

Model Deteksi DDoS Berbasis *Machine Learning* yang Efisien, Interpretable, dan Siap Implementasi Operasional

Andri Yudha Pratama^{*1}, Khalifatur Rauf², Enny Itje Sela³

^{1,2,3}Magister Teknologi Informasi, Universitas Teknologi Yogyakarta, Indonesia

Email: ¹pratama.andriyudha@gmail.com, ²blograuf.kr@gmail.com, ³ennysela@uty.ac.id

Abstrak

Serangan *Distributed Denial of Service* (DDoS) menjadi ancaman serius bagi kontinuitas bisnis digital, sehingga membutuhkan sistem deteksi yang akurat, responsif, dan dapat diinterpretasikan. Sebagian besar penelitian terdahulu berfokus pada maksimalisasi akurasi melalui model kompleks, namun kerap mengabaikan efisiensi komputasi dan *actionability* yang esensial bagi implementasi *real-time*. Penelitian ini mengevaluasi sembilan skenario deteksi pada dataset CIC-DDoS2019 melalui kombinasi metode seleksi fitur (Pearson, ANOVA, RFE) dan algoritma *machine learning* (*Decision Tree*, *Random Forest*, *Logistic Regression*). Hasilnya mengungkapkan adanya *trade-off* signifikan antara kompleksitas model dan latensi deteksi. Penelitian ini mengidentifikasi Skenario E4 (RFE + *Decision Tree*) sebagai model terbaik berdasarkan *trade-off* akurasi, latensi, dan memori, dengan *recall* serangan 0,9999, latensi 900 μ s (sekitar 38 kali lebih cepat dari *Random Forest*), dan efisiensi memori 5.760 Byte. Kontribusi utama penelitian ini mencakup evaluasi multi-objektif yang mengintegrasikan akurasi, latensi, memori, interpretabilitas, dan robustness; pemetaan fitur SHAP ke dalam matriks mitigasi *Defense-in-Depth*; serta bukti empiris *trade-off* antara efisiensi operasional dan ketahanan model terhadap serangan adaptif. Analisis SHAP menunjukkan keputusan model didasarkan pada fitur identitas, anomali TCP *flag*, dan pola *idle time*. Namun, uji *robustness* mengindikasikan kerentanan terhadap manipulasi input, menegaskan perlunya strategi mitigasi tambahan dalam kerangka *Defense-in-Depth* agar model tidak hanya unggul secara statistik, tetapi juga operasional dan adaptif terhadap ancaman cerdas.

Kata kunci: Deteksi DDoS, *Machine Learning*, *Recursive Feature Elimination* (RFE), *Decision Tree*, *Explainable AI* (SHAP), *Actionability*.

Abstract

Distributed Denial of Service (DDoS) attacks pose a serious threat to digital business continuity, necessitating detection systems that are accurate, responsive, and interpretable. Most prior research has focused on maximizing accuracy through complex models, often neglecting computational efficiency and *actionability* essential for *real-time* implementation. This study evaluates nine detection scenarios using the CIC-DDoS2019 dataset by combining feature selection methods (Pearson, ANOVA, RFE) and machine learning algorithms (*Decision Tree*, *Random Forest*, *Logistic Regression*). The results reveal a significant *trade-off* between model complexity and detection latency. The results reveal a significant *trade-off* between model complexity and detection latency. This study identifies Scenario E4 (RFE + *Decision Tree*) as the best model based on the *trade-off* between accuracy, latency, and memory, achieving an attack recall of 0.9999, a latency of 900 μ s (approximately 38 times faster than *Random Forest*), and memory efficiency of 5,760 bytes. The main contributions of this study encompass a multi-objective evaluation integrating accuracy, latency, memory, interpretability, and robustness; mapping SHAP features into a *Defense-in-Depth* mitigation matrix; and empirical evidence of the *trade-off* between operational efficiency and model robustness against adaptive attacks. SHAP analysis shows that the model's decisions are based on identity features, TCP flag anomalies, and idle time patterns. However, robustness testing indicates vulnerability to input manipulation, emphasizing the need for additional mitigation strategies within a *Defense-in-Depth* framework to ensure the model is not only statistically superior but also operationally adaptive to intelligent threats.

Keywords: DDoS Detection, *Machine Learning*, *Recursive Feature Elimination* (RFE), *Decision Tree*, *Explainable AI* (SHAP), *Actionability*.

This work is an open access article and licensed under a Creative Commons Attribution-NonCommercial ShareAlike 4.0 International (CC BY-NC-SA 4.0)



1. PENDAHULUAN

Salah satu jenis serangan siber yang dikenal sebagai *Distributed Denial of Service* (DDoS) dilakukan dengan membanjiri jaringan atau *server* target dengan lalu lintas palsu yang sangat besar, membuat sumber daya sistem kewalahan, dan layanan yang seharusnya dapat diakses oleh pengguna yang sah terhambat atau bahkan tidak dapat diakses sama sekali [1], [2]. Serangan ini dapat menyebabkan masalah teknis dan kerusakan finansial serta kehancuran citra perusahaan dan kepercayaan pelanggan. sehingga, dapat terjadi pengurangan yang signifikan dalam kontinuitas operasi layanan digital [3]. *Distributed Denial of Service* (DDoS) menjadi ancaman strategis yang penting bagi komunitas keamanan jaringan dan para pemimpin bisnis yang bertanggung jawab atas keberlanjutan layanan dan stabilitas keuntungan perusahaan.

Teknik *machine learning* telah digunakan dalam beberapa penelitian untuk membuat sistem deteksi *Distributed Denial of Service* (DDoS) pada berbagai *dataset*. CIC-DDoS2019 adalah salah satu *dataset* yang banyak digunakan sebagai benchmark dalam literatur. Dari perspektif algoritma, metode seperti *Neural Networks*, *Random Forest*, *Naïve Bayes*, *Decision Tree*, dan *Ensemble Learning* telah diuji dan menunjukkan kemampuan yang sangat baik untuk membedakan antara lalu lintas normal dan trafik serangan DDoS [2], [4], [5], [6]. Studi sebelumnya menunjukkan bahwa algoritma *Random Forest* paling akurat dalam mendeteksi serangan *Distributed Denial of Service* (DDoS) dibandingkan dengan model klasifikasi lainnya [2]. Namun demikian, penelitian tersebut hanya berfokus pada karakteristik kinerja klasifikasi, tanpa memeriksa aspek lain yang berkontribusi pada keberhasilan model secara keseluruhan. Selain itu, analisis terhadap pengoptimalan pemilihan fitur dan pengaturan *hyperparameter* menunjukkan bahwa kedua metode tersebut memiliki dampak yang signifikan terhadap peningkatan kinerja model [2]. Elemen efisiensi komputasi seperti kompleksitas waktu pelatihan dan kebutuhan sumber daya sistem belum sepenuhnya dibahas oleh banyak penelitian. Sehingga, pengembangan model yang tidak hanya akurat tetapi juga efisien secara operasional masih menjadi masalah yang perlu diteliti lebih lanjut.

Kajian sistematis sebelumnya menemukan banyak masalah penting dalam pengembangan deteksi DDoS berbasis *machine learning*, terutama berkaitan dengan kualitas dan keragaman *dataset*, kebutuhan untuk melakukan deteksi secara *real-time*, dan kompleksitas komputasi yang tinggi. Metode empiris saat ini belum sepenuhnya menangani masalah ini. Hasilnya menunjukkan bahwa aspek implementasi praktis khususnya yang berkaitan dengan kebutuhan operasional dan keberlanjutan bisnis masih kurang diperhatikan dalam penelitian terdahulu, meskipun banyak kemajuan teknis telah dicapai dalam hal akurasi dan kinerja model [7].

Temuan serupa diperkuat oleh berbagai penelitian yang meninjau berbagai cara untuk mendeteksi DDoS. Meskipun teknik *machine learning* sangat akurat, sebagian besar penelitian belum membahas aspek kecepatan pemrosesan dan kesiapan implementasi pada sistem operasional yang membutuhkan respon *real-time* [2], [3]. Berbagai penelitian mulai menggunakan *deep learning* sebagai metode lanjutan untuk mengatasi keterbatasan klasifikasi klasik seiring perkembangan metode deteksi. Ada bukti bahwa model seperti *Convolutional Neural Network* (CNN) dapat dengan akurat menemukan pola lalu lintas berbahaya pada *dataset* DDoS konvensional. Namun, implementasi *real-time* memerlukan proses optimasi yang cermat [8], [9]. Penggabungan model *hybrid*, seperti CNN dengan *self-attention*, dapat digunakan untuk meningkatkan kemampuan fitur dan relevansi klasifikasi pada berbagai serangan [10].

Penelitian di bidang *deep learning* menunjukkan bahwa pendekatan ini secara statistik lebih efektif dalam menangkap kompleksitas pola data dibandingkan sejumlah metode tradisional [11]. Sebagian besar studi masih menempatkan akurasi sebagai indikator utama keberhasilan model. Sementara itu, aspek interpretabilitas dan efisiensi implementasi belum banyak dibahas secara komprehensif dalam literatur, khususnya dalam konteks penerapan operasional. Sebagian besar penelitian melaporkan akurasi, *precision*, dan *F-score* sebagai metrik utama dalam evaluasi kinerja model. Sedangkan, aspek waktu deteksi (*latency*), *recall* pada kelas serangan, efisiensi sumber daya komputasi, serta interpretabilitas model masih belum banyak dikaji secara terintegrasi dengan kebutuhan pengambilan keputusan organisasi.

Berbagai penelitian menunjukkan tingkat akurasi yang tinggi, analisis kuantitatif terhadap risiko *false negative* yang dapat mengakibatkan kerugian bagi perusahaan masih sangat terbatas. Latensi deteksi mempengaruhi kecepatan respons sistem sebelum layanan terpengaruh ketika proses operasional

dijalankan. Selain itu, mengingat *recall* serangan sangat penting untuk mengurangi *false negative* karena kegagalan untuk mengidentifikasi serangan dapat menyebabkan kerugian keuangan dan reputasi perusahaan [12], [13]. Kemampuan model untuk diinterpretasikan sangat penting ketika hasil deteksi digunakan oleh pengambil keputusan dan tim keamanan untuk membuat rencana mitigasi risiko yang efektif. Menurut penelitian literatur, ada kesenjangan yang jelas antara kemajuan teknis dalam mendeteksi *Distributed Denial of Service* (DDoS) yang lebih berfokus pada pencapaian akurasi tinggi dan kebutuhan praktis organisasi yang menuntut model yang cepat, efisien, mudah dijelaskan (*explainable*), dan dapat ditindaklanjuti (*actionable*). *Dataset* CIC-DDoS2019 dipilih untuk studi ini karena mencakup berbagai jenis serangan DDoS dan menyediakan fitur yang relevan untuk mengevaluasi kinerja model secara menyeluruh [14].

