Penerapan Algoritma *Gradient Boosted Decision Tree* (*GBDT*) untuk Klasifikasi Serangan *DDoS*

Yusril Isra Mahendra[1], Ricky Eka Putra[2]

[1,2] Program Studi S1 Teknik Informatika, Universitas Negeri Surabaya
[1]yusril.20060@mhs.unesa.ac.id
[3]rickyeka@unesa.ac.id

Abstrak— *Cybercrime* telah menjadi isu yang meluas dengan berbagai pelanggaran seperti pencurian, penipuan, dan serangan terhadap sistem dan jaringan komputer. Serangan Distributed Denial of Service (DDoS) telah meningkat signifikan, membanjiri server dengan lalu lintas data sehingga pengguna yang sah tidak dapat mengakses layanan. XGBoost dan LightGBM adalah dua implementasi modern dari Gradient Boosted Decision Tree (GBDT), sebuah algoritma ensemble yang memadukan beberapa model regresi atau model pohon klasifikasi. GBDT mengintegrasikan fungsi-fungsi parameter sederhana dengan hasil yang 'buruk', atau tingkat kesalahan prediksi yang tinggi, untuk menghasilkan prediksi yang sangat akurat. Penelitian ini bertujuan untuk membuat model klasifikasi menggunakan dataset serangan DDoS dari CICDDoS2019 dengan menerapkan algoritma XGBoost dan LightGBM. Kedua algoritma tersebut dibandingkan berdasarkan akurasi untuk menghasilkan sistem yang akurat dalam mendeteksi serangan DDoS. Penelitian ini menghasilkan model dengan akurasi sebesar 97.62% untuk penggunaan algoritma XGBoost dan akurasi sebesar 97.39% untuk penggunaan algoritma LightGBM dengan selisih 0.23% lebih tinggi didapat oleh algoritma XGBoost.

Kata Kunci— DDoS, XGBoost, LightGBM, Hyperparameter, Website

I. PENDAHULUAN

Cybercrime telah menjadi isu yang meluas dengan komputer yang digunakan sebagai alat untuk melakukan aktivitas kriminal. Kejahatan ini mencakup berbagai pelanggaran termasuk pencurian, penipuan, dan serangan terhadap sistem dan jaringan komputer [1]. Dalam beberapa tahun terakhir, jumlah ancaman berbasis jaringan termasuk volume dan intensitas *DDoS* telah meningkat secara signifikan. Pada kuartal awal tahun 2022 terjadi peningkatan jumlah serangan *DDoS* yang signifikan. Peningkatan tersebut mencapai hampir 1,5 kali lipat relatif terhadap rekor sebelumnya, serta meningkat 4,5 kali lipat dibandingkan periode yang sama tahun sebelumnya. Krisis di Ukraina yang memicu perang dunia maya menjadi faktor utama di balik pertumbuhan ini. Distribusi serangan *DDoS* menunjukkan puncak baru terjadi pada kuartal pertama tahun 2022 [2].

Distributed Denial of Service (DDoS) merupakan serangan yang sederhana namun berpotensi merusak, dimana sumber daya dari target diserang sehingga pengguna yang sah tidak dapat mengakses layanan tersebut. Serangan ini melibatkan sejumlah sistem komputer yang berupaya membanjiri server dengan lalu lintas data sehingga server tidak mampu menangani permintaan yang datang. Perbedaan mendasar antara *DoS* dan *DDoS* terletak pada sumber serangan; *DoS* melibatkan satu komputer sementara *DDoS* melibatkan beberapa sistem komputer secara bersamaan [3].

Pertahanan terhadap serangan *DDoS* telah berkembang menjadi bagian integral dari teknologi perlindungan jaringan seperti *firewall* dan *IPS* yang semula hanya merupakan bagian dari infrastruktur. Namun, kendati *firewall* dapat menolak koneksi yang tidak diizinkan, mereka rentan terhadap serangan karena keterbatasan jumlah koneksi yang dapat mereka tangani serta keterbatasan dalam mendeteksi lalu lintas yang tidak seimbang. Solusi-solusi berbasis perangkat lunak yang berfokus pada pengembangan pola serangan (*signature*) juga memiliki keterbatasan, terutama dalam mendeteksi serangan DDoS yang baru dan dapat membanjiri jaringan dengan hasil false positive [4].

Pada Februari 2021, bursa kriptokurensi *EXMO* mengalami serangan *DDoS* dengan lalu lintas mencapai 30 GB per detik dan tidak dapat diakses selama 2 jam. Pada Desember 2020, situs pelacakan *Down Detector* juga mengalami gangguan akibat serangan *DDoS*. Serangan *DDoS* lainnya terjadi antara tahun 2018 hingga 2020. Menurut tim *NETSCOUT's ATLAS Security Engineering & Response Team (ASERT)*, pada kuartal pertama 2021, sekitar 2,9 juta serangan *DDoS* diluncurkan oleh pelaku ancaman, meningkat 31% dari periode yang sama pada tahun sebelumnya [9]. Laporan Ancaman Digital di Indonesia Tahun 2023 mengungkapkan bahwa selama periode Januari hingga Desember 2023, terjadi sejumlah 279,84 juta serangan siber di Indonesia. Meskipun jumlah ini menurun sebesar 24,4% dari tahun sebelumnya yang mencapai 370,02 juta serangan siber, ancaman digital tetap signifikan dan memerlukan perhatian serius dalam menjaga keamanan sistem dan data di negara ini [5].

Deteksi serangan *DDoS* menjadi krusial dalam melawan ancaman tersebut, dengan dua teknik utama yang digunakan : deteksi penyalahgunaan dan deteksi anomali. Teknik deteksi penyalahgunaan membandingkan aktivitas jaringan saat ini dengan pola serangan yang telah diketahui, namun sulit untuk mendeteksi serangan baru. Sebaliknya, deteksi anomali membandingkan aktivitas jaringan saat ini dengan pola aktivitas normal yang telah ditetapkan, dengan menggunakan *machine learning* untuk membuat model aktivitas normal dan mengidentifikasi aktivitas yang tidak biasa sebagai potensi serangan [6].

