E- JOINT (Electronica and Electrical Journal of Innovation Technology) ISSN : 2745 - 9292
Vol. 03, No. 1, Juni 2022
Politeknik Negeri Cilacap

Klasifikasi serangan *Malware* terhadap Lalu Lintas Jaringan *Internet of Things* menggunakan Algoritma *K-Nearest Neighbour* (K-NN)

Ari Sandriana[*1], Rianto[2], Firmansyah Maulana[3]

[3]*Jurusan Teknik Informatika Universitas Siliwangi*

Jl. Siliwangi No.24, Kahuripan, Tawang, Tasikmalaya, Jawa Barat 46115

[1]arisandriana96@gmail.com

[2]rianto@unsil.ac.id

[3]firmansyah@unsil.ac.id

Abstrak— **Penerapan *Internet of Things* (IoT) dapat membuat semuanya terhubung ke internet tetapi sistem IoT dapat menjadi sasaran yang sangat mudah untuk disusupi penyerang dengan menggunakan *malware*, lebih dari 1,6 miliar atau tepatnya 1.637.973.022 anomali *traffic* atau serangan siber (*cyberattack*) yang terjadi diseluruh wilayah Indonesia sepanjang tahun 2021, teknik *machine learning* dapat dimanfaatkan untuk proses pengklasifikasian anomali *traffic* dengan menggunakan algoritma *k-nearest neighbour* (KNN) sehingga dapat membedakan data *traffic* yang bersifat *benign* atau *malicious*. Data anomali *traffic* yang digunakan adalah *dataset* aposemat IoT-23, didalam *dataset* tersebut terdapat 23 *dataset*, lalu terbagi kedalam 20 *dataset scenario malicious* dan 3 *dataset scenario benign*. Namun dataset yang digunakan adalah 20 *dataset scenario malicious*. 20 *dataset* tersebut selanjutnya dilakukan data *preprocessing* supaya dapat digunakan untuk proses *training* model atau pengklasifikasian. Nilai akurasi yang didapatkan setelah proses *training* model sebesar 0.94 atau 94%, model yang sudah dilakukan *training* model dapat memprediksi *traffic* data baru kedalam *benign* atau *malicious*, data baru yang sudah disiapkan adalah sebanyak 25 data baru. Prediksi 25 data baru tersebut menghasilkan 20 data diprediksi benar atau sesuai dan 5 data diprediksi salah atau tidak sesuai, 5 data tersebut terbagi menjadi 3 data yang harusnya diprediksi *benign* dan 2 data yang harusnya diprediksi *malicious*.**

Kata kunci: *Internet of Things, Malware, Machine learning, Klasifikasi, K-Nearest Neighbour* (K-NN)

Abstract— ***The implementation of the internet of things (IoT) can make everything connected to the internet but the IoT system can be a very easy target for attackers to infiltrate using malware, more than 1.6 billion or 1,637,973,022 traffic anomalies or cyber attacks to be exact. occurring throughout Indonesia throughout 2021, machine learning techniques can be used for the traffic anomaly classification process using the k-nearest neighbor (KNN) algorithm so that it can distinguish benign or malicious traffic data. The traffic anomaly data used is the IoT-23 aposemat dataset, in the dataset there are 23 datasets, then divided into 20 datasets for malicious scenarios and 3 datasets for benign scenarios. However, the dataset used is 20 datasets for malicious scenarios. The 20 datasets are then preprocessed so that they can be used for model training or classification processes. The accuracy value obtained after the model training process is 0.94 or 94%, the model that has been trained in the model can predict new data traffic into benign or malicious, the new data that has been prepared is as many as 25 new data. The prediction of the 25 new data resulted in 20 data predicted to be correct or appropriate and 5 data predicted to be incorrect or inappropriate, the 5 data divided into 3 data that should be predicted to be benign and 2 data that should be predicted to be malicious.***

***Keywords: Internet of Things, Malware, Machine learning, Classification, K-Nearest Neighbour* (K-NN)**

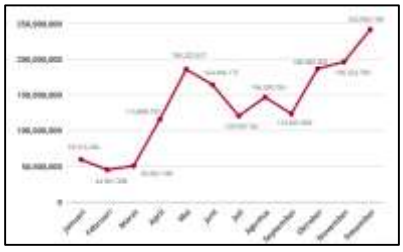
I. PENDAHULUAN

Internet of Things (IoT) terus berkembang dalam segi komunikasi antar perangkat, baik *software* maupun *hardware*. IoT membuat beragam perangkat dapat terhubung bersama ke internet. Selain itu, IoT merupakan koneksi unik perangkat komputasi yang dapat diidentifikasi, tertanam dan dapat mengirimkan data melalui jaringan tanpa interaksi antar pengguna atau perangkat. Jumlah perangkat yang terhubung ke internet dengan IoT semakin hari semakin berkembang. Teknologi IoT bisa menjadi sesuatu yang berharga dalam hal inovasi tetapi juga dapat menjadi ancaman keamanan siber yang signifikan bagi sistem yang rentan. Situasi ini adalah potensi risiko besar dalam hal keamanan siber, karena banyak titik masuk yang mungkin memiliki kerentanan keamanan. Kerentanan dalam rantai keamanan sistem dapat menimbulkan risiko keamanan untuk keseluruhan sistem dan memberikan peluang bagi penyerang untuk melancarkan aksinya. Terutama

infrastruktur penting yang harus dilindungi dari risiko keamanan siber yang sangat besar [1].

Salah satu tantangan yang harus diatasi untuk mendorong implementasi IoT secara luas adalah faktor keamanan. IoT merupakan sebuah sistem yang majemuk. Kemajemukannya bukan hanya karena keterlibatan berbagai entitas seperti data, mesin, RFID, sensor dan lain-lain, tetapi juga karena melibatkan berbagai peralatan dengan kemampuan komunikasi dan pengolahan data. Banyaknya entitas dan data yang terlibat, membuat IoT menghadapi resiko keamanan yang dapat mengancam dan membahayakan penggunanya. Ancaman ini utamanya dilakukan dengan cara memungkinkan orang yang tidak berhak untuk mengkases data dan menyalahgunakan informasi personal, memfasilitasi serangan terhadap sistem yang lain, serta mengancam keselamatan personal penggunanya [2]. Jumlah penggunaan perangkat IoT (*Internet of Things*) telah meningkat dan tersebar luas. Ancaman keamanan pada perangkat IoT juga meningkat. Berbagai serangan siber dapat dilakukan pada perangkat IoT, mulai dari mengambil hak akses, merusak data, mencuri informasi penting, dan merekam aktivitas pribadi pengguna saat menggunakan perangkat IoT. Serangan siber ini memasuki sistem melalui *malware* yang berhasil ditanam di perangkat IoT [3].

