# Klasifikasi Serangan Jaringan Menggunakan Metode *K-Nearest Neighbour* pada Data Riwayat Jaringan

*Dimas Surya Prasetyo [1)], Karina Auliasari [2)], Muhammad Ridho Putra Syalabi [3)]*

*[1),2),3 )] Teknik Informatika S1, Institut Teknologi Nasional Malang*
*Jl. Sigura-gura 2 Malang*
*Email : diimprasetyos@gmail.com*

**Abstrak.** *Metode deteksi serangan yang lebih pintar dan fleksibel diperlukan karena serangan jaringan modern semakin kompleks. Pola lalu lintas, frekuensi koneksi, dan protokol yang digunakan digunakan dalam metode K-Nearest Neighbour (KNN) dalam klasifikasi serangan jaringan berdasarkan data riwayat jaringan. Pola serangan dapat diidentifikasi dengan model KNN yang efisien, yang membedakannya dari lalu lintas jaringan biasa. Pada penelitian ini dilakukan prapemrosesan data untuk penelitian ini, yang mencakup normalisasi data, ekstraksi atribut yang relevan, korelasi antar atribut hingga menerapkan algoritma KNN pada dataset yang telah dipersiapkan setelah itu. Untuk mengukur kinerja model, pengujian dilakukan menggunakan metrik evaluasi standar seperti akurasi, presisi, dan recall. Hasil eksperimen menunjukkan bahwa metode KNN didapatkan dengan mendeteksi serangan jaringan dengan sangat akurat dan merespons secara efisien terhadap berbagai jenis serangan.*

***Katakunci:*** *Klasifikasi, Serangan, Jaringan, K-NN, Riwayat.*

## 1. Pendahuluan

Tantangan keamanan jaringan menjadi semakin penting di era telekomunikasi yang berkembang pesat. Serangan jaringan telah menjadi ancaman besar terhadap integritas, kerahasiaan, dan ketersediaan sistem komputer, jaringan, dan infrastruktur komunikasi. Upaya yang tidak sah atau jahat untuk memanipulasi, merusak, atau mengakses data dan sumber daya komputer tanpa izin dikenal sebagai serangan jaringan. Di antara konsekuensi yang signifikan termasuk ancaman terhadap informasi rahasia, kehilangan uang, dan kemungkinan gangguan terhadap layanan penting [1]. Serangan jaringan dapat menjadi pintu masuk untuk berbagai ancaman lain, seperti pencurian data sensitif, peretasan sistem, dan pencurian identitas. Serangan jaringan dapat merugikan pengguna jaringan dan menghancurkan aset informasi pemilik. Pendeteksian dini sangat penting untuk melindungi infrastruktur jaringan yang kompleks dan penting dalam menghadapi ancaman serangan jaringan yang semakin meningkat [2].

Beberapa penelitian terkait dengan topik ini telah memberikan wawasan yang berharga dalam pengembangan deteksi serangan yang lebih cerdas dan responsif. Seperti penelitian yang dilakukan oleh Buzcak dan Guven di tahun 2016 mengulas berbagai metode dalam sistem deteksi intrusi jaringan (NIDS), dengan penekanan pada klasifikasi berbasis data riwayat. Menyoroti tantangan dalam mengenali serangan yang berkembang, penelitian ini mengidentifikasi pentingnya memanfaatkan data riwayat jaringan yang kaya akan informasi untuk meningkatkan akurasi dan respons sistem deteksi [2]. Penelitian serupa juga dilakukan di tahun berikutnya oleh Jeong et. al. yang berfokus pada teknik klasifikasi, penelitian ini mencakup penggunaan data riwayat jaringan sebagai atribut utama dan menyoroti keefektifan metode ini dalam mengklasifikasikan berbagai serangan jaringan [3]. Penelitian Uhongora et. al. di tahun 2023 meneliti pendekatan jaringan saraf dalam deteksi intrusi, termasuk penggunaan data riwayat jaringan sebagai input. Menyoroti potensi jaringan saraf dalam menangani data yang kompleks, penelitian ini memberikan gambaran tentang perkembangan terbaru dalam bidang ini [4]. Seperti yang ditunjukkan oleh analisis literatur tersebut, semua orang setuju bahwa data riwayat jaringan adalah sumber informasi penting untuk klasifikasi serangan jaringan. Dengan tujuan meningkatkan akurasi dan respons sistem deteksi serangan jaringan, penelitian ini akan melanjutkan topik penelitian klasifikasi serangan jaringan dengan menerapkan metode K-Nearest Neighbour pada dataset riwayat jaringan. Dimana dalam penelitian ini metode K-Nearest Neighbour (KNN) dapat digunakan untuk klasifikasi serangan jaringan yang didasarkan pada data riwayat jaringan. KNN

