

# ANALISIS ALGORITMA MACHINE LEARNING K-NEAREST NEIGHBOR (KNN) PADA QUALITY OF SERVICE JARINGAN 5G

## ANALYSIS OF K-NEAREST NEIGHBOR (KNN) MACHINE LEARNING ALGORITHM ON 5G NETWORK QUALITY OF SERVICE

ArdaSyafianPutra<sup>1</sup>, HamzahUlinuhaMustakim<sup>2</sup>, HendyBriantoro<sup>3</sup>

<sup>1,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>2</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>3</sup>Prodi S1 Teknik Komputer, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>syafian@student.telkomuniversity.ac.id, <sup>2</sup>hamzahmustakim@telkomuniversity.co.id,

<sup>3</sup>hendybr@telkomuniversity.ac.id

### Abstrak

Dengan perkembangan pesat teknologi komunikasi, jaringan seluler telah berevolusi dari 1G hingga 5G membawa peningkatan signifikan dalam kecepatan transmisi data, kapasitas jaringan, dan efisiensi spektrum. Jaringan 5G menawarkan kecepatan hingga 10 Gbps, latensi mendekati nol dan mampu menghubungkan jutaan perangkat per kilometer persegi, mendukung aplikasi canggih seperti IoT, kendaraan otonom, AR, dan VR. Penelitian ini melibatkan pengumpulan data jaringan 5G, pra-pemrosesan, pemilihan fitur, serta penerapan algoritma machine learning seperti KNN, Decision Tree, Random Forest, dan Neural Networks. Hasilnya menunjukkan bahwa machine learning dapat meningkatkan QoS jaringan 5G, khususnya Throughput, Packet Loss, dan Delay. Algoritma KNN menghasilkan nilai Throughput terbaik pada test size 0.7 dengan akurasi 1.0000, sementara Packet Loss tidak menunjukkan pola yang signifikan. Hasil terbaik untuk Delay diperoleh pada test size 0.9 dengan akurasi 1.0000.

**Kata kunci :** perkembangan seluler, 5G, kualitas layanan, machine learning, KNN.

### Abstract

With the rapid development of communication technology, mobile networks have evolved from 1G to 5G bringing significant improvements in data transmission speed, network capacity and spectrum efficiency. 5G networks offer speeds of up to 10 Gbps, near zero latency and are capable of connecting millions of devices per square kilometer, supporting advanced applications such as IoT, autonomous vehicles, AR and VR. This research involves collecting 5G network data, pre-processing, feature selection, and applying machine learning algorithms such as KNN, Decision Tree, Random Forest, and Neural Networks. The results show that machine learning can improve 5G network QoS, especially Throughput, Packet Loss, and Delay. The KNN algorithm produces the best Throughput value at a test size of 0.7 with an accuracy of 1.0000, while Packet Loss does not show a significant pattern. The best results for Delay were obtained at a test size of 0.9 with an accuracy of 1.0000.

**Keywords:** cellular developments, 5G, service quality, machine learning, KNN

### 1. Pendahuluan

Seiring dengan pesatnya perkembangan teknologi komunikasi, jaringan seluler telah berevolusi dari generasi pertama (1G) hingga generasi kelima (5G). Setiap generasi membawa berbagai peningkatan, mulai dari kecepatan transmisi data, kapasitas jaringan, hingga efisiensi spektrum. Jaringan 5G diharapkan dapat memberikan lompatan signifikan dalam berbagai aspek, termasuk kecepatan yang lebih tinggi, latency yang lebih rendah, kapasitas yang lebih besar, serta mendukung beragam aplikasi canggih seperti Internet of Things (IoT), kendaraan otonom, augmented reality (AR), dan virtual reality (VR)[1]. Dengan kecepatan data hingga 10 Gbps dan latency yang mendekati nol, 5G tidak hanya merupakan evolusi dari jaringan sebelumnya, tetapi juga revolusi dalam bidang komunikasi seluler[2].