Malware merupakan perangkat lunak (*software)* yang memiliki tujuan negatif, seperti merusak data, mencuri informasi penting, mengganggu kinerja perangkat, dan mengambil alih sistem, ancaman ini terus meningkat setiap tahunnya. Pada tahun 2019 serangan siber menggunakan *malware* di Indonesia merupakan yang tertinggi di Asia Pasifik [4]. Selain itu, Badan Siber dan Sandi Negara (BSSN) mempublikasikan laporan tahunan bertajuk "Monitoring Keamanan Siber" untuk tahun 2021, isi didalam laporan tersebut lebih dari 1,6 miliar atau tepatnya 1.637.973.022 anomali *traffic* atau serangan siber (*cyber attack*) yang terjadi diseluruh wilayah Indonesia sepanjang tahun 2021 [5].



Gambar 1. Data statistik *monitoring* anomali *traffic* BSSN tahun 2021

Pada Gambar 1 serangan siber paling banyak terjadi pada bulan desember 2021 dengan jumlah lebih dari 242 juta anomali, sedangakan serangan siber paling sedikit terjadi pada bulan februari dengan jumlah hampir 45 juta anomali anomali *traffic* didominasi oleh serangan *malware*, statistik dari serangan sepanjang tahun 2021.

Salah satu aktivitas mencurigakan dari *malware* adalah penggunaan lalu lintas jaringan yang dapat dimanfaatkan sebagai media untuk mengirimkan informasi rahasia, informasi rekening bank, pesan pribadi, dan kata sandi. *Malware* juga bisa memanfaatkan lalu lintas jaringan sebagai pintu belakang untuk dapat dimasuki oleh *malware* lainnya [3].

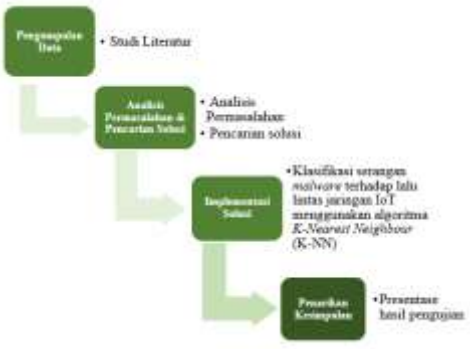
Penelitian berjudul *"Klasifikasi Malicious Websites Menggunakan Algoritma K-NN Berdasarkan Application Layers dan Network Characteristics"* yang dilakukan oleh Arther S dkk (2020) menyebutkan bahwa algoritma yang telah di evaluasi yaitu *K-Nearest Neighbor*, *Decision Tree*, *Logistic Regression*, dan *Random Forest* dapat disimpulkan bahwa algoritma K-NN memiliki performa yang paling baik di antara algoritma lainnya dengan hasil 95.51% *accuracy*, 89.42% *recall*, 89.42% *precision*, dan 0.212 RMSE untuk hasil *independent* sedangkan untuk hasil 10 *fold cross validation* memiliki hasil 93.61% *accuracy*, 85.05% *recall*, 85.25% *precision*, dan 0.251 RMSE dalam mendeteksi *malicious* dan *benign website* [6].

Algoritma *K-Nearest Neighbour* (K-NN) adalah algoritma *supervised* yang memiliki hasil *query instance* baru yang didapat dengan klasifikasi dengan nilai mayoritas dari kategori. Algoritma K-NN memiliki tujuan untuk mengklasifikasi obyek baru berdasarkan atribut data dan *training sample* [7].

Berdasarkan hasil paparan permasalahan maka akan berfokus pada klasifikasi serangan *malware* terhadap lalu lintas jaringan IoT menggunakan algoritma *K-Nearest Neighbour* (K-NN).

II. METODE

Gambar 2. merupakan tahap penelitian yang dimulai dari proses pengumpulan data, analisis permasalahan dan pencarian solusi, implementasi solusi, sampai proses penarikan kesimpulan.



Gambar 2. Tahapan Penelitian

A. Pengumpulan Data

Data yang merupakan penunjang penelitian dapat diperoleh melalui studi literatur. Studi literatur berisi uraian tentang teori, temuan, dan bahan penelitian lainnya yang diperoleh dari jurnal nasional ataupun internasional yang berupa *survey paper* dan *technical paper*. Studi literatur yang digunakan untuk menunjang penelitian ini adalah jurnal terkait data *mining* dan algoritma *machine learning* untuk digunakan pada proses klasifikasi data, *dataset* yang didapatkan berjumlah 20 *dataset* dengan kolom atau atribut yang sama lalu *dataset* tersebut dilakukan data *preprocessing* untuk membersihkan *dataset* dari data yang kosong atau *missing values*, setelah *dataset* sudah melewati data *preprocessing* kemudian dilakukan *export dataset* menjadi sebuah file berekstensi .CSV dan akan dijadikan sebagai bahan penelitian.

