VOLUME 22 | NOMOR 1
APRIL 2025 | HALAMAN 1-111

P-ISSN : 1693-9166
E-ISSN : 2685-127X

Bit

Fakultas Teknologi Informasi
Universitas Budi Luhur

Fakultas Teknologi Informasi
Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260
https://journal.budiluhur.ac.id/index.php/bit/index

Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)
Vol. 22, No. 1, April 2025
p-ISSN: 1693-9166
e-ISSN: 2685-127X

Alamat Redaksi
Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)
Fakultas Teknologi Informasi, Universitas Budi Luhur
Jl. Ciledug Raya, Petukangan Utara, Jakarta Selatan, 12260
Telp: 021-585 3753 Fax: 021-585 3752

Bit (Fakultas Teknologi Informasi Universitas Budi Luhur) p-ISSN: 1693-9166
Vol. 22, No. 1, April 2025 e-ISSN: 2685-127X

MITRA BESTARI

Bit (Fakultas Teknologi Informasi Universitas Budi Luhur)
Vol. 22, No. 1, April 2025
p-ISSN: 1693-9166
e-ISSN: 2685-127X

Bit (Fakultas Teknologi Informasi Universitas Budi Luhur) p-ISSN: 1693-9166
Vol. 22, No. 1, April 2025, hlm. 70-76 e-ISSN: 2685-127X

RANCANG BANGUN SISTEM DETEKSI MALWARE DALAM FILE GAMBAR MENGGUNAKAN ANALISIS METADATA DAN VIRUSTOTAL

Joko Christian Chandra[1*], Muhammad Aldiansyah[2]

[1,2] Fakultas Teknologi Informasi, Teknik Informatika, Universitas Budi Luhur, DKI Jakarta, Indonesia

Email: [1]joko.christian@budiluhur.ac.id, [2]aldiansyahmuhammad215@gmail.com

(*: corresponding author)

(Naskah masuk: 16 Mei 2025, diterima untuk diterbitkan: 30 April 2025)

Abstrak

Risiko keamanan sistem informasi terdapat dalam banyak bentuk, di antaranya adalah dokumen gambar yang disisipkan *malware*, dengan tujuan untuk mengeksploitasi celah keamanan dalam pemrosesan *file* gambar. Teknik ini dikenal sebagai *stegomalware*, dengan tren peningkatan jumlah kasus. Kerugian yang ditimbulkan dari infeksi dan serangan yang dibuka stegomalware mencapai nilai jutaan dolar per kasus. Produk deteksi yang tersedia bagi pengguna perorangan sangat minim, dan harganya relatif tinggi. Penelitian ini bertujuan untuk menghasilkan solusi yang membantu mitigasi risiko keamanan tersebut. Penelitian ini dilaksanakan mengikuti kaidah metodologi *Waterfall* dan *Black box Testing* untuk memastikan keakuratan dan efektivitas sistem. Luaran dari penelitian berupa sistem untuk mendeteksi *malware* yang disembunyikan dalam gambar digital menggunakan teknik steganografi. Sistem memanfaatkan kombinasi dua alat utama yaitu ExifTool untuk menganalisis *metadata* dari *file* gambar dan VirusTotal untuk memindai *file* terhadap repositori *signature malware* yang dikumpulkan dari berbagai vendor antivirus. Sistem yang dihasilkan berbasis web dengan komponen fungsi *front-end* dan *backend* menggunakan *javascript*. Hasil pengujian menunjukkan bahwa sistem mampu melakukan deteksi yang efektif terhadap *file* gambar yang memiliki *malware* tersembunyi dengan tingkat keberhasilan tinggi 95%. Studi ini berkontribusi dalam pengembangan metode deteksi yang lebih terbuka untuk mengidentifikasi ancaman tersembunyi dalam *file* gambar, sehingga pengamanan tipe ini tidak harus bergantung sepenuhnya pada produk vendor keamanan spesifik.