Penelitian mengenai ancaman *DDoS* telah banyak dilakukan sebelumnya dengan berbagai metode yang menghasilkan bermacam-macam kesimpulan. Penelitian Rozam tentang *XGBoost Classifier for DDOS Attack Detection in Software Defined Network Using sFlow Protocol* menunjukkan akurasi tinggi, tingkat false positive rendah, kecepatan deteksi yang baik, dan dapat diimplementasikan secara skalabel pada jaringan SDN [7]. *XGBoost*, yang merupakan metode pembelajaran *ensemble*. Metode ini memiliki lebih dari satu pengklasifikasi untuk meminimalkan kesalahan prediksi (*False Negative*).

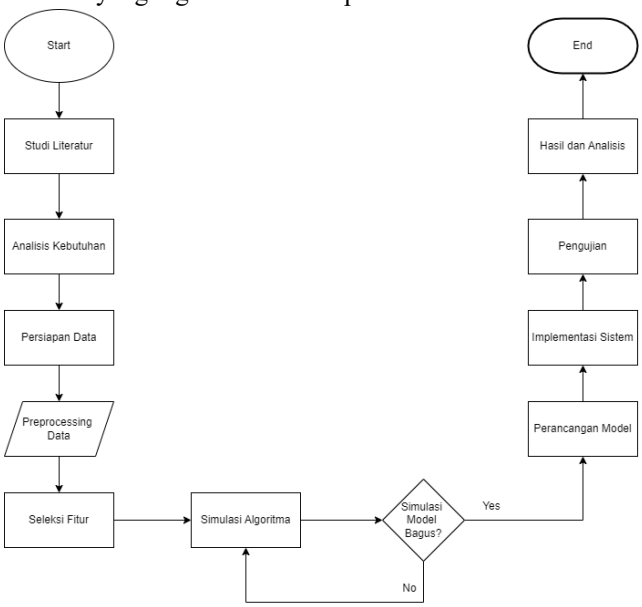
Penelitian yang serupa juga dilakukan oleh Kabirat dengan judul *LightGBM-DDoS: Intelligent Model for Detecting Distributed Denial of Service Attacks in Software-Defined Networking* menghasilkan sebuah model yang mampu mendeteksi serangan *DDoS* dalam jaringan SDN dengan tingkat akurasi sebesar 0.9972, tingkat deteksi sebesar 0.9940, skor F1 sebesar 0.9970, dan tingkat kesalahan sebesar 0.00601, yang lebih unggul dibandingkan dengan metode lain seperti *XGBoost*, *Logistic Regression*, dan *KNearest Neighbours*. Pada penelitian *LightGBM* menjadi algoritma terbaik yang dimana *LightGBM* adalah algoritma *GBDT (Gradient Boosting Decision Tree)* yang gratis dan bersifat *opensource* dari Microsoft. Algoritma berbasis histogram seperti yang digunakan dalam teknik pohon keputusan voting paralel digunakan untuk mempercepat pelatihan, mengurangi penggunaan memori, dan menggabungkan konektivitas jaringan terbaru untuk memaksimalkan potensi pembelajaran paralel.

XGBoost (Extreme Gradient Boosting) dan *LightGBM (Light Gradient Boosting Machine)* adalah dua algoritma canggih yang merupakan pengembangan lebih lanjut dari *Gradient Boosted Decision Trees (GBDT)*. *GBDT* adalah teknik *ensemble* yang menggabungkan prediksi dari beberapa model keputusan pohon untuk meningkatkan akurasi dan mengurangi kesalahan prediksi. *XGBoost* dirancang untuk memaksimalkan kecepatan dan efisiensi melalui teknik-teknik seperti pemangkasan pohon *(tree pruning)*, penanganan otomatis terhadap *missing values*, dan kemampuan paralel yang kuat. Algoritma ini terkenal karena kemampuannya menangani data dengan skala besar dan berbagai jenis data, serta memberikan hasil yang sangat baik dalam berbagai kompetisi *data mining* dan *machine learning*. *LightGBM*, yang dikembangkan oleh Microsoft, menggunakan teknik berbasis histogram untuk mempercepat proses pelatihan dan mengurangi penggunaan memori. *LightGBM* juga mendukung paralelisasi dan skalabilitas tinggi, menjadikannya pilihan ideal untuk *dataset* besar dan kompleks. Algoritma ini unggul dalam kecepatan dan efisiensi serta mampu menangani fitur kategori dan nilai yang hilang dengan sangat baik[9].

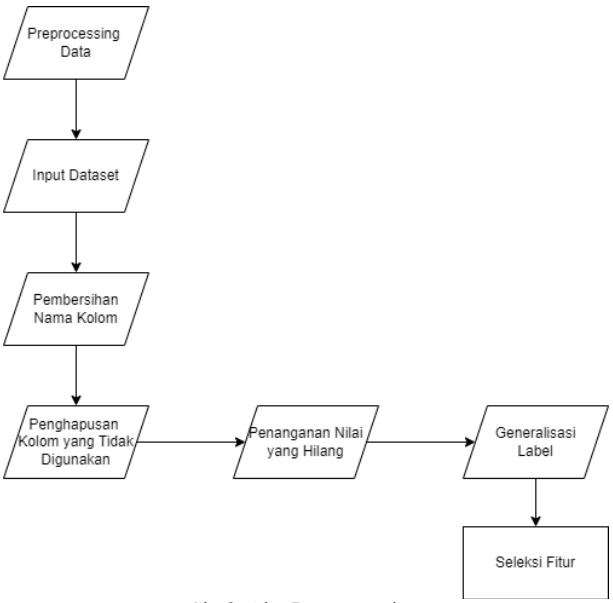
Oleh karena itu, tujuan utama penelitian adalah membuat model klasifikasi dengan menggunakan dataset serangan *DDoS* yang berasal dari *CICDDoS2019* untuk dianalisis lebih lanjut dengan menerapkan algoritma *XGBoost* dan *LightGBM* yang merupakan bagian dari algoritma *Gradient Boosted Decision Tree* untuk dibandingkan perhitungan akurasi dari kedua algoritma sehingga dapat diimplementasikan menjadi sistem yang akurat.

