Jurnal ELTIKOM, Vol. 2, No. 2, Desember 2018, hal. 58-66
ISSN 2598-3245 (Print), ISSN 2598-3288 (Online)
Tersedia online di http://eltikom.poliban.ac.id
DOI : http://doi.org/10.31961/eltikom.v2i2.88

ANALISIS KLASTERISASI MALWARE: EVALUASI DATA TRAINING DALAM PROSES KLASIFIKASI MALWARE

Denar Regata Akbi[1)], Arini R Rosyadi[2)]
[1, 2)] Universitas Muhammadiyah Malang
e-mail: dnarregata@umm.ac.id[1)], arini.rosyadi@gmail.com[2)]

ABSTRACT

Training data is an important part of the classification process. Especially if the data is used to create a malware detection system. This study compared training data generated from two previous studies, the data used in both studies were android malware data based on the frequency of the system call of 600 data. The first study classifies and produces 4 types of malware, while the second research clusters and produces 8 clusters. From the two studies, researchers evaluated training data from each study to get more accurate results of training data, using 50 test data, researchers conducted evaluations and trials using the kNN algorithm. The results obtained, the use of training data based on the results of clustering in the classification process is more recommended, the results of the first error Prediction study: 0.995 while in the second study: 0.998. Recall results and accuracy using the cross-validation method, the first study, Recall: 0.665 accuracy: 0.66, second study, Recall: 0.893 accuracy: 0.89, while the Recall Results and accuracy using the percentage split method, the first study, Recall: 0.657 accuracy: 0.65, second study, Recall: 0.798 accuracy: 0.79. Based on the results, the clustering process that uses frequency data from the malware system calls, produces training data that is more accurate than the training data produced by naming software for malware.

Keywords*: training data, classification process, clustering process, malware, system call.*

ABSTRAK

Data latih merupakan salah satu bagian penting pada proses klasifikasi. Terutama jika data tersebut digunakan untuk membuat sistem pendeteksi malware. Penelitian ini melakukan perbandingan data latih yang dihasilkan dari dua penelitian yang telah dilakukan sebelumnya, data yang digunakan pada kedua penelitian tersebut merupakan data malware android berdasarkan frekuensi system call sejumlah 600 data. Penelitian pertama melakukan klasifikasi dan menghasilkan 4 jenis malware, sedangkan penelitian kedua melakukan klastering dan menghasilkan 8 klaster. Dari kedua penelitian tersebut, peneliti melakukan evaluasi data latih dari masing - masing penelitian untuk mendapatkan hasil data latih yang lebih akurat, dengan menggunakan data uji sejumlah 50, peneliti melakukan evaluasi dan uji coba dengan menggunakan algoritme kNN. Hasil yang didapatkan, penggunaan data latih berdasarkan hasil klastering pada proses klasifikasi lebih direkomendasikan, hasil Error Prediction penelitian pertama: 0,995 sedangkan pada penelitian kedua: 0,998. Hasil Recall dan akurasi menggunakan metode cross validation, penelitian pertama, Recall: 0,665 akurasi: 0,66, penelitian kedua, Recall: 0,893 akurasi: 0,89, sedangkan Hasil Recall dan akurasi menggunakan metode precentage split, penelitian pertama, Recall: 0,657 akurasi: 0,65, penelitian kedua, Recall: 0,798 akurasi: 0,79. Berdasarkan hasil pengujian, proses klastering yang menggunakan data frekuensi system call malware menghasilkan data latih yang lebih akurat dibandingkan dengan data latih yang dihasilkan dengan menggunakan suatu situs penamaan malware.

Kata Kunci*: data latih, klasifikasi, klasterisasi, malware, system call.*

I. PENDAHULUAN

DATA latih merupakan salah satu bagian penting pada proses klasifikasi. Menurut [1] data latih yang digunakan dalam teknik pembelajaran, seperti untuk melakukan proses prediksi, harus memiliki data latih yang baik sehingga menghasil hasil prediksi yang baik pula. Pada tulisan lain [2] menyebutkan bahwa salah satu aspek penting yang wajib untuk diperhatikan dalam suatu proses klasifikasi adalah data latih yang digunakan sebagai model pembelajaran. Data yang digunakan pada penelitian ini, baik data latih ataupun data uji merupakan data *system call* android, *system call* merupakan mekanisme dari suatu program yang terpasang pada sistem operasi untuk meminta layanan

pada *kernel* sistem operasi, proses seberapa sering suatu program meminta layanan pada *kernel* sistem operasi dinamakan frekuensi *system call*, dari frekuensi permintaan layanan tersebut, didapatkan fitur – fitur dari *malware* untuk digunakan pada penelitian ini [3] [4] [5] [6].

Oleh beberapa pengembang, *malware* dirancang untuk melakukan kerusakan pada suatu sistem yang diinfeksi, aktivitas yang dilakukan diantaranya menggangu kinerja komputer, sistem operasi, mencuri data atau informasi rahasia secara ilegal [7] [8] [9]. Sehingga dengan begitu dibutuhkan adanya sistem yang dapat mendeteksi atau pun mencegah serangan *malware* [10]. Dari banyak teknik yang digunakan dalam sistem pendeteksi *malware* (*anti-malware*) teknik utama yang digunakan adalah *Signature-Based*, *Anomaly-Based* dan *Specification-Based*. Pada setiap teknik tersebut dapat digunakan pendekatan statis, dinamis maupun *hybrid* [11]. Berdasarkan karakteristiknya, *malware* terbagi menjadi beberapa variasi diantaranya adalah *virus, trojan-horse, worm, backdoor, adware, spyware, rootkit* dan lain sebagainya, walaupun memiliki karakteristik yang berbeda, tetapi tidak menutup kemungkinan terdapat *malware* yang memiliki lebih dari satu variasi dalam waktu yang sama [12].

Terdapat beberapa penelitian tentang *malware* yang melibatkan teknik data mining. Penelitian pertama dilakukan oleh Sendi dkk [13]. Penelitian ini dilakukan dengan mengumpulkan *malware*, yang nantinya digunakan pada dua aktivitas (penelitian) yang berbeda, yang berjalan pada sistem operasi android pada ponsel pintar. Selanjutnya *malware-malware* yang telah didapatkan dilakukan proses pelabelan secara manual menggunakan situs *www.virustotal.com* untuk mengetahui jenis *malware* dan banyakknya jenis *malware* yang digunakan sebagai data set. Selanjutnya penelitian ini menggunakan metode *k-fold Cross Validation* untuk menguji tingkat akurasi data latih pada proses klasifikasi menggunakan Algoritme SVM (*Support Vector Machine*).

