Perbandingan Kinerja Metode Seleksi Fitur untuk Mendeteksi Aktivitas Trojan

Performance Comparison of Feature Selection Methods for Detecting Trojan Activity

Muhammad Rijal[1)], Amil Ahmad Ilham[2)*], Ady Wahyudi Paundu[3)]

[1,2,3]Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin
[1,2,3]Jl. Poros Malino, Gowa, Sulawesi Selatan, Indonesia
muhammadrijal503@gmail.com[1)], Corresponding Author : amil@unhas.ac.id[2)*], adywp@unhas.ac.id[3)]

Diterima : 1 Juli 2022 || Revisi : 16 November 2022 || Disetujui: 20 Desember 2022

Abstrak - Virus merupakan program berbahaya yang dapat merugikan. Salah satu virus paling berbahaya adalah virus trojan, dimana virus trojan bersembunyi pada perangkat pengguna tanpa diketahui keberadaanya. Virus trojan dapat sangat sulit diketahui keberadaannya karena bersembunyi pada perangkat jaringan dan menyamar sebagai bagian dari perangkat. Namun ketika perangkat jaringan terinfeksi oleh serangan virus trojan maka aktivitas yang terjadi pada jaringan akan berbeda dari aktivitas biasanya. Pada aktivitas jaringan terdapat beragam parameter yang menyebabkan pengklasifikasian membutuhkan waktu lebih lama dalam melakukan prediksi. Pada penelitian ini dilakukan berbagai perbandingan algoritma reduksi fitur antara *Coefficient Correlation*, *Information Gain*, PCA, dan LDA serta melakukan pengujian kombinasi algoritma model klasifikasi (*Random Forest, Decision Tree*, KNN, *Naïve Bayes, AdaBoost*) terbaik untuk mendeteksi aktivitas trojan pada jaringan internet secara lebih cepat untuk meningkatkan keamanan terhadap virus trojan. Hasil dari penelitian menunjukkan klasifikasi dengan akurasi maksimal dengan waktu terbaik diperoleh kombinasi antara *Coefficient Correlation*, *Information Gain*, dan PCA menggunakan klasifikasi *Decision Tree*, dengan menggunakan kombinasi seleksi fitur dan metode klasifikasi tersebut diperoleh akurasi 99% dan waktu prediksi 0,0033 detik.

Kata Kunci : Reduksi Fitur, Klasifikasi, Optimasi Waktu Prediksi, Virus Trojan, *Random Forest, Decision Tree*, KNN, Naïve Bayes, AdaBoost, *Coefficient Correlation*, *Information Gain*, PCA, dan LDA

Abstrak *- Viruses are malicious programs that can be harmful. One of the most dangerous viruses is the trojan virus, where the trojan virus hides on the user's device without being aware of its existence. Trojan viruses can be very difficult to spot because they hide on network devices and disguise themselves as part of the device. However, when a network device is infected by a trojan virus attack, the activities that occur on the network will be different from usual activities. In network activity, there are various parameters that cause classification to take longer to predict. In this study, various comparisons of feature reduction algorithms between Coefficient Correlation, Information Gain, PCA, and LDA were carried out and tested the combination of classification model algorithms (Random Forest, Decision Tree, KNN, Naïve Bayes, AdaBoost) to detect the best trojan activity on the internet network. faster to increase security against trojan viruses. The results of the study show that the classification with maximum accuracy with the best time is obtained by a combination of Coefficient Correlation, Information Gain, and PCA using the Decision Tree classification, using a combination of feature selection and classification methods obtained 99% accuracy and prediction time of 0.0033 seconds.*

Keywords: *Feature Reduction, Classification, Prediction Time Optimization, Trojan Virus, Random Forest, Decision Tree, KNN, Naïve Bayes, AdaBoost, Coefficient Correlation, Information Gain, PCA, dan LDA*

PENDAHULUAN

Hampir seluruh aktifitas dilakukan menggunakan komputer dan jaringan. Sehingga jenis dari virus komputer semakin beragam dan terus meningkat. Dalam beberapa tahun terakhir, jumlah serangan jaringan secara bertahap meningkat seiring dengan perkembangan Internet(Yeh et al., 2021). Berdasarkan data yang diterbitkan oleh pusat operasi keamanan siber nasional terhadap serangan siber di Indonesia pada rentang waktu 1 januari – 12 april 2020 terdapat

total 88.414.296 serangan trojan, dimana dari serangan yang diterima 56% diantaranya adalah serangan Trojan. Berdasarkan data tersebut serangan trojan merupakan serangan yang sering dilakukan dan sangat penting untuk diketahui keberadaannya.

Virus Trojan bersembunyi pada perangkat pengguna. Nama lengkapnya adalah "*Trojan Horse*" yang merupakan program perangkat lunak berbasis *remote control* yang mengendalikan komputer lain berdasarkan program tertentu. Dengan fungsi implantasinya atau karakteristik aksesori pembawa virus, virus ini dapat masuk ke komputer pengguna untuk mencuri informasi pribadi dan kata sandi, merusak data, atau menghancurkan *file*. Virus *Trojan Horse* dapat dikendalikan oleh penyerang untuk melakukan perintah, sehingga sistem komputer dapat dihancurkan karena operasi ilegal mereka. Trojan horse sangat berbahaya karena penyerang dapat mengendalikan komputer pengguna dengan virus dari jarak jauh untuk mencuri atau mengubah *file*, memata-matai informasi sistem, mencuri berbagai perintah dan kata sandi, dan bahkan memformat perangkat keras pengguna. Selain itu, virus *Trojan Horse* biasanya merekam operasi *keyboard* melalui catatan *keyboard*, dan kemudian memperoleh akun dan kata sandi E-bank. Penyerang dapat langsung mencuri informasi berharga pengguna dengan mendapatkan akun dan kata sandi (Kok et al., 2019).

Virus trojan dapat sangat sulit diketahui keberadaannya secara langsung disebabkan karena trojan dapat menyamar sebagai bagian dari perangkat jaringan sehingga dapat menyembunyikan diri di komputer dan menyebarkannya di internet sementara pengguna tidak dapat melihatnya (Han & Tan, 2010). Virus trojan tampaknya tidak berbahaya karena menyembunyikan dirinya sedemikian rupa sehingga penyerang dapat mengontrol dan melakukan bentuk kerusakan yang berbahaya, seperti merusak data , menghapus data, dan lebih buruk lagi(Al-Saadoon & Al-Bayatti, 2011).

Trojan biasanya sulit ditemukan, kecuali dengan menjalankan program yang terinfeksi, misalnya untuk memeriksa bahwa file tertentu masih ada. Trojan dapat berada pada perangkat dalam jangka waktu yang lama, namun efeknya tidak langsung dirasakan dan akan diketahui pada waktu tertentu(Thimbleby et al., 1998). Namun ketika perangkat jaringan terinfeksi oleh serangan virus trojan maka aktivitas yang terjadi pada jaringan akan berbeda dari aktivitas biasanya (Tidak Terinfeksi Virus Trojan), hal tersebut disebabkan oleh adanya aktivitas lain yang terjadi pada jaringan selain aktivitas yang seharusnya dilakukan pada jaringan.

Salah satu cara mendeteksi keberadaan virus Trojan adalah dengan melakukan pemantauan aktivitas jaringan. Sejumlah algoritma telah dikembangkan dalam statistik dan pembelajaran mesin untuk mendukung dan memungkinkan pencarian informasi dari data yang besar(Plotnikova et al., 2020). Namun demikian untuk tujuan ini, diperlukan banyak parameter (fitur) yang harus dipantau untuk membedakan apakah aktivitas tersebut normal atau melibatkan Trojan. Banyaknya parameter yang harus dipantau akan berpengaruh pada dua hal yaitu membebani proses pengambilan data parameter jaringan dan membebani proses komputasi deteksi/klasifikasi aktivitas normal atau tidak (Trojan).