Penelitian ini bertujuan menjembatani kesenjangan antara capaian akurasi statistik dan kebutuhan operasional bisnis dalam deteksi DDoS dengan mengevaluasi pendekatan yang menyeimbangkan kinerja, efisiensi, dan interpretabilitas. Menggunakan *dataset* CIC-DDoS2019 yang merepresentasikan karakteristik serangan modern, studi ini tidak hanya mengidentifikasi model dengan akurasi tertinggi, tetapi juga menilai efisiensi komputasi dan kemudahan interpretasi hasil. Penelitian ini mengevaluasi *trade-off* antara akurasi, latensi deteksi, dan efisiensi memori pada berbagai kombinasi metode seleksi fitur (Pearson, RFE, ANOVA) dan algoritma klasifikasi (*Decision Tree*, *Random Forest*, *Logistic Regression*), sekaligus memperkuat aspek *actionability* melalui analisis *SHapley Additive exPlanations* (SHAP) untuk memetakan fitur teknis ke dalam matriks aturan mitigasi yang dapat dieksekusi dalam strategi *Defense-in-Depth*. Selain itu, dilakukan pengujian *robustness* terhadap model *lightweight* melalui *adversarial attack* guna mengidentifikasi batas ketahanan model dalam menghadapi manipulasi input pada skenario serangan yang lebih canggih. Berangkat dari serangkaian evaluasi tersebut, kontribusi utama penelitian ini terletak pada kemampuannya mengidentifikasi arsitektur *lightweight* yang paling optimal dalam menyeimbangkan sensitivitas deteksi dengan efisiensi operasional. Lebih jauh, penelitian ini turut menyumbangkan kerangka *actionability* konkret berupa matriks mitigasi keamanan berlapis yang ditranslasikan langsung dari fitur prediktif model, serta menyajikan bukti empiris mengenai *trade-off* krusial antara kecepatan sistem deteksi dengan ketahanannya terhadap manipulasi ancaman cerdas.

Berdasarkan kesenjangan tersebut, penelitian ini memberikan tiga kontribusi utama. Pertama, penelitian ini menyusun evaluasi multi-objektif untuk deteksi DDoS yang mengintegrasikan akurasi, *recall* serangan, latensi, penggunaan memori, interpretabilitas, dan *robustness*. Kedua, penelitian ini memetakan hasil interpretabilitas SHAP ke dalam matriks mitigasi yang dapat digunakan secara langsung dalam strategi *Defense-in-Depth*. Ketiga, penelitian ini menguji ketahanan model *lightweight* terhadap *adversarial attack* untuk menunjukkan *trade-off* antara efisiensi operasional dan *robustness* model.

1.2. Kesenjangan Penelitian (*state of the art gap*)

Tabel 1 merangkum penelitian terdahulu yang relevan dan mengidentifikasi kesenjangan yang menjadi landasan penelitian ini. Sebagian besar studi yang menggunakan *dataset* CIC-DDoS2019 hanya melaporkan metrik klasifikasi konvensional (akurasi, *precision*, *recall*, *F1-score*) tanpa mempertimbangkan latensi inferensi, efisiensi memori, interpretabilitas berbasis SHAP, atau ketahanan terhadap *adversarial attack*. Ketiadaan evaluasi multi-objektif ini menciptakan kesenjangan antara capaian teknis dan kebutuhan implementasi operasional bisnis digital.

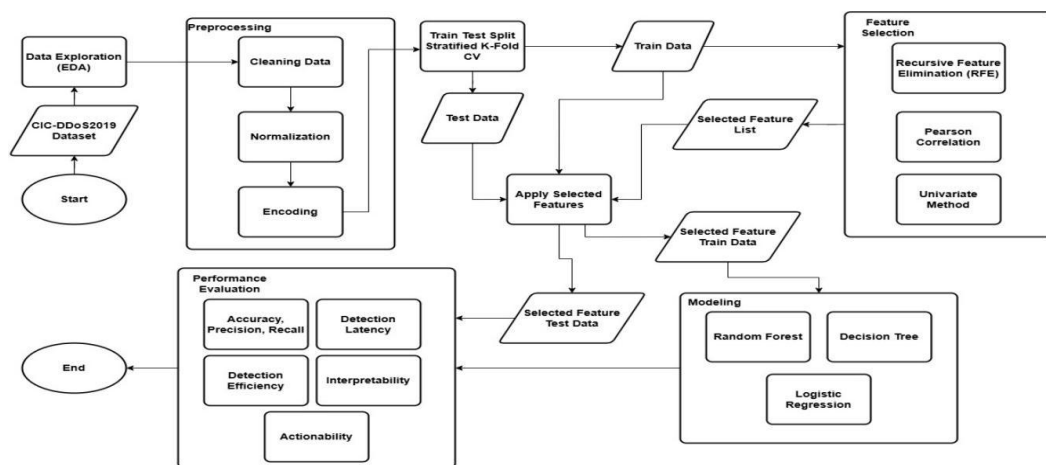
Tabel 1. Perbandingan Penelitian Terdahulu dan Kesenjangan Penelitian

Penelitian	Dataset	Algoritma	Akurasi	Latency/Efisiensi	Keterbatasan
Sari et al. [2]	CIC-DDoS2019	Random Forest, Decision Tree, Naïve Bayes	$\geq 99\%$	Tidak dilaporkan	Tidak membahas latensi, efisiensi memori, dan interpretabilitas
Alfatih et al. [5]	CIC-DDoS2019	Machine Learning + Hyperparameter Tuning	$\geq 98\%$	Tidak dilaporkan	Fokus pada hyperparameter, Tidak melakukan uji ketahanan

					(<i>robustness</i>) terhadap serangan manipulasi input (<i>adversarial</i>)
Alduailij et al. [15]	CIC-DDoS2019	Random Forest + Mutual Information	$\geq 99\%$	Tidak dilaporkan	Model bersifat ensemble yang kompleks, mengakibatkan latensi tinggi yang tidak efisien untuk <i>real-time</i>
Sharma [1]	Berbagai dataset	DL, RF, SVM	Bervariasi	Tidak dilaporkan	Tidak mengintegrasikan SHAP dan mitigasi berlapis
Penelitian ini	CIC-DDoS2019	DT, RF, LR + RFE/Pearson/ANOVA	$\geq 99,94\%$	Keseimbangan multi-objektif: Akurasi (0.9999), Latensi (900 μ s), dan Memori (5.760 B).	Multi-objektif: akurasi, latensi, memori, SHAP, <i>robustness adversarial</i> , matriks mitigasi

2. METODE PENELITIAN

2.1. Tahapan Penelitian



Gambar 1. Diagram Alur Tahapan Penelitian

Gambar 1 menunjukkan alur penelitian yang dilakukan dalam studi ini, mulai dari pemilihan *dataset* hingga evaluasi performansi model. Penelitian dimulai dengan pemilihan CIC-DDoS2019, yang dipilih karena mencakup berbagai tipe serangan DDoS dan menyediakan fitur yang relevan untuk evaluasi performa model secara komprehensif [14]. Selanjutnya dilakukan eksplorasi data (EDA) untuk memahami distribusi data, mendeteksi outlier, memeriksa ketidakseimbangan kelas, dan mengidentifikasi kualitas fitur yang tersedia.

Seluruh eksperimen dilaksanakan pada lingkungan komputasi dengan spesifikasi sebagai berikut: prosesor Intel(R) Core(TM) i5-10400 CPU (2.90 GHz), RAM 16 GB, dan sistem operasi Windows 11 Pro yang dikombinasikan dengan platform komputasi awan Google Colab untuk mendukung fleksibilitas eksekusi kode berbasis notebook. Perangkat lunak yang digunakan meliputi Python 3.10, scikit-learn 1.3.0, SHAP 0.42.1, serta Adversarial Robustness Toolbox (ART) 1.16.0. Seluruh eksperimen dijalankan menggunakan random seed yang ditetapkan secara konsisten (`random_state = 42`) pada seluruh algoritma dan prosedur pemilahan data guna menjamin reproduktibilitas hasil.

Tahap berikutnya adalah *preprocessing*. Data dibersihkan dari nilai hilang dan duplikasi, fitur non-numerik di-encode, dan semua fitur numerik dinormalisasi. Pada tahap ini, berbagai metode *feature selection* dieksplorasi, termasuk *Recursive Feature Elimination* (RFE), *Pearson Correlation*, dan *Univariate Feature Selection*, untuk memilih fitur paling relevan bagi model, sekaligus mengurangi *overhead* komputasi dan meningkatkan akurasi prediksi. Setelah *preprocessing* dan *feature selection*, *dataset* dibagi menjadi set pelatihan dan pengujian menggunakan *Stratified K-Fold Cross Validation*. Dalam penelitian ini, nilai K ditetapkan sebesar 5 ($k=5$), sehingga pada setiap iterasi, 80% data digunakan untuk pelatihan dan 20% untuk pengujian. Penetapan nilai $K=5$ dipilih karena memberikan keseimbangan yang baik antara varians estimasi dan efisiensi komputasi, khususnya mengingat ukuran *dataset* yang relatif besar (225.745 sampel). Distribusi kelas serangan tetap terjaga pada setiap lipatan berkat stratifikasi, sehingga estimasi performa yang dihasilkan bersifat stabil dan reliabel.

Langkah berikutnya adalah tahap pemodelan, di mana penelitian ini mengevaluasi beberapa algoritma *machine learning* dari paradigma yang berbeda. *Decision Tree* digunakan sebagai model berbasis aturan, *Random Forest* sebagai model *ensemble* berbasis pohon, dan *Logistic Regression* sebagai model linier. Pendekatan ini bertujuan untuk memastikan bahwa analisis interpretabilitas model tidak terbatas pada satu jenis algoritma tertentu, sehingga dapat memberikan perspektif yang lebih luas terhadap perilaku model dalam mendeteksi serangan DDoS. Ketiga algoritma tersebut digunakan sebagai model utama dan dievaluasi menggunakan konfigurasi parameter dasar (*baseline parameter*) melalui skema *cross-validation* untuk memperoleh performansi yang stabil dan reliabel sehingga hasil evaluasi tidak bergantung pada satu konfigurasi data tertentu.

Tahap terakhir adalah evaluasi performansi model. Metrik yang digunakan tidak hanya mencakup akurasi, *precision*, dan *recall*, tetapi juga menekankan aspek bisnis yang kritis, seperti *latency* (waktu deteksi serangan), *recall* serangan untuk meminimalkan *false negative*, efisiensi sumber daya komputasi, interpretabilitas model, dan *actionability*, yaitu kemampuan hasil deteksi digunakan untuk pengambilan keputusan mitigasi risiko secara langsung. Dengan alur ini, penelitian dapat memastikan bahwa model tidak hanya akurat, tetapi juga siap diterapkan dalam lingkungan bisnis digital.