Penelitian selanjutnya dilakukan oleh Denar [14] yaitu melakukan pengklasteran terhadap data set yang sama dengan data set yang digunakan pada penelitian [13] dengan menggunakan Algoritme k-Means. Penelitian ini menghasilkan nilai *Sum Square Error* yang digunakan sebagai pengujian hasil klaster dan juga penentuan nilai *k* yang tepat sehingga layak untuk digunakan sebagai data latih.

Kedua penelitian yang telah disebutkan adalah penelitian yang dilakukan dengan tujuan untuk mendapatkan data latih yang baik yang dapat digunakan pada sebuah sistem pendeteksi *malware* berdasarkan pada frekuensi *system call* dari *malware*. Selain itu berdasarkan pada [1] dengan digunakannya data latih yang baik maka sistem pendeteksi *malware* yang dikembangkan dapat mendeteksi *malware* dengan baik dan tepat. Maka berdasarkan penelitian-penelitian tersebut, penelitian ini melakukan pembandingan dan analisis terhadap data latih yang dihasilkan oleh keduanya. Dengan harapan didapatkan data latih yang baik untuk dapat digunakan dalam suatu sistem pendeteksi *malware*. Pembandingan dan pengujian dilakukan dengan menguji tingkat akurasi dari data latih.

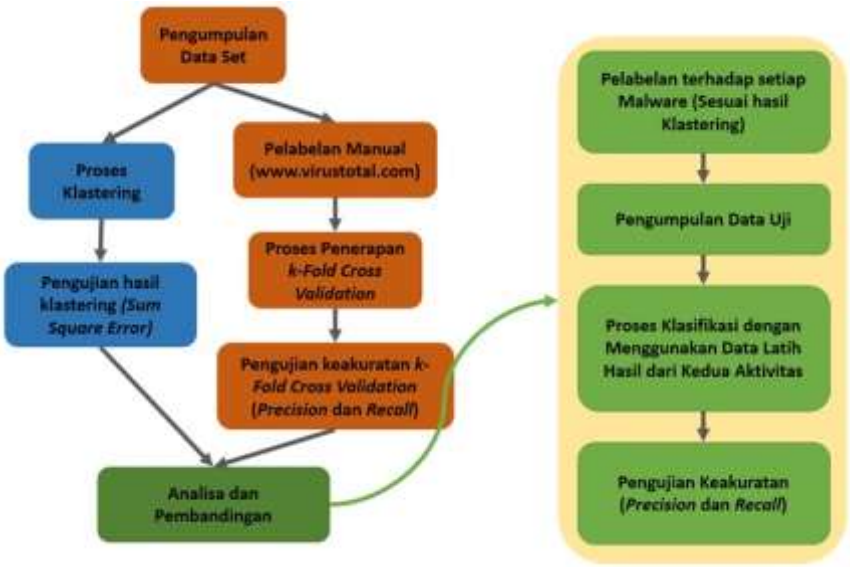
II. METODE PENELITIAN

Penelitian ini melakukan beberapa kegiatan yang diberikan pada Gambar 1. Pada beberapa aktivitas dalam penelitian ini telah dilakukan pada penelitian sebelumnya, sehingga hasil uji coba telah didapatkan pada penelitian sebelumnya. Pada Gambar 1 aktivitas yang dilakukan pada penelitian sebelumnya adalah aktivitas yang berwarna jingga dan biru. Dimana Aktivitas-aktivitas berwarna jingga adalah penelitian yang dilakukan oleh [13] dan Aktivitas-aktivitas berwarna biru adalah penelitian sebelumnya yang dilakukan oleh [14]. Sedangkan aktivitas pada penelitian ini adalah aktivitas berwarna hijau.

Secara garis besar penelitian ini melakukan analisis dari kedua penelitian yang telah dilakukan sebelumnya untuk mendapatkan data set yang baik sehingga layak untuk digunakan sebagai data latih pada suatu sistem pendeteksi *malware* pada ponsel pintar. Aktivitas pertama yang dilakukan dalam penelitian ini melakukan analisis dan perbandingan adalah melakukan pelabelan manual pada *malware* data set berdasarkan pada jumlah klaster yang dihasilkan oleh klastering. Selanjutnya mengumpulkan beberapa *malware* lain yang pada aktivitas selanjutnya digunakan sebagai data uji. Aktivitas selanjutnya adalah melakukan proses klasifikasi dengan menggunakan data latih dari kedua penelitian sebelumnya, [13] dan [14], dan mendapatkan nilai *Precision* dan *Recall* untuk menentukan keakuratan dari kedua data latih yang digunakan.

Untuk memudahkan dalam membaca dan memahami penelitian yang dilakukan, pada penjabaran

selanjutnya data latih berdasarkan hasil proses klastering [14] disebut dengan Data Latih A dan data latih berdasarkan penelitian yang dilakukan oleh [13] disebut dengan Data Latih B.



Gambar 7. Alur Kegiatan Penelitian

A. Pelabelan Data Latih A

Pada penelitian [14], data set yang digunakan hanya dilakukan proses klastering tanpa proses pelabelan terhadap setiap *malware*. Sehingga pada aktivitas ini, dilakukan pemberian label terhadap setiap *malware* yang berada pada data set sesuai dengan hasil klaster. Penamaan ini dilakukan dengan pemberian nama *malwareCluster0*, *malwareCluster1*, *malwareCluster2* sampai dengan *malwareCluster7* sesuai dengan nilai *k* yang baik berdasarkan penelitian [14].

B. Pengumpulan Data Set sebagai Data Uji

Untuk menguji layak tidaknya suatu data latih menjadi suatu model dalam proses klasifikasi, tentunya dibutuhkan adanya data uji. Data uji digunakan selama proses klasifikasi dan dilakukan perhitungan nilai *Precision* dan *Recall* untuk dapat mengetahui tingkat akurasi dari proses klasifikasi yang dilakukan.

