PENINGKATAN AKURASI DETEKSI DDOS SDN MENGGUNAKAN ALGORITMA *RANDOM FOREST* BERBASIS *GAIN RATIO*

Reza Pahlevi[1], Latifah Nur Zam Zami[2], Imelda Imelda[3]
Program Magister Ilmu Komputer, Fakultas Teknologi Informasi
Universitas Budi Luhur
Jalan Ciledug Raya, RT.10/RW.2, Petukangan Utara, Kecamatan Pesanggrahan, Jakarta Selatan
rezalevi2490@gmail.com[1], latifahnurzam9d@gmail.com[2], imelda@budiluhur.ac.id[3]

Abstrak

Perkembangan teknologi *Software-Defined Networking* (SDN) membawa fleksibilitas tinggi dalam pengelolaan jaringan, namun juga membuka celah keamanan terhadap serangan *Distributed Denial of Service* (DDoS). Serangan ini dapat melumpuhkan *controller* SDN melalui banjir paket data, sehingga mengakibatkan penurunan performa jaringan secara drastis. Penelitian ini bertujuan untuk meningkatkan akurasi deteksi serangan DDoS pada arsitektur SDN berbasis *Ryu Controller* dengan protokol OpenFlow v1.3 serta dengan mengimplementasikan algoritma *Random Forest* yang dioptimasi dengan seleksi fitur *Gain Ratio*. Keunggulan penelitian ini terletak pada penggunaan *dataset* primer yang dikumpulkan melalui simulasi *testbed* SDN mandiri menggunakan Mininet dengan topologi *Modified Tree* untuk menghasilkan skenario serangan yang lebih aktual. Metode *Gain Ratio* digunakan untuk mereduksi dimensi data dengan memilih fitur-fitur paling relevan dari lalu lintas jaringan OpenFlow, sehingga dapat mempercepat waktu pemrosesan dan meminimalkan tingkat kesalahan deteksi. Hasil pengujian menunjukkan bahwa kombinasi *Random Forest* dan *Gain Ratio* berhasil mencapai akurasi deteksi sebesar 99,84% pada lingkungan *testbed* yang terkontrol. Penelitian ini menunjukkan bahwa kombinasi *Random Forest* dan *Gain Ratio* mampu memberikan nilai akurasi yang lebih tinggi dibandingkan dengan penggunaan algoritma standar. Penggunaan seleksi fitur *Gain Ratio* terbukti mengoptimalkan akurasi dan efisiensi algoritma *Random Forest* dalam deteksi DDoS.

Kata Kunci: DDoS, SDN, *Random Forest, Gain Ratio*, Keamanan Jaringan.

Abstract

The development of Software-Defined Networking technology offers high flexibility in network management but also introduces security vulnerabilities to Distributed Denial of Service attacks. These attacks can paralyze the SDN controller through packet flooding, leading to a drastic decline in network performance. This research aims to enhance the accuracy of DDoS attack detection in an SDN architecture based on the Ryu Controller with the OpenFlow v1.3 protocol by implementing the Random Forest algorithm optimized with Gain Ratio feature selection. The novelty of this study lies in the use of a primary dataset collected through a self-developed SDN testbed using Mininet with a Modified Tree topology to generate more realistic attack scenarios. The Gain Ratio method is employed to reduce data dimensionality by selecting the most relevant features from OpenFlow network traffic, thereby accelerating processing time and minimizing detection error rates. The experimental results demonstrate that the combination of Random Forest and Gain Ratio achieved a detection accuracy of 99.84% in a controlled testbed environment. This study indicates that the integration of Random Forest and Gain Ratio provides higher accuracy compared to the standard algorithm. The use of Gain Ratio feature selection is proven to optimize both the accuracy and efficiency of the Random Forest algorithm in DDoS detection.

Keywords: DDoS*, SDN, Random Forest, Gain Ratio, Network Security.*

PENDAHULUAN

Perkembangan infrastruktur jaringan komputer saat ini telah bergeser menuju paradigma baru yang lebih dinamis dan fleksibel, yaitu *Software-Defined Networking* (SDN). Berbeda dengan jaringan tradisional yang menggabungkan fungsi kontrol dan pengiriman data dalam satu perangkat keras, SDN memisahkan antara *control plane* dan *data plane*. Pemisahan ini memungkinkan administrator jaringan untuk mengelola lalu lintas data secara terpusat melalui *controller* berbasis perangkat lunak, yang memberikan efisiensi tinggi dalam konfigurasi dan manajemen jaringan

skala besar. *Ryu Controller* merupakan salah satu *controller* berbasis *Python* yang populer karena fleksibilitasnya dalam mengelola *flow table* melalui protokol OpenFlow.

Software-Defined Networking (SDN) merupakan sebuah konsep inovatif dalam merencanakan, mengelola, serta mengimplementasikan jaringan guna menjawab kebutuhan tata kelola jaringan perusahaan yang menuntut fleksibilitas tinggi agar tetap dinamis dan adaptif (Ekawijana et al., 2024). Organisasi dapat melakukan konsolidasi pengaturan jaringan secara terpadu melalui penyediaan kontrol terpusat. *Software-Defined Networking* (SDN) memberikan harapan baru untuk merubah keterbatasan dari infrastruktur jaringan yang ada (Harto & Basuki, 2021).

