Implementasi dan Analisis Mitigasi serangan DDoS *UDP Flood* pada *Software Defined Network* dengan *Ryu Controller* menggunakan *Rate Limiting*

1st Muhammad Farhan Guslim
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
farhanguslim@student.telkomuniversity.ac.id

2nd Mochamad Teguh Kurniawan
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
teguhkurniawan@telkomuniversity.ac.id

3rd Muhammad Fathinuddin
Universitas Telkom
S1 Sistem Informasi
Bandung, Indonesia
muhammadfathinuddin@telkomuniversity.ac.id

***Abstrak* — Serangan DDoS jenis *UDP Flood* merupakan ancaman serius bagi arsitektur *Software Defined Networking* (SDN) karena sifat pengelolaannya yang terpusat, menjadikannya rentan terhadap serangan. Penelitian ini mengembangkan sistem mitigasi serangan *UDP Flood* dengan menggabungkan metode *Rate Limiting* pada *Ryu Controller* dan deteksi berbasis *Support Vector Machine* (SVM). Simulasi dilakukan pada topologi *tree* di Mininet. Hasil evaluasi menggunakan *confusion matrix* menunjukkan akurasi deteksi rata-rata sebesar 87,62% dengan 0% *false positive*. Akurasi mencapai 90% pada dua penyerang dan menurun menjadi 82% pada tujuh penyerang. Pada sisi pemulihan, ditemukan pola koneksi normal → *unreachable* → normal yang berulang, akibat aktivasi mitigasi. Rata-rata waktu pemulihan koneksi normal adalah 5,19 menit (2 penyerang), 2,30 menit (3 penyerang), 2,00 menit (5 penyerang), dan 1,48 menit (7 penyerang), namun dapat melebihi 15 menit pada kondisi ekstrem. Trafik normal baru pulih sepenuhnya setelah seluruh serangan dimatikan dan tidak ada lalu lintas mencurigakan. Sistem juga dilengkapi dashboard visual untuk pemantauan serangan dan mitigasi secara *real-time*.**

***Kata kunci*— SDN, DDoS, *UDP Flood*, *Ryu Controller*, *Rate Limiting*, SVM**

I. PENDAHULUAN

Perkembangan teknologi jaringan yang pesat membawa banyak kemudahan dalam berbagai sektor, namun juga menghadirkan tantangan baru terkait dengan ancaman terhadap keamanan jaringan. Salah satu ancaman yang terus berkembang adalah serangan *Distributed Denial of Service* (DDoS), terutama serangan jenis *UDP Flood*. Serangan ini dapat membanjiri jaringan dengan jumlah paket yang sangat besar, menyebabkan terganggunya kinerja dan ketersediaan layanan yang tergantung pada jaringan tersebut. Arsitektur *Software Defined Networking* (SDN), meskipun menawarkan fleksibilitas dan efisiensi dalam pengelolaan jaringan, ternyata juga memiliki kelemahan terkait dengan pengelolaan lalu lintas data secara terpusat, menjadikannya rentan terhadap serangan seperti DDoS [1].

Dalam upaya untuk mengatasi masalah ini, berbagai pendekatan mitigasi telah dikembangkan, salah satunya adalah dengan menggunakan metode *Rate Limiting*. Metode ini bertujuan untuk membatasi jumlah paket yang dapat dikirimkan oleh suatu sumber dalam periode waktu tertentu, sehingga serangan dapat diredam sebelum menyebabkan kerusakan lebih lanjut [2], [3].Selain itu, untuk mendeteksi serangan secara lebih efisien, digunakan algoritma *machine learning*, khususnya *Support Vector Machine* (SVM), yang dapat mengklasifikasikan lalu lintas jaringan berdasarkan pola-pola tertentu, membedakan antara trafik normal dan serangan [4] [5].

Penelitian ini bertujuan untuk mengembangkan sistem mitigasi yang memanfaatkan kedua metode tersebut *Rate Limiting* dan deteksi berbasis SVM dalam lingkungan SDN yang menggunakan *Ryu Controller*. Dengan mengimplementasikan kedua metode ini secara terintegrasi, diharapkan dapat meningkatkan keamanan dan ketersediaan jaringan SDN, serta meminimalkan dampak yang disebabkan oleh serangan DDoS, khususnya *UDP Flood*.

II. KAJIAN TEORI

II.1 Jaringan Komputer

Jaringan komputer adalah sekumpulan perangkat keras dan perangkat lunak yang saling terhubung untuk memungkinkan komunikasi antar komputer atau perangkat lain. Tujuan utama dari jaringan komputer adalah untuk berbagi sumber daya, seperti data dan aplikasi, serta meningkatkan efisiensi dalam komunikasi [6]. Jaringan ini dapat dibedakan menjadi beberapa jenis, salah satunya adalah jaringan terdistribusi yang mengandalkan perangkat untuk saling terhubung dan berkomunikasi tanpa ada kontrol terpusat yang jelas

II.2 Software Defined Network (SDN)

Software Defined Networking (SDN) merupakan paradigma arsitektur jaringan yang memungkinkan pengelolaan jaringan secara terpusat dengan memisahkan fungsi kontrol dari perangkat penerus (*data plane).* Hal ini memungkinkan pengendalian yang lebih fleksibel terhadap aliran data, serta mempermudah pemantauan dan pengelolaan jaringan [7]. Dalam SDN, komponen utama yang mengontrol lalu lintas data adalah *controller*, yang dapat dikendalikan secara terpusat untuk mengatur aliran data di seluruh jaringan. Namun, meskipun menawarkan fleksibilitas yang tinggi, SDN juga menghadapi tantangan terkait dengan

kerentanannya terhadap serangan, karena semua kontrol dikelola dalam satu titik pusat yang bisa menjadi target serangan.