Namun, memastikan kualitas layanan (QoS) yang optimal pada jaringan 5G menjadi tantangan besar. QoS mencakup parameter seperti latency, throughput, jitter, dan packet loss yang memengaruhi performa jaringan dan pengalaman pengguna[3]. Dalam konteks 5G, QoS semakin

kompleks karena jaringan harus melayani berbagai aplikasi dengan kebutuhan yang berbeda. Misalnya, AR dan VR memerlukan latency yang sangat rendah untuk menjaga pengalaman yang mulus, sedangkan aplikasi IoT membutuhkan koneksi andal untuk mengelola banyak perangkat secara bersamaan. Untuk mengatasi tantangan ini, pengelolaan jaringan yang cerdas dan adaptif sangat diperlukan, dan di sinilah peran machine learning menjadi relevan.

Machine learning menawarkan solusi potensial untuk mengoptimalkan QoS dalam jaringan 5G[4]. Dengan kemampuannya menganalisis dan memproses data dalam jumlah besar secara real-time, machine learning dapat memprediksi performa jaringan dan menyesuaikan parameter secara dinamis[5]. Penelitian ini bertujuan untuk mengeksplorasi penerapan algoritma machine learning seperti k-Nearest Neighbors (k-NN), Decision Tree, Random Forest, dan Neural Networks dalam mengoptimalkan QoS. Maka implementasi machine learning ini menggunakan algoritma KNN diharapkan dapat meningkatkan efisiensi pengelolaan jaringan, mendeteksi anomali, serta memberikan pengalaman pengguna yang konsisten dan optimal[6].

## **2. Dasar Teori /Material dan Metodologi/perancangan**

### **2.1 Dasar Teori**

Quality of service adalah metode pengukuran yang digunakan untuk menilai kemampuan transfer data dalam jaringan. QoS merujuk pada tingkat kecepatan dan keandalan data dalam komunikasi, bertujuan memberikan kualitas layanan yang beragam berdasarkan kebutuhan jaringan[7]. Dalam jaringan seluler 5G membawa peningkatan yang signifikan dalam hal kecepatan, kapasitas, latensi, dan konektivitas luar. Parameter QoS mencakup throughput (kecepatan transfer data yang diukur dalam bit per detik), packet loss (jumlah paket yang hilang akibat collision), dan delay (waktu yang dibutuhkan paket data untuk berpindah dari sumber ke tujuan)[8]. Teknologi 5G dengan unsur eMBB, URLLC, dan mMTC mendukung berbagai aplikasi seperti multimedia, IoT yang meningkatkan efisiensi transformasi digital. Machine learning sebagai cabang AI (kecerdasan buatan) digunakan untuk mempelajari pola data dan membuat prediksi atau keputusan otomatis tanpa pemrograman eksplisit[9]. Dalam QoS, machine learning membantu mengoptimalkan kinerja jaringan, memprediksi masalah dan pengalaman pengguna. Algoritma seperti KNN dalam supervised learning digunakan untuk mengklasifikasikan data QoS[10]. Untuk simulasi jaringan, OMNeT++ digunakan sebagai kerangka simulasi yang memodelkan jaringan kompleks, sementara google colab adalah platform berbasis cloud yang mendukung pemrosesan data intensif serta eksperimen machine learning dalam jupyter Notebook dalam membuat keputusan atau mempelajari pola data.

### **2.2 Metodologi Penelitian**

Penelitian ini mengaplikasikan menggunakan teknik algoritma KNN untuk teknik memetakan (klasifikasi) kinerja akurasi QoS dalam dataset yang didapatkan. Fungsi dari KNN(K-Nearest Neighbor) digunakan untuk melakukan klasifikasi pengelompokan terhadap data training yang memiliki data jumlah besar dari dataset untuk menentukan jumlah terdekat untuk ditargetkan sebagai parameter K atau target data. Sementara itu, parameter QoS yang digunakan untuk melakukan klasifikasi adalah throughput, packet loss, dan delay. Alur penelitian dapat dijelaskan pada gambar 1 berikut.