Konsentrasi fungsi kontrol pada *controller* menjadikannya target utama bagi berbagai serangan siber, di antaranya adalah serangan *Distributed Denial of Service* (DDoS). Serangan ini bertujuan untuk melumpuhkan layanan jaringan dengan membanjiri server target menggunakan lalu lintas yang sangat besar, berdasarkan laporan *Cisco Annual Internet Report*, frekuensi serangan DoS/DDoS telah meningkat lebih dari 2,5 kali lipat dalam tiga tahun terakhir, dengan ukuran rata-rata serangan mendekati 1 Gbps (Sari et al., 2025). Selain itu serangan DDoS dapat meliputi penyerangan terhadap situs web bisnis, layanan cloud, lembaga pemerintah serta infrastruktur kritis untuk menyebabkan gangguan pada layanan online (Mamuriyah et al., 2024), dalam arsitektur SDN DDoS dapat menyasar ke berbagai komponen utama seperti *control panel, data plane,* maupun *bandwidth* (Sugeng & Adhliana, 2026). Pendeteksian serangan DDoS pada lingkungan SDN telah banyak dilakukan, namun tantangan utamanya terletak pada kecepatan dan akurasi deteksi di tengah lalu lintas data yang sangat masif.

Sebagian besar penelitian sebelumnya masih mengandalkan *dataset* publik yang bersifat statis, sehingga kurang merepresentasikan dinamika trafik pada protokol OpenFlow versi terbaru. Pembaharuan dalam penelitian ini terletak pada penggunaan *dataset* primer yang dikumpulkan secara mandiri melalui simulasi *testbed* SDN, dengan menggunakan mininet serta topologi *Modified Tree,* implementasi dilakukan pada lingkungan OpenFlow v1.3 untuk menghasilkan skenario serangan yang lebih aktual dan spesifik pada *Ryu Controller.*

Random Forest adalah pengembangan dari metode *Decision Tree* yang menggunakan beberapa *Decision Tree*, dimana setiap *Decision Tree* telah dilakukan pelatihan menggunakan sampel individu dan setiap atribut dipecah pada pohon yang dipilih antara atribut subset yang bersifat acak (Supriyadi et al., 2020). Penggunaan seluruh fitur dalam dataset seringkali menyebabkan beban komputasi yang berat dan memperlambat waktu deteksi, untuk mengatasi hal tersebut teknik seleksi fitur *Gain Ratio* diintegrasikan guna menyaring fitur-fitur yang paling relevan, berbeda dengan *Information Gain* yang menggunakan pendekatan penyaringan dengan menghitung entropi setiap fitur (*filtered-based)* (Kurniabudi et al., 2020)*, Gain Ratio* memberikan penilaian yang lebih akurat dengan menormalisasi nilai perolehan informasi menggunakan s*plit entropy*. Integrasi metode memungkinkan model untuk melakukan seleksi fitur secara lebih presisi terhadap parameter lalu lintas jaringan yang kompleks (Lutfia et al., 2024). Sistem mampu meningkatkan akurasi deteksi serangan DDoS pada jaringan SDN sekaligus mereduksi waktu pemrosesan secara signifikan. Hasil penelitian ini diharapkan dapat memberikan kontribusi secara teoritis maupun praktis dalam bidang keamanan jaringan.

Model deteksi yang diusulkan memberikan kontribusi pada pengembangan literatur mengenai sistem keamanan jaringan berbasis *Software-Defined Networking* (SDN), khususnya terkait strategi mitigasi serangan DDoS berbasis OpenFlow v1.3. Hasil penelitian menjadi referensi penting bagi pengembang di masa depan, yang ingin mengembangkan *Intrusion Detection System* yang lebih ringan dan efisien pada infrastruktur jaringan modern. Administrator jaringan dapat mengimplementasikan temuan ini untuk membangun sistem pertahanan yang lebih tangguh terhadap serangan DDoS yang berpotensi melumpuhkan layanan secara tiba-tiba.

PENELITIAN RELEVAN

Penelitian oleh (Kiswanto et al., 2025) dengan judul Pengembangan dan Implementasi Sistem Deteksi Serangan DDoS Berbasis Algoritma *Random Forest.* Hasil penelitian tersebut adalah model yang dikembangkan mampu memberikan performa klasifikasi yang lebih tinggi dalam membedakan antara trafik jaringan normal dan serangan DDoS.

Penelitian oleh (Winanto et al., 2024) dengan judul Peningkatan Performa Deteksi Serangan Menggunakan Metode PCA dan *Random Forest*. Hasil penelitian tersebut adalah penggunaan

metode *Random Forest* dan PCA dapat meningkatkan akurasi deteksi pada jaringan dataset CIC IoT 2023.

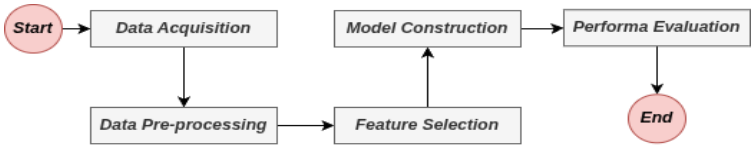
Penelitian oleh (Gudare et al., 2026) dengan judul Penerapan Algoritma *Random Forest* untuk Analisis dan Deteksi Dini Serangan Siber pada Lalu Lintas Jaringan. Hasil penelitian tersebut adalah penggunaan metode *Random Forest* dapat meningkatkan akurasi deteksi pada lalu lintas jaringan.