Beberapa penelitian terkait dengan deteksi Trojan antara lain Penelitian yang dilakukan oleh Tatsuki Kurihara, dkk "*Hardware-Trojan Classification based on the Structure of Trigger Circuits Utilizing Random Forests*" (Kurihara & Togawa, 2021), penelitian tersebut memiliki beberapa kelebihan seperti menambah jumlah fitur untuk memperbanyak informasi yang dapat meningkatkan akurasi pengklasifikasian. Namun demikian terdapat beberapa kekurangan dari penelitian terkait sebelumnya dimana dengan menambah jumlah fitur akan meningkatkan komputasi yang terjadi sehingga waktu pengklasifikasian akan menjadi lebih lama. Adapun penelitian lain yang dilakukan oleh Chee Hoo Kok, Dkk "*Net Classification Based on Testability and Netlist Structural Features for Hardware Trojan*"(Kok et al., 2019); dan penelitian yang dilakukan oleh Kristina P. Sinaga, Dkk "*Entropy K-Means Clustering With Feature Reduction Under Unknown Number of Clusters*"(Sinaga et al., 2021), pada penelitian tersebut dilakukan perhitungan nilai relevansi untuk menghilangkan fitur tidak relevan yang bertujuan meningkatkan hasil akurasi. Meski pada penelitian sebelumnya menghilangkan fitur yang tidak relevan, fitur yang memiliki korelasi tinggi antara satu fitur dan fitur lainnya tetap digunakan meski salah satu dari fitur yang memiliki korelasi tinggi dapat diabaikan karena memiliki nilai pengaruh yang sama. Selain itu, penelitian tersebut tidak berfokus terhadap kecepatan waktu prediksi dan hanya bertujuan meningkatkan hasil akurasi. Oleh sebab itu optimasi perlu dilakukan untuk meningkatkan peluang terdeteksinya virus trojan lebih cepat. Optimasi perlu dilakukan ketika terdapat data kompleks yang berskala besar(Zhan et al., 2022).

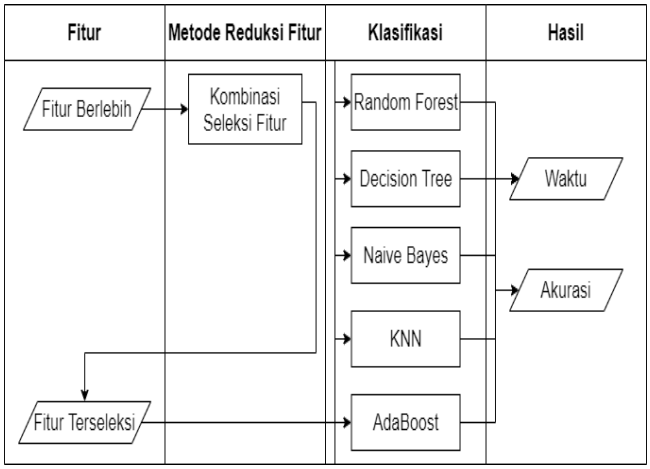
Penelitian yang dilakukan akan mengevaluasi keseluruhan parameter-parameter yang digunakan untuk mereduksi jumlah fitur dengan menggunakan seleksi fitur. Penggunaan parameter yang tidak tepat dapat menyebabkan buruknya akurasi prediksi dan dapat membuat klasifikasi tidak stabil, parameter yang besar dan kompleks menjadi salah satu tantangan utama dalam optimasi fitur(Lu et al., 2018). Penelitian ini diharapkan mampu mengatasi klasifikai trojan yang memiliki data berskala besar yang memiliki karakteristik, seperti volume data yang besar, data bervariasi, dan kecepatan pemrosesan data yang diperlukan tinggi(Choi et al., 2018). Untuk memaksimalkan pereduksian fitur akan dilakukan eliminasi terhadap fitur redundan dengan menggunakan teknik *Coefficient Correlation* untuk melihat kemiripan fitur yang ada sehingga dapat mengurangi lebih banyak jumlah fitur yang akan digunakan dalam pembangunan model klasifikasi. Selain fitur redundan, fitur yang tidak relevan akan dihilangkan. Dalam kebanyakan kasus, terdapat beberapa fitur yang tidak relevan yang selalu mempengaruhi hasil pengelompokan dan dapat menghasilkan hasil pengelompokan yang salah(Yang & Sinaga, 2019). Untuk mengurangi jumlah fitur yang tersisa digunakan metode LDA atau PCA yang merupakan metode pengurangan dimensi dan sangat berguna ketika data yang ada berukuran besar. Tujuannya adalah untuk mengurangi jumlah dimensi dengan kehilangan informasi yang minimal(Kherif & Latypova, 2019). Terdapat dua teknik pengurangan dimensi yang dilakukan yaitu supervised dan unsupervised learning, dimana LDA merupakan reduksi dimensi *supervised learning* dan PCA *unsupervised learning*(Tharwat et al., 2017). Model klasifikasi akan sangat berpengaruh terhadap proses pengklasifikasian sehingga diperlukan model klasifikasi yang tepat terhadap fitur terseleksi yang akan digunakan. Pada penelitian ini, untuk dapat memaksimalkan waktu komputasi dan akurasi prediksi dilakukan evaluasi untuk menentukan kombinasi seleksi fitur(*Coefficient Correlation, Information Gain*, PCA, dan LDA) dan algoritma model klasifikasi(*Random Forest, Decision Tree*, KNN, *Naïve Bayes, AdaBoost*) paling tepat terhadap aktivitas jaringan trojan. Dengan menggunakan berbagai seleksi fitur tersebut dapat dilakukan pereduksian terhadap fitur yang akan digunakan untuk mengurangi beban komputasi saat dilakukan pengklasifikasian dan melakukan perbandingan terhadap beberapa algoritma model klasifikasi untuk melihat perbedaan waktu prediksi yang dapat terjadi dari berbagai kombinasi seleksi fitur. Dengan waktu prediksi yang lebih cepat pada pendeteksian virus trojan dapat meningkatkan keamanan lebih baik lagi, semakin cepat virus dideteksi maka semakin cepat tindakan dapat dilakukan sehingga semakin meminimalisir kerusakan yang dapat terjadi. Oleh sebab itu, penelitian ini dilakukan untuk meningkatkan keamanan dengan melakukan pengujian terhadap berbagai kombinasi seleksi fitur dan algoritma model klasifikasi untuk memperoleh hasil prediksi yang lebih cepat.

METODOLOGI PENELITIAN

Pada penelitian ini akan dilakukan perancangan sistem identifikasi aktivitas trojan dengan teknik klasifikasi berbasis seleksi fitur. Jumlah data yang berdimensi tinggi yang tersedia untuk umum di internet semakin mudah didapatkan namun fitur yang tersedia belum tentu relevan dengan penelitian yang akan dilakukan. Oleh karena itu, metode pembelajaran mesin mengalami kesulitan dalam menangani sejumlah besar fitur input, yang merupakan tantangan menarik bagi para peneliti (Kumar, 2014). Pada penelitian ini akan dibangun beberapa kombinasi seleksi fitur dan model klasifikasi.

