitur yang memberikan kontribusi terbesar terhadap proses klasifikasi. Hal ini menunjukkan bahwa kedua model berbasis pohon (*tree-based models*) sangat mengandalkan informasi mengenai *port* tujuan dalam membedakan berbagai jenis serangan jaringan.

Sebaliknya, pada model MLP, fitur min_seg_size_forward menjadi fitur dengan kontribusi terbesar, sedangkan Destination Port berada pada urutan kedua. Hasil tersebut menunjukkan bahwa MLP lebih menitikberatkan pada karakteristik ukuran segmen paket dalam membangun representasi data sebelum memanfaatkan informasi mengenai port tujuan. Perbedaan ini menunjukkan bahwa setiap algoritma memiliki mekanisme pembelajaran yang berbeda dalam mengenali pola lalu lintas jaringan.

Selain itu, beberapa fitur hanya muncul pada model tertentu. Sebagai contoh, Average Packet Size, Total Length of Fwd Packets, dan Flow Packets/s hanya termasuk dalam sepuluh fitur teratas pada model Random Forest. Sementara itu, Packet Length Mean, Fwd IAT Min, dan Flow Bytes/s hanya muncul pada model XGBoost, sedangkan ACK Flag Count, Down/Up Ratio, Min Packet Length, dan URG Flag Count hanya ditemukan pada model MLP. Perbedaan tersebut menunjukkan bahwa setiap model memanfaatkan karakteristik lalu lintas jaringan yang berbeda sebagai dasar dalam membentuk proses klasifikasi.

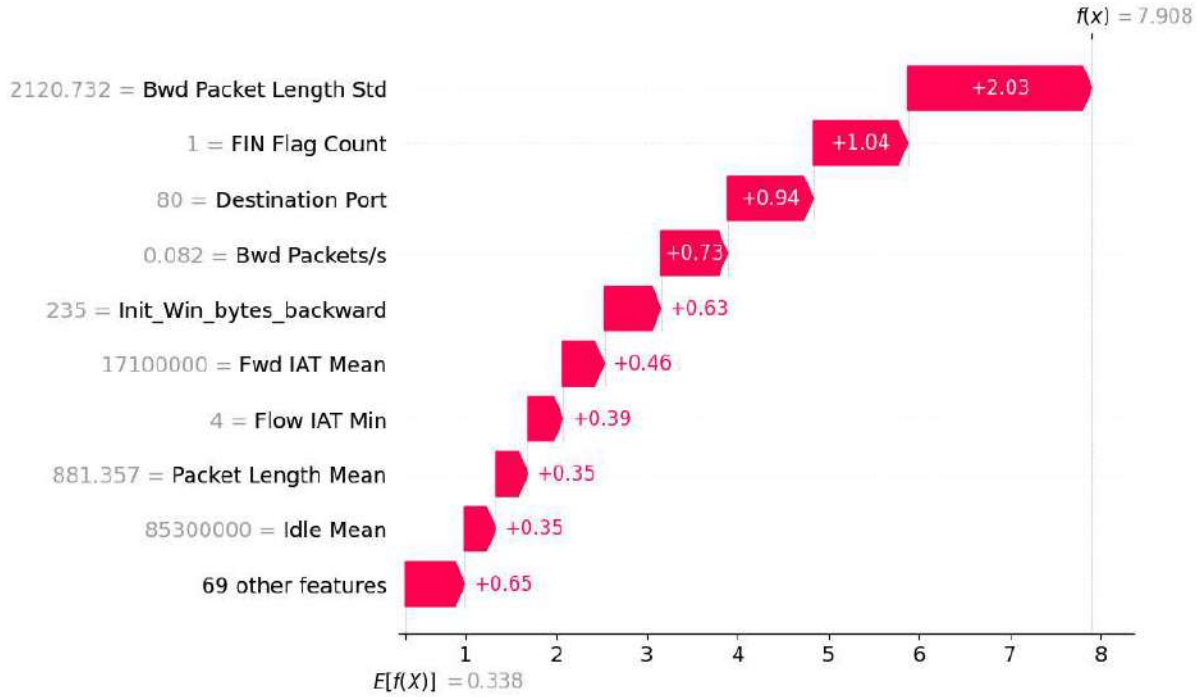
Secara keseluruhan, hasil perbandingan interpretasi global menunjukkan bahwa meskipun ketiga model memiliki fokus yang berbeda terhadap fitur-fitur tertentu, seluruh model tetap memanfaatkan kombinasi fitur yang berkaitan dengan karakteristik komunikasi jaringan, seperti informasi *port*, ukuran paket, atribut komunikasi TCP, dan pola lalu lintas jaringan. Hal ini menunjukkan bahwa fitur-fitur tersebut merupakan indikator penting dalam proses deteksi berbagai jenis serangan pada dataset CIC-IDS2017.

4.4.5 Interpretasi Lokal Menggunakan SHAP

Selain digunakan untuk memberikan interpretasi secara global, SHAP juga digunakan untuk menjelaskan proses pengambilan keputusan model pada tingkat lokal (*local interpretation*). Interpretasi lokal bertujuan untuk mengetahui kontribusi masing-masing fitur terhadap hasil prediksi pada satu sampel tertentu, sehingga dapat dijelaskan alasan model mengklasifikasikan suatu lalu lintas jaringan ke dalam kelas tertentu. Pada penelitian ini, interpretasi lokal dilakukan menggunakan model XGBoost sebagai model dengan performa terbaik berdasarkan hasil evaluasi sebelumnya. Analisis dilakukan terhadap tiga sampel yang mewakili kelas DoS, PortScan, dan Bot untuk mengetahui fitur-fitur yang memberikan kontribusi positif maupun negatif terhadap hasil prediksi pada masing-masing kelas.

4.4.5.1 Interpretasi Lokal Kelas DoS

Interpretasi lokal pada kelas Denial of Service (DoS) dilakukan untuk menjelaskan alasan model XGBoost mengklasifikasikan satu sampel lalu lintas jaringan sebagai serangan DoS. Berbeda dengan interpretasi global yang menunjukkan tingkat kepentingan fitur secara keseluruhan, interpretasi lokal menggunakan SHAP menjelaskan kontribusi masing-masing fitur terhadap hasil prediksi pada satu sampel tertentu. Visualisasi interpretasi lokal untuk kelas DoS ditunjukkan pada Gambar 4.22.



Gambar 4.22 Visualisasi Waterfall SHAP pada Sampel kelas DoS

Berdasarkan Gambar 4.22, nilai prediksi model untuk sampel yang dianalisis adalah 7,908, yang diperoleh dari nilai dasar (*base value*) sebesar 0,338 setelah mempertimbangkan kontribusi masing-masing fitur. Seluruh fitur yang ditampilkan pada visualisasi memberikan kontribusi positif terhadap hasil prediksi sehingga mendorong model untuk mengklasifikasikan sampel tersebut sebagai serangan DoS.

Fitur Bwd Packet Length Std memberikan kontribusi terbesar dengan nilai SHAP sebesar +2,03, menunjukkan bahwa variasi panjang paket pada arah komunikasi balik (*backward traffic*) menjadi indikator utama yang digunakan model dalam mengenali pola serangan DoS pada sampel ini. Kontribusi besar dari fitur tersebut mengindikasikan adanya karakteristik lalu lintas yang berbeda dibandingkan aktivitas normal, khususnya pada variasi ukuran paket yang diterima.

