Jurnal Informatika Universitas Pamulang **ISSN: 2541-1004**
Penerbit: Program Studi Teknik Informatika Universitas Pamulang **e-ISSN: 2622-4615**
Vol. 6, No. 4, Desember 2021 (672-679) **10.32493/informatika.v6i4.12680**

Kombinasi Metode K-Nearest Neighbor dengan Cosine Similarity untuk Prediksi Serangan Firewall pada Jaringan Komputer

Rahmawan Bagus Trianto[1], Andri Triyono[2], and Dhika Malita Puspita Arum[3]

[1,2,3]Ilmu Komputer, Fakultas Sains dan Kesehatan, Universitas An Nuur, Jl. Gajah Mada No.7, Majenang, Kuripan, Kec. Purwodadi, Kabupaten Grobogan, 58112
e-mail: [1]rahmawanbagust@gmail.com, [2]andritriyono1@gmail.com, [3]dhika.malita.11@gmail.com

Submitted Date: August 22nd, 2021 Reviewed Date: January 05th, 2022
Revised Date: January 06th, 2022 Accepted Date: January 31st, 2022

Abstract

The security of the computer network, especially the internet, is very crucial to note. One of the most effective ways to secure a computer network is to use a firewall. However, making a firewall that is still manual will make it difficult for network administrators to secure their computer network. The automatic detection of attacks on the firewall will further enhance the security of the computer network. Prediction or detection of attacks on the firewall automatically and intelligently can use the *K-Nearest Neighbor* algorithm by measuring the distance of data similarity using *Cosine Similarity*. The results of this study managed to achieve a high accuracy, which is 99.71%, precision is 74.70% and recall is 74.85% of predicting traffic that goes to the firewall. The results can be used as a standard of accuracy in predicting the traffic leading to the firewall, or even create an additional firewall so that the security of computer networks, especially the user data is saved.

Keywords: Computer network; Prediction; K-Nearest Neighbor; Firewall; Cosine Similarity;

Abstrak

Kemanan suatu jaringan komputer, khususnya internet sangat penting untuk diperhatikan. Salah satu cara yang paling efektif untuk mengamankan jaringan komputer adalah dengan menggunakan *firewall*. Akan tetapi pembuatan *firewall* yang masih manual akan menyulitkan *network administrator* untuk mengamankan jaringan komputernya. Dengan adanya otomatisasi deteksi adanya serangan pada *firewall* akan lebih meningkatkan keamanan jaringan komputer tersebut. Prediksi atau deteksi adanya serangan pada firewall secara otomatis dan cerdas dapat memakai algoritma *K-Nearest Neighbor* dengan pengukuran jarak kemiripan data menggunakan *Cosine Similarity*. Hasil dari penelitian ini berhasil mencapai akurasi yang cukup tinggi, yaitu sebesar 99.71%, presisi sebesar 74.70% dan recall sebesar 74.85% dalam memprediksi trafik serangan yang menuju pada *firewall*. Dengan performa tersebut dapat dijadikan standar dalam memprediksi trafik yang mengarah ke *firewall*, atau bahkan membuat *firewall* tambahan agar keamanan jaringan komputer, khususnya data pengguna tetap terjaga untuk menghindari kerugian baik yang bersifat materi maupun non materi.

Kata kunci: Jaringan komputer; Prediksi; *Firewall*; *K-Nearest Neigbour*; *Cosine Similarity*;

1. Pendahuluan

Firewall merupakan salah satu pilar penting dalam dunia keamanan jaringan komputer yang berfungsi untuk mengelola trafik data, baik trafik yang masuk maupun yang keluar (Ertam & Kaya, 2018). Data trafik tersebut tidak semuanya aman bagi pengguna jaringan komputer, hal ini dapat membuka ancaman terhadap integritas data mereka (Khosroshahi & Shahinzadeh, 2016). Pengguna jaringan komputer pada sebuah instansi yang memiliki nilai, maka akan semakin meningkatkan peluang untuk merasakan percobaan peretasan jaringan. Padahal, semua aktifitas yang menyangkut tentang keamanan pada jaringan komputer harus diatur pada *firewall*. Dengan kata lain, *firewall* merupakan lapisan pembatas yang

dapat memilah data yang aman dan data yang membahayakan (Khosroshahi & Shahinzadeh, 2016).

Sejauh ini, kebanyakan administrator jaringan komputer membuat *firewall* untuk melakukan filter trafik secara manual (Chiche & Meshesha, 2021). Pembuatan *firewall* secara manual yang tidak tepat dapat memberikan celah keamanan lain karena faktor *human error* atau kesalahan konfigurasi. Selain itu, tingkat pengetahuan para administrator jaringan juga mempengaruhi tingkat keamanan *firewall* yang dibuat. Hal ini menjadikan keamanan jaringan komputer yang awalnya diharapkan meningkat menjadi tidak dapat tercapai, bahkan dapat menimbulkan kerentanan lain di dalam jaringan.