Dalam penelitian ini dataset yang digunakan untuk membedakan Trojan dan bukan Trojan adalah *dataset Trojan Detection* yang diterbitkan oleh *Cyber Cop* dan dipublikasikan pada Kaggle. *Dataset* ini diperoleh dari situs *web* CIC yang ini berisi catatan atau aktifitas trafik yang terjadi untuk membedakan antara Trojan dan Bukan Trojan. *Dataset* yang digunakan terdiri dari 90683 data yang memiliki label Trojan dan 86799 data yang memiliki label bukan trojan dengan total data keseluruhan berjumlah 177482 data. *Dataset* yang digunakan memiliki total 84 atribut dan 1 label. *Dataset* ini diterbitkan pada tanggal 17 September 2021 sebagai sebagai versi pertama.

Kerangka pikir dapat dilihat pada gambar 1 yang menjelaskan mengenai alur penelitian yang akan dilakukan. Dimana dilakukan reduksi fitur menggunakan beberapa metode yang diusulkan. Selanjutnya seluruh fitur yang terseleksi akan dilakukan pengujian terhadap beberapa model klasifikasi untuk menentukan model klasifikasi terbaik terhadap fitur yang telah diseleksi sebelumnya.



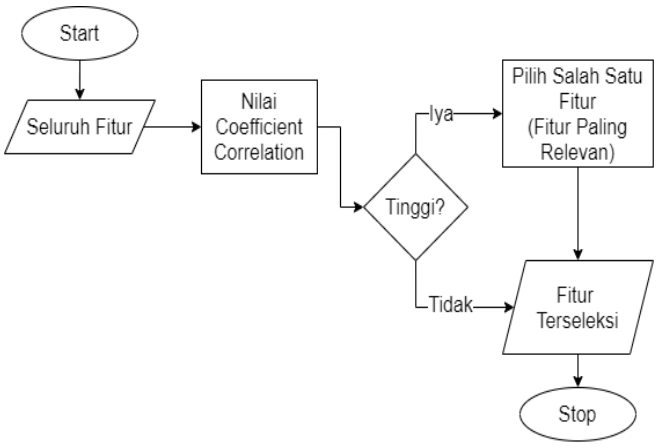
Gambar 1 Alur Kerja Penelitian

Berdasarkan alur kerja pada Gambar 1 seluruh kombinasi akan dilakukan pengujian untuk menentukan waktu dan akurasi prediksi dari setiap kombinasi seleksi fitur dan model klasifikasi yang yang dibuat. Adapun metode seleksi fitur yang diperbandingkan adalah :

1. Seluruh fitur
2. Menghilangkan fitur berkorelasi (Korelasi 50% dan Korelasi 70%)
3. Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan
4. Menggunakan metode *Linear Discriminant Analysis* (LDA) untuk memperoleh fitur terbaik
5. Menggunakan metode *Principal Component Analysis* (PCA) untuk memperoleh fitur terbaik
6. Menghilangkan fitur berkorelasi + Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan
7. Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan + Menggunakan metode *Linear Discriminant Analysis* (LDA) untuk memperoleh fitur terbaik
8. Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan + Menggunakan metode *Principal Component Analysis* (PCA) untuk memperoleh fitur terbaik
9. Menghilangkan fitur berkorelasi (Korelasi 50% dan 70%) + Menggunakan metode *Linear Discriminant Analysis* (LDA) untuk memperoleh fitur terbaik
10. Menghilangkan fitur berkorelasi (Korelasi 50% dan 70%) + Menggunakan metode *Principal Component Analysis* (PCA) untuk memperoleh fitur terbaik
11. Menghilangkan fitur berkorelasi + Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan + Menggunakan metode *Linear Discriminant Analysis* (LDA) untuk memperoleh fitur terbaik
12. Menghilangkan fitur berkorelasi + Menggunakan metode *Information Gain* dalam melakukan seleksi fitur yang relevan + Menggunakan metode *Principal Component Analysis* (PCA) untuk memperoleh fitur terbaik.

Perancangan *Coefficient Correlation*

Gambar 2 menunjukkan bagaimana proses seleksi fitur menggunakan *Coefficient Correlation*. Apabila data memiliki nilai *Coefficient Correlation* tinggi maka hanya salah satu data tersebut yang menjadi fitur terseleksi dan apabila nilai *Coefficient Correlation* rendah maka akan menjadi fitur terseleksi.



Gambar 2 *Flowchart Coefficient Correlation*

Adapun rumus dari *Coefficient Correlation* mengacu pada persamaan (1) :

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{\sum(x-\bar{x})^2 \sum(y-\bar{y})^2}} \quad (1)$$

Keterangan :

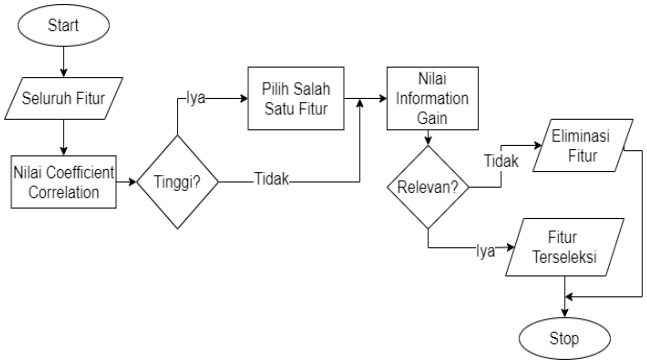
x = variabel independent

y = variable dependen

$\bar{x}$ = Rata – Rata nilai x

$\bar{y}$ = Rata – Rata nilai y

Perancangan *Information Gain*



Gambar 3 *Flowchart Information Gain*

Gambar 3 menunjukkan seluruh fitur yang ada akan dihitung tingkat relevansinya menggunakan *Information Gain*. Apabila fitur tersebut tidak relevan maka fitur tersebut akan dieliminasi dan apabila fitur tersebut relevan maka akan lanjut ke tahap selanjutnya, Adapun rumus yang digunakan mengacu pada persamaan (2) : (Pramono et al., 2019)

$$info_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|}\ x\ info(D_j) \qquad (2)$$

Keterangan :

A = Atribut,

$|D_j|$ = banyak sampel untuk nilai j,

$|D|$ = banyak sampel data,

v = kemungkinan nilai untuk atribut A.

Sedangkan rumus yang digunakan dalam menghitung Entropy, S(c) mengacu pada persamaan (3) : (Pramono et al., 2019)

$$info(D) = -\sum_{i=1}^{c} p_i\ log_2\ p(p_i) \qquad (3)$$

Keterangan :

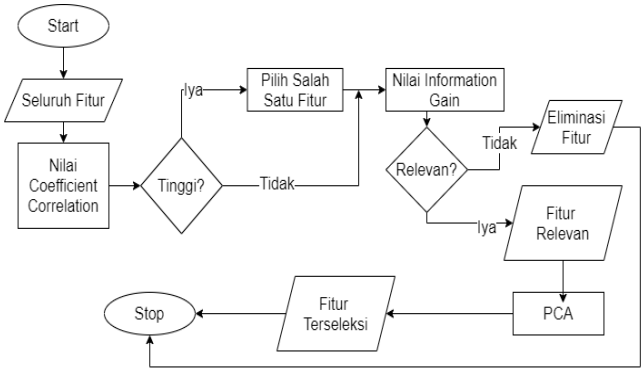
c = jumlah nilai dalam atribut target (jumlah kelas),

p_i = banyak sampe terhadap kelas i.