Kata kunci: *deteksi malware, ExifTool, metadata, stegomalware, VirusTotal.*

DESIGN OF IMAGE MALWARE DETECTION SYSTEM USING METADATA ANALYSIS AND VIRUSTOTAL

Abstract

Information system security risks come in many forms, one of which involves image files embedded with malware, aiming to exploit vulnerabilities in image file processing. This technique, called stegomalware has been widely used and have increasing trend case. The loss caused by the infection and attack subsequent from stegomalware can reach millions of dollars for each case. The detection tools available for individual users is very low, with high price tag. This study aims to develop a solution to help mitigate for such security risks. The development of the system was conducted following the principles of the Waterfall methodology and Black Box Testing to ensure the accuracy and effectiveness of the system. The output of this research is a system designed to detect malware hidden in digital images using steganographic techniques. The system utilizes two main tools: ExifTool to analyze the metadata of image files, and VirusTotal to scan files for malware using signatures repository aggregated from various antivirus vendors. The resulting system is web based with function separated component of front-end and back-end using javascript. The test results show that the system is capable of effectively detecting image files containing hidden malware with a high success rate above 95%. This study contributes to the development of more open detection methods to identify hidden threats in image files, reducing reliance on specific security vendor products for this type of protection.

***Keywords**: ExifTool, malware detection, metadata, stegomalware, VirusTotal.*

1. PENDAHULUAN

Ancaman terhadap keamanan sistem informasi terus meningkat dengan munculnya beragam jenis perangkat lunak berbahaya, atau yang biasa dikenal dengan istilah *malware*. *Malware* adalah program atau kode jahat yang secara sengaja melakukan eksekusi fungsi yang destruktif [1]. *Malware* dapat dirancang untuk merusak, mengakses, mengubah, atau mencuri data pada target [2]. Hal ini terjadi tanpa izin pengguna dan merupakan salah satu ancaman utama bagi keamanan informasi di era digital karena dapat menyebabkan kerusakan yang signifikan pada sistem komputer, jaringan dan perangkat lainnya. *Malware* dapat menyusup melalui berbagai media, termasuk *file* gambar. Penyerang atau aktor jahat dapat menyembunyikan kode berbahaya di dalam *file* gambar menggunakan teknik Steganografi. Steganografi adalah metode menyembunyikan informasi dalam material yang tidak rahasia [3]. Hal ini membuat deteksi *malware* menjadi lebih sulit. *Malware* yang tersembunyi pada *file* multimedia seperti gambar dan menggunakan teknik steganografi telah menjadi tantangan untuk perusahaan, dan biasanya sebagai prekursor dari serangan yang rumit dan berbahaya [4]. Biaya tahunan yang dikeluarkan perusahaan untuk mempertahankan dan meningkatkan sistem pengawasan keamanan digital diperkirakan sebesar £30.000. Sementara itu, potensi kerugian yang ditimbulkan dari satu keberhasilan serangan yang menyisipkan data secara tersembunyi (steganografi) ke dalam sistem, termasuk *file* gambar dapat mencapai £250.000 [5]. Stegomalware mengalami tren peningkatan dari sisi kuantitas dan kompleksitas [6], oleh karena itu proses steganalisis gambar, untuk mendeteksi data tersembunyi pada gambar digital sangat penting dalam upaya meningkatkan keamanan digital [7]

Dalam layanan berbasis web yang mengizinkan pengguna untuk melakukan *upload file*, aktor jahat dapat menggunakan karakteristik kerja ini untuk menyerang dengan *malware* yang disembunyikan dalam *file*. Kondisi ini terjadi secara aktual pada situs dan web publik. Masalah yang muncul adalah kesulitan dalam mendeteksi *malware* yang disembunyikan di dalam *file* gambar. Sebagian besar perangkat lunak antivirus dan sistem deteksi intrusi tradisional berfokus pada analisis *file* atau aplikasi yang mencurigakan secara eksplisit. Namun, gambar yang tampaknya normal dapat menyembunyikan ancaman berbahaya tanpa meninggalkan jejak yang jelas. Hal ini menjadikan *malware* yang disembunyikan dalam gambar sulit untuk dikenali, terutama jika teknik steganografi digunakan terhadap *file* gambar tersebut. Kegagalan dari antivirus untuk mendeteksi jenis serangan ini pernah diteliti, di mana program antivirus gagal mendeteksi kode jahat yang disisipkan, karena antivirus menganggap *file* gambar bebas dari kode jahat [8]. Beberapa vendor keamanan besar seperti CrowdStrike dan Microsoft Endpoint memiliki fitur deteksi yang dimaksud, dengan biaya yang relatif tinggi.

