REPOSITOR, Vol. 2, No. 7, Juli 2020, Pp. 915-922
ISSN : 2714-7975
E-ISSN : 2716-1382

Analisa Sistem Identifikasi DDoS Menggunakan KNN Pada Jaringan Software Defined Network(SDN)

Muhammad Misbahul Azis*[1] Yufis Azhar[2], Saifuddin[3]
[1,2,3]Teknik Informatika/Universitas Muhammadiyah Malang
misbahulazis@webmail.umm.ac.id[*1], syaifuddin@umm.ac.id[3]

Abstrak

Kebutuhan pada jaringan mengutamakan performa untuk mendukung sebuah efisiensi jaringan merupakan hal penting untuk saat ini. Penentuan konfigurasi yang semakin banyak dan kompleks serta kontrol jaringan yang semakin rumit, membuat jaringan semakin tidak fleksibel dan susah untuk diterapkan pada sebuah topologi jaringan yang besar. Software Defined Network (SDN) muncul dengan mekanisme yang dapat menyelesaikan masalah tersebut. Pada dasarnya konsep dari Software Defined Network (SDN) adalah memisahkan kontroller dan data/forwarding plane, sehingga mampu untuk me-menejemen jaringan yang begitu banyak dalam sebuah kontroller. Namun pada kontroller belum memiliki keamanan yang cukup untuk melindungi dari serangan jaringan seperti DDoS, SYN Flooding Attack sehingga kontroller akan menjadi target dari attacker. Sehingga penelitian ini mengusulkan penambahan aplikasi machine learning pada kontroller untuk menangani serangan seperti DDoS dan SYN Flooding Attack. Dalam penelitian ini kontroller yang digunakan adalah ryu controller yang menggunakan bahasa pemrograman python. Dalam penelitian ini menggunakan topologi linear pada mininet dan membuat paket dalam format .pcap untuk pengujian serangan yang dilakukan. Sehingga dapat mengetahui rata-rata jumlah paket yang masuk dan keluar dan keberhasilan dalam melakukan mitigasi terhadap paket yang dianggap DDoS.

Kata Kunci: *Software Defined Network, Ryu Controller, K-Nearest Neighbors, Mininet, SYN-Flooding Attack*

Abstract

The need for the network to prioritize performance to support a network efficiency is important for now. Determination of configurations that are more and more complex and increasingly complicated network control, makes the network more inflexible and difficult to apply to a large network topology. Software Defined Network (SDN) appears with a mechanism that can solve the problem. Basically, the concept of Software Defined Network (SDN) is to separate the controller and the data / forwarding plane, so that it is able to manage so many networks in a controller. But the controller does not have enough security to protect against network attacks such as DDoS, SYN Flooding Attack so the controller will be the target of the attacker. So this study proposes adding machine learning applications to controllers to handle attacks such as DDoS and SYN Flooding Attack. In this study the controller used is the Ryu controller that uses the Python programming language. In this study using a linear topology on Mininet and create a package in. Pcap format for testing attacks carried out. So as to know the average number of incoming and outgoing packages and success in mitigating packages that are considered DDoS.

Keywords: *Software Defined Network, Ryu Controller, K-Nearest Neighbors, Mininet, SYN-Flooding Attack*

1. Pendahuluan

Software Defined Network (SDN) bersifat terpusat pada sebuah kontroller yang dimana dapat mengatur semua aktivitas pada sebuah jaringan. Perbedaan vendor perangkat jaringan seperti Cisco, Jitter, dan Mikrotik bisa saling terhubung karena di software Defined Network (SDN) semua konfigurasi pada setiap vendor dapat di install/masukan pada controller [1].