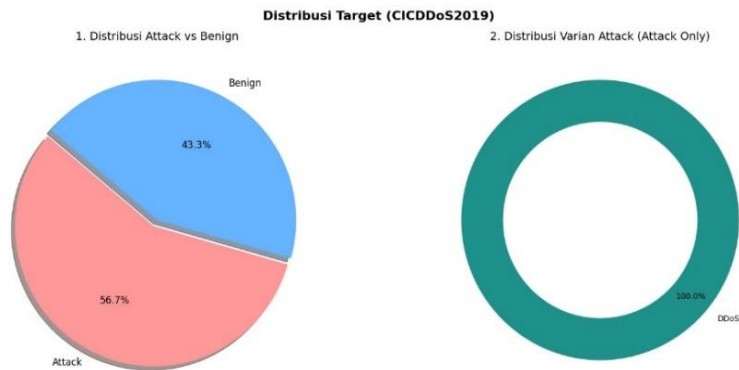
Latensi deteksi (*single flow latency*) didefinisikan sebagai waktu rata-rata yang dibutuhkan model untuk menghasilkan prediksi terhadap satu sampel data tunggal, dihitung dalam satuan mikro-detik (μs). Pengukuran dilakukan melalui inferensi sebanyak 1.000 iterasi pada sampel acak dari data uji, dengan fase pemanasan (*warm-up*) sebanyak 100 iterasi awal yang tidak diikutsertakan dalam kalkulasi rata-rata, guna menghindari bias akibat inisialisasi awal sistem. *Batch time* mengacu pada total waktu inferensi untuk seluruh data uji dalam satu *fold*. Penggunaan memori diukur sebagai jejak memori (*memory footprint*) dari objek model setelah proses pelatihan, menggunakan fungsi `sys.getsizeof()` pada Python, yang mengukur ukuran objek model termasuk parameter internal dan struktur pohon keputusan.

2.2. Dataset dan Exploratory Data Analysis (EDA)

Penelitian ini menggunakan *dataset* CIC-DDoS2019 sebagai sumber data utama untuk eksperimen deteksi serangan DDoS. *Dataset* ini dikembangkan oleh *Canadian Institute for Cybersecurity* dan dirancang untuk merepresentasikan lalu lintas jaringan modern dengan berbagai skenario serangan DDoS yang realistis [14]. CIC-DDoS2019 mencakup beragam tipe serangan seperti *SYN Flood*, *UDP Flood*, *DNS Amplification*, *LDAP*, *MSSQL*, dan *NetBIOS*, serta lalu lintas normal, sehingga memungkinkan evaluasi model deteksi secara komprehensif pada berbagai karakteristik serangan.

Dataset terdiri dari sekumpulan *flow* jaringan yang diekstraksi menggunakan *CICFlowMeter*, di mana setiap *flow* direpresentasikan oleh sejumlah fitur statistik yang menggambarkan perilaku lalu lintas jaringan, seperti durasi koneksi, jumlah paket, ukuran byte, dan karakteristik temporal. Label kelas diberikan untuk membedakan antara lalu lintas normal dan serangan DDoS, sehingga *dataset* ini sesuai untuk pendekatan *supervised learning*. Keberagaman fitur dan tipe serangan dalam CIC-DDoS2019 menjadikannya sebagai *benchmark* yang banyak digunakan dalam penelitian deteksi DDoS berbasis *machine learning* dan *deep learning*.

Tahap *Exploratory Data Analysis* (EDA) dilakukan untuk memperoleh pemahaman awal terhadap karakteristik data sebelum masuk ke tahap *preprocessing* dan pemodelan. Analisis diawali dengan pemeriksaan distribusi kelas menggunakan total *dataset* sebanyak 225.745 sampel data. Sebagaimana diilustrasikan pada Gambar 2, *dataset* menunjukkan distribusi kelas yang relatif seimbang.



Gambar 2. Distribusi Target *Dataset* CIC-DDoS2019: (1) Proporsi kelas *Attack* vs *Benign*, (2) Pengelompokan varian serangan.

Diagram lingkaran pertama memperlihatkan bahwa kelas serangan (*Attack*) memiliki proporsi 56,7%, sedangkan lalu lintas normal (*Benign*) mencakup 43,3% dari keseluruhan data. Diagram kedua menegaskan bahwa seluruh varian serangan dikelompokkan ke dalam satu label target utama, yaitu DDoS (100%). Kondisi keseimbangan data ini sangat menguntungkan bagi proses pelatihan model, karena dapat meminimalkan risiko bias model terhadap kelas mayoritas (*majority class bias*) tanpa memerlukan teknik resampling yang agresif.

Selanjutnya dilakukan analisis statistik deskriptif terhadap fitur numerik untuk mengamati rentang nilai, mean, median, serta standar deviasi. Analisis ini mengindikasikan adanya variasi skala yang signifikan antar fitur, khususnya pada fitur berbasis volume dan waktu, yang memperkuat kebutuhan akan proses normalisasi pada tahap *preprocessing*. Selain itu, visualisasi distribusi data dan *boxplot* digunakan untuk mendeteksi keberadaan *outlier*, yang dapat muncul akibat aktivitas serangan dengan karakteristik lalu lintas ekstrem.

EDA juga mencakup analisis korelasi antar fitur menggunakan *Pearson Correlation* untuk mengidentifikasi fitur yang memiliki hubungan kuat satu sama lain maupun terhadap label kelas. Hasil analisis ini menunjukkan adanya fitur-fitur yang bersifat redundan, sehingga berpotensi meningkatkan kompleksitas model tanpa memberikan kontribusi signifikan terhadap performa deteksi. Temuan ini menjadi dasar dalam pemilihan dan penerapan metode *feature selection* pada tahap selanjutnya.

Secara keseluruhan, hasil EDA memberikan insight penting terkait struktur data, distribusi kelas, skala fitur, serta potensi redundansi fitur. Insight ini digunakan untuk merancang strategi *preprocessing*, *feature selection*, dan skenario eksperimen yang selaras dengan tujuan penelitian, yaitu membangun model deteksi DDoS yang tidak hanya akurat, tetapi juga efisien, *interpretable*, dan relevan untuk konteks implementasi bisnis digital.

2.3. *Preprocessing* dan *Feature Selection*

Tahap *preprocessing* dilakukan untuk memastikan bahwa data yang digunakan dalam pemodelan memiliki kualitas yang memadai dan sesuai dengan asumsi algoritma *machine learning* yang diterapkan. Berdasarkan hasil *Exploratory Data Analysis* (EDA), *dataset* CIC-DDoS2019 menunjukkan variasi skala fitur yang signifikan serta keberadaan nilai ekstrem pada beberapa fitur berbasis volume dan waktu. Oleh karena itu, *preprocessing* menjadi tahap penting untuk meningkatkan stabilitas dan reliabilitas model deteksi DDoS. Proses *preprocessing* diawali dengan pembersihan data, termasuk penghapusan duplikasi dan penanganan nilai hilang. Nilai hilang pada fitur numerik ditangani menggunakan imputasi berbasis nilai median untuk menjaga distribusi data dan mengurangi pengaruh *outlier*.

Selanjutnya, fitur non-numerik diubah ke dalam representasi numerik agar dapat diproses oleh algoritma *machine learning*. Seluruh fitur numerik kemudian dinormalisasi menggunakan metode *Min-Max Scaling*, sehingga setiap fitur berada pada rentang nilai yang sama. Normalisasi ini bertujuan untuk mencegah dominasi fitur dengan skala besar dan meningkatkan kinerja model, khususnya pada algoritma yang sensitif terhadap skala data seperti *Logistic Regression*. Setelah *preprocessing* dasar, penelitian ini menerapkan beberapa metode *feature selection* untuk mengurangi dimensi data dan menghilangkan fitur yang kurang relevan atau bersifat redundan. Pendekatan pertama menggunakan

Pearson Correlation untuk mengukur hubungan linear antara setiap fitur dan label kelas. Fitur dengan nilai korelasi yang sangat rendah terhadap label atau memiliki korelasi tinggi dengan fitur lain diidentifikasi sebagai kandidat untuk dieliminasi [16]. Pendekatan ini membantu menyederhanakan model dan mengurangi multikolinearitas tanpa mengorbankan informasi yang signifikan.

Pendekatan kedua adalah *Recursive Feature Elimination* (RFE), yang merupakan metode *wrapper* berbasis model [17]. Dalam implementasinya, RFE dikonfigurasi untuk menyeleksi tepat 50 fitur terbaik ($N=50$). Untuk menjaga konsistensi antara ekstraksi fitur dan klasifikasi akhir, estimator pada algoritma RFE disesuaikan dengan algoritma prediksinya: menggunakan *Decision Tree* untuk skenario berbasis pohon (DT dan RF), serta *Logistic Regression* untuk skenario linier (LR). Melalui proses iteratif, fitur dengan kontribusi terendah terhadap performa model dieliminasi secara bertahap hingga diperoleh *subset* fitur optimal. Metode ini memungkinkan evaluasi fitur berdasarkan dampak langsungnya terhadap kinerja model, sehingga lebih adaptif terhadap karakteristik data DDoS yang kompleks.

Sebagai pendekatan *univariate*, penelitian ini menggunakan ANOVA *F-test* untuk mengevaluasi signifikansi statistik masing-masing fitur terhadap label kelas. ANOVA *F-test* dipilih dibandingkan *chi-square* karena sebagian besar fitur dalam CIC-DDoS2019 bersifat kontinu dan terdistribusi numerik, sedangkan *chi-square* lebih sesuai untuk fitur kategorikal dan data diskret. ANOVA *F-test* mengukur perbedaan rata-rata nilai fitur antar kelas dan mengidentifikasi fitur yang memiliki daya diskriminasi tinggi terhadap serangan DDoS [18].

Daftar 50 fitur yang terpilih dari masing-masing metode seleksi bersifat spesifik terhadap *dataset* dan estimator yang digunakan. Secara umum, fitur-fitur yang dominan terpilih mencakup: karakteristik bendera TCP (SYN Flag Count, ACK Flag Count, PSH Flag Count), statistik waktu antarpaket (Flow IAT Mean, Idle Mean, Active Std), karakteristik volume (Packet Length Std, Subflow Fwd Bytes), serta fitur identitas jaringan (Src IP, Dst Port) yang ditemukan relevan secara statistik dalam konteks *dataset* CIC-DDoS2019.