Prediksi pengelompokan aksi atau tindakan terhadap data trafik pada *firewall* perlu dilakukan agar keamanan jaringan komputer semakin kuat (Ertam & Kaya, 2018). Semakin meningkat keamanan data pengguna internet atau jaringan komputer, maka akan semakin menurunkan resiko kerugian pengguna, baik dari segi materi maupun non materi. Penanganan aktifitas yang berpotensi merugikan pengguna, bahkan sampai dapat merusak perangkat komunikasi adalah hal yang penting dan menjadi tujuan dari sebuah sistem pendeteksi serangan pada jaringan komputer (Chiche & Meshesha, 2021).

Metode-metode *machine learning* seperti *Support Vector Machine* (SVM) (Ertam & Kaya, 2018), *K-Nearest Neighbor* (K-NN) (Chen, Zhou, Li, Zhang, & Huo, 2020), Naïve Bayes dan *Neural Network* (Chicco & Jurman, 2020) sudah sering dipakai untuk prediksi ataupun klasifikasi. SVM memiliki kekurangan apabila dataset yang dipakai jumlahnya besar (Saleh, Rabie, & Abo-Al-Ez, 2016). Naïve Bayes memiliki kelemahan seperti ketergantungan pada dataset dan juga kompleksitas dalam proses komputasinya (Nazari & Mahdavi, 2018). *Neural Network* memiliki kelemahan semakin besar dataset pelatihan, waktu yang dibutuhkan juga semakin lama. Selain itu, NN untuk melakukan penghitungan dataset berupa numerik tidak bekerja dengan baik (Mujtaba et al., 2019).

Metode K-NN dipilih pada penelitian ini karena mudah diterapkan, memiliki efisiensi yang tinggi, dapat bekerja pada dataset numerik (Dewi, Obert, & Gusmana, 2018). Akan tetapi, K-NN memiliki kekurangan, yaitu akurasi bergantung pada jumlah nilai *k* serta metode dalam menghitung jarak data latih dengan objek yang akan diprediksi (Rivki, Muhammad; Bachtiar, 2017). Untuk menentukan nilai *k* optimum akan dipakai cara perulangan. Penggunaan teknik perulangan ini dapat dipakai untuk memilih nilai *k* optimal secara otomatis, sehingga dapat meningkatkan hasil akurasi prediksi (Hassanat, Abbadi, Altarawneh, & Alhasanat, 2014). Sedangkan untuk menentukan jarak terdekat data latih dengan objek tujuan menggunakan *Cosine Similarity*, di mana metode ini dapat menghitung kemiripan suatu objek data dengan data latih dengan baik (Rivki, Muhammad; Bachtiar, 2017).

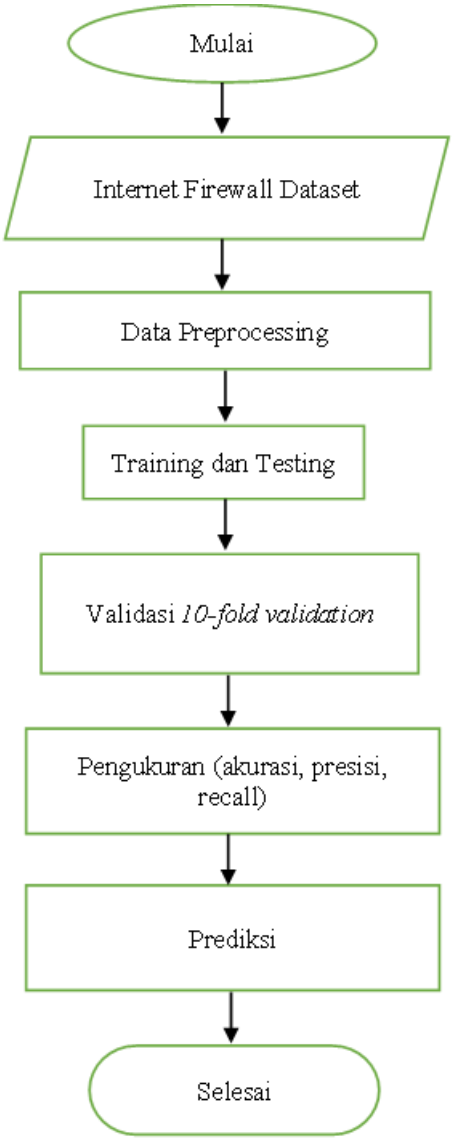
Dengan menggunakan kombinasi K-NN dan *Cosine Similarity* pada penelitian ini, diharapkan dapat membuat sebuah model prediksi secara otomatis dan cerdas yang dapat membentuk *firewall* pada jaringan komputer agar keamanan data pengguna bisa terjaga dengan baik.