I. METODE PENELITIAN

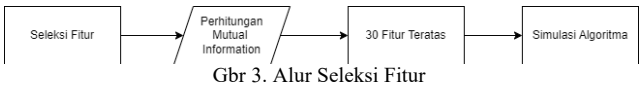
Metode penelitian yang digunakan dalam studi ini adalah kuantitatif, yang melibatkan pengumpulan dan analisis data numerik untuk mengidentifikasi pola, hubungan, dan tren yang ada dalam fenomena yang diteliti [8]. Berikut adalah alur penelitian yang digunakan dalam penelitian ini :



Gbr 1. Alur Penelitian



Gbr 2. Alur Prerprocessing



Gbr 3. Alur Seleksi Fitur

A. Studi Literatur

Tahapan awal dalam penelitian ini adalah studi literatur yang melibatkan eksplorasi berbagai sumber seperti buku, artikel, serta jurnal internasional dan nasional terkait topik *Distributed Denial-of-Service (DDoS)* dan teknik pengklasifikasian menggunakan machine learning. Analisis mendalam terhadap penggunaan machine learning dalam mengklasifikasikan serangan *DDoS* dilakukan untuk memahami tren terbaru dalam deteksi dan mitigasi serangan tersebut. Peneliti mengidentifikasi dan mengevaluasi metode-metode yang telah digunakan sebelumnya untuk membangun kerangka kerja yang komprehensif sebagai panduan langkah-langkah penelitian selanjutnya. Melalui studi literatur ini, peneliti juga memperoleh pemahaman mendalam tentang teknik-teknik *machine learning* yang telah diterapkan dalam konteks pengklasifikasian serangan *DDoS*, menjadi landasan penting dalam pemilihan dan pengembangan model yang tepat [10]. *XGBoost* dan *LightGBM* adalah dua metode machine learning yang dipertimbangkan dalam penelitian ini. *XGBoost*, yang memenangkan beberapa kompetisi di Kaggle, dikenal karena kinerjanya yang sangat baik, sementara *LightGBM*, dengan kecepatan pelatihan tinggi, efisiensi memori, dan kemampuan menangani hubungan non-linear, juga merupakan alternatif menarik. Dengan membandingkan kedua metode ini, penelitian ini bertujuan untuk memahami keunggulan dan kelemahan masing-masing pendekatan serta memilih metode yang paling sesuai untuk analisis serangan *DDoS*.

B. Analisis Kebutuhan

Analisis kebutuhan dalam penelitian ini bertujuan untuk menentukan detail kebutuhan dalam klasifikasi serangan menggunakan algoritma *machine learning XGBoost* dan *LightGBM*. Untuk mencapai tujuan penelitian, diperlukan perangkat lunak pengembangan dan pengujian seperti *Visual Studio Code, Microsoft Edge*, dan *Hping3*. Kebutuhan ini mencakup aspek fungsional dan non-fungsional. Kebutuhan fungsional mencakup kemampuan sistem untuk mengklasifikasikan serangan *DDoS* dan menampilkan output klasifikasi data testing, sementara kebutuhan non-fungsional menekankan bahwa sistem harus berjalan dengan lancar [11]. Kombinasi kebutuhan ini akan memastikan bahwa penelitian dapat dilakukan sesuai dengan tujuan yang diharapkan.

C. Persiapan Data

Persiapan data dilakukan untuk mendapatkan informasi yang diperlukan dalam pengembangan sistem. Salah satu metode yang digunakan adalah studi literatur, di mana data-data disiapkan dan referensi dari berbagai sumber yang relevan dengan penelitian dianalisis. *Dataset* yang digunakan dalam penelitian ini berasal dari *CICDDoS2019* (*Canadian Institute for Cybersecurity*) dan diunduh dari Kaggle. Dataset *CICDDoS2019* terdiri dari 88 fitur. *CICDDoS2019* berisi data *benign* dan data terbaru mengenai *common DDoS Attacks*. Dataset ini juga berisi hasil analisis lalu lintas jaringan menggunakan *CICFlowMeter-V3* dengan aliran yang diberi label berdasarkan waktu, IP sumber dan tujuan, port sumber dan tujuan, protokol, dan jenis serangan. Pada dataset ini juga terdapat berbagai serangan *DDoS* reflektif modern seperti *PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS*, dan *SNMP*.

D. Preprocessing Data

Preprocessing data merupakan tahap awal yang sangat penting dalam perancangan model untuk melatih dan menguji model klasifikasi serangan *DDoS*. Proses ini mencakup berbagai langkah untuk membersihkan, menyesuaikan, dan mengubah dataset agar sesuai dengan kebutuhan analisis, yang pada akhirnya dapat meningkatkan kualitas dan akurasi model klasifikasi. Langkah-langkah spesifik dalam preprocessing data meliputi pembersihan nama kolom, penghapusan kolom yang tidak diperlukan, penanganan nilai yang hilang, pengubahan label, dan seleksi fitur. Dengan melakukan preprocessing data yang tepat, kita dapat memastikan bahwa dataset siap digunakan untuk analisis lebih lanjut dan pengembangan model machine learning yang efektif.