II.3 DDoS dan *UDP Flood*

Serangan *Distributed Denial of Service* (DDoS) adalah upaya untuk membuat suatu layanan atau jaringan tidak dapat diakses oleh pengguna yang sah dengan membanjiri sistem target dengan lalu lintas yang sangat besar. Salah satu jenis serangan DDoS yang sering digunakan adalah *UDP Flood*, yang memanfaatkan protokol *User Datagram Protocol* (UDP) untuk mengirimkan paket secara masif ke server atau perangkat lainnya. Tujuan dari serangan ini adalah untuk menguras sumber daya sistem dan mempengaruhi ketersediaan layanan [8]. *UDP Flood* berbeda dengan serangan DDoS lainnya karena menggunakan paket UDP yang tidak memerlukan koneksi, sehingga sulit untuk dianalisis dan diblokir secara tradisional. Hal ini menjadikannya lebih efektif dalam melumpuhkan jaringan dalam waktu yang singkat.

II.4 *Rate Limiting*

Rate Limiting adalah teknik untuk membatasi jumlah data atau permintaan yang dapat dilakukan oleh pengguna atau perangkat dalam suatu periode waktu tertentu. Metode ini sangat mampu dalam mengendalikan serangan, seperti *UDP Flood*, dengan membatasi jumlah paket yang diperbolehkan masuk ke jaringan dalam interval waktu yang telah ditentukan. Implementasi *Rate Limiting* pada jaringan SDN menggunakan *controller*, seperti *Ryu Controller*, memungkinkan pembatasan aliran data dari sumber yang terdeteksi melakukan serangan, tanpa mempengaruhi koneksi normal [9]. *Rate Limiting* ini membantu menjaga ketersediaan layanan dengan mencegah jaringan dari kelebihan beban yang disebabkan oleh serangan.

II.5 *Support Vector Machine* (SVM)

Support Vector Machine (SVM) adalah salah satu algoritma *machine learning* yang digunakan untuk klasifikasi data. Dalam konteks penelitian ini, SVM digunakan untuk mendeteksi serangan dalam lalu lintas jaringan dengan menganalisis jumlah paket (*Packet Count*) yang diterima dalam periode waktu tertentu. SVM bekerja dengan cara mencari *hyperplane* yang dapat memisahkan dua kelas data yang berbeda (normal dan serangan) berdasarkan fitur yang diberikan. Algoritma ini sangat mampu dalam mengklasifikasikan data yang memiliki dimensi tinggi dan jumlah data yang terbatas, menjadikannya pilihan yang baik untuk deteksi anomali pada jaringan [4], [5]Dalam penelitian ini, SVM akan digunakan untuk mendeteksi serangan *UDP Flood* berdasarkan log lalu lintas yang tercatat oleh *controller*.

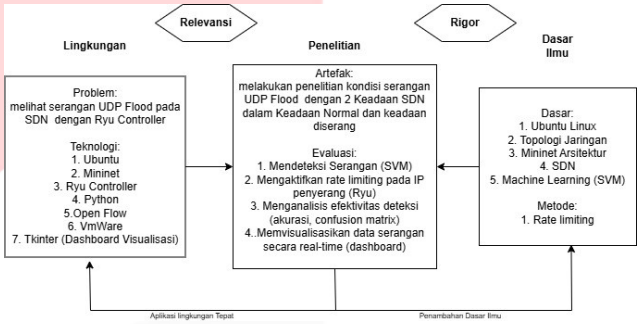
II.6 *Confusion Matrix*

Confusion Matrix adalah alat evaluasi yang digunakan untuk mengukur kinerja model klasifikasi, seperti SVM, dengan membandingkan hasil prediksi model dengan data yang sebenarnya. *Confusion Matrix*memberikan gambaran mengenai jumlah prediksi yang benar (*True Positive* dan *True Negative*) dan yang salah (*False Positive* dan *False Negative*). Metrik ini digunakan untuk menghitung berbagai indikator penting, seperti akurasi, *precision*, *recall*, dan F1-score. Penggunaan *Confusion Matrix* dalam penelitian ini bertujuan untuk mengevaluasi seberapa baik SVM dapat mendeteksi serangan *UDP Flood* dan membandingkannya dengan hasil klasifikasi yang sebenarnya[10].

III. METODE

III.1 Kerangka Berpikir

Kerangka berpikir, ini didasarkan pada konsep penelitian yang mengintegrasikan teori, fakta, dan analisis dari berbagai literatur yang relevan untuk mendukung proses penelitian. Melalui kerangka ini, variabel-variabel penelitian diuraikan secara mendalam dan relevan dengan masalah yang diteliti, sehingga memberikan fondasi yang kuat untuk menemukan solusi atas permasalahan tersebut. Kerangka berpikir ini juga bersifat dinamis, memungkinkan pembaruan agar tetap sesuai dengan perkembangan ilmu pengetahuan dan perubahan konteks yang terjadi [11].