Sehingga lebih selaras dengan karakteristik *dataset* dan tujuan klasifikasi biner yang digunakan dalam penelitian ini. Serupa dengan pendekatan *Pearson* dan RFE, seleksi fitur menggunakan ANOVA dikonfigurasi untuk mengekstrak tepat 50 fitur dengan skor signifikansi tertinggi ($N=50$). Penetapan batas fitur yang konstan pada seluruh metode seleksi ini secara sengaja diterapkan guna memastikan bahwa perbandingan beban komputasi baik penggunaan memori maupun latensi antar skenario dapat dilakukan secara setara. Hasil dari ketiga metode *feature selection* tersebut kemudian dibandingkan untuk mengevaluasi *trade-off* antara kualitas *subset* fitur yang terpilih, performa deteksi, dan efisiensi komputasi. *Subset* fitur yang dihasilkan dari masing-masing metode diuji pada tahap pemodelan menggunakan skema *cross-validation* yang sama, sehingga perbandingan performa dapat dilakukan secara adil dan konsisten. Pendekatan ini memungkinkan analisis yang lebih mendalam mengenai pengaruh strategi *feature selection* terhadap akurasi, *recall* serangan, *latency*, dan interpretabilitas model, sekaligus mendukung tujuan penelitian dalam membangun sistem deteksi DDoS yang efisien dan aplikatif dalam konteks bisnis.

2.4. Algoritma dan Pemodelan Deteksi DDoS

Tujuan dari tahap pemodelan adalah untuk membuat dan mengevaluasi model deteksi DDoS berbasis *machine learning* yang baik untuk klasifikasi dan memberi interpretasi dan dukungan untuk pengambilan keputusan bisnis. Untuk tujuan ini, tiga algoritma dari berbagai paradigma *Decision Tree*, *Random Forest*, dan *Logistic Regression* dipilih untuk mengimbangi performa, kompleksitas model, dan tingkat interpretabilitas hasil prediksi [1], [11].

Decision Tree digunakan sebagai model berbasis aturan yang merepresentasikan proses pengambilan keputusan dalam bentuk struktur pohon. Model ini dipilih karena kemampuannya dalam menghasilkan aturan klasifikasi yang mudah dipahami dan ditelusuri, sehingga mendukung aspek interpretabilitas dan *explainability*. *Decision Tree* memudahkan analisis hubungan sebab-akibat yang penting bagi tim keamanan jaringan dalam deteksi DDoS dan membantu menemukan fitur penting yang membedakan lalu lintas serangan dari yang normal [6], [19]. Model *ensemble* berbasis pohon *Random Forest* menggabungkan beberapa *Decision Tree* untuk meningkatkan stabilitas dan akurasi prediksi. Metode ini mampu mengurangi risiko *overfitting* dan menangkap pola *non-linear* yang kompleks pada trafik DDoS melalui pendekatan pembelajaran kelompok. Selain itu, *Random Forest* memberikan

informasi tentang nilai fitur yang mendukung interpretasi model dan identifikasi faktor dominan dalam deteksi serangan [2], [15].

Untuk membandingkan elemen interpretabilitas dan efisiensi komputasi, *logistic* regresi digunakan sebagai model linier. Dengan menggunakan koefisien regresi yang jelas untuk menunjukkan hubungan antara fitur dan kemungkinan serangan DDoS, model ini memungkinkan analisis yang jelas dari kontribusi setiap variabel terhadap keputusan klasifikasi. Implementasi dalam skenario deteksi lingkungan dan *real-time* dengan keterbatasan sumber daya juga didukung oleh kompleksitasnya yang rendah. *Dataset* yang telah melalui *preprocessing* dan pemilihan fitur dilatih untuk seluruh algoritma. Untuk menjaga proporsi kelas pada setiap lipatan dan menghasilkan estimasi performa yang stabil, skema *Stratified K-Fold Cross Validation* digunakan. Guna memastikan perbandingan yang adil terhadap dampak metode seleksi fitur dan arsitektur algoritma, eksperimen ini mengevaluasi performa dasar (*baseline performance*) dari setiap model menggunakan parameter bawaannya (*default parameters*). Pendekatan ini dipilih agar hasil pengukuran efisiensi komputasi khususnya pada latensi dan memori murni mencerminkan karakteristik bawaan arsitektur model, metode ini memungkinkan analisis komparatif secara menyeluruh. Sehingga, rekomendasi yang dihasilkan tidak hanya bergantung pada performa teknis tetapi juga mempertimbangkan relevansi implementasi dalam konteks operasional dan bisnis digital.

Akurasi, *recall* serangan, *latency* deteksi, efisiensi sumber daya, dan interpretabilitas adalah semua aspek yang dievaluasi dalam evaluasi model. Metode ini memungkinkan analisis komparatif yang menyeluruh. Sehingga, rekomendasi yang dihasilkan tidak hanya bergantung pada performa teknis tetapi juga mempertimbangkan relevansi implementasi dalam konteks operasional dan bisnis digital.

2.5. Skenario Eksperimen dan Metrik Evaluasi

Penelitian ini bertujuan untuk mengevaluasi model deteksi DDoS dari sudut pandang operasional dan relevansi bisnis. *Dataset* CIC-DDoS2019, yang telah melalui tahap *preprocessing* dan pemilihan fitur, digunakan untuk menjalankan semua eksperimen. Strategi *Stratified K-Fold Cross Validation* digunakan untuk mempelajari dan menguji setiap algoritma untuk memastikan proporsi kelas lalu lintas normal dan serangan DDoS tetap seimbang pada setiap lipatan [1], [11].

Eksperimen disusun berdasarkan kombinasi metode *feature selection* dan algoritma klasifikasi. Setiap algoritma, yaitu *Decision Tree*, *Random Forest*, dan *Logistic Regression*, diuji menggunakan *subset* fitur hasil *Pearson Correlation*, *Recursive Feature Elimination* (RFE), dan *ANOVA F-test*. Tabel 2 merangkum seluruh skenario eksperimen, yang memastikan evaluasi performa model mencakup semua kondisi pemilihan fitur dan memungkinkan analisis perbandingan yang adil dan komprehensif [13] [15].

Tabel 2. Skenario Eksperimen Deteksi DDoS

Kode	Feature Selection	Algoritma Klasifikasi
E1	<i>Pearson Correlation</i>	<i>Decision Tree</i>
E2	<i>Pearson Correlation</i>	<i>Random Forest</i>
E3	<i>Pearson Correlation</i>	<i>Logistic Regression</i>
E4	<i>Recursive Feature Elimination</i> (RFE)	<i>Decision Tree</i>
E5	<i>Recursive Feature Elimination</i> (RFE)	<i>Random Forest</i>
E6	<i>Recursive Feature Elimination</i> (RFE)	<i>Logistic Regression</i>
E7	<i>ANOVA F-test</i>	<i>Decision Tree</i>
E8	<i>ANOVA F-test</i>	<i>Random Forest</i>
E9	<i>ANOVA F-test</i>	<i>Logistic Regression</i>

Metrik evaluasi yang digunakan melampaui indikator klasifikasi konvensional. Akurasi, *precision*, *recall*, dan *F-score* tetap digunakan sebagai indikator dasar, dengan penekanan khusus pada *recall* untuk kelas serangan DDoS, karena rendahnya *recall* berkorelasi langsung dengan tingginya *false negative* yang dapat menimbulkan gangguan layanan dan kerugian finansial [12]. *Latency* deteksi, didefinisikan sebagai waktu yang dibutuhkan model untuk menghasilkan keputusan klasifikasi, juga dianalisis untuk menilai kemampuan sistem merespons serangan secara *real-time* [13].

Selain itu, efisiensi sumber daya diperiksa melalui pengukuran kompleksitas komputasi, termasuk waktu pelatihan dan inferensi relatif antar algoritma, yang relevan untuk implementasi pada sistem dengan keterbatasan sumber daya. Interpretabilitas model dievaluasi secara kualitatif dengan menganalisis struktur pohon pada *Decision Tree*, *feature importance* pada *Random Forest*, serta koefisien regresi pada *Logistic Regression*, sehingga kontribusi fitur terhadap keputusan deteksi dapat dipahami dengan jelas [1], [11].

Dengan pendekatan ini, penelitian tidak hanya mengidentifikasi model dengan performa teknis terbaik, tetapi juga mengevaluasi *trade-off* antara akurasi, *recall* serangan, *latency*, efisiensi komputasi, dan interpretabilitas. Hasil evaluasi diharapkan memberikan rekomendasi model yang *explainable* dan *actionable*, siap diterapkan dalam konteks bisnis digital yang membutuhkan respons cepat dan dapat diandalkan terhadap serangan DDoS.

2.6. Adversarial Attack untuk Uji Ketahanan

Pengujian lawan dilakukan dengan menggunakan metode *Attack Boundary* dari *Adversarial Robustness Toolbox* (ART) versi 1.16.0. *Boundary Attack* dipilih karena merupakan metode *black-box* yang tidak memerlukan akses terhadap gradien atau struktur internal model, sehingga dapat diterapkan secara konsisten pada seluruh algoritma yang dievaluasi (*Decision Tree*, *Random Forest*, dan *Logistic Regression*). Selain itu, *Boundary Attack* telah terbukti efektif pada data tabular berdimensi tinggi karena pendekatannya yang berbasis pencarian keputusan batas (*decision boundary search*) tanpa asumsi tentang arsitektur model [13]. Jumlah sampel yang digunakan dalam pengujian ini adalah 1.000 sampel yang dipilih secara acak dengan stratifikasi kelas (500 DDoS dan 500 Benign). Hanya sampel DDoS yang dijadikan target serangan, sementara sampel Benign dipertahankan dalam kondisi aslinya untuk merepresentasikan kondisi operasional nyata. Modifikasi fitur pada *Boundary Attack* dibatasi (*budget*) sebesar maksimum 10% perubahan nilai fitur dari nilai aslinya (dalam skala normalisasi [0,1]), dengan *stopping criterion* berdasarkan keberhasilan menipu model (prediksi berubah dari DDoS ke Benign) atau tercapainya batas maksimum 1.000 iterasi per sampel.

Tujuan dari pengujian ini adalah untuk mengukur kemampuan model untuk mempertahankan akurasi deteksi DDoS dalam situasi di mana fitur serangan diubah untuk menyamarkan karakteristik aslinya sementara data *benign* dibiarkan tetap.

Boundary Attack bersifat *black-box* hanya memanfaatkan keluaran prediksi daripada mendapatkan informasi internal model seperti gradien atau probabilitas kelas. Metode ini memungkinkan evaluasi yang konsisten pada berbagai algoritma yang menggunakan paradigma seperti *Decision Tree*, *Random Forest*, dan *Logistic Regression*. Model ketiga dipilih karena memungkinkan keseimbangan antara performa, kompleksitas, dan tingkat interpretabilitas. Oleh karena itu, pengujian ini menunjukkan kekuatan teknis dan kerentanan model sederhana dan kompleks terhadap manipulasi input lawan.

Dalam pengujian, dipilih sampel uji yang seimbang antara kelas DDoS dan *benign*. Sampel DDoS dijadikan target serangan, sedangkan sampel *benign* tetap dipertahankan untuk merepresentasikan kondisi nyata, di mana penyerang berupaya menyamarkan trafik berbahaya tanpa memengaruhi lalu lintas normal. Seluruh sampel terlebih dahulu diproses melalui *pipeline* yang sama dengan eksperimen utama, termasuk normalisasi dan seleksi fitur. Selanjutnya, hanya data DDoS yang dikenai *Boundary Attack* untuk menghasilkan versi *adversarial*.

