Terbit *online* pada laman web jurnal : http://innovatics.unsil.ac.id

Innovation in Research of Informatics (INNOVATICS)

| ISSN (Online) 2656-8993 |

Perbandingan Algoritma K-Nearest Neighbour (KNN) dan Naive Bayes pada Intrusion Detection System (IDS)

Aditya Dwi Afifaturahman [1], Firmansyah Maulana[2]

[1,2] Informatika, Fakultas Teknik, Universitas Siliwangi, Tasikmalaya, Indonesia
[1]afifaturahman@gmail.com, [2]firmansyah@unsil.ac.id

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima Redaksi: 22-03-2021
Revisi Akhir: 30-03-2021
Diterbitkan *Online*: 31-03-2021

KATA KUNCI

Classification,
IDS,
KNN,
Machine Learning,
Naive Bayes

KORESPONDENSI

Telepon: 082321960404

E-mail: afifaturahman@gmail.com

ABSTRACT

Machine learning techniques are widely used to develop Intrusion Detection Systems (IDS) to detect and classify cyber attacks at the network level and the host level in a timely and automated manner. However, many challenges arise as malicious attacks are constantly changing and occurring in very large volumes requiring a scalable solution. Therefore, this study conducted a comparison of the K-Nearest Neighbor (KNN) and Naive Bayes algorithms. The dataset used in this study is the Ddos features-IDS 2017 dataset published in 2019. This research analyzes the comparison of methods generated from the classification process based on metric accuracy, specificity and sensitivity parameters. The classification process using the K-Nearest Neighbor (KNN) and Naive Bayes algorithms, it can be concluded that the results of the three tests with a percentage split of 60%, 70% and 80% show that the K-Nearest Neighbor (KNN) algorithm gets a higher value than Naive Bayes except the error rate because the error rate indicates that the data failed to be classified properly. Testing on a percentage split of 60% KNN parameter accuracy gets a value of 99.53%, specificity 94.05%, sensitivity 75.20%, testing on a percentage split 70% KNN parameter accuracy gets a value of 99.69%, specificity 94.59%, sensitivity 78.40% and testing on percetage split 80%, KNN parameter accuracy parameter got a value of 99.70%, specificity 94.44%, sensitivity 75.85%.

1. PENDAHULUAN

Keamanan jaringan menggambarkan aspek berguna dalam bidang teknologi data dikala ini. Semakin banyak pengguna serta terus menjadi luas jangkauan komunikasi, hingga terus menjadi banyak pula kesempatan serangan. Selaku cerminan, pada survey yang dicoba oleh Lab Kaspersky 2017, 33% organisasi hadapi serbuan DDoS pada tahun 2017, dibanding dengan 17% di tahun 2016. Dari organisasi yang terserang serbuan DDoS, 20% merupakan bisnis yang sangat kecil, 33% merupakan UKM, serta 41% merupakan industri [1].

Teknik *machine learning* banyak digunakan untuk mengembangkan *Intrusion Detection System* (IDS) untuk mendeteksi dan mengklasifikasikan serangan dunia maya di tingkat jaringan dan tingkat host secara tepat waktu dan cara otomatis. Namun, banyak tantangan muncul karena serangan jahat terus berubah dan terjadi dalam volume yang sangat besar yang membutuhkan solusi yang dapat diskalakan[2]

Intrusion Detection System (IDS) biasanya menggunakan dua jenis teknik yakni *signature based intrusion detection system* dan *anomaly based intrusion detection system*[3]. Deteksi Serangan *Denial of Service* Menggunakan *Artificial Immune System* berpendapat bahwa mekanisme kinerja *dari Intrusion Detection System* (IDS) dengan menggunakan teknik *signature based* dapat mendeteksi serangan yang telah diketahui dengan efektif, tetapi belum mampu memprediksi serangan lama dengan pola yang baru. Sementara itu, *anomaly based intrusion system* bekerja dengan mengacu pada pola serangan yang ada dalam lalu lintas, tetapi bermasalah apabila lalu lintas tersebut berperilaku tidak normal sehingga tidak bisa

mengirimkan peringatan adanya serangan kepada sistem [4]. Suatu lalu lintas data dikatakan anomali, apabila terjadi peristiwa yang mencurigakan dari perspektif keamanan informasi [5].

Penelitian tentang klasifikasi *anomaly network traffic* dilakukan perbandingan metode yang dihasilkan dari proses klasifikasi bersumber pada nilai akurasi confusion matrix, precision, recall, serta f1 score. Naive Bayes, SVM Linear, SVM Polynomial serta SVM Sigmoid menciptakan persentase akurasi berturut- turut sebesar 85,055%, 99, 995%, 99, 999%, serta 99, 995%. Persentase akurasi paling tinggi diperoleh SVM Polynomial, sebaliknya Naive Bayes menciptakan persentase akurasi terendah [1].

Algoritma Naive Bayes bisa menciptakan akurasi yang optimal dengan informasi latih yang sedikit. Sebaliknya tata cara K- Nearest Neighbor diseleksi sebab tata cara tersebut tangguh terhadap informasi noise. Hasil yang didapatkan menampilkan tata cara Naive Bayes mempunyai kinerja yang lebih baik dengan tingkatan akurasi 70%, sebaliknya tata cara K- Nearest Neighbor mempunyai tingkatan akurasi yang lumayan rendah ialah 40% [6]

Algoritma *K-Nearest Neighbour* mempunyai akurasi yang tinggi dibandingkan dengan algoritma *Support Vector Machine* (SVM) dan *Neural Network* (NN) untuk kategori *accuracy, precison* dan *recall*. Hasil tersebut menunjukkan bahwa algoritma *K-Nearest Neighbour* dapat memecah data dalam keadaan *higher-feature space* sehingga dua kelas yang berbeda dapat dikelompokkan dengan baik [7] [8].

