e-ISBN 978-623-93343-3-8
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 April 2021

Perbandingan Metode Klasifikasi *Naive Bayes, Decision Tree* Dan *K-Nearest Neighbor* Pada Data *Log Firewall*

Gebrina Divva Meuthia Zulma[1], Angelika[2], Nurul Chamidah[3]
[1,2,3]Teknik Informatika / Universitas Pembangunan Nasional Veteran Jakarta
Jl. Rs. Fatmawati, Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
[1]gebgebdmz@gmail.com, [2]angellie2298@gmail.com, [3]nurul.chamidah@upnvj.ac.id

Abstrak. Penelitian ini bertujuan untuk membandingkan tiga metode klasifikasi, yaitu *K-Nearest Neighbor, Naive Bayes*, *Decision Tree.Dataset log firewall* bisa didapatkan pada situs UCI Machine Learning. Agar mendapatkan hasil yang maksimal, *dataset* harus melalui tahap *preprocessing*.Setelah itu, data divalidasi menggunakan *StratifiedKFold* dengan n_splits sebanyak 10 .Untuk pengujian performa pada ke tiga algoritma tersebut, *precision*, *recall*, akurasi serta nilai ROC AUC nya untuk digunakan sebagai bahan pertimbangan metode klasifikasi manakah yang terbaik serta dapat memproses *dataset* yang disajikan. *Decision Tree* menjadi metode dengan akurasi 100% dan nilai AUC sebesar 75%, disusul *K-Nearest Neighbor* dengan akurasi 99% dan nilai AUC sebesar 74%, namun *Naive Bayes* dianggap tidak latyak untuk memproses data karena meskipun nilai akurasinya 96%, nilai AUC termasuk rendah yaitu 46%. Hasil dari penelitian ini diharapkan dapat menjadi acuan untuk penelitian metode klasifikasi lain, baik menggunakan data yang sama maupun menggunakan klasifikasi yang berbeda.

Kata Kunci: *K-Nearest Neighbor, Naive Bayes*, *Decision Tree*.

1. Pendahuluan

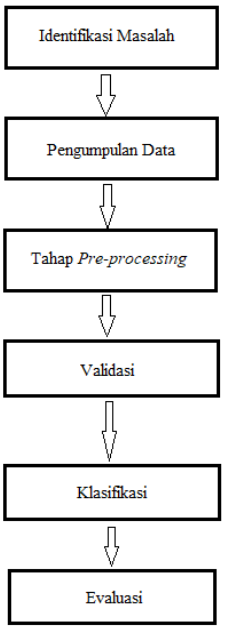
Penggunaan internet sebagai alat untuk berkomunikasi menjadi sangat vital setelah virus Covid-19 muncul pada akhir Desember 2019. Masyarakat terpaksa untuk mengurangi aktivitas di luar rumah demi menekan laju penyebaran virus. Hal ini menyebabkan kegiatan seperti sekolah, bekerja, dan belanja harus dilakukan secara online. Untuk mengamankan arus data yang masuk maupun keluar, *firewall* menjadi pelindung utama dari setiap perangkat di dunia. *Firewall* akan mencatat setiap data yang masuk, dan melalui *firewall* pengguna dapat memblokir situs tertentu untuk melindungi sistem dari hal seperti Trojan horses, *virus*, *phishing* dan *spyware* [1]. Penelitian kali ini berguna untuk menentukan algoritma manakah yang terbaik untuk mengklasifikasi data Log Firewall menggunakan metode klasifikasi *K-Nearest Neighbor, Naive Bayes* dan *Decision Tree*.

Dalam penelitian klasifikasi ini, mula-mula dilakukan tahapan *pre-processing* untuk mempersiapkan *dataset* [2]. Bila tahap *pre-processing* selesai, data harus divalidasi, salah satunya dengan menggunakan StratifiedKFold. Setelah itu, klasifikasi dilakukan pada data *log firewall*. Klasifikasi *dataset* lalu di evaluasi menggunakan *confusion matrix*, untuk diketahui *precision, recall*, akurasi dan ROC. Jika keseluruhan nilai evaluasi sudah didapatkan, maka hasil nya dapat digunakan untuk menarik kesimpulan manakah klasifikasi yang bisa memproses serta paling baik untuk digunakan pada data *log firewall*.

