Implementasi dan Analisis Mitigasi Serangan *Distributed Denial of Service SYN Flood* pada *Software Defined Network* Menggunakan *Rate Limiting*

1st Dr. Mochamad Teguh Kurniawan, S.T., M.T.
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
teguhkurniawan@telkomuniversity.ac.id

2nd Muhammad Fathinuddin, S.SI., M.T.
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
muhammadfathinuddin@telkomuniversity.ac.id

3rd Evandani Giantino Rafif
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
evandanirafif@student.telkomuniversity.ac.id

Abstrak*— Software Defined Network (SDN) memisahkan control plane dan data plane, menjadikan kontroler sebagai pusat pengendali sekaligus titik rawan serangan. Serangan DDoS jenis SYN Flood dapat membanjiri kontroler dengan koneksi palsu, menguras sumber daya, dan menghentikan pemrosesan paket normal. Penelitian ini mengembangkan sistem deteksi dan mitigasi berbasis Support Vector Machine (SVM) dan rate limiting dengan algoritma token bucket. Model SVM dilatih menggunakan dataset publik dengan skema train-validation-test dan optimasi hyperparameter menggunakan Optuna, menghasilkan akurasi 96,64%. Sistem diuji pada 23 skenario serangan, masing-masing 4 kali, mencakup IP statis, acak, dan kombinasi. Rata-rata false negative tercatat hanya 1,2–1,5 paket per trial, dengan mitigasi serangan mencapai 98–99%. Paket normal berhasil dikirim tanpa kehilangan pada 83 dari 92 trial, dan sisanya hanya mencatat packet loss sebesar 0,83%. Sistem dilengkapi Prometheus, Grafana, dan notifikasi Telegram, memungkinkan pemantauan kontroler secara pasif. Hasil menunjukkan sistem mampu menjaga layanan SDN secara otomatis, efisien, dan tangguh saat menghadapi serangan SYN Flood..

***Kata kunci*— Software Defined Network, SYN Flood, Rate Limiting, Support Vector Machine**

I. PENDAHULUAN

Software Defined Network (SDN) merupakan pendekatan arsitektur jaringan yang merevolusi pengelolaan lalu lintas dengan memisahkan logika kontrol dari perangkat keras, sehingga memberikan fleksibilitas dan kendali terpusat. Namun, sentralisasi kontroler ini menjadikannya titik kegagalan tunggal (single point of failure) yang rentan terhadap serangan Distributed Denial of Service (DDoS), khususnya jenis SYN Flood. Serangan ini mengeksploitasi proses three-way handshake TCP dengan membanjiri kontroler menggunakan permintaan koneksi palsu, yang menyebabkan lonjakan pesan Packet_In dan konsumsi sumber daya CPU, memori, serta bandwidth secara masif. Urgensi penanganan ancaman ini diperkuat oleh data dari Badan Siber dan Sandi Negara (BSSN) yang menunjukkan ratusan juta anomali lalu lintas siber di Indonesia setiap tahunnya. Untuk mengatasi permasalahan tersebut, penelitian ini mengimplementasikan dan menganalisis sebuah sistem pertahanan hibrida. Sistem ini mengintegrasikan algoritma Support Vector Machine (SVM) untuk melakukan deteksi cerdas terhadap pola serangan SYN Flood dan metode rate limiting sebagai strategi mitigasi untuk membatasi laju paket dari sumber berbahaya tanpa mengorbankan lalu lintas yang sah. Seluruh mekanisme ini diimplementasikan pada Ryu Controller dan diuji dalam lingkungan simulasi Mininet. Penelitian ini bertujuan untuk menganalisis efektivitas kombinasi SVM dan rate limiting dalam menjaga ketersediaan layanan, serta mengevaluasi dampaknya terhadap kualitas layanan (Quality of Service - QoS) jaringan, dengan dukungan sistem pemantauan modern menggunakan Prometheus dan Grafana.

II. KAJIAN TEORI

A. Software Defined Network

Software Defined Network (SDN) merupakan paradigma baru dalam arsitektur jaringan yang memisahkan fungsi kontrol dan pengolahan data melalui tiga lapisan terintegrasi: Infrastructure Plane, Control Plane, dan Application Plane. Pendekatan ini memungkinkan manajemen jaringan yang lebih fleksibel dan responsif terhadap ancaman keamanan secara real-time. Peran switch dalam SDN, dikendalikan melalui protokol seperti OpenFlow dan NETCONF, memungkinkan forwarding data yang dinamis dan adaptif dengan dukungan teknologi TCAM. Meskipun menghadirkan efisiensi tinggi dalam pengelolaan jaringan, implementasi SDN masih menghadapi tantangan teknis, seperti keterbatasan skala tabel forwarding dan potensi kerentanan keamanan. Namun, fleksibilitas dan kontrol terpusat yang ditawarkan menjadikan SDN sebagai solusi prospektif untuk infrastruktur jaringan modern[1].