**Gambar 1.** Diagram Alir Penelitian

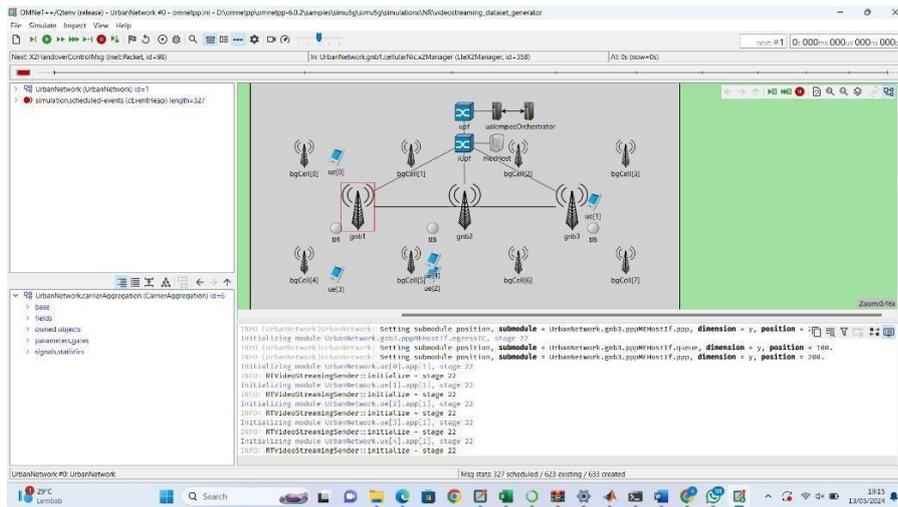
### 2.3 Pengolahan Dataset dan Perancangan KNN

Sebelum melakukan pengolahan dataset, penulis melakukan kerangka simulasi perancangan jaringan yang kompleks pada OMNeT++ dengan menggunakan model tools simu 5G. kemudian penulis melakukan desain jaringan video streaming dengan membuat topologi jaringan 5G yang mencakup node seperti, base station, UE (User Equipment) pada file .ned dan menambahkan kode untuk simulais aliran video, buffering, dan pengendalian bitrate yang dijelaskan pada gambar 2. Dalam jaringan ini komponen yang utama dalam menyediakan layanan komunikasi dan data ialah UE yang diwakili sebagai perangkat seluler terhubung ke base station yang disebut gNodeB yang seperti ditunjukkan pada gambar 3. Dengan arsitektur yang terstruktur ini, topologi jaringan mampu memenuhi kebtuhan komunikasi dan layanan data dengan efisiensi tinggi, menjaga kualitas layanan, dan pengelolaan sumber daya yang optimal.

```

1 //
2 //
3 //
4 // Authors: Giovanni Nardini, Giovanni Stea, Antonio Virdis (University of Pisa)
5 //
6 // This file is part of a software released under the license included in file
7 // "license.pdf". Please read LICENSE and README files before using it.
8 // The above files and the present reference are part of the software itself,
9 // and cannot be removed from it.
10 //
11
12 package simu5g.simulations.NR.videostreaming_dataset_generator;
13
14 import inet.networklayer.configurator.ipv4.Ipv4Ne78
15 import inet.networklayer.ipv4.RoutingTableRecorde79
16 import inet.node.ethernet.Eth10G;
17 import inet.node.inet.Router;
18 import simu5g.common.binder.Binder;
19 import simu5g.common.carrierAggregation.CarrierAg4
20 import simu5g.nodes.Upf;
21 import simu5g.nodes.NR.gNodeB;
22 import simu5g.nodes.NR.NRUE;
23 import simu5g.nodes.mec.MECHost;
24 import simu5g.nodes.mec.MecOrchestrator.MecOrche89
25 import simu5g.nodes.mec.UALCMP;
26 import simu5g.nodes.backgroundCell.BackgroundCell91
27 import simu5g.mobility.trafficLightMobility.Traffic92
28 import simu5g.world.radio.LteChannelControl;
29
30 upf: Upf {
31   @display("p=997,80005,1050");
32 }
33 iUpf: Upf {
34   @display("p=997,80005,1200");
35 }
36
37 // MEC modules
38 mechost: MECHost {
39   @display("p=1150,1200");
40 }
41 mecOrchestrator: MecOrchestrator {
42   @display("p=1300,1050");
43 }
44 ualcmp: UALCMP {
45   @display("p=1150,1050");
46 }
  