Beberapa penelitian sebelumnya telah dilakukan untuk mendeteksi stegomalware. Penggunaan *tool* berbasis *python* yang menganalisis file JPEG untuk mendeteksi keberadaan *malware* dengan lokasi penyisipannya [9]. Menggunakan *fuzzy C-means clustering algorithm* untuk melakukan klasifikasi *payload* yang dicurigai [10]. Beberapa *tool* menggunakan *deep learning* untuk pemetaan dan klasifikasi [11], dan *machine learning* [12] . Mayoritas penelitian lebih bersifat klasifikasi dan perbandingan [13]. Penelitian ini lebih berfokus pada target yang lebih praktis dan aplikatif, ketimbang pemanfaatan teknik baru atau teknik novel. Luaran berupa *tool* yang bisa digunakan dengan mudah oleh pengguna dengan akses web browser tanpa perlu melakukan konfigurasi atau persyaratan infrastruktur khusus.

Berdasarkan riset, tingkat deteksi dan akurasi dari produk *scanner* yang tersedia di pasar gagal mendeteksi stegomalware dengan tingkat deteksi hanya 15% [9]. Riset terhadap 106 stegomalware dengan *tool* tersedia di lapangan memiliki tingkat deteksi di bawah 30% [13]. Dari latar belakang yang disampaikan, didapatkan permasalahan : Diperlukan sistem tanpa afiliasi vendor tunggal yang dapat digunakan untuk dapat mendeteksi *malware* yang disembunyikan dalam *file* gambar.

Salah satu alat yang dapat digunakan untuk menganalisis *metadata* pada berkas gambar adalah ExifTool. ExifTool merupakan perangkat lunak *open-source* yang independen dari *platform* dalam *library* Perl dan aplikasi perintah teks (*command line*) digunakan untuk ekstraksi, membaca, menulis dan manipulasi *metadata* yang tertanam pada berbagai format berkas, termasuk gambar[14]. *Metadata* sering kali memuat informasi tersembunyi yang dapat menjadi indikator adanya *malware* dalam *file* gambar.

Untuk mendeteksi *malware* dan virus, secara tradisional dapat digunakan aplikasi antivirus yang dikembangkan oleh vendor industri. Tingkat efektivitas dan akurasi dari *engine* antivirus produk vendor satu dengan yang lainnya sangat bervariasi, sehingga hasil analisis dari multi vendor merupakan solusi yang paling tepat. Untuk mendapatkan hasil analisis multi vendor antivirus, dapat menggunakan layanan VirusTotal. VirusTotal adalah layanan yang digunakan untuk memindai berkas dan URL (*Uniform Resource Locators*) terhadap berbagai macam ancaman menggunakan banyak *engine* antivirus. VirusTotal menyediakan analisis secara menyeluruh terhadap berkas *file* yang dicurigai mengandung *malware*. VirusTotal adalah *scanner online* yang paling populer dan sering digunakan untuk menentukan apakah sampel yang di periksa bersifat *malicious* dan memiliki pengelompokan dan koleksi sampel yang ekstensif [15] .

Penggunaan ExifTool dan VirusTotal secara bersamaan dapat meningkatkan efektivitas dalam

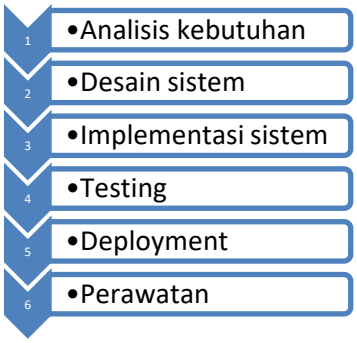
mendeteksi *malware* pada gambar. Berdasarkan kondisi dan permasalahan yang disampaikan, Penelitian ini bertujuan untuk mengidentifikasi dan menganalisis keberadaan *malware* dalam *file* gambar.

Melalui analisis dan implementasi pengembangan perangkat lunak, penelitian ini bertujuan untuk menghasilkan sistem yang dapat menganalisis keberadaan *malware* yang tersembunyi dalam gambar dengan menggunakan dua alat utama, yaitu ExifTool dan VirusTotal. Penelitian ini bermanfaat untuk mengurangi beban pengguna, khususnya administrator sistem untuk mengidentifikasi dan mengatasi ancaman *malware* dalam *file* gambar. Sistem ini berbeda dari solusi yang tersedia di lapangan karena berbasiskan mekanisme *open source* dan layanan yang tersedia cuma-cuma, memiliki keunggulan utama dari sisi biaya yang dikeluarkan oleh pengguna.