E. Seleksi Fitur

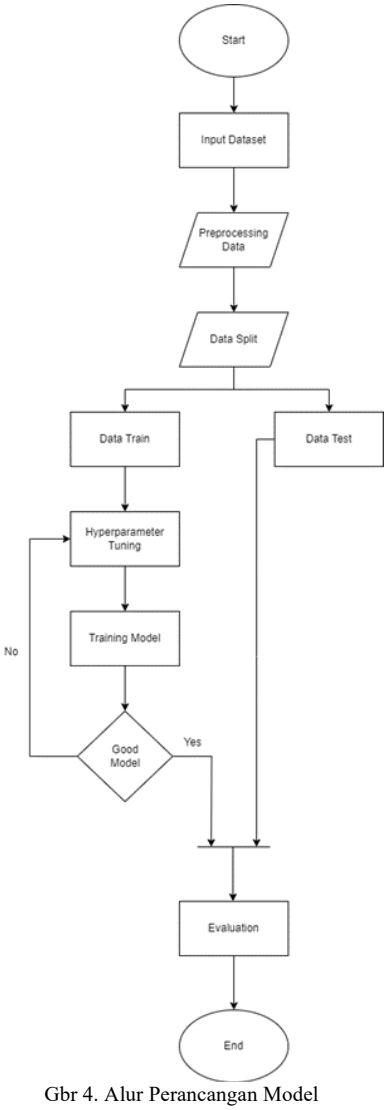
Proses seleksi fitur dilakukan terhadap dataset *CICDDoS2019* dengan pertimbangan akan dipergunakan 10-30 fitur teratas sebagai landasan dalam penyusunan model agar akurasi serta pengujian model dapat dieksplorasi secara lengkap dan mendalam untuk mendapatkan hasil terbaik. Metode seleksi fitur yang digunakan adalah *Mutual Information* yakni metrik yang berguna untuk mengevaluasi seberapa baik variabel prediktor (fitur) berhubungan dengan variabel target dalam sebuah dataset. Metrik ini mengukur seberapa banyak informasi tentang variabel target yang diberikan oleh suatu fitur [12].

F. Simulasi Metode

Simulasi metode dilakukan dengan percobaan menggunakan 10 fitur teratas yang didapat dari proses Seleksi Fitur yang selanjutnya dipilih metode terbaik dari kedua metode dengan melihat hasil akurasi, presisi, *recall*, serta *f1-score* untuk selanjutnya digunakan dalam tahap perancangan model.

G. Perancangan Sistem

Perancangan model dalam penelitian ini mencakup pengembangan model machine learning menggunakan algoritma *XGBoost* dan *LightGBM* untuk mengklasifikasikan serangan *DDoS*, serta pembuatan antarmuka yang memungkinkan pengujian dan evaluasi hasil klasifikasi secara efisien [13]. Sistem ini dirancang untuk memastikan performa yang optimal dan kemudahan penggunaan, dengan mempertimbangkan kebutuhan fungsional dan non-fungsional yang telah diidentifikasi sebelumnya. Berikut adalah alur perancangan model pada penelitian ini :



Gbr 4. Alur Perancangan Model

H. Implementasi Sistem

Tahap implementasi sistem adalah langkah di mana model terbaik hasil dari perancangan dan evaluasi diintegrasikan ke dalam sebuah aplikasi [14]. Dalam penelitian ini, model yang telah melalui proses training, hyperparameter tuning, dan evaluasi, kemudian diimplementasikan dalam bentuk web. Website ini dikembangkan menggunakan framework Streamlit, yang memungkinkan pembuatan antarmuka web interaktif dengan mudah untuk pengguna non-teknis.

II. HASIL DAN PEMBAHASAN

A. Perbandingan Algoritma XGBoost dan LightGBM menggunakan Hyperparameter GridCV

1) Data Description

Dataset yang digunakan dalam penelitian ini berasal dari *CICDDoS2019 (Canadian Institute for Cybersecurity)* dan diunduh dari Kaggle. Dataset *CICDDoS2019* terdiri dari 88 fitur. *CICDDoS2019* berisi data *benign* dan data terbaru mengenai *common DDoS Attacks*. 88 kolom/fitur akan diseleksi kembali untuk diambil 30 fitur teratas yang paling berpengaruh guna menghasilkan model yang terbaik.

2) Preprocessing Data

Pada tahap ini, dataset *CICDDoS2019* diolah melalui beberapa langkah *preprocessing* untuk memastikan data siap digunakan dalam pelatihan model *machine learning*. Berikut adalah tahap *preprocessing* yang dilakukan :

a) Pembersihan Nama Kolom
b) Penghapusan Kolom yang Tidak Diperlukan
c) Penanganan Nilai yang Hilang
d) Generalisasi Label

3) Seleksi Fitur

Pada tahap ini penggunaan fitur dilakukan seleksi sesuai dengan hasil dari proses seleksi fitur dengan menggunakan *Mutual Information* [13]. Perhitungan setiap kolom dengan Mutual Information menggunakan rumus persamaan sebagai berikut :

$$MI(x,A) = log(\frac{N \times f(x,A)}{f(x,A)+f(x,-A)+f(x,A)+f(-x,A)}) \quad (1)$$

Keterangan :
$f(x,A)$ = Frekuensi term x dalam kelas A
$f(-x,A)$ = Frekuensi term lain dalam kelas A
$f(x,-A)$ = Frekuensi term x di kelas selain A
N = Total jumlah term
Hasil seleksi di berdasarkan *score* yang diberikan oleh *Mutual Information* :

TABEL I
HASIL SELEKSI FITUR DENGAN MUTUAL INFORMATION

No.	Features	Type	Score
1.	Source IP	Categorical	0.551376
2.	Destination IP	Categorical	0.450541
3.	Inbound	Categorical	0.424765
4.	Average Packet Size	Numeric	0.272924
5.	Packet Length Mean	Numeric	0.271404
6.	Fwd Packet Length Mean	Numeric	0.268712
7.	Avg Fwd Segment Size	Numeric	0.268418
8.	Fwd Packets/s	Numeric	0.266532
9.	Max Packet Length	Numeric	0.262067