B. Analisis Permasalahan dan Pencarian Solusi

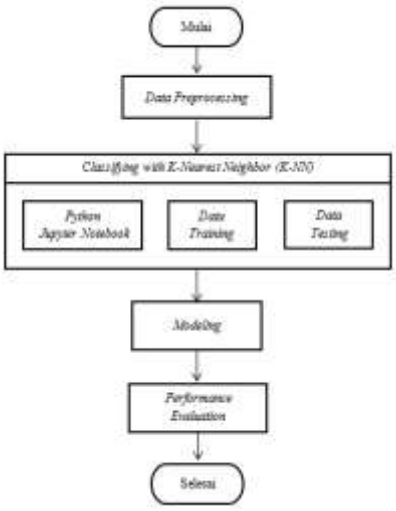
Tahapan analisis permasalahan dan pencarian solusi merupakan tahap pengembangan yang dilakukan pasca pengumpulan data. Masalah yang ditemukan pada proses studi literatur, kemudian dikaji dan dicarikan solusinya berdasarkan perkembangan ilmu pengetahuan dan teknologi yang ada. Rentannya keamanan dan banyaknya anomali pada lalu lintas jaringan IoT yang disebabkan serangan *malware* menjadi sebuah masalah yang harus dicari solusinya. Solusi yang dipilih adalah dilakukan pengklasifikasian serangan malware menggunakan algoritma K-Nearest Neighbour (K-NN) berdasarkan anomali *traffic* pada lalu lintas jaringan IoT.

Selain proses klasifikasi, pada penelitian ini juga dilakukan proses prediksi data lalu lintas jaringan IoT menggunakan model yang sudah dilakukan proses *training*, sehingga model dapat memprediksi *malware* menjadi 2 kategori yaitu *benign* dan *malicious*.

C. Implementasi Solusi

Tahap awal pada proses implementasi solusi adalah data *preprocessing* dibagi menjadi 2 bagian, yaitu: data *cleaning* dan data *reduction*. Data *cleaning* adalah proses pembersihan data *incomplete* pada *attribute* di *dataset* untuk membuat data menjadi lebih konsisten. Sedangkan, data *reduction* adalah proses untuk menghapus data pada *attribute* yang kurang dominan sehingga data bisa dikurangi, namun tetap menghasilkan data yang akurat. Tahap kedua adalah *Classifying with K-Nearest Neighbour* (K-NN) yang merupakan proses klasifikasi serangan *malware* pada data lalu lintas jaringan IoT dikategorikan bersifat *benign* atau *malicious*. Tahap terakhir adalah tahap *performance evaluation*, setelah pembuatan model maka langkah selanjutnya adalah melakukan evaluasi dengan *performance evaluation*. *Performance evaluation* berguna untuk menguji performa dari *classifier*. *recall*, *precision*, dan *accuracy*. *Recall* adalah kumpulan data positif yang diklasifikasikan dengan benar sebagai data positif. *Precision* adalah kumpulan data yang diklasifikasikan sebagai positif yang benar – benar positif. *Accuracy* adalah ketepatan klasifikasi data.

Gambar 3 merupakan *flowchart* yang menggambarkan secara utuh tahap implementasi solusi.



Gambar 3. *Flowchart* tahap implementasi solusi

Pada Gambar 3 terdapat *flowchart* yang menggambarkan alur dari proses implementasi solusi diawali dengan data *preprocessing* digunakan untuk membersihkan data dari *null* dan *missing values*, tahap kedua adalah *classifying with k-nearest neighbour* (K-NN) yang merupakan proses klasifikasi serangan *malware* pada data lalu lintas jaringan IoT dikategorikan bersifat *benign* atau *malicious*, tahap terakhir adalah tahap *performance evaluation*, setelah pembuatan model maka langkah selanjutnya adalah melakukan evaluasi dengan *performance evaluation* serta di uji dengan memprediksi data baru.

D. Penarikan Kesimpulan

Penarikan kesimpulan merupakan tahap akhir dari penelitian ini dimana model yang sudah melewati proses training model menggunakan algoritma *K-Nearest Neighbour* (K-NN) yang dapat memprediksi data-data anomali *traffic* kedalam 2 kategori seperti *benign* dan *malicious*. Model tersebut di *training* menggunakan 20 dataset anomali *traffic* yang sudah dipilih, data tersebut dibagi menjadi 60% untuk data *training* dan 40% untuk data *testing* yang menghasilkan nilai akurasi sehingga model dapat digunakan untuk memprediksi data-data anomali *traffic* selanjutnya.

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

Studi literatur dilakukan dengan melakukan pencarian *dataset* yang berkaitan dengan lalu lintas jaringan IoT. *Dataset*

diperoleh dari website stratosphereips.org dengan nama *dataset* aposemat iot-23. *Dataset* yang tersedia pada website stratosphereips.org khusunya aposemat iot-23 terdapat sebanyak 23 *scenario*, 20 diantaranya *scenario malicious* dan 3 *scenario benign*. Pada penelitian ini *dataset* yang digunakan sebanyak 20 *dataset*, *dataset* yang dipilih hanya *scenario malicious*.

B. Analisis Permasalahan dan Pencarian Solusi

Objek yang menjadi bahan pada penelitian ini merupakan data lalu lintas jaringan IoT yang didalamnya terdapat lalu lintas bersifat *malicious* dan *benign*, masalah yang ditemukan adalah bagaimana melakukan klasifikasi serangan *malware* terhadap lalu lintas jaringan IoT menggunakan algoritma *K-Nearest Neighbour* (K-NN), pencarian solusi untuk permasalahan tersebut dilakukan klasifikasi terhadap *dataset* aposemat iot-23 dengan menggunakan algoritma *K-Nearest Neighbour* (K-NN) serta *tools jupyter lab*. Lalu masalah kedua bagaimana menguji tingkat akurasi menggunakan algoritma *K-Nearest Neighbour* (K-NN) terhadap *dataset* aposemat iot-23, pencarian solusi untuk permasalahan tersebut membagi *dataset* menjadi data *training* dan data *testing* lalu digunakan algoritma *K-Nearest Neighbour* (K-NN) untuk proses klasifikasi atau *training model*, nilai akurasi akan didapatkan setelah proses klasifikasi atau *training* model yang nantinya model tersebut akan digunakan untuk memprediksi data-data baru.

C. Implementasi Solusi

1) Data Preprocessing

Data *preprocessing* merupakan tahap untuk membuat data menjadi lebih konsisten dan menghapus data pada *attribute* yang kurang dominan sehingga data bisa dikurangi, namun tetap menghasilkan data yang akurat. Pada Tabel 1 terdapat penjelasan dari masing-masing kolom atau atribut yang akan digunakan.