Berdasarkan dari paparan permasalahan maka akan berfokus pada klasifikasi dataset *anomaly network traffic* pada *Intrusion Detection System* (IDS) dengan membandingkan algoritma *K-Nearest Neighbour* (KNN) dan Naïve Bayes dengan parameter *metric accuracy, sensitivity* dan *specificity* sehingga akan dihasilkan nilai g-means yang pada penelitian sebelumnya belum dijabarkan pada parameter *metric* tersebut[7].

.

2. ULASAN PENELITIAN TERKAIT

Algoritma Naive Bayes bisa menciptakan akurasi yang optimal dengan informasi latih yang sedikit. Sebaliknya tata cara K- Nearest Neighbor diseleksi sebab tata cara tersebut tangguh terhadap informasi noise. Hasil yang didapatkan menampilkan tata cara Naive Bayes mempunyai kinerja yang lebih baik dengan tingkatan akurasi 70%, sebaliknya tata cara K- Nearest Neighbor mempunyai tingkatan akurasi yang lumayan rendah ialah 40% [6]

Algoritma *K-Nearest Neighbour* mempunyai akurasi yang tinggi dibandingkan dengan algoritma *Support Vector Machine* (SVM) dan *Neural Network* (NN) untuk kategori *accuracy, precison* dan *recall*. Hasil tersebut menunjukkan bahwa algoritma *K-Nearest Neighbour* dapat memecah data dalam keadaan *higher-feature space* sehingga dua kelas yang berbeda dapat dikelompokkan dengan baik[7].

Berdasarkan pemaparan ulasan terkait maka akan berfokus pada klasifikasi dataset *anomaly network traffic* pada *Intrusion Detection System* (IDS) dengan membandingkan algoritma *K-Nearest Neighbour* (KNN) dan Naïve Bayes dengan parameter *metric accuracy, sensitivity* dan *specificity* sehingga akan dihasilkan nilai g-means yang pada penelitian sebelumnya belum dijabarkan pada parameter *metric* tersebut[7].

.

3. METODOLOGI

Tahapan yang dimulai dari proses pengumpulan data, analisis permasalahandan pencarian solusi, implementasi solusi sampai pada proses penarikan kesimpulan yang dijelaskan pada gambar 1.

Gambar 1. Tahapan proses

Data merupakan penunjang yang diperoleh melalui studi literatur dan observasi dengan pengamatan langsung. Studi literatur berisi uraian tentang teori, temuan dan bahan penelitian lainnya yang diperoleh dari jurnal nasional maupun jurnal internasional yang berupa survey paper dan technical paper. Observasi data dilakukan dengan mencari data yang tepat untuk melakukan penelitian.

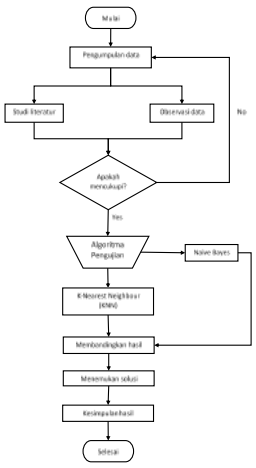
3.1 Analisis Permasalahan dan Pencarian Solusi

Tahapan analisis permasalahan dan oencarian solusi merupakan tahap pengembangan yang dilalkukan setelah pengumpulan data. Masalah yang ditemukan terdapat pada proses literatur dan observasi data, kemudian diamatai dan mencari solusi berdasarkan perkembangan ilmu pengetahuan dan teknologi yang ada. Masalah yang ditemukan adalah deteksi anomaly network traffic. Solusi yang dipilih adalah menerapkan algoritma *K-Nearest Neighbour* (KNN) dengan menggunaan tools WEKA

3.2 Implementasi Solusi

Langkah yang dilakukan pada tahap implementasi solusi adalah mencari dataset yang diperoleh setelah melakukan observasi data. Kemudian data yang telah diperoleh diubah dari format “csv” menjadi “arff”. Dataset yang ada dilakukan pemangkasan karena data yag didapat terlalu banyak dan data yang didapat sebanyak 9998 data.

Dataset yang telah didapatkan diolah menggunakan tools WEKA. Data diproses dengan Explorer dan Classify Rule, kemudian proses selanjutnya adalah menerapkan algoritma K-Nearest Neighbour (KNN) sebagai classifier. Hasil dari proses pengujian berupa item summary, detailed accuracy by class dan confusion matrix. Berikut adalah gambaran implementasi solusi dari penelitian ini.



Gambar 2. Alur Implementasi Solusi

Gambar 2 merupakan proses menemukan solusi dimana pada tahapan tersebut meliputi data collection, data preprocessing, klasifikasi menggunakan WEKA, hasil akurasi berupa precision dan recall.

Tabel summary yang menggambarkan hasil dari proses pengujian dataset secara garis besaryang dijelaskan pada tabel 1.

Tabel 1. *Overall Summary*

No.	Kategori
1.	*Correctly Classified Instances*
2.	*Incorrectly Classified Instances*
3.	*Kappa statistic*
4.	*Mean absolute error*
5.	*Root mean squared error*
6.	*Relative absolute error*
7.	*Root relative squared error*
8.	*Total Number of Instances*

Tabel tingkat keakuratan pemrosesan data berdasarkan kelas yang digunakan pada tabel 2.

Tabel 2. *Detailed Accuracy By Class*

No.	Kategori
1.	*TP Rate*
2.	*Fp Rate*
3.	*Precision*
4.	*Recall*
5.	*F-Measure*
6.	*MCC*
7.	*ROC Area*
8.	*PRC Area*

Tabel parameter pengujian merupakan proses data perbandingan berdasarkan parameter yang digunakan pada tabel 3.