No.	Features	Type	Score
10.	Destination Port	Categorical	0.259885
11.	Fwd Packet Length Max	Numeric	0.256399
12.	Init_Win_bytes_forward	Categorical	0.252515
13.	Subflow Fwd Bytes	Numeric	0.244678
14.	Total Length of Fwd Packets	Numeric	0.244281
15.	Fwd Packet Length Min	Numeric	0.243307
16.	Min Packet Length	Numeric	0.243159
17.	Flow Bytes/s	Numeric	0.240048
18.	Flow Packets/s	Numeric	0.210000
19.	Flow IAT Mean	Numeric	0.208316
20.	Flow IAT Max	Numeric	0.200112
21.	Flow IAT Std	Numeric	0.193688
22.	Bwd Packets/s	Numeric	0.193044
23.	Flow Duration	Numeric	0.184624
24.	Packet Length Std	Numeric	0.178187
25.	Packet Length Variance	Numeric	0.178130
26.	Source Port	Categorical	0.171364
27.	Protocol	Categorical	0.167626
28.	Subflow Bwd Bytes	Numeric	0.136767
29.	Total Length of Bwd Packets	Numeric	0.136511
30.	Fwd IAT Max	Numeric	0.134685
31.	Bwd Packet Length Mean	Numeric	0.132804
32.	Avg Bwd Segment Size	Numeric	0.132663
33.	Bwd Packet Length Max	Numeric	0.131818
34.	Fwd IAT Mean	Numeric	0.130007
35.	Bwd IAT Max	Numeric	0.128090
36.	Bwd IAT Total	Numeric	0.126310
37.	Bwd Header Length	Numeric	0.125400

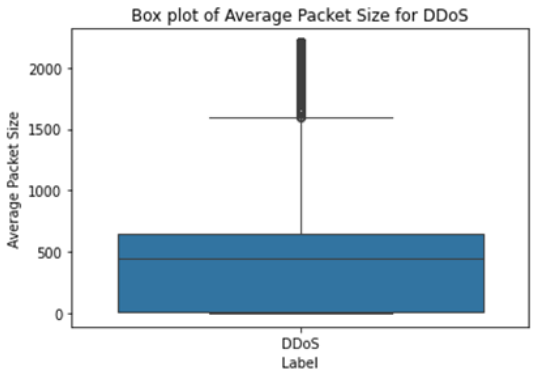
30 fitur numerik teratas diambil untuk digunakan dalam proses selanjutnya yakni *hyperparameter*. Fitur seleksi dibagi menjadi 3 untuk didapatkan hasil terbaik sebanyak 10, 20, 30 fitur agar bisa dibandingkan akurasi yang dihasilkan ketika dilakukan training dengan algoritma *XGBoost* dan *LightGBM*.

4) Resampling Data

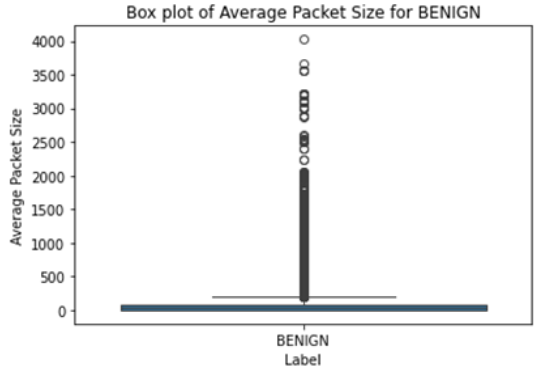
Teknik *Resampling* yang digunakan adalah *Oversampling* dan *Undersampling*. *Undersampling* dilakukan terhadap baris data yang berlabel *DDoS* sebanyak 150.000 dan *Oversampling* dilakukan terhadap baris data yang berlabel *Benign* sebanyak 150.000 untuk selanjutnya digabungkan kembali menjadi sebuah *dataframe* guna digunakan pada langkah selanjutnya. Proses ini melibatkan pengambilan sampel data '*DDoS*' tanpa penggantian, sehingga setiap baris yang dipilih adalah unik, sedangkan data '*BENIGN*' diambil dengan penggantian, memungkinkan beberapa baris untuk dipilih lebih dari sekali. Dengan *random_state*=42, hasil resampling dapat direproduksi. Kriteria pengambilan data memastikan bahwa setiap kelas memiliki jumlah yang sama, menciptakan *dataframe* seimbang yang memudahkan analisis lebih lanjut [15].

5) Penanganan Nilai Bias pada Data

Pada tahap ini dataset yang telah dilakukan *resample* kemudian diamati ulang untuk dievaluasi Kembali sehingga nilai bias yang ada dalam data bisa dihilangkan lebih dalam lagi agar performa dari model dapat maksimal dan sesuai dengan tujuan penelitian. Berikut adalah hasil pengamatan terhadap *outlier* dengan visualisasi menggunakan *box plot* :



Gbr 5. Box Plot Label DDoS



Gbr 6. Box Plot Label BENIGN

Berdasarkan bukti visual dari *box plot*, terlihat jelas bahwa terdapat nilai-nilai ekstrim pada label *DDoS* dan *BENIGN* yang dapat dianggap sebagai *outlier* [16]. *Outlier* ini dapat mempengaruhi proses pelatihan model secara signifikan, yang dapat menyebabkan *overfitting* atau bias sehingga dilakukanlah penghapusan baris nilai tersebut.