B. Ryu Controller

Ryu Controller merupakan pengendali berbasis Python yang mendukung berbagai protokol, termasuk OpenFlow, dan dirancang untuk mempermudah pengembangan aplikasi pengelolaan jaringan dalam arsitektur SDN. Sebagai elemen inti yang memisahkan fungsi kontrol dari pengalihan data,

Ryu menyediakan fleksibilitas tinggi dalam manajemen aliran lalu lintas. Berdasarkan evaluasi kinerja, Ryu menunjukkan performa yang unggul dibandingkan pengendali lain seperti POX, terutama dalam parameter delay, jitter, bitrate, dan throughput pada topologi linier dengan lima switch, menjadikannya pilihan yang efisien untuk jaringan SDN modern[2].

C. Mininet

Mininet merupakan emulator jaringan berbasis perangkat lunak yang dirancang untuk mendukung penelitian dan pembelajaran dalam arsitektur SDN. Emulator ini memungkinkan pembuatan jaringan virtual realistis dalam satu perangkat menggunakan teknologi network namespaces, sehingga efisien dalam sumber daya. Melalui antarmuka Command Line Interface (CLI), pengguna dapat membangun berbagai topologi jaringan, baik standar maupun kustom, secara cepat dan fleksibel. Mininet juga mendukung integrasi dengan alat seperti iperf3 dan ping untuk menghasilkan lalu lintas dan mengukur parameter performa seperti throughput, RTT, jitter, dan packet loss. Dengan kapabilitas tersebut, Mininet menjadi solusi uji coba jaringan SDN yang fleksibel, efisien, dan hemat biaya[3].

D. SYN Flood Attack

SYN Flood merupakan jenis serangan DDoS yang mengeksploitasi kelemahan pada mekanisme three-way handshake protokol TCP dengan membanjiri sistem target menggunakan paket SYN dalam jumlah besar. Serangan ini bertujuan menguras memori dan sumber daya pemrosesan host, sehingga sistem tidak dapat merespons koneksi dari pengguna yang sah. Tantangan utama dalam mitigasi serangan ini adalah kompleksitas pemfilteran IP, yang meningkat drastis seiring bertambahnya sumber serangan, sehingga membebani perangkat jaringan. Oleh karena itu, dibutuhkan pendekatan mitigasi yang lebih efisien dan adaptif untuk menangani serangan ini secara efektif[4].

E. Support Vector Machine

Support Vector Machine (SVM) merupakan metode pembelajaran mesin yang digunakan untuk klasifikasi data dengan membangun hyperplane optimal sebagai pemisah antar kelas. Dalam konteks deteksi intrusi jaringan, SVM bertujuan memaksimalkan margin antar kelas untuk meningkatkan akurasi dan presisi klasifikasi. Keunggulan SVM terletak pada kemampuannya menangani data non-linier melalui konsep soft margin dan kernel trick, yang memungkinkan pemetaan data ke dimensi yang lebih tinggi. Pendekatan ini menjadikan SVM sebagai model klasifikasi yang robust dan adaptif terhadap beragam karakteristik data, serta efektif untuk diterapkan dalam sistem deteksi intrusi jaringan[5].

F. Rate Limiting

Rate limiting merupakan teknik keamanan yang bertujuan membatasi jumlah permintaan ke server dalam interval waktu tertentu untuk mencegah gangguan layanan, khususnya akibat serangan DDoS. Mekanisme ini memvalidasi dan menyaring lalu lintas masuk guna mengurangi beban dari trafik tidak sah atau berlebihan. Efektivitasnya terlihat pada skenario serangan skala kecil hingga menengah, di mana sistem dapat mendeteksi dan membatasi permintaan dari alamat IP yang menunjukkan perilaku mencurigakan. Dengan demikian, rate limiting berkontribusi signifikan dalam meningkatkan ketahanan sistem terhadap ancaman siber dan meminimalkan risiko downtime akibat flooding[6].

G. Prometheus

Prometheus merupakan alat monitoring open-source yang digunakan untuk mengumpulkan dan menganalisis metrik dari layanan dan node dalam infrastruktur komputasi. Melalui eksportir khusus, Prometheus dapat mengakses data seperti beban kerja, I/O, dan penggunaan jaringan tanpa memerlukan skema metrik tetap. Didukung oleh PromQL, alat ini memungkinkan eksplorasi data secara fleksibel dan mendalam. Prometheus juga terintegrasi secara optimal dengan platform visualisasi seperti Grafana, serta mampu menyimpan data hingga 15 hari. Kemampuannya menjadikannya pilihan utama dalam pemantauan layanan produksi, termasuk sistem Linux dan klaster Kubernetes[7].

H. Grafana

Grafana adalah perangkat lunak open-source untuk analisis dan visualisasi data secara interaktif lintas platform. Dengan dukungan berbagai sumber data seperti Prometheus, Grafana memungkinkan pembuatan dasbor pemantauan yang kompleks melalui sistem plugin dan pembangun kueri interaktif. Dalam konteks pemantauan perangkat edge heterogen, Grafana menyajikan metrik seperti penggunaan CPU dan memori dalam bentuk grafik real-time, sehingga mempermudah analisis performa sistem secara menyeluruh[8].

I. Hping3

Hping3 adalah sebuah tool yang sering digunakan untuk pengujian dan analisis jaringan, yang memungkinkan pengguna untuk mengirim paket TCP/IP yang dapat membantu mendeteksi berbagai jenis serangan. Dalam berbagai studi yang diacu, penggunaan Hping3 sebagai dataset menunjukkan kemampuan sistem deteksi intrusi untuk mengidentifikasi serangan DDoS dengan menggunakan teknik seleksi fitur[9].