Tabel 3. Parameter Pengujian

No.	Parameter
1.	*Accuracy*
2.	*Precision*
3.	*Recall*
4.	*Specificity*
5.	*Sensitivity*
6.	*Error Rate*

Tabel merupakan contoh bentuk Confusion Matrix dari hasil pemrosesan data.

Tabel 4. *Confusion Matrix*

a	*b*	*< - - Classified as*
9	0	*a = yes*
1	4	*b = no*

3.3 Penarikan Kesimpulan

Penarikan kesimpulan menjadi tahapan terakhir dari proses dimana hasil yang diperoleh adalaha nilai keakuratan dari hasil uji coba penerapan algoritma K-Nearest Neighbour (KNN) dan algoritma Naive Bayes terhadap data yang memuat anomaly network traffic pada dataset Instrusion Detection System (IDS)

4. HASIL DAN PEMBAHASAN

4.1 Pengumpulan Data

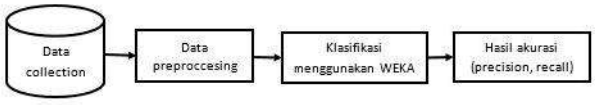
Observasi data dilakukan dengan melakukan pencarian dataset yang berkaitan dengan Intrusion Detection System (IDS). Dataset diperoleh dari kaggle.com/dataset dengan dataset yang bernama alldays_ddos.

4.2 Analisis Permasalahan dan Pencarian Solusi

Objek yang diteliti yakni deteksi anomali network traffic pada Instrusion Detection System (IDS) dan masalah yang ditemukan adalah mendeteksi lalu lintas jaringan yang bersifat anomali pada Intrusiom Detection System (IDS). Penyelesaian yang diseleksi untuk menanggulangi kasus tersebut merupakan menerapkan pengujian tingkatan akurasi dari algoritma K-Nearst Neighbour (KNN) untuk mendeteksi lalu lintas jaringan yang bersifat anomali pada Intrusiom Detection System (IDS). Tools yang digunakan sebagai penunjang penelitian ini adalah WEKA 3.9, Microsoft Excel 2016, dan sublime text.

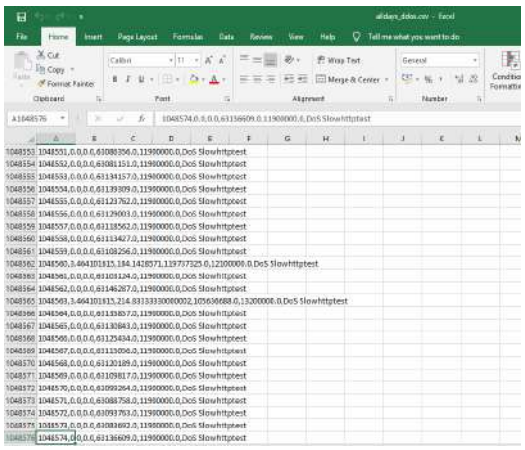
4.3 Pemrosesan Data

Pemrosesan data ini merupakan tahap optimasi klasifikasi terhadap dataset yang telah diperoleh dari kaggle.com/dataset dengan dataset yang bernama alldays_ddos. Kemudian dirumuskan di Microsoft Exel dengan format CSV.



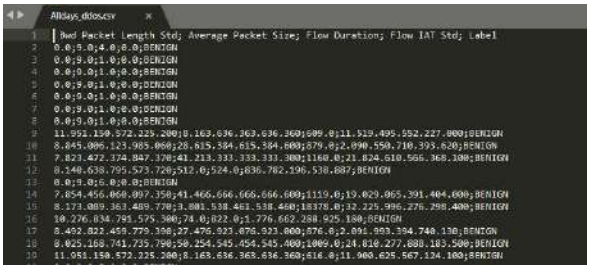
Gambar 3. Alur Proses Menemukan Solusi

Gambar 3 merupakan alur menemukan solusi yang terdiri dari data collection, data preproccesing, klasifikasi WEKA dan hasil akurasi. Data collection didapatkan dari kaggle.com/dataset yang selanjutnya diproses menggunakan microsoft excel untuk merubah format CSV ke ARFF agar bisa diproses oleh WEKA dan selanjutnya akan diproses sehingga menghasilkan hasil akurasinya.



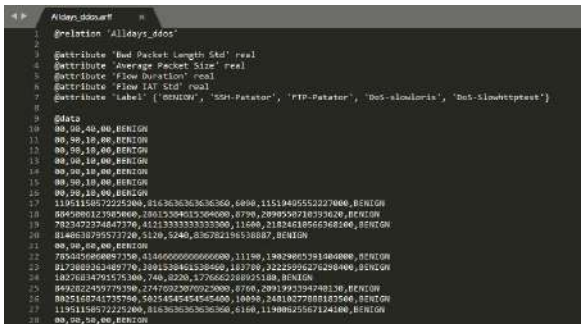
Gambar 4. dataset Alldays_ddos

Sebelum di proses dengan tools WEKA format CSV harus di konversi terlebih dahulu ke dalam format ARFF agar dapat di proses oleh tools WEKA.



Gambar 5. dataset format CSV

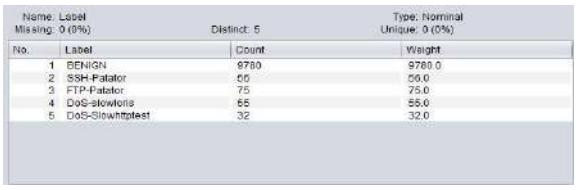
Gambar 5 merupakan dataset yang masih berformat CSV yang akan di konversikan ke dalam format ARFF yang di buka menggunakan notepad atau bisa menggunakan text editor yang lainnya seperti notepad++, sublime text dan lain-lain.