adalah algoritma pembelajaran mesin non-parametrik yang dapat digunakan baik untuk regresi maupun klasifikasi; dalam hal deteksi serangan jaringan, KNN memungkinkan sistem untuk mengenali pola serangan berdasarkan seberapa dekat mereka dengan pola yang sudah ada dalam data riwayat jaringan. Mengembangkan model KNN yang dapat diandalkan untuk mendeteksi berbagai jenis serangan jaringan adalah tujuan utama penelitian ini. Dari penelitian ini diharapkan dapat ditemukan fitur penting yang membedakan serangan dari lalu lintas jaringan biasa dengan menganalisis data riwayat jaringan dan dapat membantu dalam pengembangan sistem deteksi serangan jaringan yang lebih pintar dan responsif.

### 1.1. *Metode K-Nearest Neighbour*

*K-Nearest Neighbor* merupakan salah satu algoritma klasifikasi Supervised learning, yang bertujuan untuk mengklasifikasikan objek baru berdasarkan atribut atau ciri dari sampel data latih yang sudah ada pada sistem, menggunakan jarak ketetanggaan terdekat dari data uji ke data latih [5]. Berikut merupakan langkah-langkah perhitungan algoritma K-NN [5]:

a) Menentukan nilai K atau tentangga terdekat dari data latih terhadap data uji.
b) Menghitung jarak dengan *Euclidean* distance masing - masing objek terhadap data latih yang diberikan.
c) Kemudian mengurutkan objek-objek tersebut dari yang terkecil ke terbesar.
d) Mengelompokkan data sejumlah K yang telah ditentukan sebelumnya.
e) Memilih kategori yang paling mayoritas atau yang paling banyak muncul.

Algoritma K-Nearest Neighbor pada penelitian ini menggunakan salah satu perhitungan jarak yaitu, jarak *Euclidean Distance*, karena perhitungan jarak *Euclidean* yang paling umum digunakan pada data yang berbentuk numerik.

## 2. Pembahasan

Beberapa tahapan dalam penelitian ini dilakukan untuk menemukan hubungan antara elemen-elemen dalam kumpulan data serangan jaringan. Tahapan pertama adalah menyiapkan data yang dilakukan untuk membersihkan data seperti menghilangkan nilai yang hilang dan mengonversi atribut kategori menjadi data numerik. Kemudian dilakukan seleksi fitur untuk memilih subset atribut yang paling relevan untuk mengklasifikasikan serangan jaringan. Setelah data siap dilanjutkan ke tahap klasifikasi menggunakan algoritma K-*Nearest Neighbor* (KNN) untuk mengidentifikasi pola dan hubungan antara atribut dalam dataset. Hasil klasifikasi dievaluasi menggunakan metrik seperti nilai akurasi, recall, dan skor f1. Keseluruhan tahapan pada penelitian ini dilakukan menggunakan bahasa pemrograman *python*.

### 2.1. Pra-pemrosesan Data

Dataset yang digunakan dalam penelitian ini memiliki 42 atribut data dengan 15.800 baris data. Tiap atribut dataset mulai nama atribut, tipe data atribut dan deskripsinya ditunjukkan pada Tabel 2.

Tabel 2. Atribut dataset riwayat jaringan

| Nama atribut data | Tipe data | Deskripsi |
|---|---|---|
| Duration | float | Lamanya waktu (dalam detik) koneksi. |
| Protocol_type | object | Jenis protokol koneksi. |
| Service | object | Nama layanan jaringan pada host target, seperti HTTP, FTP, dll. |
| Flag | object | Nama layanan jaringan pada host target, seperti HTTP, FTP, dll. |
| Src_bytes | float | Jumlah byte yang ditransfer dari sumber ke tujuan. |
| Dst_bytes | float | Jumlah byte yang ditransfer dari tujuan ke sumber. |

