e-ISSN 2962-6129
Seminar Nasional Mahasiswa Ilmu Komputer dan Aplikasinya (SENAMIKA)
Jakarta-Indonesia, 20 Agustus 2022

Klasifikasi Malware Berdasarkan Fitur API *Call* dan Android *Permission* Menggunakan *Radial Basis Function Network*

Bagas Aditya Wibisono [1], Didit Widiyanto [2], Noor Falih[3,]
Program Studi Informatika / Fakultas Ilmu Komputer
Universitas Pembangunan Nasional Veteran Jakarta
Jl. RS. Fatmawati , Pondok Labu, Jakarta Selatan, DKI Jakarta, 12450, Indonesia
bagasaw@upnvj.ac.id[1] , didit.widiyanto@upnvj.ac.id [2], falih@upnvj.ac.id [3]

Abstrak. *Malware* telah menjadi ancaman besar bagi pengguna teknologi saat ini. *Malware* atau *Malicious Software* adalah sebuah perangkat lunak bersifat intrusif yang dikembangkan oleh peretas dengan tujuan utama menginfeksi, menjelajah, mencuri, atau merusak suatu perangkat yang ditargetkan demi kepentingan peretas. Berbagai jenis perangkat dapat diinfeksi oleh malware, salah satunya adalah *smarphone*, di mana kasus *malware* terbanyak didominasi pada sistem operasi *Android*. Penelitian ini bertujuan dalam mengklasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering*.

Kata Kunci: Klasifikasi *malware*, *Radial Basis Function Network*, *K-Means Clustering*

1 Pendahuluan

1.1 Latar Belakang

Malware telah menjadi ancaman besar bagi pengguna teknologi saat ini. *Malware* atau *Malicious Software* adalah sebuah perangkat lunak bersifat intrusif yang dikembangkan oleh peretas dengan tujuan utama menginfeksi, menjelajah, mencuri, atau merusak suatu perangkat yang ditargetkan demi kepentingan peretas. Berbagai jenis perangkat dapat diinfeksi oleh malware, salah satunya adalah *smarphone*, di mana kasus *malware* terbanyak didominasi pada sistem operasi *Android*. Hal ini dikarenakan pengguna *smartphone* dengan sistem operasi *Android* lebih banyak dibandingkan sistem operasi lainnya yakni sebesar 72,11% berdasarkan data dari *StatCounter GlobalStats* [1].

Kasus *malware* berdasarkan laman AV Test, dilaporkan bahwa terdapat 1139,24 juta *malware* yang tersebar di berbagai perangkat pada tahun 2020 dibandingkan tahun 2019 sebesar 1001,52 juta [2]. Kemudian, kasus *malware* berdasarkan McAfee tentang *mobile threat report*, dilaporkan bahwa terdapat peningkatan signifikan *malware* bagi perangkat *mobile* di tahun 2020 pada Q4 menjadi 43 juta malware dari yang sebelumnya sebanyak 40 juta *malware* pada Q3 [3].

Berbagai metode pendeteksi *malware* telah dikembangkan sebagai antisipasi dalam menghadapi perkembangan *malware*, salah satunya dengan *Radial Basis Function Network* yang menunjukkan hasil yang baik pada kasus klasifikasi *malware* berdasarkan fitur *permissions* dengan *centroid* yang dipilih secara acak (Abdulrahman, dkk, 2021) [4]. Adapun penelitian yang berkaitan dengan *K-Means Clustering* sebagai metode pemilihan *centroid* dari *Radial Basis Function Network* seperti pada kasus prediksi penyakit ginjal kronik yang memberikan perbedaan antara hasil *Radial Basis Function Network* yang *centroid*-nya dipilih secara acak dengan yang *centroid*-nya dipilih menggunakan *K-Means Clustering* (Santosa, dkk, 2016) [5].

1.2 Rumusan Masalah

Rumusan masalah penelitian ini adalah:

a. Bagaimana klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering*?

b. Bagaimana perbedaan performa antara *Radial Basis Function Network* yang *centroid*-nya dipilih secara acak dengan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering* pada kasus klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions*?

1.3 Tujuan Penelitian

Tujuan penelitian ini adalah:

a. Mengetahui bagaimana klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering*.
b. Mengetahui perbedaan performa antara *Radial Basis Function Network* yang *centroid*-nya dipilih secara acak dengan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering* pada kasus klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions*.