6) Penerapan Hyperparameter Tuning

a. Penerapan *hyperparameter tuning* pada *XGBoost* dalam penelitian ini melibatkan beberapa langkah utama. Pertama, parameter grid didefinisikan dengan fokus pada 'n_estimators', 'learning_rate', 'max_depth', 'subsample', dan 'gamma'. Nilai default dari parameter tersebut digunakan sebagai acuan untuk menentukan kombinasi parameter yang akan diuji, seperti 'n_estimators' dengan nilai 300, 500, dan 700; 'learning_rate' dengan nilai 0.1, 0.3, dan 0.5; 'max_depth' dengan nilai 2, 4, dan 6; 'subsample' dengan nilai 0.5, 0.7, dan 0.8; serta 'gamma' dengan nilai 0, 1, dan 2. Model *XGBoost* diinisialisasi dengan tree_method='gpu_hist' untuk memanfaatkan GPU, kemudian *GridSearchCV* digunakan dengan *2-fold cross-validation* dan *5-fold cross-validation* untuk menemukan kombinasi parameter terbaik berdasarkan akurasi.

b. Untuk *LightGBM*, *hyperparameter tuning* dilakukan dengan mendefinisikan parameter grid yang meliputi 'n_estimators', 'learning_rate', 'max_depth', 'subsample', dan 'num_leaves'. Nilai default parameter digunakan sebagai dasar untuk menentukan kombinasi yang diuji, seperti 'n_estimators' dengan nilai 300, 500, dan 700; 'learning_rate' dengan nilai 0.1, 0.3, dan 0.5; 'max_depth' dengan nilai 2, 4, dan 6; 'subsample' dengan nilai 0.5, 0.7, dan 0.8; serta 'num_leaves' dengan nilai 4, 16, dan 64. *LightGBM classifier* diinisialisasi dengan device='gpu' untuk memanfaatkan GPU, dan *GridSearchCV* diterapkan dengan *2-fold cross-validation* dan *5-fold cross-validation* menggunakan metrik akurasi untuk mengevaluasi performa model.

Kedua proses tuning ini diterapkan pada masing-masing skenario fitur (10, 20, dan 30 fitur) untuk kedua algoritma, menghasilkan total enam percobaan *Grid Search*. Hasil *grid search* ditampilkan dalam bentuk dataframe yang menunjukkan parameter, *mean test score, standard deviation dari test score, mean fit time,* dan *mean score time.* Pemilihan parameter terbaik dari hasil *grid search* ini bertujuan untuk mengoptimalkan kinerja model dalam mengklasifikasikan serangan *DDoS* secara efisien dan akurat [15]. Berikut adalah *Mean Test Score* dan *Mean Fit Time* terbaik dari setiap kombinasi fitur untuk *fold = 2* dan *fold = 5* :

TABEL II
MEAN TEST SCORE & MEAN FIT TIME UNTUK HYPERPARAMETER DENGAN FOLD 2

No.	Fitur	Param	Mean Test Score	Mean Fit Time
1.	XGBoost 10 Fitur	{'gamma': 0, 'learning_rate': 0.5, 'max_depth': 2, 'n_estimators': 700, 'subsample': 0.8}	0,999872332	1,348502159
2.	XGBoost 20 Fitur	{'gamma': 0, 'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 500, 'subsample': 0.7}	0,999861693	1,584114313
3.	XGBoost 30 Fitur	{'gamma': 0, 'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 500, 'subsample': 0.8}	0,999872332	1,684500337
4.	LightGBM 10 Fitur	{'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 300, 'num_leaves': 64, 'subsample': 0.8}	0,999861693	47,913000237
5.	LightGBM 20 Fitur	{'learning_rate': 0.1, 'max_depth': 2, 'n_estimators': 700, 'num_leaves': 4, 'subsample': 0.5}	0,999877652	22,181999968
6.	LightGBM 30 Fitur	{'learning_rate': 0.1, 'max_depth': 2,	0,999861693	22,245

No.	Fitur	Param	Mean Test Score	Mean Fit Time
		'n_estimators': 700, 'num_leaves': 4, 'subsample': 0.5}		

TABEL III
MEAN TEST SCORE & MEAN FIT TIME UNTUK HYPERPARAMETER DENGAN FOLD 5

No.	Fitur	Param	Mean Test Score	Mean Fit Time
1.	XGBoost 10 Fitur	{'gamma': 0, 'learning_rate': 0.3, 'max_depth': 4, 'n_estimators': 500, 'subsample': 0.7}	0,999931	1,7664
2.	XGBoost 20 Fitur	{'gamma': 0, 'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 300, 'subsample': 0.8}	0,999926	1,498199
3.	XGBoost 30 Fitur	{'gamma': 0, 'learning_rate': 0.3, 'max_depth': 6, 'n_estimators': 500, 'subsample': 0.8}	0,99992	2,577774
4.	LightGBM 10 Fitur	{'learning_rate': 0.1, 'max_depth': 2, 'n_estimators': 700, 'num_leaves': 4, 'subsample': 0.5}	0,999926	23,9176
5.	LightGBM 20 Fitur	{'learning_rate': 0.5, 'max_depth': 6, 'n_estimators': 300, 'num_leaves': 16, 'subsample': 0.7}	0,999931	17,68
6.	LightGBM 30 Fitur	{'learning_rate': 0.1, 'max_depth': 4, 'n_estimators': 300, 'num_leaves': 64, 'subsample': 0.7}	0,999931	28,1158

Dari percobaan diatas dapat disimpulkan bahwa penggunaan *fold = 5* lebih baik dari pada *fold = 2* dengan selisih *Mean Test Score* untuk *XGBoost* 10, 20, 30 fitur secara berurutan sebesar 0.000058515, 0.000063834, 0.000047876 dan untuk *LightGBM* dengan 10, 20,30 fitur secara berurutan sebesar 0.000063834, 0.000053195, 0.000069154. Dari percobaan yang dilakukan dapat disimpulkan bahwa penggunaan *fold* = 5 pada hyperparameter menghasilkan *Mean Test Score* yang lebih tinggi dari pada *fold* = 2. *Mean Fit Time* menunjukkan hasil yang berbeda-beda dengan didominasi oleh *fold* = 2 yang lebih cepat dikarenakan proses iterasi yang lebih singkat sehingga hanya dijadikan sebagai metrik tambahan.