2. Metode Penelitian

2.1. Data

Penelitian ini memakai *open access dataset* yang dapat diakses oleh siapa saja. *Internet Firewall Dataset* dipakai pada penelitian ini yang bisa diakses melalui laman https://archive.ics.uci.edu/ml/datasets/Internet+Firewall+Data. Dataset ini bersumber pada salah satu router di sebuah Universitas di Turki, yaitu Firat University. Dataset ini berkaitan dengan data trafik yang ada di router serta aksi yang dilakukan terhadap trafik tersebut. Aksi ini dapat dipakai sebagai firewall di router dalam rangka keamanan jaringan komputer di Universitas Firat. Jumlah dataset sebanyak 65.532 baris dengan 11 (sebelas) atribut, yaitu *Source Port, Destination Port, NAT Source Port, NAT Destination Port, Action, Bytes, Bytes Sent, Bytes Received, Packets, Elapsed Time (sec), pkts_sent, pkts_received* dan 1 (satu) kelas label, yaitu *Action*. Adapun jumlah kelas pada dataset ini berjumlah 4 (empat), yaitu *allow*, *drop*, *deny* dan *reset-both*.

2.2. Metode yang Diusulkan



Gambar 1. *Flowchart* metode yang diusulkan

Tahap awal pada proses prediksi serangan pada *firewall* di jaringan komputer dilakukan *pre-processing*. Tahap ini dilakukan proses pemilihan fitur yang dipakai. Fitur yang tidak dipakai akan dibuang. Namun, pada penelitian ini semua fitur yang ada akan dipakai karena memiliki hubungan dengan data trafik. Setelah itu dilakukan proses training dan testing. Proses ini menggunakan metode atau algoritma *K-Nearest Neighbor* (K-NN) di mana diawali dengan pemilihan nilai *k*. Pemilihan nilai *k* dilakukan secara perulangan untuk mendapatkan performansi yang maksimal. Selain itu, dilakukan juga uji coba algoritma penghitung jarak terdekat pada objek yang akan diuji dengan dataset, yaitu *Manhattan Distance, Euclidean Distance, Cosine Similarity, Canberra* dan *Jaccard*. Pengujian ini juga bertujuan untuk mencari performa akurasi terbaik pada algoritma K-NN. Selanjutnya dilakukan testing dan validasi hasil pengujian dengan *k-fold validation*, di mana pada penelitian ini menggunakan 10 *fold*. Setelah itu dilakukan evaluasi dengan mencari akurasi, presisi dan recall dari model yang telah didapatkan. Dan terakhir dilakukan prediksi serangan pada *firewall* dari model yang di dapatkan.

2.3. Tahap Pre-Processing

Pemilihan fitur dari dataset dilakukan untuk menentukan fitur-fitur apa saja yang dianggap penting dan berpengaruh di hasil akhir. Pada tahap ini semua fitur diambil untuk dipakai pada proses selanjutnya.

2.4. *K-Nearest Neighbor* (K-NN)

K-Nearest Neighbor (K-NN) merupakan sebuah algoritma klasifikasi terbimbing untuk menentukan objek baru dikelompokkan ke dalam kelas tertentu berdasarkan anggota terbanyak dari jumlah *k* tetangga terdekatnya (Hassanat et al., 2014). Oleh karena itu, penentuan nilai *k* sangat vital untuk menentukan hasil akhirnya. Pada penelitian ini memakai teknik perulangan. Teknik ini dapat dipakai karena bisa mencari nilai *k* yang optimal secara otomatis (Hassanat et al., 2014). Pencarian nilai *k* dilakukan antara 1 sampai dengan 100. Karena jumlah kelas dari dataset adalah genap, maka pemilihan nilai *k* akan diincrement secara ganjil (Wahyono, Trisna, Sariwening, Fajar, & Wijayanto, 2020). Format perulangan dapat dituliskan sebagai berikut.

for (k = 1; k <= 100; k+=2) **(1)**

Untuk *k* adalah jumlah tetangga terdekat, kemudian dilakukan increment atau penambahan nilai *k* sebanyak 2 kali, karena dimulai dari 1, agar menjadi nilai ganjil, dan proses ini berulang sampai nilai *k* menyentuh nilai 100. Di dalam proses perulangan tersebut, dilakukan pencarian jarak terdekat menggunakan beberapa metode, yaitu *Manhattan Distance, Canberra, Cosine Similarity, Euclidean Distance* dan *Jaccard*.