1.4 Manfaat Penelitian

Manfaat penelitian ini adalah:

a. Memberikan gambaran tentang bagaimana klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering*.
b. Memberikan hasil perbedaan performa antara *Radial Basis Function Network* yang *centroid*-nya dipilih secara acak dengan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering* pada kasus klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions*.

2 Tinjauan Pustaka

2.1 API Call

Application Programming Interfaces (APIs) adalah antarmuka antar perangkat lunak dengan perangkat lunak lainnya yang memungkinkan perangkat lunak berinteraksi satu sama lain melalui media yang disebut API *call*. API *call* adalah proses permintaan yang dikirimkan aplikasi klien ke API, di mana data dari server diambil oleh API untuk dikirimkan kembali kepada klien (Fitzgerald, 2021) [6].

2.2 Android Permissions

Android *app permissions* adalah sebuah izin yang dimiliki oleh aplikasi Android untuk memperoleh kontrol dan akses perangkat seperti kamera, mikrofon, pesan pribadi, percakapan, foto, dan sebagainya. *App permissions* biasanya muncul saat sebuah aplikasi pertama kali dijalankan pada perangkat (Stegner, 2020) [7].

2.3 Malware

Malware atau *malicious software* adalah perangkat lunak bersifat intrusif yang dikembangkan oleh penjahat dunia maya dengan tujuan mencuri data, merusak, atau mengancurkan komputer beserta sistemnya. Contoh umum malware berupa virus, *worm*, *Trojan*, *spyware*, *adware*, dan *ransomware* (Tahir, 2018) [8].

2.4 Radial Basis Function Network

Radial Basis Function Network adalah salah satu bagian dari jaringan saraf tiruan yang menggunakan fungsi *gaussian* sebagai fungsi aktivasinya. Struktur *Radial Basis Function Network* dari tiga lapisan (*layer*). Pertama, *input layer* sebagai lapisan awal dengan tujuan membaca data masukkan untuk diproses pada lapisan selanjutnya. Kedua, *hidden layer*

sebagai lapisan yang melakukan proses terhadap data masukkan. Ketiga, *output layer* sebagai lapisan yang memberikan hasil keluaran.

Fungsi *gaussian* dari *Radial Basis Function Network* terdapat pada *hidden layer* serta memerlukan perhitungan *euclidean distance*. Adapun rumus fungsi *Gaussian* ditunjukkan pada persamaan (1) dan *euclidean distance* ditunjukkan pada persamaan (2) (Patmasari, 2017) [9].

$$\varphi_{ik} = \varphi\|x_{ij} - c_{kj}\| \quad \textbf{(1)}$$

$$d_{ik} = \sqrt{\sum_{j=1}^{m}\left(x_{ij} - c_{kj}\right)^2} \quad \textbf{(2)}$$

2.5 K-Means Clustering

K-Means Clustering merupakan algoritma pembelajaran *unsupervised* dalam *machine learning*. Algoritma ini bekerja dengan cara pembagian kumpulan data menjadi K *cluster* berbeda, setiap kumpulan data dengan karakteristik atau properti serupa dikelompokkan ke dalam satu cluster (Murti, 2017) [10]. Data dikelompokkan oleh *K-Means Clustering* dengan cara penentuan nilai K sebagai jumlah *cluster*, kemudian *centroid* awal dari *cluster* dipilih secara acak, lalu jarak antara tiap data dengan tiap *centroid cluster* dihitung dengan *euclidean dictance* pada persamaan (2).

3 Metodologi Penelitian

3.1 Identifikasi Permasalahan

Permasalahan *malware* diidentifikasi tentang bagaimana cara *malware* dapat diklasifikasi berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network*, serta penggunaan *K-Means Clustering* sebagai metode pemilihan *centroid* dari *Radial Basis Function Network*-nya.

3.2 Studi Literatur

Referensi dikumpulkan untuk dijadikan sebagai acuan dalam membantu proses penelitian, beberapa sumber referensi seperti jurnal, buku, karya ilmiah, *website*, *ebook*, serta penelitian terkait klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* dengan *K-Means Clustering* sebagai metode pemilihan *centroid*-nya.

3.3 Pengumpulan Data

Data dikumpulkan untuk digunakan selama penelitian berlangsung terkait klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* dengan *K-Means Clustering* sebagai metode pemilihan *centroid*-nya.