Penelitian oleh (Sugeng & Adhliana, 2026) dengan judul Deteksi Serangan DDoS Pada Jaringan SDN Menggunakan *Random Forest* dan *Particle Swarm Optimization*. Hasil penelitian tersebut adalah penggunaan metode *Random Forest* yang dioptimasi dengan PSO dapat meningkatkan akurasi deteksi DDoS secara signifikan pada arsitektur SDN.

Penelitian oleh (Kurniabudi et al., 2020) dengan judul Seleksi Fitur dengan *Information Gain* untuk Meningkatkan Deteksi Serangan DDoS Menggunakan *Random Forest*. Hasil penelitian tersebut adalah penggunaan seleksi *Information Gain* mampu meningkatkan performa metode klasifikasi khususnya *Random Forest.*

METODE PENELITIAN

Pendekatan penelitian ini menggabungkan simulasi jaringan pada lingkungan *testbed* SDN dengan teknik *machine learning* yang dioptimasi menggunakan seleksi fitur berbasis *Gain Ratio*, tahapan metodologi tersebut diilustrasikan pada Gambar 1.

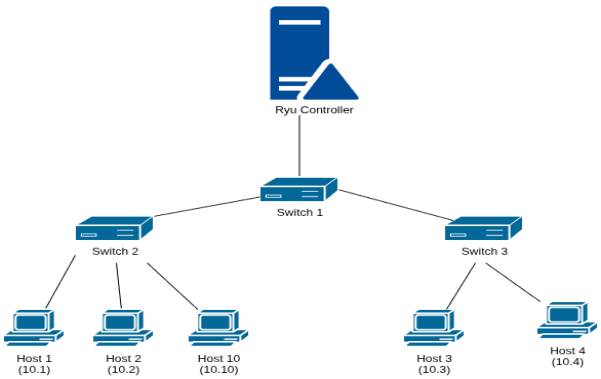


Gambar 1. Tahapan Penelitian

a. Skenario Pengumpulan Data (*Data Acquisition)*

Ryu Controller merupakan *controller* berbasis *open-source* yang dikembangkan menggunakan bahasa pemrograman Python dan mendukung penuh protokol OpenFlow (Pramudita & Suartana, 2020)**.**

Proses pengumpulan data dilakukan melalui simulasi trafik secara *real-time* pada *testbed* untuk menghasilkan *dataset internal* yang aktual dengan menggunakan simulator Mininet untuk infrastruktur jaringan dan *Ryu Controller* yang berfungsi sebagai *control plane* serta Open vSwitch. Trafik normal disimulasikan melalui aktivitas protokol HTTP *(wget/curl)* dan *iperf*, sedangkan trafik serangan diinjeksikan menggunakan *hping3* untuk meluncurkan *ICMP Flood* dan *UDP Flood* intensitas tinggi. *Ryu Controller* mengekstraksi data statistik aliran tersebut setiap 10 detik ke dalam format CSV sebagai basis data analisis, sebagaimana ditunjukkan pada Gambar 2 dan Tabel 1.



Gambar 2. *Modified Tree Topology* yang Diimplementasikan

Tabel 1. Daftar Fitur Statistik Aliran (*Flow Stats)* Awal

Nama Fitur	Deskripsi	Satuan
Duration_sec	Durasi waktu aliran data aktif dalam *table flow*	Detik
Packet_count	Jumlah total paket dalam satu aliran (*flow)*	Paket
byte_count	Jumlah total byte dalam satu aliran	Byte
packet_rate	Kecepatan pengiriman paket per satuan waktu	Pkt/Detik
byte_rate	Kecepatan pengiriman byte per satuan waktu	Byte/Detik
flow_iat_mean	Rata-rata jeda waktu antar paket (Inter-Arrival Time)	Detik
flow_iat_max	Jeda waktu maksimal antar paket dalam satu aliran	Detik
flow_iat_std	Standar deviasi dari jeda waktu antar paket	Detik

b. Pra-pemrosesan Data (*Data Pre-processing)*

Data mentah yang telah dikumpulkan kemudian diproses melalui beberapa tahapan pembersihan untuk menjamin kualitas input model. Tahap pertama adalah *labeling*, di mana setiap baris data diberi penanda numerik berupa nilai 0 untuk merepresentasikan trafik normal dan nilai 1 untuk trafik serangan DDoS. Proses pelabelan telah selesai setelah itu dilakukan proses data *cleaning* yang bertujuan untuk mengeliminasi fitur-fitur bersifat statis seperti identitas *flow*, alamat MAC, atau alamat IP yang tidak memiliki nilai kontributif dalam proses pembelajaran mesin.

c.* Seleksi Fitur Berbasis *Gain Ratio* (*Feature Selection)