2. METODE PENELITIAN

2.1 Metodologi penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah *Waterfall* dan *Black box Testing*. Metode *Waterfall* adalah salah satu model pengembangan perangkat lunak yang sekuensial dari sisi desain, umumnya digunakan pada proses pengembangan perangkat lunak, di mana kemajuan mengalir berurutan melalui fase-fase, di mana setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya [16]. Dalam konteks penelitian ini, model *Waterfall* digunakan untuk merencanakan dan melaksanakan pengembangan sistem deteksi *malware* pada gambar. Meskipun termasuk metodologi lama, pola pengembangan ini paling tepat untuk pengembangan sistem yang memiliki kebutuhan terdefinisi jelas, dan target luaran yang statis. Tahapan dalam model *Waterfall* terdiri dari beberapa fase seperti pada Gambar 1.



Gambar 1. Tahapan metodologi Waterfall

Berikut adalah penjelasan langkah-langkah yang dilakukan pada tiap tahap

1. Fase Analisis Kebutuhan: Pada tahap ini, peneliti mengidentifikasi kebutuhan sistem dan menentukan apa yang perlu dicapai dalam penelitian. Dalam penelitian ini, kebutuhan utama yang ditangkap adalah diperlukan sistem yang mampu mendeteksi dan menganalisis *malware* yang disembunyikan dalam *metadata* gambar atau menggunakan teknik steganografi. Kebutuhan turunan yang ditangkap adalah kebutuhan fungsional terkait fungsi yang harus tersedia dan kebutuhan non-fungsional yang berkaitan dengan karakteristik sistem.
2. Fase Desain Sistem: Desain sistem dilakukan setelah kebutuhan fungsional dan non-fungsional dianalisis. Peneliti merancang arsitektur sistem, antarmuka pengguna (UI), rancangan algoritma deteksi *malware* beserta integrasinya dengan ExifTool dan VirusTotal. Juga dilakukan desain rencana pengujian sistem.
3. Fase Implementasi: Fase implementasi adalah tahap pengembangan perangkat lunak sesuai dengan desain. Pada tahap ini, peneliti melakukan pemrograman dengan *javascript* dalam runtime node.js.
4. Fase Pengujian: Setelah sistem selesai dibangun, sistem akan diuji untuk memastikan bahwa semua fungsionalitas bekerja dengan baik dan sesuai dengan kebutuhan yang telah ditentukan. Pola pengujian menggunakan rencana yang telah ditentukan dan diukur hasilnya.
5. Fase Perawatan: Setelah sistem berhasil diuji dan diterapkan, fase ini akan fokus pada pemeliharaan sistem, perbaikan *bug* / kesalahan sistem, dan pengoptimalan sistem berdasarkan masukan dan umpan balik yang diterima dari pengguna atau hasil pengujian. Penelitian ini tidak mencakup fase perawatan secara ekstensif.

2.2 Metodologi pengujian

Pengujian dilakukan dengan pola black box testing untuk fungsionalitas sistem. Pengujian black box merupakan metode pengujian yang tidak memerlukan akses terhadap struktur internal atau kode sumber dari objek yang diuji [17]. Untuk mengukur akurasi dari sistem, peneliti juga menyiapkan sebanyak 200 gambar sebagai data set. 100 gambar disisipi dengan teknik steganografi (Least Significant Bit, masking dan transformasi domain) berisi berbagai jenis malware. 100 gambar sebagai kontrol tanpa disisipi malware. Pada Tabel 1 dan Tabel 2 adalah rencana pengujian *black box* yang digunakan.

Tabel 1. Tabel rencana pengujian

Fungsi utama	Kasus uji	Input	Output yang diharapkan
Upload file	Upload gambar valid (JPG/PNG, ≤ ukuran maksimum)	Gambar valid dengan ukuran valid	Status: Upload berhasil, file disimpan, pemeriksaan dimulai
Deteksi data dengan ExifTool	Gambar dengan metadata	Gambar dengan metadata (baik stego maupun normal)	Metadata terdeteksi
Deteksi steganografi	Gambar dengan stego tersembunyi	Gambar dengan stego tersembunyi	Stego ditemukan atau tidak ditemukan

