Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Deteksi Malware Trojan Klasik Berbasis Web: Studi Kasus Miralab

Michelino Fransiskus Sinurat [1], Nur Cahyono Kuhardianto[2]
[1,2]Teknik Informatika, Rakayasa Keamanan Siber, Politeknik Negeri Batam
michelinofransiskussinurat@gmail.com[1] , anung@polibatam.ac.id[2]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword: *Windows Security, Trojan Detection, CEX, ILOVEYOU, WannaCry, Signature-based Detection, Malware Analysis, Web-based System*

ABSTRACT

Lingkungan Windows merupakan target utama berbagai serangan malware klasik, termasuk *CEX*, *ILOVEYOU*, dan *WannaCry* ransomware, yang memanfaatkan kelemahan sistem maupun interaksi pengguna untuk menyebar secara cepat. Beberapa penelitian terdahulu berfokus pada deteksi *WannaCry* melalui pendekatan *rule-based detection* terhadap eksploitasi celah *SMB* di *Windows*, kajian historis mengenai penyebaran *ILOVEYOU* dan *WannaCry*, serta tinjauan literatur terkait evolusi teknik *malware* dari generasi awal hingga serangan global. Selain itu, pendekatan berbasis *machine* learning telah diusulkan untuk mendeteksi varian baru pada sistem *Windows* tanpa bergantung pada signature tradisional. Namun, metode tersebut sering membutuhkan dataset besar, sumber daya komputasi tinggi, dan infrastruktur yang kompleks. Berbeda dari penelitian sebelumnya, Miralab dirancang sebagai sistem berbasis web untuk deteksi multi-trojan klasik pada *Windows*, dengan mengombinasikan *signature-based detection* menggunakan *yara-python*, analisis statis file PE dengan pefile, serta fitur *auto-scan* folder yang ringan. Hasil pengujian menunjukkan bahwa Miralab mampu mendeteksi file berbahaya pada Windows dengan akurasi tinggi dan menyajikan laporan terstruktur secara otomatis. Dengan demikian, penelitian ini menghadirkan solusi sederhana, fleksibel, dan aplikatif untuk mendukung keamanan siber pada sistem operasi Windows.

I. PENDAHULUAN

Perkembangan teknologi informasi telah membawa kemudahan dalam aktivitas digital, tetapi juga memunculkan ancaman serius berupa serangan siber. Salah satu bentuk ancaman yang paling menonjol adalah malware trojan, yang bekerja dengan menyamar sebagai perangkat lunak sah untuk kemudian merusak sistem atau mencuri data. Serangan trojan klasik seperti *ILOVEYOU* yang merebak pada tahun 2000 serta ransomware *WannaCry* pada tahun 2017 menjadi contoh nyata betapa besar dampak yang ditimbulkan, mulai dari kerugian finansial hingga gangguan sistem informasi berskala global[1]. Penelitian terkini menegaskan bahwa malware berbasis trojan tetap relevan hingga saat ini karena sifatnya yang mudah dimodifikasi dan disebarkan melalui berbagai vektor serangan[2].

Metode deteksi malware yang dominan masih berbasis tanda tangan (*signature-based*), yang meski efektif untuk varian lama, kerap lemah dalam mengantisipasi *trojan* dengan sifat polimorfik dan kemampuan penyamaran canggih[3]. Beberapa platform analisis malware berbasis web telah dikembangkan untuk menjawab kebutuhan ini. Namun, keterbatasan seperti batas ukuran file unggahan, kapasitas deteksi, dan keterbatasan akses membuat layanan tersebut kurang fleksibel dalam menangani jumlah data besar[4]. Hal ini menimbulkan kebutuhan akan sistem deteksi berbasis web yang tidak hanya mampu mengidentifikasi trojan klasik, tetapi juga memberikan fleksibilitas unggahan dan kecepatan dalam proses analisis.

Dalam konteks kebutuhan tersebut, penulis telah mengembangkan aplikasi Miralab sebagai solusi inovatif yang dirancang khusus untuk mengatasi tantangan deteksi *malware trojan* klasik. Aplikasi ini merupakan karya original penulis yang dikembangkan dari nol dengan mempertimbangkan kekurangan-kekurangan sistem existing yang ada di pasaran. Penulis memperkenalkan Miralab sebagai alternatif baru yang tidak hanya mampu mendeteksi *malware trojan* klasik seperti *CEX, WannaCry*, dan *ILOVEYOU*, tetapi juga menawarkan pendekatan berbasis web yang ringan, praktis, dan *user-friendly*. Miralab dikembangkan dengan filosofi *democratization of cybersecurity*, dimana penulis berupaya memberikan akses deteksi *malware* berkualitas tanpa ketergantungan pada layanan komersial mahal atau infrastruktur kompleks.

e-ISSN: 2548-6861

Selain berfokus pada pengembangan aplikasi berbasis web, penelitian ini juga menitikberatkan pada penerapan deteksi berbasis signature menggunakan yara-python. Library ini memungkinkan pembentukan aturan (YARA rules) untuk mengenali pola byte spesifik dari berbagai jenis malware tanpa perlu membangun model machine learning baru.

II. TINJAUAN PUSTAKA

A. Klasifikasi Pendekatan Deteksi Malware

1) Pendekatan Rule-based Detection:

Pendekatan *rule-based detection* memanfaatkan aturan dan pola spesifik untuk mengidentifikasi malware berdasarkan karakteristik tertentu. Lu et al. mengembangkan sistem deteksi *WannaCry* secara komprehensif melalui pemanfaatan prinsip dasar serangan, perumusan aturan spesifik, serta eksperimen validasi yang menekankan rule-based detection untuk mengenali pola *byte* maupun perilaku jaringan, termasuk eksploitasi celah SMB pada sistem operasi *Windows*, dan hasilnya terbukti efektif dalam mencegah penyebaran *ransomware* tersebut[5].