Tahap seleksi fitur merupakan langkah krusial dalam mereduksi dimensi data guna meningkatkan efisiensi komputasi dan akurasi model klasifikasi. Metode *Gain Ratio* diterapkan untuk mengevaluasi dan memberikan bobot kontribusi pada setiap fitur dalam dataset. Penerapan *Gain Ratio* bertujuan untuk menyediakan penilaian yang lebih adil terhadap kepentingan fitur-fitur dengan variasi nilai yang beragam dalam pemrosesan data (Joses et al., 2024). *Gain Ratio* memberikan bobot pada setiap fitur jaringan. Fitur dengan nilai *Split Entropy* tinggi akan dikoreksi agar tidak mendominasi, sehingga fitur yang benar-benar informatif (seperti flow_iat_mean dan duration_sec) terpilih. Rumus yang digunakan dan keterangan pada Tabel 2 adalah :

$$\boldsymbol{GainRatio(A)} = \frac{\boldsymbol{Gain\,(S,A)}}{\boldsymbol{SplitInfo_A\,(S)}} \quad (1)$$

Tabel 2. Keterangan Variabel pada Rumus Gain Ratio

Simbol	Keterangan
S	Himpunan data (dataset internal SDN).
A	Atribut/fitur yang diuji (contoh: byte_count)
Pi	Probabilitas munculnya label (0 atau 1)
SplitInfo	Nilai penyeimbang agar fitur dengan banyak variasi tidak menang secara tidak adil

d. Implementasi Algoritma *Random Forest* (*Model Construction)*

Algoritma *machine learning* yang sering digunakan untuk klasifikasi adalah *Random Forest* karena algoritma ini dikenal mampu menangani data berdimensi tinggi dan mengurangi risiko *overfitting (*Gudare et al., 2026). Implementasi algoritma *Random Forest* yang membangun sekumpulan pohon keputusan (*ensemble of decision trees*) menggunakan metode yang dikembangkan oleh Breiman (2001) dengan membangun sejumlah *decision tree* dari subset data latih yang dipilih secara acak (Ikhwanul Uzlah et al., 2024).

Salah satu tahapan yang berpengaruh pada hasil akhir pada pengembangan machine learning adalah parameter tuning (Kiswanto et al., 2025), untuk memastikan eksperimen dapat diproduksi, model diimplementasikan menggunakan pustaka *Scikit-Learn* pada *Python* 3.12.3 dengan parameter teknis sebagaimana ditunjukkan Tabel 3.

Tabel **3.** Parameter Model *Random Forest*

Parameter	Nilai
Jumlah Tree (n_estimators)	100
Max Depth	*Default*

Random State	42
Criterion	*Gini Impurity*
Software	Python (Scikit-Learn)

Algoritma seperti *Random Forest* menunjukkan performa yang baik dalam melakukan klasifikasi antara trafik normal dan trafik serangan, Namun demikian, efektivitas setiap algoritma sangat dipengaruhi oleh karakteristik dataset serta proses *pre-processing data*, termasuk tahap seleksi fitur dan standardisasi (Fathurrahman & Prabowo, 2025). Pelatihan model dilakukan dalam dua skenario utama, yaitu skenario *baseline* (menggunakan seluruh fitur) dan skenario integrasi *Gain Ratio* (menggunakan fitur hasil seleksi). Model dilatih menggunakan pembagian data *Training* 80% dan *Testing* 20%.

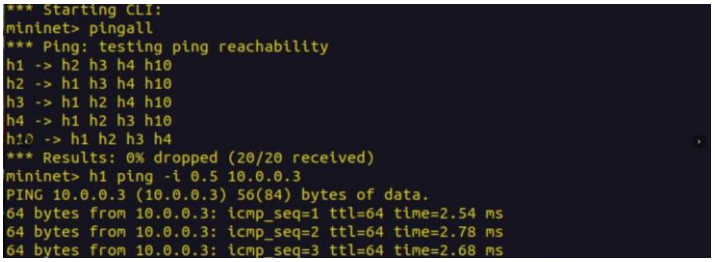
e. Evaluasi Performa *(*Performa *Evaluation)*

Tahap terakhir adalah mengukur efektivitas model *Random Forest* yang diusulkan, dilakukan evaluasi menggunakan metrik *Accuracy, Precision, Recall, dan Confusion Matrix*. Penggunaan *Confusion Matrix* memberikan gambaran detail mengenai klasifikasi trafik normal dan serangan (DDoS). Evaluasi ini bertujuan untuk membuktikan bahwa integrasi *Gain Ratio* mampu melakukan seleksi fitur secara optimal sehingga model dapat mempertahankan akurasi deteksi yang tinggi. Penggunaan fitur yang lebih ringkas diharapkan dapat meringankan beban komputasi pengontrol SDN, yang dibuktikan melalui waktu pemrosesan data yang lebih efisien.

HASIL DAN PEMBAHASAN

Skenario Pengambilan *Dataset* (Trafik Normal)

Proses pengumpulan data dilakukan melalui simulasi trafik pada *testbed* SDN menggunakan Mininet. Skenario ini dirancang untuk menghasilkan data yang merepresentasikan dua kondisi jaringan, yaitu kondisi normal dan kondisi di bawah serangan DDoS. Trafik normal disimulasikan untuk membangun profil perilaku jaringan yang valid. Proses pengujian awal menggunakan perintah pingall setelah itu untuk trafik normal dibangkitkan dengan mengirimkan paket ICMP dari h1 ke h3 secara berkala menggunakan ping –i 0.5.