```

**Gambar 2.** Source code Perancangan Jaringan



Gambar 3. Topologi Jaringan

## 2.4 Perancangan KNN

Setelah penulis melakukan perancangan topologi OMNeT++ video streaming, selanjutnya penulis melakukan pengujian selama 300detik untuk mendapatkan parameter scalar yang dihasilkan dari topologi video streaming. Langkah selanjutnya, penulis melakukan pembedahan parameter yang didapatkan untuk mendapatkan 100 data parameter QoS seperti throughput, packet loss, dan delay yang dijadikan sebagai pengolahan dataset kinerja KNN. Dataset selanjutnya diekstrak dalam bentuk CSV dalam format file log yang terstruktur dengan nama dataset vidstream.csv. Yang dapat dijelaskan pada gambar 4. Dataset yang telah diproses kemudian diunggah ke google colab, dimana penulis dapat melakukan algoritma KNN untuk melatih model pola kinerja machine learning yang dijelaskan pada gambar 5. Source code yang terdapat pada google colab meliputi pandas (sebagai mengelola dan memanipulasi data), tensorflow (sebagai membuat dan melatih model machine learning), scikit-learn (sebagai membagi dataset menjadi data training dan data testing dan melakukan praproses data), dan matplotlib (digunakan sebagai membuat visualisasi grafis data dan peforma model kinerja).

	A	B	C	D	E	F	G	H	I
1	Throughput	Packet Loss	Delay						
2	403903	0.080804803	0.004						
3	581039	0.027848101	0.004						
4	139994	0.101928996	0.010668018						
5	137348	0.050051778	0.009415698						
6	139994	0.027895182	0.008601739						
7	409624	0.057511328	0.009392751						
8	434291	0.040862982	0.007						
9	133087	0.027415966	0.004						
0	528962	0.097396487	0.004						
1	563917	0.003245132	0.004						
2	855112	0.022443036	0.004						
3	812041	0.004074402	0.003						
4	692878	0.017530006	0.004						
5	192518	0.012186865	0.007						
6	940035	0.00789564	0.008						
7	204414	0.032951786	0.00840495						
8	625341	0.034673072	0.008249634						
9	498340	0.043970009	0.004						
0	708680	0.016565869	0.045						
1	184977	0.029187936	0.030231413						
2	567290	0.022443036	0.030231413						
3	156944	0.004324934	0.007941741						
4	156944	0.021654287	0.032215073						
5	350574	0.017530006	0.004998581						
6	811942	0.018032887	0.008289021						
7	812041	0.021644749	0.008690419						
8									
9									

Gambar 4. Dataset CSV

```
import pandas as pd
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

# Muat data dari file CSV
data = pd.read_csv('vidstreams.csv')

# Tampilkan beberapa baris pertama dari dataset
print(data.head(10))

# Tampilkan nama kolom dalam DataFrame
print("Kolom dalam DataFrame:", data.columns)

# Tentukan kolom label dan fitur berdasarkan nama kolom yang benar
label_column = 'Delay' # Gantilah dengan nama kolom label yang benar
feature_columns = ['Throughput', 'Delay', 'Packet Loss'] # Gantilah dengan nama kolom fitur yang benar

# Tampilkan nama kolom label dan fitur
print("Kolom label:", label_column)
print("Kolom fitur:", feature_columns)

# Periksa apakah kolom-kolom tersebut ada dalam DataFrame
print("Kolom yang digunakan sebagai fitur:", feature_columns)
print("Kolom yang digunakan sebagai label:", label_column)

# Pastikan semua kolom fitur dan label ada dalam DataFrame
missing_columns = [col for col in feature_columns + [label_column] if col not in data.columns]
if missing_columns:
```

