PERANCANGAN SISTEM ANALISIS DINAMIS DAN KLASIFIKASI MALWARE OTOMATIS DENGAN ALGORITMA K-NEAREST NEIGHBORS

DYNAMIC ANALYSIS SYSTEM DESIGN AND AUTOMATIC MALWARE CLASSIFICATION USING THE K-NEAREST NEIGHBORS ALGORITHM

Wasilaturrahmi[1], Yudha Purwanto[2], Muhammad Faris Ruriawan[3]
[1,2,3] Universitas Telkom, Bandung
wasilaturrahmi@student.telkomuniversity.ac.id[1], **omyudha@telkomuniversity.ac.id**[2], **muhammadfaris@telkomuniversity.ac.id**[3]

Abstrak

***Malware* merupakan sebuah perangkat lunak untuk merusak dan mencuri informasi berbagai sumber jaringan atau server yang tidak diketahui oleh pemiliknya, virus *malware* yang berbahaya rentan terhadap keamanan pada sistem agar *malware* mendapatkan akses untuk mengganggu layanan pada sistem serta mengambil informasi yang ditargetkan. Proses analisis yang dilakukan untuk mengidentifikasi apakah suatu file itu *malware* atau bukan.**

Semakin meningkat perkembangan *malware* bisa membuat *malware* tidak bisa terdeteksi karena dapat menghindari teknik analisis *malware* yang dilakukannya untuk mendeteksi *malware* dengan analisis *malware*, perkembangan *malware* dapat menghindari teknik analisis agar *malware* tidak dapat terdeteksi. Oleh karena itu, penelitian ini membuat atau merancang sebuah sistem otomatis mendeteksi *malware* menggunakan proses analisis dinamis dengan cuckoo *Sandbox* untuk dapat menghasilkan data yang diklasifikasikan. Proses analisis yang diklasifikasikan menggunakan sebuah algoritma *K-Nearest Neighbors* yang digunakan untuk mengklasifikasi file *malware*, pada metode ini *K-Nearest Neighbors* lebih memilih jalur terdekat dari tetangganya agar mendapatkan hasil akurasi yang optimal.

Kata kunci: ***Malware, Analisis Dinamis,* Klasifikasi*, K-Nearest Neighbors,* Cuckoo *Sandbox***

Abstract

Malware is software to damage and steals information on various network resources or servers that are not known by the owner, virus malware is malicious is vulnerable to system security so that malware gains access to disrupt services on the system and retrieve targeted information. The analysis process is carried out to identify whether a file is a malware or not.

The increasing development of malware can make the malware undetectable because it can avoid the malware analysis techniques it does to detect malware with malware analysis, the development of malware can avoid analysis techniques so that malware cannot be detected. Therefore, this research creates or designs an automatic malware detection system using a dynamic analysis process with cuckoo sandbox to be able to generate classified data. The analysis process is classified using a K-Nearest Neighbors algorithm which is used to classify malware files, in this method K-Nearest Neighbors prefer the closest path from its neighbors in order to get optimal accuracy results.

Keywords: ***Malware, Analisis Dinamis,* Klasifikasi*, K-Nearest Neighbors,* Cuckoo *Sandbox***

1. Pendahuluan

Perkembangan teknologi pada saat sekarang begitu sangat pesat begitu sebaliknya semakin cangih teknologi maka semakin tinggi tingkat kejahatan di dunia maya maupun digital. Penyalahgunaan kecangihan teknologi yang pesat salah satun pengambilan data informasi yang diambil tanpa sepengetahuan pemiliknya. Saat sekarang ancaman yang sangat besar dan sangat merugikan bagi seseorang ataupun kelompok yang disebut malicious software atau yang dikenal dengan sebutan *malware*.