Tabel 1. Penjelasan masing-masing kolom

No	Nama Kolom	Penjelasan
1	ts	Waktu paket pertama
2	uid	ID identifikasi koneksi (unik)
3	id.orig_h	4-tupel alamat/port titik akhir koneksi.
4	id.orig_p	4-tupel alamat/port titik akhir koneksi.
5	id.resp_h	4-tupel alamat/port titik akhir koneksi.
6	id.resp_p	4-tupel alamat/port titik akhir koneksi.
7	proto	Protokol koneksi.
8	service	Identifikasi protokol aplikasi yang dikirim melalui koneksi.
9	duration	Berapa lama koneksi berlangsung
10	orig_bytes	Jumlah byte payload yang dikirim oleh originator. Untuk TCP ini diambil dari nomor urut dan mungkin tidak akurat (misalnya, karena koneksi yang besar).
11	resp_bytes	Jumlah byte payload yang dikirim oleh responden. Lihat orig_bytes .
12	conn_state	Pencatatan informasi umum mengenai lalu lintas
13	local_orig	Jika koneksi berasal secara lokal, nilai ini akan menjadi T. Jika berasal dari jarak jauh akan menjadi F. Jika Site::local_netsvariabel tidak ditentukan, bidang ini akan dibiarkan kosong setiap saat.
14	local_resp	Jika koneksi ditanggapi secara lokal, nilai ini akan menjadi T. Jika ditanggapi dari jarak jauh akan menjadi F. Jika Site::local_netsvariabel tidak terdefinisi, bidang ini akan dibiarkan kosong setiap saat.
15	missed_bytes	Menunjukkan jumlah byte yang terlewatkan dalam celah konten, yang mewakili kehilangan paket. Nilai selain nol biasanya akan menyebabkan analisis protokol gagal tetapi beberapa analisis mungkin telah diselesaikan sebelum paket hilang.
16	history	Merekam riwayat status koneksi sebagai string huruf.
17	orig_pkts	Jumlah paket yang dikirim oleh originator. Hanya disetel jika use_conn_size_analyzer= T.
18	orig_ip_bytes	Jumlah byte level IP yang dikirim oleh originator (seperti yang terlihat pada kabel, diambil dari bidang header total_length IP). Hanya disetel jika use_conn_size_analyzer= T.
19	resp_pkts	Jumlah paket yang dikirim oleh responden. Hanya disetel jika use_conn_size_analyzer= T.
20	resp_ip_bytes	Jumlah byte level IP yang dikirim oleh responden (seperti yang terlihat pada kabel, diambil dari bidang header total_length IP). Hanya disetel jika use_conn_size_analyzer= T.
21	tunnel_parents	Jika koneksi ini melalui terowongan, tunjukkan nilai uid untuk koneksi induk enkapsulasi yang digunakan selama masa pakai koneksi dalam ini.
22	label	Kategori dari data lalu lintas seperti malicious atau benign

23	detailed-label	Kategori yang lebih detail seperti jenis-jenis serangan

Tabel 1 merupakan penjelasan dari masing-masing kolom yang terdapat dalam *dataset* aposemat IoT-23, jumlah kolom sebanyak 23 kolom beserta penjelasannya.

a) Data Cleaning

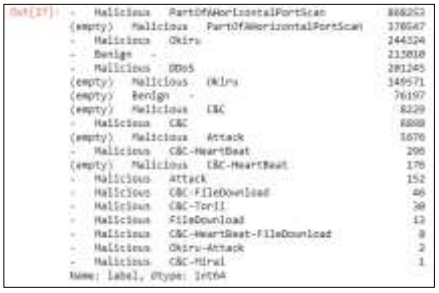
Data *cleaning* merupakan proses pembersihan data *incomplete* pada *attribute* di *dataset* pada Tabel 2 untuk membuat data menjadi lebih konsisten.

Tabel 2. Nama *dataset* beserta ukuran *file* dan jumlah data

No	Nama *Dataset*	Ukuran File (KB)	Jumlah Data Sebelum (*Row*)	Jumlah Data Sesudah (*Row*)
1	CTU-IoT-Malware-Capture-34-1 (Mirai)	2.949	23.146	23.142
2	CTU-IoT-Malware-Capture-43-1 (Mirai)	9.143.811	67.321.810	149.999
3	CTU-IoT-Malware-Capture-44-1 (Mirai)	31	238	234
4	CTU-IoT-Malware-Capture-49-1 (Mirai)	833.624	5.410.562	149.999
5	CTU-IoT-Malware-Capture-52-1 (Mirai)	2.998.360	19.781.379	149.999
6	CTU-IoT-Malware-Capture-20-1 (Torii)	410	3.210	3.206
7	CTU-IoT-Malware-Capture-21-1 (Torii)	419	3.287	3.283
8	CTU-IoT-Malware-Capture-42-1 (Trojan)	573	4.427	4.423
9	CTU-IoT-Malware-Capture-60-1 (Gagfyt)	456.602	3.581.029	149.999
10	CTU-IoT-Malware-Capture-17-1 (Kenjiro)	7.948.604	54.659.864	149.999
11	CTU-IoT-Malware-Capture-36-1 (Okiru)	1.745.950	13.645.107	149.999
12	CTU-IoT-Malware-Capture-33-1 (Kenjiro)	7.683.516	54.454.592	149.999
13	CTU-IoT-Malware-Capture-8-1 (Hakai)	1.398	10.404	10.400
14	CTU-IoT-Malware-Capture-35-1 (Mirai)	1.308.769	10.447.796	149.999
15	CTU-IoT-Malware-Capture-48-1 (Mirai)	519.276	3.394.347	149.999
16	CTU-IoT-Malware-Capture-39-1 (IRCBot)	10.630.236	73.568.982	149.999
17	CTU-IoT-Malware-Capture-7-1 (Linux.Mirai)	1.544.203	11.454.723	149.999
18	CTU-IoT-Malware-Capture-9-1 (Linux.Hajime)	970.176	6.378.294	149.999
19	CTU-IoT-Malware-Capture-3-1 (Muhstik)	23.813	156.104	149.999
20	CTU-IoT-Malware-Capture-1-1 (Hide and Seek)	144.830	1.008.749	149.999
Jumlah		**45.957.550**	**325.308.050**	**2.144.674**

Dataset yang sudah dilakukan proses data *cleaning*, setiap *dataset* nya dipilih 150.000 data, tetapi terdapat beberapa *dataset* yang kurang dari 150.000 data karena ukuran *dataset* nya yang kecil. Pada setiap *dataset* terdapat *attribute* atau kolom seperti *ts, uid, id.orig_h, id.resp_p, proto, service, duration, orig_bytes, resp_bytes, conn_state, local_orig, local_resp, missed_bytes, history, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes*.