Pada infrastruktur jaringan SDN DDoS sangat mungkin untuk menyerang perangkat kontroller karena otak dari SDN adalah kontroller. Distributed Denial of Service (DDoS) jenis serangan yang sering dipakai untuk melakukan kejahatan pada sebuah perangkat jaringan. Seiring berkembangnya teknologi pada jaringan maka jenis serangan DDoS juga semakin

meningkat, beberapa tahun ini serangan DDoS meningkat jumlahnya dan semakin kompleks pada setiap jenis yang diserangannya [2]. Konsep dasar dari DDoS adalah memanfaatkan banyak ip untuk melakukan serangan pada jaringan dengan mengirim paket request yang sangat banyak membuat padatnya lalu lintas pada sebuah jaringan sehingga kerja dari perangkat menjadi lebih keras dari biasanya. Hal ini mengakibatkan kerusakan pada perangkat jaringan, yang menjadi salah satu faktor pendorong utama yang dilakukan oleh para penyerang [3].

Mitigasi DDoS salah satu upaya untuk mengatasi masalah serangan DDoS. Mekanisme dari mitigasi DDoS adalah dengan memperkecil bandwidth dan blocking agar jumlah paket permintaan dapat berkurang durasinya. Sehingga lalu lintas jaringan tidak terjadi kepadatan yang membuat kerja switch dan router menjadi terbebani, karena hal ini berpengaruh terhadap kerusakan perangkat jaringan tersebut. Mitigasi DDoS pada jaringan SDN diusulkan untuk mengatasi masalah serangan DDoS dengan mekanisme yang sama dengan jaringan lama, namun, pada SDN berpengaruh pada server kontroller yang dapat berakibat kerusakan pada perangkat jaringan tersebut [4].

Dalam penelitian sebelumnya, mendeteksi DDoS pada jaringan SDN telah dikombinasikan dengan Intrusion Detection System (IDS) yang di implementasikan pada kontroller. Terdapat 2 modul dalam IDSnya, modul pertama memproses permintaan dan menemukan host yang memiliki perilaku normal dan anomali jika terdapat perilaku yang anomali maka host tersebut akan dikirimkan pada modul kedua yang dimana akan memeriksa paket pada host tersebut apakah benar bahwa perilakunya anomali. Dengan adanya 2 modul ini pada penelitian sebelumnya berhasil mempersingkat waktu pemrosesannya. Hal ini dikarenakan modul satu dijalankan terlebih dahulu untuk mengurangi jumlah host yang akan periksa paket datanya [5].

Dalam penelitian lainnya, mendeteksi DDoS pada jaringan SDN dan dilakukan tindakan mitigasi dengan mengkombinasikan sFlow dan OpenFlow. sFlow adalah teknologi kolektor yang berfungsi sebagai mengumpulkan data pada lalu lintas jaringan yang ada pada Switch dan Router sehingga membuat jaringan mudah terlihat. OpenFlow merupakan protokol komunikasi yang berfungsi sebagai pemberi hak akses forwarding data pada Switch dan Router melalui jaringan. Dua teknologi tersebut digunakan peneliti untuk mengembangkan sebuah metode baru yakni “FlowTrApp” yang digunakan untuk mendeteksi dan me-mitigasi serangan dengan cara mengumpukan data lalu lintas pada jaringan dan mengambil tindakan mitigasi apabila aliran data yang berjalan tidak sesuai dengan batas yang telah ditentukan pada jaringan [1].

Pada penelitian sebelumnya tentang Detection of Distributed Denial of Service Attack in Sofware Defined Network masih belum ada tindakan apa yang akan dilakukan setelah benar menemukan penyerang [5]. Maka dalam penelitian ini penulis menggunakan DDoS dengan tipe TCP Flood Attack pada jaringan Software Defined Network (SDN) untuk dilakukan DDoS mitigasi di jaringan SDN dengan menggunakan K-Nearest Neighbors (KNN) pada kontroller. Aplikasi machine learning dalam kontroller berfungsi sebagai deteksi dan mitigasi paket DDoS dengan cara menginstruksi pada switch untuk menginstall flow mitigasi DDoS. Mengenali pola serangan yang sudah dipelajari oleh K-Nearest Neighbors (KNN) melalui dataset akan mempercepat membuat kesimpulan bahwa adanya penyerang. Ryu kontroller akan melakukan tindakan mitigasi terhadap penyerang yang telah dibenarkan oleh K-Nearest Neighbors (KNN). Penerapan K-Nearest Neighbors (KNN) pada jaringan SDN bertujuan untuk meningkatkan akurasi dalam mendeteksi penyerang sehingga identitas dari penyerang dapat terlihat. Sehingga penelitian ini diharapkan dapat berkontribusi banyak dalam menunjang tingkat keamanan pada jaringan SDN, khususnya pada keamanan kontroller yang ada di jaringan Software Defined Network (SDN).

