JOURNAL OF COMPUTER SCIENCE AND INFORMATICS ENGINEERING
V VOL. 05, NO. 2, APRIL 2026: 147-156
Doi: https://doi.org/10.55537/cosie.v5i2.1520

e-ISSN 2827-8356
p-ISSN 2827-9999

Klasifikasi Tingkat Bahaya Malware Android Menggunakan Reverse Engineering Dengan Metode SAW dan Integrasi VirusTotal

Implementation of Reverse Engineering Techniques for Malware Risk Level Categorization Using Simple Additive Weighting

Muhammad Firdaus Aldiansyah*[1], Khaerul Ma'mur[2]
[1,2]Prodi Teknik Informatika, Universitas Pamulang, Jl. Raya Puspitek, Tangerang Selatan, Banten, 15310, Indonesia
[1]tomsbreak@gmail.com, [2]dosen00844@unpam.ac.id

Received: Januari 12, 2026 | Revised: Februari 02, 2026 | Accepted: March 09, 2026

Abstrak

Perkembangan pesat penggunaan perangkat Android diiringi dengan meningkatnya ancaman *malware*, khususnya *spyware* yang dapat mencuri informasi sensitif pengguna. Penelitian ini bertujuan untuk mengklasifikasikan tingkat bahaya *spyware* Android dengan memanfaatkan teknik *reverse engineering*, integrasi data dari *VirusTotal API*, dan metode *Simple Additive Weighting (SAW)*. Proses *reverse engineering* dilakukan untuk mengekstrak atribut internal *malware* seperti jumlah *permission* dan kategori *permission* berbahaya dari file APK. Selain itu, atribut eksternal diperoleh melalui integrasi *VirusTotal API* yang meliputi jumlah *engine* antivirus yang mendeteksi serta kategori jenis ancaman. Seluruh atribut tersebut kemudian diolah menggunakan metode SAW untuk menghasilkan nilai preferensi yang digunakan dalam pengelompokan tingkat bahaya *spyware* ke dalam kategori rendah, sedang, dan tinggi. Sampel yang digunakan dalam penelitian ini adalah *malware AhMyth* sebagai representasi *spyware* Android. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu memberikan klasifikasi tingkat bahaya *spyware* secara sistematis serta menyajikan informasi risiko yang lebih informatif bagi pengguna. Penelitian ini diharapkan dapat membantu meningkatkan pemahaman pengguna terhadap ancaman *spyware* serta mendukung pengembangan sistem deteksi *malware* yang lebih informatif.

Kata kunci: *Spyware*, *Reverse Engineering*, SAW, Klasifikasi *Malware*, Keamanan Siber

Abstract

The rapid growth of Android devices has been accompanied by an increasing number of malware threats, particularly spyware that can steal sensitive user information. This study aims to classify the risk level of Android spyware using reverse engineering techniques, VirusTotal API integration, and the Simple Additive Weighting (SAW) method. Reverse engineering is performed to extract internal malware attributes such as the number of permissions and the category of dangerous permissions from APK files. In addition, external attributes are obtained through VirusTotal API integration, including the number of antivirus engines detecting the malware and the threat category. These attributes are then processed using the SAW method to generate preference values used for classifying spyware risk levels into low, medium, and high categories. The malware sample used in this study is AhMyth, which represents Android spyware. The results indicate that the developed system is capable of providing systematic spyware risk classification and presenting more informative risk information to users. This research is expected to improve user awareness of spyware threats and support the development of more informative malware detection systems.

JOURNAL OF COMPUTER SCIENCE AND INFORMATICS ENGINEERING
V VOL. 05, NO. 2, APRIL 2026: 147-156
Doi: https://doi.org/10.55537/cosie.v5i2.1520
e-ISSN 2827-8356
p-ISSN 2827-9999

Keywords: Spyware, Reverse Engineering, SAW, Malware Classification, Cybersecurity