Malware merupakan sebuah perangkat lunak yang bertujuan untuk merusak ataupun mencuri informasi sistem dan menganggu layanan untuk mengambil informasi yang ditargetkan dan memanipulasi data [1]. *Malware* diciptakan untuk merusak software atau sistem operasi. Perangkat lunak untuk mengidentifikasi perangkat yang tedeteksi sebagai analisis malware yang dibutuhkan untuk mendeteksi tujuan analisis pada file *malware*. Proses analisis yang dilakukan dengan file malware analisis dinamis berdasarkan *malware* perilakunya [2]. Analisis

dinamis dilakukan dengan memeriksa dan menjalankan *malware* dengan sampel kode yang berbahaya dan mengamati jenis dari perilaku *malware* [3].

Banyak tujuan yang dilakukan pelaku seperti aktifitas yang dilakukan berbahaya berdampak pada korban yang dirugikan, salah satu tujuannya untuk mengambil serta mencuri informasi penting dari korban yang berisikan informasi pribadi. *Malware* biasanya dapat dari sistem file yang diunduh yang tidak diketahui oleh pengguna, akibat *malware* yang telah aktif di sistem maka *malware* melakukan aktivitas yang bertujuan untuk merusak seluruh bagian dari sistem [4].

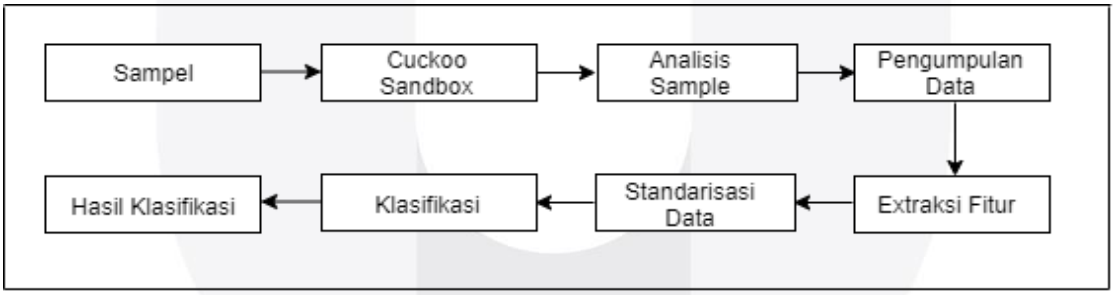
Berdasarkan studi literatur sebelumnya yang dilakukan dalam proses mendeteksi dan mengklasaifikasi analisis *malware* yang digunakan lebih efektif dan dinamis menggunakan analisis *malware*. Cuckoo Sandbox yang digunakan untuk menganalisis dan mengeksekusi file [1]. Pada penelitian ini penulis ingin merancang sebuah sistem otomatis untuk mengidentifikasi jenis file malware dengan algoritma K-Nearest Neighbors. Proses identifikasi menggunakan analisis *malware* berdasarakan jenis dari data karakteristik *malware*.

2. Perancangan Sistem

2.1 Gambaran Umum Sistem

Gambaran umum sistem yang digunakan untuk mengambarkan sistem yang dijalankan. Sistem ini dibuat untuk menganalisis file *malware* yang dimana sistem ini menentukan apakah file *malware* jenisnya tersebut *malware* atau *goodware*, sistem menggunakan algoritma yang digunakan untuk mengklasifikasi K-Nearest Neighbors. Tugas Akhir ini sistem menggunakan data dari VirusShare.com, The Zoo, portableapps dan filehippo.

Proses sistem yang dilakukan untuk mendeteksi *malware* menggunakan tahapan data dari karakteristik dinamis pada dataset perangkat lunak. Sistem analisis *malware* menggunakan analisis dinamis untuk menjalankan dan menganalisis *malware* pada cuckoo sandbox untuk mendapatkan hasil dari karakteristik dari hasil output *malware* yang berjalan berupa report.json. Pada penelitian ini penulis melakukan proses identifikasi *malware* dengan algoritma K-Nearest Neighbors untuk dapat memberikan hasil keluaran yang di klasifikasikan. Gambaran secara umun sistem sebagai berikut:



Gambar 1. Gambaran Umum Sistem

Gambar 1. Menjelaskan tentang bagaimana sistem berkerja mulai dari tahapan pengambilan data atau sampel sampai dengan tahapan hasil dari klasifikasi. Sampel dianalisis menggunakan wadah terisolasi dari cuckoo sandbox, dari hasil analisis mendapatkan hasil output berupa report.json. Setelah itu Proses yang dilakukan yaitu pengumpulan data proses ini merupakan pengambilan data atau sampel *malware* dari hasil karakteristik yang digunakan sebagai dataset yang telah dikumpulkan sebelumnya. Jenis perangkat lunak yang digunakan format executable. File *malware* yang di dapatkan dari beberapa sumber seperti Virusshare, The Zoo sedangkan perangkat lunak goodware dari Portableapps dan Filehippo.

Setelah penggumpulan data selanjutnya tahapan yang dilakukan yaitu dengan melakukan extraksi fitur proses ini merupakan proses dari ektraksi fitur yang digunakan, dimana proses analisis dinamis ini menggunakan proses pemanggilan atribut Application Programming Interfaces (API).

Klasifiksi *malware* menggunakan algoritma *K-Nearest Neighbors* dengan hasil dari ekstraksi fitur pada dataset yang ada sebelumnya. Pada algoritma *K-Nearest Neighbors* ini digunakan jarak yang lebih optimal dengan

parameter nilai K yang digunakan sebagai penentu jumlah tentangga yang dilakukan. Pada metode ini dilakukan dengan klasifikasi dari jarak yang lebih dekat dari jarak tetangga lainnya, klasifikasi yang digunakan *K-Nearest Neighbors* lebih sedikit karena *K-Nearest Neighbors* hanya memiilih jarak yang lebih dekat [11]. *K-Nearest Neighbors* biasanya dihitung berdasarkan rumus *euclidean* [12], sebagai berikut:

$$d(a,b) = \sqrt{\sum_{i=1}^{n}(ai - bi)^2} \quad (1)$$

Keterangan :

d : jarak dekat dari titik
a : titk awal
b : titik akhir
i : jumlah data yang digunakan
n : banyak data yang digunakan

Proses dari implementasi dengan algoritma K-Nearest Neighbors dengan menggunakan percobaaan data traning dan data testing. Untuk mnggukur kinerja menggunakan algoritma K-Nearest Neighbors menggunakan klasifikasi *confosion matrix*. Rumus *confosion matrix* sebagai berikut:

$$Akurasi = \frac{TP + TN}{TP + FP + TN + FN} \quad (2)$$

$$Presisi = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

Keterangan:

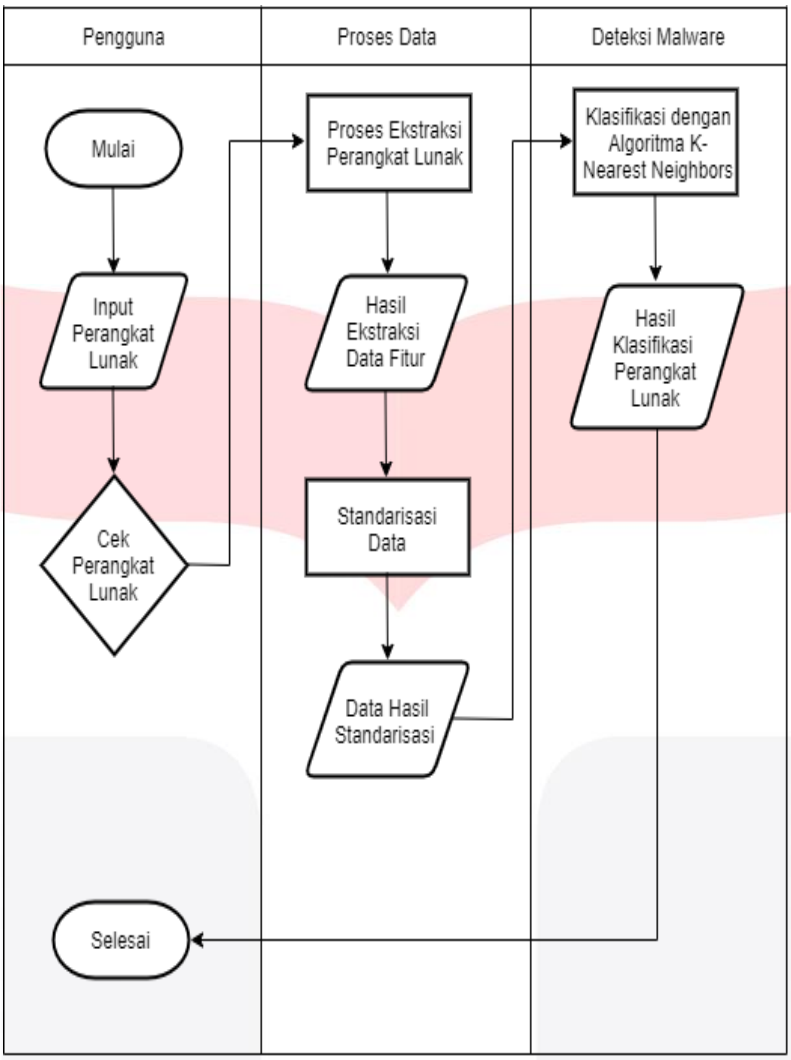
TP : *True Positive*

TN : *True Negative*

FP : *False Positive*

FN : *False Negative*

Pada Tugas akhir ini dilakukan sistem pelatihan dan pengujian terhadap proses analisis dinamis yang dijalankan. Pada data pelatihan pada sistem dataset yang telah dilakukan ekstraksi fitur dengan fitur pemanggilan dengan pelatihan model mendapatkan proses yang dilakukan oleh sistem dengan berbagai macam proses agar mendapatkan nilai yang optimal. Pada saat melakukan proses pengujian dengan menggunakan beberapa jumlah data proses ini sama dengan proses data pelatihan.



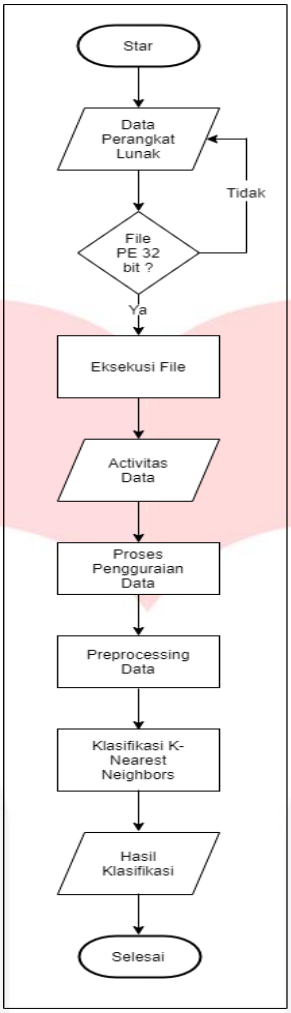
Gambar 2. Diagram Alir Sistem

Pada gambar 2 menjelaskan tentang diagram alir sistem secara keseluruhan. Diagram menjelaskan proses identifikasi *malware* yang dimulai dari pengguna menginput file *malware* kemudian melakukan proses data, klasifikasi data dan mendapatkan hasil klasifikasi *malware* berupa jenis hasil identifikasi dari perangkat *malware* atau *goodware*.

2.2 Perancangan sistem analisis dinamis

Proses sistem analisis dinamis, Proses analisis dinamis yang dilakukan pada perangkat lunak yang dijalankan dimesin virtual pada cuckoo *sandbox* untuk mendapatkan hasil output dari karakteristik dari *malware*. Proses dilakukan untuk menganalisis *malware* untuk mendapatkan hasil dari karakteristik dalam keluaran dalam bentuk jenis file *Java Script Object Notation* (JSON) jenis file ini yang didapatkan berbagai informasi aktivitas dari perangkat lunak *malware* yang dijalankan.