Setelah proses penggabungan data, lalu pada kolom *label* terdapat data dengan kategori *benign* dan *malicious*, untuk data kategori *malicious* terdapat beberapa jenis serangan sehingga perlu dihitung jumlah datanya berdasarkan jenis serangannya maka akan menghasilkan jumlah seperti pada Gambar 4.



Gambar 4. Hasil jumlah data berdasarkan jenis serangannya

Gambar 4 merupakan hasil menghitung jumlah data berdasarkan jenis serangannya yang menghasilkan 19 jenis serangan berkategori *benign* dan *malicious* dengan nilai terbesar 868.253 data pada jenis serangan Malicious *Part of a Horizontal Port Scan*. Dengan didapatkannya jumlah data berdasarkan jenis serangan maka diperlukan untuk mengubah data pada kolom *label* hanya kedalam kategori *benign* dan *malicious* tanpa menggunakan jenis serangannya, sehingga menghasilkan jumlah data seperti pada Gambar 5. Gambar 5 menghasilkan jumlah data *malicious* sebanyak 1.855.467, lalu data *benign* sebanyak 289.207 data.

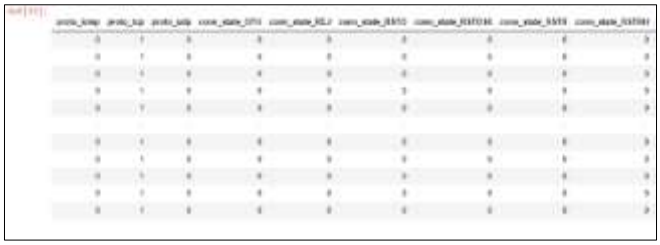
```
Out[29]: Malicious    1855467
         Benign        289207
         Name: label, dtype: int64
```

Gambar 5. Jumlah data berdasarkan *label benign* dan *malicious*

b) Data Reduction

Data *reduction* merupakan proses untuk menghapus data pada *attribute* yang kurang dominan sehingga data bisa dikurangi, namun tetap menghasilkan data yang akurat. *Dataset* yang sudah didapatkan setelah proses data *cleaning* masih terdapat kumpulan data yang kurang berpengaruh untuk dijadikan sebagai *features* maka harus dilakukan proses pembersihan untuk data beserta *attribute* yang kurang berpengaruh, pembersihan tersebut meliputi *attribute ts, uid, id.orig_h, id.orig_p, id.resp_h, id.resp_p, service, local_orig, local_resp, history* dan hanya *attribute proto, duration, orig_bytes, resp_bytes, conn_state, missed_bytes, orig_pkts, orig_ip_bytes, resp_pkts, resp_ip_bytes, label* yang tidak dihapus karena *attribute* tersebut sangat berpengaruh untuk proses penelitian ini.

Gambar 6 merupakan hasil memecah kolom *proto* dan *conn_state* menjadi kolom-kolom baru berdasarkan datanya, maka akan terbuat kolom-kolom baru dari kolom *proto* dan *conn_state* sesuai dengan datanya. Dalam *dataset* yang sudah digabungkan masih terdapat data-data *null* pada bagian kolom *duration, orig_bytes, resp_bytes*, data-data tersebut masih kurang baik untuk dijadikan bahan pengujian, kolom yang masih terdapat data-data *null* untuk dijadikan bahan pengujian diubah menjadi 0, sehingga menghasilkan seperti Gambar 7.



Gambar 6. Hasil pemisahan data kolom *proto* dan *conn_state*

```
Out[36]: duration          0
         orig_bytes        0
         resp_bytes        0
         missed_bytes      0
         orig_pkts         0
         orig_ip_bytes     0
         resp_pkts         0
         resp_ip_bytes     0
         label             0
         proto_icmp        0
         proto_tcp         0
         proto_udp         0
         conn_state_OTH    0
         conn_state_REJ    0
         conn_state_RSTO   0
         conn_state_RSTOS0 0
         conn_state_RSTR   0
         conn_state_RSTRH  0
         conn_state_S0     0
         conn_state_S1     0
         conn_state_S2     0
         conn_state_S3     0
         conn_state_SF     0
         conn_state_SH     0
         conn_state_SHR    0
         dtype: int64
```

Gambar 7. Hasil pengecekan data *null* atau *missing values*

Berdasarkan Gambar 7 sudah tidak ada kolom yang bernilai *null* atau *missing value*, setelah itu dilakukan proses cetak kolom-kolom yang akan dijadikan bahan pengujian, kolom-kolom tersebut terdapat pada Tabel 3.

Tabel 3. *Attribute* atau kolom yang digunakan

No	Nama *Attribute* atau Kolom
1	duration
2	orig_bytes
3	resp_bytes
4	missed_bytes
5	orig_pkts
6	orig_ip_bytes
7	resp_pkts
8	resp_ip_bytes
9	label
10	proto_icmp
11	proto_tcp
12	proto_udp
13	conn_state_OTH
14	conn_state_REJ
15	conn_state_RSTO

16	conn_state_RSTOS0
17	conn_state_RSTR
18	conn_state_RSTRH
19	conn_state_S0
20	conn_state_S1
21	conn_state_S2
22	conn_state_S3
23	conn_state_SF
24	conn_state_SH
25	conn_state_SHR

Tabel 3 merupakan *attribute* atau kolom yang akan digunakan pada saat *training* model dan proses pengujian yang sebelumnya sudah melewati proses data *preprocessing*. *Dataset* yang sebelumnya terdapat 20 file berbeda-beda sudah melewati tahap data *preprocessing* sehingga data yang akan digunakan sebagai bahan pengujian sudah siap untuk digunakan, dari 20 file berbeda sudah disatukan menjadi satu file yaitu dengan nama file iot23_combined.csv.