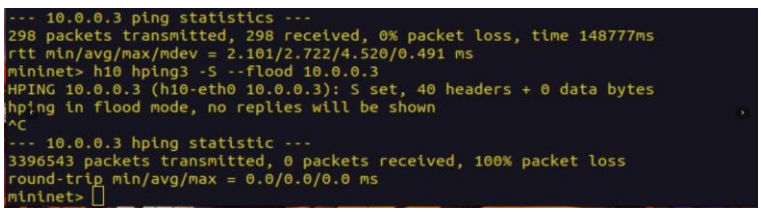


Gambar 3. Pengumpulan data menggunakan trafik normal

Proses pengumpulan data dilakukan melalui simulasi trafik pada *testbed* SDN menggunakan Mininet (Gambar 3). Skenario ini dirancang untuk menghasilkan data yang merepresentasikan dua kondisi jaringan, yaitu kondisi normal dan kondisi di bawah serangan DDoS. Trafik normal disimulasikan untuk membangun profil perilaku jaringan yang valid.

Skenario Serangan DDoS (SYN *Flood*)

Serangan DDoS disimulasikan menggunakan perangkat lunak hping3 yang ditargetkan ke alamat IP 10.0.0.3 (h3). Metode yang di lakukan dengan menggunakan *mode flood* dan pengiriman paket SYN dari h10.



Gambar 4. Pengumpulan data menggunakan serangan DDoS

Trafik normal tercatat sebanyak 298 paket terkirim berdasarkan hasil pengujian pada Gambar 4. Rata-rata *Round-Trip Time* (RTT) yang dihasilkan adalah sebesar 2.722 ms. Data ini merepresentasikan kondisi jaringan yang stabil sedangkan untuk serangan yang menggunakan hping3 mengirimkan sebanyak 3.396.543 paket dengan tingkat *packet loss* mencapai 100% pada sisi target. Proses sebelumnya menunjukkan bahwa *flooding* berhasil membanjiri antrean paket pada *switch* dan melumpuhkan layanan pada target, yang merupakan karakteristik utama serangan DDoS.

Seluruh trafik yang melewati *Open vSwitch* (OVS) diekstraksi *oleh Ryu Controller* menjadi fitur-fitur statistik aliran (*flow statistics*) dalam format CSV. Hasil verifikasi pada file verification.py, *dataset* yang berhasil dikumpulkan memiliki karakteristik sebagai berikut (Gambar 5):

Gambar 5. Distribusi Dataset

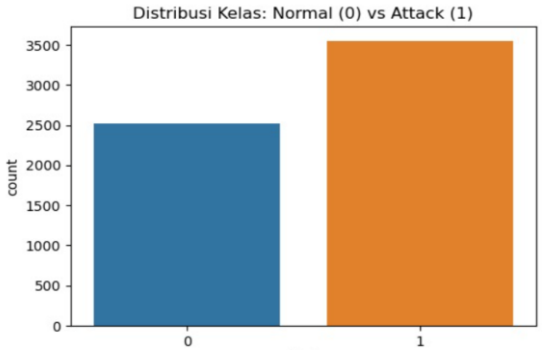
Analisis Eksplorasi Data (*Exploratory Data Analysis* - EDA)

Ekstraksi fitur adalah proses yang berfungsi dalam mengidentifikasi fitur-fitur penting dari data (Winanto et al., 2024). Tahapan EDA dilakukan setelah *dataset* berhasil diekstraksi ke dalam format CSV, dilakukan tahapan EDA untuk memahami struktur, sebaran, dan hubungan antar variabel di dalam *dataset* sebelum masuk ke proses klasifikasi. Karakteristik statistik dari *dataset* merangkum hasil ekstraksi sebanyak 6.075 records.

Tabel 4. Ringkasan Statistik Deskriptif *Dataset*

Statistik	Packet_count	Byte_count	Flow_iat_max	Label
Count	6075.0	6075.0	6075.0	6075.0
Mean	33.40	2137.65	0.259	0.585
Std Dev	19.71	1261.46	6.755	0.492
Min	10.0	640.0	0.0003	0.0
Max	50.0	3200.0	219.596	1.0

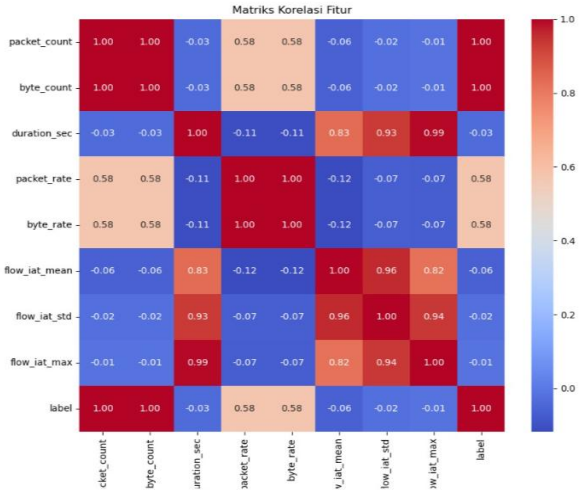
Nilai Standard Deviation (Std Dev) pada fitur byte_count tercatat cukup tinggi sebesar 1261.46 (Tabel 4), nilai *Standard Deviation* (Std Dev) yang tinggi pada byte_count (1261.46) menunjukkan variasi data yang sangat kontras, yang mencerminkan perbedaan antara volume trafik normal dan serangan.