Hasil dari keluaran file JSON yang dijalankan di cuckoo *sandbox*, keluaran dari hasil dilakukan pemilihan beberapa fitur yang diperlukan dalam mengklasifikasikan jenis perangkat lunak. Proses ini dilakukan untuk urutan pemanggilan *API calls*. Atribut yang digunakan dalam proses klasifikasi yang diambil dari urutan pemangilan API *calls* yang dijalankan di dalam wadah terisolasi cuckoo *sandbox*. Atribut pemanggilan ini digunakan untuk proses klasifikasi dengan algoritma *K-Nearest Neighbors*. Gambaran proses perancangan sistem dinamis sebagai berikut:



Gambar 3 Perancangan sistem analisis dinamis

3. Hasil dan Pengujian sistem

3.3 Implementasi Algoritma K-Nearest Neighbors (KNN)

Berdasarkan rumus *confosion matrix* mendapatkan hasil dari kinerja algoritma *K-Nearest Neighbors* untuk mendapatkan hasil dari performa bobot dalam klasifikasi dengan menghitung nilai dari akurasi, presisi, dan recall dengan nilai K : 3 pada proses klasifikasi perangkat lunak dengan data training 90 persen, 80 persen dan 70 persen.

Tabel 1 *Confusion* matrix dengan data 90 :10

	Positive	**Negative**
Positive	21	1
Negative	1	16
Akurasi		95%
Presisi		95%
Recall		95%

Tabel 2 *Confusion* matrix dengan data 80 :20

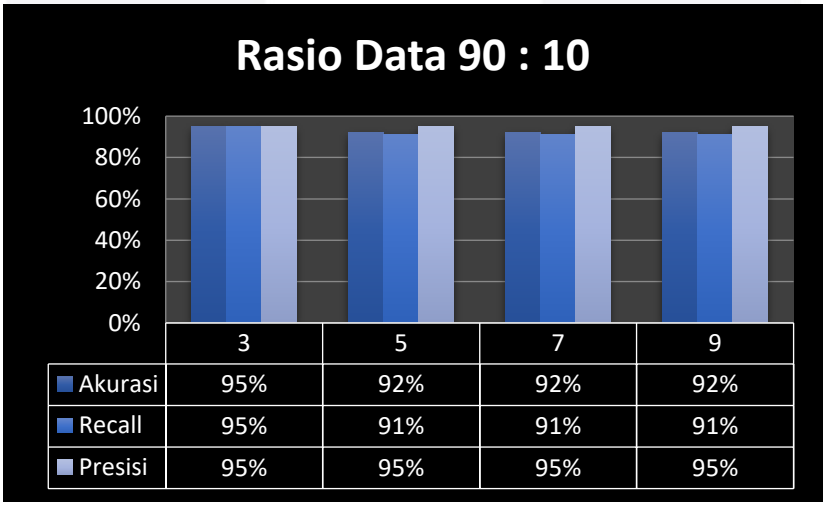
	Positive	**Negative**
Positive	34	3
Negative	3	37
Akurasi		92%
Presisi		92%
Recall		92%

Tabel 3 *Conusion* matrix dengan data 70 :30

	Positive	**Negative**
Positive	57	3
Negative	3	52
Akurasi		95%
Presisi		95%
Recall		95%