| Nama atribut data | Tipe data | Deskripsi |
|---|---|---|
| Land | float | Jumlah byte yang ditransfer dari tujuan ke sumber. |
| Wrong_fragment | float | Jumlah fragmen yang "salah" dalam suatu koneksi. |
| Urgent | float | Jumlah paket mendesak dalam suatu koneksi. |
| Hot | float | Jumlah indikator "hot" dalam sambungan. |
| Num_failed_logins | float | Jumlah indikator "panas" dalam sambungan. |
| Logged_in | float | Menunjukkan apakah login berhasil dilakukan (1 jika ya, 0 jika tidak). |
| Num_compromised | float | Jumlah host yang dikompromikan melalui koneksi. |
| Root_shell | float | Menunjukkan apakah shell root telah diperoleh (1 jika ya, 0 jika tidak). |
| Su_attempted | float | Menunjukkan apakah perintah "su root" telah dicoba (1 jika ya, 0 jika tidak). |
| Num_root | float | Jumlah akses root yang diperoleh. |
| Num_file_creations | float | Jumlah operasi pembuatan file. |
| Num_shells | float | Jumlah perintah shell. |
| Num_access_files | float | Jumlah operasi pada file kontrol akses. |
| Num_outbound_cmds | float | Jumlah perintah keluar dalam sesi FTP. |
| Is_hot_login | float | Menunjukkan apakah login termasuk dalam daftar "panas" (1 jika ya, 0 jika tidak). |
| Is_guest_login | float | Menunjukkan apakah login tersebut merupakan login tamu (1 jika ya, 0 jika tidak). |
| Count | float | Jumlah koneksi ke host yang sama dengan koneksi saat ini dalam dua detik terakhir. |
| Srv_count | float | Jumlah sambungan ke layanan yang sama dengan sambungan saat ini dalam dua detik terakhir. |
| Serror_rate | float | Persentase koneksi yang mengalami kesalahan "SYN". (SYN: SINKRONISASI) |
| Srv_serror_rate | float | Persentase koneksi ke layanan yang sama yang mengalami kesalahan "SYN". |
| Rerror_rate | float | Persentase koneksi yang mengalami kesalahan "REJ". (REJ : PENOLAKAN) |
| Srv_rerror_rate | float | Persentase koneksi ke layanan yang sama yang mengalami kesalahan "REJ". |
| Same_srv_rate | float | Persentase koneksi ke layanan yang sama. |
| Diff_srv_rate | float | Persentase koneksi ke layanan berbeda. |
| Srv_diff_host_rate | float | Persentase koneksi ke host yang berbeda. |
| Dst_host_count | float | Jumlah koneksi ke host tujuan yang sama dalam dua detik terakhir. |
| Dst_host_srv_count | float | Jumlah koneksi ke layanan tujuan yang sama dalam dua detik terakhir. |
| dst_host_same_srv_rate | float | Persentase koneksi ke layanan tujuan yang sama. |
| dst_host_diff_srv_rate | float | Persentase koneksi ke layanan tujuan yang berbeda. |
| dst_host_same_src_port_rate | float | Persentase koneksi ke host tujuan yang sama yang berasal dari port sumber yang sama. |
| dst_host_srv_diff_host_rate | float | Persentase koneksi ke layanan tujuan yang sama namun berasal dari host berbeda. |
| dst_host_serror_rate | float | Persentase koneksi ke host tujuan tertentu yang |

| Nama atribut data | Tipe data | Deskripsi |
|---|---|---|
| | | mengalami kesalahan “SYN”. |
| dst_host_srv_serror_rate | float | Persentase koneksi ke layanan tujuan tertentu yang mengalami kesalahan “SYN”. |
| dst_host_rerror_rate | float | Persentase koneksi ke host tujuan tertentu yang mengalami kesalahan “REJ”. |
| dst_host_srv_rerror_rate | float | Persentase koneksi ke layanan tujuan tertentu yang memiliki kesalahan ”REJ” |
| type_of_attack | object | Jenis serangan. |

Dalam langkah awal pemrosesan data, dilakukan pembersihan data untuk memastikan bahwa data dalam kondisi yang baik untuk analisis lebih lanjut. Berikut langkah-langkah yang diambil dalam tahap pembersihan data:

a) Ukuran Data Awal: Dataset awalnya memiliki bentuk (*shape*) yang dapat diidentifikasi menggunakan *df.shape*. Menggunakan fungsi ini didapatkan informasi sejumlah baris dan kolom.

b) Informasi Data: Melalui *df.info()*, didapatkan informasi tentang tipe data setiap kolom dan apakah ada nilai yang hilang (missing values).

c) Pengecekan *Missing Values*: Untuk memastikan bahwa tidak ada data yang hilang, digunakan *df.isnull()*, sedangkan.*sum()* merupakan jumlah nilai yang hilang dalam setiap kolom akan ditampilkan.

d) *Total Missing Values*: Total keseluruhan nilai yang hilang dalam dataset dihitung dengan *df.isnull().sum().sum()*. Hasilnya adalah 0, yang menunjukkan bahwa tidak ada nilai yang hilang dalam dataset.

e) Penghapusan Baris dengan Nilai Hilang: Meskipun telah diperiksa dan ditemukan bahwa dataset tidak memiliki nilai yang hilang, namun harus tetap dilakukan langkah pencegahan dengan menjalankan *df.dropna(axis=0, inplace=True)*. Hal ini dilakukan untuk memastikan bahwa baris yang mengandung nilai yang hilang, akan dihapus dari dataset.