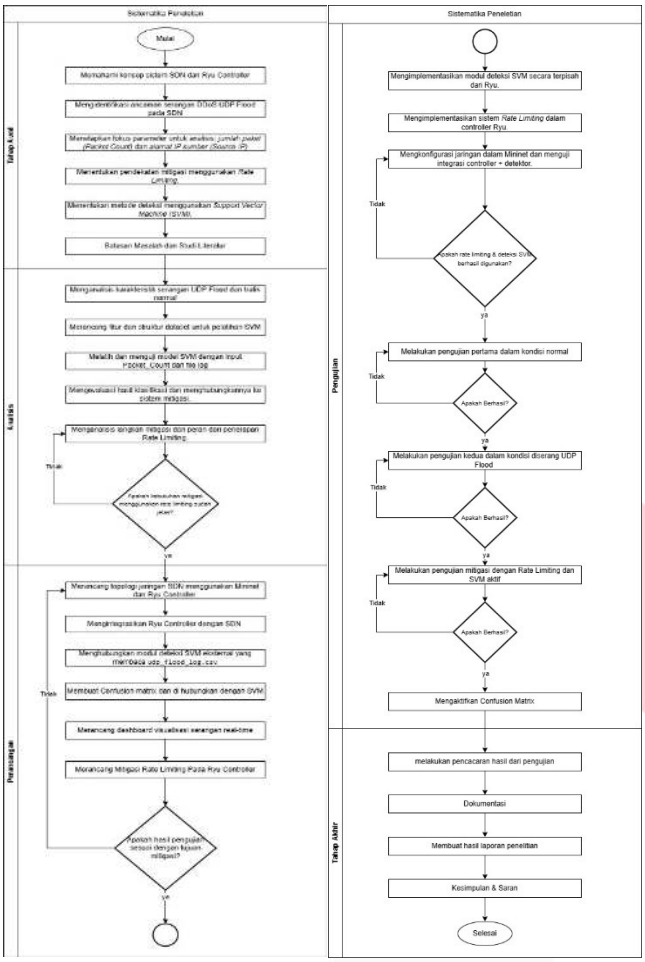


GAMBAR 1
Kerangka Berpikir

Kerangka berpikir penelitian ini menjelaskan keseluruhan proses penelitian. Pada bagian kiri (Lingkungan), dijelaskan konteks permasalahan dan teknologi yang digunakan, seperti Ubuntu, *Mininet*, dan *Ryu Controller*, yang diterapkan untuk simulasi serangan *UDP Flood*. Di bagian tengah (Penelitian), penelitian difokuskan pada artefak utama, yaitu sistem deteksi dan mitigasi serangan *UDP Flood* yang menggabungkan algoritma SVM dan *Rate Limiting*, yang kemudian dievaluasi melalui analisis log, klasifikasi SVM, serta implementasi *Rate Limiting*. Sisi kanan (Dasar Ilmu) merinci teori yang mendasari sistem, seperti arsitektur SDN, SVM, dan pengelolaan aliran data menggunakan *Ryu Controller*. Kerangka ini menunjukkan bagaimana data log dapat menghubungkan aspek teknis dan dasar ilmiah untuk implementasi mitigasi serangan di jaringan SDN.

III.2 Sistematika penyelesaian Masalah

Berikut adalah sistematika penelitian yang dapa di jelaskan pada gamba III.2 :



GAMBAR 2
Sistematika Penyelesaian Masalah

Penelitian ini dimulai dengan pemahaman tentang Software Defined Network (SDN) dan fungsi *Ryu Controller* dalam pengelolaan lalu lintas jaringan. Peneliti mengidentifikasi jenis serangan *UDP Flood* dan parameter utama, yaitu jumlah paket (Packet Count) dan alamat IP sumber (Source IP). Metode mitigasi menggunakan *Rate Limiting* dan deteksi dengan SVM dipilih sebagai solusi gabungan.

Tahap analisis bertujuan untuk memahami karakteristik *Traffic* normal dan serangan, serta menyusun dataset berbasis log jaringan untuk melatih model SVM. Evaluasi dilakukan untuk mengukur performa model dan memperbaiki fitur jika diperlukan.

Selanjutnya, tahap perancangan dilakukan dengan mendesain topologi jaringan SDN menggunakan *Mininet* dan *Ryu Controller*, serta mengimplementasikan deteksi SVM dan aktivasi *Rate Limiting*. Visualisasi status jaringan disediakan dalam bentuk Dashboard *real-time*.

Pada tahap pengujian, sistem diuji dalam dua skenario: kondisi normal dan serangan *UDP Flood* yang otomatis memicu mitigasi *Rate Limiting* dan deteksi dengan SVM. Hasil pengujian dievaluasi menggunakan *Confusion Matrix* dan catatan log.

Akhirnya, semua temuan dan hasil pengujian didokumentasikan dalam laporan akhir, termasuk grafik, log, dan Dashboard yang menunjukkan kinerja sistem.

III.3 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan melalui beberapa teknik sebagai berikut:

1. Teknik pertama adalah Studi Pustaka, di mana peneliti mengumpulkan berbagai topik referensi yang relevan, seperti paper, jurnal, dan sumber lain yang dapat dijadikan bahan rujukan untuk penelitian.
2. Teknik kedua adalah Dataset, di mana peneliti menghasilkan dataset dari simulasi yang kemudian digunakan untuk proses pengujian lebih lanjut.
3. Teknik ketiga adalah Pengujian, yang dilakukan dengan menggunakan berbagai alat pendukung seperti *Ryu Controller*, *Mininet*, VMware, *OpenFlow* dan SVM. Pengujian ini juga mencakup instalasi perangkat lunak yang diperlukan, seperti *Python*, *Hping3*, dan *Wireshark*, agar alat tersebut dapat berfungsi dengan optimal.

III.4 Pengolahan Data

Proses pengolahan data dilakukan melalui langkah-langkah berikut:

1. Desain: Peneliti merancang sistem yang meliputi *controller* SDN (Ryu), modul mitigasi *Rate Limiting*, dan deteksi serangan berbasis SVM. Sistem ini terpisah dari *controller* untuk memantau log lalu lintas UDP, mengklasifikasikan aktivitas, dan memicu mitigasi bila perlu.
2. Simulasi dan Pengujian: Serangan *UDP Flood* disimulasikan menggunakan *Hping3* di *Mininet*, dengan data performa jaringan dikumpulkan melalui *log controller* dan *Wireshark*.
3. Analisis Data: Data dianalisis untuk mengevaluasi akurasi deteksi serangan menggunakan *confusion matrix*, ketangguhan mitigasi *Rate Limiting*, dan kesesuaian log serangan dengan klasifikasi yang terdeteksi.

III.5 Metode Evaluasi

Metode evaluasi dalam penelitian ini bertujuan untuk menilai performa sistem dalam mendeteksi dan mengatasi serangan *UDP Flood* secara tepat dan cepat. Evaluasi dilakukan dengan pendekatan berikut:

1. Evaluasi Deteksi: Hasil klasifikasi model SVM divalidasi menggunakan *confusion matrix*, yang membandingkan data aktual dan prediksi. Matriks ini mengukur akurasi, *precision*, *recall,* dan kesalahan klasifikasi.
2. Evaluasi Mitigasi: Keberhasilan mitigasi diukur berdasarkan jumlah IP penyerang yang berhasil dikenali dan dibatasi dengan *Rate Limiting*, serta berkurangnya aktivitas serangan setelah pembatasan diterapkan.
3. Evaluasi Visualisasi: Evaluasi juga dilakukan dengan Dashboard *real-time* yang menunjukkan tren serangan dan status sistem, membantu memvisualisasikan kemampuan sistem dalam mendeteksi dan merespons ancaman.