Untuk mendapatkan nilai *Information Gain* akan menggunakan perhitungan yang mengacu pada persamaan (4) : (Pramono et al., 2019)

$$Gain(A) = |info(D) - info_A(D)| \qquad (4)$$

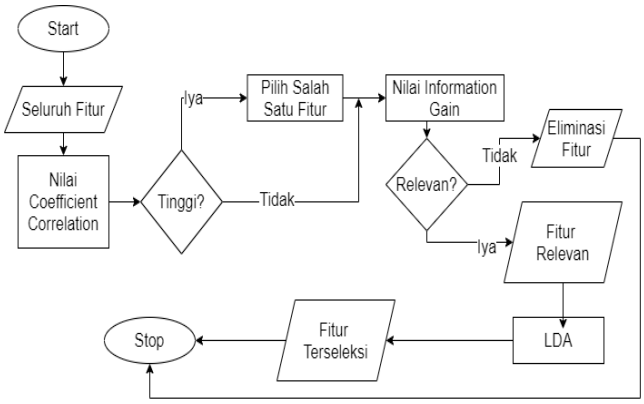
Perancangan *Principal Component Analysis*



Gambar 4 *Flowchart Principal Component Analysis* (PCA)

Gambar 4 menunjukkan seleksi fitur menggunakan metode *Principal Component Analysis* (PCA), dimana dilakukan pengurangan jumlah fitur menggunakan Metode PCA untuk mereduksi dimensi seluruh fitur tanpa kehilangan banyak informasi yang terkandung didalamnya. Analisis komponen utama (PCA) merupakan metode yang dapat mereduksi dimensi data dengan cara melakukan analisis kovarians antara variabel. Keuntungan utama dari analisis komponen utama yaitu mengurangi jumlah dimensi tanpa harus banyak kerugian informasi (Saed-moucheshi et al., 2013).

Perancangan *Linear Discriminant Analysis*



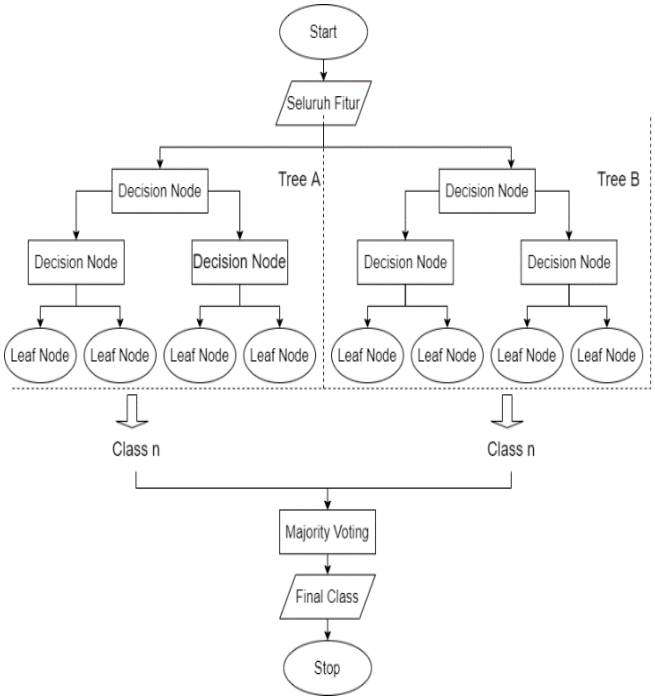
Gambar 5 *Flowchart Linear Discriminant Analysis* (LDA)

Gambar 5 menunjukkan seleksi fitur menggunakan metode *Linear Discriminant Analysis* (LDA) untuk mengurangi dimensi. LDA mengurangi dimensi dan juga menyimpan semua kemungkinan informasi diskriminatif. LDA menggunakan teknik ekstraksi ciri untuk mereduksi dimensi(Ghosh & Shuvo, 2019). LDA melakukan reduksi dengan cara mencari

kombinasi atribut terbaik yang dapat memisahkan kelas-kelas dalam kumpulan data dan meminimalkan varians di setiap kelas.

Perancangan Model Klasifikasi

Perancangan Model *Random Forest*



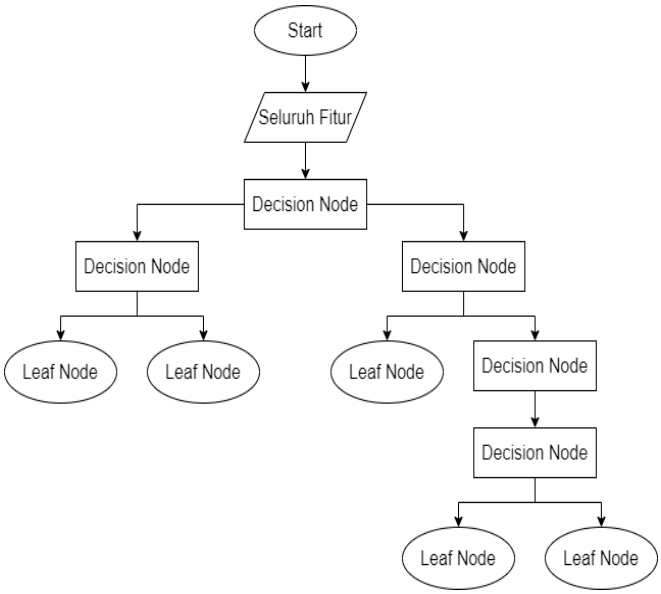
Gambar 6 Alur Sistem *Random Forest*

Gambar 6 menunjukkan alur sistem dari *Random Forest. Random Forest* merupakan pengklasifikasi yang mengkombinasi beberapa prediktor pohon keputusan sehingga setiap pohon bergantung pada nilai vektor acak dari sampel independen dan memiliki distribusi yang sama untuk semua pohon yang ada (Tian et al., 2009). *Random Forest* bekerja dengan membangun beberapa pohon keputusan dan menggabungkannya untuk prediksi yang lebih stabil dan akurat. "Hutan" yang dibangun oleh *Random Forest* adalah kumpulan pohon keputusan, biasanya dilatih dengan metode *bagging*. Metode *bagging* adalah metode yang menggabungkan model pembelajaran untuk meningkatkan efek keseluruhan Algoritma *Random Forest*. Berdasarkan gabungan keputusan yang diperoleh dari kumpulan pohon keputusan selanjutnya akan menjadi model akhir yang digunakan dalam klasifikasi.

Perancangan Model *Decision Tree*

Algoritma *Decision Tree* adalah algoritma berbasis aturan sederhana yang didasarkan pada seperangkat aturan yang memanfaatkan struktur sekuensial cabang pohon keputusan sehingga urutan aturan pemeriksaan dan tindakan terkait segera terlihat (Tian et al., 2009).

Pohon keputusan merupakan pohon yang mengklasifikasikan instance dengan mengurutkannya berdasarkan nilai fitur (Kotsiantis, 2007). Pada *Decision Tree* seluruh fitur yang digunakan akan menjadi pertanyaan(*Decision Node*), selanjutnya seluruh pertanyaan akan mengarah kedua buah simpul keputusan(jawaban), jika jawaban telah ditemukan maka pencarian selesai, jika jawaban belum ditemukan maka akan menuju ke pertanyaan(*Decision Node*) selanjutnya hingga menemukan jawaban. Gambar 7 menunjukkan alur sistem dari *Decision Tree*.



Gambar 7 *Flowchart Decision Tree*

Perancangan Model KNN

Metode *k-Nearest Neighbor* (kNN) adalah metode pengklasifikasi populer dalam penambangan data dan statistik karena implementasinya yang sederhana dan dan kinerja pengklasifikasi yang baik. Algoritma K-*Nearest Neighbors* bekerja dengan cara mengambil K data terdekat (tetangga) sebagai acuan untuk menentukan kelas dari data baru. Algoritma akan mengklasifikasikan data berdasarkan kesamaan atau kemiripan dengan data lain (Zhang et al., 2017).