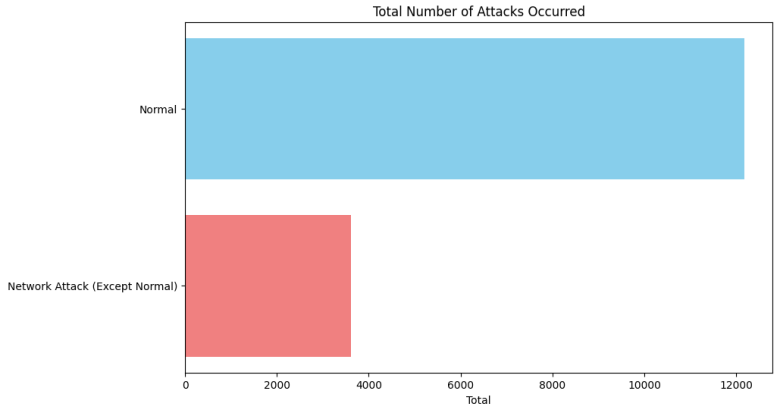
f) Pengecekan *Missing Values* tahap akhir: Setelah penghapusan nilai yang hilang, kembali dilakukan pemeeriksaan apakah ada nilai yang hilang dalam kolom '*type_of_attack*' dengan menggunakan *df['type_of_attack'].isnull().sum()*. Hasilnya juga menunjukkan bahwa tidak ada nilai yang hilang dalam kolom ini.

Langkah selanjutnya dalam pra-pemrosesan data, dilakukan konversi atribut kategori *'type_of_attack'*, *'service'*, *'protocol_type'* dan *'flag'* menjadi bentuk yang sesuai untuk digunakan dalam analisis lebih lanjut, dimana di penelitian ini atribut kategri diubah menjadi bentuk numerik.

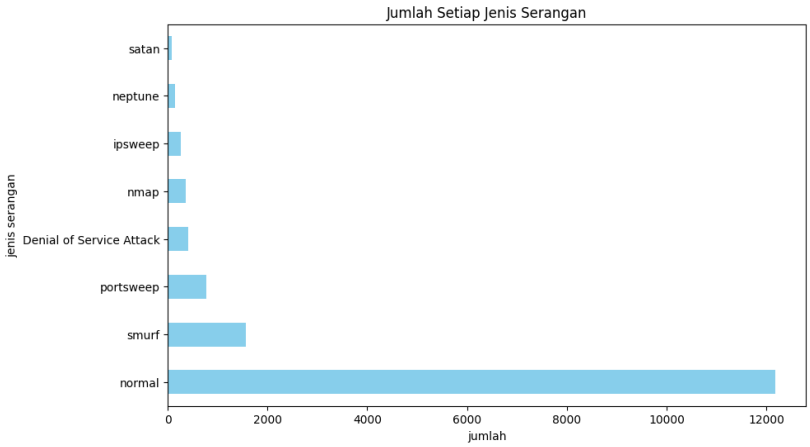
### 2.2. Visualisasi dan Korelasi Data

Visualisasi data dilakukan dalam pemrosesan data yang membantu memahami, mengidentifikasi pola, dan mendapatkan wawasan dari dataset riwayat jaringan. Dari 15.800 baris data riwayat jaringan jika divisualisasikan ada 12.184 baris data yang masuk ke dalam kondisi "normal," yang mencerminkan operasi jaringan dalam keadaan biasa tanpa serangan. Sedangkan sisanya ada 3.616 baris data yang masuk ke dalam kategori "serangan" yang menunjukkan jaringan sedang diserang hal ini seperti yang ditunjukkan pada Gambar1. Jika 3.616 data serangan diperincikan lagi berdasarkan jenis serangannya maka ditunjukkan pada Gambar 2.

SENIATI 2023
ISSN 2085-4218
ITN Malang, 9 Desember 2023




Gambar 1. Grafik visualisasi pengelompokan normal dan serangan pada data riwayat jaringan