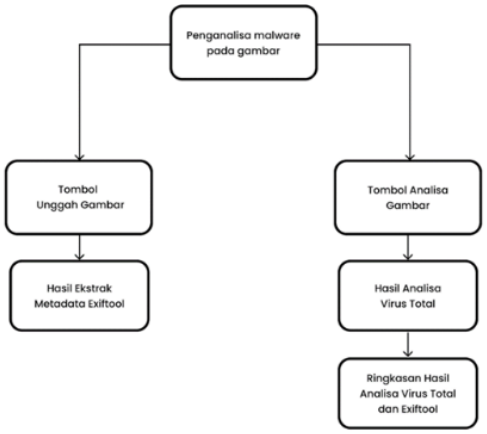
Tabel 2. Tabel rencana pengujian (Lanjutan)

Fungsi utama	Kasus uji	Input	Output yang diharapkan
Cek Malware	File hash match dengan signature malware pada VirusTotal	Gambar dengan hash yang diketahui mengandung malware	Output: Malware terdeteksi, hasil VirusTotal, detail vendor yang mendeteksi
Penilaian keseluruhan	Integrasi dari seluruh proses	Semua input valid + input gambar mengandung malware	Output: hasil pemeriksaan sesuai deteksi

3. HASIL DAN PEMBAHASAN

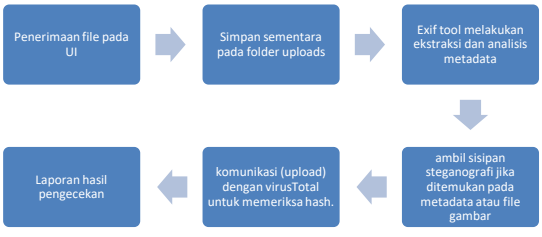
3.1 Perancangan

Mempertimbangkan kemudahan akses dan *deployment*, sistem dirancang berbasis web. Secara spesifik akan memisahkan komponen *user interface* (front end) dengan komponen pengolahan (*back-end*). Pada tahap desain dihasilkan beberapa rancangan, di antaranya rancangan menu *User interface* sesuai Gambar 2.



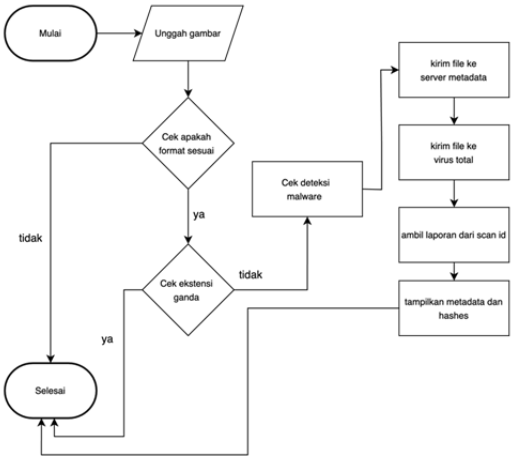
Gambar 2. Rancangan menu user interface

Pola kerja sistem yang disusun secara sekuensial dengan urutan seperti pada Gambar 3.



Gambar 3. Urutan kerja sistem

Berdasarkan urutan kerja tersebut disusun diagram alir seperti pada gambar 4.



Gambar 4. Diagram alir sistem

Berikut adalah penjelasan dari urutan kerja sistem dan diagram alir :