3.4 Praproses Data

Data dilakukan praproses terlebih dahulu sebelum digunakan pada proses klasifikasi. Praproses yang dilakukan seperti *label encoding* yang bertujuan agar data lebih mudah dibaca oleh model klasifikasi, *data balancing* yang bertujuan agar distribusi data lebih seimbang, serta *feature selection* yang bertujuan memilih fitur yang relevan dari seluruh data untuk digunakan selama proses klasifikasi.

3.5 Klasifikasi Malware

Klasifikasi *malware* dilakukan berdasarkan fitur Android API *call* dan *permissions*, menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih dengan *K-Means Clustering*. Proses klasifikasi dilakukan dengan beberapa percobaan *hyperparameter K-fold, learning rate, epoch*, dan jumlah *hidden unit*. Adapun rancangan proses penelitian dapat dilihat pada tabel 1 sampai tabel 4.

Tabel 1. Rancangan proses untuk hyperparameter K-fold

Hyperparameter			
K-fold	Learning Rate	Epoch	Hidden Unit
3	0.001	100	2
5			
7			
10			
15			

Tabel 2. Rancangan proses untuk *hyperparameter learning rate*

Hyperparameter			
K-Fold	Learning Rate	Epoch	Hidden Unit
Nilai terbaik dari *K-Fold*	0.001	100	2
	0.01		
	0.1		

Tabel 3. Rancangan proses untuk *hyperparameter epoch*

Hyperparameter			
K-Fold	Learning Rate	Epoch	Hidden Unit
Nilai terbaik dari *K-Fold*	Nilai terbaik dari *learning rate*	50	2
		100	
		150	
		200	

Tabel 4. Rancangan proses untuk *hyperparameter hidden unit*

Hyperparameter			
K-Fold	Learning Rate	Epoch	Hidden Unit
Nilai terbaik dari *K-Fold*	Nilai terbaik dari *learning rate*	Nilai terbaik dari *epoch*	2
			3
			5
			7

			10

3.6 Evaluasi Hasil

Hasil pengujian model *Radial Basis Function* dengan *K-Means Clustering* sebagai metode pemilihan *centroid*-nya pada kasus klasifikasi malware berdasarkan API *call* dan Android *permissions* dihitung nilai akurasi, *precision, recall*, dan F1 *score*-nya. Akurasi adalah pengukuran tentang seberapa dekat nilai prediksi dengan aktual. *Precision* adalah pengukuran tentang seberapa tepat informasi yang diminta dengan jawaban yang diberikan. *Recall* adalah pengukuran tingkat keberhasilan dalam menemukan kembali informasi. Sedangkan, F1 *score* adalah gabungan dari *precision* dan *recall*. Berikut adalah persamaan dari akurasi (3), *precision* (4), *recall* (5), dan *F1 score* (6).

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad \textbf{(3)}$$

$$Precision = \frac{TP}{TP + FP} \quad \textbf{(4)}$$

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(5)}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad \textbf{(6)}$$

Perhitungan tersebut dilakukan berdasarkan *confusion matrix* yang terdiri dari *True Positive* (TP) berupa hasil kelas positif yang diprediksi dengan benar sebagai kelas positif, *True Negative* (TN) berupa hasil kelas negatif yang diprediksi dengan benar sebagai kelas negatif, *False Positive* (FP) berupa hasil dari kelas negatif yang salah diprediksi sebagai kelas positif, dan *False Negative* (FN) kelas positif yang salah diprediksi sebagai kelas negatif, ditunjukkan pada tabel 1.

Tabel 5. Confussion Matrix

		Aktual	
		Malware (positif)	*Benign* (negatif)
Prediksi	*Malware* (positif)	*True Positive* (TP)	*False Positive* (FN)
	Benign (negatif)	*False Negative* (FN)	*True Negative* (TN)

3.7 Kesimpulan

Hasil akhir penelitian disimpulkan terkait pengaruh penggunaan *K-Means Clustering* sebagai metode pemilihan *centroid* dalam klasifikasi *malware* berdasarkan Android API *call* dan *permissions* menggunakan *Radial Basis Function Network*, untuk dijadikan sebagai pertimbangan pada penelitian selanjutnya.

4 Hasil dan Pembahasan

4.1 Pengumpulan Data

Data yang digunakan penelitian ini adalah *malgenome-215-dataset*, diunduh dari repository *figshare* [11]. Dataset berupa file CSV (*Comma Separated Value*) terdiri dari 3799 sampel aplikasi anonim dengan 215 fitur dan 1 *class*. Fitur tersebut terdiri dari 73 fitur API *call signature*, 113 fitur *Manifest Permission*, 23 fitur *Intent*, dan 6 fitur *Commands signature*.