Gambar 6. dataset dengan format ARFF

Gambar 6 menunjukan proses konversi CSV ke dalam format ARFF yang dibuka dengan text editor sublime text. Merubah data CSV kedalam format ARFF mengharuskan mengubah model atau bentuk konten data. Line 1 yang berisi @relation Allday_ddos merupakan judul atau topik dari dataset. Atribut-atribut yang sebelumnya berbentuk tabel, diubah menjadi parafrase script seperti pada line 3 sampai line 7.

Proses optimasi klasifikasi tersebut dilakukan dengan menggunakan tools WEKA untuk melakukan prepoccess terhadap dataset dan buka allday_ddos yang telah di konversikan kedalam format ARFF. Berikut merupakan data dari Allday_ddos berdasarkan label BENIGN, SSH-Patator, FTP Patator, Dos-slowris dan Dos-Slowhttptest.

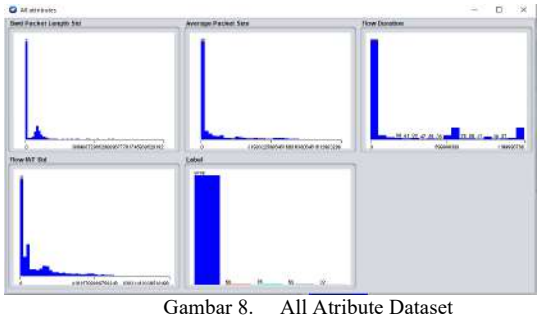


Name: Label
Missing: 0 (0%)
Distinct: 5
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	BENIGN	9780	9780.0
2	SSH-Patator	56	56.0
3	FTP-Patator	75	75.0
4	DoS-slowloris	55	55.0
5	DoS-Slowhttptest	32	32.0

Gambar 7. Perhitungan Atribut pada Label Network Traffic

Gambar 7 merupakan perhitungan jumlah atribut kategori berdasarkan label network traffic yang isinya adalah BENIGN, SSH-Patator, FTP-Patator, Dos-slowloris, Dos-slowhttptest. Perhitungan tersebut menunjukan bahwa dari 9998 data yang ada pada dataset sebanyak 9780 data benign, 56 data SSH-Patator, 75 data FTP-Patator , 55 data Ddos-slowloris, 32 data Ddos-slowhttptest.



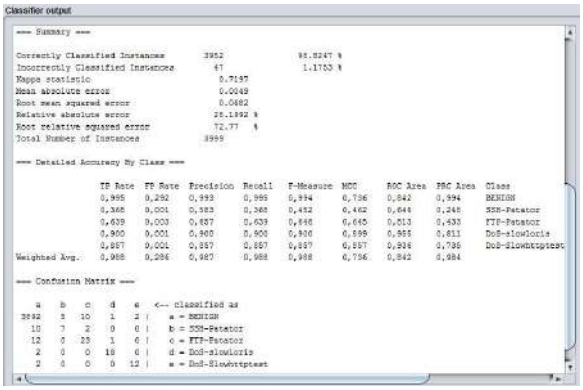
Gambar 8. All Atribute Dataset

Gambar 8 merupakan seluruh atribut yang ada pada dataset Allday_ddos atribut tersebut meliputi Bwd packet length Std, Average Packet Size, Flow duration, Flow IAT Std dan Label.

Pengujian dilakukan sebanyak 3 kali dengan pengujian pertama dilakukan pada percentage split 60%, pengujian kedua dilakukan terhadap percentage split 70% dan pengujian ketiga dilakukan terhadap percentage split 80%. Proses split bertujuan untuk menghindari proses overviting karena jumlah data yang banyak dari dataset dan maksud dari hasil split ialah untuk membagi antara data latih dan data uji.

4.1 Pengujian 1

Data yang sebelumnya telah dilakukan preprocess kemudian di split menjadi 60%. Proses tersebut merupakan pembagian data latih dan uji dimana 60% merupakan data latih dan 40% data uji. Hasil pengujian ditunjukan pada gambar 4.14.



Gambar 9. *Classifier Output* KNN Pengujian 1

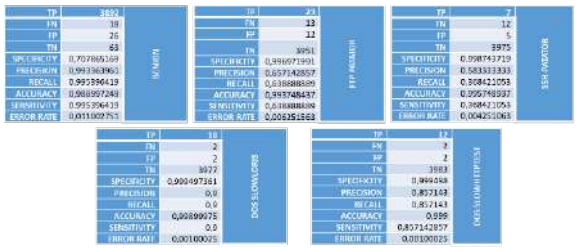
Gambar 9 merupakan hasil pengujian data dari algoritma KNN yang telah di split menjadi 60%. *Classifier output* terdapat 3 bentuk yaitu *Summary*, *Detailed Accuracy By Class* dan *Confusion Matrix*. Penelitian ini hanya menggunakan 2 bentuk *classifier output* yaitu *Detailed Accuracy By Class* dan *Confusion Matrix*.

```
=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 3892    5   10    1    2 |    a = BENIGN
   10    7    2    0    0 |    b = SSH-Patator
   12    0   23    1    0 |    c = FTP-Patator
    2    0    0   18    0 |    d = DoS-slowloris
    2    0    0    0   12 |    e = DoS-Slowhttptest
```

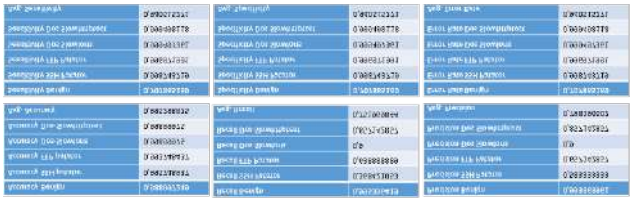
Gambar 10. *Confusion Matrix* KNN Pengujian 1

Gambar 12 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label BENIGN terdapat 3.892 data TP (*True Positive*), 26 data FP (*False Positive*), 63 data TN (*True Negative*) dan 18 data FN (*False negative*) yang terdeksi. Label SSH-Patator terdapat 7 data TP, 5 data FP, 3975 data TN, 12 FN yang terdeksi Label FTP-Patator terdapat 23 data TP, 12 data FP, 3951 data TN dan 13 data FN yang terdeksi. Label Dos-slowloris terdapat 18 data TP, 2 data FP, 3977 data TN dan 2 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 12 data TP, 2 data FP, 3983 data TN dan 2 data FN yang terdeksi.