2. Algoritma K - Nearest Neighbors dan SDN

2.1 K – Nearest Neighbors

K-Nearest Neighbor merupakan algoritma sederhana yang mengklasifikasikan data berdasarkan ukuran kesamaan. Klasifikasi dilakukan berdasarakan data pembelajaran dengan objek yang terdekat dari data pembelajaran tersebut. Penulis menggunakan jarak sebagai ukuran kesamaan[5]. Pada penulisan ini metode KNN yang digunakan adalah library dari scikit-learning metode dari scikit learning ini adalah menemukan jumlah sampel pelatihan yang telah ditentukan jarak terdekatnya ke titik sampel baru, dan memprediksi label dari ini. Jumlah sampel berupa konstanta yang ditentukan pengguna (pembelajaran tetangga terdekat k), atau bervariasi berdasarkan kepadatan titik lokal (pembelajaran tetangga berbasis radius). Proses klasifikasi yang dilakukan dalam algortima KNN akan dijelaskan secara rinci dibawah ini [5].

2.1.1 Classification

Klasifikasi merupakan teknik yang digunakan untuk mengelompokkan data berdasarkan variabel yang dipelajari dengan tujuan untuk memprediksi data baru yang belum memiliki kategori atau kelas. Klasifikasi mempelajari data sebelumnya untuk memprediksi data baru yang tidak memiliki kategori atau kelas agar dapat di prediksi secara akurat. Klasifikasi pada KNN terbagi menjadi 2 pengklasifikasian tetangga yang berbeda. Pertama teknik klasifikasi yang mengimplementasikan pembelajaran berdasarkan tetangga terdekat dari setiap titik kueri, di mana nilai integer ditentukan oleh pengguna. Kedua teknik klasifikasi yang mengimplementasikan pembelajaran berdasarkan jumlah tentangga dalam radius atau jarak yang tetap dimana radius akan ditentukan oleh pengguna [6]. Adapun tahapan perhitungan manual dari metode KNN.

1. Menghitung jarak Eucledian
   $d\ (xi, xj) = \sqrt{\sum_{r=1}^{n}(ar\ (xi) - ar(xj))^2}$
   Keterangan:
   d(xi,xj) = jarak *Eucledian*
   xi = *record ke-i*
   xj = *record ke-j*
   ar = *data ke-r*
2. Mengurutkan berdasarkan nilai *Eucledian*
3. Menentukan nilai *K* klasifikasi terdekat
4. Terget *output* merupakan kategori mayoritas

2.1.2 Accuracy & F1-Score

Klasifikasi KNN dapat dihitung hasil prediksinya dalam hal ini digunakan untuk mengetahui seberapa bagus atau baik klasifikasi KNN dalam memprediksi data baru yang ada didalam dataset atau diluar dataset. Berikut ini adalah rumus untuk mencari nilai f1-score dengan menggunakan library sklearn. Nilai akurasi merupakan rasio kebenaran aplikasi KNN dalam memprediksi seluruh dataset yang digunakan dalam penelitian ini.