Penelitian serupa oleh Studi lain menunjukkan bahwa pendekatan berbasis aturan mampu mencapai akurasi tinggi untuk malware dengan signature yang sudah dikenal, namun memiliki keterbatasan dalam mendeteksi varian baru atau malware dengan kemampuan polimorfik[6].

2) Pendekatan Machine Learning dan AI:

Perkembangan teknologi *machine learning* telah membuka peluang baru dalam deteksi malware adaptif. Park et al. mengembangkan sistem deteksi malware menggunakan *static analysis* dan *machine learning* pada file PE dengan memanfaatkan header *feature values* untuk mengidentifikasi pola malware dengan akurasi tinggi[6]. Penelitian ini mendemonstrasikan efektivitas kombinasi *static analysis* dengan *stacking methods* untuk deteksi *malware* baru. Namun, pendekatan ini memerlukan *dataset* pelatihan yang besar dan preprocessing yang intensif untuk ekstraksi fitur dari PE header.

Owoh et al. mengusulkan *model hybrid* menggunakan *Gated Recurrent Unit-Generative Adversarial Network* (GRU-GAN) untuk mendeteksi malware berdasarkan sekuensi *API call*, yang terbukti efektif dalam mengidentifikasi varian malware baru dengan *F1-score* 94.2%[4]. Meskipun menjanjikan, implementasi model ini membutuhkan infrastruktur komputasi yang kompleks dan biaya operasional tinggi, sehingga kurang praktis untuk pengguna umum atau institusi dengan keterbatasan sumber daya.

3) Analisis Historis dan Evolusi Malware:

Sanmorino dan Zahra melakukan tinjauan historis komprehensif mengenai evolusi ancaman digital, menekankan pola penyebaran berulang malware klasik seperti ILOVEYOU dan *WannaCry*, serta urgensi pengembangan sistem deteksi dini yang adaptif[7]. Studi ini memberikan perspektif penting mengenai siklus hidup malware dan pola serangan yang cenderung berulang dengan modifikasi teknis.

Alenezi et al. menyoroti evolusi malware dengan meninjau berbagai serangan mulai dari *PCwrite Trojan* (1986), *ILOVEYOU worm*, hingga *WannaCry* ransomware, serta menekankan bahwa perkembangan teknik penyebaran malware semakin kompleks seiring waktu[8]. Penelitian ini mengidentifikasi bahwa malware klasik tetap relevan karena:

- Kemudahan modifikasi kode,
- Efektivitas vektor serangan sosial engineering, dan
- Kelemahan keamanan yang persisten pada sistem legacy.

B. Analisis Komparatif Penelitian Terdahulu

Untuk memahami posisi penelitian ini dalam konteks yang lebih luas, dilakukan analisis komparatif terhadap berbagai pendekatan deteksi malware yang telah dikembangkan:

TABEL I
PERBANDINGAN PENELITIAN PENDEKATAN DETEKSI MALWARE

Aspek	Lu et al.[5]	Sanmorino & Zahra[7]	Alenezi et al.[8]	Park et al.[6]	Owoh et al.[4]	Miralab
Metode Deteksi	*Rule-based (SMB exploitation)*	*Historical analysis*	*Literature review*	*ML + Static PE analysis*	*GRU-GAN hybrid*	*Signature + PE analysis*
Target Malware	*WannaCry* spesifik	*Multiple classic malware*	*General malware evolution*	*PE-based malware*	*API-based malware*	*Classic trojans (CEX, ILOVEYOU, WannaCry)*
Platform	*Windows (SMB focus)*	*Cross-platform analysis*	*General review*	*Windows PE files*	*Cross-platform*	*Windows web-based*

Resource Requirements	*Low-moderate*	N/A (*review*)	N/A (*review*)	*Moderate (ML training)*	*High (complex infrastructure)*	*Low (local processing)*
Implementasi Praktis	Ya (*rule engine*)	Tidak	Tidak	Ya (*prototype*)	Ya (*proof-of-concept*)	Ya (*production-ready*)
Scalability	*Moderate (rule expansion)*	N/A	N/A	*High (model scaling)*	*High (distributed)*	*Moderate (local scanning)*
Accessibility	*Technical users*	*Academic*	*Academic*	*Technical users*	*Technical experts*	*General users*
Privacy	*Local processing*	N/A	N/A	*Dataset dependent*	*Cloud/local hybrid*	*Full local privacy*
Cost	*Free (if implemented)*	N/A	N/A	*Moderate (infrastructure)*	*High (development)*	*Free (open source)*

C. Identifikasi Research Gap

Berdasarkan analisis komparatif di atas, teridentifikasi beberapa kekosongan penelitian yang signifikan:

1) Keterbatasan Fokus dan Cakupan:

Penelitian existing menunjukkan dua ekstrem: terlalu spesifik (seperti fokus hanya pada WannaCry [5]) atau terlalu umum tanpa implementasi praktis[7][8]. Belum ada solusi yang secara khusus menangani *multiple classic trojans* dengan pendekatan terintegrasi yang praktis.