3.1 Pengujian Sistem Klasifikasi

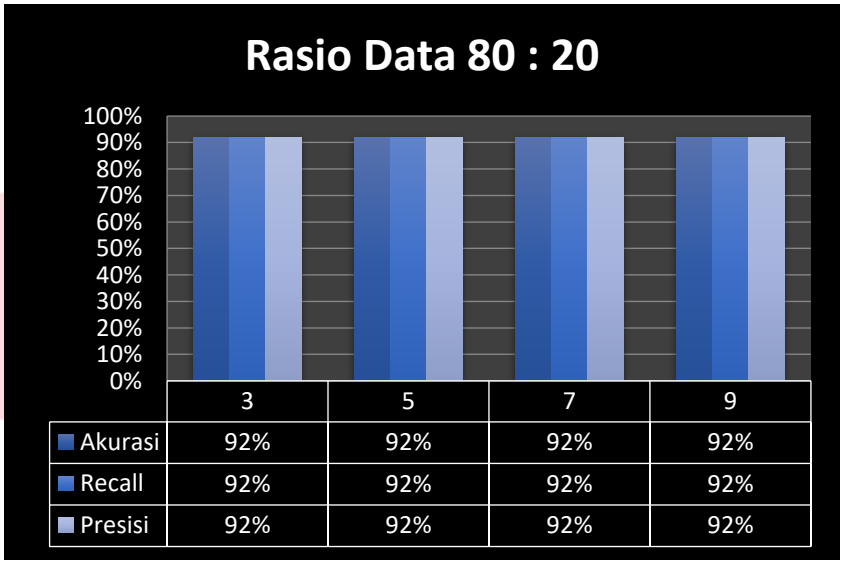
Pada pengujian dilakukan proses dalam mencari nilai K dengan rasio data latih dan data uji yang mendapatkan hasil yang optimal. Pada pengujian dilakukan nilai K yang diuji dengan nilai 3, 5, 7 dan 9 dengan menggunakan rasio data yang ditentukan pada sistem ini yaitu dengan pengujian : 90:10, 80:20 dan 70:30 yang diuji.



Gambar 4. Grafik Pengujian dengan Rasio Data 90:10

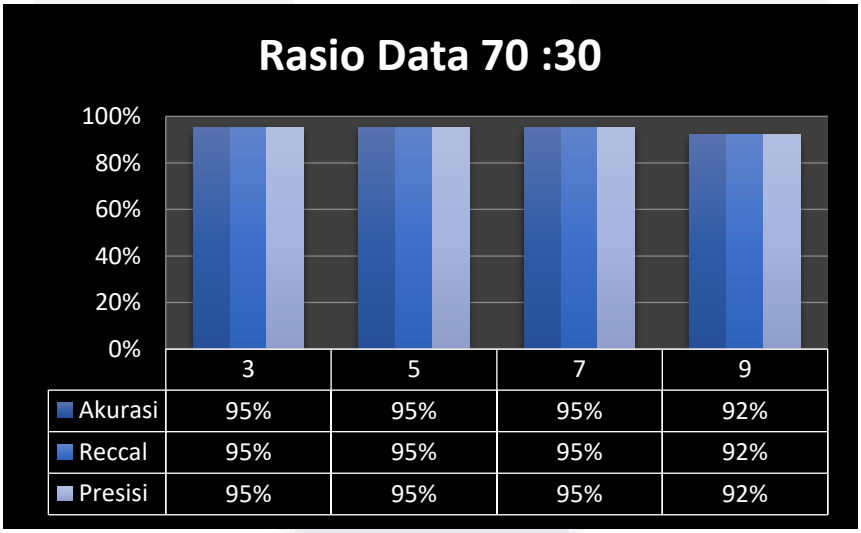
Pada gambar 4. Menjelaskan hasil pengujian yang dilakukan sebanyak 5 kali pengujian dengan mendapatkan hasil parameter nilai k : 3 mendapatkan hasil nilai presisi sebesar 95%, nilai recall 95% dan nilai akurasi 95% pada proses klasifikasi data dinamis. Parameter nilai k terdapat banyak tetangga yang dibandingkan dalam data latih dan data uji, sehingga dapat diklasifikasikan sesuai dengan kelasnya. Pada nilai K 5, 7, dan 9 dalam pengujian data

90:10 menggalami penaikan dan penurunan karena didalam data latih dan data uji tidak sesuai dan jika nilai K semakin tinggi maka jumlah tetangga samakin kabur.