III.6 Alasan pemilihan Metode

Bagian ini menjelaskan alasan pemilihan metode yang digunakan untuk menjawab rumusan masalah, dengan mempertimbangkan beberapa opsi yang relevan. Justifikasi pemilihan metode dapat dilihat pada tabel berikut.

TABEL 1
Pemilihan Metode

Rate Limiting	*Blackholing*	*Sinkholing*	*Load balancing*
Metode *Rate Limiting* untuk membatasi jumlah paket yang diterima dalam interval waktu tertentu, untuk mencegah kelebihan beban akibat serangan.	Metode *Blackholing* Mengarahkan lalu lintas mencurigakan ke *blackhole* untuk diabaikan sepenuhnya	Metode *Sinkholing* Mengalihkan lalu lintas serangan ke server *sinkhole* untuk tujuan analisis tanpa mengganggu target utama.	Metode *Load Balancing* itu membagi beban lalu lintas jaringan ke berbagai server untuk mencegah kelabihan beban
Pemilihan kerangka kerja	Kerangka kerja yang dipilih adalah *Rate Limiting* karena diterangkan dengan terpusat menggunakan *Ryu Contoller*, sangat baik untuk membatasi serangan *UDP Flood*.		

IV. ANALISIS DAN PERANCANGAN

IV.1 Perangkat yang digunakan

Pada tahap ini, peneliti menyiapkan perangkat keras dan perangkat lunak yang diperlukan untuk simulasi SDN, memastikan semua perangkat terpasang dan berfungsi dengan baik sebelum simulasi dimulai.

IV.1.1 Spesifikasi Hardware

Pada spesidfikasi *Hardware* ini dapat di lihat pada Table IV.1:

TABEL 2
Spesifikasi Hardware

Perangkat	Versi	Spesifikasi
Laptop	Asus Tuf Gaming A15	AMD Ryzen 9 5900HX
		16.0 GB
		64-bit operating system, x64-based processor

IV.1.2 Spesifikasi Software

Pada spesifikasi *Software* ini dapat di lihat pada Table IV.2:

TABEL 3
Spesifikasi Software

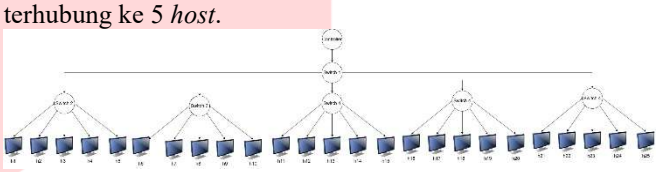
Perangkat	Versi	Deskripsi
VMWare Workstasion	16 Pro	Perangkat virtualisasi untuk menjalankan beberapa sistem operasi dalam satu komputer fisik.
Linux Ubuntu	20.04.6 LTS	Sistem operasi Linux stabil dengan dukungan jangka panjang (LTS).
Mininet	2.3.1b4	Emulator jaringan untuk simulasi topologi SDN pada satu mesin.
Python	3.8.10	Bahasa pemrograman untuk pengembangan *controller*, deteksi, dan visualisasi.
Hping3	3.0.0-alpha-2	Alat untuk mengirim paket TCP/IP, digunakan dalam simulasi serangan jaringan seperti *UDP Flood*.

IV.2 Alur Perancangan

Penelitian ini dimulai dengan merancang jaringan menggunakan *Mininet* dan mengonfigurasikan *Ryu Controller*. *Ryu Controller* mencatat lalu lintasi UDP dan mengirimkan ke SVM untuk mendeteksi. Jika ditemuka trafik yang melebihi batas normal, *controller* akan membatasi paket. Proses ini di pantau melalu Dashbord yang menampilkan status normal, serangan dan mitigasi.

IV.2.1 Perancangan Topologi

Topologi jaringan dibuat menggunakan *Mininet* dengan struktur *tree*, terdiri dari 1 Root *Switch* (s1), 5 *Switch* menengah (s2-s6), dan 25 *host*, di mana setiap *switch* terhubung ke 5 *host*.



GAMBAR 3
Topologi SDN

IV.2.2 Klasifikasi *Host*

Subbab ini menjelaskan klasifikasi *host* dalam topologi SDN yang disimulasikan menggunakan emulator *Mininet*.

TABEL 4
Jumlah Perangkat

NO	Perangkat	Jumlah
1	*Controller*	1
2	*Switch*	6
3	*Host*	25

IV.2.3 *Ryu Controller*

Ryu Controller berfungsi sebagai pengendali utama lalu lintas jaringan dengan tugas-tugas berikut:

1. Menerima seluruh paket dari jaringan.
2. Menghitung jumlah paket UDP berdasarkan alamat IP sumber.
3. Mencatat lalu lintas UDP dalam file log (UDP_*Flood*_log.csv).
4. Menerapkan *Rate Limiting* otomatis ketika IP melebihi batas tertentu (misalnya 100 paket dalam 30 detik).
5. Membatasi bandwidth atau membuang paket dari IP yang terdeteksi.

Log yang tercatat dapat digunakan untuk evaluasi dan deteksi lebih lanjut.

GAMBAR 4
Running Controller

IV.2.4 Menguji Konektivitas

Subbab ini menguji konektivitas antar *host* dalam topologi yang telah dibuat di *Mininet*. Pengujian ini penting untuk memastikan bahwa semua *host* dapat saling terhubung dengan baik, sebelum melanjutkan ke tahap berikutnya. Jika ping antar *host* gagal, maka pengujian selanjutnya tidak dapat dilanjutkan. Perintah yang digunakan untuk memastikan *host* terhubung adalah sebagai berikut:

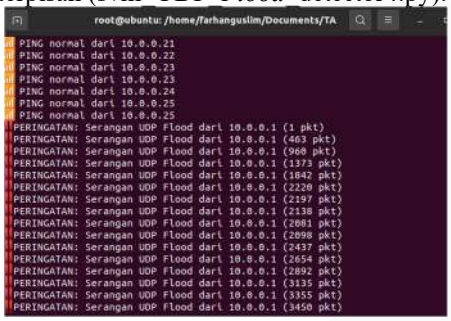
GAMBAR 5
Ping antar *Host*

Gambar IV.3 menunjukkan hasil pengujian konektivitas antar *host*, di mana setelah menjalankan perintah "pingall", pesan "h1 – h25" muncul menandakan bahwa koneksi antar *host* berjalan lancar.