2) Kompleksitas Implementasi dan Resource Requirements:

Pendekatan *machine learning modern*[4] meskipun akurat, membutuhkan:

- *Dataset* pelatihan besar (100K+ *samples*)
- Infrastruktur komputasi tinggi (*GPU clusters*)
- Expertise teknis mendalam (*data science*, ML *engineering*)
- Biaya operasional tinggi untuk *maintenance* dan *update model*

Hal ini menciptakan *barrier* yang tinggi untuk adopsi oleh institusi pendidikan, UKM, atau pengguna individual.

3) Keterbatasan Aksesibilitas dan Privacy:

Solusi *commercial* seperti *VirusTotal* dan *MetaDefender* memiliki limitasi:

- Batasan ukuran file upload (650MB - 200MB)
- *Rate limiting* pada *API* gratis
- Data sharing ke komunitas (*privacy concerns*)
- Ketergantungan pada koneksi internet

4) Kurangnya Solusi Hybrid Sederhana

Belum ada penelitian yang mengombinasikan kelebihan signature-based detection (ringan, cepat) dengan analisis statis file PE dalam satu sistem terintegrasi yang *user-friendly* untuk deteksi *multi-trojan* klasik.

D. Positioning Miralab dalam Landscape Penelitian

Miralab dirancang untuk mengisi research gap yang teridentifikasi dengan menghadirkan kontribusi berupa:

1) Hybrid Approach yang Efisien:

Berbeda dari pendekatan tunggal pada penelitian terdahulu, Miralab mengombinasikan:

- *Signature-based detection* menggunakan *yara-python* untuk pengenalan pola spesifik *trojan* klasik
- *Static analysis* dengan pefile untuk ekstraksi metadata dan struktur PE files
- *Automated scanning* dengan file *watcher* untuk monitoring

2) Fokus Praktis pada Classic Trojans:

Sementara penelitian lain berfokus pada malware umum[4] atau *single malware*[5], Miralab secara spesifik menargetkan *trojan* klasik (*CEX*, *ILOVEYOU*, *WannaCry*) yang masih relevan dan sering menjadi basis pengembangan varian baru.

3) Accessibility dan Democratization:

Miralab mengatasi barrier accessibility dengan menyediakan:

- *Web-based interface yang user-friendly*
- *Local processing* tanpa ketergantungan *cloud*
- *Resource-efficient operation* (standard *PC capable*)
- *Free* dan *open-source approach*

4) Privacy-Preserving Design:

Berbeda dari solusi cloud-based, Miralab memastikan:

- Semua *processing* dilakukan secara *local*
- Tidak ada data sharing atau upload ke server eksternal
- *User* memiliki kontrol penuh terhadap hasil *scanning*
- Mendukung *compliance* dengan regulasi *privacy*

E. Justifikasi Penelitian

Berdasarkan analisis gap dan *positioning* di atas, penelitian Miralab dapat dijustifikasi sebagai kontribusi yang signifikan karena:

1) Mengisi Kekosongan Solusi Praktis Multi-Trojan:

e-ISSN: 2548-6861

Penelitian ini menghadirkan solusi pertama yang secara spesifik menangani deteksi multi-trojan klasik dengan pendekatan hybrid yang praktis dan dapat diimplementasikan langsung, mengisi gap antara solusi *academic* (*theoretical*) dan *commercial* (*expensive/privacy-invasive*).

2) Menyediakan Alternatif Lightweight terhadap ML-Heavy Solutions

Miralab menawarkan alternative approach yang tidak memerlukan machine learning infrastructure yang *complex*, namun tetap mampu memberikan *detection accuracy* yang *acceptable* untuk target specifik (*classic trojans*), menjadikannya *viable option* untuk *resource-constrained environments*.

3) Mendemokratisasi Akses Deteksi Malware

Penelitian ini berkontribusi dalam mendemokratisasi akses terhadap malware detection tools untuk pengguna non-technical, institusi pendidikan dengan budget terbatas, dan small-medium enterprises yang membutuhkan proteksi namun tidak mampu invest dalam expensive commercial solutions.

4) Mempertahankan Privacy dalam Era Data Surveillance

Miralab memberikan alternatif *privacy-preserving* dalam landscape yang didominasi *cloud-based solutions* yang berpotensi menimbulkan data *privacy risks*, *supporting the growing demand for local-first security tools*.

Dengan demikian, Miralab bukan hanya memberikan solusi teknis yang functional, tetapi juga menjawab kebutuhan praktis dan strategik dari komunitas pengguna yang selama ini kurang terakomodasi oleh penelitian akademis existing maupun solusi komersial available di market.

III. METODE PENGEMBANGAN

Dalam penelitian ini, metodologi yang digunakan adalah model Waterfall dipilih karena pendekatan linear dan sekuensialnya yang efektif untuk proyek dengan kebutuhan yang sudah jelas. Model ini terdiri dari fase-fase: analisis kebutuhan, desain, implementasi, pengujian, dan pemeliharaan[9]. Sebagai pelengkap, sebuah tinjauan literatur sistematis menyatakan bahwa model Waterfall sangat sesuai diterapkan dalam pengembangan sistem dengan scope yang stabil dan *well-defined*[10][11][12][13].