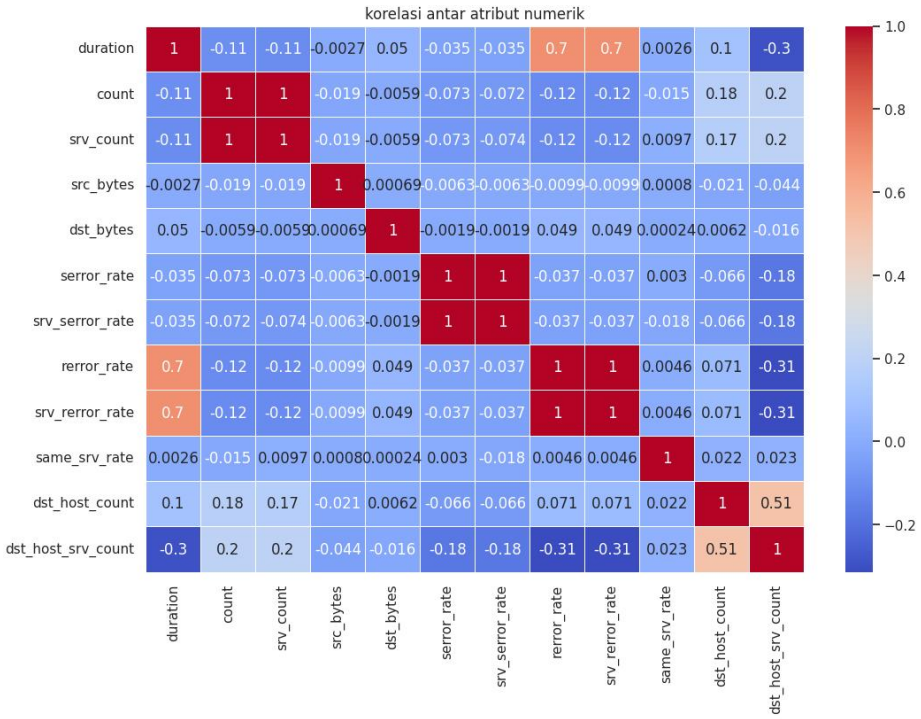



Gambar 2. Grafik visualisasi serangan berdasarkan jenisnya

Pada Gambar 2 terdapat 1.565 (sekitar 9.9%) serangan tipe Smurf, yang umumnya berhubungan dengan serangan Denial of Service (DoS) yang membanjiri target dengan permintaan ICMP Echo Request. Port Sweep: Data mencatat 773 (sekitar 4.9%) serangan jenis Port Sweep, yang mencoba mengakses sejumlah besar port jaringan pada tujuan tertentu untuk mencari kerentanannya. Denial of Service Attack: Terdapat 416 (sekitar 2.6%) serangan DoS Attack, yang bertujuan membuat sumber daya jaringan tidak tersedia untuk pengguna yang sah. Nmap: Ada 367 (sekitar 2.3%) serangan Nmap, yang mungkin mencerminkan upaya pemindaian jaringan oleh perangkat lunak Nmap. IP Sweep: 255 (sekitar 1.6%) serangan tipe IP Sweep yang mencoba mengidentifikasi alamat IP yang aktif dalam jaringan. Neptune: 154 (kurang lebih 1%) serangan tipe Neptune, yang termasuk dalam serangan DoS yang bertujuan menghabiskan sumber daya jaringan yang tersedia pada target. Satan: Terdapat 86 (kurang lebih 0.5%) serangan Satan, yang mencoba mengidentifikasi kerentanan di sistem target.

Selanjutnya adalah pengujian hubungan atau korelasi antara setiap atribut numerik seperti yang ditunjukan pada Gambar 3. Korelasi Positif Kuat, Atribut 'srv_count' dan 'count' memiliki korelasi positif sangat kuat dengan nilai sekitar 0.9998. Ini menunjukkan bahwa jumlah koneksi yang ada dalam satu detik ('count') sangat berkorelasi dengan jumlah koneksi yang memiliki layanan aktif ('srv_count'). Korelasi Negatif Kuat, Atribut 'rerror_rate' dan 'srv_rerror_rate' memiliki korelasi negatif sangat kuat dengan nilai sekitar -1. Ini mengindikasikan bahwa tingkat kesalahan yang terkait dengan koneksi yang diterima ('rerror_rate') sangat berkorelasi dengan tingkat kesalahan pada layanan koneksi yang diterima ('srv_rerror_rate'). Korelasi Positif Antara 'duration' dan 'rerror_rate', Atribut 'duration' memiliki korelasi positif yang cukup kuat dengan 'rerror_rate' sekitar 0.9406. Hal ini mengindikasikan bahwa semakin lama durasi koneksi, semakin tinggi kemungkinan terjadi kesalahan pada koneksi tersebut. Korelasi Negatif dengan 'dst_host_count' dan 'dst_host_srv_count', Atribut 'dst_host_count' memiliki korelasi negatif dengan 'dst_host_srv_count' sekitar -0.6458. Ini menunjukkan bahwa semakin banyak tuan rumah (host) yang terdeteksi dalam jaringan, semakin sedikit host yang memberikan layanan aktif. Korelasi Positif antara 'same_srv_rate' dan 'dst_host_count', Atribut