IV.2.5 Model SVM (Support Vector Machine)

Model *Support Vector Machine* (SVM) digunakan untuk mengklasifikasikan IP berdasarkan jumlah paket yang dikirim, dengan dua label: normal dan *attack*.

1. Fitur input model adalah packet_count.
2. Klasifikasi output adalah "normal" atau "*attack*".
3. Model dilatih dan disimpan dalam file svm_model.pkl.
4. Proses klasifikasi dijalankan oleh skrip *Python* terpisah (svm_UDP_*Flood*_detector4.py).



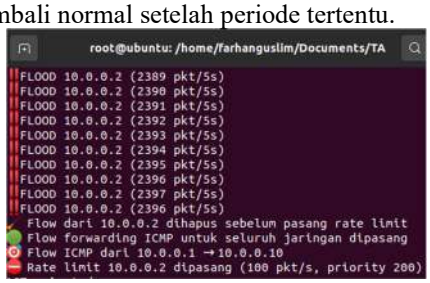
GAMBAR 6
SVM

Pada gambar IV.4 merupakan output SVM yang Dimana jika terjadinya *Traffic* normal dan serangan.

IV.2.6 Implementasi Mitigasi *Rate Limiting*

Rate Limiting diterapkan pada *Ryu Controller* berdasarkan statistik trafik UDP dari setiap IP. Mekanisme mitigasi bekerja sebagai berikut:

1. *Controller* memantau jumlah paket UDP dari setiap IP dalam periode waktu tertentu.
2. Jika jumlah paket melebihi ambang batas (misalnya 100 paket dalam 5 detik), IP tersebut dianggap mencurigakan.
3. Ryu memasang flow rule untuk men-drop paket jika diperlukan.
4. Tindakan mitigasi dicatat dengan status "Rate_Limited" di log.
5. Pembatasan akan dihapus otomatis jika trafik kembali normal setelah periode tertentu.



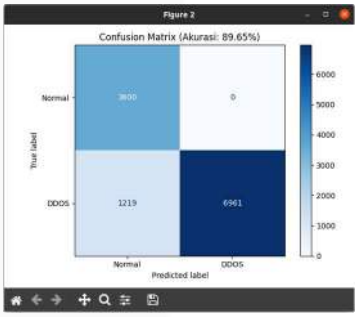
GAMBAR 7
Rate Limiting

Pada gamabar IV.5 merupakan bukti jika *Rate Limiting* itu berjalan, jika serangan *UDP Flood* sudah masuk selama 5 detik otomatis terpasang.

IV.2.7 *Confusion Matrix*

Evaluasi klasifikasi SVM dilakukan menggunakan *Confusion Matrix*untuk mengukur kinerja model dalam mendeteksi serangan. Parameter yang dihitung meliputi:

1. *True Positive* (TP): Serangan yang terdeteksi.
2. *True Negative* (TN): *Traffic* normal yang terklasifikasi dengan benar.
3. *False Positive* (FP): *Traffic* normal yang salah terdeteksi sebagai serangan.
4. *False Negative* (FN): Serangan yang tidak terdeteksi.



GAMBAR 8
Confusion Matrix

Pada gamabar IV.6 ini merupakan output *Confusion Matrix*.

IV.2.8 Dataset dan Pencacatan Log

Data utama diperoleh dari:

1. File UDP_*Flood*_log.csv yang dihasilkan oleh *Ryu Controller*.
2. Kolom utama: timestamp, Source_IP, Packet_Count, Status (PING, *Flood*_Detected, Rate_Limited).

Dataset ini digunakan untuk:

1. Evaluasi performa deteksi dan mitigasi.
2. Input klasifikasi ke model SVM.
3. Visualisasi pada Dashboard.

Dataset otomatis tercatat dalam file log CSV setelah pengujian ping normal dan serangan, yang disimpan di *library*.

IV.2.9 Dashboard Visualisasi

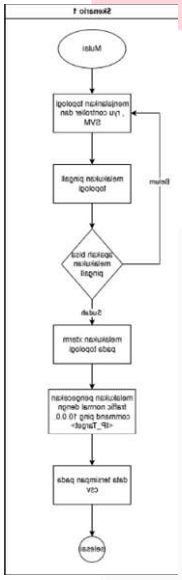
Dashboard dibangun menggunakan *Python* dengan Tkinter dan Matplotlib, menampilkan informasi seperti:

1. Jumlah IP penyerang yang terdeteksi.
2. Grafik tren serangan berdasarkan waktu.
3. Status mitigasi yang aktif pada IP tertentu.

GAMBAR 9
Dashboard

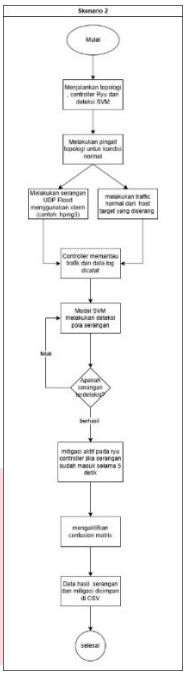
Pada gambar IV.7 merupakan gambar dari Dashboard visualisasi, yang dimana bisa mendeteksi dalam keaadaan normal, terserangan dan termitigasi.

IV.3 Skenario Pengujian



GAMBAR 10
Skenario Pengujian 1

Skenario 1 menguji kondisi normal pada jaringan SDN dengan topologi dan *controller* Ryu yang telah dibuat. Pengujian dimulai dengan menjalankan topologi dan *controller*, lalu menggunakan perintah pingall untuk memastikan semua node terhubung. Setelah itu, terminal (xterm) digunakan untuk memeriksa trafik normal dengan perintah ping ke alamat IP tertentu, yang dilakukan berulang hingga kondisi normal terkonfirmasi. Data hasil pengamatan disimpan dalam file CSV untuk analisis lebih lanjut. Skenario ini memastikan jaringan berfungsi dengan baik sebelum menguji serangan atau mitigasi.