Aktivitas ini dilakukan berdasarkan pada penelitian [13]. *Malware* yang dikumpulkan pada aktivitas ini selanjutnya digunakan sebagai data uji pada proses klasifikasi. Sehingga sampai pada aktivias ini terdapat dua jenis data yang digunakan, yaitu data latih dari kedua penelitian sebelumnya (Data Latih A dan Data Latih B) dan data uji yang dikumpulkan pada aktivitas ini.

C. Proses Klasifikasi dengan Menggunakan Dua Data Latih

Pada aktivitas ini, dilakukan dua proses klasifikasi dengan menggunakan algoritme klasifikasi yang sama yaitu, Algoritme kNN, dan juga data uji yang sama. Yang membedakan dari kedua proses ini adalah data latih yang digunakan, yaitu Data Latih A dan Data Latih B.

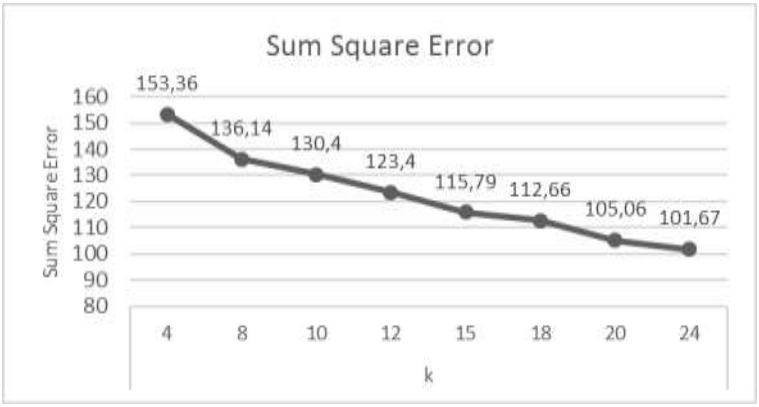
D. Pengujian Keakuratan Data Latih Kedua Proses Klasifikasi

Aktivitas ini melakukan pengujian terhadap tingkat keakuratan dari kedua data latih yang digunakan pada proses klasifikasi. Hasil ini selanjutnya digunakan untuk proses analisis dan perbandingan. Pengujian keakuratan ini dilakukan dengan menghitung nilai dari *Precision* dan *Recall*.

III. HASIL DAN PEMBAHASAN

A. Pelabelan Data Latih Berdasarkan Hasil Klastering

Berdasarkan pada penelitian [14] didapatkan nilai *k* yang baik berdasarkan perhitungan nilai *Sum Square Error* adalah *k* sama dengan 8 dari beberapa percobaan variasi nilai *k*. Hal ini dapat dilihat pada Gambar 2 yang menunjukkan *elbow* pada grafik *Sum Square Error* berada pada *k* sama dengan 8.



Gambar 8. Grafik Nilai Sum Square Error dari Hasil Proses Klastering dengan Variasi Nilai k

Berdasarkan pada Tabel I pada nilai *k* sama dengan 8 maka didapatkan bahwa setiap klaster memiliki beberapa instans didalamnya. Terlihat bahwa klaster dengan jumlah terbanyak adalah klaster ketiga dengan jumlah instans adalah 410 *malware* dan klaster dengan jumlah instans terendah adalah klaster keempat dengan jumlah 2 *malware*. Dan delapan jenis *malware* sesuai dengan proses klastering yang dilakukan pada penelitian [14] adalah *malwareCluster0*, *malwareCluster1*, *malwareCluster2*, *malwareCluster3*, *malwareCluster4*, *malwareCluster5*, *malwareCluster6* dan *malwareCluster7*.

TABEL I.
HASIL PELABELAN MALWARE DALAM DATASET

Cluster0	***Cluster1***	***Cluster2***	***Cluster3***	***Cluster4***	***Cluster5***	***Cluster6***	***Cluster7***
25	46	1	2	298	3	128	42
76	101	12	4	335	6	130	103
82	112	41	5		7	141	105
96	154	48	8		20	143	124
98	170	90	9		52	147	
…	…	122	…		…	…	
…	…	251	…		…	…	
585	389		599		244	240	
592	584		600		246	247	
61	**11**	**7**	**410**	**2**	**92**	**13**	**4**

B. Pengumpulan Data Uji

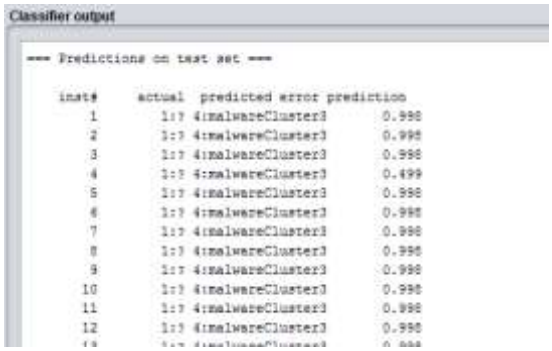
Data uji yang dikumpulkan pada aktivitas ini, adalah data *malware* yang sejenis dengan data set yang digunakan sebagai data latih. Yaitu berupa *malware* untuk sistem operasi android pada ponsel pintar yang nantinya didapatkan frekuensi *system call* yang dipanggil selama *malware* tersebut berjalan dalam sistem operasi. Situs yang digunakan untuk mendapatkan *malware* yang digunakan sebagai data uji adalah *www.virusshare.com.*

C. Pengujian Keakuratan Data Latih Berdasarkan Kedua Penelitian

Pada tahapan ini, dilakukan pengujian terhadap tingkat keakuratan dari data latih berdasarkan kedua penelitian yang telah dilakukan sebelumnya. Pengujian dilakukan dengan beberapa jenis percobaan, yaitu (1) Pengujian menggunakan data uji, (2) Pengujian menggunakan metode *Cross Validation* dan (3) Pengujian menggunakan metode *Percentage Split*. Ketiga pengujian tersebut menggunakan Aplikasi WEKA untuk mendapatkan nilai akurasinya.