Selain itu, fitur FIN Flag Count (+1,04) dan Destination Port (+0,94) juga memberikan pengaruh yang cukup besar terhadap keputusan model. Fitur FIN Flag Count menggambarkan jumlah paket yang membawa FIN flag pada komunikasi TCP, sedangkan Destination Port menunjukkan layanan tujuan yang diakses oleh lalu lintas jaringan. Kedua fitur tersebut membantu model dalam membedakan pola komunikasi yang berkaitan dengan aktivitas serangan.

Fitur lainnya, seperti Bwd Packets/s, Init_Win_bytes_backward, Fwd IAT Mean, Flow IAT Min, Packet Length Mean, dan Idle Mean, turut memberikan kontribusi positif meskipun dengan nilai yang lebih kecil. Kombinasi kontribusi dari seluruh fitur tersebut meningkatkan nilai prediksi dari 0,338 menjadi 7,908, sehingga model memberikan keyakinan yang tinggi dalam mengklasifikasikan sampel sebagai serangan DoS. Hasil interpretasi ini menunjukkan bahwa keputusan model tidak dipengaruhi oleh satu fitur saja, melainkan oleh kombinasi beberapa karakteristik lalu lintas jaringan yang saling melengkapi.

4.4.5.2 Interpretasi Lokal Kelas PortScan

Interpretasi lokal pada kelas PortScan dilakukan untuk menjelaskan alasan model XGBoost mengklasifikasikan satu sampel lalu lintas jaringan sebagai serangan PortScan. Melalui metode SHAP, kontribusi masing-masing fitur terhadap hasil prediksi dapat diketahui sehingga proses pengambilan keputusan model menjadi lebih mudah dipahami. Visualisasi interpretasi lokal untuk sampel kelas PortScan ditunjukkan pada Gambar 4.23.



Gambar 4.23 Visualisasi Waterfall SHAP pada Sampel Kelas PortScan

Berdasarkan Gambar 4.23, model menghasilkan nilai prediksi sebesar 7,71, yang diperoleh dari nilai dasar (*base value*) sebesar 0,454 setelah mempertimbangkan kontribusi masing-masing fitur. Seluruh fitur yang ditampilkan memberikan kontribusi positif terhadap hasil prediksi sehingga mendorong model untuk mengklasifikasikan sampel tersebut sebagai serangan PortScan.

Fitur PSH Flag Count memberikan kontribusi terbesar dengan nilai SHAP sebesar +1,22, diikuti oleh Bwd Packets/s sebesar +1,09 dan Packet Length Mean sebesar +0,97. Ketiga fitur tersebut menunjukkan bahwa model memanfaatkan karakteristik komunikasi TCP, laju pengiriman paket pada arah komunikasi balik, serta rata-rata ukuran paket sebagai indikator utama dalam mengenali pola aktivitas PortScan. Karakteristik tersebut sesuai dengan perilaku PortScan yang umumnya menghasilkan banyak percobaan koneksi dalam waktu singkat dengan pola paket tertentu.

Selain itu, fitur Total Fwd Packets, min_seg_size_forward, Flow Bytes/s, Bwd Packet Length Min, Init_Win_bytes_backward, dan Destination Port juga memberikan kontribusi positif terhadap keputusan model. Meskipun nilai kontribusinya lebih kecil dibandingkan tiga fitur utama, kombinasi seluruh fitur tersebut meningkatkan nilai prediksi dari 0,454 menjadi 7,71, sehingga model memiliki keyakinan yang tinggi dalam mengklasifikasikan sampel sebagai serangan PortScan.

Hasil interpretasi lokal ini menunjukkan bahwa model XGBoost tidak hanya mempertimbangkan informasi mengenai *port* tujuan, tetapi juga memanfaatkan karakteristik komunikasi TCP, ukuran paket, jumlah paket, serta laju lalu lintas jaringan dalam membedakan aktivitas PortScan dari aktivitas normal maupun jenis serangan lainnya. Kombinasi berbagai fitur tersebut menunjukkan bahwa keputusan model dibentuk berdasarkan representasi karakteristik lalu lintas jaringan secara menyeluruh, bukan hanya bergantung pada satu indikator tertentu.

4.4.5.3 Interpretasi Lokal Kelas Bot

Interpretasi lokal pada kelas Bot dilakukan untuk menjelaskan alasan model XGBoost mengklasifikasikan satu sampel lalu lintas jaringan sebagai aktivitas Bot. Melalui metode SHAP, kontribusi setiap fitur terhadap hasil prediksi dapat dianalisis sehingga proses pengambilan keputusan model dapat dipahami secara lebih rinci. Visualisasi interpretasi lokal untuk sampel kelas Bot ditunjukkan pada Gambar 4.24.



Gambar 4.24 Visualisasi Waterfall SHAP pada Sampel Kelas Bot

Berdasarkan Gambar 4.24, model menghasilkan nilai prediksi sebesar 6,573, yang diperoleh dari nilai dasar (*base value*) sebesar 0,365 setelah mempertimbangkan kontribusi masing-masing fitur. Berbeda dengan interpretasi lokal pada kelas DoS dan PortScan, visualisasi pada kelas Bot menunjukkan bahwa terdapat fitur yang memberikan kontribusi positif maupun negatif terhadap hasil prediksi. Kondisi tersebut menunjukkan bahwa proses klasifikasi pada sampel ini dipengaruhi oleh interaksi beberapa fitur yang saling memperkuat maupun mengurangi kecenderungan prediksi model.

Fitur Destination Port memberikan kontribusi positif terbesar dengan nilai SHAP sebesar +5,99, sehingga menjadi faktor utama yang mendorong model mengklasifikasikan sampel sebagai aktivitas Bot. Besarnya kontribusi tersebut menunjukkan bahwa informasi mengenai *port* tujuan memiliki peran yang sangat dominan dalam mengenali pola komunikasi yang berkaitan dengan aktivitas botnet. Selain itu, fitur Init_Win_bytes_backward, Bwd IAT Std, min_seg_size_forward, dan Init_Win_bytes_forward juga memberikan kontribusi positif, meskipun nilainya relatif lebih kecil dibandingkan Destination Port.

Di sisi lain, beberapa fitur memberikan kontribusi negatif terhadap hasil prediksi, yaitu URG Flag Count, Fwd Header Length, Flow IAT Min, dan Flow Bytes/s. Kontribusi negatif tersebut menunjukkan bahwa nilai pada fitur-fitur tersebut mengurangi kecenderungan model untuk mengklasifikasikan sampel sebagai Bot. Meskipun demikian, pengaruh negatif tersebut tidak mampu mengimbangi besarnya kontribusi positif dari Destination Port, sehingga nilai prediksi tetap meningkat dari 0,365 menjadi 6,573.