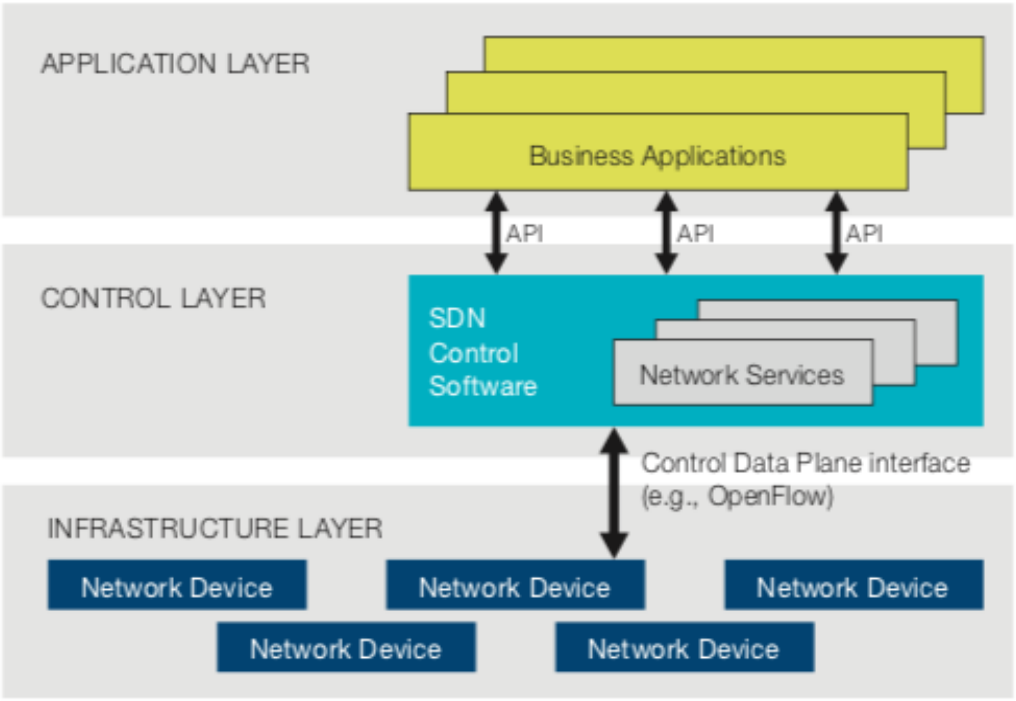
Akurasi = (TP + TN)/ (TP+FP+FN+TN)
F1 = 2 * (precision * recall) / (precision + recall)
Precission = (TP) / (TP+FP)
Recall = (TP) / (TP + FN)
Keterangan :
TP : True Positif
TN : True Negatif
FP : False Positif
FN : False Negatif

Akurasi digunakan untuk mencari nilai prediksi yang dilakukan oleh KNN dengan diuji oleh x test. Recall digunakan untuk mencari nilai hasil prediksi yang positif daripada nilai keseluruhan data yang positif benar. Precission digunakan untuk mencari nilai hasil prediksi yang negatif daripada nilai keseluruhan data yang negatif benar. F1 digunakan untuk mencari nilai perbandingan antara precission dan recall. Pada rumus diatas digunakan untuk melakukan pengujian 2 yang dilakukan pada penelitian ini [7].

2.2 Software Defined Network

Software defined network (SDN) istilah kosep baru dalam mendesain, mengelola dan mengimplementasikan jaringan. Kebutuhan dan inovasi yang semakin lama semakin kompleks membuat SDN hadir untuk mendukung semua kebutuhan dan inovasi dalam bidang jaringan. Menurut Nisharani Meti dkk software defined network adalah kemampuan dalam menyediakan virtualisasi jaringan, membuat aturan jaringan yang dinamis, dan kontrol yang lebih besar atas entitas jaringan di seluruh struktur jaringan dengan mengurangi biaya operasional. SDN memberikan beban yang besar pada administrator untuk secara manual menjamin keamanan dan fungsi yang benar dari seluruh jaringan, dimana administrator membuat aplikasi pada kontroller untuk mengatur dan memberi keamanan pada seluruh jaringan karena SDN bersifat terpusat [5].

Dikutip dari Open Network Foundation (ONF) dengan judul “Software Define Networking : The New Norm For Networks” SDN didefinisikan sebagai arsiktektur jaringan yang memisahkan antara fungsi control dan forwarding, sehingga operator jaringan dan administrator dapat mengkonfigurasi jaringan secara sederhana dan terpusat di antara ribuan perangkat. Dalam arsitektur jaringan SDN terdapat 3 layer yang memiliki fungsi penting dalam jaringan yaitu application layer, control layer, dan infrastructure/data layer. Aplication layer adalah interface yang mengelola atau mengembangkan sebuah jaringan SDN. Control layer adalah sebuah kontroller pusat yang mengatur jaringan yang berbasis software. infrastructure/data layer adalah perangkat jaringan yang terhubung dengan controller [8].