2) Classifying with K-Nearest Neighbour (K-NN)

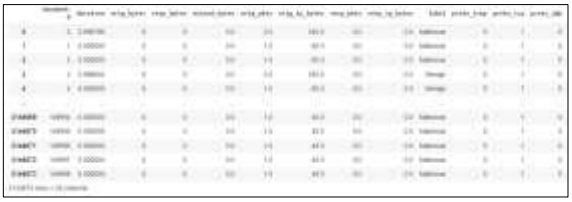
Tahapan setelah data *preprocessing* merupakan proses klasifikasi *dataset* menggunakan algoritma *K-Nearest Neighbour* (K-NN). Dalam proses klasifikasi ini ada beberapa tahapan yang harus dilakukan meliputi *import library*, *load dataset*, memilih data target dan *attribute* yang akan dijadikan sebagai *features*, mengkonversi data *benign* dan *malicious* pada kolom *label* menjadi angka 0 dan 1, *splitting dataset* dilakukan untuk membagi *dataset* menjadi 2 bagian yaitu sebagai data *training* dan data *test*, mencari nilai tetangga terdekat (K) dengan akurasi paling tinggi berdasarkan *dataset* yang sudah melewati data *preprocessing*, *training* model dilakukan untuk melatih model supaya dapat menghasilkan nilai akurasi yang baik.

a) Import Library

Sebelum melakukan *load dataset* dilakukan terlebih dahulu *import library* yang dibutuhkan diantaranya ada *library pandas* yang digunakan untuk membaca *dataset* dan mengubahnya dalam bentuk struktur tabel, selain itu *pandas* juga berfungsi untuk mengubah dimensi data, mengecek data, lalu setelah itu proses *import library numpy* yang digunakan untuk membentuk data menjadi sebuah *array*.

b) Load Dataset

Sesudah dilakukan *import library* maka langkah selanjutnya yang dilakukan merupakan proses *load dataset* atau memasukan *dataset* kedalam tools *jupyter notebook* untuk dilakukan pengujian, proses *load dataset* menampilkan data seperti pada Gambar 8.



Gambar 8. Hasil tampilan data proses *load dataset*

Pada Gambar 8. terdapat satu kolom yang perlu dihapus yaitu kolom *unnamed:0*, kolom tersebut dihapus karena tidak akan digunakan pada proses pengujian.

c) Data Target dan Features

Tahap selanjutnya yaitu pemilihan data target dan *features*, data target merupakan *attribute* atau kolom yang dijadikan sebagai target, lalu *features* merupakan *attribute* atau kolom yang menjadi karakteristik atau ciri-ciri dari target.

Tabel 4. Data *Features*

No	***Data Features***
1	duration
2	orig_bytes
3	resp_bytes
4	missed_bytes
5	orig_pkts
6	orig_ip_bytes
7	resp_pkts
8	resp_ip_bytes
9	proto_icmp
10	proto_tcp
11	proto_udp
12	conn_state_OTH
13	conn_state_REJ
14	conn_state_RSTO
15	conn_state_RSTOS0
16	conn_state_RSTR
17	conn_state_RSTRH
18	conn_state_S0
19	conn_state_S1
20	conn_state_S2
21	conn_state_S3
22	conn_state_SF
23	conn_state_SH
24	conn_state_SHR

Data *features* pada Tabel 4. merupakan *attribute* atau kolom yang menjadi karakteristik atau ciri-ciri dari target, sedangkan targetnya terdapat pada Tabel 5 yaitu kolom *label*.

Tabel 5. Data *Target*

Data *Target*
label

Attribute yang sebelumnya sudah dijadikan sebagai *target* merupakan *attribute label*, isi dari data *target* tersebut berupa *string* yang hanya memiliki 2 nilai yaitu *benign* dan *malicious*, 2 nilai tersebut perlu dikonversi kedalam sebuah nilai numerik seperti pada Tabel 6.

Tabel 6. Mengubah *label* string menjadi numerik

Label String	***Label* Numerik**
Benign	0
Malicious	1

Pada Tabel 6 data pada kolom *label* yang memiliki nilai *benign* dan *malicious* tersebut diubah menjadi bertipe numerik 0 dan 1, data *benign* diubah menjadi 0 dan data *malicious* diubah menjadi 1.

d) Splitting Dataset

Splitting dataset digunakan untuk membagi *dataset* menjadi 2 yaitu data *training* dan data *testing*, pembagian untuk data *test* nya sebesar 40% dan data *training* nya sebesar 60% dari total keseluruhan data sehingga menghasilkan jumlah masing-masing data seperti pada Gambar 9

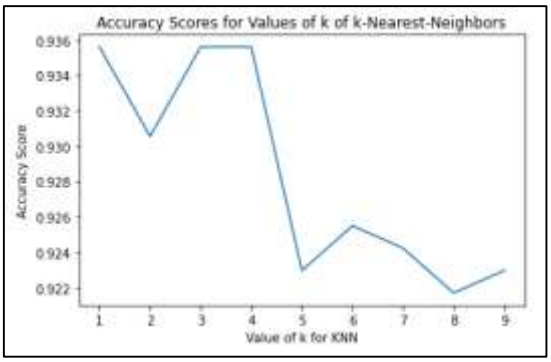
```
training dataset
(1286804, 24)
(1286804,)

testing dataset:
(857870, 24)
(857870,)
```

Gambar 9. Jumlah data pada data *training* dan data *test*

Gambar 9 menampilkan hasil pembagian *dataset* menjadi data *training* sebesar 60% dan data *testing* sebesar 40% dengan jumlah *features* sebanyak 24 *attribute*, sehingga jumlah data *training* sebanyak 1.286.804 dan data *testing* sebanyak 857.870.

e) Training Model

Training model dilakukan untuk melatih model supaya mendapatkan akurasi yang baik, model tersebut nantinya akan dilatih dengan data *training* yang sudah disiapkan sebesar 60% dari *dataset*, lalu akan di uji tingkat akurasinya terhadap data *testing* yang sudah disediakan sebesar 40%. Sebelum *training* model, dilakukan terlebih dahulu pencarian tetangga terdekat (K) supaya mendapatkan nilai akurasi tertinggi dan terbaik, dari tetangga terdekat rentang 1 sampai 10 didapatkan visualisasi seperti pada Gambar 10. Ada beberapa nilai tetangga terdekat yang memiliki akurasi baik diantaranya tetangga terdekat (K) sebanyak 1 dan 3 mempunyai akurasi yang paling baik. Maka dengan didapatkannya nilai tetangga terdekat 1 dan 3 dapat menjadi pilihan untuk dilakukan *training* model atau prediksi selanjutnya. Jadi jumlah tetangga terdekat (K) yang digunakan untuk proses *training* model adalah sebanyak 3 tetangga terdekat (K), sehingga setelah dilakukan *training* model menghasilkan nilai akurasi seperti pada Gambar 11.