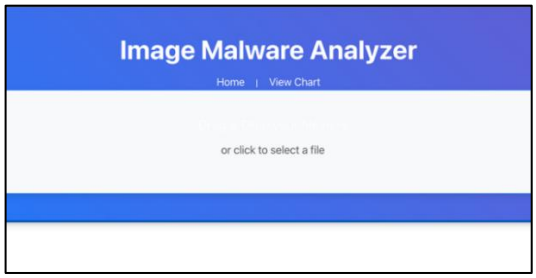
1. Penerimaan *file* pada *user interface* mencakup proses *user* melakukan pemilihan *file* gambar dan melakukan proses *upload*. Format *file* bukan gambar tidak di izinkan untuk di *upload*.
2. Proses cek pertama berupa apakah memiliki ekstensi ganda. Jika tipenya ekstensi ganda maka ditolak.
3. Pengecekan *malware* dimulai dengan ekstraksi *metadata* dan hasilnya dikirim ke *back-end* server *metadata* untuk memeriksa hal berikut:
   a. Ukuran *File*: Perubahan signifikan pada ukuran *file* dapat mengindikasikan adanya penyisipan data.
   b. Format *File*: Memastikan format *file* tetap sesuai (misalnya, .jpg atau .png) dan tidak memiliki ekstensi ganda.
   c. Atribut *Metadata*: Memeriksa entri mencurigakan seperti atribut yang dimodifikasi untuk menyembunyikan keberadaan *malware*.
   d. Struktur *metadata* Cari entri yang tidak lazim atau aneh, seperti atribut yang terlalu panjang atau data yang dienkripsi.
   e. Mendeteksi Tanda Penyisipan: Atribut tambahan atau data yang tidak dikenal di dalam *metadata*.
4. *File* dan *metadata* yang ada, atau sisipan steganografi yang ditemukan dikirim ke VirusTotal. Pengiriman menggunakan API VirusTotal metode HTTP POST dengan header autentikasi yang berisi kunci API.
5. Ambil hasil *scan-id* VirusTotal untuk kemudian mengambil laporan deteksi
6. Pengambilan hasil deteksi VirusTotal menggunakan HTTP GET
7. Penyajian hasil dari hasil keseluruhan cek ke UI.

3.2 Implementasi

Implementasi menggunakan bahasa pemrograman *javascript* dalam dua bentuk: yang pertama memanfaatkan *library* react.js untuk *front-*

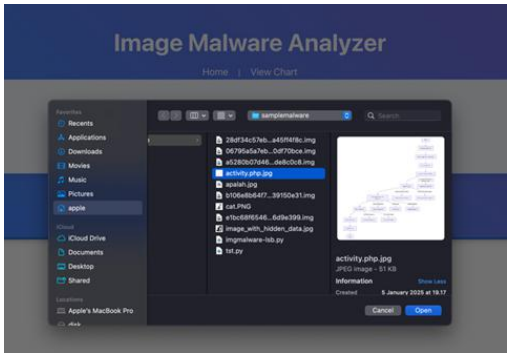
end, yang kedua menggunakan run-time environment Node.js untuk API di sisi *back-end*.

Tampilan layar awal dari sistem menggunakan UI sederhana yang meminta *input file* dari *user*, seperti pada Gambar 5.



Gambar 5. Halaman utama

Saat *user* melakukan pemilihan gambar, akan ditampilkan *open file dialog box* sesuai dengan OS dan *browser* yang digunakan, seperti pada Gambar 6.

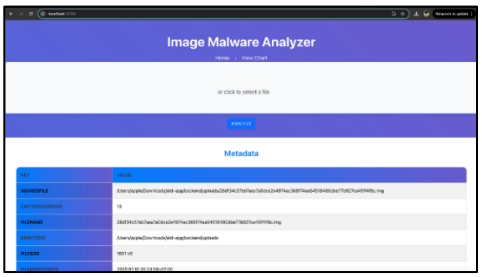


Gambar 6. Antarmuka pemilihan file

Setelah *upload* selesai, maka tombol "*Analyze*" akan ditampilkan seperti pada Gambar 7.

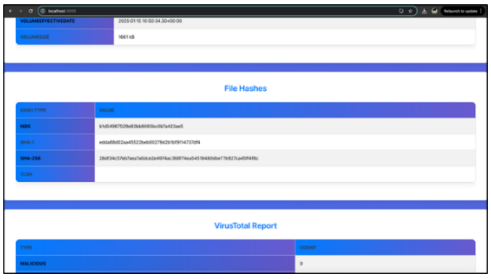
Gambar 7. Antarmuka setelah upload

Setelah *user* melakukan klik "*Analyze*", proses akan berjalan di latar belakan dan menampilkan hasil pengecekan berupa informasi tentang status gambar, apakah terdeteksi mengandung *malware* atau tidak, serta jenis *malware* yang terdeteksi. Gambar 8 menunjukkan *metadata file* gambar.



Gambar 8. Metadata file gambar

Pada Gambar 9 menunjukkan contoh tampilan hasil deteksi *malware* nol (bersih) dari kasus uji yang diberikan.



Gambar 9. Tampilan layar hasil deteksi bersih

Pada Gambar 10 menunjukkan contoh tampilan hasil deteksi *malware* yang positif pada *file* gambar yang diberikan.