Terdapat banyak penelitan terkait perbandingan metode klasifikasi *K-Nearest Neighbor, Naive Bayes* dan *Decision Tree*. Penelitian yang dilakukan oleh Wahyuningsih dan Utari [3] menyebutkan bahwa dalam memprediksi kelayakan pemberian kredit, *Decision Tree* memiliki akurasi tertinggi yaitu 92,21%, disusul dengan *Naive Bayes* dengan akurasi 81,32% dan *K-Nearest Neighbor* dengan nilai 81,82%. Pada penelitian oleh Marutho [4], Decision Tree ditetapkan sebagai algoritma yang berfungsi secara maksimal dalam mengklasifikasikan *dataset* untuk digunakan pada laporan *water level* Jakarta dengan 96.56 % sebagai akurasinya, dengan *K-Nearest Neighbor* memiliki akurasi sebesar 95,98% dan *Naive Bayes* sebesar 94,32%. Sementara itu, penelitian yang dilakukan oleh Setiyorini & Asmono [5] mencoba untuk melakukan komparasi terhadap metode ketiga klasifikasi yang disebutkan di atas terhadap kinerja siswa dalam lembaga pendidikan yang bertujuan untuk meningkatkan kinerja siswa agar mendapatkan pendidikan yang berkualitas, dengan hasil evaluasi berupa akurasi yang didapatkan dari analisa berbagai faktor-faktor yang masuk dalam penentu kinerja siswa mendapatkan hasil akurasi sebesar 78,85% pada Decision Tree, sedangkan pada Naïve Bayes didapatkan akurasi 77,69%, dan pada K-Nearest Neighbor mendapat akurasi yang lebih baik yaitu sebesar 79,31%.

2. Metode Penelitian

Tahapan penelitian dituangkan pada gambar dibawah ini:



Gambar 1. Tahapan Penelitian

2.1. Identifikasi Masalah

Tahapan ini dilakukan dengan cara membaca literasi seperti buku dan jurnal guna menemukan masalah yang bisa diangkat dari penelitian kali ini dengan dataset berupa *log* yang tercatat pada saat penggunaan arus internet yang melewati *firewall* yang berfungsi untuk pelindung perangkat. Dataset ini akan digunakan untuk melakukan klasifikasi dan pemilihan algoritma yang paling relevan dengan penelitian. Untuk algoritma yang digunakan dalam penelitian di antaranya adalah Naïve Bayes, Decision Tree dan K-Nearest Neighbor untuk hasil evaluasi.

2.2. Pengumpulan Data

Data yang digunakan untuk penelitian ini diambil dari *website* UCI *Machine Learning Repository* berupa data *firewall log files* yang digunakan untuk melakukan penelitian ini. Dataset mencakup 12 atribut berupa fitur pada data log firewall. Atribut Action digunakan sebagai kelas untuk penelitian ini. Untuk mendapatkan hasil yang lebih dapat dipercaya, data ini harus diolah melalui beberapa tahapan, yaitu tahap *pre-processing* yang berfungsi untuk membersihkan data yang terduplikat dan *noise*, validasi data, serta normalisasi data sehingga tahap ini membutuhkan waktu yang lebih lama.

2.3. Tahap *Pre-processing*

Pre-processing dilakukan untuk memproses data menjadi agar menjadi bersih dan bisa digunakan dengan maksimal sehingga mendapatkan hasil evaluasi yang lebih baik**.** Pada tahap ini dilakukan beberapa proses seperti melakukan pengecekan melalui seleksi dan pemilihan pada tiap atribut, normalisasi data dan penanganan pada *missing value*. Untuk tahap ini, hal pertama yang perlu dilakukan yaitu melakukan pengecekan *missing value* pada dataset yang digunakan. Untuk mempermudah proses klasifikasi, kelas Action disarankan untuk dirubah dari huruf menjadi angka.