1. PENDAHULUAN

Jumlah perangkat Android yang meningkat membuat platform ini menjadi sasaran utama serangan malware. Berbagai jenis malware, termasuk trojan, spyware, dan backdoor, memiliki kemampuan untuk mengeksploitasi izin aplikasi dan berkomunikasi secara tersembunyi dengan server eksternal, yang memungkinkan pencurian data pengguna yang sensitif[1]. Dibutuhkan metode yang tidak hanya dapat mendeteksi *malware* tetapi juga dapat mengukur tingkat bahayanya secara kuantitatif, karena ancaman semakin kompleks. Sebagian besar penelitian terbaru tentang klasifikasi malware Android menggunakan metode *machine learning* seperti *XGBoost, Random Forest, Supervised, dan* Variasi Model Algoritma [2, 3, 4, 5] untuk menemukan pola berbahaya dalam kumpulan data yang sangat besar. Meskipun metode ini menunjukkan akurasi yang tinggi, itu membutuhkan proses pelatihan model, bergantung pada dataset yang besar dan terlabel dengan baik, dan memiliki beberapa keterbatasan dalam hal interpretabilitas hasil klasifikasi. Transparansi dan tingkat risiko yang jelas sangat penting untuk keamanan pengguna umum. Di sisi lain, penelitian berbasis *reverse engineering* [6, 7, 8, 9] umumnya berfokus pada analisis teknis perilaku *malware*, seperti identifikasi *permission* berbahaya, komunikasi *Command and Control*, serta teknik *persistence*. Namun, pendekatan tersebut masih bersifat deskriptif dan belum mengintegrasikan hasil analisis ke dalam model kuantitatif yang mampu menghasilkan skor tingkat bahaya yang terukur. *Metode Multi-Criteria Decision Making (MCDM)*, khususnya *Simple Additive Weighting (SAW)*, untuk mengukur tingkat bahaya *malware Android* yang didasarkan pada hasil *reverse engineering* dan integrasi data *VirusTotal* masih terbatas. Padahal, pendekatan ini memungkinkan penggabungan berbagai indikator risiko ke dalam satu skor komposit yang mudah direplikasi dan transparan. Oleh karena itu, penelitian ini mengusulkan model klasifikasi tingkat bahaya malware Android menggunakan metode SAW yang mengintegrasikan hasil *reverse engineering* dan data deteksi *multi-engine* dari *VirusTotal API*. Penting bagi penelitian ini adalah desain model penilaian risiko berbasis MCDM, yang secara sistematis menghasilkan skor preferensi untuk menunjukkan tingkat bahaya *malware*. Dibandingkan dengan metode deskriptif konvensional, metode ini tidak hanya menemukan keberadaan *malware* tetapi juga memberikan pengukuran tingkat risiko yang lebih interpretatif.

2. METODE PENELITIAN

Metode penelitian dilakukan untuk mencari sesuatu secara sistematis dengan menggunakan metode ilmiah serta sumber yang berlaku. Penelitian ini menggunakan metode studi literatur untuk mengumpulkan data dan mengolah informasi. Tujuan dari metode ini adalah untuk melihat berbagai sumber tertulis, seperti jurnal ilmiah, buku referensi, dan sumber online, yang terkait dengan topik penelitian.

2.1 Pengumpulan Data

Penelitian ini menggunakan sampel pada penelitian sebelumnya yaitu *Ahmyth.Apk* dan juga untuk sampel lainnya didapatkan dari situs MalwareBazzaar (bazaar.abuse.ch) dan ApkPure (apkpure.com).

2.2 Teknik Reverse Engineering

Reverse engineering adalah suatu proses analisis yang dilakukan dengan membongkar sistem, baik perangkat lunak maupun perangkat keras, untuk mendapatkan pemahaman tentang mekanisme kerja internalnya. Tujuan utama dari *reverse engineering* adalah mengidentifikasi serta mengkaji bagaimana komponen-komponen penyusun sistem berinteraksi satu sama lain saat menjalankan fungsinya[10]. Pada analisis *malware* Android, *reverse engineering* umumnya dilakukan melalui dua pendekatan, yaitu analisis statis dan analisis dinamis[11]. Analisis statis