J. VMware Workstation

VMware Workstation adalah perangkat lunak virtualisasi desktop yang memungkinkan pengguna untuk menjalankan beberapa sistem operasi sekaligus pada satu komputer fisik. Perangkat lunak ini menciptakan lingkungan Virtual Machine (VM) yang memungkinkan pengguna untuk menjalankan berbagai sistem operasi seperti Windows, Linux, atau lainnya, tanpa perlu melakukan instalasi langsung pada perangkat keras. VMware Workstation digunakan untuk eksperimen dalam lingkungan forensik digital berbasis cloud. Perangkat lunak ini memungkinkan pengambilan snapshot, yaitu salinan kondisi sistem pada waktu tertentu, yang sangat penting dalam investigasi forensik untuk mendokumentasikan bukti digital tanpa memodifikasi keadaan asli sistem tersebut[10].

K. Ubuntu Linux

Ubuntu Linux merupakan sistem operasi open-source turunan Debian yang dikenal karena kemudahan instalasi, antarmuka yang ramah pengguna, serta dukungan komunitas yang luas. Stabilitas, keamanan, dan kompatibilitas dengan berbagai perangkat lunak menjadikannya pilihan ideal untuk penelitian, simulasi jaringan, pengujian keamanan, dan pengembangan aplikasi. Dengan pembaruan sistem yang konsisten dan pustaka yang lengkap, Ubuntu banyak digunakan di lingkungan akademik maupun industri teknologi[11].

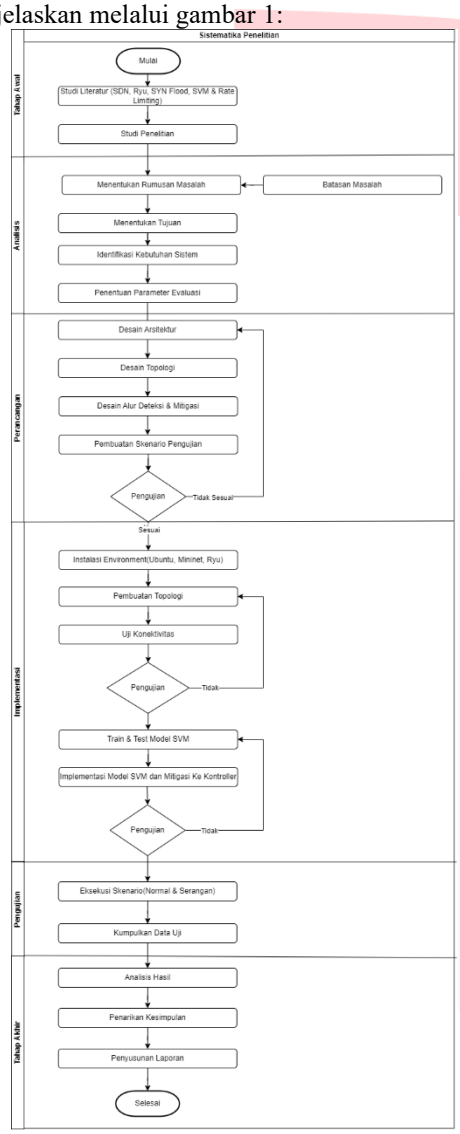
III. METODE

A. Kerangka Berpikir

Kerangka berpikir merupakan landasan konseptual yang disusun berdasarkan sintesis teori, observasi, dan kajian literatur, yang digunakan untuk menjelaskan hubungan antar variabel dalam penelitian. Kerangka ini membantu peneliti dalam merumuskan argumen, menyusun asumsi, serta mengarahkan analisis terhadap permasalahan yang diteliti. Dalam penelitian kuantitatif, kerangka berpikir berfungsi untuk menguji hipotesis, sedangkan dalam penelitian deskriptif, berperan sebagai dasar penjelasan teoritis atas data yang dikumpulkan.

B. Sistematika Penyelesaian Masalah

Berikut adalah Langkah-langkah sistematis penelitian yang dapat dijelaskan melalui gambar 1:



Gambar 1 Sistematika Penyelesaian Masalah

Gambar sistematika penelitian menunjukkan alur kerja yang terdiri dari tahapan studi literatur, analisis masalah, perancangan sistem, implementasi, hingga pengujian dan penyusunan laporan. Setiap tahap saling terintegrasi untuk mencapai tujuan penelitian, yaitu membangun sistem mitigasi serangan SYN Flood berbasis SDN dengan integrasi model SVM dan metode rate limiting. Pendekatan ini dilakukan secara sistematis dan iteratif guna memastikan solusi yang dikembangkan efektif dan sesuai dengan permasalahan yang diangkat.

1. Tahap Awal

Tahap awal diawali dengan studi literatur mengenai konsep dasar SDN, Ryu, serangan SYN Flood, metode klasifikasi SVM, serta mekanisme mitigasi rate limiting. Langkah ini dilakukan untuk membangun landasan teori yang kuat dan memahami penelitian terdahulu yang relevan, sebagai dasar dalam merancang solusi dan metodologi penelitian yang akan dikembangkan.