7) Training Model XGBoost dan LightGBM

Training model dilakukan dengan menggunakan kombinasi parameter terbaik yang didapat dari *Hyperparameter* berdasarkan akurasi [17]. Berikut adalah hasil akurasi dari setiap fitur yang dilatih menggunakan parameter terbaik masing-masing :

a. XGBoost

TABEL IV
HASIL TRAINING ALGORITMA XGBOOST

No.	Jumlah Fitur	Parameter	Akurasi
1.	10	n_estimators=500 max_depth=4 learning_rate=0.3 subsample=0.7 gamma=0	99.99574440921761 %
2.	20	n_estimators=300 max_depth=6	99.99574440921761 %

No.	Jumlah Fitur	Parameter	Akurasi
		learning_rate=0.3 subsample=0.8 gamma=0	
3.	30	n_estimators=500 max_depth=6 learning_rate=0.3 subsample=0.8 gamma=0	99.99787220460881 %

b. LightGBM

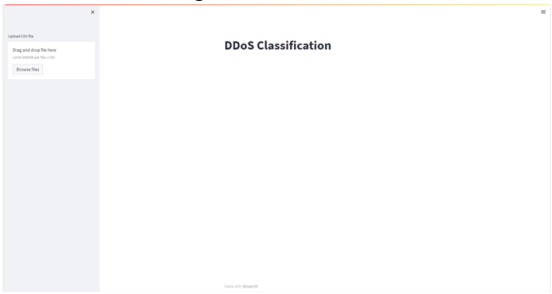
TABEL V
HASIL TRAINING ALGORITMA LIGHTGBM

No.	Jumlah Fitur	Parameter	Akurasi
1.	10	n_estimators=700 max_depth=2 learning_rate=0.1 subsample=0.5 num_leaves=4	99.99574440921761 %
2.	20	n_estimators=700 max_depth=6 learning_rate=0.5 subsample=0.7 num_leaves=16	99.99574440921761 %
3.	30	n_estimators=300 max_depth=4 learning_rate=0.1 subsample=0.7 num_leaves=64	99.99574440921761 %

Dari percobaan yang telah dilakukan didapatkan model terbaik yakni *XGBoost* yang menggunakan 30 fitur teratas dari *Mutual Information* dengan akurasi sebesar 99.99787220460881 % dan *LightGBM* yang menggunakan 10, 20, 30 fitur teratas dari *Mutual Information* dengan akurasi sama sebesar 99.99574440921761 %. *XGBoost* mendapatkan skor lebih tinggi dengan selisih 0.00212.779539120 %.

8) Implementasi Model ke dalam Bentuk Website

Model terbaik yang didapat dari proses *Training Data* selanjutnya dikonversi kedalam bentuk file *Pickle* sehingga bisa diimplementasikan ke dalam website yang dibuat menggunakan *framework Streamlit*. Berikut adalah tampilan antarmuka dari implementasi model :



Gbr 6. Antarmuka Implementasi

9) Pengujian Model dengan Data Tes DDoS Skenario

Pengujian dilakukan sebanyak 3 kali dengan pengumpulan data dari skenario *DDoS*, *DoS*, dan *Benign* yang didapat dari tools hping3 kemudian dilakukan sniffing dengan tools tcpdump yang kemudian dilakukan konversi menjadi *dataset test* dengan menggunakan *CICFlowmeter* untuk melihat performa dari model ketika berhadapan dengan skenario yang menyerupai real case sehingga dapat dilakukan evaluasi [18] Berikut langkah-langkah yang dilakukan pada skenario :

a) *Botnet* melakukan serangan ke mesin korban
b) Mesin korban melakukan *capture traffic network* dengan menggunakan *tcpdump* menjadi sebuah ekstensi file PCAP
c) File pcap hasil *capture traffic tcpdump* dikonversi menjadi dataset tes skenario menggunakan *CICFlowmeter* menjadi ekstensi file CSV

Berikut adalah tabel lengkap hasil akurasi dari *Confusion Matrix* untuk setiap skenario :

TABEL VI
HASIL PERHITUNGAN CONFUSION MATRIX

Skenario	XGBoost	LightGBM
Tes 1 (DDoS)	97.62%	97.39%
Tes 2 (DoS)	95.63%	94.11%
Tes 3 (Benign)	87.71%	75.43%

Terlihat pada tabel bahwa akurasi dari model *XGBoost* lebih baik daripada *LightGBM* dalam semua skenario pengujian dengan selisih 0.23% untuk skenario 1, 1.52% untuk skenario 2 dan 11.28% untuk skenario 3.

III. KESIMPULAN

Berdasarkan hasil dan pembahasan penelitian "Penerapan Algoritma *Gradient Boosted Decision Tree (GBDT)* Untuk Klasifikasi Serangan *DDoS*", dapat disimpulkan bahwa

penelitian ini menggunakan dataset publik dari Kaggle dengan 1.130.650 data dan 88 kolom, serta menguji tiga skema Hyperparameter *XGBoost* dan *LightGBM* dengan 10, 20, dan 30 fitur. Hasil terbaik untuk *XGBoost* dengan 30 fitur adalah n_estimators=500, max_depth=6, learning_rate=0.3, subsample=0.8, gamma=0, mencapai akurasi 99.99787220460881%, sedangkan *LightGBM* dengan 30 fitur adalah n_estimators=300, max_depth=4, learning_rate=0.1, subsample=0.7, num_leaves=64, mencapai akurasi 99.99574440921761%. *XGBoost* menunjukkan hasil lebih baik daripada *LightGBM* dengan selisih 0.00212779539120%. Implementasi model diuji dengan tiga skenario serangan, di mana *XGBoost* consistently menunjukkan akurasi lebih tinggi: skenario tes 1 (*XGBoost* 97.62%, *LightGBM* 97.39%, selisih 0.23%), skenario tes 2 (*XGBoost* 95.63%, *LightGBM* 94.11%, selisih 1.52%), dan skenario tes 3 (*XGBoost* 87.71%, *LightGBM* 75.43%, selisih 11.28%). Kombinasi parameter yang tepat meningkatkan akurasi dan kinerja model secara signifikan, sehingga model yang dihasilkan dapat digunakan secara efektif dalam aplikasi web untuk mendeteksi serangan *DDoS* dengan keandalan yang sangat tinggi.