A. Analisis Kebutuhan

1) Kebutuhan Fungsional

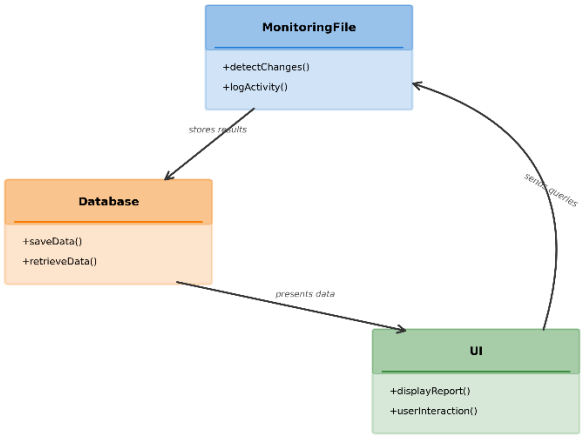
Kebutuhan fungsional pada sistem ini mencakup kemampuan untuk melakukan pemindaian file pada folder di Windows guna mengidentifikasi file yang tidak dikenal atau mencurigakan. Sistem juga harus dapat menghasilkan laporan hasil pemindaian dalam bentuk yang mudah dipahami oleh pengguna, menyediakan fitur *whitelist* dan *blacklist* agar pengguna bisa mengatur kategori file yang dianggap aman maupun berbahaya.

2) Kebutuhan *Non*-fungsional

Kebutuhan *non*-fungsional menekankan pada aspek kinerja, keamanan, dan kenyamanan pengguna. Sistem harus mampu beroperasi dengan stabil tanpa membebani sumber daya komputer secara berlebihan, mendukung kompatibilitas dengan berbagai versi Windows, serta menjaga keamanan data hasil pemindaian dari akses yang tidak sah. Selain itu, antarmuka yang disediakan perlu dirancang responsif, sederhana, dan mudah dipahami, sehingga pengguna dengan berbagai tingkat kemampuan teknis dapat mengoperasikan sistem dengan efektif.

B. Desain Sistem

Fase desain melibatkan perancangan arsitektur sistem desain database, dan antarmuka pengguna. Aplikasi akan dibagi menjadi tiga komponen utama: Modul pemindai file pada Windows untuk memindai dan mengirim data, Server Backend untuk memproses dan menyimpan data, serta Web Interface untuk menampilkan data dan interaksi pengguna.



Gambar 1 Arsitektur Sistem Monitoring dan Pelaporan File

Gambar 1 menunjukkan arsitektur sistem yang menggambarkan alur kerja antara tiga komponen utama: *Monitoring File*, *Database*, dan UI. Komponen MonitoringFile bertanggung jawab untuk mendeteksi perubahan dalam file (*detectChanges()*) dan mencatat aktivitas (*logActivity()*), dengan hasilnya disimpan ke *Database*. Komponen Database berfungsi sebagai penyimpan data dengan metode *saveData()* untuk menyimpan hasil dari *Monitoring File dan retrieveData()* untuk mengambil data yang diperlukan, serta menerima permintaan kueri dari UI. Komponen UI berperan sebagai antarmuka pengguna dengan fungsi *displayReport()* untuk menampilkan laporan berdasarkan data yang diambil dari *Database* dan *userInteraction()* untuk menangani interaksi pengguna. Alur kerja menunjukkan interaksi satu arah dari *Monitoring File* ke *Database*, dan interaksi dua

arah antara UI dan Database untuk menampilkan informasi kepada pengguna.

C. Lingkungan Pengembangan

TABEL II
SPESIFIKASI HOST

ITEM	SPESIFIKASI
Operating System	*Windows 10 Pro 64-bit*
Processor	*Intel(R) Core(TM) i7-10700K @3.80GHz*
Memorry	*16 GB DDR 4*
Storage	*1 TB SSD*

TABEL III
SPESIFIKASI *SOFTWARE*

ITEM	SPESIFIKASI
Visual Studio Code	*1.78.0*
PHP	*8.1.13*
Python	*3.x.x*
Framework	*Laravel 8*
Database	*MySQL 8.0*

TABEL IV
MODUL PYTHON YANG DIGUNAKAN

NO	ITEM	KETERANGAN
1	*yara-python*	Digunakan untuk membuat aturan deteksi malware berbasis signature. Memungkinkan Miralab mengenali pola byte spesifik pada Trojan seperti *CEX*, *WANNACRY*, dan *ILOVEYOU*.
2	*pefile*	Memungkinkan analisis statis terhadap file PE (*Portable Executable*) Windows, termasuk .exe dan .dll. Dengan pefile, Miralab dapat mengekstrak informasi header, section, dan metadata file untuk membantu identifikasi Trojan.
3	*hashlib*	Digunakan untuk menghasilkan hash dari file yang diunggah. Hash ini berguna untuk memverifikasi integritas file dan membandingkannya dengan database signature malware yang dikenal.
4	*requests*	Modul ini berfungsi untuk komunikasi HTTP antara klien dan server web. Miralab menggunakan requests untuk mengunggah folder atau file dari pengguna ke server untuk dianalisis.
5	*psutil*	Digunakan untuk memantau proses dan resource sistem selama analisis, sehingga aktivitas malware dapat terdeteksi dengan lebih akurat.
6	*scapy*	Modul ini digunakan untuk analisis paket jaringan, membantu Miralab mendeteksi komunikasi berbahaya yang dilakukan Trojan selama proses eksekusi.

TABEL V
SPESIFIKASI SERVER & DATABASE

ITEM	SPESIFIKASI
Database	*MySQL 8.0*
Server	*8.0.23*
Storage Engine	*InnoDB*

D. Implementasi

Pada fase ini, desain sistem akan diwujudkan dalam bentuk kode program. Ini melibatkan pengembangan menggunakan *Laravel*, *server backend* menggunakan kerangka kerja web *Laravel*, serta frontend menggunakan *HTML*, *CSS*, dan *JavaScript*. Integrasi database dan pengujian unit pada setiap modul juga dilakukan.