2. Tahap Analisis

Pada tahap analisis, dilakukan identifikasi permasalahan, penentuan tujuan dan batasan penelitian, serta pengumpulan kebutuhan sistem. Tahap ini dilanjutkan dengan penetapan parameter evaluasi yang akan digunakan untuk mengukur efektivitas sistem. Proses ini bertujuan memperjelas ruang lingkup dan arah penelitian agar tetap fokus dan terarah.

3. Tahap Perancangan

Tahap ini mencakup perancangan arsitektur sistem, topologi jaringan, alur deteksi dan mitigasi, serta skenario pengujian. Setiap rancangan diuji secara konseptual, dan bila ditemukan ketidaksesuaian, maka dilakukan perbaikan hingga diperoleh rancangan yang sesuai untuk diimplementasikan.

4. Tahap Implementasi

Tahap implementasi dimulai dengan instalasi lingkungan pengujian (Ubuntu, Mininet, dan Ryu), dilanjutkan dengan pembuatan topologi dan pengujian konektivitas. Setelah itu dilakukan pelatihan dan pengujian model SVM, lalu integrasi model dan mekanisme mitigasi ke dalam kontroler SDN. Proses ini berlangsung secara iteratif hingga sistem dapat berjalan sesuai skenario yang dirancang.

5. Tahap Pengujian

Pengujian dilakukan dengan mengeksekusi skenario jaringan, baik dalam kondisi normal maupun saat terjadi serangan SYN Flood. Data performa dikumpulkan berdasarkan metrik tertentu, seperti throughput, delay, dan tingkat deteksi, untuk dianalisis lebih lanjut.

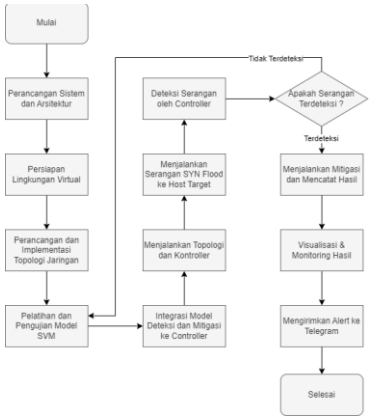
6. Tahap Akhir

Tahap akhir mencakup analisis hasil pengujian, penarikan kesimpulan berdasarkan temuan yang diperoleh, serta penyusunan laporan penelitian secara menyeluruh. Hasil akhir diharapkan dapat menjawab rumusan masalah dan memberikan kontribusi terhadap pengembangan solusi keamanan di jaringan SDN.

IV. PENYELESAIAN MASALAH

A. Alur Perancangan

Alur perancangan sistem ini menunjukkan tahapan utama dalam membangun sistem deteksi dan mitigasi serangan DDoS SYN Flood pada jaringan SDN. Setiap langkah disusun secara logis, mulai dari perancangan hingga implementasi, termasuk proses identifikasi serangan dan pengiriman notifikasi.



Gambar 2 Alur Perancangan

1. Tahap ini menyusun komponen utama sistem seperti kontroler, modul deteksi, mekanisme mitigasi, serta topologi jaringan.
2. Dilakukan instalasi Ubuntu, Mininet, dan Ryu Controller dalam lingkungan virtual sebagai persiapan eksperimen jaringan SDN.
3. Topologi jaringan virtual dibangun menggunakan Mininet dengan komposisi Host penyerang, target, dan switch.
4. Model SVM dilatih menggunakan dataset lalu lintas jaringan untuk membedakan trafik normal dan berbahaya.
5. Model deteksi dan logika mitigasi diintegrasikan ke dalam kontroler menggunakan metode token bucket.
6. Topologi jaringan dan kontroler dijalankan untuk memulai simulasi dan pengujian sistem.
7. Host penyerang mengirimkan paket SYN berlebih ke Host target menggunakan hping3 sebagai simulasi serangan.
8. Kontroler memproses koneksi masuk dan mendeteksi pola serangan berdasarkan klasifikasi SVM.
9. Jika serangan terdeteksi, mitigasi dijalankan; jika tidak, model SVM diulang untuk pelatihan ulang.
10. Data performa jaringan dikirim ke Prometheus dan divisualisasikan di Grafana secara real-time.
11. Sistem mengirimkan notifikasi otomatis ke Telegram untuk memperingatkan adanya serangan.

B. Spesifikasi Perangkat

Implementasi sistem memerlukan perangkat dengan spesifikasi untuk menjalankan pelatihan model SVM, dan monitoring. Dalam penelitian ini, seluruh proses dijalankan pada satu perangkat utama melalui VMware, dengan rincian spesifikasi tercantum pada Tabel 1

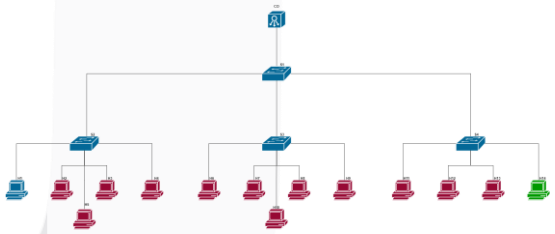
Perangkat	Versi	Spesifikasi
Laptop	Asus Vivobook Pro 15 M3500QC	AMD Ryzen™ 9 5900HX Mobile Processor (8-core/16-thread, 20MB cache, up to 4.6 GHz max boost)
		16GB DDR4
		Windows 11 Home Single Language 64-bit(10.0, Build 22631)

Tabel 1 Spesifikasi Perangkat

Spesifikasi perangkat keras yang digunakan telah memenuhi kebutuhan untuk menjalankan simulasi jaringan virtual dan pelatihan model klasifikasi. Dukungan CPU multi-core dan RAM yang memadai memungkinkan operasi paralel antar komponen sistem dalam lingkungan virtual, sehingga menjamin stabilitas dan kelancaran implementasi sistem deteksi dan mitigasi serangan DDoS dalam penelitian ini.