- *Pengujian Keakuratan Data Latih A*

Data latih yang digunakan dalam pengujian ini adalah data latih yang dimana setiap *malware* telah masuk dalam salah satu dari delapan jenis *malware* yang ditentukan. Jumlah *malware* dalam tiap jenis *malware* diberikan pada Tabel 1. Pengujian tingkat akurasi pada data latih ini dilakukan pada tiga jenis pengujian, yaitu:



Classifier output

=== Predictions on test set ===

inst#	actual	predicted	error	prediction
1	1:?	4:malwareCluster3		0.998
2	1:?	4:malwareCluster3		0.998
3	1:?	4:malwareCluster3		0.998
4	1:?	4:malwareCluster3		0.499
5	1:?	4:malwareCluster3		0.998
6	1:?	4:malwareCluster3		0.998
7	1:?	4:malwareCluster3		0.998
8	1:?	4:malwareCluster3		0.998
9	1:?	4:malwareCluster3		0.998
10	1:?	4:malwareCluster3		0.998
11	1:?	4:malwareCluster3		0.998
12	1:?	4:malwareCluster3		0.998

Gambar 9. Pengujian Data Latih A Menggunakan Data Uji

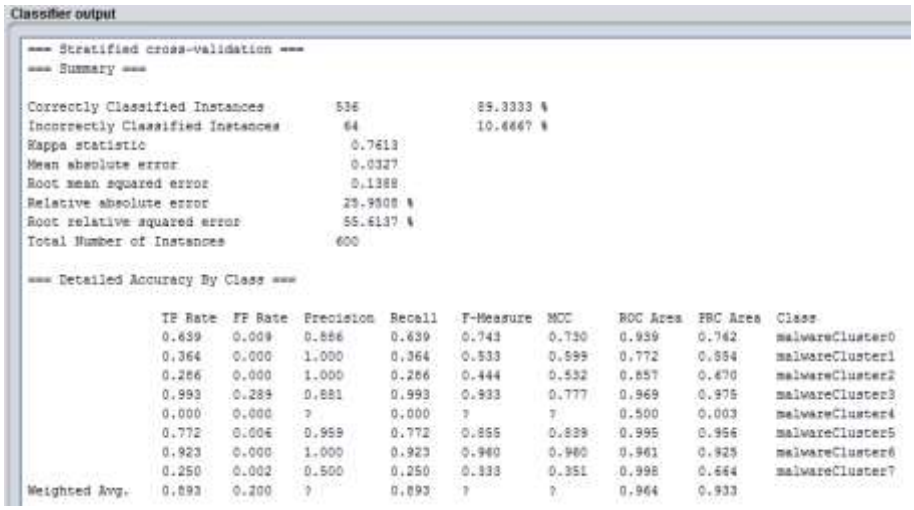
(1) Pengujian menggunakan data uji.

Data uji yang digunakan pada perbandingan ini adalah data uji yang berisi 50 *malware* yang didapatkan secara acak dan tidak diketahui jenis dari *malware* tersebut. Sehingga dengan menggunakan proses klasifikasi menggunakan WEKA dan Algoritme kNN, dapat diketahui jenis dari setiap *malware*.

Gambar 3 menunjukkan hasil Dari proses klasifikasi yang dilakukan dengan menggunakan data uji. Pengujian ini menghasilkan prediksi dari setiap *malware* yang terdapat dalam data uji bahwa seluruh *malware* masuk dalam jenis *malwareCluster3* dengan nilai *Error Prediction* adalah 0.998 dan 0.499.

(2) Pengujian Menggunakan Metode *Cross Validation.*

Berdasarkan pada Gambar 4, pengujian dilakukan dengan proses klasifikasi yang pilihan pengujiannya menggunakan metode *Cross Validation*. Dari pengujian ini didapatkan nilai akurasi sebesar 89.33% dengan jumlah *malware* yang berada pada jenis *malware* yang benar adalah 536 dari total 600 *malware* yang digunakan. Sedangkan untuk tingkat akurasi data latih berdasarkan dari rata-rata nilai *Precision* adalah tidak diketahui (?) dan rata-rata nilai *Recall* adalah 0.893.



Classifier output

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	536	89.3333 %
Incorrectly Classified Instances	64	10.6667 %
Kappa statistic	0.7613	
Mean absolute error	0.0327	
Root mean squared error	0.1388	
Relative absolute error	29.9508 %	
Root relative squared error	55.6137 %	
Total Number of Instances	600	

=== Detailed Accuracy By Class ===

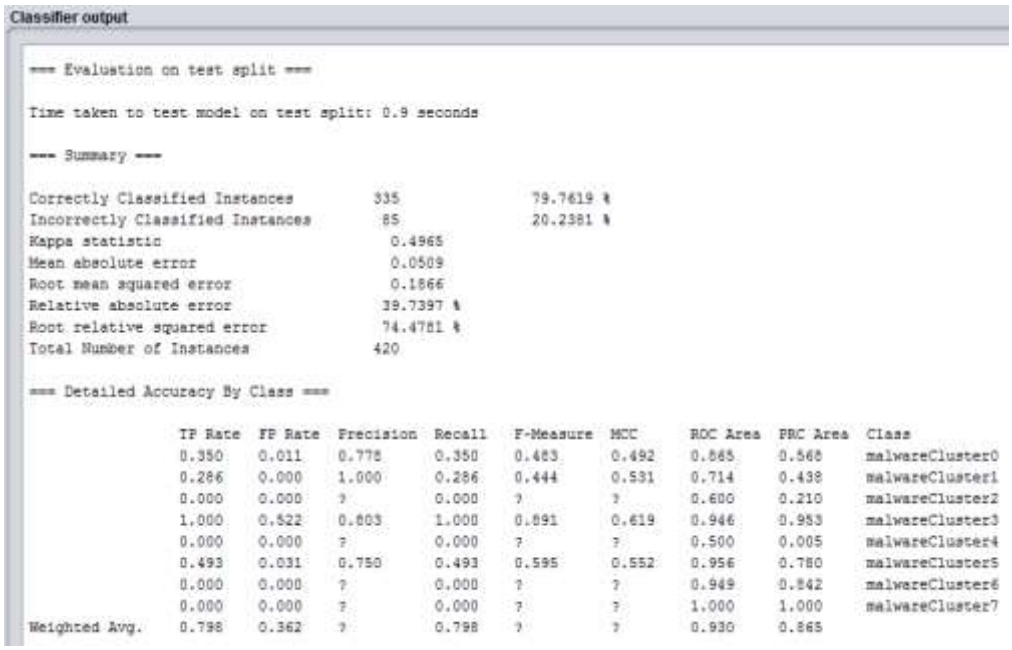
	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.639	0.009	0.886	0.639	0.743	0.730	0.939	0.762	malwareCluster0
	0.364	0.000	1.000	0.364	0.533	0.599	0.772	0.554	malwareCluster1
	0.286	0.000	1.000	0.286	0.444	0.532	0.857	0.670	malwareCluster2
	0.993	0.289	0.881	0.993	0.933	0.777	0.969	0.975	malwareCluster3
	0.000	0.000	?	0.000	?	?	0.500	0.003	malwareCluster4
	0.772	0.006	0.959	0.772	0.855	0.839	0.995	0.956	malwareCluster5
	0.923	0.000	1.000	0.923	0.960	0.960	0.961	0.925	malwareCluster6
	0.250	0.002	0.500	0.250	0.333	0.351	0.998	0.664	malwareCluster7
Weighted Avg.	0.893	0.200	?	0.893	?	?	0.964	0.933	