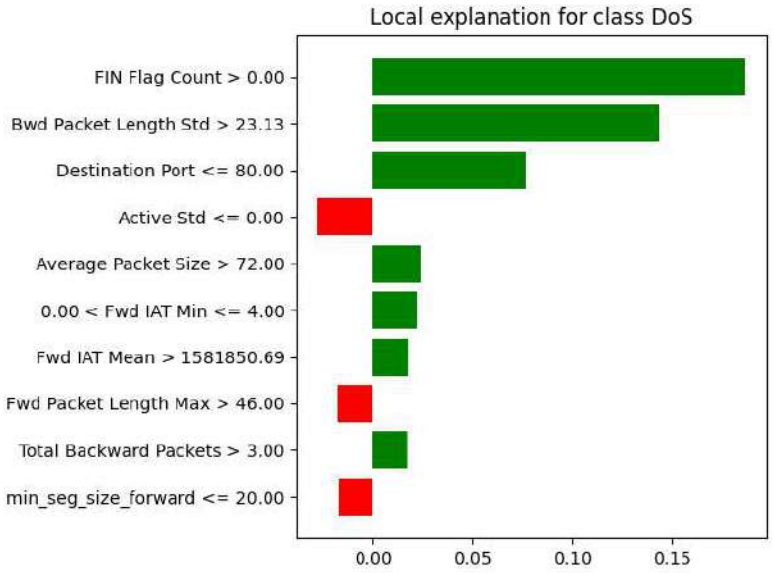
Secara keseluruhan, hasil interpretasi lokal menunjukkan bahwa proses klasifikasi pada kelas Bot dipengaruhi oleh kombinasi kontribusi positif dan negatif dari beberapa fitur. Dominannya kontribusi Destination Port mengindikasikan bahwa model sangat mengandalkan informasi mengenai *port* tujuan dalam mengenali aktivitas Bot, sedangkan fitur-fitur lain berperan sebagai faktor pendukung yang memperkuat maupun menyesuaikan keputusan model. Hasil ini menunjukkan bahwa SHAP tidak hanya mengidentifikasi fitur yang meningkatkan hasil prediksi, tetapi juga mampu memperlihatkan fitur-fitur yang mengurangi keyakinan model dalam proses klasifikasi.

4.5 Interpretasi Model Menggunakan LIME

Selain menggunakan SHAP sebagai metode interpretasi utama, penelitian ini juga menerapkan LIME untuk memberikan interpretasi pada tingkat lokal terhadap hasil prediksi model. LIME bekerja dengan membangun model sederhana di sekitar sampel yang dianalisis sehingga dapat menjelaskan fitur-fitur yang paling berpengaruh terhadap keputusan model pada satu instance tertentu. Pada penelitian ini, interpretasi menggunakan LIME dilakukan pada model XGBoost sebagai model dengan performa terbaik berdasarkan hasil evaluasi sebelumnya. Analisis dilakukan terhadap tiga sampel yang mewakili kelas DoS, PortScan, dan Bot untuk memperoleh pemahaman mengenai faktor-faktor yang memengaruhi keputusan model pada masing-masing kelas.

4.5.1 Interpretasi Lokal Kelas DoS Menggunakan LIME

Interpretasi lokal pada kelas DoS menggunakan metode LIME dilakukan untuk menjelaskan fitur-fitur yang memengaruhi keputusan model XGBoost dalam mengklasifikasikan satu sampel lalu lintas jaringan sebagai serangan DoS. Berbeda dengan SHAP yang menghitung kontribusi fitur berdasarkan nilai SHAP, LIME membangun model lokal di sekitar sampel yang dianalisis sehingga dapat menunjukkan fitur-fitur yang mendukung maupun mengurangi hasil prediksi model. Visualisasi hasil interpretasi lokal menggunakan LIME ditunjukkan pada Gambar 4.25.



Gambar 4.25 Interpretasi Lokal LIME pada Sampel Kelas DoS

Berdasarkan Gambar 4.25, beberapa fitur memberikan kontribusi positif terhadap keputusan model dalam mengklasifikasikan sampel sebagai serangan DoS, sedangkan beberapa fitur lainnya memberikan kontribusi negatif. Semakin besar nilai bobot (*weight*) suatu fitur, semakin besar pula pengaruhnya terhadap hasil prediksi model.

Fitur FIN Flag Count > 0,00, Bwd Packet Length Std > 23,13, dan Destination Port ≤ 80,00 merupakan tiga fitur yang memberikan kontribusi positif terbesar terhadap hasil prediksi. Sebaliknya, fitur Active Std ≤ 0,00, Fwd Packet Length Max > 46,00, dan min_seg_size_forward ≤ 20,00 memberikan kontribusi negatif sehingga sedikit mengurangi kecenderungan model dalam mengklasifikasikan sampel sebagai serangan DoS. Meskipun demikian, besarnya kontribusi positif dari fitur-fitur utama menyebabkan model tetap memberikan prediksi pada kelas DoS.

Tabel 4.13 Bobot Fitur Hasil Interpretasi Lokal LIME pada Sampel Kelas DoS

No	Feature	Weight
1	FIN Flag Count > 0.00	0.186669
2	Bwd Packet Length Std > 23.13	0.143739
3	Destination Port ≤ 80.00	0.076706
4	Active Std ≤ 0.00	-0.027475
5	Average Packet Size > 72.00	0.024090
6	0.00 < Fwd IAT Min ≤ 4.00	0.022435
7	Fwd IAT Mean > 1581850.69	0.017941
8	Fwd Packet Length Max > 46.00	-0.017661
9	Total Backward Packets > 3.00	0.017352

10	min_seg_size_forward ≤ 20.00	-0.017085

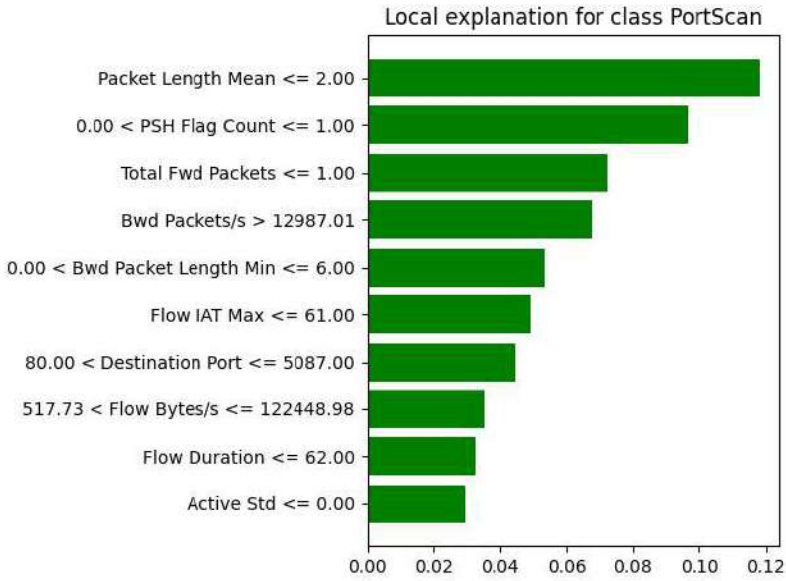
Berdasarkan Tabel 4.13 Bobot Fitur Hasil Interpretasi Lokal LIME pada Sampel Kelas DoS, fitur FIN Flag Count > 0,00 memiliki bobot terbesar, yaitu 0,186669, sehingga menjadi faktor yang paling memengaruhi keputusan model dalam mengklasifikasikan sampel sebagai serangan DoS. Fitur Bwd Packet Length Std > 23,13 dan Destination Port ≤ 80,00 juga memberikan kontribusi positif yang cukup besar, menunjukkan bahwa karakteristik komunikasi TCP, variasi panjang paket pada arah komunikasi balik, serta informasi mengenai port tujuan merupakan indikator penting dalam proses identifikasi serangan DoS.