GAMBAR 11
Skenario Pengujian 2

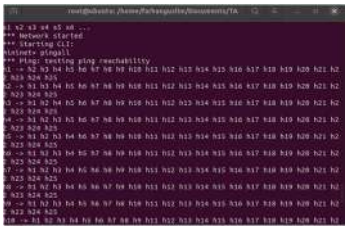
Skenario 2 menguji jaringan SDN yang diserang dengan *UDP Flood*. Pengujian dimulai dengan menjalankan topologi dan *Ryu Controller* di *Mininet*, memverifikasi konektivitas menggunakan perintah pingall, dan kemudian melakukan simulasi serangan dengan *Hping3* untuk mengirim paket UDP ke *host* target. *Ryu Controller* memantau dan mencatat lalu lintas ke dalam log secara *real-time*, yang kemudian dianalisis oleh model SVM untuk mendeteksi apakah lalu lintas tersebut normal atau serangan. Jika serangan terdeteksi dan berlangsung lebih dari 5 detik, *Ryu Controller* mengaktifkan mitigasi *Rate Limiting* untuk membatasi paket dari IP penyerang tanpa mengganggu trafik normal. Evaluasi dilakukan dengan *Confusion Matrix*untuk mengukur akurasi deteksi. Pengujian mencakup empat skenario dengan jumlah penyerang yang berbeda: 2 penyerang tanpa *Traffic* normal, 3 penyerang dan 1 *host* normal, 5 penyerang dan 1 *host* normal, serta 7 penyerang dan 1 *host* normal. Akurasi deteksi oleh SVM menurun seiring bertambahnya jumlah penyerang, namun mitigasi berhasil membatasi serangan tanpa mengganggu trafik normal.

V. Hasil dan Aanalisis

V.1 Skenario 1Pengujian Kondisi Normal

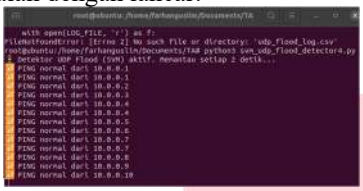
Pada skenario 1, topologi SDN dan *controller* Ryu dijalankan tanpa serangan. Konektivitas diuji dengan perintah pingall untuk memastikan semua node terhubung, kemudian ping dilakukan secara berkala dari beberapa *host* ke *host* target untuk memastikan trafik normal. Seluruh aktivitas dicatat dalam file log CSV.

V.1.1 Hasil Pengujian



GAMBAR 12
Menjalankan Topologi

Pada Gambar V.1 merupakan bukti menjalankan *Mininet* topologi berjalan dengan lancar.



GAMBAR 13
Menjalankan *Ryu Controller*

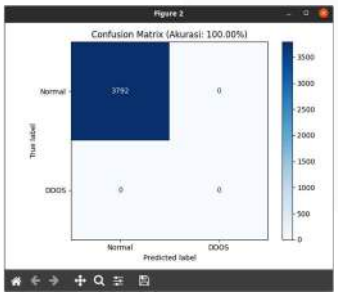
Pada Gambar V.2 merupakan bukti menjalankan *Ryu Controller* yang dimana pada saat pingall otomatis *controller* akan membaca ping nya.

V.1.2 Analisis Pengujian
Dari hasil pengujian ini dapat dilihat data nya dalam bentuk csv dan *confusion matrix*.

TABEL 5
Data CSV Normal

No	Timestamp	Source_IP	Packet_count	Status
1	2025-05-27 05:33:04	10.0.0.1	1	PING
2	2025-05-27 05:33:04	10.0.0.2	1	PING
3	2025-05-27 05:33:04	10.0.0.3	1	PING
4	2025-05-27 05:33:04	10.0.0.4	1	PING

Pada Tabel V.1 meruapakan dataset keaadan ping normal.



GAMBAR 14
*Confusion Matrix*Normal

Pada gambar V.3 merupakan output *Confusion Matrix*jika menjalankan Ping Normal, otomatis akurasi nya 100% dikarenakan tidak ada serangan.

V.2 Skenario Pengujian 2 Serangan *UDP Flood* pada SDN

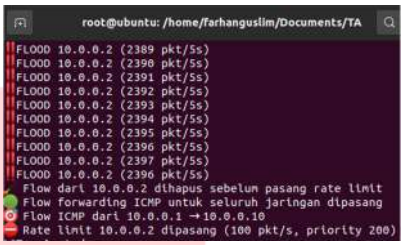
Pada skenario 2, setelah memastikan kondisi jaringan normal, dilakukan simulasi serangan *UDP Flood* dengan mengirimkan paket UDP dari beberapa *host* ke *host* target menggunakan perintah :

Hping3 --udp --*Flood* -p 80 10.0.0.<IP_Target>

Ryu Controller yang terintegrasi dengan SVM memantau paket secara *real-time*, mengenali serangan setelah 5 detik, dan mengaktifkan mitigasi *Rate Limiting* dengan membatasi IP penyerang. Model SVM terpisah juga dijalankan untuk mendeteksi trafik normal dan serangan tanpa mitigasi. Semua aktivitas tercatat dalam file log CSV.

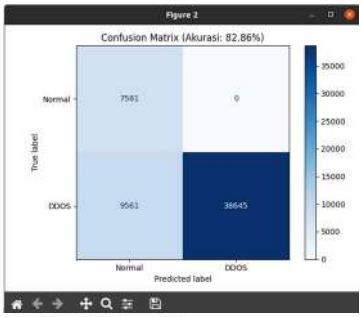
V.2.1 Hasil Pengujian

Pengujian dilakukan dengan serangan DDoS *UDP Flood* menggunakan 2-7 penyerang terhadap *host* target. Berikut hasilnya.



GAMBAR 15
Tampilan Seranagan *Ryu Controller*

Gambar V.4 menunjukkan pengujian serangan DDoS *UDP Flood*, di mana *Ryu Controller* mendeteksi serangan dan secara otomatis menerapkan mitigasi *Rate Limiting* setelah 5 detik serangan terus-menerus.