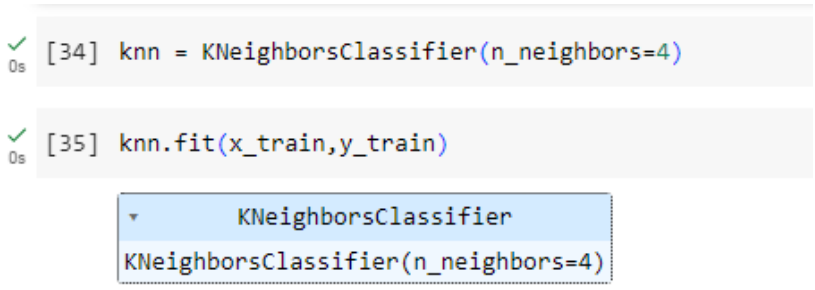
'same_srv_rate' memiliki korelasi positif dengan 'dst_host_count' sekitar 0.7253. Hal ini mengindikasikan bahwa semakin banyak host dengan tingkat layanan yang sama yang terdeteksi, semakin banyak host yang ada dalam jaringan.




Gambar 3. Grafik korelasi antar atribut yang bernilai numerik

**2.3. Penerapan metode K-Nearest Neighbor (KNN)**

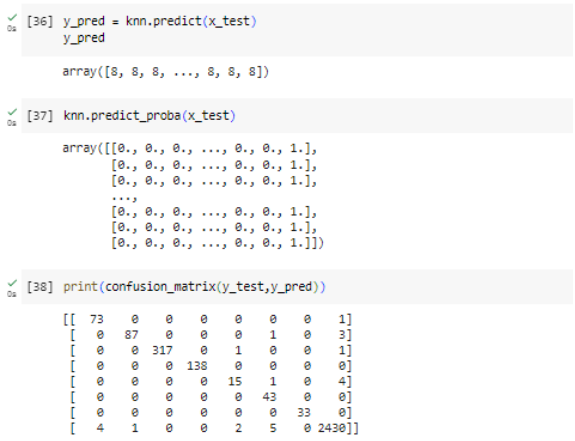
Setelah dilakukan korelasi antara atribut pada dataset, langkah selanjutnya adalah menentukan variabel dependen dan variabel independen. Variabel independen (X) telah ditentukan dengan menghapus kolom 'type_of_attack_num' dari dataset, sehingga X berisi atribut-atribut lain yang akan digunakan dalam proses pemodelan. Sementara itu, variabel dependen (Y) adalah kolom 'type_of_attack_num' yang menyimpan label numerik yang mengidentifikasi jenis serangan. Dengan variabel independen dan dependen yang telah ditentukan seperti yang ditunjukkan pada Gambar 4, selanjutnya akan dipersiapkan data untuk proses menggunakan metode *K-Nearest Neighbor* (KNN).




Gambar 4. Inisialisasi model KNN

Setelah model *K-Nearest* Neighbor (KNN) diinisialisasi dengan parameter-parameter yang sesuai, langkah selanjutnya adalah menentukan prediksi dari model ini. Dalam konteks deteksi serangan jaringan, prediksi akan memberikan informasi tentang jenis serangan yang kemungkinan terjadi pada suatu titik data. Model KNN akan "memutuskan" jenis serangan dengan menganalisis karakteristik data, membandingkannya dengan tetangga terdekat, dan menentukan kelas yang paling mendominasi. Selain prediksi kelas, model KNN juga memberikan probabilitas dari hasil prediksi seperti yang ditunjukkan pada Gambar 5. Probabilitas ini mencerminkan tingkat keyakinan model terhadap