Gambar 10. Tampilan layar malware terdeteksi

Tabel 3. Hasil pengujian fungsionalitas

Fungsi utama	***Output* yang diharapkan**	***Output* yang dihasilkan**
Upload file	Status: *Upload* berhasil, *file* disimpan, pemeriksaan dimulai	*Upload file* berhasil. Pemeriksaan oleh *backend* server berjalan normal
Deteksi data dengan ExifTool	*Metadata* terdeteksi	*Metadata* terdeteksi
Deteksi steganografi	Stego ditemukan atau tidak ditemukan	*Stego file* ditemukan pada *file* yang terinfeksi, sedangkan tidak ada *stego file* pada *file* yang bersih.
Cek *Malware*	*Output*: *Malware* terdeteksi, hasil VirusTotal, detail vendor yang mendeteksi	*Output* hasil deteksi *malware* dan VirusTotal terdeteksi.
Penilaian keseluruhan	*Output*: hasil pemeriksaan sesuai deteksi	Hasil pemeriksaan sesuai dengan deteksi.

3.3 Pengujian

Pengujian fungsionalitas sesuai dengan rencana pengujian dapat dilihat pada Tabel 3. Hasil yang didapat adalah fungsi berjalan sesuai dengan rencana dan rancangan.

Pengujian secara kuantitatif menggunakan sampel 200 *file* gambar, terdiri dari 100 *file* gambar yang terinfeksi dan 100 *file* gambar normal. Hasilnya dapat dilihat pada Tabel 4.

Tabel 4. Pengujian akurasi sistem

Alat Deteksi	Gambar dengan *malware* ter-deteksi (*True Positive*)	Gambar dengan *malware* Tidak Ter-deteksi (*False Negative*)	Gambar tanpa *malware* Ter-deteksi (*False Positive*)	Gambar tanpa *malware* Tidak Ter-deteksi (*True Negative*)
ExifTool	95	5	5	95
VirusTotal	98	2	2	98

Berdasarkan data hasil pengujian akurasi sistem, kedua komponen sistem berfungsi dengan baik. Menggunakan data yang terkontrol sebanyak 100 file, yang sudah diketahui kondisi positif-negatif *malware*. Akurasi dari ExifTool sebesar 95% untuk mendeteksi keberadaan *malware* dan ketiadaan *malware*. Sebesar 5% kesalahan deteksi. Akurasi dari VirusTotal sebesar 98% untuk mendeteksi keberadaan dan ketiadaan *malware*, dengan 2 % kesalahan deteksi. Ini merupakan hasil yang menunjukkan tingkat akurasi tinggi.

4. KESIMPULAN

Berdasarkan penelitian yang dilaksanakan, kesimpulan yang didapatkan adalah :

1. Luaran dari penelitian ini berupa sistem yang fungsional untuk mendeteksi *malware* yang disembunyikan secara steganografi pada *file* gambar digital.
2. Dalam pengujian kuantitatif efektivitas deteksi ExifTool adalah 95%, dan efektivitas VirusTotal adalah 98%, karena kedua jenis deteksi digunakan, secara rerata didapatkan 96.5% efektivitas. Penggunaan keduanya menyajikan informasi yang lebih lengkap dan mendalam.
3. Mekanisme kerja sistem ini meningkatkan deteksi *malware* pada file, karena tidak seperti *tool* umum di lapangan, bagian yang dicurigai berupa *malware* di pisahkan terlebih dahulu sebelum diperiksa *signature*nya.
4. Bagian *backend* dari sistem bisa digunakan sebagai API untuk implementasi sistem lain. Sehingga aplikasi pada *user interface* lain, atau automasi berbasis script untuk pemeriksaan massal dapat diterapkan.
5. Sistem ini belum menerapkan automasi tindakan terhadap *file* yang diperiksa, intervensi manual masih perlu dilakukan pengguna terhadap *file* gambar yang diperiksa.

Keilmuan keamanan digital terus berkembang, sehingga untuk peningkatan kualitas deteksi dan disarankan penambahan mekanisme cek *file* dapat ditambah, seperti pemeriksaan anomali pada bagian akhir *file*, atau menggunakan analisis statistik dengan bantuan *Artificial Intelligence*.

DAFTAR PUSTAKA

[1] V. Yadav and others, “A survey on machine learning based malware detection in executable files,” *Journal of Systems Architecture*, vol. 112, p. 101861, 2021, doi: 10.1016/j.sysarc.2020.101861.