Tiap fitur dari sampel aplikasi diwakili dengan angka 0 sebagai fitur non-aktif dan 1 sebagai fitur aktif. Serta, *class* terdiri dari *malware* (S) dan *benign* (B).

4.2 Praproses Data

Praproses data dilakukan terhadap *malgenome-215-dataset* dengan tujuan agar proses klasifikasi lebih mudah serta menghasilkan keluaran produktif dan informatif. Praproses pertama yakni melakukan *label encoding* untuk mengubah representasi *class* dari *string* ke dalam angka agar memudahkan model dalam membaca data. Kemudian, dilakukan *data balancing* dengan *random undersampling* agar distribusi antara *class malware* dan *benign* lebih seimbang, dari perbandingan 2539 *malware* dan 1260 *benign* menjadi 1260 *malware* dan 1260 *benign*. Terakhir, dilakukan *feature selection* dengan memilih 100 fitur relevan dari keseluruhan data menggunakan *mutual information*.

4.3 Pemilihan Centroid dengan K-Means

Centroid dari model *Radial Basis Function* dipilih dengan *K-Means Clustering*. *Centroid* bertujuan untuk menghitung jarak *euclidean* pada *Radial Basis Function* untuk digunakan pada fungsi *Gaussian*-nya, jumlah *centroid* menentukan jumlah *hidden unit* pada *hidden layer*. Pemilihan *centroid* dengan *K-Means Clustering* dilakukan dengan pembagian data *input* ke dalam kelompok (*cluster*) K yang telah ditentukan dengan perhitungan *euclidean distance*

4.4 Hasil Pengujian Model Radial Basis Function Network

Tabel 6. Pengujian model dengan fokus *hyperparameter* K-fold

K-fold	Learning Rate	Epoch	Unit	Akurasi	Loss	Waktu (detik)
3	0.001	100	2	94,52%	4,81%	29,099
5	0.001	100	2	94,25%	4,70%	30,635
7	0.001	100	2	93,33%	5,16%	31,866
10	0.001	100	2	96,03%	4,26%	32,014
15	0.001	100	2	93,45%	4,62%	34,605

Tabel 7. Pengujian model dengan fokus *hyperparameter learning rate*

K-fold	Learning Rate	Epoch	Unit	Akurasi	Loss	Waktu (detik)
10	0.001	100	2	96,03%	4,26%	32,014
10	0.01	100	2	95,63%	3,52%	40,591
10	0.1	100	2	42,86%	50%	32,662

Tabel 8. Pengujian model dengan fokus *hyperparameter epoch* dan *learning rate* 0.001

K-fold	Learning Rate	Epoch	Unit	Akurasi	Loss	Waktu (detik)
10	0.001	50	2	91,27%	8,55%	16,997
10	0.001	100	2	96,03%	4,26%	32,014
10	0.001	150	2	96,43%	3,65%	47,213

10	0.001	200	2	97,22%	3,37%	63,498

Tabel 9. Pengujian model dengan fokus *hyperparameter epoch* dan *learning rate* 0.01

K-fold	Learning Rate	Epoch	Unit	Akurasi	Loss	Waktu (detik)
10	0.01	50	2	97,22%	3,80%	18,323
10	0.01	100	2	95,63%	3,52%	32,150
10	0.01	150	2	96,43%	3,32%	47,751
10	0.01	200	2	97,22%	3,93%	62,391

Tabel 10. Pengujian model dengan fokus *hyperparameter hidden unit*

K-fold	Learning Rate	Epoch	Unit	Akurasi	Loss	Waktu (detik)
10	0.001	200	2	97,22%	3,37%	63,498
10	0.001	200	3	98,02%	2,32%	65,133
10	0.001	200	5	97,62%	2,02%	65,567
10	0.001	200	7	98,02%	1,96%	66,481
10	0.001	200	10	98,41%	1,94%	68,683

4.5 Evaluasi Hasil

Hasil pengujian model di evaluasi dengan *confusion matrix* yang ditunjukkan pada tabel untuk menghitung akurasi, *precision*, *recall*, dan F1 *score*. Perhitungan dilakukan dari jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang dihasilkan.