Di sisi lain, fitur Active Std ≤ 0,00, Fwd Packet Length Max > 46,00, dan min_seg_size_forward ≤ 20,00 memiliki bobot negatif. Hal tersebut menunjukkan bahwa nilai pada fitur-fitur tersebut cenderung mengurangi keyakinan model terhadap kelas DoS. Namun, pengaruh negatif tersebut relatif kecil dibandingkan kontribusi positif dari fitur-fitur utama sehingga tidak mengubah hasil prediksi model.

Secara keseluruhan, hasil interpretasi menggunakan LIME menunjukkan bahwa keputusan model XGBoost pada sampel kelas DoS dipengaruhi oleh kombinasi beberapa fitur dengan arah kontribusi yang berbeda. Temuan ini sejalan dengan hasil interpretasi menggunakan SHAP, di mana fitur-fitur seperti FIN Flag Count, Bwd Packet Length Std, dan Destination Port juga muncul sebagai faktor penting dalam proses klasifikasi. Dengan demikian, kedua metode memberikan interpretasi yang saling mendukung terhadap alasan model mengklasifikasikan sampel sebagai serangan DoS.

4.5.2 Interpretasi Lokal Kelas PortScan Menggunakan LIME

Interpretasi lokal pada kelas PortScan menggunakan metode LIME dilakukan untuk menjelaskan fitur-fitur yang memengaruhi keputusan model XGBoost dalam mengklasifikasikan satu sampel lalu lintas jaringan sebagai serangan PortScan. Melalui pendekatan interpretasi lokal, LIME menunjukkan kontribusi setiap fitur terhadap hasil prediksi sehingga karakteristik yang menjadi dasar keputusan model dapat dipahami secara lebih jelas. Hasil interpretasi lokal untuk sampel kelas PortScan ditunjukkan pada Gambar 4.26.



Gambar 4.26 Interpretasi Lokal LIME pada Sampel Kelas PortScan

Berdasarkan Gambar 4.26, seluruh fitur yang ditampilkan memberikan kontribusi positif terhadap keputusan model dalam mengklasifikasikan sampel sebagai serangan PortScan. Besarnya bobot (*weight*) pada masing-masing fitur menunjukkan tingkat pengaruh fitur tersebut terhadap hasil prediksi, di mana semakin besar nilai bobot maka semakin besar kontribusinya dalam meningkatkan keyakinan model.

Fitur Packet Length Mean ≤ 2,00, 0,00 < PSH Flag Count ≤ 1,00, dan Total Fwd Packets ≤ 1,00 merupakan tiga fitur yang memberikan kontribusi terbesar terhadap hasil prediksi. Selain itu, fitur Bwd Packets/s > 12987,01 dan 0,00 < Bwd Packet Length Min ≤ 6,00 juga memberikan pengaruh yang cukup besar dalam mendukung keputusan model. Hasil ini menunjukkan bahwa karakteristik ukuran paket, komunikasi TCP, serta jumlah paket yang dikirim dan diterima menjadi indikator penting dalam mengenali aktivitas PortScan.

Tabel 4.14 Bobot Fitur Hasuk Interpretasi Lokal LIME pada Sampel Kelas Portscan

No	Feature	Weight
1	Packet Length Mean ≤ 2.00	0.118009
2	0.00 < PSH Flag Count ≤ 1.00	0.096558
3	Total Fwd Packets ≤ 1.00	0.072586
4	Bwd Packets/s > 12987.01	0.067734
5	0.00 < Bwd Packet Length Min ≤ 6.00	0.053507
6	Flow IAT Max ≤ 61.00	0.049463
7	80.00 < Destination Port ≤ 5087.00	0.044659

8	517.73 < Flow Bytes/s ≤ 122448.98	0.035150
9	Flow Duration ≤ 62.00	0.032555
10	Active Std ≤ 0.00	0.029407

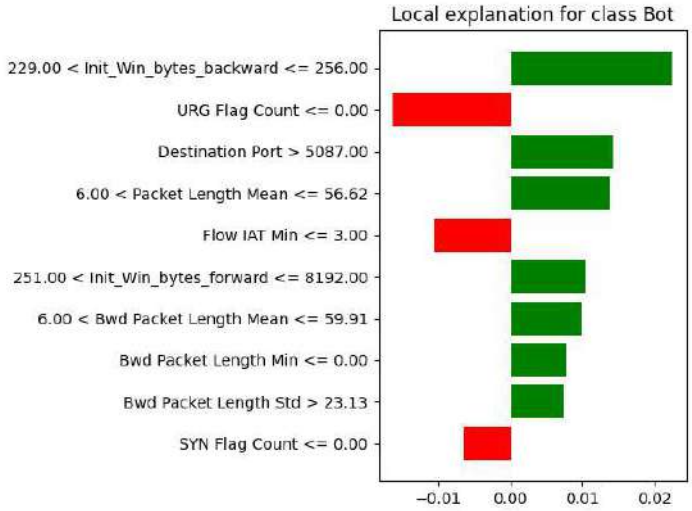
Berdasarkan Tabel 4.14, fitur Packet Length Mean ≤ 2,00 memiliki bobot terbesar, yaitu 0,118009, sehingga menjadi faktor yang paling memengaruhi keputusan model dalam mengklasifikasikan sampel sebagai serangan PortScan. Fitur 0,00 < PSH Flag Count ≤ 1,00 dan Total Fwd Packets ≤ 1,00 juga memberikan kontribusi positif yang cukup besar, menunjukkan bahwa karakteristik ukuran paket, status komunikasi TCP, serta jumlah paket yang dikirim merupakan indikator penting dalam mengidentifikasi aktivitas PortScan.

Selain itu, fitur Bwd Packets/s, Bwd Packet Length Min, Flow IAT Max, Destination Port, Flow Bytes/s, Flow Duration, dan Active Std turut memberikan kontribusi positif terhadap keputusan model. Kombinasi fitur-fitur tersebut menggambarkan bahwa model tidak hanya mempertimbangkan satu karakteristik tertentu, tetapi juga memanfaatkan informasi mengenai ukuran paket, laju lalu lintas, waktu komunikasi, serta informasi port dalam mengenali pola aktivitas PortScan.

Secara keseluruhan, hasil interpretasi menggunakan LIME menunjukkan bahwa seluruh fitur utama pada sampel ini memberikan kontribusi positif terhadap hasil prediksi. Temuan ini sejalan dengan hasil interpretasi menggunakan SHAP, di mana fitur-fitur seperti Packet Length Mean, PSH Flag Count, Bwd Packets/s, dan Destination Port juga teridentifikasi sebagai faktor penting dalam proses klasifikasi. Dengan demikian, kedua metode memberikan interpretasi yang saling mendukung terhadap alasan model mengklasifikasikan sampel sebagai serangan PortScan.