Gambar 1. Arsitektur Jaringan SDN

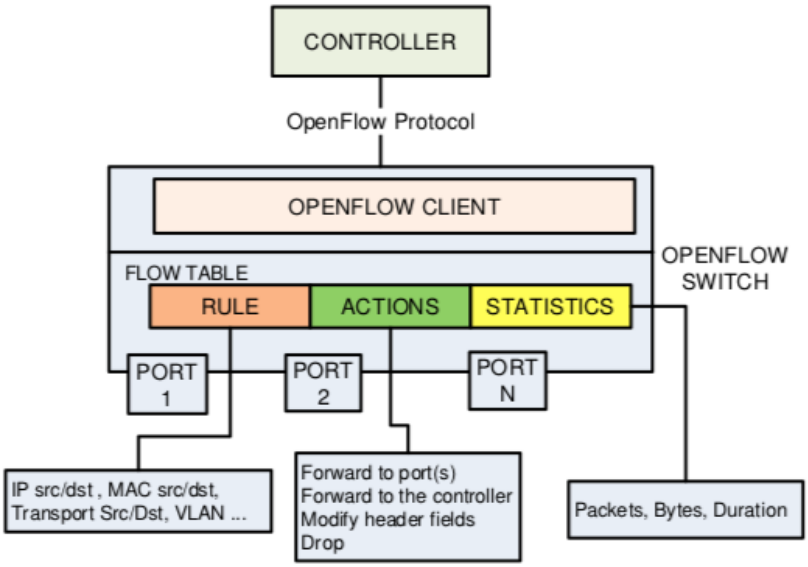
Pada Gambar 1 application layer dapat meminta dan mendapatkan layanan dari infrastruktur jaringan yang mendasarinya. Kemampuan ini memberikan dampak pada application layer lebih proaktif dan dinamis dalam meningkatkan pengalaman pengguna. SDN mengubah cara kerja jaringan, di mana aplikasi memiliki kontrol lebih besar pada konfigurasi infrastruktur jaringan. SDN menawarkan peluang untuk membangun jaringan dengan meningkat kesadaran dan kecerdasan aplikasi tentang atribut protokol Layer 4 - Layer 7 untuk memungkinkan jaringan menjadi otomatis dan lebih mandiri[9].

2.2.1 OpenFlow

OpenFlow merupakan suatu protokol komunikasi yang menghubungkan antara perangkat jaringan dengan sebuah kontroller. Dapat melakukan pemrograman pada data plane dan memberikan akses pada forwarding plane dari switch atau router melalui jaringan. Dengan menerapkan OpenFlow pada switch atau router maka dapat mengatur sebuah paket yang masuk secara terpusat pada kontroller. Dalam teknologi OpenFlow memiliki 2 bagian penting yaitu OpenFlow Switch dan Flow Table yang akan di jelaskan dibawah ini[10][11][12][13].

2.2.2 OpenFlow Switch

OpenFlow Switch atau Forwarding Device telah aktif apabila OpenFlow sudah dimasukan dalam switch atau router. OpenFlow switch memiliki satu atau lebih flow table dan layer abstrak. Protokol openflow adalah komunikasi yang digunakan kontroller untuk berkomunikasi pada switch atau router. Dalam flow table memiliki beberapa informasi yang dimana akan digunakan untuk menentukan bagaimana paket akan diproses dan diteruskan[10][11][14].



Gambar 2. Arsitektur OpenFlow Switch

Pada Gambar 2 merupakan arsitektur dari OpenFlow Switch. Dalam flow table fungsi rule adalah mencocokan paket yang masuk berupa informasi header paket, port masuk dan metadata. Apabila telah ditemukan maka paket akan diteruskan menuju tujuan sesuai informasi yang ada pada header paket. Fungsi statitics merupakan pengumpulan data statistik untuk flow table seperti jumlah paket yang diterima, jumlah byte dan durasi aliran paket. Flow table merupakan tabel yang berisi entri, dimana masing-masing entri ditentukan bagaimana paket yang masuk diubah menjadi aliran (flow) yang akan diproses dan diteruskan [12].