Setelah proses *labeling* kelas Action, *dataset* harus melalui normalisasi. *Data scaling* atau normalisasi data merupakan teknik mengubah nilai numerik pada suatu dataset skala yang lebih umum, tanpa merubah perbedaan dalam rentang nilai [6]. Normalisasi yang digunakan pada dataset penelitian ini yaitu normalisasi *Min-Max*. Normalisasi *Min-Max* berguna untuk mentransformasi dataset asli secara linear sehingga tercipta nilai yang seimbang antar data sebelum dan sesudah proses [7]. Rumus yang digunakan normalisasi ini yaitu:

$$Min - Max(x) = \frac{minRange + (x - minValue)(maxRange - minRange)}{\max Value - \min Value} \quad (1)$$

2.4. Validasi

Pada tahap ini, *StratifiedKFold* digunakan untuk memvalidasi data. *StratifiedKFold* merupakan salah satu variasi dari *KFold*, dimana data dibagi menjadi data latih dan data uji. Tahap dari *StratifiedKFold* yaitu data yang akan digunakan di acak, lalu di pisah sebanyak n_splits [8]. Nilai n_splits yang digunakan yaitu sebanyak 10, dengan perbandingan penggunaan 1 *instance* digunakan untuk *testing*, 9 sisanya digunakan untuk *training*.

2.5. Klasifikasi

Untuk metode klasifikasi yang digunakan dalam penelitian ini yaitu Naïve Bayes, Decision Tree dan K-Nearest Neighbor. Untuk indikator evaluasi menggunakan kelas atribut Action dengan 4 macam kelas yaitu ‘allow’, ‘deny’, ‘drop’, dan ‘reset-both’.

2.5.1. Naive Bayes

Menurut Syarli dan Muin [9], *Naive Bayes* merupakan salah satu algoritma pembelajaran induktif yang paling efektif dan efisien untuk *machine learning* dan *data mining*. Algoritma ini berfungsi dengan cara memprediksi peluang di masa depan berdasarkan masa lalu. Rumus Naïve Bayes berada dibawah ini:

$$P(C|X) = \frac{P(x|c)P(c)}{\mathrm{P(x)}} \quad (2)$$

2.5.2. Decision Tree

Decision Tree merupakan struktur *flowchart* yang berbentuk seperti pohon, dimana setiap simpul dalam menandakan suatu tes pada atribut, dengan cabang yang dihasilkan menunjukkan hasil tes, dan simpul daun merepresentasikan persebaran kelas [10].

2.5.3. K-Nearest Neighbor

Algoritma ini memiliki metode dengan cara mengklasifikan suatu objek dengan data pelatihan terdekat dengan objek tersebut. Data yang digunakan untuk pelatihan lalu direpresentasikan ke ruang berdimensi banyak, dimana masing-masing dimensi mewakilkan fitur dari suatu dataset [11]. Pada penelitian ini, n_neighbors yang digunakan yaitu sebanyak 3, dengan *weights* yang digunakan yaitu *distances*. Rumus pada *K-Nearest Neighbor* untuk menghitung seberapa dekat c kelas klasifikasi dengan k buah tetangga:

$$Jarak = \sqrt{\sum_{i=1}^{n} (X_{latih}^{i} - X_{test})^2} \quad (3)$$

2.6. Evaluasi

Untuk mengetahui performa dari setiap algoritma klasifikasi yang diuji, dilakukan penghitungan *confusion matrix*. Hasil dari *confusion matrix* akan digunakan untuk menghitung besar dari nilai *precision, recall* dan akurasinya.

Precision

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (4)$$

Recall

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

Akurasi

$$Akurasi = \frac{True\ Positive + True\ Negative}{False\ Positive + False\ Negative + True\ Positive + True\ Negative} \quad (6)$$

Setelah itu, *Receiver Operating Charateristics* (ROC) akan digunakan untuk memvisualisasikan secara dua dimensi performa dari setiap dari klasifikasi yang diujikan, dimana garis *horizontal* merupakan nilai *false positive*, sedangkan garis *vertical* berupa *true positive* [12]. Nilai *Area Under Curve* (AUC) merupakan area dibawah grafik ROC. Untuk pengkategorian hasil AUC, nilai kualitas suatu klasifikasi berdasarkan nilai AUC nya bisa dilihat pada tabel dibawah ini.

Tabel. 1. Kriteria AUC

Nilai AUC	Penjelasan
90%-100%	Excellent
80%-90%	Good
70%-80%	Fair
60%-70%	Poor
<60%	Failure

3. Analisis dan Pembahasan

3.1. Pengumpulan Data

Data ini diambil dari *website* UCI Machine Learning Repository , dan digunakan pada penelitian yang dilakukan oleh Kaya dan Ertam [13] dengan metode *Support Vector Machine* (SVM). Data memiliki jumlah sebanyak 65532 entri, dengan atribut sebanyak 12, salah satunya digunakan sebagai kelas pada penelitian kali ini yaitu atribut *Action*.