C. Perancangan Topologi

Topologi jaringan dirancang menggunakan Mininet berbasis arsitektur SDN, terdiri dari satu switch, satu kontroler Ryu, satu host target, beberapa host normal, dan host penyerang untuk simulasi SYN Flood. Seluruh koneksi dikendalikan oleh kontroler terpusat, memungkinkan pemantauan dan mitigasi lalu lintas secara langsung. Topologi disusun melalui skrip Python agar fleksibel terhadap skenario pengujian yang telah ditentukan.



Gambar 3 Topologi Jaringan

Topologi jaringan yang dibangun menggunakan Mininet terdiri dari satu kontroler Ryu, empat switch, dan empat belas host yang terbagi dalam tiga segmen. Switch utama (S1) terhubung ke tiga switch lainnya (S2, S3, S4), yang masing-masing melayani kelompok host berbeda. Host h1 bertindak sebagai target, h14 sebagai pengirim lalu lintas normal, sementara host lainnya (h2–h13) berperan sebagai penyerang. Lalu lintas yang tidak dikenali oleh switch diteruskan ke kontroler untuk diklasifikasi menggunakan algoritma SVM, dan IP penyerang dikenakan pembatasan melalui rate limiting. Topologi ini dirancang menyerupai kondisi jaringan nyata untuk menguji efektivitas sistem deteksi dan mitigasi serangan SYN Flood dalam lingkungan SDN.

D. Klasifikasi Host

Untuk mendukung proses simulasi dan pengujian, konfigurasi host pada topologi jaringan dijabarkan dalam tabel berikut, yang mencakup informasi alamat IP serta switch yang terhubung dengan masing-masing host.

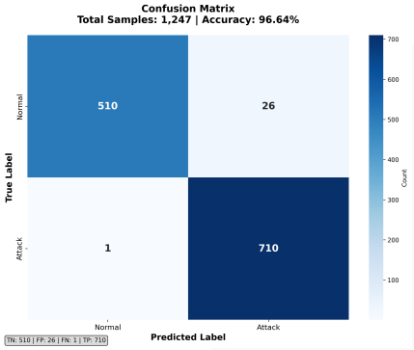
No.	Perangkat	Jumlah
1.	Kontroler	1
2.	*Switch*	4
3.	*PC (Personal Computer)*	14

Tabel 2 Perangkat yang Digunakan

Topologi jaringan yang digunakan dalam penelitian ini terdiri dari satu kontroler, empat switch, dan lima belas host (PC) yang disimulasikan dalam lingkungan virtual menggunakan Mininet. Seluruh perangkat tersebut dihubungkan dalam struktur topologi bertingkat untuk mendukung proses implementasi serta pengujian sistem deteksi dan mitigasi serangan SYN Flood.

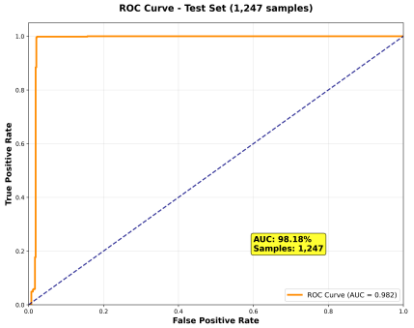
E. Pelatihan dan Uji Model

Setelah proses pelatihan model klasifikasi selesai dilakukan, tahap selanjutnya adalah menguji dan mengevaluasi kinerja model secara kuantitatif untuk memastikan efektivitasnya dalam mengklasifikasikan lalu lintas jaringan. Evaluasi ini penting tidak hanya untuk mengukur akurasi secara umum, tetapi juga untuk menilai performa berdasarkan metrik klasifikasi lainnya seperti Precision, Recall, dan F1-Score, yang seluruhnya merujuk pada hasil Confusion Matrix.



Gambar 4 Confusion Matrix

Confusion matrix merupakan alat penting untuk memahami jenis prediksi yang dilakukan model serta kesalahan yang mungkin terjadi. Berdasarkan hasil pengujian, model menghasilkan 510 True Negative (TN), 710 True Positive (TP), 26 False Positive (FP), dan 1 False Negative (FN). Dari data tersebut, diperoleh tingkat akurasi sebesar 96,64%, yang menunjukkan bahwa model mampu mengklasifikasikan lalu lintas secara akurat pada sebagian besar kasus uji.