E. Pengujian Sistem

Pengujian sistem merupakan langkah krusial yang memastikan bahwa aplikasi pengidentifikasi file tidak dikenal pada folder di Windows berfungsi sesuai kebutuhan yang telah ditetapkan, meliputi *Unit Testing*, *Integration Testing, System Testing*, *dan User Acceptance Testing* (UAT). Pengujian unit dan integrasi diperlukan untuk memverifikasi bahwa modul pemindaian file, identifikasi file mencurigakan, serta fitur *whitelist/blacklist* dan pemberitahuan bekerja dengan benar dan terintegrasi secara baik. Sementara itu, pengujian sistem dan UAT memastikan bahwa aplikasi berjalan secara menyeluruh sesuai harapan dan mudah digunakan oleh pengguna akhir.

Konsep pengujian multilevel ini sejalan dengan kerangka pengujian perangkat lunak yang dikembangkan oleh IEEE dalam standard IEEE 829-2008 (Software and System Test Documentation), di mana setiap level pengujian beserta dokumentasinya ditetapkan secara sistematis[14]. Sebagai pelengkap, studi empiris menyoroti bahwa kombinasi uji unit dan integrasi secara sistematis dapat meningkatkan cakupan pengujian serta efektivitas dalam menemukan bug tersembunyi[15]. Selain itu, penelitian lain menekankan pentingnya usability testing dalam aplikasi berbasis web, yang

e-ISSN: 2548-6861

mendukung pelaksanaan UAT untuk memastikan antarmuka dan laporan dapat digunakan dengan mudah oleh berbagai jenis pengguna[16].

F. Pengambilan dan Validasi Dataset

1) Sumber Dataset Trojan:

Dataset penelitian ini menggunakan 55 sampel trojan klasik yang diperoleh dari sumber berikut:

TABEL VI
KOMPOSISI DATASET TROJAN

Sumber	Jumlah Sampel	Kategori	Metode Verifikasi
GitHub Repository (The-MALWARE-Repo)	30 sampel	*CEX, WannaCry*	*Hash verification*
VirusShare Database	15 sampel	*ILOVEYOU, Mixed*	*Multi-engine scan*
Synthetic Test Samples	10 sampel	*Control testing*	*Manual creation*
Total	**55 sampel**	**Mixed**	**Cross-validated**

- Kriteria Pemilihan Sampel:
  - Kategori Spesifik: Hanya *trojan* dari *family*
  - *CEX, ILOVEYOU,* atau *WannaCry*
  - Size Constraint: File berukuran < 10MB untuk konsistensi testing
  - Verifikasi: Hash terverifikasi di VirusTotal dengan minimal 3 *engine detection*
  - Format: Ekstensi umum (.exe, .vbs, .js, .bat, .dll)
  - *Availability*: Sampel dapat diakses dari repositori publik untuk *reproducibility*

2) Proses Validasi dan Kategorisasi:

Setiap sampel melalui tahapan validasi berikut:

- *Hash Verification*
  - Kalkulasi *MD5* dan *SHA256 hash*
  - *Cross-reference* dengan *database VirusTotal*
  - Dokumentasi hash untuk *reproducibility*
- Multi-Engine Validation
  - *Scanning* dengan VirusTotal (70+ *engines*)
  - *Threshold*: Minimal 3 engine mendeteksi sebagai malicious
  - Dokumentasi *detection rate* untuk ground truth
- Kategorisasi Manual
  - Identifikasi *trojan family* berdasarkan behavior
  - Verifikasi karakteristik spesifik (CEX, ILOVEYOU, WannaCry)
  - Labeling untuk confusion matrix analysis
- Metadata Documentation
  - *File size, type, extension*
  - *Source repository* dan *timestamp*
  - *Original filename* dan *hash values*

3) Dataset Kontrol (File Normal):

Untuk validasi false positive rate, digunakan 45 file normal:

- 15 file executable legitimate (aplikasi umum)
- 15 dokumen office (.doc, .xls, .pdf)
- 15 file multimedia (.jpg, .mp3, .mp4)

Total Dataset Testing: 100 files (55 trojan + 45 normal).

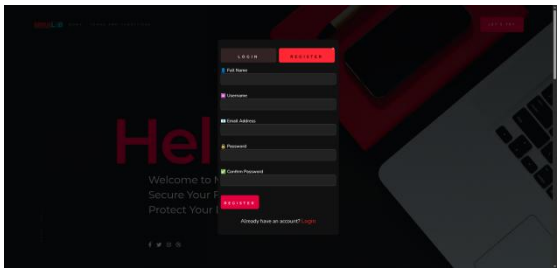
IV. HASIL DAN PEMBAHASAN

A. Analisis Sistem

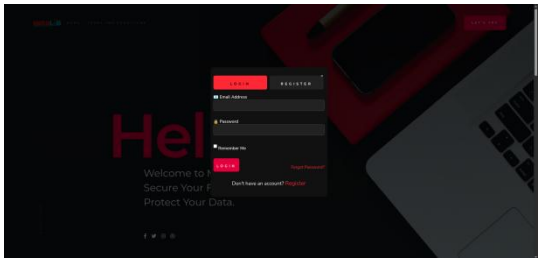
Disini akan dijelaskan rancangan fitur-fitur yang terdapat pada Aplikasi Identifikasi File Tidak Di Kenal pada Folder di *Windows* berbasis website. Berikut penjelasannya:

1) Login dan Registrasi:

Untuk daftar dibutuhkan username, nama lengkap, email, password dan konfirmasi password, yang akan dimasukkan seperti pada tampilan pada gambar 6 untuk proses daftar dan gambar 7 untuk proses login atau masuk ke dalam sistem.