GAMBAR 16
*Confusion Matrix*7 Serangan & 1 *Traffic* Normal

Pada Gambar V.5 merupakan *Confusion Matrix*yang dimana data sudah masuk ke SVM yang akurasi semakin banyak serangan akurasi nya menurun.

V.2.2 Analisis Pengujian
Hasil pengujian serangan *UDP Flood* dengan variasi jumlah penyerang menunjukkan bahwa akurasi deteksi SVM menurun seiring bertambahnya penyerang. Namun, dengan dua penyerang aktif, sistem mencapai akurasi 90,03%, meskipun terjadi 1306 *false negative*, menunjukkan deteksi yang hampir sempurna.

TABEL 6
Analisis Data 2 serangan

No	Jumlah Penyerang	*Traffic* Normal	Akurasi (%)	*False Negative*
1	2	0	90.03	1306

Pada table V.2 merupakan tabel analisis data 2 penyerang.

TABEL 7
Analisis Keseluruhan Data

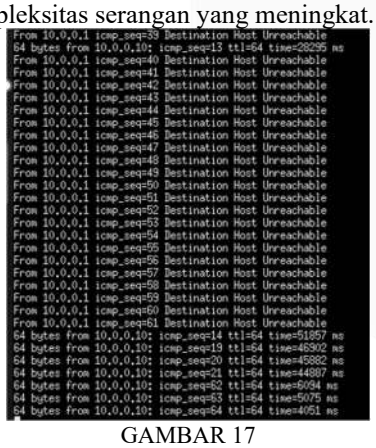
No	Jumlah Penyerang	*Traffic* Normal	Akurasi (%)	*False Negative*
1	2	0	90.03	1306
2	3	1	88.18	2864
3	5	1	87.39	4394
4	7	1	82.86	9561

Tabel V.3 menunjukkan bahwa meskipun tidak ada *false positive*, di mana trafik normal tidak terdeteksi sebagai serangan, model SVM tetap mampu mendeteksi trafik biasa dengan baik. Mekanisme *Rate Limiting* berhasil membatasi serangan setelah terdeteksi selama 5 detik. Secara keseluruhan, meskipun akurasi deteksi menurun dengan bertambahnya jumlah penyerang dari 90,03% dengan 2 penyerang menjadi 82,86% dengan 7 penyerang sistem tetap efektif. Jumlah *false negative* juga meningkat, tetapi tidak ada trafik normal yang salah diklasifikasikan sebagai serangan. Ini menunjukkan sistem masih andal dalam mengenali aktivitas normal meskipun mengalami penurunan akurasi saat serangan meningkat.

TABEL 8
Rata rata metrik

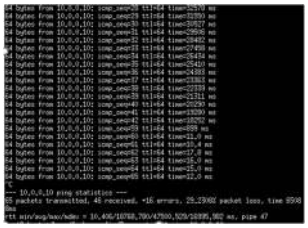
No	Metrik	Rata-rata
1	Akurasi (%)	87.62
2	*False Negative*	4531.25
3	*False Positive*	0.00
4	Penurunan Akurasi (%)	2.39

Tabel V.8 menunjukkan bahwa rata-rata akurasi deteksi sistem adalah 87,62%, yang menunjukkan kemampuan sistem dalam mengenali serangan meskipun jumlah penyerang bertambah. Namun, *false negative* yang tinggi (4531,25) mengindikasikan banyak serangan yang terdeteksi saat terjadi serangan bersamaan. Yang positif, tidak ada *false positive*, artinya trafik normal tidak salah diklasifikasikan sebagai serangan. Penurunan akurasi rata-rata sebesar 2,39% menunjukkan bahwa kemampuan deteksi menurun seiring dengan kompleksitas serangan yang meningkat.



GAMBAR 17
Traffic normal pada saat disereng dan di mitigasi

Pada Gambar V.6 merupakan output *Traffic* normal yang diserangan setelah itu di mitigasi, maka *Traffic* normal nya akan unreacble setelah itu akan ping normal secara berulang.



GAMBAR 18
Setelah Serangan dimatikan

Pada Gambar V.7 merupakan merupakan output *Traffic* normal yang diserangan setelah itu di mitigasi, maka *Traffic* normal nya akan menurun ms nya.
Berikut ini adalah estimasi waktu pemulihan koneksi normal berdasarkan jumlah *host* penyerang yang terlibat:

Penyerang	Estimasi Kondisi *Traffic* Normal					Rata – Rata
	Normal	Unreachable	Normal	Unreachable	Normal	Normal
2	3.23 menit	3.5 menit	6.13 menit	6.31 menit	5 menit	5.19 menit
3	2.50 menit	6.56 menit	2.24 menit	11.19 menit	2.17 menit	2.30 menit
5	2.38 menit	20.47 menit	1.30 menit	1.2 menit	1.17 menit	2 menit
7	1.48 menit	5 menit	1.49 menit	15.30 menit	1.49 menit	1.48menit

GAMBAR 19
Estimasi *Traffic* Normal Pulih

Waktu pemulihan rata-rata koneksi normal dihitung dengan memperhitungkan kondisi *Unreachable* dan Normal setelah mitigasi diterapkan dan pengujian ini di uji selama 25 menit. Estimasi waktu pemulihan berdasarkan jumlah *host* penyerang adalah sebagai berikut:

- 2 penyerang: rata-rata 5,19 menit
- 3 penyerang: rata-rata 2,30 menit
- 5 penyerang: rata-rata 2 menit
- 7 penyerang: rata-rata 1,48 menit

Hasil menunjukkan bahwa meskipun mitigasi *Rate Limiting* mampu dalam mengembalikan koneksi, waktu pemulihan sangat dipengaruhi oleh jumlah penyerang dan intensitas serangan. Semakin banyak penyerang, semakin lama waktu pemulihan yang diperlukan. Pada skenario dengan 2-3 penyerang, pemulihan lebih cepat, namun waktu pemulihan tetap bervariasi tergantung pada kondisi serangan dan kemampuan jaringan

Berikut Dashboard Visualisasi jika terjadi Serangan , *Rate Limiting*, dan *Traffic*

GAMBAR 20
Dashboard Normal

Pada Gambar V.9 merupakan Dashboard Normal.