Gambar 5. Source code python google colab

Penerapan KNN pada QoS diterapkan pada mengklasifikasikan jenis trafik jaringan berdasarkan parameter seperti throughput, packet loss, dan delay. Jenis-jenis trafik ini dijadikan pengelompokan yang digunakan untuk memprediksi pola kinerja penggunaan jaringan dan peforma berdasarkan pola pengujian test size yang dijelaskan pada gambar 6. Pada source code label\_column ini digunakan untuk mempresentasikan variabel target yang akan diprediksi oleh model dari salah satu parameter yang akan dilakukan sebuah test pola kinerja machine learning. Sedangkan feature\_column sebagai nama kolom fitur untuk memprediksi label yang ada dalam fitur dataset CSV. Kode dari kedua ini untuk mencetak nama kolom label dan fitur untuk verifikasi dan

memeriksa keberadaan kolom-kolom tersebut yang digunakan benar dan terdapat ke dalam dataset CSV.

```
# Tentukan kolom label dan fitur berdasarkan nama kolom yang benar
label_column = 'Packet Loss' # Gantilah dengan nama kolom label yang benar
feature_columns = ['Throughput', 'Packet Loss', 'Delay'] # Gantilah dengan nama kolom fitur yang benar

# Tampilkan nama kolom label dan fitur
print("Kolom Label:", label_column)
print("Kolom Fitur:", feature_columns)

# Periksa apakah kolom-kolom tersebut ada dalam DataFrame
print("Kolom yang digunakan sebagai fitur:", feature_columns)
print("Kolom yang digunakan sebagai label:", label_column)
```

**Gambar 6.** Penerapan KNN

### 3. Pembahasan

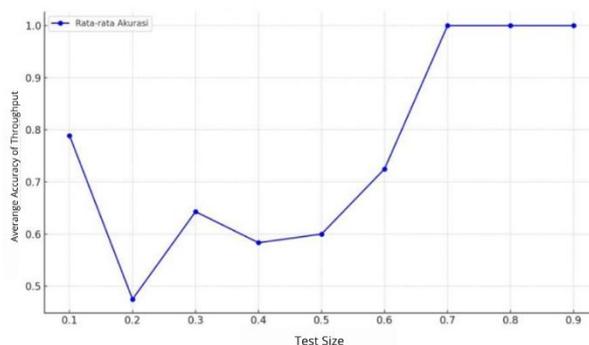
#### 3.1. Penentuan Test size

Pemilihan test size yang digunakan sebagai ukuran data uji yang optimal untuk melatih model bergantung pada beberapa faktor antara lain; ukuran total dataset, kompleksitas model, dan tujuan spesifik dari eksperimen atau aplikasi. Dalam konteks machine learning, pemisahan antara dataset menjadi data latih (training data) dan data uji (test data) adalah langkah awal untuk mengevaluasi kinerja model yang akan dibentuk. Berikut tabel 1. pemilihan test size ukuran data latih dan data uji yang digunakan untuk melatih model mengevaluasi pola kinerja machine learning.

Tabel 1. Penentuan *Test Size*

Test size	Data latih	Data uji
0.1	90% data latih	10% data uji
0.2	80% data latih	20% data uji
0.3	70% data latih	30% data uji
0.4	60% data latih	40% data uji
0.5	50% data latih	50% data uji
0.6	40% data latih	60% data uji
0.7	30% data latih	70% data uji
0.8	20% data latih	80% data uji
0.9	10% data latih	90% data uji