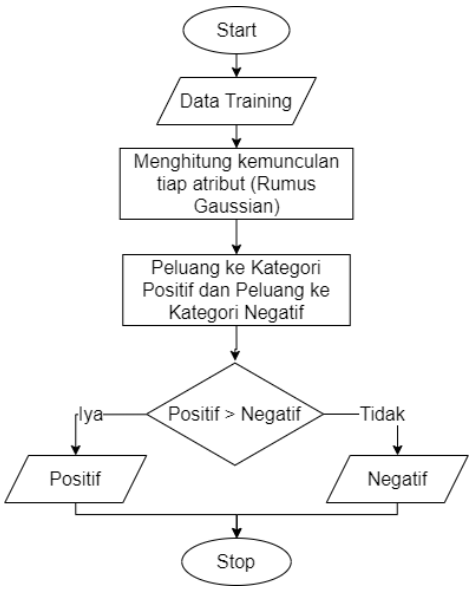
Tahap pertama dalam KNN adalah menentukan jumlah tetangga (K) yang akan dipertimbangkan untuk penentuan kelas. Selanjutnya menghitung jarak data baru ke setiap titik data dalam kumpulan data menggunakan *Euclidean Distance*. Selanjutnya mengambil beberapa K data yang paling dekat satu sama lain, kemudian tentukan kelas dari data baru tersebut berdasarkan rata – rata hasil dari K data yang paling dekat. Gambar 8 menunjukkan alur sistem dari KNN.

Gambar 8 *Flowchart* KNN

Perancangan Model *Naïve Bayes*

Pengklasifikasi *Naïve Bayes* merupakan pengklasifikasi probabilistik yang sederhana berlandaskan penerapan teorema Bayes (statistik Bayesian) berdasarkan asumsi independensi kuat. Keuntungan dari *Naïve Bayes classifier* yaitu hanya membutuhkan sedikit data pelatihan untuk memperkirakan parameter yang diperlukan untuk klasifikasi. (Kaur & Oberai, 2014).

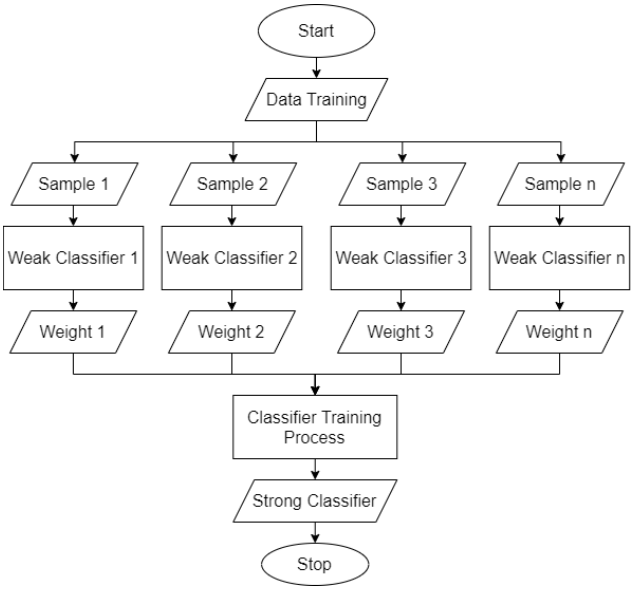
Algoritma *Naïve bayes* melakukan klasifikasi dengan cara menghitung probabilitas suatu peristiwa yang diperoleh dari hasil perhitungan kemunculan tiap atribut menggunakan rumus gaussian. Selanjutnya dilakukan perhitungan peluang ke kategori positif dan negative, apabila peluang positif lebih besar dari peluang *negative* maka masuk ke klasifikasi positif apabila tidak maka masuk ke klasifikasi *negative*. Gambar 9 menunjukkan alur sistem dari *Naïve Bayes*.



Gambar 9 *Flowchart Naïve Bayes*

Perancangan Model *AdaBoost*

AdaBoost adalah algoritma pembelajaran mesin yang berjalan secara iteratif. Dalam setiap iterasi, bobot sampel yang salah diklasifikasikan meningkat sedangkan bobot sampel yang diklasifikasikan dengan benar dikurangi. Pembaruan bobot semacam ini paling penting karena *AdaBoost* selanjutnya dapat fokus pada klasifikasi sampel yang sulit untuk mengurangi kesalahan pelatihan(Wu & Nagahashi, 2014). Oleh sebab itu pengkasifikasian lemah dapat meningkat, dengan melakukan penggabungan sejumlah fungsi klasifikasi lemah. Oleh sebab itu, pengklasifikasi yang kuat akan mengambil bentuk perceptron, yang merupakan kombinasi berbobot dari pengklasifikasi lemah. Gambar 8 menunjukkan alur sistem dari *AdaBoost.*



Gambar 10 *Flowchart AdaBoost*

Pengujian Akurasi

Metode yang akan digunakan dalam melakukan pengujian kinerja atau keakuratan model yang dibuat adalah metode *Cross-validation* atau validasi silang. validasi silang adalah sebuah metode yang dapat digunakan untuk menguji dan mengevaluasi algoritma atau model pembelajaran. Validasi silang secara acak akan membagi data menjadi data pembelajaran untuk pelatihan dan data pengujian untuk evaluasi model.

Confusion Matrix digunakan untuk memperlihatkan perbandingan antara hasil prediksi yang diperoleh dari model klasifikasi dengan data sebenarnya atau data label. Adapun perhitungan akurasi mengacu pada persamaan (5):

$$Akurasi = \frac{TP+TN}{TP+TN+FP+TN} X\ 100\% \quad (5)$$

Keterangan :

True Positive (TP) = Data positif diprediksi benar.

True Negative (TN) = Data negatif diprediksi benar.

False Positive (FP) = Data negatif diprediksi salah.

False Negative (FN) = Data positif diprediksi salah.

Pengujian Waktu

Dalam pengambilan sampel waktu digunakan rumus yang mengacu pada persamaan (6):

$$Waktu = Waktu\ Akhir - Waktu\ Awal \quad (6)$$

Dimana waktu awal adalah waktu saat model melakukan prediksi data uji terhadap model yang telah dibuat menggunakan data latih. Waktu akhir adalah waktu disaat model menampilkan hasil prediksi.

HASIL DAN PEMBAHASAN

Hasil Seleksi Fitur

1. Korelasi 50

Tabel 1 menunjukkan fitur yang memiliki tingkat korelasi sebesar 50% yang adakan digunakan pada pembuatan model:

Tabel 1 Fitur Tanpa Korelasi 50%

Nama Fitur	Nama Fitur
Flow ID	*PSH Flag Count*
Destination Port	*Down/Up Ratio*
Total Fwd Packets	*Init_Win_bytes_backward*
Total Length of Fwd Packets	*min_seg_size_forward*
Fwd Packet Length Min	*Active Std*
Bwd Packet Length Min	*Idle Min*
Flow Bytes/s	*Destination IP*
Flow Packets/s	*Timestamp*
Bwd IAT Mean	*Flow IAT Mean*
Fwd PSH Flags	*URG Flag Count*
Bwd Packets/s	*Active Mean*
FIN Flag Count	*Idle Std*

2. Korelasi 70

Tabel 2 menunjukkan nilai fitur yang memiliki korelasi lebih besar dari 70% dapat dilihat pada tabel sebelumnya. Berikut seluruh fitur yang digunakan tanpa adanya korelasi sebesar 70% terhadap masing-masing fitur menggunakan *Coefficient Correlation*:

Tabel 2 Fitur Tanpa Korelasi 70%

NamaFitur	NamaFitur
ACKFlagCount	FlowPackets/s
ActiveMean	FwdIATStd
ActiveStd	FwdPacketLengthMax
BwdIATMean	FwdPacketLengthMin
BwdIATStd	FwdPSHFlags
BwdPacketLengthMax	IdleMin
BwdPacketLengthMin	IdleStd
BwdPackets/s	Init_Win_bytes_backward
DestinationIP	min_seg_size_forward
DestinationPort	MinPacketLength
Down/UpRatio	Protocol
FINFlagCount	PSHFlagCount
FlowBytes/s	Timestamp
FlowDuration	TotalFwdPackets
FlowIATMean	TotalLengthofFwdPackets
FlowID	URGFlagCount