Gambar 10. Pengujian Data Latih A Menggunakan Metode Cross Validation

(3) Pengujian Menggunakan Metode *Percentage Split*.

Pada pengujian ini digunakan metode *Percentage Split*, dimana data latih yang dimiliki dibagi sesuai dengan porsi yang ditentukan untuk digunakan sebagai data latih dan data uji. Pada pengujian ini, pembagian dilakukan dengan porsi 70% dari total data latih dan 30% digunakan sebagai data uji.

Berdasarkan pada Gambar 5, pada pengujian proses klasifikasi, yang pilihan pengujiannya menggunakan metode *Percentage Split*, didapatkan nilai akurasi sebesar 79.77% dengan jumlah *malware* yang berada pada jenis yang benar adalah 335 dari total 420 *malware* (70%) yang digunakan sebagai data latih dan 85 *malware* berada pada jenis yang tidak sesuai. Sedangkan untuk tingkat akurasi data latih berdasarkan dari rata-rata nilai *Precision* adalah tidak diketahui (?) dan rata-rata nilai *Recall* adalah 0.798.



Classifier output

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.9 seconds

=== Summary ===

Correctly Classified Instances         335               79.7619 %
Incorrectly Classified Instances        85               20.2381 %
Kappa statistic                          0.4965
Mean absolute error                      0.0509
Root mean squared error                  0.1866
Relative absolute error                 39.7397 %
Root relative squared error             74.4781 %
Total Number of Instances              420

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.350    0.011    0.778      0.350   0.483      0.492    0.865     0.568     malwareCluster0
               0.286    0.000    1.000      0.286   0.444      0.531    0.714     0.438     malwareCluster1
               0.000    0.000    ?          0.000   ?          ?        0.600     0.210     malwareCluster2
               1.000    0.522    0.803      1.000   0.891      0.619    0.946     0.953     malwareCluster3
               0.000    0.000    ?          0.000   ?          ?        0.500     0.005     malwareCluster4
               0.493    0.031    0.750      0.493   0.595      0.552    0.956     0.780     malwareCluster5
               0.000    0.000    ?          0.000   ?          ?        0.949     0.842     malwareCluster6
               0.000    0.000    ?          0.000   ?          ?        1.000     1.000     malwareCluster7
Weighted Avg.  0.798    0.362    ?          0.798   ?          ?        0.930     0.865
```

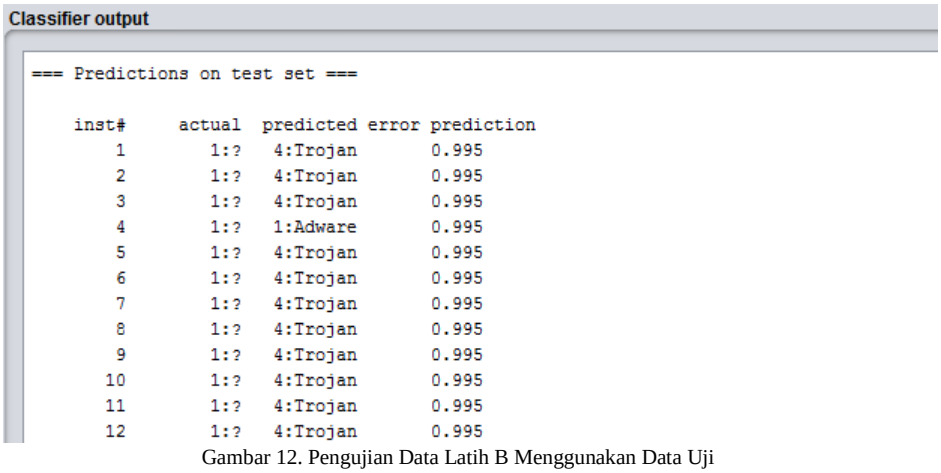
Gambar 11. Pengujian Data Latih A Menggunakan Metode Percentage Split

- *Pengujian Keakuratan Data Latih B*

Data latih ini didapatkan berdasarkan pada penelitian [13], dimana terdapat empat jenis *malware* yaitu *Adware*, *Trojan*, *Plankton* dan *DroidKungfu*. Seperti pada data latih berdasarkan hasil klastering, data latih ini juga dilakukan tiga pengujian yang berbeda untuk mendapatkan tingkat akurasinya. Pengujian ini menggunakan data uji dan skenario pengujian yang sama dengan yang digunakan pada pengujian menggunakan data uji pada Data Latih A.

(1) Pengujian menggunakan data uji.

Berdasarkan pada Gambar 6 didapatkan bahwa dari total 50 *malware* yang terdapat dalam data uji, *malware-malware* tersebut terbagi dalam beberapa jenis *malware* yang berbeda. Hasil klasifikasi menyebutkan bahwa 40 *malware* diprediksi sebagai *Trojan*, 6 *malware* sebagai *Adware* dan 4 *malware* sebagai *Plankton*. Dengan jumlah *malware* yang berbeda pada setiap jenis *malware* yang ada didapatkan nilai *Error Prediction* dari seluruh *malware* adalah 0.995.



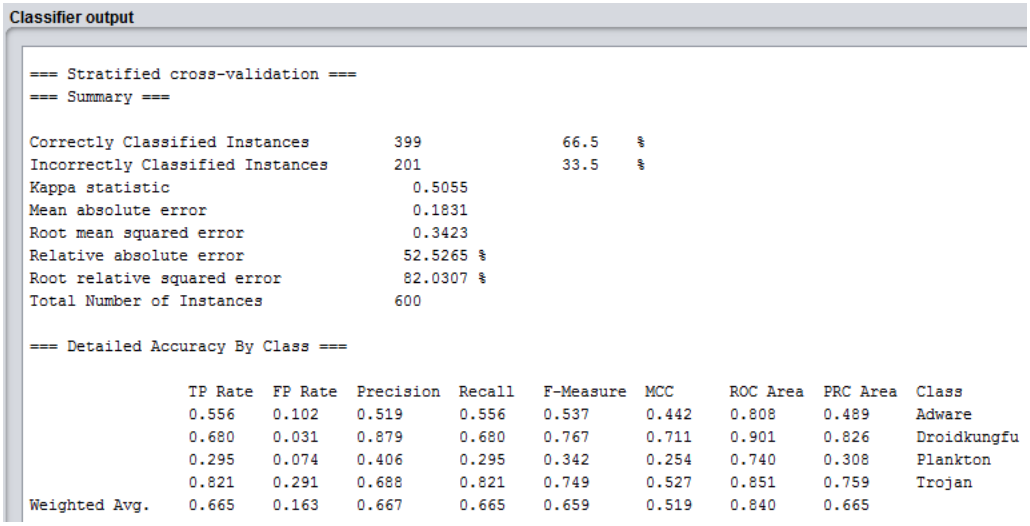
Classifier output