Tabel. 2. Penjelasan fitur pada dataset.

Fitur	Deskripsi
Source Port	*Source Port* pada *Client*
Destination Port	*Destination Port* pada *Client*
Nat Source Port	*Network Address Translation Source Port*
Nat Destination Port	*Network Address Translation Destination Port*
Elapsed Time (sec)	Lama waktu berjalan
Bytes	Jumlah keseluruhan *Bytes*
Bytes Sent	*Bytes* yang dikirim
Bytes Received	*Bytes* yang diterima
Packets	Jumlah keseluruhan *Packet*
pkts_sent	*Packet* yang dikirim
pkts_received	*Packet* yang diterima
Action	Kelas (allow, deny, drop, reset-both)

Tabel. 3. Pengelompokan *Dataset* berdasarkan kelas *Action* beserta jumlahnya.

Jenis Kelas	Jumlah
allow	37640
deny	14988
drop	12850
reset-both	54

Total	65532

3.2. Tahap *pre-processing*

Bila *dataset* sudah didapatkan, pastikan tidak terdapat *missing value* didalamnya . Hasil pengecekan *missing value* pada *dataset* adalah sebagai berikut:

```
Source Port             0
Destination Port        0
NAT Source Port         0
NAT Destination Port    0
Action                  0
Bytes                   0
Bytes Sent              0
Bytes Received          0
Packets                 0
Elapsed Time (sec)      0
pkts_sent               0
pkts_received           0
dtype: int64
```

Gambar. 1. Pengelompokan *Dataset* berdasarkan kelas *Action* beserta jumlahnya.

Kelas-kelas pada fitur Action bisa dirubah dari huruf lalu menjadi angka. Setelah itu, dilakukan proses normalisasi min-max pada dataset tersebut.

Tabel. 4. Kelas Asli berisi kelas sebelum dirubah, hasil dari konversi ditampilkan pada kolom Setelah Konversi.

Kelas Asli	Setelah Konversi
allow	0
deny	1
drop	2
reset-both	3

3.3. Klasifikasi dan Evaluasi

Hasil klasifikasi yang didapatkan dibuat dalam tabel sebagai berikut:

Tabel. 5. Hasil *precision* dan *recall* pada *Naive Bayes.*

Kelas	*Precision*	*Recall*
allow	100%	59%
deny	91%	97%
drop	94%	100%
reset-both	0%	0%

Pada tabel 4 terlihat bahwa hasil evaluasi algoritma *Naïve Bayes*, terdapat nilai 0% pada kedua nilai *precision* dan *recall* pada kelas reset-both. Meskipun bisa mengklasifikasikan kelas allow, deny dan drop dengan baik, kelas reset-both menjadi kelas yang gagal untuk terklasifikasi oleh metode *Naïve Bayes*.

Tabel. 6. Hasil *precision* dan *recall* pada *Decision Tree*.

Kelas	*Precision*	*Recall*
allow	100%	100%
deny	99%	99%
drop	99%	100%
reset-both	25%	29%

Decision Tree memiliki hasil *precision* dan *recall* dengan nilai diatas 25% pada hasil keduanya. Hanya nilai evaluasi pada kelas reset-both yang tidak memiliki nilai yang sempurna, dengan nilai *precision* sebesar 25% dan *recall* sebesar 29%

Tabel. 7. Hasil *precision* dan *recall* pada *K-Nearest Neighbor.*

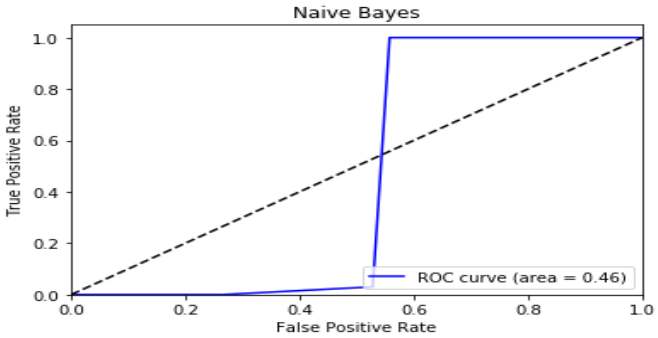
Kelas	*Precision*	*Recall*
allow	100%	99%
deny	98%	98%
drop	99%	100%
reset-both	0%	0%

K-Nearest Neighbor memiliki nilai *precision* dan *recall* yang tinggi pada kelas allow, deny dan drop, tetapi nilai 0% pada kelas reset-both pada kedua nilai *precision* dan *recall*.