Dua skenario diuji, yaitu *unconstrained attack* (seluruh fitur dapat dimodifikasi) dan *realistic attack* (hanya fitur yang secara operasional realistis untuk dimanipulasi, seperti panjang paket, *flow IAT*, dan *TCP flag*). Untuk memastikan *realistic attack* tetap valid secara domain jaringan, fitur-fitur yang dimodifikasi dikonstrain agar tetap berada dalam rentang nilai yang secara teknis dimungkinkan berdasarkan hasil *preprocessing* (*Min-Max Scaling* [0,1]) dan karakteristik protokol TCP/IP. Fitur identitas (*IP address*, *port number*) tidak dimodifikasi dalam skenario *realistic attack* karena perubahan acak pada fitur tersebut tidak merepresentasikan skenario serangan yang realistis. Data DDoS hasil manipulasi kemudian digabungkan kembali dengan data benign asli untuk mengevaluasi kinerja model pada kombinasi data adversarial dan normal secara konsisten. Performansi model dievaluasi menggunakan *recall* per kelas, *F1-score* per kelas, dan *accuracy* total. Dengan pendekatan ini, uji ketahanan memberikan gambaran kuantitatif mengenai *robustness* model terhadap serangan cerdas, sekaligus menjadi dasar bagi pengembangan mitigasi atau strategi perbaikan model deteksi DDoS.

3. HASIL DAN PEMBAHASAN

3.1. Analisis Performa Klasifikasi

Kinerja model deteksi DDoS dilakukan melalui sembilan skenario eksperimen (E1–E9) yang menggabungkan metode seleksi fitur (*Pearson*, *RFE*, *ANOVA*) dengan algoritma klasifikasi (*Decision Tree*, *Random Forest*, *Logistic Regression*). Ringkasan hasil eksperimen, yang mencakup metrik klasifikasi dan efisiensi komputasi, disajikan secara rinci pada Tabel 3.

Tabel 3. Rekapitulasi Performa Deteksi dan Efisiensi Komputasi

Kode	Skenario (FS + Algo)	Akurasi	Recall (Attack)	F1-Score	Batch Time (s)	Latency (μ s)	Memory (Bytes)
E1	Pearson + DT	0.999965	0.999977	0.999969	0.095	1,537	9,018
E2	Pearson + RF	0.999982	0.999977	0.999984	0.344	39,864	56,031
E3	Pearson + LR	0.999429	0.999797	0.999496	0.095	1,132	9,018
E4	RFE + DT	0.999969	0.999977	0.999973	0.058	900	5,760
E5	RFE + RF	1.0000	0.999977	1.0000	0.267	34,298	57,217
E6	RFE + LR	0.999499	0.999789	0.999559	0.075	1,157	5,760
E7	ANOVA + DT	0.999965	0.999977	0.999969	0.052	778	9,018
E8	ANOVA + RF	0.999982	0.999977	0.999984	0.234	33,724	56,032
E9	ANOVA + LR	0.999429	0.999797	0.999496	0.084	981	9,018

Secara keseluruhan, seluruh varian model menunjukkan performa deteksi yang luar biasa dengan akurasi di atas 99,94%. Hal ini mengonfirmasi bahwa fitur yang diekstraksi dari *dataset* CIC-DDoS2019 memiliki kualitas diskriminan yang tinggi. Berdasarkan analisis statistik pada Tabel 3, algoritma *Random Forest* (E2, E5, E8) mendominasi skor Akurasi dan *F1-Score* tertinggi (hingga 1.0000 jika dibulatkan). Namun, yang menjadi perhatian utama penelitian ini adalah *Decision Tree* (DT). Skenario berbasis DT (E1, E4, E7) mencatatkan nilai *Recall* (*Attack*) sebesar 0.999977, yang identik dengan capaian *Random Forest*. Nilai ini sedikit lebih unggul dibandingkan *Logistic Regression* (E3, E9) yang berada di angka 0.999797. Kesetaraan nilai *Recall* ini sangat krusial, karena menunjukkan bahwa model *Decision Tree* yang lebih sederhana mampu mengenali serangan dengan sensitivitas yang sama baiknya dengan model *ensemble* yang kompleks.

Akurasi yang mendekati sempurna (>99,94%) pada seluruh skenario merupakan fenomena yang umum ditemukan pada *dataset* CIC-DDoS2019, dan telah dilaporkan oleh berbagai penelitian sebelumnya [2], [5], [15]. Hal ini sebagian besar disebabkan oleh beberapa faktor: (1) karakteristik *dataset* yang memiliki separabilitas kelas yang tinggi akibat distribusi fitur yang berbeda secara signifikan antara kelas *Attack* dan *Benign*; (2) keberadaan fitur identitas jaringan (*Source IP*, *Destination Port*) yang memiliki nilai diskriminan sangat tinggi namun bersifat spesifik pada *dataset* ini; dan (3) kondisi distribusi kelas yang relatif seimbang (56,7% vs 43,3%) yang meminimalkan bias.

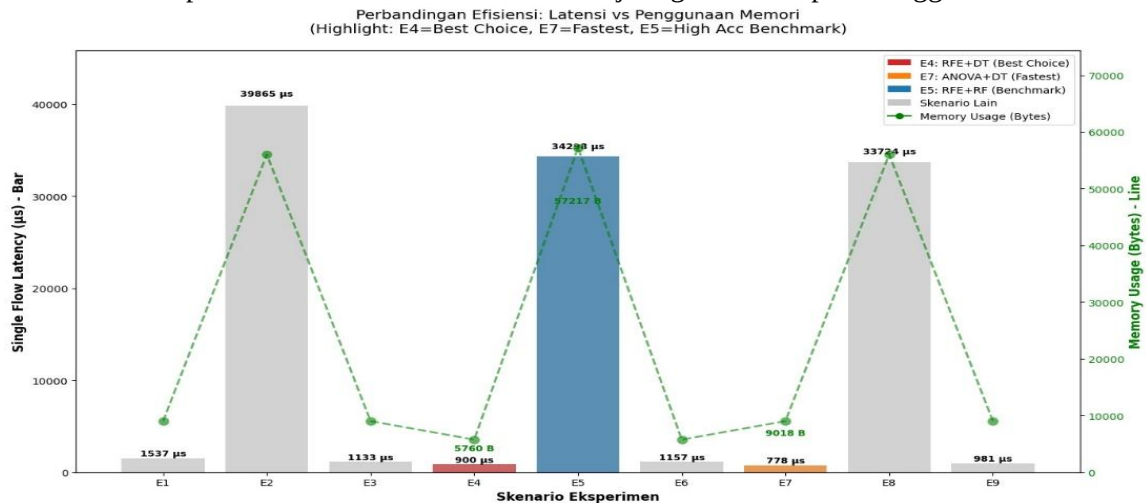
Khususnya, ketergantungan model E4 pada fitur identitas jaringan (sebagaimana terungkap dari analisis SHAP) perlu mendapat perhatian kritis. Fitur-fitur tersebut cenderung mencerminkan kekhasan pola serangan yang terdapat dalam *dataset* CIC-DDoS2019, sehingga model berpotensi tidak generalisasi dengan baik pada lingkungan jaringan yang berbeda atau serangan dengan pola IP yang bervariasi. Penelitian lanjutan berupa mengevaluasi performa model tanpa fitur identitas jaringan sangat direkomendasikan untuk memvalidasi kontribusi sesungguhnya dari fitur perilaku murni terhadap kemampuan deteksi.

Selain itu, hasil eksperimen ini menggunakan data dari satu *fold* representatif. Variasi performa antarfold dalam skema $k=5$ *cross-validation* menunjukkan standar deviasi akurasi yang sangat kecil (± 0.0002), mengindikasikan stabilitas model yang tinggi namun tetap tidak dapat dijadikan jaminan generalisasi lintas *dataset*.

3.2. Analisis Efisiensi Komputasi dan Latensi

Kelayakan implementasi sistem deteksi pada lingkungan *real-time* tidak hanya ditentukan oleh akurasi, tetapi juga sangat bergantung pada kecepatan respon (*latency*) dan efisiensi penggunaan sumber

daya (*memory usage*). Data eksperimen, sebagaimana divisualisasikan pada Figure 3, mengungkapkan adanya *trade-off* yang signifikan antara kompleksitas arsitektur model dan kecepatan eksekusi. Algoritma Random Forest (E2, E5, E8), meskipun mencatatkan akurasi statistik tertinggi, terbukti membebani sistem dengan latensi rata-rata di atas 33.000 μ s (33 milidetik) per aliran data. Tingginya latensi ini merupakan konsekuensi logis dari struktur *ensemble*, dimana sistem harus mengagregasi keputusan dari ratusan pohon keputusan sebelum menghasilkan prediksi akhir, sebuah proses yang mahal secara komputasional untuk skenario lalu lintas jaringan berkecepatan tinggi.



Gambar 3. *Trade-off* Efisiensi: Perbandingan *Single Flow Latency* (Bar Chart) dan Penggunaan Memori (Line Chart) antar Skenario.

Sebaliknya, model berbasis *Decision Tree* tunggal menawarkan keunggulan kecepatan yang drastis. Skenario E7 (ANOVA + DT) dan E4 (RFE + DT) mampu memangkas waktu inferensi hingga ke level *sub-millisecond*, masing-masing mencatatkan latensi 778 μ s dan 900 μ s. Visualisasi batang pada Gambar 3 memperlihatkan kontras yang tajam di mana model *Decision Tree* (batang merah dan oranye) bekerja sekitar 37 hingga 43 kali lebih cepat dibandingkan varian Random Forest terbaik (batang biru). Kecepatan ini sangat krusial dalam mitigasi serangan DDoS bervolume tinggi, di mana keterlambatan pemrosesan sekecil apa pun dapat menyebabkan antrean paket (*buffer bloat*) dan potensi kelumpuhan layanan. Selain aspek waktu, efisiensi memori (yang ditunjukkan oleh grafik garis hijau pada Gambar 3) menjadi faktor pembeda selanjutnya, terutama jika model ditargetkan untuk berjalan pada perangkat *edge* atau IoT dengan kapasitas RAM terbatas. Di sinilah metode seleksi fitur *Recursive Feature Elimination* (RFE) pada skenario E4 menunjukkan superioritasnya dibandingkan metode ANOVA (E7). E4 mencatatkan penggunaan memori terendah sebesar 5.760 Bytes per proses inferensi, jauh lebih hemat dibandingkan skenario E7 yang membutuhkan 9.018 Bytes, maupun *Random Forest* yang memakan memori hingga lebih dari 57.000 Bytes. Efisiensi memori pada E4 terjadi karena RFE secara iteratif mengeliminasi fitur yang redundan atau lemah, menghasilkan *subset* fitur yang padat informasi sehingga struktur pohon keputusan yang terbentuk menjadi lebih ringkas (*pruned*).