```
=== Predictions on test set ===

    inst#     actual  predicted error prediction
        1        1:?   4:Trojan       0.995
        2        1:?   4:Trojan       0.995
        3        1:?   4:Trojan       0.995
        4        1:?   1:Adware       0.995
        5        1:?   4:Trojan       0.995
        6        1:?   4:Trojan       0.995
        7        1:?   4:Trojan       0.995
        8        1:?   4:Trojan       0.995
        9        1:?   4:Trojan       0.995
       10        1:?   4:Trojan       0.995
       11        1:?   4:Trojan       0.995
       12        1:?   4:Trojan       0.995
```

Gambar 12. Pengujian Data Latih B Menggunakan Data Uji

(2) Pengujian Menggunakan Metode *Cross Validation*.

Berdasarkan pada Gambar 7, pada pengujian terhadap data latih B, didapatkan nilai akurasi dari data latih yang digunakan sebesar 66.5% dengan jumlah *malware* yang berada pada jenis *malware* yang benar adalah 399 dari total 600 *malware* yang digunakan dan *malware* yang tidak berada pada jenis yang benar adalah 201 *malware*. Sedangkan untuk tingkat akurasi data latih berdasarkan dari rata-rata nilai *Precision* dan *Recall* adalah 0.667 dan 0.665.



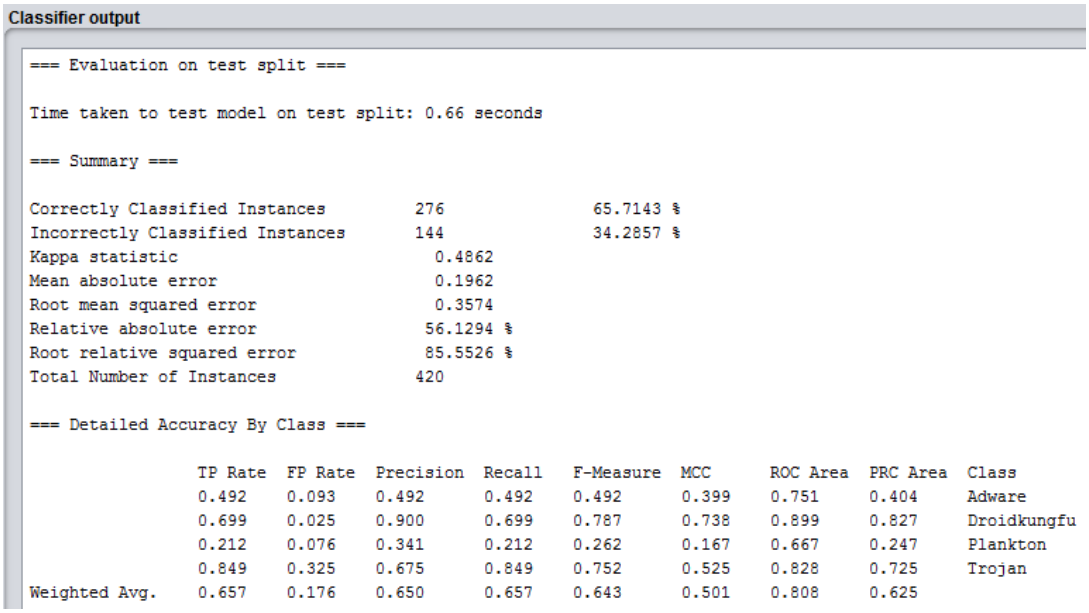
Classifier output

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         399               66.5    %
Incorrectly Classified Instances       201               33.5    %
Kappa statistic                          0.5055
Mean absolute error                      0.1831
Root mean squared error                  0.3423
Relative absolute error                 52.5265 %
Root relative squared error             82.0307 %
Total Number of Instances              600

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.556     0.102     0.519       0.556   0.537      0.442    0.808     0.489     Adware
               0.680     0.031     0.879       0.680   0.767      0.711    0.901     0.826     Droidkungfu
               0.295     0.074     0.406       0.295   0.342      0.254    0.740     0.308     Plankton
               0.821     0.291     0.688       0.821   0.749      0.527    0.851     0.759     Trojan
Weighted Avg.  0.665     0.163     0.667       0.665   0.659      0.519    0.840     0.665
```

Gambar 13. Pengujian Data Latih B Menggunakan Metode Cross Validation

(3) Pengujian Menggunakan Metode *Percentage Split*.

Berdasarkan pada Gambar 8, pada pengujian ini didapatkan nilai akurasi dari data latih sebesar 65.72% dengan jumlah *malware* yang berada pada jenis *malware* yang benar adalah 276 dari total 420 *malware* (70%) dan *malware* yang tidak berada pada jenis yang benar adalah 144 *malware*. Sedangkan untuk tingkat akurasi data latih berdasarkan dari nilai rata-rata *Precision* dan *Recall* adalah 0.650 dan 0.657.