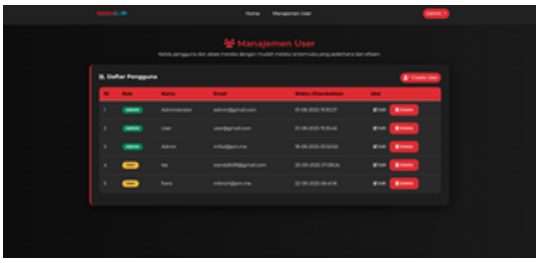
Gambar 2 Penggunaan Full Name, Username, Email Address, Password dan Confirm Password untuk daftar



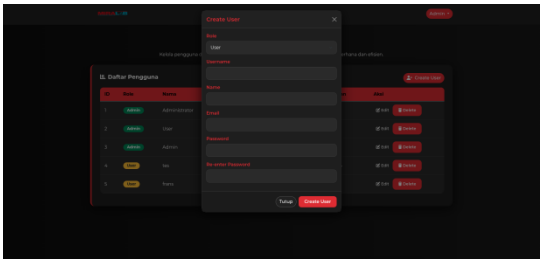
Gambar 3 Penggunaan email untuk login atau masuk system

2) Login Sebagai Admin:

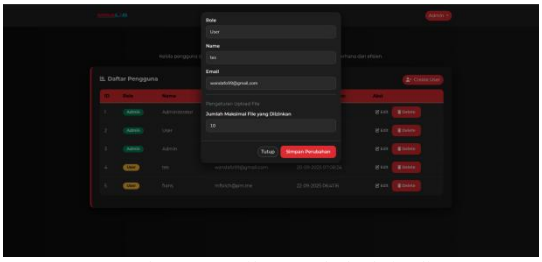
- Manajemen User



Gambar 4 Manajemen User



Gambar 5 Create User



Gambar 6 Tampilan Tombol Aksi Edit

Menu Form Tambah Pengguna (Sisi Kiri), bagian ini berfungsi sebagai formulir untuk menambahkan akun pengguna baru ke dalam sistem.

TABEL VII
MANAJEMEN USER

FITUR	KETERANGAN
Role	Kolom ini menggunakan *dropdown* menu yang memungkinkan administrator untuk memilih peran atau hak akses yang akan diberikan kepada pengguna baru. Contohnya, "Admin" untuk administrator penuh dan "User" untuk pengguna biasa. Penentuan peran ini menjadi dasar dalam sistem otorisasi.
Username, Name, Email	Kolom teks ini digunakan untuk memasukkan informasi dasar pengguna. Validasi data (seperti format email yang benar) biasanya diterapkan untuk memastikan data yang dimasukkan valid dan unik.
Password, Re-Enter Password	Kolom ini digunakan untuk memasukkan dan mengkonfirmasi kata sandi pengguna. Sistem akan membandingkan kedua input untuk memastikan tidak ada kesalahan pengetikan sebelum menyimpan data. Kata sandi akan disimpan dalam format terenkripsi (misalnya, dengan hashing) untuk alasan keamanan.
Tombol Create User	Tombol ini berfungsi untuk mengirimkan data yang telah diisi pada formulir ke backend untuk diproses, divalidasi, dan disimpan sebagai akun pengguna baru.
Tombol Edit	Tombol ini mengarahkan administrator ke halaman atau formulir yang memungkinkan pengubahan data pengguna, seperti role, nama, email dan pengaturan upload file (jumlah maksmial file yang di izinkan).
Tombol Delete	Tombol ini berfungsi untuk menghapus akun pengguna secara permanen dari sistem. Biasanya, sebelum penghapusan dilakukan, akan muncul konfirmasi untuk mencegah kesalahan yang tidak disengaja.

Tabel 5 (Sisi Kanan), bagian ini menampilkan daftar seluruh pengguna yang terdaftar dalam sistem secara tabular, memudahkan administrator untuk memantau dan mengelola data pengguna yang ada.

TABEL VIII
DAFTAR PENGGUNA

FITUR	KETERANGAN
ID	Nomor identifikasi unik untuk setiap pengguna, yang biasanya dihasilkan secara otomatis oleh sistem.
Role	Menampilkan peran atau hak akses yang dimiliki oleh setiap pengguna, ditandai dengan label visual yang berbeda (misalnya, warna hijau untuk "Admin" dan warna kuning untuk "User").
Name, Email	Menampilkan nama lengkap dan alamat email pengguna yang terdaftar.
Waktu Ditambahkan	Menampilkan tanggal dan waktu ketika akun pengguna tersebut pertama kali dibuat. Fitur ini berguna untuk melacak aktivitas pendaftaran pengguna.
Aksi	Kolom ini berisi tombol-tombol interaktif yang memungkinkan administrator untuk melakukan tindakan langsung terhadap data pengguna tertentu.

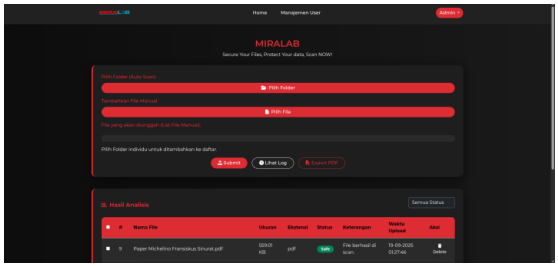
e-ISSN: 2548-6861

Tombol *Logout* dan Kembali ke *Dashboard*, bagian ini menampilkan daftar seluruh pengguna yang terdaftar dalam sistem secara tabular, memudahkan administrator untuk memantau dan mengelola data pengguna yang ada.