4.5.3 Interpretasi Lokal Kelas Bot Menggunakan LIME

Interpretasi lokal pada kelas Bot menggunakan metode LIME dilakukan untuk menjelaskan fitur-fitur yang memengaruhi keputusan model XGBoost dalam mengklasifikasikan satu sampel lalu lintas jaringan sebagai aktivitas Bot. Melalui pendekatan interpretasi lokal, LIME menunjukkan kontribusi setiap fitur terhadap hasil prediksi sehingga karakteristik yang menjadi dasar keputusan model dapat dipahami secara lebih rinci. Hasil interpretasi lokal untuk sampel kelas Bot ditunjukkan pada Gambar 4.27.



Gambar 4.27 Interpretasi Lokal LIME pada Sampel Kelas Bot

Berdasarkan Gambar 4.27, beberapa fitur memberikan kontribusi positif terhadap keputusan model dalam mengklasifikasikan sampel sebagai aktivitas Bot, sedangkan beberapa fitur lainnya memberikan kontribusi negatif. Nilai bobot (*weight*) menunjukkan besarnya pengaruh masing-masing fitur terhadap hasil prediksi, di mana bobot positif meningkatkan keyakinan model terhadap kelas Bot, sedangkan bobot negatif mengurangi kecenderungan prediksi tersebut.

Fitur 229,00 < Init_Win_bytes_backward ≤ 256,00 memberikan kontribusi positif terbesar terhadap hasil prediksi, diikuti oleh Destination Port > 5087,00 dan 6,00 < Packet Length Mean ≤ 56,62. Ketiga fitur tersebut menunjukkan bahwa karakteristik komunikasi TCP, informasi mengenai port tujuan, serta ukuran paket merupakan faktor utama yang digunakan model dalam mengenali aktivitas Bot. Di sisi lain, fitur URG Flag Count ≤ 0,00, Flow IAT Min ≤ 3,00, dan SYN Flag Count ≤ 0,00 memberikan kontribusi negatif sehingga mengurangi kecenderungan model dalam mengklasifikasikan sampel sebagai Bot. Meskipun demikian, besarnya kontribusi positif dari fitur-fitur utama menyebabkan model tetap memberikan prediksi pada kelas Bot.

Tabel 4.15 Bobot Fitur Hasil Interpretasi Lokal LIME pada Sampel Kelas Bot

No	Feature	Weight
1	229.00 < Init_Win_bytes_backward ≤ 256.00	0.022540
2	URG Flag Count ≤ 0.00	-0.016260
3	Destination Port > 5087.00	0.014225
4	6.00 < Packet Length Mean ≤ 56.62	0.013764

5	Flow IAT Min ≤ 3.00	-0.010661
6	251.00 < Init_Win_bytes_forward ≤ 8192.00	0.010371
7	6.00 < Bwd Packet Length Mean ≤ 59.91	0.009970
8	Bwd Packet Length Min ≤ 0.00	0.007736
9	Bwd Packet Length Std > 23.13	0.007441
10	SYN Flag Count ≤ 0.00	-0.006418

Berdasarkan Tabel 4.15, fitur 229,00 < Init_Win_bytes_backward ≤ 256,00 memiliki bobot terbesar, yaitu 0,022540, sehingga menjadi faktor yang paling memengaruhi keputusan model dalam mengklasifikasikan sampel sebagai aktivitas Bot. Selain itu, fitur Destination Port > 5087,00 dan 6,00 < Packet Length Mean ≤ 56,62 juga memberikan kontribusi positif yang cukup besar, menunjukkan bahwa informasi mengenai port tujuan, karakteristik komunikasi TCP, serta ukuran paket berperan penting dalam proses identifikasi aktivitas Bot.

Di sisi lain, fitur URG Flag Count ≤ 0,00, Flow IAT Min ≤ 3,00, dan SYN Flag Count ≤ 0,00 memiliki bobot negatif sehingga cenderung mengurangi keyakinan model terhadap kelas Bot. Namun, kontribusi negatif tersebut relatif lebih kecil dibandingkan kontribusi positif dari fitur-fitur utama, sehingga keputusan akhir model tetap mengarah pada kelas Bot.

Secara keseluruhan, hasil interpretasi menggunakan LIME menunjukkan bahwa keputusan model XGBoost pada sampel kelas Bot dipengaruhi oleh kombinasi beberapa fitur dengan kontribusi positif dan negatif. Temuan ini sejalan dengan hasil interpretasi menggunakan SHAP, di mana Destination Port dan atribut komunikasi TCP juga menjadi faktor penting dalam proses klasifikasi. Dengan demikian, kedua metode memberikan interpretasi yang saling melengkapi dalam menjelaskan alasan model mengidentifikasi sampel sebagai aktivitas Bot.

4.5.4 Perbandingan Hasil Interpretasi SHAP dan LIME

Setelah dilakukan interpretasi lokal menggunakan SHAP dan LIME pada tiga sampel yang mewakili kelas DoS, PortScan, dan Bot, langkah selanjutnya adalah membandingkan hasil interpretasi dari kedua metode tersebut. Perbandingan ini bertujuan untuk mengetahui tingkat konsistensi fitur-fitur yang dianggap penting oleh SHAP dan LIME dalam menjelaskan keputusan model XGBoost. Hasil perbandingan tersebut disajikan pada Tabel 4.16

Tabel 4.16 Perbandingan Hasil Interpretasi SHAP dan LIME

Kelas	Fitur Dominan SHAP	Fitur Dominan LIME	Analisis
DoS	Bwd Packet Length Std, FIN Flag Count, Destination Port	FIN Flag Count, Bwd Packet Length Std, Destination Port	Kedua metode secara konsisten menunjukkan bahwa karakteristik komunikasi TCP, variasi panjang paket, dan informasi port tujuan merupakan faktor utama dalam proses klasifikasi kelas DoS.
PortScan	PSH Flag Count, Bwd Packets/s, Packet Length Mean	Packet Length Mean, PSH Flag Count, Bwd Packets/s	SHAP dan LIME sama-sama menyoroti karakteristik paket, komunikasi TCP, dan laju lalu lintas jaringan sebagai dasar pengenalan aktivitas PortScan, meskipun urutan fitur yang dianggap paling penting berbeda.
Bot	Destination Port, Init_Win_bytes_backward	Init_Win_bytes_backward, Destination Port	Kedua metode menunjukkan bahwa atribut komunikasi TCP dan informasi port tujuan memiliki peran penting dalam mengidentifikasi aktivitas Bot, dengan perbedaan pada urutan tingkat kepentingan fitur.

Berdasarkan Tabel 4.16, hasil interpretasi menggunakan SHAP dan LIME menunjukkan tingkat konsistensi yang cukup baik dalam mengidentifikasi fitur-fitur penting pada ketiga kelas serangan yang dianalisis. Meskipun terdapat perbedaan pada urutan maupun besar kontribusi setiap fitur, kedua metode secara umum menyoroti karakteristik lalu lintas jaringan yang sama sebagai dasar dalam proses klasifikasi.

Pada kelas DoS, baik SHAP maupun LIME menunjukkan bahwa Bwd Packet Length Std, FIN Flag Count, dan Destination Port merupakan fitur-fitur yang paling berpengaruh terhadap keputusan model. Hasil tersebut menunjukkan bahwa variasi panjang paket pada arah komunikasi balik, karakteristik komunikasi TCP, serta informasi mengenai *port* tujuan menjadi indikator utama dalam membedakan aktivitas DoS dari lalu lintas jaringan normal.