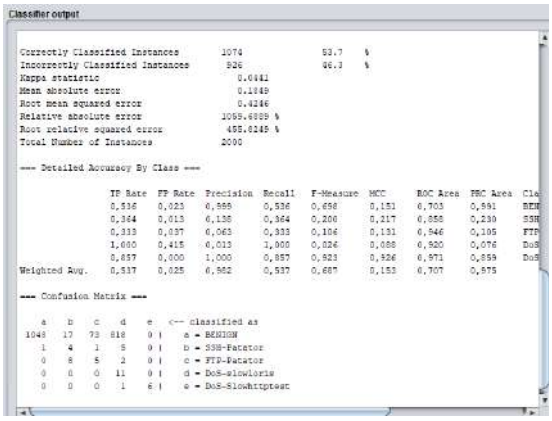
Gambar 11. Parameter KNN Pengujian 1

Gambar 11 merupakan parameter klasifikasi setiap label pada K-Nearest Neighbour dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity* dan *error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix*.



Gambar 12. *Average* pengujian 1 KNN

Gambar 12 merupakan nilai rata-rata *Accuracy, Precision, Recall, specificity, sesnsitivity* dan *Error Rate* dari Algortima KNN dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai setiap label dibagi jumlah label itu sendiri.



Gambar 13. *Classifier Output* Naive Bayes Pengujian 1

Gambar 13 merupakan hasil pengujian data dengan algoritma Naive Bayes yang telah di split menjadi 60%. *Classifier output* terdapat 3 bentuk yaitu *Summary, Detailed Accuracy By Class* dan *Confusion Matrix*. Penelitian ini hanya menggunakan 2 bentuk *classifier output* yaitu *Detailed Accuracy By Class* dan *Confusion Matrix.*

```
    a    b    c    d    e   <-- classified as
 1048   17   73  818    0 |    a = BENIGN
    1    4    1    5    0 |    b = SSH-Patator
    0    8    5    2    0 |    c = FTP-Patator
    0    0    0   11    0 |    d = DoS-slowloris
    0    0    0    1    6 |    e = DoS-Slowhttptest
```

Gambar 14. *Confusion Matrix* NB Pengujian 1

Gambar 14 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label BENIGN terdapat 2151 data TP, 4 data FP, 85 data TN dan 1755 data FN yang terdeksi. Label SSH-Patator terdapat 6 data TP, 107 data FP, 85 data TN, 13 FN yang terdeksi Label FTP-Patator terdapat 12 data TP, 78 data FP, 3881 data TN dan 24 data FN yang terdeksi. Label Dos-slowloris terdapat 20 data TP, 1604 data FP, 2371 data TN dan 0 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 12 data, 1 data FP, 3980 data TN dan 2 data FN yang terdeksi.

Gambar 15. *Parameter* NB Pengujian 1

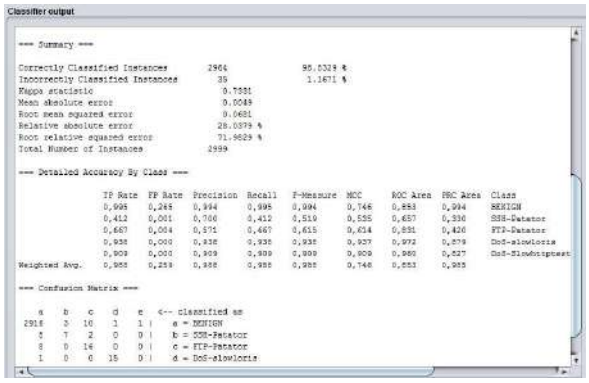
Gambar 15 merupakan parameter klasifikasi setiap label pada Naive Bayes dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity dan error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix.*

Gambar 16. *Average* Pengujian 1 NB

Gambar 16 merupakan nilai rata-rata *Accuracy, Precision, Recall, specificity, sesnsitivity* dan *Error Rate* dari Algortima NB dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai setiap label dibagi jumlah label itu sendiri.

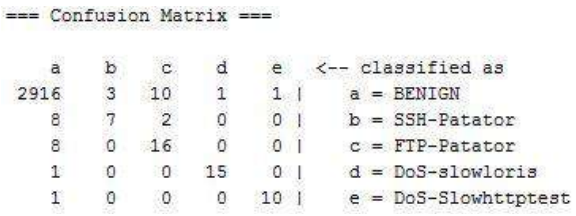
4.2 Pengujian 2

Data yang sebelumnya telah dilakukan preprocess kemudian di split menjadi 70%. Proses tersebut merupakan pembagian data latih dan uji dimana 70% merupakan data latih dan 30% data uji. Hasil pengujian ditunjukan pada gambar 19.