IV. SARAN

Penelitian ini memberikan beberapa saran untuk pengembangan lebih lanjut. Pertama, disarankan untuk menggunakan algoritma *GBDT* lainnya guna memperoleh hasil penelitian yang berbeda agar dapat dibandingkan dan ditemukan hasil yang terbaik. Kedua, untuk meningkatkan akurasi serta kinerja model dari kedua algoritma, disarankan menggunakan kombinasi parameter yang lebih banyak serta fitur yang lebih beragam. Ketiga, pada bagian validasi, disarankan untuk menggunakan *Hold-Out Validation* untuk mendapatkan wawasan baru dari hasil penelitian. Terakhir, penelitian ini diharapkan dapat dilanjutkan dan dikembangkan menjadi sebuah aplikasi *IDS (Intrusion Detection System)*.

REFERENSI

[1] Arif Wirawan Muhammad, Muhammad Nur Faiz, & Ummi Athiyah. (2022). Pengembangan Perangkat Lunak Untuk Deteksi DDoS Berbasis Neural Network. Infotekmesin, 13(2), 301–307. https://doi.org/10.35970/infotekmesin.v13i2.1544

[2] Ayuningtyas, A. D. (n.d.). DETEKSI SERANGAN DISTRIBUTED DENIAL OF SERVICE (DDOS) MENGGUNAKAN CATBOOST CLASSIFIER.

[3] Firmansyah, H., & Abidin, Z. (2022). PENERAPAN ALGORITMA GRADIENT BOOSTED DECISION TREES PADA ADABOOST UNTUK KLASIFIKASI STATUS DESA. 1(1).

[4] Herni Yulianti, S. E., Oni Soesanto, & Yuana Sukmawaty. (2022). Penerapan Metode Extreme Gradient Boosting (XGBOOST) pada Klasifikasi Nasabah Kartu Kredit. Journal of Mathematics: Theory and Applications, 21–26. https://doi.org/10.31605/jomta.v4i1.1792

[5] Irham, L. G., Adiwijaya, A., & Wisesty, U. N. (2019). Klasifikasi Berita Bahasa Indonesia Menggunakan Mutual Information dan Support Vector Machine. JURNAL MEDIA INFORMATIKA BUDIDARMA, 3(4), 284. https://doi.org/10.30865/mib.v3i4.1410

[6] Kabirat, K. M., Olaniyi, A. D., Adebukola, O. S., & Kehinde, T. O. (2023). LightGBM-DDoS: Intelligent Model for Detecting Distributed Denial of Service Attacks in Software-Defined Networking.

[7] Laila Qadrini, Andi Seppewali, & Asra Aina. (2021). Decision Tree dan Adaboost pada Klasifikasi Penerima Program Bantuan Sosial. Jurnal Inovasi Penelitian, 2(7), 1959–1966. https://doi.org/10.47492/jip.v2i7.1046

[8] Maulana, I. (n.d.). Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer.

[9] Mittal, M., Kumar, K., & Behal, S. (2023). Deep learning approaches for detecting DDoS attacks: A systematic review. Soft Computing, 27(18), 13039–13075. https://doi.org/10.1007/s00500-021-06608-1

[10] Pavlík, B. Š. (2020). Identification of Network Traffic of Communication Applications.

[11] Praptodiyono, S., Firmansyah, T., Anwar, M. H., Wicaksana, C. A., Pramudyo, A. S., & Al-Allawee, A. (2023). Development of hybrid intrusion detection system based on Suricata with pfSense method for high reduction of DDoS attacks on IPv6 networks. Eastern-European Journal of Enterprise Technologies, 5(9 (125)), 75–84. https://doi.org/10.15587/1729-4061.2023.285275

[12] Rozam, N. F., & Riasetiawan, M. (2023). XGBoost Classifier for DDOS Attack Detection in Software Defined Network Using sFlow Protocol. International Journal on Advanced Science, Engineering and Information Technology, 13(2), 718–725. https://doi.org/10.18517/ijaseit.13.2.17810

[13] Salsabila, S. (n.d.). MODUL DATA MINING FUTURE SELECTION PERTEMUAN 11 (ONLINE).

[14] Santoso, D., Noertjahyana, A., & Andjarwirawan, J. (n.d.). Implementasi dan Analisa Snort dan Suricata Sebagai IDS dan IPS Untuk Mencegah Serangan DOS dan DDOS.

[15] Tandon, R. (2020). A Survey of Distributed Denial of Service Attacks and Defenses. https://doi.org/10.48550/ARXIV.2008.01345

[16] Tiwari, K. K. N. (2020). Denial of Service attack using Slowloris. 07(07).

[17] Wicaksana, M. R. N. (n.d.). JURUSAN SISTEM KOMPUTER FAKULTAS ILMU KOMPUTER UNIVERSITAS SRIWIJAYA 2023.

[18] Zidane, M. (n.d.). Klasifikasi Serangan Distributed Denial-of-Service (DDoS) menggunakan Metode Data Mining Naïve Bayes.