Gambar 5 Kurva ROC

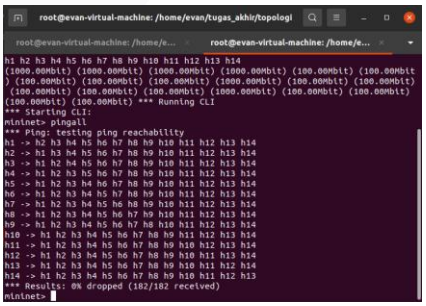
Sebagai pelengkap analisis, dilakukan visualisasi melalui Kurva ROC (Receiver Operating Characteristic) untuk menggambarkan kemampuan diskriminatif model terhadap dua kelas, yaitu lalu lintas normal dan serangan. Kurva ROC yang dihasilkan menunjukkan bentuk yang mendekati sudut kiri atas grafik, menandakan tingkat deteksi yang tinggi dengan tingkat kesalahan (False Positive Rate) yang rendah. Nilai Area Under Curve (AUC) yang diperoleh adalah 0.98, yang berarti terdapat peluang sebesar 98% bahwa model dapat membedakan secara benar antara sampel serangan dan sampel normal yang dipilih secara acak. Nilai ini menegaskan bahwa model memiliki performa klasifikasi yang sangat baik. Selain menunjukkan performa yang unggul dibandingkan algoritma lainnya, model deteksi yang baik juga harus memiliki kinerja yang konsisten serta kemampuan generalisasi terhadap data baru. Evaluasi dilakukan terhadap model SVM pada tiga jenis data berbeda, yaitu data pelatihan, validasi, dan pengujian.

Hasil pengujian menunjukkan bahwa model mencapai akurasi sebesar 96,9% pada data pelatihan, kemudian pada data validasi dengan akurasi mendapat nilai sama 96,9%, dan tetap tinggi pada data pengujian dengan akurasi 96,64%. Konsistensi nilai akurasi ini menunjukkan bahwa tidak terjadi degradasi performa yang signifikan, sehingga model mampu bekerja dengan baik pada data yang belum pernah dilihat sebelumnya.

Berdasarkan evaluasi yang telah dilakukan, pemilihan model Support Vector Machine (SVM) dapat dikatakan tepat. Selain unggul dalam berbagai metrik evaluasi, model ini juga terbukti stabil dan andal, sehingga layak digunakan sebagai komponen utama dalam sistem deteksi serangan SYN Flood pada jaringan berbasis SDN.

F. Uji Koneksi Jaringan

Sebelum melakukan pengujian terhadap metode deteksi dan mitigasi serangan, langkah awal yang dilakukan adalah memastikan konektivitas antar host dalam topologi jaringan berjalan dengan baik. Pengujian ini bertujuan untuk memverifikasi bahwa seluruh host yang disimulasikan di Mininet dapat saling terhubung tanpa hambatan, serta memastikan bahwa konfigurasi jaringan telah diterapkan dengan benar dan siap digunakan dalam skenario pengujian berikutnya.



Gambar 6 Hasil Uji Ping

G. Integrasi Monitoring dan Notifikasi

Sistem deteksi dan mitigasi serangan dalam penelitian ini dilengkapi fitur pemantauan dan notifikasi real-time menggunakan Prometheus, Grafana, dan Telegram. Prometheus bertugas mengumpulkan metrik dari Ryu Controller, sementara Grafana menyajikannya dalam bentuk

visualisasi dashboard. Untuk memudahkan respons cepat, notifikasi otomatis dikirimkan ke Telegram saat terjadi anomali pada lalu lintas jaringan.

Proses dimulai dari integrasi Prometheus sebagai *data source* di Grafana, kemudian dilanjutkan dengan ekspos metrik dari Ryu Controller melalui fungsi update_all_metrics() dan metrics_updater() yang memperbarui data setiap dua detik. Dashboard Grafana menampilkan informasi status kontroler, lalu lintas SYN, deteksi serangan, dan efektivitas mitigasi.

Notifikasi Telegram dikonfigurasi melalui fitur Contact Point di Grafana menggunakan Bot API dan chat ID. Alert dikirim otomatis berdasarkan query tertentu, seperti:

1. Status Kontroler Terhubung/Terputus: menggunakan metrik up, dengan nilai 1 (aktif) atau 0 (putus).
2. Status Serangan atau Jaringan Normal: menggunakan metrik sdn_threat_level, di mana nilai di atas 2 menandakan serangan aktif, dan di bawah 2 menandakan kondisi stabil.

Dengan sistem ini, administrator dapat mengetahui status jaringan secara cepat dan akurat tanpa harus terus-menerus memantau dashboard.

V. HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengembangkan sistem deteksi dan mitigasi serangan Distributed Denial of Service (DDoS) jenis SYN Flood pada jaringan berbasis Software Defined Networking (SDN). Sistem dirancang dengan menggabungkan algoritma klasifikasi Support Vector Machine (SVM) sebagai pendeteksi lalu lintas berbahaya dan mekanisme *rate limiting* sebagai strategi mitigasi. Pengujian dilakukan dalam berbagai skenario serangan menggunakan platform Mininet untuk mensimulasikan kondisi jaringan secara realistis.

Model SVM yang dibangun menunjukkan performa klasifikasi yang sangat baik. Berdasarkan hasil pengujian terhadap data uji, diperoleh akurasi sebesar 96,64% dan nilai AUC sebesar 0,98, yang menunjukkan kemampuan model dalam membedakan lalu lintas normal dan berbahaya secara efektif. Rangkuman performa model disajikan pada Tabel 3.