Gambar 5. Grafik Pengujian dengan Rasio Data 80:20

Pada gambar 5. Menjelaskan nilai paramater menggunakan nilai K : 3 memiliki nilai akurasi yang tinggi dengan data latih 80% dan data uji 20%, sesuai dengan kelasnya mendapatkan hasil nilai presisi 92%, nilai recall 92% dan nilai akurasi 92%, pada saat parameter nilai k : 5, 7 dan 9 mendapatkan akurasi yang sama dan tidak ada penurunan karena nilai semakin naik dan nilai k yang tinggi yang dipakai tidak mempengaruhi banyaknya tetangga.



Gambar 6. Grafik Pengujian dengan Rasio Data 70:30

Pada gambar 6. Menjelaskan pengujian yang dilakukan sebanyak 5 kali pada tabel dijelaskan bahwa pengujian yang dilakukan dengan dengan K=3 mendapatkan hasil akurasi tinggi pada pengujian dengan data uji dan data latih 70:30 dengan tingkat nilai presisi sebesar 95%, recall 95% dan nilai akurasi mencapai 95%. Nilai k 3, 5 dan 7 memiliki hasil yang sama dan pada nilai k: 9 mendapatkan nilai akurasi, presisi dan reccal 92% karena semakin tinggi nilai k yang dimasukan maka semakin sedikit jumlah tetangga.

3.2 Pengujian Kfold Cross Validation

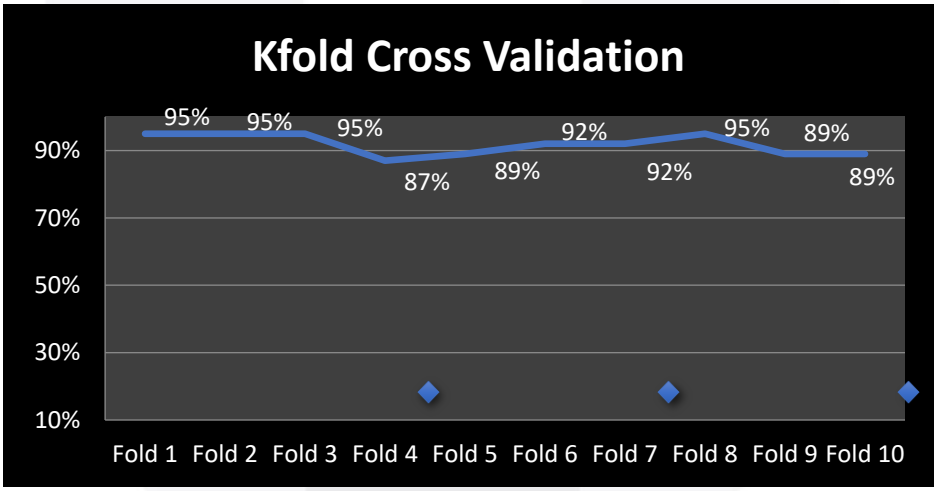
Cross validation merupakan pemodelan validasi yang digunakan untuk menilai dan mengeneralisasikan dari kumpulan data dalam melakukan prediksi. Teknik *cross validation* yang digunakan untuk membagi data menjadi k sebagai dataset yang dibagi sama rata dengan jumlah dan ukuran yang sama.

Pada pengujian ini dilakukan dengan pembagian data train dan data test dengan semua dataset diuji.

Tabel 4. Partisi 10 CV Data Train dan Data Test

Fold	Train Set	Test Set
Fold 1	343	39
Fold 2	343	39
Fold 3	344	38
Fold 4	344	38
Fold 5	344	38
Fold 6	344	38
Fold 7	344	38
Fold 8	344	38
Fold 9	344	38
Fold 10	344	38

Setelah melakukan pembagian data testing dan data training, pada pengujian ini menggunakan 10 cross validation yang dimana melakukan proses testing pertama samapai 10 fold dengan melakukan data diuji dengan secara keseluruhan.



Gambar 7. Grafik Pengujian menggunakan 10 fold

Pada grafik di atas merupakan proses pengujian yang dilakukan menggunakan 10 data yang telah dibagi menjadi data train dan data test dengan hasil yang didapat saat pengujian 10 fold yaitu nilai fold 1, fold 2 dan fold 3 mendapatkan hasil nilai tertinggi dan nilai terendah pada fold 4 dengan jumlah nilai rata-rata dalam pengujian *cross validation* 92%.