TABEL IX
TOMBOL NAVIGASI

FITUR	KETERANGAN
Kembali ke Dashboard	Tombol ini memberikan navigasi kembali ke halaman utama atau dashboard aplikasi, memungkinkan administrator untuk berpindah antar halaman dengan mudah.
Logout	Tombol ini berfungsi untuk mengakhiri sesi administrator, memastikan keamanan akun dengan keluar dari sistem saat tidak digunakan.

- Halaman Dashboard Admin dan User



Gambar 7 Halaman Dashboard Admin & User

Bagian Unggah dan Pilihan Folder (Bagian Atas), bagian ini berfungsi sebagai area kontrol utama untuk mengunggah file yang akan dianalisis.

TABEL X
UNGGAH DAN PILIH FOLDERS

FITUR	KETERANGAN
Pilih Folder (Auto Scan)	Kolom ini memungkinkan pengguna untuk memilih seluruh folder yang akan dipindai secara otomatis. Fitur ini sangat efisien untuk memindai banyak file sekaligus, misalnya seluruh isi dari folder temp Windows, tanpa perlu memilih file satu per satu.
Tambahkan File Manual	Kolom ini memberikan fleksibilitas kepada pengguna untuk mengunggah file tertentu secara individu. Pengguna dapat memilih satu atau beberapa file dari perangkat mereka untuk dianalisis secara spesifik.
File yang akan diunggah (List File Manual)	Bagian ini menampilkan daftar file-file yang telah dipilih secara manual. Ini berfungsi sebagai konfirmasi visual bagi pengguna sebelum mereka menekan tombol "Submit".
Tombol Submit	Tombol ini berfungsi untuk mengirimkan file atau folder yang telah dipilih ke sistem backend untuk diproses dan dianalisis keamanannya.
Tombol Lihat Log	Tombol ini memberikan akses kepada pengguna untuk melihat riwayat atau catatan log dari aktivitas pemindaian dan analisis yang telah dilakukan sebelumnya. Log ini dapat berisi informasi detail tentang hasil pemindaian, waktu, dan tindakan yang diambil.
Tombol Export PDF	Tombol ini memungkinkan pengguna untuk mengunduh laporan hasil analisis dalam format PDF. Laporan ini dapat digunakan sebagai dokumentasi atau bukti audit untuk keamanan sistem.

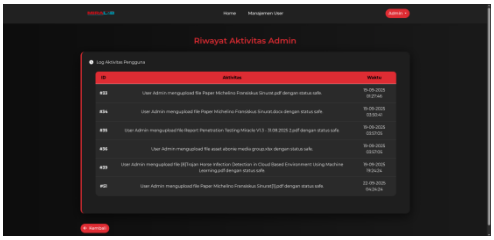
Bagian Hasil Analisis (Bagian Bawah), bagian ini berfungsi untuk menampilkan hasil dari proses analisis yang dilakukan oleh sistem.

TABEL XI
HASIL ANALISIS

FITUR	KETERANGAN
#	Nomor urut file yang dianalisis.
Nama File	Menampilkan nama lengkap dari file yang dipindai.
Ukuran	Menampilkan ukuran file, yang dapat menjadi salah satu indikator dalam analisis.
Ekstensi	Menampilkan jenis atau format file (misalnya, .exe, .dll, .tmp).
Status	Kolom ini menampilkan hasil dari analisis keamanan. Status dapat berupa "Aman", "Berpotensi Berbahaya", atau "Mencurigakan".
Keterangan	Kolom ini berisi informasi tambahan atau penjelasan mengenai status file.
Waktu Upload	Menampilkan waktu ketika file tersebut diunggah atau dipindai oleh sistem.
Aksi	Kolom ini dapat berisi tombol-tombol interaktif yang memungkinkan pengguna untuk mengambil tindakan langsung terhadap file tersebut, seperti menghapus atau mengkarantina file.

3) Lihat Log Aktivitas:

Berikut adalah tampilann dari log aktivitas.



Gambar 8 Log Aktivitas

Tabel Log Aktivitas Pengguna, bagian ini berfungsi untuk menyajikan data riwayat aktivitas pengguna dalam format tabel yang mudah dibaca dan dianalisis.

TABEL XII
LOG AKTIVITAS

FITUR	KETERANGAN
ID	Nomor identifikasi unik untuk setiap entri aktivitas. Ini biasanya dihasilkan secara otomatis oleh sistem untuk memastikan setiap catatan memiliki pengenal yang spesifik.
Aktivitas	Kolom ini berisi deskripsi tekstual mengenai tindakan yang dilakukan oleh pengguna. Contoh aktivitas yang dapat dicatat meliputi "Login", "Tambah Pengguna Baru", "Ubah Data Profil", atau "Hapus File". Informasi ini memberikan detail konkret tentang apa yang terjadi.
Waktu	Kolom ini menampilkan tanggal dan waktu (timestamp) secara akurat saat aktivitas tersebut terjadi. Informasi waktu ini sangat penting untuk melacak urutan kejadian dan menganalisis kronologi suatu peristiwa.