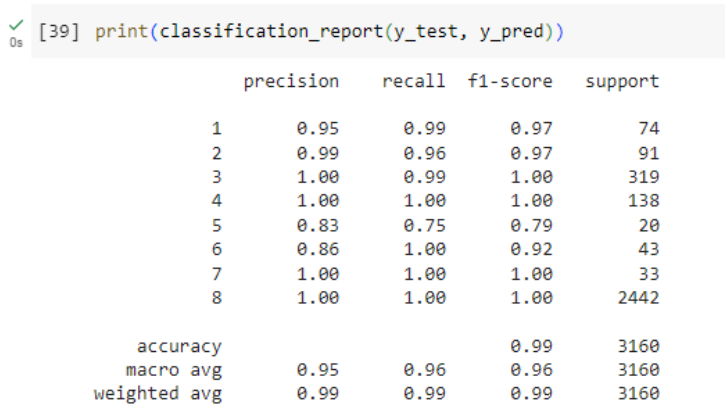
prediksi yang dibuat. Sehingga dapat diketahui sejauh mana model yakin terhadap klasifikasi yang diberikan. Hasil prediksi dan probabilitas ini memiliki dampak besar dalam keamanan jaringan. Hal ini memungkinkan sistem untuk mengidentifikasi serangan secara real-time, memungkinkan respons yang cepat dan tepat terhadap ancaman potensial. Seiring dengan evaluasi kinerja model, ini adalah langkah penting dalam memastikan bahwa jaringan dan sistem tetap aman dari serangan jaringan yang beragam.



```
[36] y_pred = knn.predict(x_test)
     y_pred

     array([8, 8, 8, ..., 8, 8, 8])

[37] knn.predict_proba(x_test)

     array([[0., 0., 0., ..., 0., 0., 1.],
            [0., 0., 0., ..., 0., 0., 1.],
            [0., 0., 0., ..., 0., 0., 1.],
            ...,
            [0., 0., 0., ..., 0., 0., 1.],
            [0., 0., 0., ..., 0., 0., 1.],
            [0., 0., 0., ..., 0., 0., 1.]])

[38] print(confusion_matrix(y_test,y_pred))

     [[  73    0    0    0    0    0    0    1]
      [   0   87    0    0    0    1    0    3]
      [   0    0  317    0    1    0    0    1]
      [   0    0    0  138    0    0    0    0]
      [   0    0    0    0   15    1    0    4]
      [   0    0    0    0    0   43    0    0]
      [   0    0    0    0    0    0   33    0]
      [   4    1    0    0    2    5    0 2430]]
```

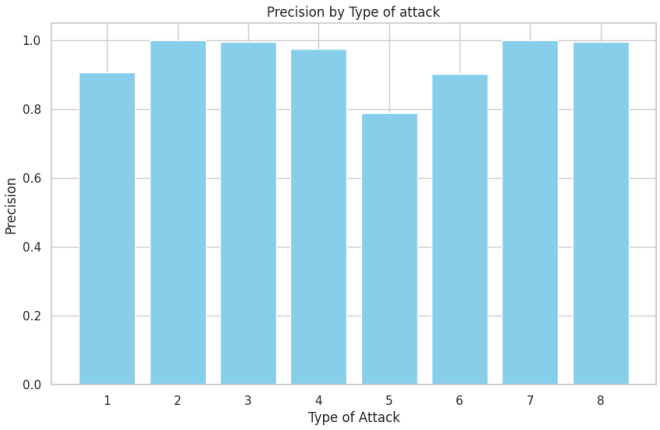
Gambar 5. Prediksi dan pelatihan model KNN

Dari hasil kinerja model KNN yang pertama yaitu memiliki Presisi Tinggi, Presisi mengukur sejauh mana prediksi positif yang dibuat oleh model KNN benar seperti yang ditunjukkan pada Gambar 6. Hasil menunjukkan bahwa model memiliki presisi yang tinggi untuk hampir semua kelas serangan (1-8), dengan skor presisi berkisar antara 0.83 hingga 1.00. Hal ini menandakan bahwa sebagian besar prediksi positif yang dibuat oleh model adalah benar. Recall yang Baik, Recall mengukur sejauh mana model mampu mengidentifikasi kasus positif dengan benar. Model KNN memiliki tingkat recall yang baik untuk semua kelas serangan, mencapai skor yang mendekati 1.00 untuk sebagian besar kelas. Ini berarti model mampu mengidentifikasi sebagian besar kasus serangan dengan benar. F1-Score Tinggi, F1-score adalah pengukuran gabungan antara presisi dan recall. Hasil menunjukkan bahwa model memiliki F1-score yang tinggi untuk semua kelas serangan, mendekati 1.00. Ini menunjukkan bahwa model KNN memiliki keseimbangan yang baik antara presisi dan recall, sehingga cocok untuk tugas klasifikasi serangan jaringan. Akurasi Tinggi, Tingkat akurasi keseluruhan model adalah sekitar 99%, yang merupakan indikasi bahwa model mampu mengklasifikasikan serangan jaringan dengan tingkat keakuratan yang sangat tinggi. Kinerja Rata-rata yang Baik, Rata-rata (macro avg) dari presisi, recall, dan f1-score adalah sekitar 0.95 hingga 0.96, menunjukkan kinerja model yang baik secara keseluruhan. Begitu pula dengan rata-rata tertimbang (weighted avg), yang juga mendekati 0.99.



```
[39] print(classification_report(y_test, y_pred))

                precision    recall  f1-score   support

             1       0.95      0.99      0.97        74
             2       0.99      0.96      0.97        91
             3       1.00      0.99      1.00       319
             4       1.00      1.00      1.00       138
             5       0.83      0.75      0.79        20
             6       0.86      1.00      0.92        43
             7       1.00      1.00      1.00        33
             8       1.00      1.00      1.00      2442

      accuracy                           0.99      3160
     macro avg       0.95      0.96      0.96      3160
  weighted avg       0.99      0.99      0.99      3160
```