Classifier output

```
=== Evaluation on test split ===

Time taken to test model on test split: 0.66 seconds

=== Summary ===

Correctly Classified Instances         276               65.7143 %
Incorrectly Classified Instances       144               34.2857 %
Kappa statistic                          0.4862
Mean absolute error                      0.1962
Root mean squared error                  0.3574
Relative absolute error                 56.1294 %
Root relative squared error             85.5526 %
Total Number of Instances              420

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.492     0.093     0.492       0.492   0.492      0.399    0.751     0.404     Adware
               0.699     0.025     0.900       0.699   0.787      0.738    0.899     0.827     Droidkungfu
               0.212     0.076     0.341       0.212   0.262      0.167    0.667     0.247     Plankton
               0.849     0.325     0.675       0.849   0.752      0.525    0.828     0.725     Trojan
Weighted Avg.  0.657     0.176     0.650       0.657   0.643      0.501    0.808     0.625
```

Gambar 14. Pengujian Data Latih B Menggunakan Metode Percentage Split

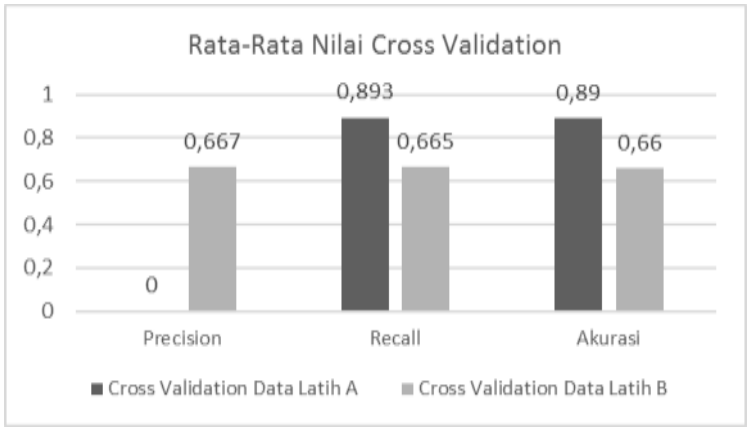
D. Anlisa dan Pembandingan

Berdasarkan pengujian yang telah dilakukan didapatkan hasil-hasil terhadap nilai *Precision*, *Recall* dan akurasi dari beberapa metode pengujian yang dilakukan. Sehingga untuk membandingkan hasil-hasil dari pengujian diatas diberikan perbandingan dibawah ini:

- *Pengujian menggunakan Data Uji*

Pada pengujian data uji, 100% data uji dengan menggunakan Data Latih A berada pada satu jenis *malware* saja yaitu *malwareCluster3* dengan variasi nilai *Error Prediction* adalah 0.495 dan 0.998, sedangkan pada data uji dengan menggunakan Data Latih B masuk pada tiga jenis *malware*, yaitu *Adware*, *Trojan* dan *Plankton*. Pada ketiga jenis *malware* tersebut 80% (40) *malware* data uji berada pada jenis *Trojan*, 12% (6) *malware* berada pada jenis *Adware* dan 8% (4) *malware* berada pada jenis *Plankton*. Dan nilai *Error Prediction* pada pengujian Data Latih B adalah 0.995.

ISSN 2598-3245 (Print)
ISSN 2598-3288 (Online)



Gambar 15. Perbandingan Rata-Rata Nilai Precision, Recall dan Akurasi dari Metode Cross Validation antara Data Latih A dan Data Latih B

- *Pengujian menggunakan Metode Cross Validation*

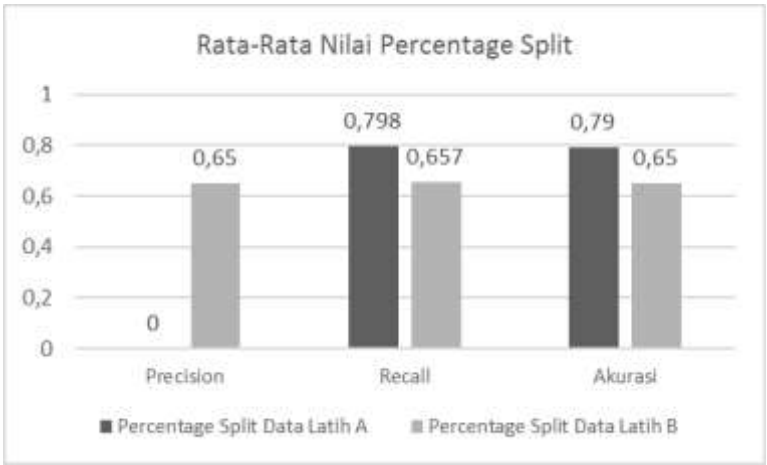
Berdasarkan hasil pengujian kedua data latih menggunakan metode *Cross Validation* didapatkan perbandingan seperti pada Gambar 9. Pada grafik tersebut terlihat bahwa pada nilai *Recall* dan akurasi, Data Latih A memiliki nilai yang lebih tinggi dibandingkan pada Data Latih B. Khusunya pada nilai akurasi, Data Latih A lebih tinggi 0.23 atau 23% dari pada nilai akurasi pada Data Latih B.

Sedangkan untuk rata-rata nilai *Precision* dianggap bernilai nol karena pada salah satu klaster, yaitu *malwareCluster4*, nilai *False Positive* dan *True Positive* bernilai 0. Dengan kata lain, tidak terdapat malware yang dikatakan berjenis *malwareCluster4* yang diklasifikasikan secara benar atau salah oleh WEKA berada pada klaster *malwareCluster4.*

- *Pengujian menggunakan Metode Percentage Split*

Hasil pengujian kedua data latih menggunakan metode *Percentage Split*, dengan pembagian 70% untuk data latih dan 30% untuk data uji, didapatkan perbandingan seperti pada Gambar 10. Pada grafik tersebut terlihat bahwa pada nilai *Recall* dan akurasi, Data Latih A memiliki nilai yang lebih tinggi dibandingkan pada Data Latih B. Pada nilai akurasi, Data Latih A lebih tinggi 0.14 atau 14% dari pada nilai akurasi pada Data Latih B.