e-ISSN 2827-8356
p-ISSN 2827-9999

dilakukan tanpa mengeksekusi aplikasi, dengan cara membongkar struktur file APK untuk mengekstraksi komponen seperti *AndroidManifest.xml*, daftar *permission*, serta *bytecode*. Sementara itu, analisis dinamis dilakukan dengan menjalankan aplikasi dalam lingkungan terkontrol untuk mengamati perilaku runtime, seperti komunikasi jaringan atau aktivitas sistem yang mencurigakan. Dalam penelitian ini, pendekatan yang digunakan adalah analisis statis. Proses *reverse engineering* dilakukan menggunakan *Apktool* dan *Dex2Jar* untuk mengekstraksi struktur aplikasi serta mengidentifikasi *permission* dan komponen yang berpotensi berbahaya. Informasi tersebut digunakan sebagai indikator internal dalam proses perhitungan tingkat bahaya menggunakan metode SAW. Selain itu, data eksternal berupa jumlah deteksi dan kategori ancaman diperoleh melalui integrasi *VirusTotal API*.

Pendekatan analisis statis dipilih karena lebih aman dan tidak memerlukan eksekusi *malware* dalam lingkungan terisolasi. Namun demikian, metode ini memiliki keterbatasan. Analisis statis tidak mampu mendeteksi perilaku runtime seperti analisis dinamis[12], komunikasi *Command and Control* yang hanya aktif saat aplikasi dijalankan, serta teknik *obfuscation* lanjutan yang dapat menyembunyikan string atau fungsi tertentu. Oleh karena itu, model penilaian risiko dalam penelitian ini berfokus pada indikator struktural dan metadata aplikasi yang dapat diidentifikasi secara statis[6].

2.3 Metode Simple Additive Weighting

Salah satu metode pengambilan keputusan multikriteria adalah metode SAW, juga dikenal sebagai metode penjumlahan terbobot [13]. Prinsip utama metode SAW adalah menghitung nilai preferensi untuk setiap opsi dengan menambahkan hasil dari pengalian antara bobot dan nilai kinerja setiap atribut . Dalam penerapan metode ini, matriks keputusan memerlukan proses normalisasi agar semua nilai atribut berada pada skala yang sebanding. Tujuan dari proses normalisasi ini adalah untuk memungkinkan perbandingan yang adil antar alternatif yang didasarkan pada selur . Berikut ini adalah alur perhitungan metode SAW yang digunakan.

1. Menentukan kriteria (Cj) dan bobot (Wj).
2. Membentuk nilai alternatif (Ai).
3. Membentuk matriks keputusan (Xij).
4. Menghitung nilai normalisasi (Rij).
5. Menghitung nilai preferensi (Vi).
6. Mengklasifikasi tingkat bahaya.

Dalam penelitian ini, metode SAW digunakan sebagai model penilaian tingkat risiko dengan menggabungkan beberapa indikator bahaya menjadi satu skor komposit.

1. Kriteria dan subkriteria, setiap faktor penilaian disebut kriteria (*Cj*) dan memiliki tingkat kepentingan yang ditunjukkan sebagai bobot (*Wj*). Ketika suatu kriteria memiliki nilai kategorikal, kategori tersebut digambarkan sebagai subkriteria dan dipetakan ke nilai numerik agar dapat diproses secara matematis.
2. Konversi kategori ke nilai numerik, Dalam beberapa situasi, satu objek dapat memiliki lebih dari satu kategori pada saat yang sama. Oleh karena itu, untuk mendapatkan nilai representatif, digunakan metode rata-rata berbobot:

$$Rkategori = \frac{\sum_{i=1}^{n}(fi\ x\ wi)}{\sum_{i=1}^{n} fi}$$

Keterangan:
Rkategori = nilai kriteria hasil rata-rata kategori
fi = jumlah deteksi pada kategori ke-i
wi = bobot risiko kategori ke-i (0-1)
n = jumlah kategori yang terdeteksi

e-ISSN 2827-8356
p-ISSN 2827-9999

3. Normalisasi, karena seluruh atribut merepresentasikan tingkat risiko, maka normalisasi menggunakan skala maksimum:

$$Rij = \frac{Xij}{\mathrm{Xj(max)}}$$

Keterangan :
R_{ij} : nilai atribut ke-j yang telah dinormalisasi pada alternatif ke-i
X_{ij} : nilai asli atribut ke-j pada alternatif ke-i
X_j(max) : nilai maksimum pada kriteria ke-j