Gambar 10. Visualisasi hasil pencarian tetangga terdekat (K)

```
0.9350880669565319
```

Gambar 11. Hasil nilai akurasi

Pada hasil akhir proses *training* model menggunakan jumlah tetangga terdekat sebanyak 3, maka hasil *classification report* terdapat pada Gambar 12.

	precision	recall	f1-score	support
0	0.96	0.54	0.69	115576
1	0.93	1.00	0.96	742294
accuracy			0.94	857870
macro avg	0.95	0.77	0.83	857870
weighted avg	0.94	0.94	0.93	857870

Gambar 12. *Classification report* hasil proses *training* model

Pada Gambar 12 nilai yang dihasilkan dari pengujian *dataset* menggunakan algoritma KNN didapatkan nilai *precision* untuk data bernilai 0 yaitu 0.96, lalu untuk data bernilai 1 0.93, nilai *recall* yang didapatkan dari data bernilai 0 sebesar 0.54, lalu untuk nilai data bernilai 1 sebesar 1.00, sedangkan untuk *f1-score* data bernilai 0 mendapatkan nilai akurasi sebesar 0.69 dan data bernilai 1 mendapatkan nilai akurasi sebesar 0.96, lalu didapatkan untuk nilai akurasinya sebesar 0.94 atau 94%.

3) Performance Evaluation

Pada tahap *performance evaluation*, model yang sudah melalui proses *training* akan di uji dengan data baru yang sudah ditentukan sebanyak 25 data tanpa menggunakan *attribute label*, sehingga model dapat memprediksi ke 25 data baru tersebut termasuk kedalam *benign* atau *malicious*. Prediksi data ke-1 menghasilkan prediksi yang salah karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*. Prediksi data ke-2 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*. Prediksi data ke-3 pada Gambar 13 menghasilkan prediksi yang salah karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*.

ISSN : 2745 - 9292

```
Out[119]: array([1])
```
Gambar 13. Hasil prediksi data ke-3

Prediksi data ke-4 yang terdapat pada menghasilkan prediksi yang salah karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 14.

```
Out[122]: array([1])
```
Gambar 14. Hasil prediksi data ke-4

Prediksi data ke-5 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 15.

```
Out[125]: array([0])
```
Gambar 15. Hasil prediksi data ke-5

Prediksi data ke-6 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 16.

```
Out[128]: array([0])
```
Gambar 16. Hasil prediksi data ke-6

Prediksi data ke-7 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 17.

```
Out[131]: array([0])
```
Gambar 17. Hasil prediksi data ke-7

Prediksi data ke-8 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 18.

```
Out[134]: array([0])
```
Gambar 18. Hasil prediksi data ke-8

Prediksi data Ke-9 yang terdapat pada menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 19.

```
Out[137]: array([0])
```
Gambar 19. Hasil prediksi data Ke-9

Prediksi data Ke-10 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 20.

```
Out[140]: array([0])
```
Gambar 20. Hasil prediksi data Ke-10

Prediksi data ke-12 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada gambar 21.

```
Out[146]: array([1])
```
Gambar 21. Hasil prediksi data Ke-12

Prediksi data ke-13 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 22.

```
Out[149]: array([1])
```
Gambar 22. Hasil prediksi data Ke-13

Prediksi data ke-14 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 23.

```
Out[152]: array([1])
```
Gambar 23. Hasil prediksi data Ke-14

Prediksi data ke-15 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 24.

```
Out[56]: array([1])
```
Gambar 24. Hasil prediksi data ke-15

Prediksi data ke-16 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 25.

```
Out[158]: array([1])
```
Gambar 25. Hasil prediksi data Ke-16

Prediksi data ke-17 yang terdapat pada tabel 23. menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 26.

```
Out[59]: array([1])
```
Gambar 26. Hasil prediksi data Ke-17

Prediksi data ke-18 yang terdapat pada tabel 24. menghasilkan prediksi yang salah karena prediksi yang dihasilkan bernilai 0 yang berarti *benign*, seperti pada Gambar 27.

```
Out[164]: array([0])
```

Gambar 27. Hasil prediksi data ke-18

Prediksi data ke-19 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 28.

```
Out[167]: array([1])
```

Gambar 28. Hasil prediksi data Ke-19

Prediksi data ke-20 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 29.

```
Out[170]: array([1])
```

Gambar 29. Hasil prediksi data Ke-20

Prediksi data ke-21 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 30.

```
Out[173]: array([1])
```

Gambar 30. Hasil prediksi data Ke-21

Prediksi data ke-22 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 31.

```
Out[176]: array([1])
```

Gambar 31. Hasil prediksi data Ke-22

Prediksi data ke-23 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 32.

```
Out[179]: array([1])
```

Gambar 32. Hasil prediksi data Ke-23

Prediksi data Ke-24 yang terdapat pada tabel 30. menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 33. Prediksi data ke-25 menghasilkan prediksi benar karena prediksi yang dihasilkan bernilai 1 yang berarti *malicious*, seperti pada Gambar 34.