Secara umum algoritma K-NN dapat dilihat sebagai berikut (Dinata, Akbar, & Hasdyna, 2020).

a. Menentukan nilai *k*
b. Menghitung jarak dataset dengan objek baru
c. Mengurutkan nilai terkecil dari jarak yang telah dihitung
d. Tentukan jarak terkecil sampai urutan ke *k*
e. Kelompokkan pada kelas yang terpilih

Adapun persamaan perhitungan jarak dapat dilihat pada persamaan-persamaan berikut, dimulai dari *Manhattan Distance* dapat dilihat pada persamaan 2 berikut.

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \quad \textbf{(2)}$$

Di mana d(x,y) adalah jarak atau *distance* dengan menghitung selisih dari jarak x_i dan y_i (Latifah, 2015). Persamaan 3 memuat persamaan perhitungan jarak dengan pendekatan *Canberra Distance*.

$$d(x,y) = \sum_{i=1}^{n} \frac{|x_{ai} - x_{bi}|}{|x_{ai}| - |x_{bi}|} \quad \textbf{(3)}$$

Di mana d(x,y) merupakan jarak dengan membagikan selisih nilai x_i dan y_i secara absolut (Wurdianarto, Wurdianarto, Novianto, & Rosyidah, 2014). Kemudian persamaan 4 adalah *Euclidean Distance* sebagai berikut.

$$d_{(x1,x2)} = \sqrt{\sum_{i=1}^{n} (x_{2i} - x_{1i})^2} \quad \textbf{(4)}$$

Di mana untuk mencari nilai d_i menggunakan akar kuadrat dari selisih nilai dataset dengan objek yang baru (x2 dan x1) (Samuel, Natan, & Syafiqoh, 2018). Persamaan 5 merupakan persamaan *Cosine Similarity*, yaitu dengan cara menghitung tingkat kemiripan antara dua buah objek (Nurdiana, Jumadi, & Nursantika, 2016).

$$CosSim(x,y) = \frac{x.y}{|x|\,|y|} = \frac{\sum_{i=1}^{a} (xi\,.\,yi)}{\sqrt{\sum_{i=1}^{a} (xi)^2 \,.\sum_{i=1}^{a} (yi)^2}} \quad \textbf{(5)}$$

Kemudian pada persamaan 6 berikut ini untuk menghitung jarak menggunakan *Jaccard Similarity*. Metode ini hamper sama seperti *Cosine Similarity* dengan mengihitung tingkat kemiripan dua buah objek, hanya saja persamaannya saja yang berbeda (Nurdiana et al., 2016).

$$J(x,y) = \frac{\sum_{i=1}^{a} (xi\,.\,yi)}{\sum_{i=1}^{a} (xi)^2 + \sum_{i=1}^{a} (yi)^2 - \sum_{i=1}^{a} x_i . y_i} \quad \textbf{(6)}$$

Di mana x dan y merupakan objek dan data dari dataset.

2.5. Validasi dan Pengukuran

Validasi yang dipakai pada penelitian ini adalah *k-fold validation* dengan *k* sebanyak 10 bagian. Pembagian dataset dijadikan 10 bagian, 9 bagian dipakai untuk pelatihan dan sisa bagian lainnya digunakan untuk proses pengujian. Tahap ini diulang sebanyak 10 kali (Wahono, Herman, & Ahmad, 2014). Proses ini dapat dilihat pada tabel 1 berikut.

Tabel 1. Pembagian dataset pada *10-fold validation*

Validasi ke-k	Pembagian dataset									
1	■									
2		■								
3			■							
4				■						
5					■					
6						■				
7							■			
8								■		
9									■	
10										■

Bagian yang berwarna hitam menunjukkan dataset yang dipakai untuk proses pelatihan. Sedangkan bagian sisanya berarti dataset yang dipakai untuk proses pelatihan.

Performa prediksi perlu dihitung untuk mengetahui berapa tingkat efektifitasnya. Oleh sebab itu diperlukan suatu pengukuran, yaitu akurasi, presisi dan *recall* (Deolika, Kusrini, & Luthfi, 2019). Pengukuran ini sering disebut dengan *confusion matrix*. Pada *confusion matrix*, data pada kolom menunjukkan data yang diharapkan, sedangkan data pada baris menunjukkan data yang diprediksi (Dhande & Patnaik, 2014). Untuk mendapatkan akurasi, presisi dan *recall* pada proses prediksi dengan label lebih dari 2 (dua) menggunakan rata-rata (Sokolova & Lapalme, 2009). Perhitungan akurasi, presisi dan *recall* dapat dilihat pada persamaan 7, 8 dan 9.