4. Nilai preferensi, nilai tingkat bahaya dihitung dengan rumus:

$$Vi = \sum_{j=1}^{n} (Wj \; x \; Rij)$$

Keterangan :
V_i : nilai preferensi / skor tingkat bahaya alternatif ke-i
W_j : bobot kepentingan pada kriteria ke-j
R_{ij} : nilai atribut yang telah dinormalisasi pada kriteria ke-j
n : jumlah kriteria
Semakin besar nilai Vi, semakin tinggi tingkat bahaya objek yang dianalisis.

3. HASIL DAN PEMBAHASAN

3.1 Perhitungan Metode Simple Additive Weighting

Contoh perhitungan untuk file *malware* Bernama *ahmyth.apk* yang dianalisis menggunakan hasil *reverse engineering* integrasi *Virustotal API* diberikan di sini untuk memperjelas penerapan metode SAW dalam menentukan tingkat bahaya *malware*.

Langkah Pertama: Menentukan kriteria (*Cj*) dan Bobot (*Wj*)

Langkah pertama adalah menentukan kriteria dan bobot yang didapat dari hasil *VirusTotal API* dan penelitian sebelumnya [6].

Tabel 1 Kriteria dan bobot

Kriteria	Keterangan	Sumber	Bobot
C1	Jumlah mesin terdeteksi	Virustotal	0.15
C2	Kategori ancaman terdeteksi	Virustotal	0.40
C3	Jumlah izin berbahaya	[6]	0.15
C4	Kategori izin berbahaya	[6]	0.30

Setiap jenis ancaman berdampak berbeda terhadap sistem dan privasi pengguna, menurut penelitian yang dilakukan oleh [14]. Nilai tingkat bahaya ditetapkan secara konseptual berdasarkan sifat aktivitas ancaman terhadap sistem. Nilai diatur dalam skala dari 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan tingkat risiko yang lebih besar terhadap pengguna. Tabel berikut menunjukkan hasil penelitian:

e-ISSN 2827-8356
p-ISSN 2827-9999

Tabel 2 Nilai kategori ancaman

Kategori Ancaman (C2)	Nilai	Dasar Penetapan
Trojan	0.9	[14]
Backdoor	0.85	
Spy	0.8	
Stealer	0.75	
Dropper	0.7	
Downloader	0.7	
Generic	0.6	
Adware	0.5	
Riskware	0.4	

Sebagaimana dijelaskan dalam dokumentasi resmi Android, setiap jenis izin aplikasi memiliki tingkat dampak terhadap sistem dan privasi pengguna yang berbeda[15]. Oleh karena itu, tingkat risiko izin ditetapkan secara konseptual berdasarkan sifat akses dan kemungkinan penyalahgunaan sumber daya sistem. Nilai risiko dinormalisasi dari 0 hingga 1, dengan nilai yang lebih tinggi menunjukkan risiko yang lebih besar terhadap pengguna. Tabel berikut menunjukkan hasil pengelompokan dan pembobotan kategori izin aplikasi berdasarkan tingkat perlindungan Android.

Tabel 3 Nilai kategori permission

Jenis Permission (C3)	Nilai	Dasar Penetapan
Signature-Privileged	0.9	[15]
Dangerous	0.8	
Signature	0.6	
Normal	0	

Langkah Kedua: Membentuk Nilai Alternatif

Langkah kedua adalah membentuk nilai alternatif dari sampel *Ahmyth.apk* dengan mengambil data menggunakan integrasi *Virustotal* dan juga hasil *reverse engineering AndroidManifest*.