Pada kelas PortScan, kedua metode sama-sama mengidentifikasi Packet Length Mean, PSH Flag Count, dan Bwd Packets/s sebagai fitur penting dalam proses klasifikasi. Meskipun urutan tingkat kepentingannya berbeda, SHAP dan LIME tetap memberikan interpretasi yang sejalan bahwa karakteristik ukuran paket, komunikasi TCP, dan intensitas lalu lintas jaringan merupakan faktor utama dalam mengenali aktivitas PortScan.

Sementara itu, pada kelas Bot, SHAP dan LIME sama-sama menunjukkan bahwa Destination Port dan Init_Win_bytes_backward menjadi fitur yang dominan dalam proses pengambilan keputusan model. Hasil tersebut menunjukkan bahwa informasi mengenai port tujuan dan atribut komunikasi TCP memiliki peran penting dalam membedakan aktivitas Bot dari kelas lainnya.

Secara keseluruhan, hasil perbandingan menunjukkan bahwa SHAP dan LIME menghasilkan interpretasi yang saling melengkapi. SHAP memberikan informasi mengenai besarnya kontribusi setiap fitur terhadap hasil prediksi berdasarkan nilai SHAP, sedangkan LIME menjelaskan pengaruh fitur melalui model lokal yang dibangun di sekitar sampel yang dianalisis. Meskipun pendekatan yang digunakan berbeda, kedua metode mampu mengidentifikasi fitur-fitur utama yang relatif konsisten pada setiap kelas serangan. Hal ini menunjukkan bahwa kombinasi SHAP dan LIME dapat meningkatkan transparansi model XGBoost serta memberikan pemahaman yang lebih komprehensif mengenai proses pengambilan keputusan pada Network Intrusion Detection System.

4.6 Diskusi Hasil

4.6.1 Diskusi Performa Model

Hasil evaluasi menunjukkan bahwa ketiga model yang digunakan, yaitu Random Forest, XGBoost, dan MLP, mampu menghasilkan performa klasifikasi yang sangat baik pada dataset CIC-IDS2017. Berdasarkan hasil pengujian, model XGBoost memperoleh nilai *accuracy, precision, recall*, dan F1-*Score* tertinggi dibandingkan dua model lainnya, diikuti oleh Random Forest, sedangkan MLP menghasilkan performa yang sedikit lebih rendah. Temuan tersebut menunjukkan bahwa seluruh model mampu mempelajari karakteristik lalu lintas jaringan dengan baik, namun masing-masing memiliki kemampuan yang berbeda dalam membangun pola klasifikasi.

Performa XGBoost yang lebih tinggi dapat dikaitkan dengan mekanisme gradient boosting yang membangun model secara bertahap melalui proses perbaikan kesalahan (residual) dari model sebelumnya. Pendekatan tersebut memungkinkan XGBoost mempelajari

pola yang kompleks secara lebih efektif sehingga mampu membedakan aktivitas normal dan berbagai jenis serangan dengan tingkat kesalahan yang lebih rendah. Selain itu, penggunaan *ensemble* dari banyak *decision tree* membuat model lebih adaptif dalam menangkap hubungan antarfitur yang beragam pada dataset CIC-IDS2017.

Random Forest juga menunjukkan performa yang sangat kompetitif dengan selisih yang relatif kecil dibandingkan XGBoost. Hal ini menunjukkan bahwa pendekatan berbasis *ensemble tree* efektif dalam menangani data klasifikasi pada NIDS. Random Forest membangun sejumlah *decision tree* secara independen, kemudian menggabungkan hasil prediksi melalui mekanisme *majority voting*. Pendekatan tersebut menghasilkan model yang stabil, mampu mengurangi risiko *overfitting*, serta tetap memberikan tingkat akurasi yang tinggi meskipun menggunakan parameter yang relatif sederhana.

Sementara itu, model MLP memperoleh performa yang sedikit lebih rendah dibandingkan kedua model berbasis *tree ensemble*. Salah satu penyebabnya adalah karakteristik MLP yang sangat bergantung pada proses optimasi bobot selama pelatihan. Meskipun telah dilakukan *feature scaling* menggunakan *StandardScaler*, proses pembelajaran jaringan saraf tetap dipengaruhi oleh konfigurasi arsitektur, parameter pelatihan, serta kompleksitas pola pada dataset. Akibatnya, kemampuan MLP dalam mempelajari karakteristik lalu lintas jaringan pada penelitian ini belum mampu melampaui performa model berbasis *ensemble tree*.

Perbedaan performa tersebut juga didukung oleh hasil confusion matrix. XGBoost menghasilkan jumlah kesalahan klasifikasi yang paling sedikit, sedangkan Random Forest menunjukkan pola kesalahan yang hampir serupa dengan jumlah yang sedikit lebih banyak. Di sisi lain, MLP masih mengalami kesalahan klasifikasi pada beberapa kelas, terutama pada kelas Benign dan WebAttack. Meskipun demikian, seluruh model tetap mampu mempertahankan nilai evaluasi di atas 97%, sehingga dapat dikatakan memiliki kemampuan klasifikasi yang baik pada dataset CIC-IDS2017.

Secara keseluruhan, hasil penelitian menunjukkan bahwa algoritma berbasis *ensemble tree*, khususnya XGBoost, lebih sesuai digunakan pada klasifikasi serangan jaringan menggunakan dataset CIC-IDS2017 dibandingkan model MLP. Temuan ini menjadi dasar dalam pemilihan XGBoost sebagai model yang digunakan pada proses interpretasi menggunakan metode XAI, yaitu SHAP dan LIME, karena model tersebut tidak hanya memberikan performa terbaik, tetapi juga menyediakan hasil prediksi yang lebih representatif untuk dianalisis lebih lanjut.

4.6.2 Diskusi Hasil Interpretasi SHAP dan LIME

Hasil interpretasi menggunakan metode SHAP dan LIME menunjukkan bahwa kedua metode mampu memberikan penjelasan mengenai proses pengambilan keputusan model XGBoost dalam mengklasifikasikan aktivitas lalu lintas jaringan. Berbeda dengan pendekatan *black box* yang hanya menghasilkan prediksi tanpa memberikan alasan di balik keputusan model, penerapan XAI pada penelitian ini memungkinkan identifikasi fitur-fitur yang paling berpengaruh terhadap hasil klasifikasi. Dengan demikian, model tidak hanya dinilai berdasarkan tingkat akurasi yang tinggi, tetapi juga berdasarkan kemampuan menjelaskan alasan di balik setiap prediksi yang dihasilkan.

Hasil interpretasi global menggunakan SHAP menunjukkan bahwa hanya sebagian kecil fitur yang memiliki kontribusi dominan terhadap proses klasifikasi, meskipun dataset CIC-IDS2017 terdiri atas puluhan fitur. Temuan tersebut mengindikasikan bahwa model XGBoost berhasil mempelajari karakteristik lalu lintas jaringan yang paling relevan sehingga proses pengambilan keputusan tidak bergantung pada seluruh fitur secara merata. Fitur-fitur yang berkaitan dengan komunikasi TCP, karakteristik paket, waktu komunikasi, serta informasi mengenai *port* tujuan secara konsisten muncul sebagai faktor yang memberikan pengaruh besar terhadap proses klasifikasi. Kondisi ini menunjukkan bahwa model mampu mengenali pola-pola lalu lintas jaringan yang memang memiliki keterkaitan dengan karakteristik berbagai jenis serangan siber.