Metriks Evaluasi	Nilai
Akurasi	96.6%
Precision	97.0%
Recall	97.5%
F1-Score	98.1%
AUC	98.1%

Tabel 3Metriks Evaluasi Model

Sistem mitigasi berbasis rate limiting diuji dalam tiga jenis serangan dengan variasi jumlah host penyerang dari 4 hingga 12. Hasil pengujian dirata-rata untuk menunjukkan konsistensi performa sistem.

Metriks	Nilai Rata-Rata
Total Paket Diproses	24,769
Paket Terblokir	24,542
Tingkat Mitigasi	99.08%
False Negative	1.5

Tabel V.1 Metriks Pengujian Serangan IP Statis

Hasil pada Tabel V.2 menunjukkan bahwa sistem mampu secara konsisten memblokir hampir seluruh lalu lintas serangan IP statis, dengan tingkat keberhasilan di atas 99.08%.

Metriks	Nilai Rata-Rata
Total Paket Diproses	23,662
Paket Terblokir	23,315
Tingkat Mitigasi	98.60%
False Negative	1

Tabel 4 Metriks Pengujian Serangan IP Acak

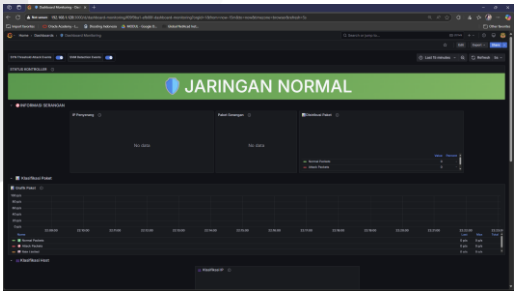
Meskipun serangan IP acak lebih sulit dikendalikan, sistem tetap menunjukkan performa yang solid, dengan tingkat mitigasi mendekati 98.60%dan jumlah false negative yang masih dalam batas aman.

Metriks	Nilai Rata-Rata
Total Paket Diproses	24,280.97
Paket Terblokir	24,034.80
Tingkat Mitigasi	99.01%
False Negative	1.55

Tabel 5 Metriks Pengujian Serangan IP Kombinasi

Pada skenario kombinasi, sistem tetap mampu mempertahankan efektivitas dengan tingkat mitigasi di atas 99.01%, menunjukkan kemampuan adaptif terhadap variasi serangan.

Sistem ini juga dilengkapi monitoring real-time menggunakan Prometheus dan Grafana, serta notifikasi otomatis ke Telegram. Dashboard Grafana menampilkan status kontroler, jumlah paket SYN, status mitigasi, serta klasifikasi lalu lintas. Notifikasi dikirim secara otomatis saat terjadi serangan, saat jaringan normal kembali, atau ketika koneksi ke kontroler terganggu.

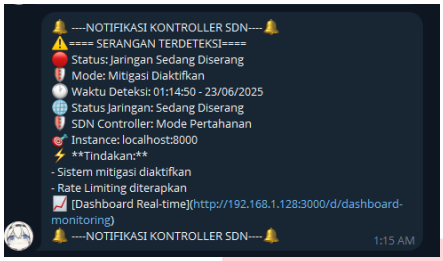


Gambar 7 Dashboard Grafana saat Jaringan Normal

Sebelum terjadi serangan, dashboard Grafana menyajikan status lalu lintas yang stabil, tingkat ancaman yang rendah, serta tidak adanya indikasi aktivitas mencurigakan. Tampilan ini menjadi referensi awal untuk membedakan kondisi normal dan ketika terjadi anomali. Dashboard ini memberikan gambaran umum bahwa seluruh metrik sistem masih berada dalam ambang batas wajar. Indikator “JARINGAN

NORMAL" juga aktif, menandakan tidak ada status firing alert yang sedang berjalan.

Ketika sistem mendeteksi serangan SYN Flood melalui peningkatan metrik sdn_threat_level, Grafana akan memicu alert dan mengirimkan pesan otomatis ke Telegram. Pesan ini berisi informasi penting seperti waktu kejadian, status mitigasi, dan jumlah IP penyerang.



Gambar 8 Notifikasi Telegram saat Serangan SYN Flood Terdeteksi

Notifikasi ini memberikan peringatan instan kepada administrator, sekaligus menyampaikan bahwa mekanisme rate limiting telah diaktifkan oleh kontroler SDN sebagai bentuk respons otomatis terhadap serangan.

Dari hasil pengujian, sistem yang dibangun terbukti andal dalam mendeteksi dan memitigasi serangan SYN Flood secara akurat dan efisien. Kemampuan klasifikasi yang kuat, efektivitas mitigasi yang tinggi di berbagai skenario, serta dukungan monitoring real-time dan alerting menjadikan sistem ini solusi yang adaptif, scalable, dan siap diterapkan pada jaringan SDN untuk menghadapi tantangan keamanan yang dinamis.