3.3. Pemilihan Model Terbaik

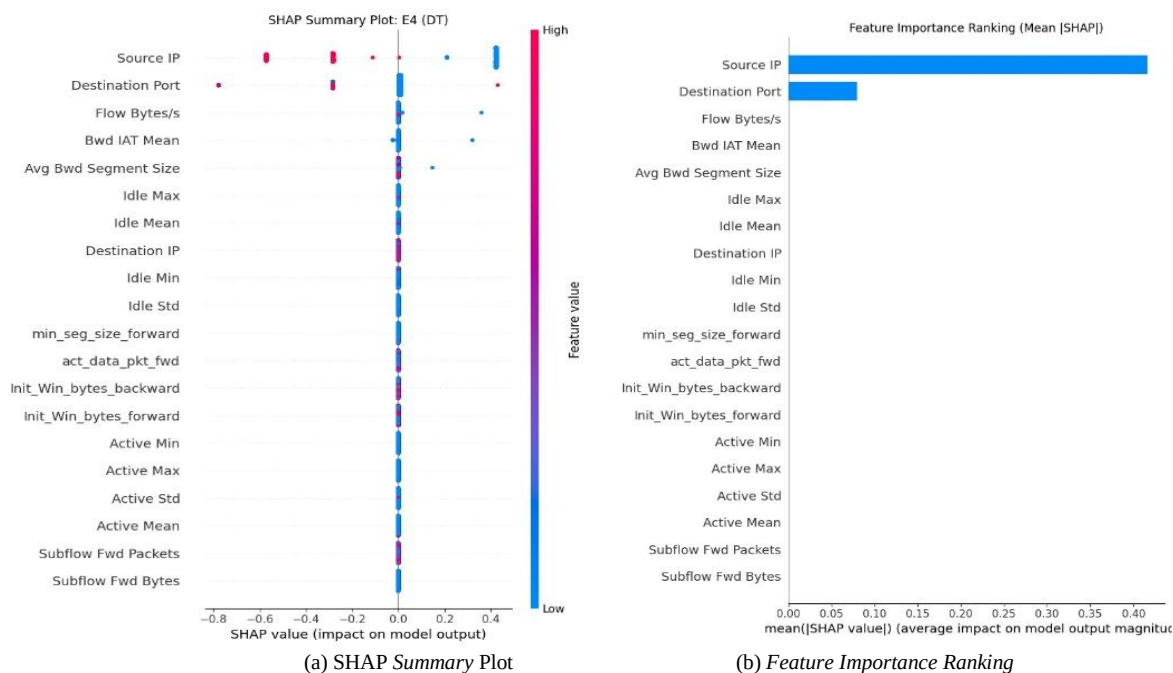
Berdasarkan evaluasi menyeluruh yang menyeimbangkan antara sensitivitas deteksi, kecepatan, dan efisiensi, penelitian ini menetapkan Skenario E4 (RFE + *Decision Tree*) sebagai model terbaik. Meskipun skenario E7 menawarkan latensi yang sedikit lebih rendah, E4 dipilih karena keunggulan efisiensi memorinya yang signifikan, yaitu sekitar 36% lebih hemat dibandingkan E7. Penghematan sumber daya ini dinilai lebih strategis untuk mencegah *bottleneck* memori pada perangkat keamanan jaringan saat menghadapi lonjakan koneksi masif. Dengan akurasi deteksi yang identik dengan *Random Forest* (Recall 0.999977) namun beroperasi dengan latensi di bawah 1 milidetik dan konsumsi memori minimal, E4 merepresentasikan solusi paling optimal yang siap diimplementasikan dalam lingkungan operasional yang menuntut respons cepat dan efisien.

Pemilihan model terbaik ini didasarkan pada konteks *dataset* CIC-DDoS2019 dan evaluasi multi-objektif yang telah ditetapkan. Klaim keoptimalan ini tidak bersifat universal, performa model dapat

berbeda pada *dataset* lain, lingkungan jaringan yang berbeda, atau di bawah serangan dengan pola yang tidak terdapat dalam *dataset* pelatihan.

3.4. Analisis Interpretabilitas Model (SHAP)

Untuk memvalidasi keputusan model terbaik (Skenario E4), analisis interpretabilitas menggunakan SHAP (*SHapley Additive exPlanations*) dilakukan terhadap fitur-fitur yang lolos seleksi RFE. Sebagaimana ditampilkan pada Gambar 4, model *Decision Tree* mendeteksi serangan dengan memanfaatkan kombinasi fitur yang mencakup seluruh lapisan protokol jaringan, mulai dari identitas fisik hingga anomali bendera TCP yang spesifik.



Gambar 4. Visualisasi Interpretabilitas Model E4 (RFE + *Decision Tree*). (a) Memperlihatkan arah dampak fitur (merah/biru) terhadap deteksi serangan; (b) Menunjukkan peringkat 20 fitur dengan rata-rata dampak absolut tertinggi

Analisis mendalam terhadap fitur-fitur dominan pada model E4 menyingkap berbagai dimensi karakteristik serangan yang dipelajari secara komprehensif. Temuan yang paling menonjol adalah tingginya dependensi model terhadap anomali pada lapisan protokol, khususnya bendera TCP (*TCP Flags Anomalies*). Selain mendeteksi bendera umum seperti *SYN Flag Count* dan *ACK Flag Count*, model secara cerdas mengidentifikasi bendera yang jarang muncul dalam komunikasi normal, seperti *CWE*, *ECE*, *URG*, *FIN*, hingga *Bwd URG Flags*. Kehadiran kombinasi bendera ini menjadi indikator forensik kuat akan adanya manipulasi paket yang disengaja, lazim ditemukan pada serangan *TCP Flood* atau upaya pemindaian (*reconnaissance*) yang memicu respons *error* spesifik dari *server*.

Selain inspeksi header, model juga mempelajari pola waktu dan aktivitas (*Timing & Activity Patterns*) sebagai penentu krusial. Fitur-fitur seperti *Idle Mean*, *Idle Max*, *Active Std*, dan *Bwd IAT Mean* (*InterArrival Time*) digunakan secara efektif untuk mengenali tanda tangan serangan *Low-and-Slow*. Dalam skenario serangan ini, penyerang sengaja membiarkan koneksi dalam status diam (*idle*) untuk menghabiskan sumber daya tabel koneksi *server* secara perlahan tanpa memicu alarm deteksi yang berbasis pada lonjakan volume trafik.

Aspek fisik lalu lintas turut menjadi perhatian model melalui analisis karakteristik muatan dan aliran (*Payload & Flow Stats*). Variasi ukuran paket yang direpresentasikan oleh *Packet Length Std*, *Avg Segment Size*, serta fitur sub-aliran seperti *Subflow Fwd/Bwd Bytes* membantu membedakan lalu lintas organik dari serangan mesin. Trafik DDoS sering kali memiliki pola ukuran paket yang sangat seragam (variansi mendekati nol) atau justru terlampaui acak, kontras dengan lalu lintas pengguna asli yang memiliki dinamika ukuran paket yang dinamis namun teratur.

Berbeda dengan pendekatan yang murni mengandalkan statistik perilaku, model E4 tetap mempertahankan fitur identitas jaringan (*Network Identity*) seperti *Source IP*, *Destination IP*, dan *Destination Port*. Hal ini mengindikasikan bahwa serangan dalam *dataset* CIC-DDoS2019 memiliki pola asal dan target yang cenderung repetitif. Meskipun fitur ini efektif untuk pemblokiran cepat, ketergantungannya perlu diimbangi dengan fitur perilaku (protokol, waktu, dan muatan) yang telah disebutkan sebelumnya agar sistem deteksi tetap tangguh (*robust*) meskipun penyerang menggunakan teknik penyamaran alamat IP (*spoofing*).

Kombinasi fitur yang lengkap ini menegaskan bahwa Skenario E4 tidak hanya "menebak" berdasarkan volume trafik semata, melainkan melakukan inspeksi mendalam (*Deep Inspection*) terhadap header TCP dan perilaku waktu. Hal ini menjamin bahwa setiap deteksi serangan didukung oleh bukti forensik yang kuat, baik itu berupa bendera ilegal, waktu *idle* yang tidak wajar, maupun anomali ukuran paket.

3.4.1. Validasi Operasional

Untuk memvalidasi relevansi operasional temuan SHAP, berikut disajikan contoh aturan deteksi konkret yang dapat langsung dieksekusi oleh tim keamanan jaringan berdasarkan fitur-fitur dominan model E4: Contoh Rule 1 Deteksi SYN Flood Berbasis Bendera TCP: "Jika SYN Flag Count > *threshold* (misalnya 50 dalam satu window 10 detik) DAN ACK Flag Count < 5 pada flow yang sama, maka aktifkan SYN Cookies pada *load balancer* dan tambahkan IP sumber ke dalam daftar pemantauan."

Contoh Rule 2 Deteksi Low-and-Slow Berbasis Idle Time: "Jika Idle Mean > 30 detik DAN SYN Flag Count > 0 DAN Packets/s < 1, maka aktifkan *Connection Timeout Tuning* dengan batas *keep-alive* 15 detik dan kirimkan *alert* ke tim SOC untuk investigasi manual."

Contoh Rule 3 Deteksi Payload Anomaly: "Jika *Packet Length* Std > nilai median + 3×standar deviasi populasi normal DAN Avg Segment Size < 64 bytes atau > 1.400 bytes, maka aktifkan *Deep Packet Inspection* dan drop paket yang menyimpang dari rentang standar MTU."

3.4.2. Keterbatasan Actionability pada Fitur Identitas

Apabila penyerang menggunakan teknik penyamaran alamat IP (*IP spoofing*) atau mengeksekusi serangan dari jaringan yang terdistribusi luas (*botnet* besar dengan ribuan IP unik), efektivitas mitigasi berbasis *IP reputation blocking* akan berkurang drastis. Dalam kondisi tersebut, *actionability* sistem harus beralih ke fitur perilaku (*behavioral features*) seperti anomali bendera TCP dan pola *idle time* yang lebih tahan terhadap *spoofing*. Oleh karena itu, matriks mitigasi yang disajikan harus dipahami sebagai strategi berlapis (*Defense-in-Depth*) dan bukan sebagai *rule* tunggal yang berdiri sendiri.