Interpretasi lokal menggunakan SHAP memperlihatkan bahwa setiap kelas serangan memiliki kombinasi fitur penting yang berbeda. Pada kelas DoS, keputusan model lebih banyak dipengaruhi oleh karakteristik panjang paket dan komunikasi TCP. Sementara itu, pada kelas PortScan, model lebih menitikberatkan pada karakteristik ukuran paket, jumlah paket, serta aktivitas komunikasi TCP. Adapun pada kelas Bot, informasi mengenai Destination Port menjadi faktor yang paling dominan dalam memengaruhi hasil prediksi. Perbedaan tersebut menunjukkan bahwa model tidak menggunakan satu pola keputusan yang sama untuk seluruh kelas serangan, melainkan membangun mekanisme klasifikasi yang adaptif dalam menyesuaikan karakteristik masing-masing jenis serangan.

Hasil interpretasi menggunakan LIME memberikan temuan yang sejalan dengan SHAP. Meskipun kedua metode menggunakan pendekatan yang berbeda di mana LIME membangun model lokal linier yang sederhana, sedangkan SHAP menggunakan pendekatan teoritis dari *game theory* fitur-fitur utama yang diidentifikasi cenderung menunjukkan konsistensi pada setiap kelas serangan. Penggunaan SHAP dan LIME secara bersamaan memberikan informasi

yang saling melengkapi. SHAP memberikan gambaran komprehensif mengenai tingkat kepentingan fitur baik secara global maupun lokal, sementara LIME memberikan penjelasan yang lebih intuitif pada sampel tunggal.

Lebih jauh lagi, pemahaman teknis dari XAI ini tidak hanya bermanfaat bagi pengembangan model, tetapi juga memiliki nilai praktis dan bisnis yang sangat tinggi apabila diimplementasikan dalam sebuah organisasi. Dari kacamata manajemen dan operasional keamanan siber, integrasi model klasifikasi dengan SHAP dan LIME ddadpat mentransformasi sistem pemantauan menjadi asisten analitik yang transparan bagi tim Security Operations Center (SOC). Terdapat beberapa nilai praktis utama yang dapat dicapai:

1. Efisiensi waktu investigasi (Pengurangan *Downtime*): ketika NIDS mendeteksi adanya serangan, analisis jaringan tidak perlu lagi melakukan penelusuran manual yang memakan waktu untuk mencari anomali yang memicu alarm. Penjelasan dari LIME dan SHAP dapat langsung menyoroti bahwa aktivitas diblokir karena adanya ketidakwajaran pada Destination Port atau ukuran Packet Length. Hal ini memangkas waktu investigasi dari hitungan jam menjadi hitungan menit, sehingga organisasi dapat meminimalisir *downtime* layanan bisnis akibat serangan.
2. Optimalisasi Kebijakan Keamanan (Tindakan Preventif): Pengetahuan mengenai fitur yang paling sering dieksploitasi dalam serangan (*global importance*) dapat digunakan oleh manajamen TI sebagai landasan strategis. Mengetahui bahwa informasi port sangat krusial bagi model dalam mengenali Botnet atau PortScan, administrator jaringan dapat secara proaktif merumuskan dan memperketat kebijakan *firewall* pada *port-port* yang rentan sebelum insiden keamanan benar-benar terjadi.
3. Mengurangi Alert Fatigue (Kelelahan Analis): Sistem pendeteksi intrusi seringkali menghasilkan alarm palsu (*false positive*) yang membuat staf keamanan siber kelelahan dan mengabaikan peringatan penting. Dengan adanya penjelasan lokal yang intuitif, analis pemula sekalipun dapat dengan cepat memvalidasi apakah alarm tersebut merupakan ancaman nyata (misalnya karena flag TCP seperti FIN atau PSH melonjak secara tidak wajar) atau hanya anomali trafik biasa. Pengambilan keputusan menjadi jauh lebih cepat, tepat, dan dapat dipertanggungjawabkan kepada pihak manajemen.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa tujuan penelitian ini telah berhasil dicapai. Penelitian ini menunjukkan bahwa algoritma Random Forest, XGBoost, dan MLP mampu digunakan untuk mendeteksi serangan jaringan menggunakan dataset CIC-IDS2017 dengan performa klasifikasi yang sangat baik. Di antara ketiga algoritma tersebut, XGBoost menghasilkan performa terbaik sehingga dipilih sebagai model utama pada proses interpretasi menggunakan metode XAI. Hasil tersebut menunjukkan bahwa model berbasis ensemble tree memiliki kemampuan yang baik dalam mempelajari karakteristik lalu lintas jaringan pada NIDS.

Penerapan metode SHAP berhasil memberikan interpretasi terhadap keputusan model baik pada tingkat global maupun lokal. Interpretasi global menunjukkan bahwa model memanfaatkan sejumlah fitur utama yang berkaitan dengan karakteristik komunikasi TCP, ukuran paket, laju lalu lintas jaringan, dan informasi *port* tujuan dalam melakukan proses klasifikasi. Sementara itu, interpretasi lokal menunjukkan bahwa setiap kelas serangan memiliki kombinasi fitur yang berbeda dalam memengaruhi keputusan model. Hasil tersebut menunjukkan bahwa SHAP mampu meningkatkan transparansi model dengan menjelaskan alasan di balik setiap hasil prediksi yang dihasilkan.

Selain SHAP, penelitian ini juga menerapkan metode LIME sebagai pembanding interpretasi lokal pada model XGBoost. Hasil perbandingan menunjukkan bahwa SHAP dan LIME memiliki tingkat kesesuaian yang baik dalam mengidentifikasi fitur-fitur penting pada setiap kelas serangan yang dianalisis. Meskipun terdapat perbedaan pada urutan maupun besarnya kontribusi fitur karena mekanisme interpretasi yang berbeda, kedua metode secara konsisten menunjukkan karakteristik lalu lintas jaringan yang relevan sebagai dasar pengambilan keputusan model. Temuan tersebut menunjukkan bahwa SHAP dan LIME saling melengkapi dalam memberikan interpretasi terhadap hasil klasifikasi.

Secara keseluruhan, penelitian ini menunjukkan bahwa penerapan Explainable Artificial Intelligence menggunakan SHAP dan LIME tidak hanya mampu meningkatkan pemahaman terhadap proses pengambilan keputusan model, tetapi juga meningkatkan transparansi pada NIDS. Dengan demikian, model yang dihasilkan tidak hanya memiliki performa klasifikasi

yang baik, tetapi juga mampu memberikan penjelasan yang dapat membantu pengguna memahami alasan di balik setiap keputusan yang dihasilkan.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dipertimbangkan untuk pengembangan penelitian selanjutnya sebagai berikut.