2.2.3 Ryu Controller

Ryu kontroller adalah kerangka kerja yang mendukung software defined network. Ryu menyediakan komponen perangkat lunak dengan API yang memudahkan dalam melakukan pengembangan aplikasi pada kontroller, melakukan aplikasi manajemen dan kontrol pada jaringan SDN. Ryu kontroller menggunakan basis bahasa python untuk menjalakan aplikasi dan memanajemen jaringan di dalamnya [15]. Ryu memiliki arsitektur monolitik yang dimana proses utama dijalankan sebagai proses tunggal. Setiap aplikasi dan layanan akan dieksekusi pada instruksi yang berbeda. Setiap aplikasi menggambarkan proses yang dapat diminati misalnya packet in, flow mod, flow removed, dll. Setelah semua proses telah diamati maka proses aliran paket akan diteruskan sesuai aliran yang telah diamati.

2.2.4 Mininet

Mininet adalah emulator CLI yang digunakan untuk membuat sebuah topologi jaringan secara virtual pada jaringan software defined network. Di dalam mininet sudah ada beberapa contoh topologi yang di sediakan oleh mininet, beberapa diantaranya adalah minimal, single, reversed, linear, dan tree. Membuat topologi secara manual dapat dilakukan dalam mininet dengan memasukan beberapa baris kode python [16].

3. Hasil Penelitian dan Pembahasan

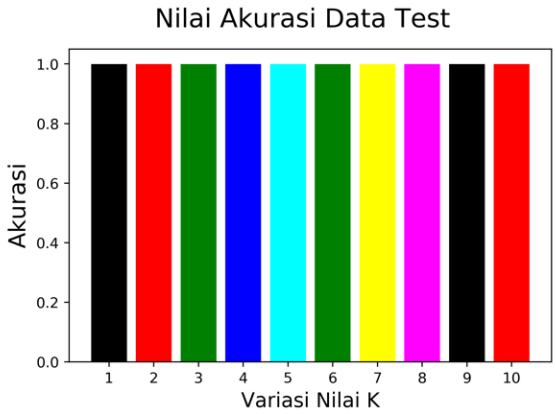
3.1 Data

Data yang digunakan dalam penelitian ini adalah dataset CICIDS (Canadian Institute for Cybersecurity Intrusion Detection Systems) tahun 2017 yang berisikan serangan umum yang digunakan oleh penyerang yang menyamai data dunia nyata (PCAP) [17]. Aliran label dalam dataset meliuti ip source, ip destinations, port source, port destinations, start time dan stop time, dll. Data tersebut digunakan sebagai data tipe serangan yang akan dipelajari oleh KNN. Dataset memiliki demensi 225745 x 85 dengan jumlah baris data 225745 dan 85 kolom. Tipe data didalam dataset ada tiga yang meliputi int64, float64, dan object. Terdapat 2 kategori dalam dataset yaitu DDoS dan BENIGN dengan masing-masing jumlah dari setiap kategori adalah 128027 untuk DDoS dan 97718 untuk BENIGN. Hasil modifikasi dataset yang digunakan untuk penelitian ini

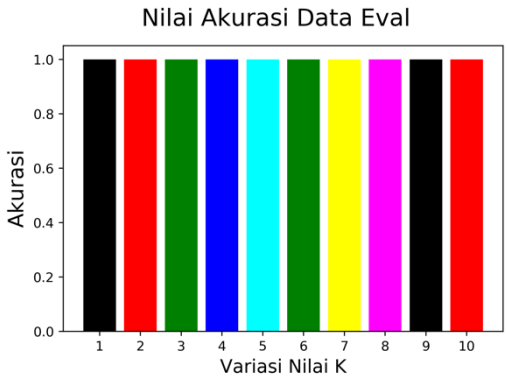
adalah mengurangi jumlah data 225745, dengan tujuan menyeimbangkan data yang digunakan untuk training dan test. Data yang digunakan berjumlah 180000 dengan jumlah proporsi ketegori dari masing-masing label adalah 90000.