$$Akurasi = \frac{\sum_{i=1}^{c} \frac{TPi + TNi}{TPi + TNi + FPi + FNi}}{c} * 100\% \quad (7)$$

$$Presisi = \frac{\sum_{i=1}^{c} TPi}{\sum_{i=1}^{c} (FPi + TPi)} * 100\% \quad (8)$$

$$Recall = \frac{\sum_{i=1}^{c} TPi}{\sum_{i=1}^{c} (TPi + FNi)} * 100\% \quad (9)$$

Di mana:

- c = jumlah label
- TPi = jumlah data positif yang diklasifikasikan benar oleh sistem pada label ke i

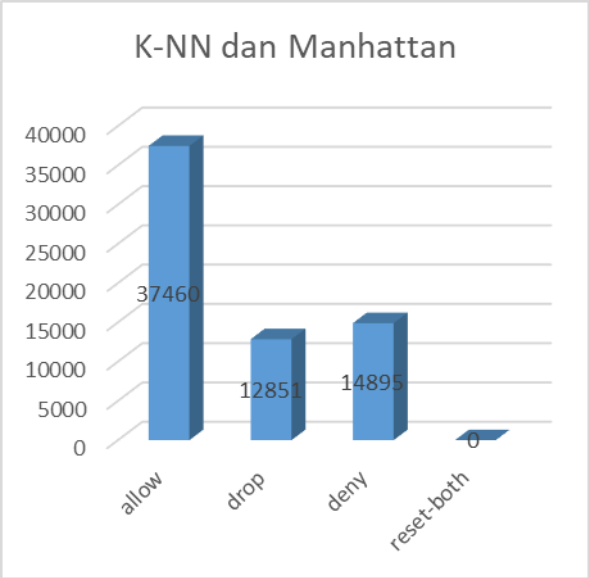
- TNi = jumlah data negatif yang diklasifikasikan benar oleh sistem pada label ke i
- FNi = jumlah data negatif yang diklasifikasikan salah oleh sistem pada label ke i
- FPi = jumlah data positif yang diklasifikasikan salah oleh sistem pada label ke i

3. Hasil dan Pembahasan

Tabel 2. Hasil pengukuran prediksi $k = 5$

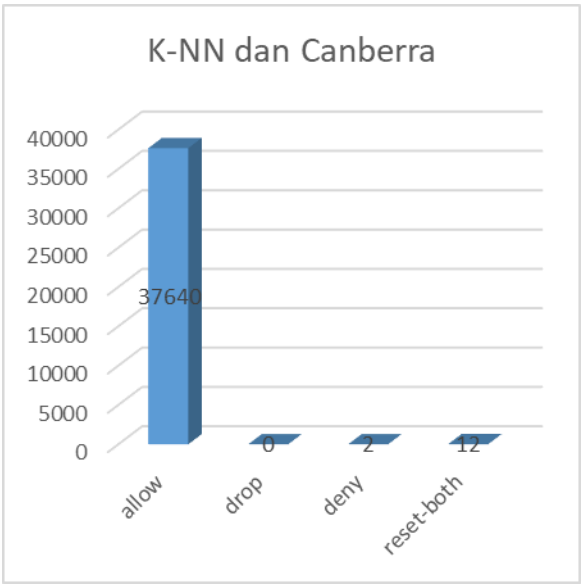
Model	Akurasi	Presisi	Recall
K-NN dengan Manhattan	99,50%	74.50%	74.73%
K-NN dengan Canberra	57.46%	39.36%	30.59%
K-NN dengan Cosine Similarity	**99.71%**	**74.70%**	**74.85%**
K-NN dengan Euclidean Distance	99.40%	74.43%	74.65%
K-NN dengan Jaccard	45.71%	17.21%	54.29%

Tabel 2 di atas memperlihatkan hasil pengukuran prediksi trafik serangan pada *firewall router*. Didapatkan bahwa nilai *k* terbaik adalah 5. Hal ini juga sesuai dengan penelitian sebelumnya yang mengatakan bahwa jumlah *k* berkebalikan dengan jumlah label atau kelasnya (Wahyono et al., 2020). Model terbaik dihasilkan oleh *K-NN* dengan menggunakan *Cosine Similarity* dengan nilai akurasi sebesar 99.71%, presisi sebesar 74.70% dan recall sebesar 74.85%. Performa terbaik kedua diperoleh dari model *K-NN* dengan *Manhattan* diikuti *K-NN* dengan *Euclidean Distance* diurutan ketiga, kemudian *K-NN* dengan *Canberra* dan *Jaccard* diurutan keempat dan kelima. *K-NN* dengan *Canberra* dan *Jaccard* memiliki performa yang rendah, baik dari sisi akurasi, presisi dan recall, berbeda dengan tiga pendekatan penghitungan jarak yang lain yang memiliki selisih tidak terlalu banyak.