Gambar 17. *Classifier Output* KNN Pengujian 2

Gambar 17 merupakan hasil pengujian data yang telah di split menjadi 70%. *Classifier output* terdapat 3 bentuk yaitu *Summary, Detailed Accuracy By Class* dan *Confusion Matrix.* Pengujian kedua ini hanya menggunakan 2 bentuk *classifier output* yaitu *Detailed Accuracy By Class* dan *Confusion Matrix.*

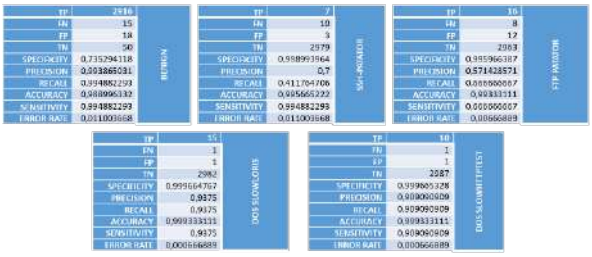


```
=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 2916    3   10    1    1 |    a = BENIGN
    8    7    2    0    0 |    b = SSH-Patator
    8    0   16    0    0 |    c = FTP-Patator
    1    0    0   15    0 |    d = DoS-slowloris
    1    0    0    0   10 |    e = DoS-Slowhttptest
```

Gambar 18. *Confusion Matrix* KNN Pengujian 2

Gambar 18 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label BENIGN terdapat 2916 data TP, 18 data FP, 50 data TN dan 15 data FN yang terdeksi. Label SSH-Patator terdapat 7 data TP, 10 data FP, 2979 data TN, 10 FN yang terdeksi Label FTP-Patator terdapat 16 data TP, 12 data FP, 2963 data TN dan 8 data FN yang terdeksi. Label Dos-slowloris terdapat 15 data TP), 1 data FP, 2982 data TN dan 1 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 10 data TP, 1 data FP, 2987 data TN dan 1 data FN yang terdeksi.

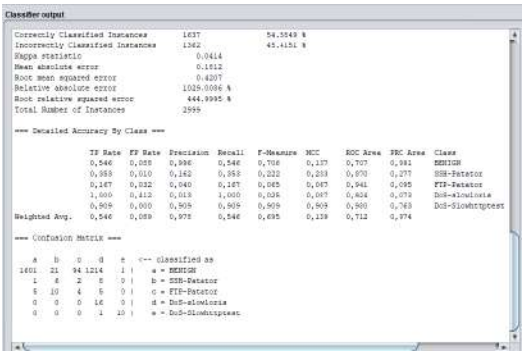


Gambar 19. *Parameter* KNN Pengujian 2

Gambar 19 merupakan parameter klasifikasi setiap label pada KNN dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity* dan *error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix.*

Gambar 20. *Average* Pengujian 2 KNN

Gambar 20 merupakan nilai rata-rata dari Algortima KNN dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai setiap label dibagi jumlah label itu sendiri.



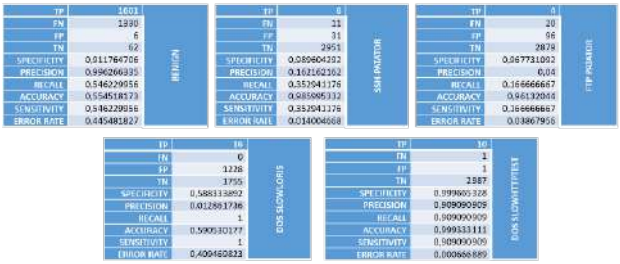
Gambar 21. *Classifier ouput* NB Pengujian 2

Gambar 21 merupakan hasil pengujian data dengan algoritma Naive Bayes yang telah di split menjadi 70%. *Classifier output* terdapat 3 bentuk yaitu *Summary, Detailed Accuracy By Class* dan *Confusion Matrix*. Penelitian ini hanya menggunakan 2 bentuk classifier output yaitu *Detailed Accuracy By Class* dan *Confusion Matrix.*

```
    a    b    c    d    e   <-- classified as
 1601   21   94 1214    1 |    a = BENIGN
    1    6    2    8    0 |    b = SSH-Patator
    5   10    4    5    0 |    c = FTP-Patator
    0    0    0   16    0 |    d = DoS-slowloris
    0    0    0    1   10 |    e = DoS-Slowhttptest
```

Gambar 22. *Confusion Matrix* NB Pengujian 2

Gambar 22 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label *BENIGN* terdapat 1601 data TP, 6 data FP, 62 data TN dan 1330 data FN yang terdeksi. Label SSH-Patator terdapat 6 data TP, 31 data FP, 2951 data TN, 11 FN yang terdeksi Label FTP-Patator terdapat 4 data TP, 96 data FP, 2879 data TN dan 20 data FN yang terdeksi. Label Dos-slowloris terdapat 16 data TP, 1228 data FP, 1755 data TN dan 0 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 10 data, 1 data FP, 2987 data TN dan 1 data FN yang terdeksi.



Gambar 23. *Parameter* NB Pengujian 2

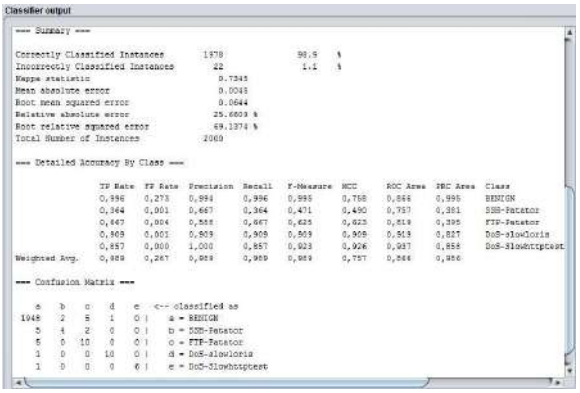
Gambar 23 merupakan parameter klasifikasi pada label *benign* pada Naive Bayes dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity* dan *error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix*.

Gambar 24. *Average* Pengujian 2 NB

Gambar 24 merupakan nilai rata-rata akurasi dari Algortima Naive Bayes dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai dari setiap label dibagi jumlah label itu sendiri.