3.2 Pengujian

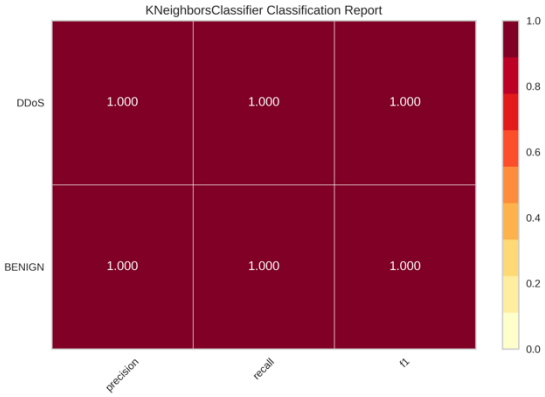
Pada Gambar 3, Gambar 4, Gambar 5, dan Gambar 6 pengujian dilakukan perbandingan antara data test dan data evaluasi pada sisi *machine learning* dan jaringan SDN, kedua mengukur rata-rata jumlah packet in dan out dengan menambahkan mitigasi serangan pada ryu controller. Pertama perbandingan data test dan data evaluasi dalam hasil prediksi dengan memvariasikan nilai k pada saat pembuatan model KNN yang digunakan untuk memprediksi data test dan data evaluasi. Pengujian ini dilakukan dengan menggunakan google colab. Tujuan dari penggunaan google colab untuk memudahkan dalam pengembangan penelitian.



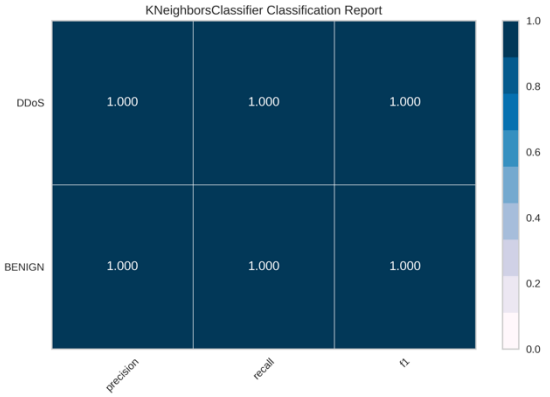
Gambar 3. Hasil dari Akurasi Data Test



Gambar 4. Hasil dari Akurasi Data Evaluasi



Gambar 5. Hasil dari F1-Score Data Test



Gambar 6. Hasil dari F1-Score Data Evaluasi

Pada Tabel 1, Tabel 2, dan Tabel 3 pengujian klasifikasi dalam jaringan software defined network (SDN) penulis mengukur jumlah paket yang masuk dan keluar, aturan mitigasi dan akurasi dalam melakukan klasifikasi serangan DDoS. Pengujian dilakukan dengan menggunakan simulasi jaringan dengan mininet dan ryu sebagai controller. Bahasa yang digunakan ada python dengan tujuan agar lebih mudah untuk melakukan pengembangan selanjutnya.

Tabel 1. Hasil Evaluasi Klasifikasi pada SDN

Jenis Data	Akurasi Nilai Prediksi	F1-Score
Data Test	0.999	0.999
Data Evaluasi	0.999	0.999

Tabel 2. Hasil Pengujian Packet IN

Frekuensi Serangan SYN-Flood *Attack*	Jumlah Paket Masuk		Waktu	Rata-rata Jumlah *Packet In* $Avg = \frac{Jumlah\ Paket\ Masuk}{Waktu\ Paket\ Masuk}$	
	Data Test	Data Evaluasi		Data Test	Data Evaluasi
1000 pkt/d	101 pkt	92 pkt	36 detik	2.805 pkt/d	2.555 pkt/d
2000 pkt/d	358 pkt	717 pkt	18 detik	19.888 pkt/d	39.833 pkt/d