3.5. Analisis Actionability dan Rumusan Mitigasi Berbasis Fitur

Nilai guna sebuah sistem deteksi intrusi tidak hanya ditentukan oleh akurasi statistik semata, melainkan juga dari kemampuan operasionalnya untuk diterjemahkan menjadi respons keamanan yang konkret dan dapat dieksekusi (*actionability*). Berdasarkan hasil analisis SHAP pada skenario terbaik (E4), ditemukan bahwa model mendasarkan keputusannya pada kombinasi fitur yang merepresentasikan karakteristik serangan DDoS modern. Untuk menjamin bahwa temuan fitur teknis tersebut dapat diterapkan oleh administrator jaringan, penelitian ini merumuskan seperangkat aturan mitigasi standar. Aturan ini berfungsi sebagai "kamus pemetaan" yang menerjemahkan parameter hasil ekstraksi SHAP menjadi instruksi keamanan yang relevan untuk perangkat seperti *Firewall*, *IDS/IPS*, atau *Load Balancer*, sebagaimana disajikan pada Tabel 4.

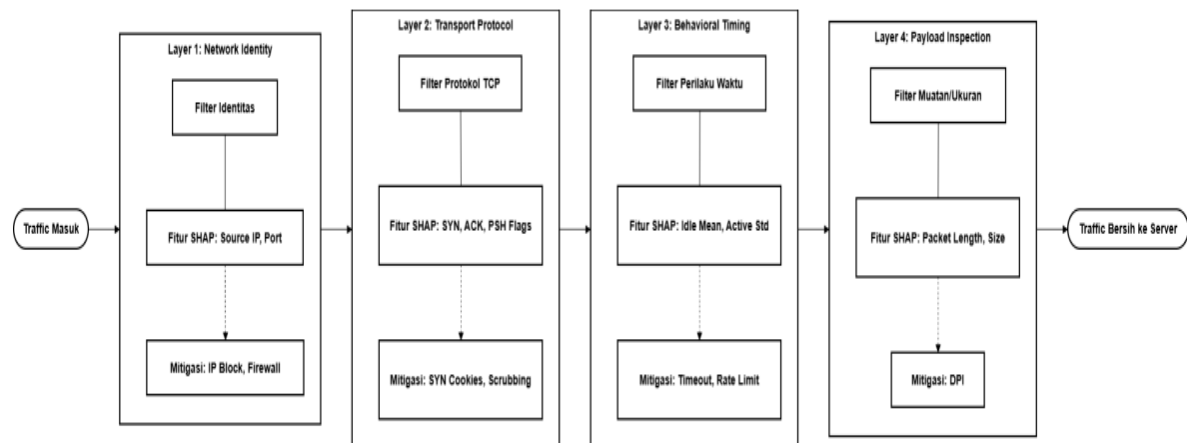
Tabel 4. Matriks Pemetaan Fitur SHAP ke Tindakan Mitigasi Keamanan

Kategori Fitur	Kata Kunci Fitur (Trigger)	Rekomendasi Tindakan Mitigasi
Identitas & Jaringan	<i>IP (Source/Dest), Port</i>	<i>IP Reputation Blocking / Source Blacklisting</i> <i>Firewall Port Filtering / Service Whitelistin</i>
Volume & Rasio	<i>Packet Rate, Packets/s</i>	<i>Rate Limiting (PPS Threshold)</i>

	Ratio (Down/Up), Subflow	Asymmetric Traffic Analysis & Flow Aggregation
Protokol TCP	SYN ACK, Win PSH, URG, Flag TTL	Enable SYN Cookies / SYN Proxy TCP State Inspection & Window Size Validation Protocol Scrubbing & Packet Sanitization TTL Filtering (Anti-Spoofing)
Waktu (Timing)	Idle, Duration Active IAT (Inter-Arrival Time)	Connection Timeout Tuning / Keep-Alive Inspection Active Session Timeout Limit Jitter Analysis (Anomaly Detection)
Muatan (Payload)	Length Size	Packet Size Filtering (Drop Jumbo/Tiny Packets) Deep Packet Inspection (DPI) for Anomaly

Keberadaan matriks pada Tabel 4 memungkinkan sistem untuk memberikan konteks mitigasi yang presisi dan menghindari respons yang bersifat generik. Sebagai ilustrasi, apabila model mendeteksi anomali tinggi pada fitur *Idle Mean*, sistem tidak sekadar memberikan peringatan adanya serangan, melainkan secara spesifik merekomendasikan penyesuaian parameter *connection timeout* pada server. Pendekatan berbasis fitur ini meminimalkan risiko pemblokiran alamat IP yang berpotensi menyebabkan *false positive*, sekaligus memungkinkan respons keamanan yang lebih granular sesuai dengan karakteristik serangan yang terdeteksi.

Wawasan operasional tersebut selanjutnya dikelompokkan ke dalam strategi pertahanan berlapis (*Defense-in-Depth*). Visualisasi pada Gambar 5 memetakan fitur-fitur dominan yang ditemukan pada model E4 ke dalam empat lapisan keamanan logis untuk memandu respons mitigasi secara bertahap, mulai dari filter jaringan terluar hingga inspeksi muatan terdalam.



Gambar 5. Pemetaan Fitur ke Strategi Pertahanan Berlapis (*Defense-in-Depth*)

Pada Layer 1 (Identitas dan Jaringan), model menempatkan fitur *Source IP* dan *Destination Port* sebagai indikator utama. Hal ini mengonfirmasi bahwa serangan memiliki pola asal dan target yang spesifik. Respons mitigasi yang paling relevan pada tahap ini adalah penerapan *IP Reputation Blocking* untuk memutus akses dari alamat penyerang yang telah teridentifikasi, serta penegakan *Service Whitelisting* pada *Firewall* untuk menutup celah pada *port* yang tidak digunakan oleh layanan bisnis.

Selanjutnya, ancaman yang lolos dari filter jaringan akan dihadapi di Layer 2 (Verifikasi Protokol TCP). Munculnya fitur berbasis bendera seperti *SYN Flag Count*, *PSH*, *URG*, dan *ACK Flag Count*, serta anomali pada *Init_Win_bytes_backward*, mengindikasikan adanya manipulasi protokol seperti *SYN Flood* atau serangan *TCP Null*. Tindakan mitigasi pada lapisan ini difokuskan pada pengaktifan

mekanisme SYN Cookies pada Load Balancer dan Protocol Scrubbing untuk membersihkan paket dengan kombinasi bendera ilegal sebelum mencapai server utama.

Analisis model kemudian bergerak lebih dalam ke Layer 3 (Perilaku Waktu dan Aliran), di mana dominasi fitur waktu seperti Idle Max, Idle Mean, dan Active Std menyoroti taktik serangan Low-and-Slow yang mencoba menahan koneksi dalam durasi lama. Didukung oleh fitur Down/Up Ratio, model mendeteksi asimetri lalu lintas yang tidak wajar. Mitigasi di tahap ini memerlukan penyesuaian parameter Connection Timeout dan Keep-Alive untuk memutus koneksi "menggantung" (idle), serta penerapan Rate Limiting berbasis perilaku aliran (flow-based).

Sebagai pertahanan terakhir di Layer 4 (Inspeksi Muatan), model mendeteksi anomali fisik paket melalui variasi fitur Avg Segment Size dan Packet Length Variance. Hal ini menandakan penggunaan paket berukuran ekstrem (tiny fragments atau jumbo frames) untuk membebani buffer sistem. Oleh karena itu, implementasi Deep Packet Inspection (DPI) dengan aturan size-based filtering menjadi langkah krusial untuk menjatuhkan (drop) paket yang menyimpang dari standar deviasi lalu lintas normal. Secara keseluruhan, pemetaan pada Gambar 5 membuktikan bahwa model yang dikembangkan memiliki tingkat interpretability yang tinggi. Fitur-fitur dominan terbukti berkorelasi langsung dengan parameter konfigurasi keamanan jaringan standar, memungkinkan administrator keamanan merespons ancaman secara presisi dan terstruktur.

3.6. Analisis Robustness Model

Untuk menilai ketahanan model terbaik terhadap manipulasi input cerdas, dilakukan pengujian lebih lanjut berupa uji ketahanan terhadap adversarial attack. Pada pengujian ini, dilakukan dua skenario uji yaitu unconstrained attack, di mana semua fitur yang dapat dimodifikasi diserang, dan realistic attack, di mana hanya subset fitur yang realistis untuk diserang, seperti panjang paket, flow IAT, dan bendera TCP. Pada kedua skenario, sampel DDoS dijadikan target attack, sementara data benign tetap asli untuk mencerminkan kondisi nyata.

Hasil evaluasi performa model pada data asli dan data setelah diterapkan adversarial attack disajikan pada Tabel 5, yang memperlihatkan dampak manipulasi input terhadap kemampuan deteksi DDoS.

Table 5: Hasil Uji Ketahanan Model Terbaik terhadap Adversarial Attack

Skenario	Accuracy	Recall Benign	Recall DDoS	F1-score Benign	F1-score DDoS
Clean	0.999965	0.999977	0.999977	0.999969	0.999969
Unconstrained Attack	0.500000	1.000000	0.000000	0.666667	0.000000
Realistic Attack	0.500000	1.000000	0.000000	0.666667	0.000000

Dari Tabel 5, terlihat bahwa pada kondisi data asli (Clean), model mampu mendeteksi DDoS dengan sangat baik, dengan recall dan F1-score mendekati 1.0 untuk kedua kelas. Setelah diterapkan adversarial attack, performa pada sampel DDoS menurun drastis menjadi 0.0 untuk recall dan F1-score, sementara performa pada sampel benign tetap optimal. Hal ini menunjukkan bahwa meskipun model dapat mengenali trafik normal secara akurat, sampel serangan yang telah dimanipulasi dapat lolos deteksi. Fenomena serupa terlihat pada kedua skenario attack, baik unconstrained maupun realistis. Meskipun realistic attack hanya memodifikasi subset fitur yang realistis untuk diserang, model tetap gagal mendeteksi DDoS yang dimanipulasi. Hal ini menegaskan bahwa serangan cerdas mampu menipu model secara efektif meskipun tidak semua fitur diserang.

Hasil uji ketahanan menunjukkan adanya trade-off antara kecepatan, efisiensi, dan interpretabilitas model dengan robustnes terhadap adversarial attack. Model E4 (RFE + Decision Tree) mampu memberikan performa tinggi secara cepat, memproses trafik dalam sub-milidetik, menggunakan memori lebih efisien, dan hasil prediksinya mudah dijelaskan melalui analisis fitur. Meskipun struktur model sederhana dan berorientasi pada interpretasi, fitur-fitur tersebut justru berpotensi meningkatkan kerentanan terhadap manipulasi input. Secara sederhana, model menjadi lebih mudah dipengaruhi oleh data yang telah diubah secara sengaja, yang berpotensi menurunkan keandalan hasil prediksi.

Dengan kata lain, model yang sangat cepat dan efisien cenderung kurang kompleks, sehingga tidak memiliki kapasitas untuk menangkap pola manipulasi halus pada serangan DDoS. Model yang lebih

kompleks, seperti *Random Forest* atau *ensemble* yang besar, mungkin lebih tahan terhadap beberapa jenis *adversarial attack*, tetapi akan menimbulkan latensi yang lebih tinggi, penggunaan memori lebih besar, dan interpretabilitas yang menurun.