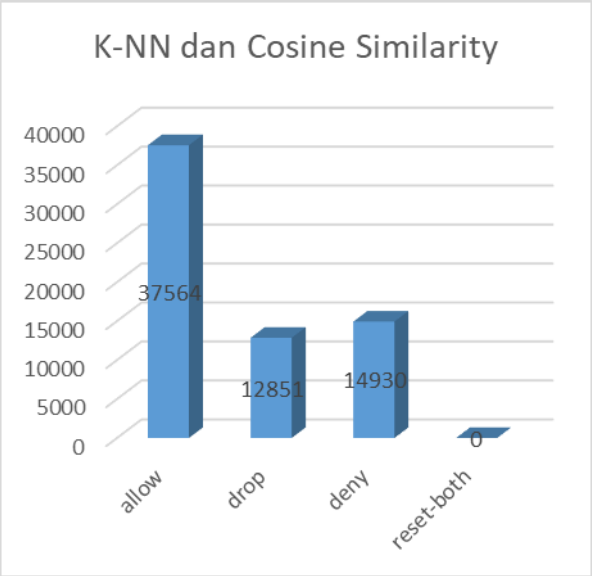
Gambar 2. Grafik sebaran Prediksi *K-NN* dan *Manhattan*

Gambar 2 menunjukkan sebaran data prediksi serangan *firewall* menggunakan *K-NN* dan *Manhattan* dengan aksi *allow* sebanyak 37.460 data *true*, aksi *drop* sebanyak 12.851 data *true*, aksi *deny* sebanyak 14.895 data *true* dan aksi *reset-both* sebanyak 0 data *true*.



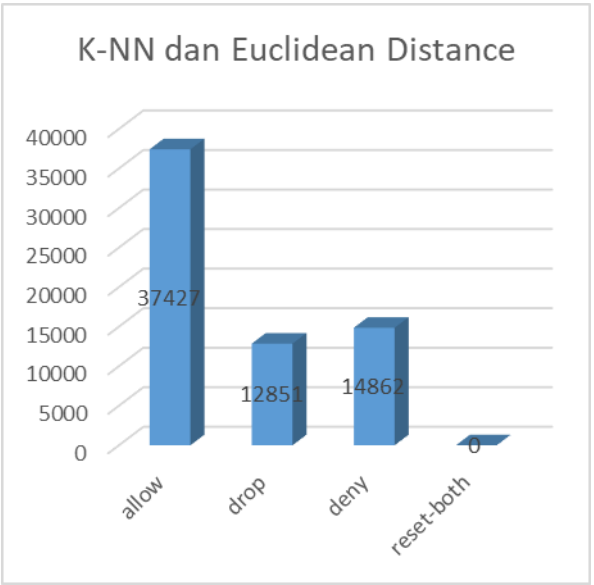
Gambar 3. Grafik sebaran prediksi *K-NN* dan *Canberra*

Gambar 3 menunjukkan sebaran data prediksi serangan *firewall* menggunakan *K-NN* dan *Canberra* dengan aksi *allow* sebanyak 37.640 data *true*, aksi *drop* sebanyak 0 data *true*, aksi *deny* sebanyak 2 data *true* dan aksi *reset-both* sebanyak 12 data *true*.



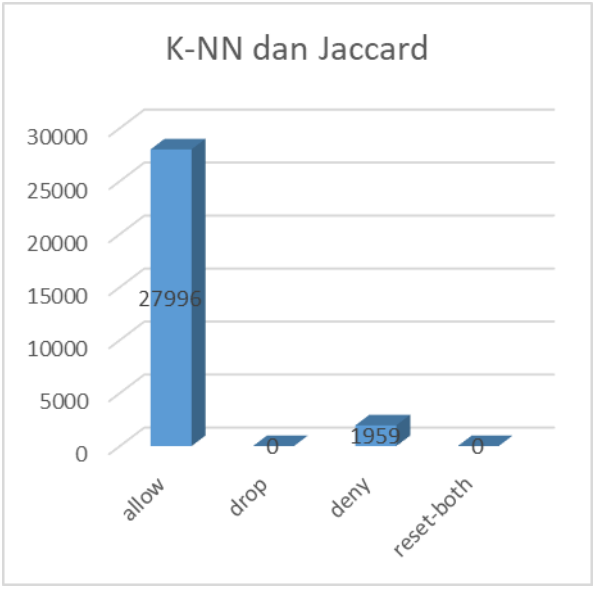
Gambar 4. Grafik sebaran prediksi *K-NN* dan *Cosine Similarity*

Gambar 4 menunjukkan sebaran data prediksi serangan *firewall* menggunakan *K-NN* dan *Cosine Similarity* dengan aksi *allow* sebanyak 37.564 data *true*, aksi *drop* sebanyak 12.851 data *true*, aksi *deny* sebanyak 14.930 data *true* dan aksi *reset-both* sebanyak 0 data *true*.