Tabel 4 Nilai alternatif

Kriteria	Data
C1	Mesin antivirus mendeteksi berbahaya sebanyak 31 dari 65 total mesin antivirus
C2	Dropper 1, Spy+Trojan 3, Trojan 5, Backdoor 5, Spy 6, Generic 11. Semuanya berjumlah 31 (sesuai dengan C1)
C3	Jumlah *permission* berbahaya yang terdeteksi sebanyak 10 dari 15 total *permission*
C4	Permission yang dianggap *Dangerous* berjumlah 10 (sesuai dengan C3)

Langkah Ketiga: Pembentukan Matriks Keputusan (X)

Langkah ketiga adalah pembentukan matriks keputusan berdasarkan data yang ada pada nilai alternatif.

e-ISSN 2827-8356
p-ISSN 2827-9999

Tabel 5Matriks keputusan

Kriteria	Nilai (*Xij*)
C1	31
C2	(1 x 0.70) + (3 x 0.90) + (5 x 0.85) + (5 x 0.90) + (6 x 0.80) + (11 x 0.60) = 23.55
C3	10
C4	(10 x 0.8) = 8
X	[31, 23.55, 10, 8]

Langkah Keempat: Menghitung Nilai Normalisasi

Langkah keempat adalah menormalisasi nilai setiap kriteria. Dengan cara membagi nilai kriteria (*xij*) dengan nilai maksimum (*xj max*). Sesuai dengan rumus di bawah ini:

$$Rij = \frac{Xij}{\mathrm{Xj(max)}}$$

Tabel 6. Nilai Normalisasi

Kriteria (*Cj*)	Nilai (*xij*)	Nilai Maksimum (*xj max*)	Nilai Normalisasi (*Rij*)
C1	31	65	0.476
C2	23.55	31 (Nilai C1)	0.759
C3	10	15	0.666
C4	8	10 (Nilai C3)	0.8

Langkah Kelima: Menghitung Nilai Preferensi

Langkah kelima adalah menghitung nilai preferensi. Dengan cara mengalikan masing-masing nilai bobot dengan hasil normalisasi tiap kriteria. Sesuai dengan rumus di bawah ini:

$$Vi = \sum_{j=1}^{n} (Wj \; x \; Rij)$$

Tabel 7. Nilai Preferensi

Kriteria	Bobot (*Wj*)	Nilai Normalisasi (*Rij*)	Nilai Preferensi (*Vi*)
C1	0.15	0.476	0.0715
C2	0.40	0.759	0.303
C3	0.15	0.666	0.0999
C4	0.30	0.8	0.24
		V	0.715

Langkah Keenam: Klasifikasi Tingkat Bahaya

Langkah keenam adalah proses klasifikasi tingkat bahaya berdasarkan hasil perhitungan yang sudah dilakukan pada langkah sebelumnya.

Tabel 8. Klasifikasi Tingkat Bahaya

Rentang Nilai	Tingkat Bahaya
0.00 - 0.30	Rendah
0.31 – 0.60	Sedang
0.61 – 1.00	Tinggi

Berdasarkan hasil perhitungan metode SAW didapat nilai 0.715 dan masuk kategori Tinggi yang berada pada rentang nilai 0.61 – 1.00.

e-ISSN 2827-8356
p-ISSN 2827-9999

3.2 *Hasil Perhitungan Sampel Malware dan Clean*

Untuk mengevaluasi konsistensi model klasifikasi tingkat bahaya berbasis SAW, pengujian dilakukan terhadap sejumlah sampel APK yang terdiri dari aplikasi bersih (sumber: apkpure.com) dan *malware* (sumber: bazaar.abuse.ch). Tabel berikut menyajikan ringkasan hasil perhitungan skor SAW untuk masing-masing sampel bersama dengan kategori tingkat bahaya yang dihasilkan.