1. Penelitian selanjutnya dapat menggunakan dataset Network Intrusion Detection System lainnya, seperti CSE-CIC-IDS2018, UNSW-NB15, atau dataset yang lebih baru, untuk mengevaluasi konsistensi performa model serta hasil interpretasi menggunakan SHAP dan LIME pada karakteristik lalu lintas jaringan yang berbeda.
2. Penelitian selanjutnya dapat mengeksplorasi algoritma machine learning maupun deep learning lainnya, seperti LightGBM, CatBoost, TabNet, atau arsitektur jaringan saraf yang lebih kompleks, sehingga dapat diperoleh perbandingan performa dan interpretabilitas yang lebih komprehensif pada sistem deteksi intrusi jaringan.
3. Penelitian selanjutnya dapat membandingkan SHAP dan LIME dengan metode Explainable Artificial Intelligence (XAI) lainnya, seperti Integrated Gradients, Anchors, atau Counterfactual Explanations, sehingga dapat diketahui kelebihan dan keterbatasan masing-masing metode dalam menjelaskan keputusan model pada Network Intrusion Detection System.
4. Penelitian selanjutnya dapat mengembangkan implementasi model pada lingkungan Network Intrusion Detection System secara real-time sehingga tidak hanya mengevaluasi performa klasifikasi dan interpretabilitas model pada dataset, tetapi juga menguji penerapannya dalam kondisi operasional yang sesungguhnya.

DAFTAR PUSTAKA

Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1). https://doi.org/10.1002/ett.4150

AL, S., & Sagiroglu, S. (2025). Explainable artificial intelligence models in intrusion detection systems. *Engineering Applications of Artificial Intelligence*, *144*. https://doi.org/10.1016/j.engappai.2025.110145

Alcineide, P., Gean, S., Luiz, M. F. M., Felipe, C. A., & Débora, de G. S. (2021). Cost forecasting of public construction projects using multilayer perceptron artificial neural networks: A case study. *Ingenieria e Investigacion*, *41*(3). https://doi.org/10.15446/ing.investig.v41n3.87737

Arreche, O., Bibers, I., & Abdallah, M. (2024). A Two-Level Ensemble Learning Framework for Enhancing Network Intrusion Detection Systems. *IEEE Access*, *12*, 83830–83857. https://doi.org/10.1109/ACCESS.2024.3407029

Arreche, O., Guntur, T. R., Roberts, J. W., & Abdallah, M. (2024). E-XAI: Evaluating Black-Box Explainable AI Frameworks for Network Intrusion Detection. *IEEE Access*, *12*, 23954–23988. https://doi.org/10.1109/ACCESS.2024.3365140

Bibers, I., & Abdallah, M. (2025). Ensemble Feature Selection for Network Intrusion Detection Systems Using Explainable AI: A Frequency-Based Approach. *2025 IEEE 4th International Conference on Computing and Machine Intelligence, ICMI 2025 - Proceedings*. https://doi.org/10.1109/ICMI65310.2025.11141068

Breiman, L. (2001). *Random Forests* (Vol. 45).

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.1145/2939672.2939785

Corea, P. M., Liu, Y., Wang, J., Niu, S., & Song, H. (2024). Explainable AI for Comparative Analysis of Intrusion Detection Models. *2024 IEEE International Mediterranean Conference on Communications and Networking, MeditCom 2024*, 585–590. https://doi.org/10.1109/MeditCom61057.2024.10621339

EvidentlyAITeam. (2023). *Accuracy, precision, and recall in multi-class classification.*

Farrukh, Y. A., Wali, S., Khan, I., & Bastian, N. D. (2025). XG-NID: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model. *Expert Systems with Applications*, *287*. https://doi.org/10.1016/j.eswa.2025.128089

Gu, X., Howells, G., & Yuan, H. (2024). A soft prototype-based autonomous fuzzy inference system for network intrusion detection. *Information Sciences*, *677*. https://doi.org/10.1016/j.ins.2024.120964

Hasnain, M., Javaid, N., Saudagar, A. K. J., & Kumar, N. (2026). An intelligent and explainable intrusion detection framework for Internet of Sensor Things using generalizable optimized active Machine Learning. *Journal of Network and Computer Applications*, *245*, 104358. https://doi.org/10.1016/j.jnca.2025.104358

Hooshmand, M. K., Huchaiah, M. D., Alzighaibi, A. R., Hashim, H., Atlam, E. S., & Gad, I. (2024). Robust network anomaly detection using ensemble learning approach and explainable artificial intelligence (XAI). *Alexandria Engineering Journal*, *94*, 120–130. https://doi.org/10.1016/j.aej.2024.03.041

Kostas, K., Just, M., & Lones, M. A. (2025). IoTGeM: Generalizable models for behaviour-based IoT attack detection. *Computer Networks*, *270*. https://doi.org/10.1016/j.comnet.2025.111550

Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: A review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159–190. https://doi.org/10.1007/s10462-007-9052-3

Lundberg, S., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. http://arxiv.org/abs/1705.07874

Nada Marey, Ahmad A. Abu-Musa, & Mona Ganna. (2026). Integrating Deep Learning and Explainable Artificial Intelligence Techniques for Stock Price Predictions: An Empirical Study Based on Time Series Big Data. *International Journal of Accounting and Management Sciences*, *3*(4). https://doi.org/10.56830/IJAMS10202401

Nigar, N., Umar, M., Shahzad, M. K., Islam, S., & Abalo, D. (2022). A Deep Learning Approach Based on Explainable Artificial Intelligence for Skin Lesion Classification. *IEEE Access*, *10*, 113715–113725. https://doi.org/10.1109/ACCESS.2022.3217217

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should i trust you?" Explaining the predictions of any classifier. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Sadhwani, S., Navare, A., Mohan, A., Muthalagu, R., & Pawar, P. M. (2025). IoT-based intrusion detection system using explainable multi-class deep learning approaches. *Computers and Electrical Engineering*, *123*. https://doi.org/10.1016/j.compeleceng.2025.110256

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSP 2018 - Proceedings of the 4th International Conference on Information Systems Security and Privacy*, *2018-January*, 108–116. https://doi.org/10.5220/0006639801080116

Sharma, B., Sharma, L., & Lal, C. (2022). Anomaly Based Network Intrusion Detection for IoT Attacks using Convolution Neural Network. *2022 IEEE 7th International Conference for Convergence in Technology, I2CT 2022*. https://doi.org/10.1109/I2CT54291.2022.9824229

Shoukat, S., Gao, T., Javeed, D., Saeed, M. S., & Adil, M. (2025). Trust my IDS: An explainable AI integrated deep learning-based transparent threat detection system for industrial networks. *Computers and Security*, *149*. https://doi.org/10.1016/j.cose.2024.104191

Sison, A., Claudy, M., Leicht-Deobald, U., & Leben, D. (2023). *Explainable AI as evidence of fair decisions*.

Wakili, A., & Bakkali, S. (2025). A resilient IoT intrusion detection system using hybrid feature selection and explainable ensemble learning. *Results in Engineering*, *28*. https://doi.org/10.1016/j.rineng.2025.107392

Wali, S., Farrukh, Y. A., & Khan, I. (2025). Explainable AI and Random Forest based reliable intrusion detection system. *Computers and Security*, *157*. https://doi.org/10.1016/j.cose.2025.104542

LAMPIRAN

Link Repository Kode Eksperimen: https://github.com/vitooo2/XAI-IDS-CICIDS2017