4.3 Pengujian 3

Data yang sebelumnya telah dilakukan preprocess kemudian di split menjadi 80%. Proses tersebut merupakan pembagian data latih dan uji dimana 80% merupakan data latih dan 20% data uji. Hasil pengujian ditunjukan pada gambar 27.



Gambar 25. *Classifier Output* KNN Pengujian 3

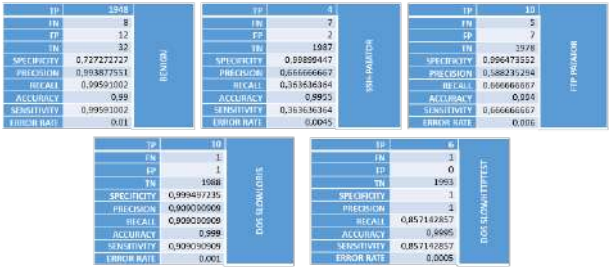
Gambar 25 merupakan hasil pengujian data yang telah di split menjadi 80%. *Classifier output* terdapat 3 bentuk yaitu *Summary, Detailed Accuracy By Class* dan *Confusion Matrix*. Penelitian ini hanya menggunakan 2 bentuk classifier output yaitu *Detailed Accuracy By Class dan Confusion Matrix*.

```
=== Confusion Matrix ===

    a    b    c    d    e   <-- classified as
 1948    2    5    1    0 |    a = BENIGN
    5    4    2    0    0 |    b = SSH-Patator
    5    0   10    0    0 |    c = FTP-Patator
    1    0    0   10    0 |    d = DoS-slowloris
    1    0    0    0    6 |    e = DoS-Slowhttptest
```

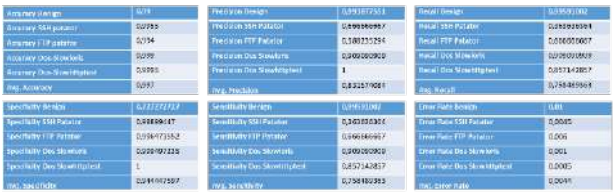
Gambar 26. *Confusion Matrix* KNN Pengujian 3

Gambar 26 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label *BENIGN* terdapat 1948 data TP, 12 data FP, 32 data TN dan 8 data FN yang terdeksi. Label SSH-Patator terdapat 4 data TP, 2 data FP, 1987 data TN, 7 FN yang terdeksi Label FTP-Patator terdapat 10 data TP, 7 data FP, 1987 data TN dan 5 data FN yang terdeksi. Label Dos-slowloris terdapat 10 data TP, 1 data FP, 1988 data TN dan 1 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 6 data 0 data FP, 1993 data TN dan 1 data FN yang terdeksi.



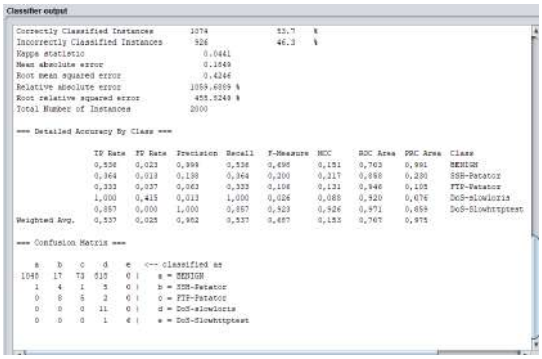
Gambar 27. *Parameter* KNN Pengujian 3

Gambar 27 merupakan parameter klasifikasi pada label *benign* pada *K-Nearest Neighbour* dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity* dan *error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix*.



Gambar 28. *Average* Pengujian 3 KNN

Gambar 28 merupakan nilai rata-rata akurasi dari Algortima KNN dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai dari setiap label dibagi jumlah label itu sendiri.



Gambar 29. *Classifier Output* NB Pengujian 3

Gambar 29 merupakan hasil pengujian data dengan algoritma Naive Bayes yang telah di *split* menjadi 80%. *Classifier output* terdapat 3 bentuk yaitu *Summary, Detailed Accuracy By Class* dan *Confusion Matrix.* Penelitian ini hanya menggunakan 2 bentuk *classifier output* yaitu *Detailed Accuracy By Class* dan *Confusion Matrix.*

```
   a    b    c    d    e   <-- classified as
1048   17   73  818    0 |    a = BENIGN
   1    4    1    5    0 |    b = SSH-Patator
   0    8    5    2    0 |    c = FTP-Patator
   0    0    0   11    0 |    d = DoS-slowloris
   0    0    0    1    6 |    e = DoS-Slowhttptest
```

Gambar 30. *Confusion Matrix* NB Pengujian 3

Gambar 30 merupakan *confusion matrix* dimana setiap label *network traffic* diklasifikasikan menggunakan variabel a,b,c,d dan e. Label *BENIGN* terdapat 1048 data TP, 1 data FP, 43 data TN dan 908 data FN yang terdeksi. Label SSH-Patator terdapat 4 data TP, 25 data FP, 1964 data TN, 7 FN yang terdeksi Label FTP-Patator terdapat 5 data TP, 74 data FP, 1911 data TN dan 10 data FN yang terdeksi. Label Dos-slowloris terdapat 11 data TP, 826 data FP, 1163 data TN dan 0 data FN yang terdeksi. Label Dos-Slowhttptest terdapat 6 data, 0 data FP, 1993 data TN dan 1 data FN yang terdeksi.

Gambar 31. *Parameter* KNN Pengujian 3

Gambar 31 merupakan parameter klasifikasi pada label *benign* pada Naive Bayes dimana perhitungan tersebut diperoleh dari rumus *specificity, precision, recall, accuracy, sensitivity* dan *error rate* setelah mengetahui nilai dari TP, FP, TN, FN pada *confusion matrix*.