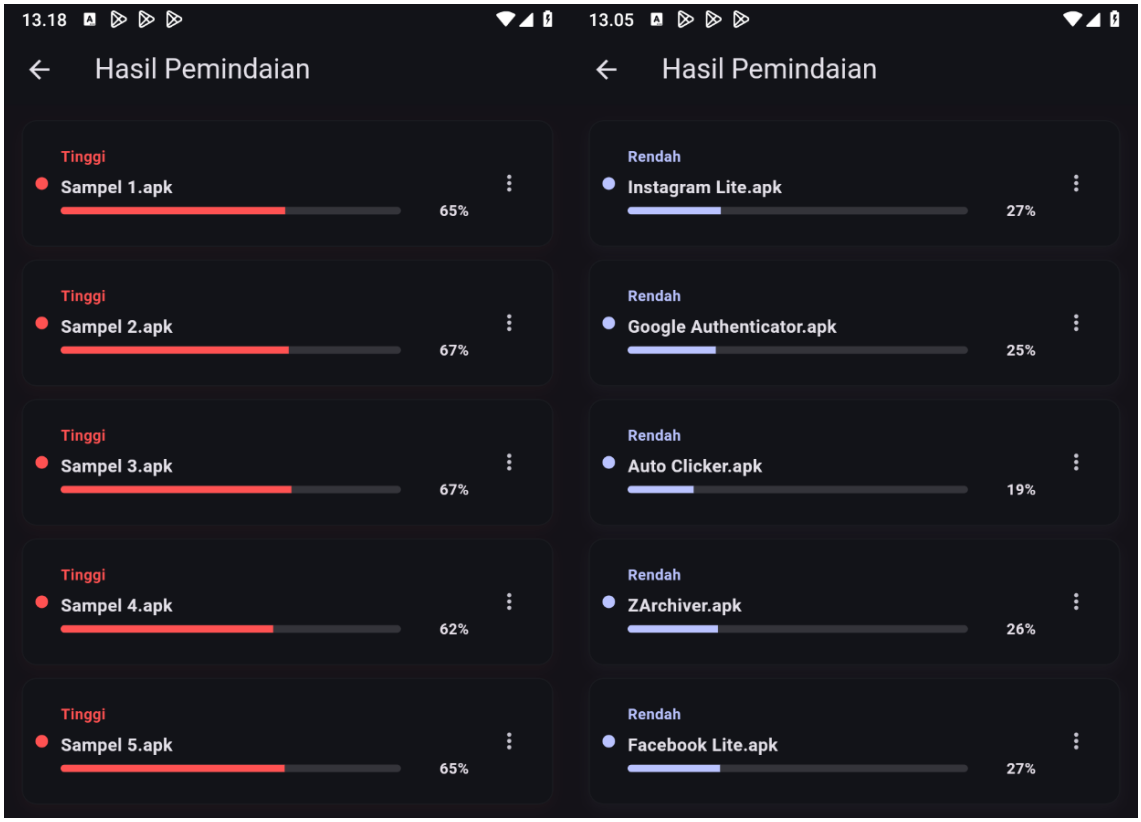
Tabel 9 Hasil Perhitungan Sampel malware dan clean

No	Sampel	Tipe	C1	C2	C3	C4	Skor SAW	Tingkat Bahaya
1	Sampel 1	*Malware*	$\frac{23}{68}$	$\frac{17.5}{23}$	$\frac{15}{31}$	$\frac{11.6}{15}$	0.65	Tinggi
2	Sampel 2	*Malware*	$\frac{15}{67}$	$\frac{11.4}{67}$	$\frac{3}{4}$	$\frac{2.2}{3}$	0.67	Tinggi
3	Sampel 3	*Malware*	$\frac{26}{67}$	$\frac{20.5}{26}$	$\frac{15}{31}$	$\frac{11.6}{15}$	0.67	Tinggi
4	Sampel 4	*Malware*	$\frac{15}{63}$	$\frac{11.7}{15}$	$\frac{11}{31}$	$\frac{8.2}{11}$	0.62	Tinggi
5	Sampel 5	*Malware*	$\frac{18}{67}$	$\frac{14.1}{18}$	$\frac{15}{31}$	$\frac{11.6}{15}$	0.65	Tinggi
6	Instagram Lite	*Clean*	$\frac{0}{67}$	0	$\frac{19}{61}$	$\frac{14.4}{19}$	0.27	Rendah
7	Google Authenticator	*Clean*	$\frac{0}{67}$	0	$\frac{1}{8}$	$\frac{0.8}{1}$	0.25	Rendah
8	Auto Clicker	*Clean*	$\frac{0}{68}$	0	$\frac{1}{11}$	$\frac{1}{0.6}$	0.19	Rendah
9	ZArchiver	*Clean*	$\frac{0}{68}$	0	$\frac{3}{10}$	$\frac{2.2}{3}$	0.26	Rendah
10	Facebook Lite	*Clean*	$\frac{0}{68}$	0	$\frac{21}{65}$	$\frac{15.6}{21}$	0.27	Rendah