4. Kesimpulan

Berdasarkan pengujian yang telah dilakukan di penelitian Tugas Akhir ini dapat disimpulkan bahwa :

1. Pada pengujian yang dilakukan dalam klasifikasi perangkat lunak Malware dan goodware menggunakan proses analisis dinamis, karakteristik dalam perangkat lunak sebagai acuan data untuk proses klasifikasi menggunakan algoritma *K-Nearst Neighbors*.
2. Proses pelatihan klasifikasi menggunkan algoritma *K-Nearest neighbors* menggunakan parameter nilai K : 3 dengan rasio data 90% dapa latih dan 10% data uji, berdasarkan pengujian yang telah dilakukan mendapatkan hasil akurasi sebesar 95% dan menggunakan Kflop Cross Validation mendapatkan hasi 92%.

Referensi:

[1] Sethi, K., Kumar, R., Sethi, L., Bera, P., & Patra, P. K. "A novel machine learning based malware detection and classification framework," In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*, IEEE,2019, pp. 1-4.

[2] K. O. Babaagba and S. O. Adesanya, "A study on the effect of feature selection on malware analysis using machine learning," *in ACM International Conference Proceeding Series*, 2019, vol. Part F1481, pp. 51–55.

[3] O. Aslan and R. Samet, "Investigation of possibilities to detect malware using existing tools," *in Proceedings of IEEE/ACS International Conference on Computer Systems and Applications*, AICCSA, 2018, vol. 2017-Octob, pp. 1277–1284, doi: 10.1109/AICCSA.2017.24.

[4] T. A. Cahyanto, V. Wahanggara and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Analisis Dinamis," JUSTINDO, Jurnal Sistem & Teknologi Informasi Indonesia , vol. II, no. 1, pp. 19-30, 2017.

[5] Or-Meir, O., Nissim, N., Elovici, Y., & Rokach, L. "Dynamic malware analysis in the modern era—A state of the art survey," *ACM Computing Surveys (CSUR)*, *52*(5), 1-48, 2019.

[6] Eze, A. O., & Chukwunonso, C. (2018). "Malware Analysis and Mitigation in Information Preservation," *IOSR Journal of Computer Engineering* (IOSRJCE),20(4), 1st ser., 53-62. Diakses pada 1 Juni, 2019

[7] Deka, D., Sarma, N., & Panicker, N. J. "Malware detection vectors and analysis techniques: A brief survey," In *2016 International Conference on Accessibility to Digital World (ICADW)* (pp. 81-85). IEEE.

[8] Ray, S., "A quick review of machine learning algorithms," In *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)* (pp. 35-39). IEEE,2019.

[9] Ramadhan, B., Purwanto, Y., & Ruriawan, M. F., "Forensic Malware Identification Using Naive Bayes Method," In *2020 International Conference on Information Technology Systems and Innovation (ICITSI)* (pp. 1-7). IEEE,2020.

[10] Sandag, G. A., Tedry, N. E., & Lolong, S., "Classification of lower back pain using K-Nearest Neighbor algorithm," In *2018 6th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-5). IEEE.

[11] Ghazy Irfan Rusydy, Budhi Irawan, dan Casi Setianingsih, *" Android Based Sea Wave Detection System Usbig KNN(K-NEAREST NEIGHBOR) Algorithm,"*e-Proceeding of Engineering : Vol.7, 2020.

[12] Ndaumanu, R. I., & Arief, M. R., "Analisis prediksi tingkat pengunduran diri mahasiswa dengan metode K-Nearest Neighbor," *Yogyakarta: STMIK AMIKOM Yogyakarta, 2014*.

[13] Ijaz, M., Durad, M. H., & Ismail, M., "Static and dynamic malware analysis using machine learning," In *2019 16th International bhurban conference on applied sciences and technology (IBCAST)* (pp. 687-691). IEEE.