Gambar 5. Grafik sebaran prediksi *K-NN* dan *Euclidean Distance*

Gambar 5 menunjukkan sebaran data prediksi serangan *firewall* menggunakan *K-NN* dan *Euclidean Distance* dengan aksi *allow* sebanyak 37.427 data *true*, aksi *drop* sebanyak 12.851 data *true*, aksi *deny* sebanyak 14.862 data *true* dan aksi *reset-both* sebanyak 0 data *true*.



Gambar 6. Grafik sebaran prediksi *K-NN* dan *Jaccard*

Gambar 6 menunjukkan sebaran data prediksi serangan *firewall* menggunakan *K-NN* dan *Jaccard* dengan aksi *allow* sebanyak 27.996 data *true*, aksi *drop* sebanyak 0 data *true*, aksi *deny* sebanyak 1.959 data *true* dan aksi *reset-both* sebanyak 0 data *true*.

Pada pengukuran jarak menggunakan metode *Manhattan*, *Cosine Similarity* dan *Euclidean Distance* memiliki persamaan rumus yang mirip, sehingga menghasilkan prediksi yang tidak terlalu jauh berbeda. *Jaccard* juga memiliki kemiripan dari sisi persamaan namun memiliki hasil performa yang berbeda jauh. Sedangkan *Canberra* memiliki perbedaan persamaan dan terlihat juga performa yang juga sangat berbeda dibandingkan metode yang lainnya.

4. Kesimpulan

Prediksi serangan pada trafik internet ke *firewall* dapat dilakukan dengan beberapa teknik data mining seperti *K-Nearest Neighbor* dengan penghitungan jarak seperti *Manhattan Distance*, *Canberra*, *Cosine Similarity*, *Euclidean Distance* dan *Jaccard*. Proses prediksi diawali dengan tahap *pre-processing* dilanjutkan dengan training dan testing dataset menggunakan proses perulangan untuk mencari nilai *k* optimal. Selanjutnya dilakukan penghitungan jarak menggunakan masing-masing metode untuk kemudian dibandingkan performa terbaiknya

Setelah dilakukan penelitian dan uji coba, didapatkan bahwa model *K-NN* dengan penghitung jarak *Cosine Similarity* dapat menghasilkan performa terbaik, yaitu dengan akurasi sebesar 99.71%, presisi sebesar 74.70% dan recall sebesar

74.85%. Diurutan kedua dan selanjutnya masing-masing diperoleh dari metode *K-NN* dengan *Manhattan*, *Euclidean Distance*, *Canberra* dan *Jaccard*. Dengan hasil penelitian ini diharapkan dapat dijadikan acuan bagi peneliti lain yang ingin mengembangkan prediksi serangan pada *firewall* sebuah *router*. Selain itu, berdasarkan hasil performa prediksi yang sangat tinggi dapat dijadikan acuan untuk membuat sebuah *firewall* tambahan yang cerdas, sehingga keamanan data pengguna terjaga dengan baik. Dengan keamanan data pengguna yang terjaga, dapat menghindarkan mereka dari kerugian baik yang bersifat materi maupun non materi.

Referensi

Chen, Z., Zhou, L. J., Li, X. Da, Zhang, J. N., & Huo, W. J. (2020). The Lao text classification method based on KNN. *Procedia Computer Science*, Vol. 166, hal. 523–528. Elsevier B.V. https://doi.org/10.1016/j.procs.2020.02.053

Chicco, D., & Jurman, G. (2020). Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. *BMC Medical Informatics and Decision Making*, *20*(1), 1–16. https://doi.org/10.1186/s12911-020-1023-5

Chiche, A., & Meshesha, M. (2021). Towards a Scalable and Adaptive Learning Approach for Network Intrusion Detection. *Journal of Computer Networks and Communications*, *2021*. https://doi.org/10.1155/2021/8845540

Deolika, A., Kusrini, K., & Luthfi, E. T. (2019). Analisis Pembobotan Kata Pada Klasifikasi Text Mining. *Jurnal Teknologi Informasi*, *3*(2), 179. https://doi.org/10.36294/jurti.v3i2.1077

Dewi, R. F. K., Obert, & Gusmana, R. (2018). Implementasi Metode K-Nearest Neighbor (KNN) dalam Pengelompokan Status Ekonomi Warga. *Journal of Big Data Analytic and Artificial Intelligence*, *4*(1), 15–22.