Gambar 32. *Average* Pengujian 3 NB

Gambar 32 merupakan nilai rata-rata akurasi dari Algortima Naive Bayes dimana nilai rata-rata tersebut diperoleh dari penjumlahan nilai dari setiap label dibagi jumlah label itu sendiri.

4.4 Penarikan Kesimpulan

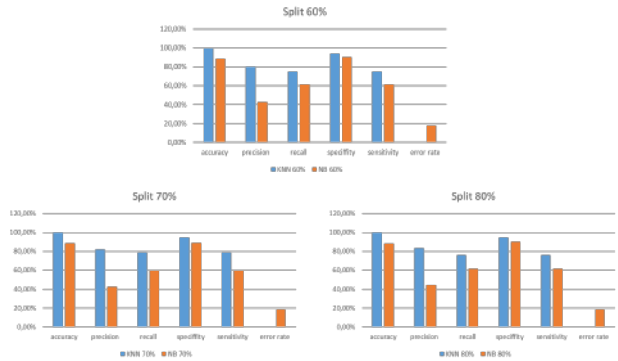
Algoritma	Accuracy	Precision	Recall	Specificity	Sensitivity	Error Rate	Avg.
K-NN 60%	99,53%	79,82%	75,20%	94,05%	75,20%	0,47%	70,55%
Naive Bayes 60%	88,55%	42,40%	61,14%	90,09%	61,14%	17,96%	54,23%

Algoritma	Accuracy	Precision	Recall	Specificity	Sensitivity	Error Rate	Avg.
K-NN 70%	99,69%	82,24%	78,40%	94,59%	78,40%	0,47%	72,14%
Naive Bayes 70%	88,43%	42,41%	59,50%	89,14%	59,50%	18,17%	53,47%

Algoritma	Accuracy	Precision	Recall	Specificity	Sensitivity	Error Rate	Avg.
K-NN 80%	99,70%	83,16%	75,85%	94,44%	75,85%	0,44%	71,43%
Naive Bayes 80%	88,21%	44,27%	61,80%	90,24%	61,80%	18,52%	54,63%

Gambar 33. Rekap Data Pengujian

Gambar 33 merupakan rekap data dari pengujian yang telah dilakukan *percentage split* 60%, 70% dan 80%.



Gambar 34. Diagram Perbandingan *percentage Split* 60%,70% dan 80%

Gambar 34 merupakan diagram perbandingan antara *K-Nearest Neighbour* dengan Naive Bayes dimana *K-Nearest Nighbour* lebih tinggi dibanding Naive Bayes dari Parameter *Accuracy, Precision, Recall, Specifity* dan *Sensitivity* sedangakan pada *Error Rate* Naive Bayes lebih tinggi dari *K-Nearest Neighbour*.

5. KESIMPULAN

Berdasarkan hasil penelitian pada Instrusion Detection System (IDS) yang memuat data dari berbagai kategori *anomaly traffic* yang selanjutnya dilakukan proses klasifikasi dengan menggunakan algoritma *K-Nearest Neighbour* (KNN) dan Naive Bayes, maka dapat disimpulkan hasil dari ketiga pengujian dengan *percentage split* 60%, 70% dan 80% menunjukkan bahwa algoritma K-Nearest Neighbour (KNN) mendapatkan nilai yang lebih tinggi dari Naive Bayes kecuali *error rate* karena *error rate* menunjukan bahwa data gagal diklasifikasi dengan baik. Pengujian pada percetage split 60% KNN parameter *accuracy* mendapatkan nilai 99,53%, *specificity* 94,05%, *sensitivity* 75,20%, pengujian pada percentage split 70% KNN parameter *accuracy* mendapatkan nilai 99,69%, *specificity* 94,59%, *sensitivity* 78,40% dan pengujian pada percetage split 80% parameter KNN parameter *accuracy* mendapatkan nilai 99,70%, *specificity* 94,44%, *sensitivity* 75,85% untuk melakukan klasifikasi pada dataset *anomaly network traffic* dengan menggunakan bantuan *tools* WEKA.

.

DAFTAR PUSTAKA

[1] M. F. Fibrianda and A. Bhawiyuga, "Analisis Perbandingan Akurasi Deteksi Serangan Pada Jaringan Komputer Dengan Metode Naïve Bayes Dan Support Vector Machine (SVM)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 9, pp. 3112–3123, 2018.

[2] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[3] Candra Adi Winanto, "Deteksi Serangan Denial of Service Menggunakan Artificial Immune System," vol. 2, no. 1, pp. 456–459, 2016.

[4] B. Agarwal and N. Mittal, "Hybrid Approach for Detection of Anomaly Network Traffic using Data Mining Techniques," *Procedia Technology*, vol. 6, pp. 996–1003, 2012.

[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, no. 1–2, pp. 18–28, 2009.

[6] R. N. Devita, H. W. Herwanto, and A. P. Wibawa, "Perbandingan Kinerja Metode Naive Bayes dan K-Nearest Neighbor untuk Klasifikasi Artikel Berbahasa indonesia," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 5, no. 4, p. 427, 2018.

[7] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," *Proceedings - 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018*, no. Ml, pp. 29–35, 2018.

[8] I. N. Rizkiana, A. Rahmatulloh, and R. Gunawan, "Penerapan Metode Clustering K-Means Untuk Menentukan Nilai Burst Header Packet Flooding Attack Pada Optical Burst Switching," *Indonesian Journal of Applied Informatics*, vol. 4, no. 2, p. 107, Aug. 2020.

BIODATA PENULIS

Aditya Dwi Afifaturahman
Mahasiswa Prodi Informatika Universitas Siliwangi Tasikmalaya

Firmansyah Maulana SN., S.T., M.Kom
Dosen Prodi Informatika Universitas Siliwangi Tasikmalaya