Gambar 6. Grafik *Count Plot*

Visualisasi distribusi kelas menunjukkan kondisi dataset yang seimbang (Gambar 6). Dataset tersebut terdiri dari 6.075 *records* ini berada dalam kondisi seimbang (*well-balanced*), dengan rincian 2.521 sampel trafik normal (Label 0) dan 3.554 sampel trafik serangan DDoS (Label 1). Perbandingan proporsi sebesar 41,5% : 58,5%, grafik ini membuktikan secara visual bahwa tidak

terdapat dominasi kelas yang ekstrem, sehingga algoritma *Random Forest* dapat mempelajari karakteristik kedua jenis trafik secara objektif tanpa risiko *majority class* bias.



Gambar 7. Matriks Korelasi

Matriks korelasi merupakan instrumen krusial dalam tahapan seleksi fitur (Gambar 7). Visualisasi ini menunjukkan kekuatan hubungan linier antar variabel terhadap label target. Analisis pada baris label menunjukkan bahwa fitur packet_count dan byte_count memiliki skor korelasi sempurna (1.00), yang mengindikasikan bahwa kedua fitur ini memiliki pengaruh paling dominan namun juga menunjukkan adanya redundansi data yang tinggi. Fitur packet_rate dan byte_rate memberikan dimensi informasi tambahan dengan skor korelasi 0.58, mencerminkan karakteristik frekuensi trafik yang berbeda dari sekadar volume paket. Identifikasi fitur dengan korelasi tinggi ini memberikan justifikasi kuat bagi penggunaan algoritma *Gain Ratio* untuk mengeliminasi variabel yang redundan, sehingga model *Random Forest* dapat bekerja lebih efisien dengan hanya menggunakan fitur-fitur yang paling informatif tanpa membebani kinerja komputasi *controller*.

Seleksi Fitur Menggunakan *Gain Ratio*

Optimasi *dataset* guna memilih fitur yang paling informatif dalam membedakan trafik normal dan serangan DDoS. Perhitungan *Information Gain* yang dinormalisasi dengan *Split Entropy*, dihasilkan peringkat fitur sebagai berikut:

```
================================================
      LAPORAN PERINGKAT FITUR (GAIN RATIO)
================================================
         Fitur  Information_Gain  Split_Entropy  Gain_Ratio
 duration_sec          0.503863       0.022568   22.326587
flow_iat_mean          0.471232       0.022195   21.231400
 flow_iat_std          0.408236       0.023556   17.330828
 flow_iat_max          0.229289       0.022568   10.159990
   byte_count          0.679443       0.979041    0.693988
 packet_count          0.679360       0.979041    0.693904
    byte_rate          0.442740       2.120750    0.208766
  packet_rate          0.442677       2.120750    0.208736
================================================
[!] Berhasil menyimpan hasil ke 'hasil_gain_ratio.csv'
```

Gambar 8. Laporan Peringkat Fitur *Gain Ratio*

Fitur berbasis waktu seperti duration_sec dan flow_iat_mean memiliki nilai *Gain Ratio* tertinggi (Gambar 8). Meskipun byte_count dan packet_count memiliki *Information Gain* yang besar (0.67), namun nilai *Split Entropy*-nya juga tinggi, sehingga skor *Gain Ratio*-nya menjadi lebih kecil dibandingkan fitur berbasis durasi. Pola durasi aliran (*flow*) jauh lebih konsisten untuk mengidentifikasi serangan DDoS pada *testbed* SDN ini.

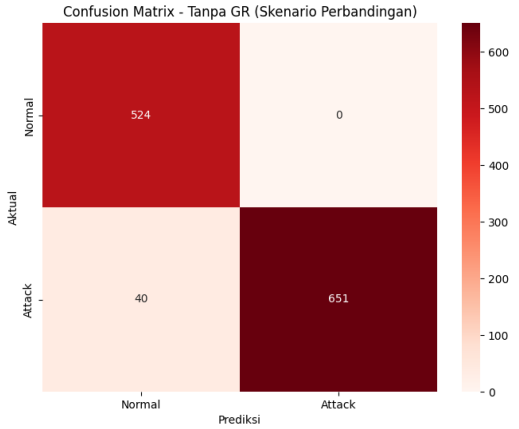
Evaluasi Model Random Forest

Pengujian menggunakan algoritma *Random Forest* dengan pembagian data 80% training dan 20% testing dilakukan setelah proses seleksi. Proses evaluasi diawali dengan melihat performa model standar menggunakan seluruh fitur tanpa seleksi sebagai pembanding.

```
==========================================
   HASIL EVALUASI MODEL (TANPA GAIN RATIO)
==========================================
Akurasi Model    : 96.71%
Waktu Training   : 0.0111 detik
Jumlah Data Test : 1215 sampel
------------------------------------------
Classification Report:
              precision    recall  f1-score   support

           0       0.93      1.00      0.96       524
           1       1.00      0.94      0.97       691

    accuracy                           0.97      1215
   macro avg       0.96      0.97      0.97      1215
weighted avg       0.97      0.97      0.97      1215

==========================================
```

Gambar 9. Hasil Evaluasi Tanpa *Gain Ratio*

Pengujian awal menggunakan algoritma *Random Forest* tanpa *Gain Ratio* Gambar 9, model menghasilkan akurasi sebesar 96,71% dengan memanfaatkan seluruh fitur yang tersedia dalam *dataset*. Angka akurasi yang ditunjukkan tersebut meskipun tergolong tinggi, model masih menunjukkan keterbatasan dalam mengenali pola serangan secara presisi, yang dibuktikan dengan nilai recall sebesar 0,94 pada kelas serangan. Indikasi adanya margin kesalahan deteksi (*False Negative*) yang cukup signifikan, di mana sistem masih gagal mengidentifikasi sebagian trafik serangan *DDoS* sebagai ancaman. Penggunaan seluruh fitur tanpa seleksi mengakibatkan model bekerja dengan beban komputasi yang lebih besar.