Dhande, L. L., & Patnaik, P. G. K. (2014). Analyzing Sentiment of Movie Review Data using Naive Bayes Neural Classifier. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, *3*(4), 313–320. Diambil dari www.ijettcs.org

Dinata, R. K., Akbar, H., & Hasdyna, N. (2020). Algoritma K-Nearest Neighbor dengan Euclidean Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus. *ILKOM Jurnal Ilmiah*, *12*(2), 104–111. https://doi.org/10.33096/ilkom.v12i2.539.104-111

Ertam, F., & Kaya, M. (2018). Classification of firewall log files with multiclass support vector machine. *6th International Symposium on Digital Forensic and Security, ISDFS 2018 - Proceeding*, *2018-Janua*(2), 1–4. https://doi.org/10.1109/ISDFS.2018.8355382

Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., & Alhasanat, A. A. (2014). Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach. *International Journal of Computer Science and Information Security*, *12*(8), 33–39. Diambil dari http://arxiv.org/abs/1409.0919

Khosroshahi, A. H., & Shahinzadeh, H. (2016). Security Technology by using Firewall for Smart Grid. *Bulletin of Electrical Engineering and Informatics*, *5*(3), 366–372. https://doi.org/10.11591/eei.v5i3.545

Latifah, K. (2015). Kombinasi Algorithma K-NN dan Manhattan Distance untuk Menentukan Pemenang Lelang. *Jurnal Informatika Upgris (JIU)*, *1*, 49–58. Diambil dari https://docplayer.info/34038947-Kombinasi-algorithma-k-nn-dan-manhattan-distance-untuk-menentukan-pemenang-lelang.html

Mujtaba, G., Shuib, L., Idris, N., Hoo, W. L., Raj, R. G., Khowaja, K., … Nweke, H. F. (2019). Clinical text classification research trends: Systematic literature review and open issues. *Expert Systems with Applications*, *116*, 494–520. https://doi.org/10.1016/j.eswa.2018.09.034

Nazari, N., & Mahdavi, M. (2018). A survey on Automatic Text Summarization. *Journal of AI and Data Mining*, *0*(0), 121–135. https://doi.org/10.22044/jadm.2018.6139.1726

Nurdiana, O., Jumadi, J., & Nursantika, D. (2016). Perbandingan Metode Cosine Similarity Dengan Metode Jaccard Similarity Pada Aplikasi Pencarian Terjemah Al-Qur'an Dalam Bahasa Indonesia. *Jurnal Online Informatika*, *1*(1), 59–63. https://doi.org/10.15575/join.v1i1.12

Rivki, Muhammad; Bachtiar, A. M. (2017). Implementasi Algoritma K-Nearest Neighbor Dalam Pengklasifikasian Follower Twitter Yang Menggunakan Bahasa Indonesia. *Jurnal Sistem Informasi (Journal of Information System)*, *13*(1), 31–37.

Saleh, A. I., Rabie, A. H., & Abo-Al-Ez, K. M. (2016). A data mining based load forecasting strategy for smart electrical grids. *Advanced Engineering Informatics*, *30*(3), 422–448. https://doi.org/10.1016/j.aei.2016.05.005

Samuel, R., Natan, R., & Syafiqoh, U. (2018). Penerapan Cosine Similarity dan K-Nearest Neighbor (K-NN) pada Klasifikasi dan Pencarian Buku. *Journal of Big Data Analytic and Artificial Intelligence*, *1*(1), 9–14.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, *45*(4), 427–437. https://doi.org/10.1016/j.ipm.2009.03.002

Wahono, R. S., Herman, N. S., & Ahmad, S. (2014). A comparison framework of classification models for software defect prediction. *Advanced Science Letters*, *20*(10–12), 1945–1950. https://doi.org/10.1166/asl.2014.5640

Wahyono, W., Trisna, I. N. P., Sariwening, S. L., Fajar, M., & Wijayanto, D. (2020). Comparison of distance measurement on k-nearest neighbour in textual data classification. *Jurnal Teknologi dan Sistem Komputer*, *8*(1), 54–58. https://doi.org/10.14710/jtsiskom.8.1.2020.54-58

Wurdianarto, S., Wurdianarto, S. R., Novianto, S., & Rosyidah, U. (2014). Perbandingan Euclidean Distance Dengan Canberra Distance Pada Face Recognition. *Techno.Com*, *13*(1), 31–37. Diambil dari https://publikasi.dinus.ac.id/index.php/technoc/article/view/539