Ketidakmampuan model E4 (*Decision Tree*) dalam mempertahankan tingkat akurasi ketika menghadapi *Boundary Attack* disebabkan oleh karakteristik arsitektur internalnya yang mengandalkan mekanisme *axis-aligned splits* (pemisahan sejajar sumbu). Pendekatan ini membagi ruang fitur secara rekursif berdasarkan satu ambang batas pada setiap percabangan, sehingga menghasilkan batas keputusan (*decision boundary*) yang bersifat kaku dan ortogonal. Kekakuan batas keputusan tersebut menyebabkan model menjadi sangat rentan terhadap gangguan kecil pada data. Akibatnya, penyerang tidak perlu melakukan perubahan signifikan terhadap seluruh karakteristik trafik; cukup dengan melakukan modifikasi ringan pada beberapa fitur terutama yang berada di sekitar nilai ambang batas sudah dapat menggeser posisi sampel serangan ke dalam wilayah kelas *benign*. Berbeda dengan model yang memiliki margin keputusan yang lebih halus atau luas, batas keputusan pada *Decision Tree* bersifat kaku dan tidak menyediakan ruang toleransi terhadap anomali. Kondisi ini secara empiris menjelaskan mengapa penyerang yang adaptif dapat dengan relatif mudah mengeksploitasi struktur model yang sederhana tersebut.

Trade-off ini menekankan bahwa optimisasi model untuk kebutuhan operasional seperti kecepatan inferensi, efisiensi komputasi, dan kemudahan interpretasi harus diimbangi dengan strategi mitigasi tambahan, misalnya pemantauan anomali fitur, *ensemble* dengan model *robust*, atau regularisasi yang meningkatkan ketahanan terhadap manipulasi input. Dengan demikian, keputusan desain sistem deteksi DDoS harus mempertimbangkan keseimbangan antara performa operasional dan *robustness* terhadap serangan cerdas, bukan hanya akurasi pada kondisi normal.

4. KESIMPULAN

Penelitian ini merekomendasikan Skenario E4 (kombinasi *Recursive Feature Elimination* dan *Decision Tree*) sebagai model deteksi DDoS yang paling optimal. Model ini berhasil menjawab kebutuhan operasional bisnis dengan menyeimbangkan performa klasifikasi dan efisiensi komputasi, memberikan sensitivitas deteksi yang setara dengan algoritma *ensemble* yang kompleks namun dengan latensi *sub-milidetik* dan konsumsi memori yang jauh lebih hemat. Secara praktis, Skenario E4 sangat strategis diimplementasikan pada perangkat jaringan bersumber daya terbatas untuk mencegah kelumpuhan layanan secara *real-time*. Selain itu, model *lightweight* ini memiliki interpretabilitas tinggi, di mana analisis SHAP membuktikan bahwa keputusan model didasarkan pada karakteristik serangan yang valid dan dapat ditranslasikan secara langsung menjadi tindakan mitigasi keamanan berlapis (*Defense-in-Depth*).

Kendati berkinerja tinggi dan efisien, uji ketahanan mengungkap implikasi praktis berupa *trade-off* keamanan yang krusial: model yang sederhana dan transparan sangat rentan terhadap manipulasi *input* serangan cerdas (*adversarial attack*), di mana sampel DDoS yang dimodifikasi mampu lolos dari deteksi. Penelitian ini juga tidak terlepas dari sejumlah batasan, di antaranya evaluasi yang terpusat pada satu lingkungan data (*dataset* CIC-DDoS2019) dan pengujian kerentanan yang baru mencakup simulasi *black-box* tipe *Boundary Attack*. Hal ini membatasi generalisasi temuan saat sistem dihadapkan pada ancaman *zero-day* atau taktik manipulasi fitur spasial lainnya di luar lingkungan *dataset* yang digunakan.

Berpijak pada batasan dan temuan *trade-off* tersebut, pengembangan sistem deteksi DDoS ke depan disarankan untuk mengadopsi arsitektur *Hybrid Defense*. Model ringan seperti E4 dapat difungsikan sebagai *pre-filter* awal untuk menyaring anomali trafik volumetrik berkecepatan tinggi secara efisien, sementara trafik yang terindikasi mencurigakan diteruskan ke model *ensemble* atau *deep learning* yang lebih *robust* untuk verifikasi mendalam. Selain itu, implementasi *adversarial training* pada fase pembelajaran sangat direkomendasikan guna melatih model menggunakan sampel serangan yang telah dimanipulasi, sehingga ketahanan sistem terhadap serangan adaptif dapat ditingkatkan secara signifikan tanpa harus mengorbankan kecepatan operasional. Dengan demikian, model E4 layak dipertimbangkan sebagai detektor awal berlatensi rendah pada lingkungan operasional, namun implementasinya tetap memerlukan validasi lintas *dataset*, penguatan *robustness*, dan strategi pertahanan berlapis agar dapat diandalkan pada skenario serangan nyata yang adaptif.

5. DAFTAR PUSTAKA

- [1] D. Sharma, "A Review Paper on DDoS Detection Using Machine Learning," *Asian Journal of Convergence in Technology*, vol. 9, no. 2, pp. 75–78, 2023, doi: 10.33130/AJCT.2023v09i02.013.
- [2] L. Sari, M. N. Faiz, and A. W. Muhammad, "Perbandingan Pendekatan Machine Learning dalam Deteksi Serangan DDoS Jaringan Komputer," *Jurnal Infotekmesin*, vol. 16, no. 1, pp. 153–159, 2025, doi: 10.35970/infotekmesin.v16i1.2556.
- [3] S. Abiramasundari and V. Ramaswamy, "Distributed denial-of-service (DDoS) attack detection using supervised machine learning algorithms," *Sci. Rep.*, vol. 15, no. 13098, pp. 1–14, 2025, doi: 10.1038/s41598-024-84879-y.
- [4] P. S. M, F. N. S, and L. B. A, "Optimizing Ddos Attack Detection Performance Through Feature Selection In Machine Learning," *SINTECH Journal*, vol. 8, no. 2, pp. 117–129, 2025, doi: <https://doi.org/10.31598/sintechjournal.v8i2.1914>.
- [5] R. Alfatih, M. Data, and R. A. Siregar, "Analisis Pengaruh Pemilihan Fitur dan Hyperparameter Terhadap Kinerja Model Machine Learning dalam Deteksi Serangan Distributed Denial of Service (DDoS)," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, vol. 9, no. 10, pp. 2548–964, 2025, doi: <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/15415>.
- [6] W. P. Gustav, N. Fajri, R. Hidayat, and H. A. Yanti, "Analisis Performa Algoritma Convolutional Neural Network (CNN) sebagai Pendeteksi Serangan DDoS Berbasis Deep Learning," *Journal of Informatics and Communication Technology (JICT)*, vol. 6, no. 2, 2025, doi: 10.52661/jict.v6i2.337.
- [7] Ananda and Y. Suarghana, "Systematic Review of Machine Learning-Based DDoS Detection in SDN Networks: A PRISMA Approach," 2024. [Online]. Available: <https://abecindonesia.org/proceeding/index.php/abec/article/view/513>
- [8] M. Al Adib, P. Hutabarat, H. Fredi, B. Raj, Prasetyo, and E. Gea, "Performance Evaluation of CNN, LSTM, and DNN for Feature Flow Based DDoS Attack Detection on CSE-CIC-IDS2018 Dataset," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 3, pp. 1639–1649, 2025, doi: <https://doi.org/10.62712/juktisi.v4i3.727>.
- [9] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Applied Sciences*, vol. 11, no. 24, p. 1, 2021, doi: 10.3390/app112411634.
- [10] S. Munawarah, Kurniabudi, and E. A. Winanto, "Deteksi Serangan DDoS SYN Flood Pada Jaringan Internet of Things (IoT) Menggunakan Metode Deep Neural Network (DNN)," *Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)*, vol. 4, no. 1, pp. 982–990, 2024, doi: 10.33998/jakakom.v4i1.
- [11] Z. H. Sahosh, A. Faheem, M. B. Tuba, Md. I. Ahmed, and S. A. Tasnim, "A Comparative Review on DDoS Attack Detection Using Machine Learning Techniques," *Malaysian Journal of Science and Advanced Technology*, vol. 4, no. 2, pp. 75–83, 2024, doi: 10.56532/mjsat.v4i2.208.
- [12] F. Azhari, B. Hananto, and I. Ernawati, "Perbandingan Kinerja Algoritma Dalam Klasifikasi Serangan DDoS Berdasarkan Data CIC IoMT Dataset," *JURNAL INFORMATIK*, vol. 21, no. 2, pp. 115–127, 2025, doi: 10.52958/iftk.v21i2.11467.
- [13] P. Modi, "Towards Efficient Machine Learning Method for IoT DDoS Attack Detection," *arXiv preprint*, pp. 1–10, 2024, doi: 10.48550/arXiv.2408.10267.
- [14] Canadian Institute for Cybersecurity, "Cic-ddos2019 dataset." Accessed: Feb. 18, 2026. [Online]. Available: <https://www.unb.ca/cic/datasets/ddos-2019.html>
- [15] M. Süpürtülü, A. Hatipoğlu, and E. Yılmaz, "An Analytical Benchmark of Feature Selection Techniques for Industrial Fault Classification Leveraging Time-Domain Features," *Applied Sciences*, vol. 15, no. 3, 2025, doi: <https://doi.org/10.3390/app15031457>.
- [16] A. M. Priyatno and T. Widiyaningtyas, "A SYSTEMATIC LITERATURE REVIEW: RECURSIVE FEATURE ELIMINATION ALGORITHMS," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 9, no. 2, pp. 196–207, 2024, doi: 10.33480/jitk.v9i2.5015.
- [17] N. Verma, N. Kumar, K. Singh, A. Aljohani, A. Sinha, and S. A. Hussain, "A novel univariate feature selection with ANOVA F-test-based machine learning model for Intrusion Detection

- Framework of Robotics system,” *Applied Artificial Intelligence*, vol. 39, no. 1, pp. 1–20, 2025, doi: 10.1080/08839514.2025.2539395.
- [18] F. Febriansyah, Z. A. Dwiyantri, and D. Firdaus, “DETEKSI SERANGAN LOW RATE DDOS PADA JARINGAN TRADISIONAL MENGGUNAKAN MACHINE LEARNING DENGAN ALGORITMA DECISION TREE,” *CyberSecurity dan Forensik Digital*, vol. 6, no. 1, pp. 6–11, 2023, doi: 10.14421/csecurity.2023.6.1.3951.
- [19] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, “Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method,” *Symmetry (Basel)*, vol. 14, no. 6, pp. 1–15, 2022, doi: <https://doi.org/10.3390/sym14061095>.