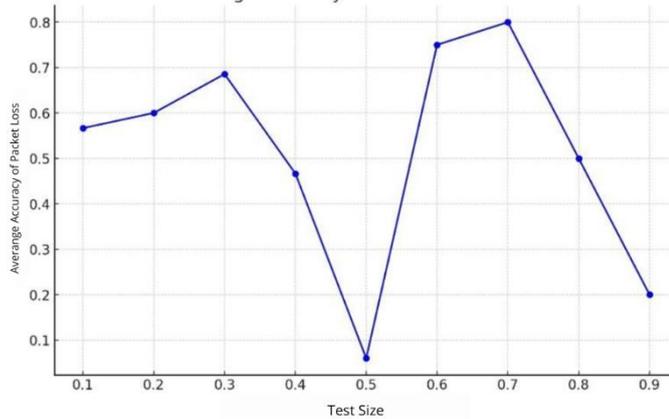
#### 3.2 Throughput



Hasil throughput yang didapat dari melakukan pengujian test size yang dilakukan. Pada gambar 7. Menunjukkan hasil grafik nilai akurasi yang tinggi pada test size 0.7 dengan nilai akurasi

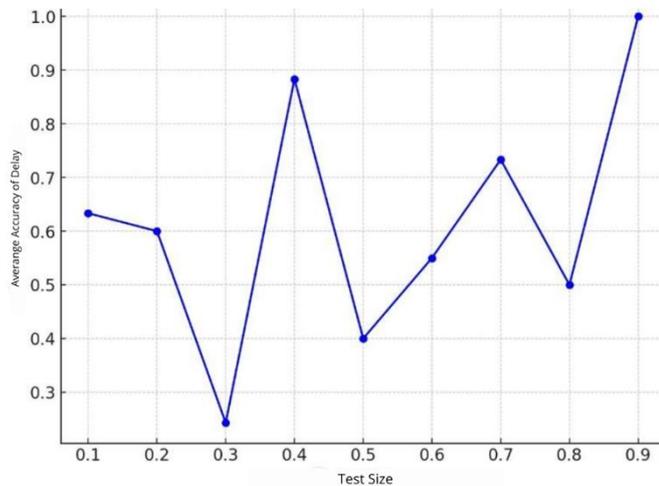
1.0. Hal ini menunjukkan bahwa, pengujian pada test size tinggi dan dengan dataset yang besar pada 100 data pola kinerja yang dihasilkan menunjukkan nilai akurasi stabil dan bagus pada data besar. Maka throughput dengan pola test size yang besar dengan jumlah data besar bisa mempelajari pola kinerja dan memprediksi pola kinerja yang baik.

### 3.3 Packet Loss



Pada pengujian packet loss yang dilakukan pada gambar 8. diatas menunjukkan bahwa hasil grafik yang dihasilkan mengalami pola kinerja fluktuasi dengan penurunan yang sangat tajam. Hal ini disebabkan oleh dataset yang didapatkan belum bisa mampu memprediksi pola kinerja akurasi yang diperoleh dari parameter packet loss, sehingga belum mampu membaca pola kinerja yang dibentuk dalam klasifikasi packet loss yang terbaik.

### 3.4 Delay



Hasil delay yang dilakukan pengujian test size pada gambar 8 diatas menunjukkan bahwa hasil nilai akurasi yang terbaik ditunjukkan pada test size 0.9 dengan nilai akurasi 1.0. Hal ini menunjukkan bahwa pengujian pada test size yang dilakukan tinggi dapat mempelajari pola kinerja dari dataset yang diberikan. Namun, di tiap tiap test size terjadi penurunan, mungkin ini disebabkan oleh data yang diperoleh masih belum cukup memenuhi standart klasifikasi model dan belum bisa mampu mempelajari pola model yang diberikan

### 3.5 Analisis

Berdasarkan hasil pengujian klasifikasi dengan algoritma KNN yang telah dilakukan, dengan melakukan ketiga parameter yang dijadikan sebagai uji klasifikasi. Hasil menunjukkan bahwa akurasi KNN yang dilakukan pada throughput menghasilkan nilai yang terbaik pada pengujian test size 0.7 dengan nilai akurasi maksimal yaitu 1.0. Sedangkan pada packet loss menunjukkan belum mampu membaca pola kinerja nilai akurasi yang dibentuk oleh klasifikasi KNN, ini memungkinkan dataset yang diterima belum mampu memprediksi pola KNN yang