3. Information Gain

Tabel 3 menunjukkan nilai relevansi dari setiap fitur yang diperoleh menggunakan perhitungan nilai *Information Gain.*

Tabel 3 Nilai *Information Gain* Masing - masing Fitur yang Digunakan

Feature	**Skor**	***Feature***	**Skor**
Timestamp	0.691646	Bwd IAT Total	0.012072
Flow ID	0.480247	min_seg_size_forward	0.011851
Source IP	0.126759	Protocol	0.011167
Destination IP	0.122846	Bwd IAT Mean	0.010984
Source Port	0.106542	Bwd Header Length	0.010266

Tabel 4 menunjukkan seluruh fitur yang digunakan dimana fitur yang digunakan adalah fitur yang memiliki relevansi diatas 0.01:

Tabel 4 Fitur Dengan Nilai *Information Gain* > 0.01

Nama Fitur
Timestamp
Flow ID
Source IP
Destination IP
Source Port

4. LDA

Pada reduksi fitur menggunakan LDA pada penelitian ini *explained variance ratio* yang diperoleh dan digunakan adalah 1, sehingga fitur yang digunakan pada pembuatan model sebanyak 1 fitur, tabel 5 menunjukkan potongan nilai fitur LDA yang digunakan:

Tabel 5 Potongan Nilai Fitur LDA

Potongan Nilai Fitur LDA
1.92
23534
26277
…
15357
0.4703
-0.5442

5. PCA

Pada reduksi fitur menggunakan PCA pada penelitian ini banyak komponen PCA yang digunakan adalah 2, sehingga fitur yang digunakan pada pembuatan model sebanyak 2 fitur, tabel 6 menunjukkan potongan nilai fitur PCA yang digunakan:

Tabel 6 Potongan Nilai Fitur PCA

PCA1	**PCA2**
-60.159.118.945	-95.430.171.836
-220.958.471.767	-35.088.895.333
-214.415.523.942	-29.452.636.464
...	…
-218.200.354.254	-34.600.003.326
-223.913.931.599	-35.373.643.309
-218.198.743.922	-28.659.437.594

6. Korelasi, *Information Gain*

Tabel 7 menunjukkan seluruh fitur yang digunakan dimana fitur yang digunakan adalah fitur tanpa korelasi dan memiliki relevansi diatas 0.01:

Tabel 7 Fitur Tanpa Korelasi dan Relevansi > 0.01

Nama Fitur
Timestamp
Flow ID
Destination IP

7. Korelasi 50, LDA

Fitur yang digunakan adalah fitur yang tidak memiliki korelasi sebesar 50% dan reduksi fitur menggunakan LDA pada penelitian ini *explained variance ratio* yang diperoleh dan digunakan adalah 1, sehingga fitur yang digunakan pada pembuatan model sebanyak 1 fitur, tabel 8 menunjukkan potongan nilai fitur LDA yang digunakan:

Tabel 8 Potongan Nilai Fitur LDA Tanpa Korelasi 50%

PotonganNilai Fitur LDA Tanpa Korelasi 50%
-1.8051
-2.1442
-2.5739
...
-1.3319
-0.561
0.6202

8. Korelasi 70, LDA

Fitur yang digunakan adalah fitur yang tidak memiliki korelasi sebesar 70% dan reduksi fitur menggunakan LDA pada penelitian ini *explained variance ratio* yang diperoleh dan digunakan adalah 1, sehingga fitur yang digunakan pada pembuatan model sebanyak 1 fitur, tabel 9 menunjukkan potongan nilai fitur LDA yang digunakan:

Tabel 9 Potongan Nilai Fitur LDA Tanpa Korelasi 70%

Potongan Nilai Fitur LDA Tanpa Korelasi 70%
-1.7991
-2.2543
-2.5409
...
-1.3354
-0.4627
0.5104

9. Korelasi 50, PCA

Fitur yang digunakan adalah fitur yang tidak memiliki korelasi sebesar 50% dan reduksi fitur menggunakan PCA pada penelitian ini banyak komponen PCA yang digunakan adalah 2, sehingga fitur yang digunakan pada pembuatan model sebanyak 2 fitur, tabel 10 menunjukkan potongan nilai fitur PCA yang digunakan:

Tabel 10 Potongan Nilai Fitur PCA Tanpa Korelasi 50%

PCA1	PCA2
-40.916.006.581	-804730.8541
-4680582.165	-2598568.0393
-4563804.816	-1611928.5236
...	…
-4677456.906	-2591444.3474
-4683202.2162	-2590630.9547
-4643584.6404	-22.659.351.047

10. Korelasi 70, PCA

Fitur yang digunakan adalah fitur yang tidak memiliki korelasi sebesar 50% dan reduksi fitur menggunakan PCA pada penelitian ini banyak komponen PCA yang digunakan adalah 2, sehingga fitur yang digunakan pada pembuatan model sebanyak 2 fitur, tabel 11 menunjukkan potongan nilai fitur PCA yang digunakan:

Tabel 11 Potongan Nilai Fitur PCA Tanpa Korelasi 70%

PCA1	PCA2
-25.030.855.671	1830665.3128
-12697046.6032	-2051487.2133
-11914579.3999	-1337390.023
...	…
-12564524.7395	-2000136.4939
-12896686.8632	-2125681.0332
-12570870.077	-18.641.608.535

11. *Information Gain*, LDA

Fitur yang digunakan dimana fitur yang digunakan adalah fitur yang memiliki relevansi diatas 0.01 dan reduksi fitur menggunakan LDA pada penelitian ini *explained variance ratio* yang diperoleh dan digunakan adalah 1, sehingga fitur yang digunakan pada pembuatan model sebanyak 1 fitur, tabel 12 menunjukkan potongan nilai fitur LDA yang digunakan:

Tabel 12 Potongan Nilai Fitur LDA Relevansi > 0.01

Potongan Nilai Fitur LDA Relevansi > 0.01
-1.7972
-2.1104
-2.5842
...
-1.353
-0.6579
0.6683

12. *Information Gain*, PCA

Fitur yang digunakan dimana fitur yang digunakan adalah fitur yang memiliki relevansi diatas 0.01 dan reduksi fitur menggunakan PCA pada penelitian ini banyak komponen PCA yang digunakan adalah 2, sehingga fitur yang digunakan pada pembuatan model sebanyak 2 fitur, tabel 13 menunjukkan potongan nilai fitur PCA yang digunakan:

Tabel 13 Potongan Nilai Fitur PCA Relevansi > 0.01

PCA1	PCA2
52.880.654	-12317.6492
-23058.807	-7536.6933
43625.859	-5425.3008
...	…
-25495.5988	-9040.0133
26327.8993	26023.1836
10296.5015	-102.412.401

13. Korelasi, *Information Gain*, LDA

Fitur yang digunakan dimana fitur yang digunakan adalah fitur tanpa korelasi, memiliki relevansi diatas 0.01, dan reduksi fitur menggunakan LDA pada penelitian ini *explained variance ratio* yang diperoleh dan digunakan adalah 1, sehingga fitur yang digunakan pada pembuatan model sebanyak 1 fitur, tabel 14 menunjukkan potongan nilai fitur LDA yang digunakan:

Tabel 14 Potongan Nilai Fitur LDA Tanpa Korelasi dan Relevansi > 0.01

Nilai Fitur LDA Tanpa Korelasi dan Relevansi > 0.01
-1.8229
-2.142
-2.598
...
-1.384
-0.6818