Seperti halnya pada pengujian menggunakan metode *Cross Validation*, untuk rata-rata nilai *Precision* pada pengujian ini dianggap bernilai nol karena pada beberapa klaster, yaitu *malwareCluster2*, *malwareCluster4*, *malwareCluster6*, *malwareCluster7* nilai *False Positive* dan *True Positive* bernilai 0. Dengan kata lain, pada *malware-malware* yang dikatakan berjenis *malware-malware* tersebut, tidak terdapat *malware* yang diklasifikasikan secara benar atau salah oleh WEKA berada pada klaster *malwareCluster2*, *malwareCluster4*, *malwareCluster6*, *malwareCluster7.*



Gambar 16. Perbandingan Rata-Rata Nilai Precision, Recall dan Akurasi dari Metode Precentage Split antara Data Latih A dan Data Latih B

Berdasarkan pada kedua pengujian (Pengujian menggunakan Metode *Cross Validation* dan Metode *Percentage Split*) Data Latih A memiliki nilai yang lebih baik. Hal ini dikarenakan pelabelan terhadap *malware* yang terdapat dalam data latih dilakukan dengan mengenali karakter *system call* dari setiap *malware* sesuai dengan penelitian [14] (dimana penelitian tersebut menggunakan *system call* sebagai fitur), dan pada proses pembadingan ini fitur yang digunakan adalah fitur-fitur yang sama pada penelitian Denar. Sedangkan pada penelitian Sendi, penamaan atau pelabelan *malware* dilakukan dengan menggunakan situs yang belum diketahui secara pasti parameter yang digunakan sebagai pengenalnya.

Menilik pada nilai rata-rata *Precision* pada pengujian pada Data Latih A, walaupun tidak didapatkan nilai yang tepat. Tetapi berdasarkan pada nilai *Precision* dari setiap kelas (klaster) yang dimiliki oleh setiap kelas (pada Gambar 4, Gambar 5, Gambar 7 dan Gambar 8), Data Latih A memiliki nilai *Precision* dan *Recall* yang lebih tinggi dibandingkan dengan Data Latih B. Sebagai contoh pada Data Latih A, pada pengujian menggunakan metode *Cross Validation*, setiap kelas memiliki nilai *Precision* diatas 0.5 bahkan terdapat beberapa kelas yang memiliki nilai *Precision* sama dengan 1.0. Sedangkan pada Data Latih B dengan metode pengujian yang sama memiliki nilai *Precision* diantara 0.4 sampai dengan 0.9.

IV. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan pada penelitian ini, data latih yang dihasilkan melalui proses klastering menggunakan frekuensi *system call malware* lebih akurat dibandingkan dengan data latih yang dihasilkan dengan menggunakan suatu situs penamaan *malware*. Walaupun penamaan jenis *malware* yang dilakukan tidak dapat spesifik tetapi pengelompokan *malware* lebih tepat. Sehingga proses klasifikasi berdasarkan frekuensi *system call malware* yang menggunakan data latih yang dihasilkan melalui proses klastering pada lebih direkomendasikan dari pada data latih yang menggunakan situs yang ada seperti *www.virustotal.com*.

System call pada suatu proses yang sedang berjalan dalam sistem operasi komputer memiliki banyak informasi yang bisa didapatkan. Sehingga selain menggunakan frekuensi dari *system call malware*, pengembangan sistem pendeteksi malware juga dapat menggunakan informasi lain yang tersimpan dalam *system call* suatu malware.

DAFTAR PUSTAKA

[1] K. Madasamy and M. Ramaswami, "Data Imbalance and Classifiers: Impact and Solutions," *International Journal of Computational Intelligence Research*, vol. 13, pp. 2267-2281, 2017.

[2] K. Millard and M. Richardson, "On the Importance of Training Data Sample Selection in Random Forest Image Classification: A Case Study in Peatland Ecosystem Mapping," *Remote Sens*, vol. 7, no. 7, pp. 8489-8515, 2015.

[3] M. Dimjašević, S. Atzeni, Z. Rakamarić and I. Ugrina, "Evaluation Of Android Malware Detection Based on System Calls," in *International Workshop on Security And Privacy Analytics. ACM*, Salt Lake, 2016.

[4] R. Canzanese, S. Mancoridis and M. Kam, "System Call-Based Detection of Malicious Processes," in *IEEE*, Vancouver, 2015.

[5] R. Canzanese, S. Mancoridis and M. Kam, "Run-Time Classification of Malicious Processes Using System Call Analysis," in *International Conference on Malicious and Unwanted Software (MALWARE)*, Fajardo, 2015.

[6] S. Malik, "Android System Call Analysis for Malicious Application Detection," *International Journal of Computer Sciences and Engineering*, vol. 5, no. 11, pp. 105-108, 2017.

[7] T. K. Barsiya, M. Gyanchandani and R. Wadhwani, "ANDROID MALWARE ANALYSIS : A SURVEY PAPER," *International Journal of Control, Automation, Communication and Systems (IJCACS)*, pp. 35-42, 2016.

[8] M. Christodorescu, S. Jha, S. A. Seshia, D. Song and R. E. Bryant, "Semantics-aware malware detection," in *IEEE Symposium on Security and Privacy*, Oakland, 2005.

[9] S. Pai, A Comparison of Clustering Techniques for Malware Analysis, San Jose: San Jose State University, 2015.

[10] U. Bayer, P. M. Comparetti, C. Hlauschek, C. Kruegel and E. Kirda, "Scalable, Behavior-Based Malware Clustering," *NDSS*, pp. 8-25, 2009.

[11] N. Idika and A. P. Mathur, A Survey of Malware Detection Techniques, West Lafayette: Purdue University, 2007.

[12] E. Gandotra, D. Bansal and S. Sofa, "Malware Analysis and Classification: A Survey," *Journal of Information Security*, pp. 56-64, 2014.

[13] S. Herlambang, S. Basuki, D. R. Akbi and Z. Sari, "Deteksi Malware Android Berdasarkan System Call Menggunakan Algoritma Support Vector Machine," in *SENTRA*, Malang, 2018.

[14] D. R. Akbi and A. R. Rosyadi, "Klastering Android Malware Berdasarkan Frekuensi System Call Menggunakan K-Means," in *SENTRA*, Malang, 2018.