Gambar 6. Hasil kinerja dari klasifikasi menggunakan model KNN




Gambar 7 Grafik presisi model KNN

Gambar 7 menunjukkan bahwa model KNN telah berhasil dalam mengklasifikasikan berbagai jenis serangan jaringan dengan tingkat akurasi dan keandalan yang tinggi. Kinerja model ini penting dalam menjaga keamanan jaringan dan sistem komputer, dengan kemampuan untuk mendeteksi serangan dengan tingkat keakuratan yang tinggi. model KNN ini merupakan alat yang kuat dalam upaya melindungi jaringan dan sistem dari ancaman serangan jaringan yang beragam.

## 3. Kesimpulan

Dalam rangka mengatasi ancaman serangan jaringan yang semakin kompleks, pengembangan dan penerapan model K-Nearest Neighbor (KNN) telah membuktikan keberhasilannya. Model ini mencapai tingkat akurasi yang sangat tinggi dan memiliki presisi serta recall yang luar biasa baik dalam mengklasifikasikan berbagai jenis serangan jaringan. Hal ini menandakan kemampuan model dalam mendeteksi serangan dengan tingkat keakuratan yang tinggi, yang sangat penting dalam menjaga keamanan jaringan dan sistem komputer. Kombinasi antara presisi dan recall yang tinggi, ditunjukkan oleh F1-score yang tinggi, memperkuat keyakinan bahwa model ini dapat menjadi alat yang kuat dalam upaya melindungi jaringan dan sistem dari ancaman serangan jaringan yang beragam.

**Ucapan Terima Kasih**

Terimakasih kepada orang tua dan keluarga yang selalu memberikan semangat dan doa dalam setiap langkah penulisan artikel hasil penelitian ini. Terimakasih juga kepada Ibu Karina Auliasari atas bimbingannya dalam penyelesaian proses penelitian hingga penulisan artikel ilmiah ini. Terimakasih juga kepada teman-teman tercinta asisten Laboratorium Pemrograman Teknik Informatika S1 ITN Malang atas dukungan dan memberikan semangat kepada penulis.

## Daftar Pustaka

[1]. Patcha, A., & Park, J. M. (2007). An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. Computer Network Journal. Volume 51, Issue 12, 22 August 2007, Pages 3448-3470. Elsevier.

[2]. Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, Vol. 18, No. 2, Second Quarter 2016.

[3]. Jeong, I. S., Kim, H. K., Kim T. H., Lee D. H., Kim, K. J. and Kang. S.H. A Feature Selection Approach Based on Simulated Annealing for Detecting Various Denial of Service Attacks urnal of Software Networking, 173–190. doi: 10.13052/jsn2445-9739.2016.010. 2016 River Publishers.

[4]. Uhongora, U., Mulinde, R., Law, Y.W. and Slay, J. (2023). Deep-learning-based Intrusion Detection for Software-defined Networking Space Systems. European Conference on Cyber Warfare and Security. 22. 639-647. 10.34190/eccws.22.1.1085.

[5]. Ardiyasa, I. W. (2020). Penerapan K-Means Clustering untuk Klasifikasi Serangan Cyber pada Syslog File. Jurnal Sistem dan Informatika (JSI). p-ISSN: 1858-473X, e-ISSN: 2460-3732, DOI: 10.30864/jsi.v14i2.305.

[6]. Ibisa. (2011) Keamanan Sistem Informasi, Andi Offset.
[7]. Iku, M. H., Moch., I. S., Mustofa. (2019). Metode K-Nearest Neighbor untuk Memprediksi Harga Eceran Beras di Pasar Tradisional Gorontalo. Jurnal Nasional cosPhi, vol. 3.
[8]. Chasanah, D. N., Rahmat, A. M. (2022). Klasifikasi Kelayakan Siswa dalam Menentukan Kelas Unggulan Menggunakan Algoritma K-Nearest Neighbor. Scientific Student Journal for Information, Technology and Science, vol. III.
[9]. Abrar I. N., Abdullah M. S. (2023). Klasifikasi Penyakit Liver Menggunakan Metode Elbow untuk Menentukan K optimal pada Algoritma K-Nearest Neighbor. Jurnal SISFOKOM (Sistem Informasi dan Komputer), vol. 12.
[10]. Khotimah, H. H. (2020). Analisis Seleksi Fitur Dalam Menentukan Serangan Menggunakan Algoritma Decision Tree. Skripsi. Program Studi Teknik Informatika Universitas Pembangunan Nasional “Veteran”.