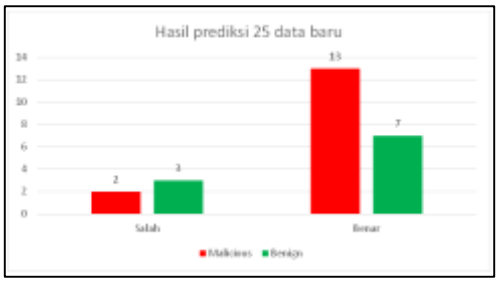
```
Out[182]: array([1])
```

Gambar 33. Hasil prediksi data Ke-24

```
Out[62]: array([1])
```

Gambar 34. Hasil prediksi data Ke-25

Hasil Prediksi 25 Data baru



Gambar 35. Visualisasi hasil prediksi 25 data baru

Gambar 35. merupakan visualisasi hasil dari prediksi 25 data baru menggunakan model yang sudah melewati proses *training*, prediksi 25 data baru tersebut menghasilkan 5 data diprediksi salah atau tidak sesuai, lalu 20 data diprediksi benar atau sesuai. 5 data yang diprediksi salah merupakan 3 data yang seharusnya bernilai *benign* dan 2 data yang seharusnya bernilai *malicious*.

Berdasarkan pemaparan yang telah dilakukan menggunakan algoritma *k-nearest neighbour* (KNN) untuk proses *machine learning* terhadap *dataset* anomali *traffic*, *dataset* aposemat IoT-23 dipilih sebagai *dataset* anomali *traffic* menghasilkan nilai akurasi sebesar 0.94 atau 94%, data yang terdapat pada *dataset* aposemat IoT-23 tersebut sebanyak 325.308.050 data sehingga perlu dilakukan data *preprocessing*, proses data *preprocessing* dapat mengurangi jumlah data yang akan digunakan tetapi tidak mengurangi tingkat akurasinya, data yang dihasilkan setelah data *preprocessing* sebanyak 2.144.674 data, kemudian data tersebut dijadikan sebagai bahan untuk proses *training* model, model yang telah di*training* menghasilkan nilai akurasi 94%, model yang sudah dilakukan proses *training* kemudian diuji terhadap 25 data baru, 25 data baru tersebut dijadikan sebagai bahan untuk pengujian dengan cara prediksi data kedalam *benign* dan *malicious* serta kesesuaiannya dengan data aslinya, hasil dari proses prediksi 25 data baru menghasilkan 20 data diprediksi benar atau sesuai, dan 5 data diprediksi salah atau tidak sesuai dengan data aslinya.

IV. KESIMPULAN

Bedasarkan hasil penelitian yang mengklasifikasikan data anomali lalu lintas jaringan IoT bersifat benign atau malicious

menggunakan dataset aposemat IoT-23 sebanyak 23 dataset yang tersedia namun hanya 20 dataset yang digunakan, maka dapat disimpulkan bahwa :

1) Dari 20 dataset yang dipilih didapatkan total data sebanyak 325.308.050 data.
2) Proses data preprocessing membuat data yang digunakan menjadi lebih sedikit dari total keseluruhan data, diambil data dari masing-masing dataset sebanyak 150.000 data sehingga menghasilkan total data sebanyak 2.144.674 data.
3) Data dibagi menjadi 2, data training sebesar 60% yang menghasilkan data sebanyak 1.286.804 data dan data testing sebesar 40% yang menghasilkan data sebanyak 857.870 data.
4) Klasifikasi atau training model dilakukan dengan menggunakan algoritma K-Nearest Neighbour (K-NN) berdasarkan data training dan data testing yang sudah ditentukan sehingga menghasilkan nilai akurasi sebesar 94%.
5) Untuk menguji tingkat akurasi yang sudah didapatkan dipilih 25 data baru dengan label data bersifat benign sebanyak 10 data, dan 15 data bersifat malicious.
6) Proses prediksi 25 data baru yang sudah dipilih lalu menghasilkan 20 data diprediksi sesuai atau benar, lalu 5 data diprediksi salah atau tidak sesuai, 3 data yang seharusnya benign diprediksi bersifat malicious, 2 data yang seharusnya malicious diprediksi bersifat benign.

REFERENSI

[1] Das R, Zekeriya, M. "*Analysis of Cyber-Attacks in IoT-based Critical Infrastructures*". International Journal Of Information Security Science, 2019.
[2] Meutia Dewi, E. "*Internet of Things – Keamanan dan Privasi*". Seminar Nasional dan Expo Teknik Elektro, 2015.
[3] Lewi, J., Joshua, E., & Maoeretz, M. "*Detection of Cyber Malware Attack Based on Network Traffic Features Using Neural Network*". Jurnal Ilmu Komputer dan Informatika, 2020.
[4] katadata.co.id. "*Microsoft : Serangan Malware di Indonesia tertinggi di Asia Pasifik*", 2 Oktober 2020. https://katadata.co.id/desysetyowati/digital/5f76 e3df376e1/microsoft-serangan-malware-di-indonesia-tertinggi-di-asia-pasifik. [Diakses, 19 Juli 2021]
[5] bssn.go.id. "*Laporan Tahunan Monitoring Keamanan Siber*", Desember 2022. https://cloud.bssn.go.id/s/Lyw8E4LxwNiJoNw. [Diakses, 1 April 2022]
[6] Arther, G., Leopold, J., & Fransiscus, V. "*Klasifikasi Malicious Websites Menggunakan Algoritma K-NN Berdasarkan Application Layers dan Network Characteristics*". Cogito Smart Journal, 2018.
[7] Yudhana, A., Sunardi, & Jaka, A. "*Algoritma K-Nn Dengan Euclidean Distance Untuk Prediksi Hasil Penggergajian Kayu Sengon*". Transmisi, 2020.
[8] Keoh, Kumar, & Tschofenig "*Securing the Internet of Things : A Standardization Perspective*". IEEE Internet Of Things Journal, 2014.
[9] Adi, T., Wahanggara, V., & Ramadana, D. "*Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis*". Jurnal Sistem & Teknologi Informasi Indonesia (Justindo), 2017.
[10] Mardi, Y. "*Data Mining : Klasifikasi Menggunakan Algoritma C4.5*". Jurnal Edik Informatika, 2017.
[11] Yustanti, W. "*Algoritma K-Nearest Neighbour untuk Memprediksi Harga Jual Tanah*". Jurnal Matematika, Statistika, & Komputasi (JMSK), 2012.
[12] Pratiwi, & Nugroho "*Prediksi Rating Film Menggunakan Metode Naïve Bayes*". Jurnal Teknik Elektro, 2016.