dilakukan untuk melakukan klasifikasi tersebut. Dan untuk hasil pada delay menunjukkan nilai akurasi yang terbaik pada test size 0.9 dengan nilai akurasi 1.0, ini menunjukkan bahwa parameter delay pada dataset tersebut mampu memprediksi pola kinerja KNN yang dibentuk. Akan tetapi, terjadi fluktuasi yang memungkinkan bahwa dataset tersebut masih kurang mempelajari pola data KNN. Pada penelitian ini menegaskan bahwa kinerja yang dibentuk untuk mengklasifikasi KNN machine learning ini sangat dipengaruhi oleh kinerja parameter QoS untuk mendapatkan hasil pengolahan dataset yang terbaik. Maka untuk mengklasifikasikan ke dalam KNN dapat menghasilkan performa akurasi yang optimal dan stabil.

#### 4. Kesimpulan

Hasil yang didapatkan pada pengujian test size yang dilakukan, pada throughput menghasilkan nilai yang terbaik pada test size 0.7 dengan nilai akurasi 1.0. Sedangkan pada packet loss tidak memiliki hubungan antara nilai akurasi yang dihasilkan dengan test size yang diuji, karena dataset yang diolah ini belum mampu membaca pola kinerja yang dilakukan. Dan pada nilai delay memiliki nilai akurasi di test size maksimal 0.9 dengan nilai akurasi sebesar 1.0, meskipun pada test rendah terjadi (fluktuasi penurunan dan kenaikan). Pada penelitian ini menegaskan bahwa kinerja yang dibentuk untuk mengklasifikasi KNN machine learning ini sangat dipengaruhi oleh kinerja parameter QoS untuk mendapatkan hasil pengolahan dataset yang terbaik. Maka untuk mengklasifikasikan ke dalam KNN dapat menghasilkan performa akurasi yang optimal dan stabil.

#### Daftar Pustaka:

- [1] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. Emad Ul Haq, "Machine Learning Techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021, doi: 10.1109/ACCESS.2021.3051557.
- [2] A. Imanbayev *et al.*, "Research of Machine Learning Algorithms for the Development of Intrusion Detection Systems in 5G Mobile Networks and Beyond," *IEEE Access*, vol. 22, no. 24, Dec. 2022, doi: 10.3390/s22249957.
- [3] N. Azura, "ANALISIS KINERJA JARINGAN WIRELESS LAN DENGAN MENGGUNAKAN METODE QUALITY OF SERVICE (QoS)," 2018.
- [4] C. Chrysoulas and M. Fasli, "A Service Oriented QoS Architecture Targeting the Smart Grid World & Machine Learning Aspects," 2019.
- [5] S. H. Alsamhi *et al.*, "Machine Learning for Smart Environments in B5G Networks: Connectivity and QoS," 2021, *Hindawi Limited*. doi: 10.1155/2021/6805151.
- [6] A. Kecerdasan Artifisial dalam Klasifikasi Sampah sebagai Upaya Pengolahan Sampah Plastik Latifah Listyalina *et al.*, "The Application of Artificial Intelligence in Waste Classification as an Effort In Plastic Waste Management," *Jurnal Informatika dan Teknologi Informasi*, vol. 21, no. 1, pp. 1–13, 2024, doi: 10.31515/telematika.v21i1.11977.
- [7] S. Tari, "ANALISIS QUALITY OF SERVICE (QoS) RADIO STREAMING MENGGUNAKAN ICECAST PADA Wi-Fi 802.11n," 2023.
- [8] X. Wang *et al.*, "QoS and Privacy-Aware Routing for 5G-Enabled Industrial Internet of Things: A Federated Reinforcement Learning Approach," *IEEE Access*, vol. 18, no. 6, pp. 4189–4197, Jun. 2022, doi: 10.1109/TII.2021.3124848.

- [9] V. Nasteski, "An overview of the supervised machine learning methods," *IEEE Access*, vol. 4, pp. 51–62, Dec. 2017, doi: 10.20544/horizons.b.04.1.17.p05.
- [10] J. Homepage, S. R. Cholil, T. Handayani, R. Prathivi, and T. Ardianita, "IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa," 2021.