Gambar 10. *Confusion Matrix Detection DDoS* SDN Tanpa *Gain Ratio*

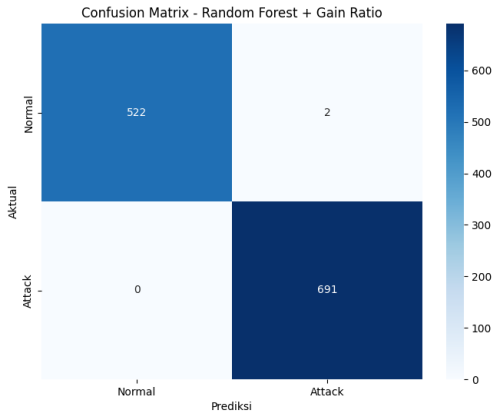
Perbandingan kedua *Confusion Matrix* tersebut pada Gambar 10, menunjukkan bahwa model *Random Forest* tanpa *Gain Ratio* memiliki akurasi 96,71% dengan kelemahan fatal berupa 40 sampel serangan yang gagal terdeteksi (*False Negative*).

Pengujian model dilakukan setelah proses seleksi fitur menggunakan *Gain Ratio* selesai dilakukan, tahap berikutnya adalah pengujian model menggunakan algoritma *Random Forest* dengan menerapkan skema pembagian data (*data splitting*) sebesar 80% untuk data pelatihan (*training set*) dan 20% untuk data pengujian (*testing set*). Langkah validasi efektivitas fitur-fitur terpilih dalam meningkatkan akurasi deteksi sekaligus memastikan model memiliki kemampuan generalisasi yang baik terhadap data trafik jaringan yang belum pernah dipelajari sebelumnya.

```
======================================
      HASIL EVALUASI MODEL RF
======================================
Akurasi Model    : 99.84%
Waktu Training   : 0.4445 detik
Jumlah Data Test : 1215 sampel
--------------------------------------
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       524
           1       1.00      1.00      1.00       691

    accuracy                           1.00      1215
   macro avg       1.00      1.00      1.00      1215
weighted avg       1.00      1.00      1.00      1215

======================================
```

Gambar 11. Hasil Evaluasi Model

Akurasi model mencapai nilai 99,84% berdasarkan hasil pengujian algoritma *Random Forest* dan *Gain Ratio* (Gambar 11), klasifikasi data berhasil dilakukan dengan benar pada 1.213 data dari total 1.215 data uji, sementara hanya terdapat 2 data (0,16%) yang mengalami kesalahan deteksi. Nilai *Precision dan Recall* pada kelas 1 (*Attack)* tetap menunjukkan angka sempurna yaitu 1.00, yang berarti sistem memiliki tingkat keberhasilan 100% dalam mendeteksi seluruh serangan DDoS yang masuk tanpa ada satupun serangan yang lolos (*False Negative)*.



Gambar 12. *Confusion Matrix Detection DDoS* SDN

Visualisasi *Confusion Matrix* pada Gambar 12, Y merupakan metrik evaluasi krusial yang menyajikan perbandingan antara nilai aktual kondisi jaringan dengan nilai prediksi model *Random Forest* berbasis *Gain Ratio*, di mana dari total 1.215 data uji, model berhasil mencapai tingkat akurasi yang sangat tinggi sebesar 99,84%. Matriks ini menunjukkan bahwa sebanyak 522 sampel trafik normal berhasil diklasifikasikan secara tepat sebagai *True Negative* (TN) dan 691 sampel serangan DDoS berhasil diidentifikasi sebagai *True Positive* (TP), dengan margin kesalahan yang sangat minim berupa 2 sampel *False Positive* (FP) dan tanpa adanya kegagalan deteksi atau *False Negative (*FN). Hasil membuktikan secara empiris bahwa fitur-fitur yang telah diseleksi melalui metode *Gain Ratio* memiliki daya pembeda yang sangat signifikan, sehingga model mampu mengenali pola serangan SYN *Flood* dengan akurasi mutlak dan menjamin keandalan sistem deteksi dalam lingkungan jaringan SDN yang diuji.

Tabel 5. Perbandingan Performa Model *Random Forest*

Matriks Evaluasi	**Tanpa *Gain Ratio* (*Baseline)***	***Gain Ratio (Proposed)***
Jumlah Fitur	8 Fitur	4 Fitur Utama
Akurasi (Accuracy)	96,71%	99,84%
Presisi (*Precision*)	1.00	1.00
*Recall (*Serangan DDoS)	0.94	1.00
*False Negative (*FN)	40 Sampel	0 Sampel
*False Positive (*FP)	0 Sampel	2 Sampel

Integrasi *Gain Ratio* dan *Random Forest* pada jaringan SDN berhasil meningkatkan akurasi deteksi dari 96,71% menjadi 99,84% dengan nilai *Recall* sempurna (1.00). Seleksi fitur berhasil mereduksi data dari 8 menjadi 4 fitur utama, performa deteksi pada lingkungan *testbed* tetap optimal dan *reliabel* tanpa indikasi *overfitting*.