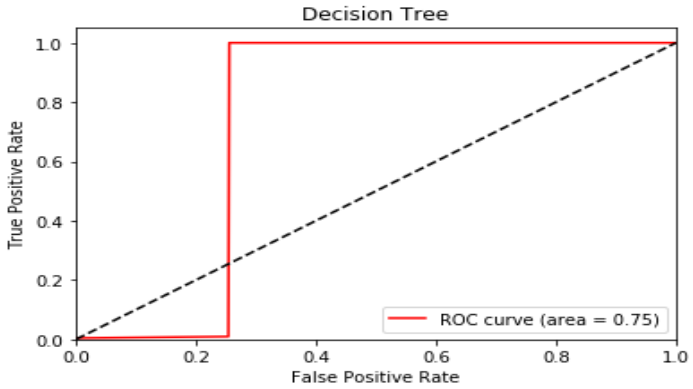
Tabel. 8. Perbandingan akurasi pada metode klasifikasi *K-Nearest Neighbor, Naive Bayes* dan *Decision Tree.*

Metode klasifikasi	Nilai akurasi
Naive Bayes	96%
Decision Tree	100%
K-Nearest Neighbor	99%

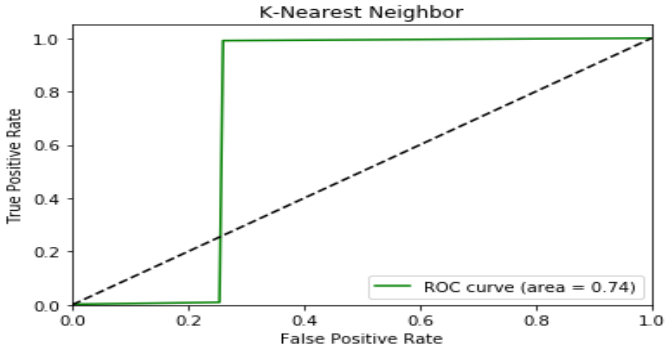
Setelah membandingkan nilai *precision* dan *recall,* nilai akurasi akan dibandingkan untuk mengetahui seberapa akurat klasifikasi yang digunakan.Hasil perhitungan akurasi pada ketiga metode klasifikasi menunjukkan bahwa *Decision Tree* memiliki nilai tertinggi yaitu 100%, disusul dengan *K-Nearest Neighbor* dengan akurasi 99% dan *Naïve Bayes* dengan presentase akurasi sebesar 96%. Meskipun nilai Akurasi pada *Decision Tree,* nilai *recall* dan *precision* yang dihasilkan tidak semuanya mencapai 100%, sehingga menjadi sangat penting untuk menghitung ROC dan AUC setiap metode klasifikasi pada penelitian ini Dibawah ini merupakan gambar kurva ROC pada setiap metode klasifikasi:



Gambar. 2. Kurva ROC dan nilai AUC pada *Naïve Bayes.*



Gambar. 3. Kurva ROC dan nilai AUC pada *Decision Tree.*



Gambar. 4. Kurva ROC dan nilai AUC pada *K-Nearest Neighbor.*

Gambar grafik AUC diatas menunjukkan bahwa *Naïve Bayes* termasuk pada kateogori *failure* dengan nilai AUC sebesar 46% pada kriteria AUC yang telah ditetapkan diatas. Sedangkan *Decision Tree* memiliki nilai 75% pada AUC, dan termasuk pada kategori *fair*. Pada *K-Nearest Neighbor* , nilai AUC yang dihasilkan yaitu sebesar 74% termasuk pada kategori *fair.* Hal ini menunjukkan bahwa meskipun nilai akurasinya yang tinggi, *Naïve Bayes* tidak dapat digunakan untuk pengklasikasian data log firewall, sedangkan klasifikasi *Decision Tree* dan *K-Nearest Neighbor* bisa digunakan nilai akurasi yang tinggi dan nilai AUC yang tergolong *fair*.