Tabel 3. Hasil Pengujian Packet Out

Frekuensi Serangan SYN-Flood *Attack*	Jumlah Paket Keluar		Waktu	Rata-rata Jumlah *Packet Ou* $Avg = \frac{Jumlah\ Paket\ Keluar}{Waktu\ Paket\ Keluar}t$	
	Data Test	Data Evaluasi		Data Test	Data Evaluasi
1000 pkt/d	22 pkt	13 pkt	36 detik	0.611 pkt/d	0.361 pkt/d
2000 pkt/d	59 pkt	137 pkt	18 detik	3.277 pkt/d	7.611 pkt/d

4. Kesimpulan

Berdasarkan penelitian ini didapatkan sebuah kesimpulan bahwa dalam mengklasifikasi paket DDoS bisa dilakukan dengan menggunakan bahasa python. Metode KNN menjadi salah satu metode yang cocok untuk mengklasifikasikan paket DDoS pada jaringan SDN karena pada hasil yang di dapat dalam penelitian ini mencapai nilai 1 pada prediksi dan pengujian f1-score. Mitigasi yang dilakukan dalam penelitian ini berhasil dengan menggabungkan aplikasi KNN pada kontroller untuk memprediksi paket serangan yang datang dan sebagai acuan untuk mengaktifkan flowmod mitigasi. Hal ini ditunjukan pada tabel packet out yang memiliki rata-rata rendah. Untuk pengembangan selanjutnya mungkin bisa ditambahkan algoritma clustering untuk memprediksi paket serangan yang tidak memiliki label pada dataset sehingga dapat di labelkan oleh clustering dan diklasifikasikan.

Referensi

[1] C. Buragohain and N. Medhi, "FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers," *3rd Int. Conf. Signal Process. Integr. Networks, SPIN 2016*, pp. 519–524, 2016.

[2] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "ArOMA: An SDN based autonomic DDoS mitigation framework," *Comput. Secur.*, vol. 70, pp. 482–499, 2017.

[3] J. N. Bakker, "Intelligent Traffic Classification for Detecting DDoS Attacks using SDN/OpenFlow," 2017.

[4] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions," *Arab. J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, 2017.

[5] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," *2017 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2017*, vol. 2017-Janua, pp. 1366–1371, 2017.

[6] David Cournapeau, "Nearest Neighbors — scikit-learn 0.21.0 documentation." [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html. [Accessed: 10-May-2019].

[7] B. Y. Pratama, "Personality Classification Based on Twitter Text Using Naive Bayes , KNN and SVM," pp. 170–174, 2015.

[8] O. N. F. W. Paper, "Software-Defined Networking : The New Norm for Networks," 2012.

[9] A. VMDC, "Evolution of Software Defined Networking within Cisco ' s VMDC Challenges within the Data Center SDN Architectural Framework and Solution Characteristics," pp. 1–10, 2013.

[10] M. H. Hidayat, N. R. Rosyid, Y. Sekip, U. Iv, and Y. Indonesia, "Analisis Kinerja dan Karakteristik Arsitektur Software-Defined Network Berbasis OpenDaylight Controller," pp. 194–200, 2017.

[11] R. M. Negara and R. Tulloh, "Analisis Simulasi Penerapan Algoritma OSPF Menggunakan RouteFlow pada Jaringan Software Defined Network (SDN)," vol. 9, no. 1, pp. 75–83, 2017.

[12] F. Dwi, S. Sumadi, and D. R. Chandranegara, "Controller Based Proxy for Handling NDP in OpenFlow Network," vol. 4, no. 1, pp. 55–62, 2019.

[13] R. T. Kokila, S. Thamarai Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," *6th Int. Conf. Adv. Comput. ICoAC 2014*, pp. 205–210, 2015.

[14] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection Using NOX / OpenFlow," pp. 408–415, 2010.

[15] Admin, "Ryu SDN Framework - Build SDN Agiley." [Online]. Available: https://osrg.github.io/ryu/. [Accessed: 08-May-2019].

[16] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as Software Defined Networking Testing Platform," *Int. Conf. Commun. Comput. Syst.*, pp. 3–6, 2014.

[17] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," no. Cic, pp. 108–116, 2018.