Nilai Fitur LDA Tanpa Korelasi dan Relevansi > 0.01
0.6435

14. Korelasi, *Information Gain*, PCA

Fitur yang digunakan dimana fitur yang digunakan adalah fitur tanpa korelasi, memiliki relevansi diatas 0.01, dan reduksi fitur menggunakan PCA pada penelitian ini banyak komponen PCA yang digunakan adalah 2, sehingga fitur yang digunakan pada pembuatan model sebanyak 2 fitur, tabel 15 menunjukkan potongan nilai fitur PCA yang digunakan :

Tabel 15 Potongan Nilai Fitur PCA Tanpa Korelasi dan Relevansi > 0.01

PCA1	PCA2
69.461.574	15378.5615
-21793.5886	18874.0375
43940.9416	20506.9691
...	…
-23991.7589	12584.0188
22429.8209	5424.1205
11630.9385	-52.793.328

Hasil Pengujian

Pada penelitian ini pengujian dilakukan dengan membandingkan kombinasi antara seleksi fitur *Coefficient Correlation, Information Gain,* PCA, dan LDA terhadap *Random Forest, Decision Tree, KNN, Naive Bayes, AdaBoost*. Seluruh kombinasi yang ada dilakukan perhitungan terhadap waktu yang dibutuhkan dalam melakukan klasifikasi dan hasil akurasi prediksinya untuk menemukan kombinasi yang paling tepat terhadap aktifitas jaringan Trojan. Hasil penelitian berdasarkan kombinasi seleksi fitur dengan algoritma klasifikasi terhadap seluruh percobaan yang dilakukan menggunakan satuan detik(*second/s*) untuk waktu dan persentase(%) untuk akurasi. Pengujian dilakukan dengan melakukan percobaan sebanyak 5 kali untuk setiap kombinasi. Tabel 16 menunjukkan waktu terbaik yang diperoleh algoritma klasifikasi berdasarkan kombinasi metode seleksi fitur (*Coefficient Correlation, Information Gain*, PCA, dan LDA) dengan tetap mempertahankan akurasi terbaiknya:

Tabel 16 Rata – rata Waktu dan Akurasi Prediksi Terbaik

	Waktu (S)	Akurasi (%)	Metode Reduksi
AdaBoost	0,1341	0,99	Korelasi, *Information Gain*, dan PCA
Decision Tree	0,0034	0,99	Korelasi, *Information Gain*, dan PCA
KNN	0,7945	0,99	Korelasi, *Information Gain*, dan PCA
Naïve Bayes	0,0022	0,92	Korelasi 70 dan LDA
Random Forest	0,1966	0,99	*Information Gain*

Dengan menggunakan metode reduksi fitur dapat menigkatkan waktu prediksi dengan signifikan dibandingkan tanpa menggunakan redukasi fitur. Tabel 17 menunjukkan waktu prediksi tanpa menggunakan seleksi fitur.

Tabel 17 Rata - rata Waktu dan Akurasi prediksi Tanpa Seleksi

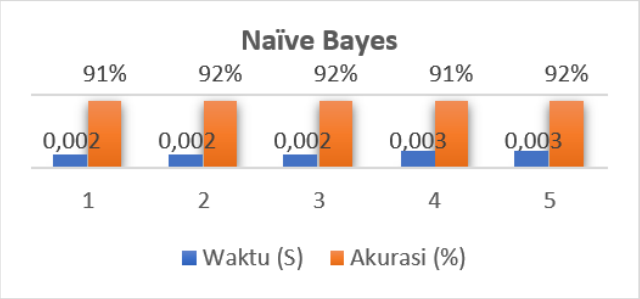
	Waktu (S)	Akurasi (%)
AdaBoost	0,4370	99%
Decision Tree	0,0178	99%
KNN	59,4920	83%
Naïve Bayes	0,0513	88%
Random Forest	0,5297	99%

Pembahasan

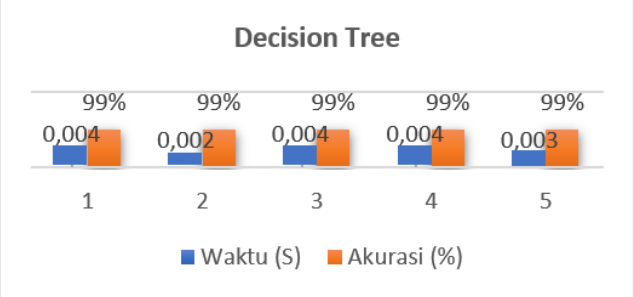
Berdasarkan hasil penelitian yang telah dilakukan, diperoleh hasil yang menunjukkan algoritma *Naïve Bayes* selalu menunjukkan waktu prediksi dibawah 0,1 detik dengan rata – rata waktu prediksi 0,0096 detik, dimana waktu prediksi tertinggi diperoleh dari tanpa reduksi fitur dengan waktu 0,0513 detik dan waktu prediksi terendah diperoleh dari kombinasi seleksi fitur Korelasi 70% dan LDA dengan waktu 0,0022 detik. Namun akurasi rata – rata yang diperoleh cukup rendah atau yang terendah dibandingkan Algoritma lainnya yaitu 80% dimana akurasi terendah diperoleh dari kombinasi antara *Coefficient Correlation* dan PCA dengan akurasi 50%.

Algoritma *Decision Tree* menunjukkan salah satu hasil akurasi terbaik dengan akurasi rata – rata 89% terhadap seluruh kombinasi. Dimana akurasi terendah yang diperoleh 58% menggunakan kombinasi *Coefficient Correlation* dan PCA, dan memiliki akurasi tidak kurang dari 90% untuk kombinasi tanpa

menggunakan PCA dengan akurasi tertinggi 99%. Waktu prediksi yang diperoleh cukup baik dimana waktu prediksi tertinggi yang diperoleh 0,0178 detik tanpa menggunakan reduksi fitur.



Gambar 11 Waktu dan Akurasi Prediksi Dengan Waktu Terbaik



Gambar 12 Waktu dan Akurasi Prediksi Dengan Akurasi Terbaik

Berdasarkan hasil yang diperoleh (Gambar 11 dan 12) untuk klasifikasi yang membutuhkan waktu prediksi yang cepat diperoleh kombinasi seleksi fitur Korelasi 70% dan LDA dengan menggunakan metode pengklasifikasi *Naïve Bayes*, dimana dengan menggunakan kombinasi antara pemilihan fitur dan metode tersebut diperoleh waktu prediksi dengan rata – rata 0,0022 detik dengan akurasi mencapai 92%. Sedangkan untuk klasifikasi yang membutuhkan akurasi maksimal diperoleh kombinasi antara *Coefficient Correlation*, *Information Gain*, dan PCA dengan algoritma *Decision Tree*, dimana dengan menggunakan kombinasi seleksi fitur dan algoritma tersebut diperoleh akurasi 99% dan waktu prediksi 0,0034 detik.

Terdapat beberapa faktor yang menyebabkan terjadinya perbedaan waktu komputasi. Dengan menggunakan *Coefficient Correlation* redundansi data dapat dihilangkan sehingga menghilangkan parameter yang memiliki nilai yang relatif sama. Untuk melihat fitur yang relevan terhadap prediksi (label) yang digunakan dalam pembuatan model dilakukan perhitungan relevansi fitur menggunakan *Information Gain* sehingga fitur yang tidak memiliki relevansi (hubungan) terhadap klasifikasi yang dilakukan akan dihilangkan. Dengan menggunakan PCA dapat dilakukan peringkasan atau pereduksian dimensi informasi yang terdapat pada parameter data besar hingga menjadi beberapa kumpulan parameter ringkasan yang lebih kecil. Dengan jumlah parameter yang lebih sedikit dengan kualitas data yang baik penggunaan algoritma medel klasifikasi *Decision Tree* yang merupakan model klasifikasi pemilihan keputusan bercabang akan menjadi lebih baik dan efisien

KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh hasil yang menunjukkan Relevansi fitur sangat berpengaruh terhadap hasil akurasi pengklasifikasian dan dengan menghilangkan redundansi fitur dapat meningkatkan waktu prediksi dan menjaga akurasi prediksi pengklasifikasian tetap maksimal, serta dengan menggunakan reduksi fitur LDA dan PCA dapat lebih memangkas waktu prediksi lebih baik lagi.