Beberapa aplikasi bersih tetap memperoleh skor antara 0,19 dan 0,27 meskipun *engine* antivirus tidak menemukannya, meskipun model terus mengklasifikasikan sampel *malware* dan aplikasi bersih secara kategorikal. Hal ini disebabkan oleh kriteria izin dalam perhitungan SAW, yang memungkinkan aplikasi bersih tetap memiliki sejumlah izin dengan kategori izin *dangerous*. Oleh karena itu, bukannya menunjukkan keamanan absolut suatu aplikasi, model ini

e-ISSN 2827-8356
p-ISSN 2827-9999

menunjukkan tingkat risiko berdasarkan indikator struktural. Berikut ini tampilan aplikasi hasil pemindaian.



Gambar 1 Hasil pemindaian malware dan clean

Studi sebelumnya menemukan *malware* sebagai *spyware* dengan kemampuan komunikasi *Command and Control* dan eksploitasi izin sistem. Hasil klasifikasi sampel *AhMyth.apk* juga sejalan. Penelitian ini mencakup analisis teknis deskriptif dengan menggunakan *reverse engineering*. Namun, untuk menghasilkan tingkat bahaya yang sistematis dan dapat diukur, penelitian ini memasukkan indikator hasil analisis ke dalam model kuantitatif berbasis SAW. Untuk mengecek kemampuan sistem secara berdasarkan angka, dilakukan pengujian keakuratan terhadap 10 sampel yang terdiri dari 5 jenis *malware* dan 5 aplikasi yang *clean*. Semua sampel *malware* dikelompokkan dalam kategori Tinggi dan semua aplikasi yang bersih dikelompokkan dalam kategori Rendah, sehingga menghasilkan nilai akurasi 100% dengan tingkat *False Positive Rate (FPR)* sebesar 0%. Hasil tersebut menunjukkan bahwa model bisa membedakan antara sampel *malware* dan aplikasi yang bersih secara terus-menerus di dataset yang digunakan. Namun, hasil ini masih terbatas karena jumlah sampel yang digunakan relatif kecil dan diperlukan pengujian tambahan menggunakan dataset yang lebih besar agar dapat menguji seberapa baik model tersebut bekerja pada kondisi yang berbeda.

4. KESIMPULAN

Hasil penelitian menunjukkan bahwa metode *Simple Additive Weighting (SAW)* dapat digunakan untuk menentukan tingkat bahaya *malware* Android. Empat kriteria utama yaitu jumlah mesin terdeteksi, kategori ancaman, jumlah izin berbahaya, dan kategori izin berbahaya yang digunakan dalam pembuatan model, yang diproses melalui proses normalisasi dan

e-ISSN 2827-8356
p-ISSN 2827-9999

pembobotan. Berdasarkan pengujian terhadap sepuluh sampel, lima aplikasi bersih dan lima *malware*, sistem menunjukkan tingkat akurasi sebesar 100% pada dataset yang digunakan, dengan nilai *False Positive Rate (FPR)* sebesar 0. Penelitian ini mengintegrasikan hasil reverse engineering dan data deteksi VirusTotal ke dalam model penilaian risiko berbasis MCDM, yang secara sistematis menghasilkan skor tingkat bahaya. Dengan demikian, metode ini tidak hanya menemukan keberadaan malware, tetapi juga memberikan pengukuran tingkat risiko yang lebih interpretatif dibandingkan dengan metode deskriptif konvensional. Namun, model ini hanya dapat digunakan dengan empat kriteria dan analisis statis. Untuk meningkatkan generalisasi sistem, pengembangan lebih lanjut dapat dilakukan dengan menambahkan indikator perilaku dinamis atau memperluas jumlah sampel.