4. Penutup

Penelitian ini bertujuan untuk membandingkan tiga model algoritma untuk mengklasifikasi data log firewall, yaitu *K-Nearest Neighbor, Naive Bayes* dan *Decision Tree*. Setelah dilakukan tahap preprocessing, validasi data menggunakan StratifiedKfold dan klasifikasi dataset , Decision Tree memiliki performa terbaik dengan akurasi sebesar 100% dengan nilai AUC sebesar 75% dan kategori AUC yang didapatkan yaitu *fair* . Algoritma yaitu *K-Nearest Neighbor* masih bisa digunakan untuk pengklasifikasian data log firewall karena nilai akurasinya sebesar 99% dan hasil AUC nya tergolong *fair* dengan nilai sebesar 74% .Namun metode klasifikasi *Naïve Bayes* tidak disarankan untuk digunakan karena meskipun memiliki akurasi yang tinggi yaitu 96%, evaluasi menggunakan ROC dan AUC menunjukkan bahwa nilai AUC yang dihasilkan pada algoritma ini yaitu sebesar 46%, dimana pada kriteria AUC termasuk pada kategori *failure*. Dari penjelasan diatas, bisa ditarik kesimpulan bahwa *Decision Tree* merupakan algoritma terbaik untuk mengolah data *log firewall*.

5. Referensi

[1] Oktaviani. 2007. *Mengenal Sistem Firewall*. 1-12.
[2] Kana, Saputra., Siahaan, Andysah Putera U. 2007. *Klasifikasi Data Minuman Wine Menggunakan Algoritma K-Nearest Neighbor*. 2–4.
[3] Wahyuningsih, Sri., Utari, Dyah Retno. 2018. Perbandingan Metode *K-Nearest Neighbor*, *Naïve Bayes* dan *Decision Tree* untuk Prediksi Kelayakan Pemberian Kredit. *Konferensi Nasional Sistem Informasi 2018*, 619-623.
[4] Marutho, Dhendra. (2019). Perbandingan Metode Naïve Bayes, KNN, Decision Tree Pada Laporan Water Level Jakarta. *INFOKAM*. Vol. 15 No. 2:90-97.
[5] Setiyorini, Tyas,. Asmono, Rizky T. 2018. Komparasi Metode *Decision Tree, Naïve Bayes* dan *K-Nearest Neighbor* Pada Klasifikasi Kinerja Siswa, *Jurnal TECHNO Nusa Mandiri.* Vol. 15, No.2.
[6] Ambarwari, Agus., Adrian, Qadhli Jafar., Herdiyeni, Yeni. (2020). Analisis Pengaruh Data Scaling Terhadap Performa Algoritme Machine Learning untuk Identifikasi Tanaman. *Jurnal Resti*. Vol 4 No. 1:117-122.
[7] Nasution, Darnisa Azzahra, et. al. 2019. *"Perbandingan Normalisasi Data Untuk Klasifikasi Wine Menggunakan Algoritma KNN"* dalam *Journal of Computer Engineering System and Science*, Vol. 4, No. 1:78-82.
[8] Z^2 Little. 2020. *StratifiedKFold v.s KFold v.s StratifiedShuffleSplit*.
[9] Syarli., Muin, Asrul Ashari. 2016. Metode *Naive Bayes* Untuk Prediksi Kelulusan (Studi Kasus: Data Mahasiswa Baru Perguruan Tinggi). *Jurnal Ilmiah Ilmu Komputer*, Vol. 2, No. 1:22-26.
[10] Kasih, Patmi. 2019. Pemodelan Data Mining *Decision Tree* Dengan Classification Error Untuk Seleksi Calon Anggota Tim Paduan Suara. *Innovation in Research of Informatics*, Vol. 1, No.2:63-69.
[11] Yustanti, Wiyli. 2012. Algoritma K-Nearest Neighbour untuk Memprediksi Harga Jual Tanah. *Jurnal Matematika, Statistika dan Komputasi*, Vol. 9, No.1:57-68.
[12] Pudjiarti, Eni. 2016. Prediksi Spam Email Menggunakan Metode *Support Vector Machine* dan *Particle Swarm Optimization. Jurnal Pilar Nusa Mandiri*, Vol.12, No.2:171-181.
[13] Kaya, Mustafa., Ertam, Fatih. 2018. Classification of Firewall Log Files with Multiclass Support Vector Machine. *IEEE*.