[2] A. M. Montoya-Martínez and others, “A Review of Attacks, Vulnerabilities, and Defenses in Industry 4.0 with New Challenges on Data Sovereignty Ahead,” *Sensors*, vol. 21, no. 15, p. 5189, 2021, doi: 10.3390/s21155189.

[3] M. Płachta, M. Krzemień, K. Szczypiorski, and A. Janicki, “Detection of Image Steganography Using Deep Learning and Ensemble Classifiers,” *Electronics (Basel)*, vol. 11, no. 10, p. 1565, 2022, doi: 10.3390/electronics11101565.

[4] R. Chaganti, V. Ravi, M. Alazab, and T. D. Pham, “Stegomalware: A systematic survey of malware hiding and detection in images, machine learning models and research challenges,” 2021. [Online]. Available: https://arxiv.org/abs/2110.02504

[5] P. X. and others, “Steganography and Probabilistic Risk Analysis: A game theoretical framework for quantifying adversary advantage and impact,” 2024. [Online]. Available: https://arxiv.org/abs/2412.17950v1

[6] C. Jeffcoat, “Stegomalware’s Growing Role in Modern Cyberthreats: Trends, Techniques, and the Need for Advanced Detection,” https://www.wetstonelabs.com/stegomalwares-growing-role-in-modern-cyberthreats-trends-techniques-and-the-need-for-advanced-detection/.

[7] B. Li, N. Li, J. Yang, and others, “Image Steganalysis using Active Learning and Hyperparameter Optimization,” *Sci Rep*, vol. 15, p. 7340, 2025, doi: 10.1038/s41598-025-92082-w.

[8] M. D. S. S. R. N. Wijesinghe and E. P. U. Ratnayake, “Framework for Malware Triggering Using Steganography,” *Applied Sciences*, vol. 12, no. 16, p. 8176, 2022, doi: 10.3390/app12168176.

[9] V. Verma, S. K. Muttoo, and V. B. Singh, “Detecting Stegomalware: Malicious Image Steganography and Its Intrusion in Windows,” 2022, pp. 103–116. doi: 10.1007/978-981-16-9089-1_9.

[10] A. Monika and R. Eswari, "An ensemble-based stegware detection system for information hiding malware attacks," *J Ambient Intell Humaniz Comput*, vol. 14, no. 4, pp. 4401–4417, Apr. 2023, doi: 10.1007/s12652-023-04559-z.

[11] A. Liguori, M. Zuppelli, D. Gallo, M. Guarascio, and L. Caviglione, "A deep learning-based approach for stegomalware sanitization in digital images," *J Intell Inf Syst*, Apr. 2025, doi: 10.1007/s10844-025-00936-6.

[12] P. Kadebu *et al.*, "A hybrid machine learning approach for analysis of stegomalware," *International Journal of Industrial Engineering and Operations Management*, vol. 5, no. 2, pp. 104–117, Jun. 2023, doi: 10.1108/IJIEOM-01-2023-0011.

[13] F. Strachanski, D. Petrov, T. Schmidbauer, and S. Wendzel, "A Comprehensive Pattern-based Overview of Stegomalware," in *Proceedings of the 19th International Conference on Availability, Reliability and Security*, New York, NY, USA: ACM, Jul. 2024, pp. 1–10. doi: 10.1145/3664476.3670886.

[14] K. R. Rimkus and K. M. Hess, "ExifTool: platform-independent Perl library and command-line application for manipulating embedded metadata in diverse file formats," 2024.

[15] K. van Liebergen, J. Caballero, P. Kotzias, and C. Gates, "A deep dive into VirusTotal: characterizing and clustering a massive file feed," 2022. [Online]. Available: https://arxiv.org/abs/2210.15973

[16] A. Mishra, S. K. Sahu, and P. P. Panda, "Structured software development versus agile software development: a comparative analysis," *International Journal of System Assurance Engineering and Management*, vol. 14, pp. 1504–1522, 2023.

[17] Z. Aghababaeyan, M. Abdellatif, L. Briand, and M. Bagherzadeh, "Black Box Testing of Deep Neural Networks Through Test Case Diversity," *IEEE Transactions on Software Engineering*, pp. 1–26, 2023, doi: 10.1109/TSE.2023.3243522.