DAFTAR PUSTAKA

[1] R. P. Sari, "Apa itu Malware? Jenis dan Cara Pencegahannya." Accessed: Jan. 07, 2025. [Online]. Available: https://www.cloudcomputing.id/pengetahuan-dasar/apa-itu-malware-jenis

[2] T. A. Aziz, Z. Sari, C. Sri, and K. Aditiya, "Klasifikasi Malware android dengan menggunakan metode XGBoost Algoritma," *REPOSITOR*, vol. 7, no. 1, pp. 103–110, 2025, doi: https://doi.org/10.22219/repositor.v7i1.36564.

[3] T. N. Turnip, C. F. Manurung, and Y. S. Lubis, "Klasifikasi Malware Android Aplikasi Menggunakan Random Forest Berdasarkan Fitur Statik," *JATISI*, vol. 10, pp. 926–936, Mar. 2023, Accessed: Mar. 05, 2026. [Online]. Available: https://jurnal.mdp.ac.id/index.php/jatisi/article/download/3164/1181/

[4] R. B. Hadiprakoso, W. Rendra Aditya, F. N. Pramitha, P. Siber, and S. Negara, "Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning," 2022. doi: https://doi.org/10.14421/csecurity.2022.5.1.3116.

[5] A. Maslan *et al.*, "JEPIN (Jurnal Edukasi dan Penelitian Informatika) Klasifikasi dan Deteksi Malware Menggunakan Variasi Model Algoritma Machine learning," 2025, doi: https://doi.org/10.26418/jp.v11i1.87924.

[6] N. Widiyasono, H. Mubarok, and A. Fatwa, "Analisis Malware Ahmyth pada Platform Android Menggunakan Metode Reverse Engineering," 2022. doi: https://doi.org/10.29407/gj.v6i2.17749.

[7] F. De Santonario Magno Moises and A. Yogyakarta Joko Dwi Santoso, "Analisis Malware Android Menggunakan Metode Reverse Engineering," *JIKMA)*, vol. 1, no. 2, 2023, doi: https://doi.org/10.54066/jikma-itb.v1i2.169.

[8] M. K. Umam, A. Kafa Abdillah, A. Izzudin, and S. Nooriansyah, "Analisis Aplikasi Malware pada Android Menggunakan Reverse Engineering dan Framework D4I," *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 10, no. 2, 2025, doi: https://doi.org/10.55732/3zf9tx13.

[9] I. Puji Saputra and A. Hidayat, "Analisis Trojan Apk Menggunakan Metode Reverse Engineering Pada Serangan Phising," 2023. doi: https://doi.org/10.24127/jiki.v4i2.4619.

[10] Prasatya, "Mengenal Reverence Engineering dalam Dunia Cyber Security - CODEPOLITAN." Accessed: Jan. 07, 2025. [Online]. Available: https://www.codepolitan.com/blog/mengenal-reverence-engineering-dalam-dunia-cyber-security/

[11] O. E. Saied and K. H. Thanoon, "A Survey on Static and Dynamic Android Malware Analysis," Institute of Electrical and Electronics Engineers (IEEE), Dec. 2025, pp. 1–6. doi: 10.1109/icbats66542.2025.11258178.

[12] Y. W, Y. B. Fitriana, S. Esabela, and F. Hamdani, "Deteksi Serangan Malware Pada Web Aplikasi Menggunakan Metode Malware Analis Dinamis dan Statis," *Digital*

e-ISSN 2827-8356
p-ISSN 2827-9999

Transformation Technology, vol. 4, no. 1, pp. 461–470, Jul. 2024, doi: 10.47709/digitech.v4i1.4270.

[13] W. Yusnaeni and Y. Tajul Arifin, "Klasifikasi Tingkat Kepuasan Pengguna dengan Menggunakan Metode SAW Pada Layanan PTSP," *Jurnal INSAN (Journal of Information Systems Management Innovation*, vol. 5, no. 1, 2025, [Online]. Available: http://jurnal.bsi.ac.id/index.php/jinsan

[14] H. S. Gill, "Malware: Types, Analysis and Classifications." Accessed: Mar. 05, 2026. [Online]. Available: https://engrxiv.org/preprint/download/2423/4675

[15] Android, "App Architecture." Accessed: Feb. 23, 2026. [Online]. Available: https://developer.android.com/guide/topics/manifest/permission-element?utm_source=chatgpt.com