Pengaruh kombinasi antara seleksi fitur dan algoritma klasifikasi sangat signifikan, dimana waktu prediksi dan akurasi pengklasifikasian sangat dipengaruhi oleh kombinasi keduanya. Pada penelitian yang dilakukan diperoleh hasil dengan kombinasi seleksi fitur Korelasi 70% dan LDA dengan menggunakan Algoritma klasifikasi *Naïve Bayes* diperoleh waktu prediksi dengan rata – rata 0,0022 detik dengan akurasi mencapai 92% untuk waktu prediksi terbaik. Sedangkan untuk klasifikasi yang membutuhkan akurasi maksimal dengan waktu terbaik diperoleh kombinasi antara *Coefficient Correlation*, *Information Gain*, dan PCA dengan algoritma *Decision Tree*, dimana diperoleh akurasi 99% dan waktu prediksi 0,0034 detik.

Pada penelitian kedepannya diharapkan dapat mengembangkan atau meningkatkan waktu prediksi Aktifitas Trojan Pada Jaringan Internet pada sisi yang berbeda, seperti meningkatkan waktu pengambilan data yang dibutuhkan. Penelitian berikutnya juga diharapkan melakukan perbandingan kombinasi lainnya pada sistem yang serupa.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih atas waktu dan dukungan yang diberikan oleh pembimbing dan seluruh pihak yang telah berkontribusi dalam penelitian ini.

DAFTAR PUSTAKA

Al-Saadoon, G. M. W., & Al-Bayatti, H. M. Y. (2011). *A Comparison of Trojan Virus Behavior in Linux and Windows Operating Systems*. *1*(3), 56–62. http://arxiv.org/abs/1105.1234

Choi, T. M., Wallace, S. W., & Wang, Y. (2018). Big Data Analytics in Operations Management. *Production and Operations Management*, *27*(10), 1868–1883. https://doi.org/10.1111/poms.12838

Ghosh, J., & Shuvo, S. B. (2019). Improving Classification Model's Performance Using Linear Discriminant Analysis on Linear Data. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, 8–12. https://doi.org/10.1109/ICCCNT45670.2019.8944632

Han, X., & Tan, Q. (2010). Dynamical behavior of computer virus on Internet. *Applied Mathematics and Computation*, *217*(6), 2520–2526. https://doi.org/10.1016/j.amc.2010.07.064

Kaur, G., & Oberai, N. (2014). A Review Article on Naïve Bayes Classifier with Various Smoothing Techniques. *International Journal of Computer Science and Mobile Computing*, *3*(10), 864–868. www.ijcsmc.com

Kherif, F., & Latypova, A. (2019). Principal component analysis. *Machine Learning: Methods and Applications to Brain Disorders*, *1*(C), 209–225. https://doi.org/10.1016/B978-0-12-815739-8.00012-2

Kok, C. H., Ooi, C. Y., Inoue, M., Moghbel, M., Baskara Dass, S., Choo, H. S., Ismail, N., & Hussin, F. A. (2019). Net Classification Based on Testability and Netlist Structural Features for Hardware Trojan Detection. *Proceedings of the Asian Test Symposium*, *2019-Decem*, 105–110. https://doi.org/10.1109/ATS47505.2019.00020

Kotsiantis, S. B. (2007). Supervised Machine Learning: A Review of Classification Techniques. *Hyperfine Interactions*, *1*, 12.

Kumar, V. (2014). Feature Selection: A literature Review. *The Smart Computing Review*, *4*(3). https://doi.org/10.6029/smartcr.2014.03.007

Kurihara, T., & Togawa, N. (2021). Hardware-trojan classification based on the structure of trigger circuits utilizing random forests. *Proceedings - 2021 IEEE 27th International Symposium on On-Line Testing and Robust System Design, IOLTS 2021*, 24–27. https://doi.org/10.1109/IOLTS52814.2021.9486700

Lu, J., Chen, Y., Herodotou, H., & Babu, S. (2018). Speedup your analytics: Automatic parameter tuning for databases and big data systems. *Proceedings of the VLDB Endowment*, *12*(12), 1970–1973. https://doi.org/10.14778/3352063.3352112

Plotnikova, V., Dumas, M., & Milani, F. (2020). Adaptations of data mining methodologies: A systematic literature review. *PeerJ Computer Science*, *6*, 1–43. https://doi.org/10.7717/PEERJ-CS.267

Pramono, F., Didi Rosiyadi, & Windu Gata. (2019). Integrasi N-gram, Information Gain, Particle Swarm Optimation di Naïve Bayes untuk Optimasi Sentimen Google Classroom. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, *3*(3), 383–388. https://doi.org/10.29207/resti.v3i3.1119

Saed-moucheshi, A., Fasihfar, E., Hasheminasab, H., Rahmani, A., & Ahmadi, A. (2013). A Review on Applied Multivariate Statistical Techniques in Agriculture and Plant Science. *International Journal of Agronomy and Plant Production*, *4*(1), 127–141.

Sinaga, K. P., Hussain, I., & Yang, M. S. (2021). Entropy K-Means Clustering with Feature Reduction under Unknown Number of Clusters. *IEEE Access*, *9*, 67736–67751. https://doi.org/10.1109/ACCESS.2021.3077622

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications*, *30*(2), 169–190. https://doi.org/10.3233/AIC-170729

Thimbleby, H., Anderson, S., & Cairns, P. (1998). A Framework for Modelling Trojans and Computer Virus Infection. *Computer Journal*, *41*(7), 443–458.

Tian, R., Batten, L., Islam, R., & Versteeg, S. (2009). An automated classification system based on the strings of trojan and virus families. *2009 4th International Conference on Malicious and Unwanted Software, MALWARE 2009*, 23–30. https://doi.org/10.1109/MALWARE.2009.5403021

Wu, S., & Nagahashi, H. (2014). Parameterized adaboost: Introducing a parameter to speed up the training of real adaboost. *IEEE Signal Processing Letters*, *21*(6), 687–691. https://doi.org/10.1109/LSP.2014.2313570

Yang, M. S., & Sinaga, K. P. (2019). A feature-reduction multi-view k-means clustering algorithm. *IEEE Access*, *7*, 114472–114486. https://doi.org/10.1109/ACCESS.2019.2934179

Yeh, W. C., Lin, E., & Huang, C. L. (2021). Predicting Spread Probability of Learning-Effect Computer Virus. *Complexity*, *2021*. https://doi.org/10.1155/2021/6672630

Zhan, Z. H., Shi, L., Tan, K. C., & Zhang, J. (2022). A survey on evolutionary computation for complex continuous optimization. In *Artificial Intelligence Review* (Vol. 55, Nomor 1). Springer Netherlands. https://doi.org/10.1007/s10462-021-10042-y

Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R., Zhang, S., Zong, M., & Zhu, X. (2017). Efficient kNN Classification With Different Numbers of Nearest Neighbors. *Ieee Transactions on Neural Networks and Learning Systems*, 1–12. http://ieeexplore.ieee.org.

Halaman ini sengaja dikosongkan