Tabel 11. Confussion Matrix hasil percobaan

		Nilai aktual	
		Malware (positif)	Benign (negatif)
Nilai prediksi	Malware (positif)	141 (TP)	1 (FP)
	Benign (negatif)	3 (FN)	107 (TN)

Confusion matrix kemudian digunakan pada perhitungan akurasi, *precision*, *recall*, dan F1 *score*. Berikut adalah hasil perhitungannya.

$$Akurasi = \frac{141 + 107}{141 + 107 + 1 + 3} = 0{,}9841$$

$$Precision = \frac{141}{141 + 1} = 0{,}993$$

$$Recall = \frac{141}{141 + 3} = 0{,}9792$$

$$F1\ score = 2 \times \frac{0{,}993 \times 0{,}9792}{0{,}993 + 0{,}9792} = 0{,}986$$

5 Kesimpulan dan Saran

Klasifikasi *malware* berdasarkan fitur API *call* dan Android *permissions* menggunakan *Radial Basis Function Network* dengan *K-Means Clustering* sebagai metode pemilihan *centroid*-nya menunjukkan hasil yang baik, akurasi yang diperoleh sebesar 98,41%, *precision* 99,3%, *recall* 97,92%, dan F1 *score* 98,6%, dengan waktu komputasi 68,683 detik. Hasil tersebut diperoleh dari percobaan dengan *hyperparameter K-fold* = 10, *learning rate* = 0.001, *epoch* = 200, dan *hidden unit* = 10. Hasil ini lebih baik dari klasifikasi *malware* menggunakan *Radial Basis Function Network* yang *centroid*-nya dipilih secara acak [12] yang akurasinya 97,20%.

Dari kesimpulan di atas, saran untuk penelitian selanjutnya adalah sebagai berikut:

a. Melakukan percobaan untuk melihat kemungkinan hasil yang berbeda dari model *Radial Basis Function Network* sebagai pengklasifikasi *malware* dengan menggunakan *hyperparameter* atau menggabungkannya dengan metode lainnya.
b. Percobaan dapat dilakukan dengan dataset *malware* lainnya dengan fitur yang berbeda atau jumlah data yang lebih banyak terhadap model *Radial Basis Function Network* dalam kasus klasifikasi *malware*.

Referensi

[1] StatCounter. (2022). *Mobile Operating System Market Worldwide*. StatCounter GlobalStats: https://gs.statcounter.com/os-market-share/mobile/worldwide/2020

[2] AV Test Organization. (2022). *Statistics: Malware*. AV Test: https://www.av-test.org/en/statistics/malware

[3] McAfee. (2020). *2020 mobile threat report*. McAfee: https://www.mcafee.com/en-us/consumer-support/2020-mobile-threat-report.html

[4] Abdulrahman, A., Hashem, K., Adnan, G., & Ali, W. (2021). Intelligent Android Malware Detection Using Radial Basis Function Networks and Permissions Features. *IJCSNS International Journal of Computer Science and Network Security*.

[5] Santosa, S., Widjanarko, A., & Supriyanto, C. (2016). Model Prediksi Penyakit Ginjal Kronik Menggunakan Radial Basis Function. *Jurnal Pseudocode Politeknik Negeri Semarang*.

[6] Fitzgerald, A. (2021, September 20). *API Calls: What They Are & How to Make Them in 5 Easy Steps*. HubSpot: https://blog.hubspot.com/website/api-calls

[7] Stegner, B. (2020, August 27). *How Do Android App Permissions Work? What You Need to Know*. Make Of Use: https://www.makeuseof.com/tag/what-are-android-permissions-why-should-you-care/

[8] Tahir, R. (2018, March 8). A Study on Malware and Malware Detection Techniques. *I.J Education and Management Engineering, 2*, 20-30.

[9] Patmasari, A. (2017). Penerapan Metode Jaringan Syaraf Tiruan Radial Basis Function Untuk Klasifikasi Status Gizi Balita. Pekanbaru, Riau: Universitas Islam Negeri Sultan Syarif Kasim Riau.

[10] Murti, M. A. (2017). Penerapan Metode K-Means Clustering Untuk Mengelompokkan Potensi Produksi Buah-buahan di Provinsi Daerah Istimewa Yogyakarta. Yogyakarta: Universitas Sanata Dharma.

[11] Yerima, S. (2018). *Android Malware Dataset for Machine Learning 1*. Figshare: https://figshare.com/articles/dataset/Android_malware_dataset_for_machine_learning_1/5854590/1