VI. KESIMPULAN DAN SARAN

Penelitian ini menunjukkan bahwa arsitektur Software Defined Network (SDN) memiliki kerentanan signifikan terhadap serangan SYN Flood, khususnya pada kontroler sebagai pusat pengendali. Pengujian menunjukkan bahwa serangan dari empat host penyerang tanpa mitigasi menyebabkan packet loss sebesar 99,17%, dengan hanya satu dari 120 paket normal yang berhasil dikirimkan, serta latensi ekstrem mencapai 18.174 ms. Untuk mendeteksi serangan ini, dikembangkan model klasifikasi berbasis Support Vector Machine (SVM) yang dilatih menggunakan dataset publik, dengan skema train-validation-test dan optimasi hyperparameter melalui Optuna. Model mencapai akurasi 96,64% dan nilai AUC 0,98, menandakan kemampuan klasifikasi tinggi. Dalam pengujian nyata pada 23 skenario (92 kali uji coba), model mencatat rata-rata false negative sebesar 1 hingga 1,5 paket per trial, baik pada skenario IP statis, acak, maupun kombinasi. Mitigasi dilakukan dengan metode rate limiting berbasis token bucket dan terbukti efektif mempertahankan integritas lalu lintas, dengan rata-rata tingkat pemblokiran paket serangan di atas 98% dan packet loss 0% pada 83 dari 92 trial. Peningkatan latensi terjadi sebagai konsekuensi proses penyaringan lalu lintas, namun layanan tetap tersedia. Sistem monitoring menggunakan Prometheus dan Grafana berhasil memberikan visibilitas real-time dan notifikasi otomatis melalui Telegram untuk mendeteksi kondisi krusial, seperti pemutusan koneksi kontroler atau deteksi serangan. Temuan ini mengindikasikan bahwa kombinasi antara deteksi berbasis SVM, mitigasi rate limiting, dan sistem monitoring real-time mampu membentuk sistem pertahanan jaringan SDN yang responsif, akurat, dan adaptif. Penelitian selanjutnya disarankan untuk menguji algoritma lain yang lebih efisien, mengembangkan mitigasi adaptif, menguji sistem pada perangkat fisik nyata, serta menambahkan fitur forensik dan laporan insiden untuk meningkatkan kesiapan dan keandalan sistem dalam skala produksi..

REFERENSI

[1] S. Khorsandroo, A. G. Sánchez, A. S. Tosun, J. M. Arco, and R. Doriguzzi-Corin, "Hybrid SDN evolution: A comprehensive survey of the state-of-the-art," *Computer Networks*, vol. 192, Jun. 2021, doi: 10.1016/j.comnet.2021.107981.

[2] N. M. Kazi, S. R. Suralkar, and U. S. Bhadade, "Performance Evaluation of RYU SDN Controller Using Mininet," *International Research Journal of Engineering and Technology*, 2021, [Online]. Available: www.irjet.net

[3] M. T. Islam, N. Islam, and M. Al Refat, "Node to Node Performance Evaluation through RYU SDN Controller," *Wirel Pers Commun*, vol. 112, no. 1, pp. 555–570, May 2020, doi: 10.1007/s11277-020-07060-4.

[4] M. Dimolianis, A. Pavlidis, and V. Maglaris, "SYN Flood Attack Detection and Mitigation using Machine Learning Traffic Classification and Programmable Data Plane Filtering," in *2021 24th Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2021*, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 126–133. doi: 10.1109/ICIN51074.2021.9385540.

[5] K. Johnson Singh, D. Maisnam, and U. S. Chanu, "Intrusion Detection System with SVM and Ensemble Learning Algorithms," *SN Comput Sci*, vol. 4, no. 5, Sep. 2023, doi: 10.1007/s42979-023-01954-3.

[6] B. Asa'ari Lubis, D. Yanuar Ar-Rafi, I. Widiyani, K. I. Lestari, and A. T. Zy, "NOVICE RESEARCH EXPLORATION (NRE) Analysis and Mitigation Technique of DDoS on Server Networks Based on Modern Technology," Jul. 2024.

[7] C. Ariza-Porras, V. Kuznetsov, and F. Legger, "The CMS monitoring infrastructure and applications," *Comput Softw Big Sci*, vol. 5, no. 1, Dec. 2021, doi: 10.1007/s41781-020-00051-x.

[8] H. Fathoni, H.-Y. Yen, C.-T. Yang, C.-Y. Huang, and E. Kristiani, "A Container-Based of Edge Device Monitoring on Kubernetes," in *Frontier Computing*, J.-W. Chang, N. Yen, and J. C. Hung, Eds., Singapore: Springer Singapore, 2021, pp. 231–237.

[9] D. Kshirsagar and S. Kumar, "A feature reduction based reflected and exploited DDoS attacks detection system," *J Ambient Intell Humaniz Comput*, vol. 13, no. 1, pp. 393–405, Jan. 2022, doi: 10.1007/s12652-021-02907-5.

[10] E. E. D. Hemdan and D. H. Manjaiah, "An efficient digital forensic model for cybercrimes investigation

in cloud computing," *Multimed Tools Appl*, vol. 80, no. 9, pp. 14255–14282, Apr. 2021, doi: 10.1007/s11042-020-10358-x.

[11] M. Abubakre, I. Faik, and M. Mkansi, "Digital entrepreneurship and indigenous value systems: An Ubuntu perspective," *Information Systems Journal*, vol. 31, no. 6, pp. 838–862, Nov. 2021, doi: 10.1111/isj.12343.