GAMBAR 21
Dashboard Serangan

Pada Gambar V.10 merupakan Dashboard di serang.

GAMBAR 22
Dashboard *Rate Limiting*

Pada Gambar V.11 merupakan Dashboard di *Rate Limiting*.

VI. Kesimpulan dan Saran

VI.1 Kesimpulan

Berdasarkan perancangan, pengujian, dan analisis yang dilakukan, dapat disimpulkan bahwa:

1. Mitigasi Serangan *UDP Flood*: Penggunaan *Rate Limiting* pada *Ryu Controller* mampu dalam membatasi serangan *UDP Flood*, dengan kontrol terpusat yang membantu menjaga stabilitas jaringan.
2. Deteksi Serangan dengan SVM: Algoritma SVM menunjukkan hasil memuaskan dalam membedakan trafik normal dan serangan *UDP Flood*, meskipun akurasinya menurun dengan meningkatnya jumlah penyerang. *Acuracy* mencapai 90% pada dua penyerang, namun turun menjadi sekitar 82% pada tujuh penyerang.
3. Waktu Pemulihan Jaringan: Proses pemulihan koneksi menunjukkan pola berulang berupa normal → unreachable → normal sebelum stabil sepenuhnya. Estimasi rata-rata waktu pemulihan berdasarkan jumlah host penyerang adalah: 5,19 menit (2 penyerang), 2,30 menit (3 penyerang), 2,00 menit (5 penyerang), dan 1,48 menit (7 penyerang). Dalam beberapa kasus, pemulihan dapat memakan waktu lebih dari 15 menit. Koneksi jaringan baru benar-benar pulih jika serangan telah dimatikan dan tidak ada lalu lintas mencurigakan yang tersisa.
4. Pemantauan dan Visualisasi: Sistem dilengkapi dengan Dashboard visual untuk pemantauan *real-time* status serangan dan mitigasi, memungkinkan pengguna menilai kinerja sistem dalam menangani serangan.

VI.2 Saran

Berdasarkan hasil penelitian, beberapa saran yang dapat diberikan adalah:

1. Peningkatan Akurasi Deteksi SVM: Meskipun SVM memadai, akurasi deteksi masih dapat ditingkatkan dengan melatih model menggunakan dataset yang lebih besar dan menambahkan fitur yang lebih rinci tentang karakteristik paket.
2. Penyesuaian *Rate Limiting*: Batasan laju *Rate Limiting* bisa dioptimalkan lebih lanjut dengan menyesuaikan dinamis sesuai dengan jenis serangan atau kondisi jaringan yang terdeteksi.
3. Metode Mitigasi Lain: Penggunaan teknik mitigasi tambahan, seperti *blackholing* atau *sinkholing*, dapat memperkuat perlindungan jaringan.
4. Pengujian pada Topologi Lebih Besar: Pengujian dengan topologi yang lebih besar dan kompleks, serta uji coba di dunia nyata, akan memberikan gambaran lebih akurat tentang kemampuan sistem.
5. Perbaikan Visualisasi: Dashboard visual dapat dikembangkan lebih lanjut dengan menambahkan fitur seperti notifikasi otomatis, grafik waktu nyata, dan analisis prediktif untuk mempermudah pemantauan jaringan.

REFERENSI

[1] T. Irfan, "*Software Defined Networking* (SDN) dalam Manajemen Jaringan Kontemporer: Survey Komprehensif dan Evaluasi Tren Terkini."

[2] admin leravio, "Apa itu *Rate Limiting* ?," admin leravio. Accessed: Nov. 14, 2024. [Online]. Available: https://leravio.com/blog/apa-itu-rate-limiting/

[3] W. Haniyah, M. C. Hidayat, Z. F. I. Putra, V. A. Pertama, and A. Setiawan, "Simulasi Serangan Denial of Service (DoS) menggunakan *Hping3* melalui Kali Linux," *Journal of Internet and Software Engineering*, vol. 1, no. 2, p. 8, Jun. 2024, doi: 10.47134/pjise.v1i2.2654.

[4] K. Putri *et al.*, "Implementasi Algoritma *Support Vector Machine* dalam Klasifikasi Deteksi Depresi dari Postingan pada Media Sosial," *Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 2, no. 1, 2023.

[5] Muhammad Salim Mursid, "SISTEM DETEKSI SERANGAN DDoS PADA SOFTWARE DEFINED NETWORK MENGGUNAKAN METODE SUPPORT VECTOR MACHINE," Bandung, 2024.

[6] kyndryl, "Software-defined networking (SDN) vs traditional networking explained," kyndryl. Accessed: Dec. 05, 2024. [Online]. Available: https://www.kyndryl.com/us/en/learn/sdn-vs-traditional-networking

[7] N. Abyan, D. / Asisten, E. Ahmad, and Z. Hamidi, "F (1147070055), Meta Ayu lestari (1147070043), Saepul Kholik (1147070067)/ Kelompok 3," 2017.

[8] Q. Syahputra, D. Akbi, and D. Risqiwati, "Deteksi Dan Mitigasi Serangan DDoS Pada Software Defined Network Menggunakan Algoritma Decision Tree," *REPOSITOR*, vol. 2, no. 11, pp. 1491–1502, 2020.

[9] PT. Letun cloud asia, "Pemahaman Mendalam tentang *Rate Limiting* dan Manfaatnya ." Accessed: Nov. 19, 2024. [Online]. Available: https://www.leyun.asia/id/apa-itu-rate-limiting-dan-manfaatnya/#:~:text=Rate%20limiting%20adalah%20proses%20membatasi,ketersediaan%20layanan%20bagi%20pengguna%20lain.

[10] Nisha Arya Ahmed, "What is A *Confusion Matrix*in *Machine learning*? The Model Evaluation Tool Explained," datacamp.com.

[11] F. D. F. R. S. Addini Zahra Syahputri1, "Tarbiyah: Jurnal Ilmu Pendidikan dan Pengajaran," 2023. [Online]. Available: https://jurnal.diklinko.id/index.php/tarbiyah/https://jurnal.diklinko.id/index.php/tarbiyah/