SIMPULAN

Penelitian ini berhasil membuktikan bahwa integrasi seleksi fitur *Gain Ratio* pada algoritma *Random Forest* dengan lingkungan *testbed* yang terkontrol mampu meningkatkan efisiensi dan akurasi deteksi DDoS pada jaringan SDN. Hasil pengujian menunjukkan peningkatan akurasi dari 96,71% menjadi 99,84% dengan nilai *Recall* mencapai 1.00, yang berarti sistem berhasil mendeteksi seluruh serangan tanpa ada yang lolos *(zero False Negative)*. Capaian akurasi yang sangat tinggi ini diperoleh melalui penggunaan *dataset* primer pada lingkungan *testbed* yang terkontrol, di mana skenario serangan *SYN Flood* memiliki pola yang sangat diskriminatif terhadap

trafik normal, sehingga hasil tersebut mencerminkan performa optimal model pada infrastruktur spesifik. Sinergi ini tidak hanya mereduksi dimensi data dari 8 menjadi 4 fitur utama guna meringankan beban komputasi *controller*, tetapi juga memberikan solusi mitigasi yang andal dalam menjaga ketersediaan layanan jaringan SDN.

DAFTAR PUSTAKA

Ekawijana, A., Bakhrun, A., & Kurniawan, M. T. (2024). Deteksi Serangan DDOS Pada Jaringan SDN dengan Metode Random Forest. Jurnal media informatika budidarma, 8(1), 685

Harto, M. K., & Basuki, A. (2021). Deteksi Serangan DDos Pada Jaringan Berbasis Sdn Dengan Klasifikasi Random Forest. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 5, 1329–1333.

Sari, L., Faiz, M. N., & Muhammad, A. W. (2025). Perbandingan Pendekatan Machine Learning dalam Deteksi Serangan DDoS Jaringan Komputer. Infotekmesin, 16(1), 153–159.

Mamuriyah, N., Prasetyo, S. E., & Sijabat, A. O. (2024). Rancangan Sistem Keamanan Jaringan dari serangan DDoS Menggunakan Metode Pengujian Penetrasi. Jurnal Teknologi Dan Sistem Informasi Bisnis, 6(1), 162–167.

Sugeng, W., & Adhliana, F. (2026). Deteksi serangan DDoS pada jaringan sdn menggunakan random forest dan particle swarm optimization. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 13, 39–46.

Supriyadi, R., Gata, W., Maulidah, N., & Fauzi, A. (2020). Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah. E-Bisnis : Jurnal Ilmiah Ekonomi dan Bisnis, 13(2), 67–75.

Kurniabudi, K., Harris, A., & Rahim, A. (2020). Seleksi Fitur Dengan Information Gain Untuk Meningkatkan Deteksi Serangan DDoS menggunakan Random Forest. Techno.Com, 19(1), 56–66.

Lutfia, A., Gunawan, G., Rohman, R. S., & Gunawan, A. (2024). Penerapan seleksi fitur gain ratio pada prediksi penyakit jantung berbasis naïve bayes. Jurnal Responsif : Riset Sains dan Informatika, 6(1), 1–10.

Kiswanto, D., Ramadhani, F., Surbakti, N. M., & Nasution, N. A. (2025). Pengembangan dan Implementasi Sistem Deteksi Serangan DDoS Berbasis Algoritma Random Forest. 6(3), 247–256.

Winanto, E. A., Novianto, Y., Sharipuddin, S., Wijaya, I. S., & Jusia, P. A. (2024). Peningkatan performa deteksi serangan menggunakan metode pca dan random forest. Jurnal Teknologi Informasi dan Ilmu Komputer, 11(2), 285–290.

Gudare, Y. B., Bria, Y. P., & Tedy, F. (2026). Penerapan Algoritma Random Forest untuk Analisis dan Deteksi Dini Serangan Siber pada Lalu Lintas Jaringan. 3.

Pramudita, A. Z., & Suartana, I. M. (2020). Perbandingan Performa Controller OpenDayLight dan Ryu pada Arsitektur Software Defined Network. Journal of Informatics and Computer Science (JINACS), 1(04), 174–178.

Joses, S., Quinevera, S., Mardianto, R., Yulvida, D., & Shiddiqi, A. M. (2024). Pendekatan Metode Ensemble Learning untuk Deteksi Serangan DDoS menggunakan Soft Voting Classifier. Jurnal Edukasi dan Penelitian Informatika (JEPIN), 10(1), 79.

Ikhwanul Uzlah, L., Adi Saputra, R., & Isnawaty, I. (2024). Deteksi serangan siber pada jaringan komputer menggunakan metode random forest. JATI (Jurnal Mahasiswa Teknik Informatika), 8(3), 2787–2793.

Fathurrahman, M. A., & Prabowo, D. W. (2025). Perbandingan Performa Algoritma Random Forest dan SVM Dalam Mendeteksi serangan DDoS di Jaringan Cloud. Jurnal Riset Rekayasa Elektro, 7(2), 193–202.