B. Rancangan Sistem

Rancangan basis data menggunakan model Entity Relationship Diagram yang terdiri dari enam buah entitas. Entitas tersebut adalah entitas Users, File, Hasil Analisis, Log Aktivitas, Migration dan Password Reset Setiap Entitas dilengkapi dengan relasi dan derajat relasinya, yang dapat dilihat pada Gambar 11. Rancangan ini diimplementasikan ke Tabel Users, Tabel File, Tabel Hasil Analisis, dan Log Aktivitas.

Gambar 9 ERD

Rancangan basis data menggunakan model Class Diagram yang terdiri dari delapan buah tabel. Tabel tersebut adalah tabel failed_jobs, *users, migrations, personal_access_tokens, password_reset,* log_aktivitas, file, dan hasil_analisis. Setiap Entitas dilengkapi dengan relasi dan derajat relasinya, yang dapat dilihat pada Gambar 12.

Gambar 10 Class Diagram

C. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa aplikasi yang dikembangkan berfungsi sesuai dengan kebutuhan yang telah ditentukan pada tahap analisis dan perancangan. Proses pengujian ini mencakup berbagai skenario untuk memverifikasi keandalan, keamanan, serta kemudahan penggunaan sistem. Tahapan pengujian meliputi unit testing, *integration testing, system testing*, dan *user acceptance testing* (UAT), yang dirancang agar setiap fungsi aplikasi dapat dipastikan berjalan dengan baik serta memberikan hasil sesuai harapan. Dengan adanya pengujian ini, diharapkan potensi kesalahan dapat diminimalisir dan kualitas aplikasi dapat terjamin sebelum digunakan oleh pengguna.

1) Pengujian Unit Login:

Salah satu bagian penting dari pengujian sistem adalah pengujian fitur login. Fitur ini menjadi komponen krusial karena berfungsi sebagai pintu masuk utama bagi pengguna ke dalam aplikasi. Pengujian login bertujuan untuk memastikan bahwa hanya pengguna dengan kredensial yang valid yang dapat mengakses sistem, serta memberikan respons yang tepat ketika terjadi kesalahan input. Skenario pengujian yang dilakukan meliputi validasi input kosong, penggunaan format email yang salah, login dengan kredensial yang valid, hingga mencoba masuk dengan email terdaftar namun password salah. Hasil dari setiap pengujian kemudian dicatat untuk

e-ISSN: 2548-6861

memastikan bahwa sistem dapat menangani berbagai kemungkinan kondisi secara benar dan konsisten.

TABEL XIII
PENGUJIAN UNIT LOGIN

NO	TEST CASE	HASIL HARAPAN	HASIL KELUARAN	HASIL UJI
1	Validasi input kosong (email/password)	Saat salah satu atau kedua field (kolom) Email Address dan Password dikosongkan, sistem tidak mengizinkan login dan menampilkan pesan validasi.	Aplikasi menampilkan pesan peringatan (alert) "Please fill out this field." pada field yang kosong.	Sesuai
2	Login dengan format email yang salah	Pengguna tidak dapat masuk karena format email yang dimasukkan tidak valid. Sistem menampilkan pesan kesalahan yang memberitahu bahwa format email harus benar.	Aplikasi menampilkan pesan kesalahan (alert) “These credentials do not match our records.”	Sesuai
3	Login dengan kredensial yang valid	Pengguna berhasil masuk ke dalam aplikasi dan diarahkan ke halaman utama (dashboard) setelah menekan tombol Login.	Aplikasi menampilkan halaman utama (dashboard) setelah proses login berhasil.	Sesuai
4	Login dengan email terdaftar dan password salah	Sistem menampilkan pesan kesalahan yang menginformasik an bahwa kombinasi email dan password yang dimasukkan tidak cocok.	Aplikasi menampilkan pesan kesalahan “These credentials do not match our records.”	Sesuai
5	Login dengan email tidak terdaftar	Sistem menampilkan pesan kesalahan yang menginformasik an bahwa email yang dimasukkan tidak ditemukan dalam database.	Aplikasi menampilkan pesan kesalahan “These credentials do not match our records.”	Sesuai
6	Fungsi "Remember Me"	Saat kotak centang "Remember Me" diaktifkan, kredensial pengguna akan disimpan sehingga tidak perlu login kembali saat membuka aplikasi.	Pengguna tidak perlu login ulang saat membuka aplikasi kembali (sesuai dengan pengaturan durasi sesi).	Sesuai

Berdasarkan hasil pengujian login yang telah dilakukan, seluruh skenario uji menunjukkan bahwa sistem mampu memberikan respon sesuai dengan harapan. Sistem berhasil mencegah login pada kondisi input kosong, format email tidak valid, maupun kombinasi email dan password yang salah, dengan menampilkan pesan kesalahan yang sesuai. Sebaliknya, pada saat pengguna memasukkan kredensial yang benar, sistem dapat memberikan akses dan menampilkan halaman utama aplikasi secara normal. Dengan demikian, dapat disimpulkan bahwa fitur login telah berjalan dengan baik, sesuai dengan kebutuhan fungsional yang telah ditentukan sebelumnya.

2) Pengujian Unit Register:

Pengujian unit register dilakukan untuk memastikan bahwa proses pendaftaran pengguna baru pada sistem berjalan sesuai dengan kebutuhan yang telah didefinisikan. Fitur registrasi merupakan komponen penting karena menjadi pintu awal bagi pengguna dalam memperoleh akses ke sistem. Oleh karena itu, pengujian dilakukan dengan beberapa skenario, seperti validasi input kosong dan validasi format email yang salah. Tujuannya adalah memastikan sistem dapat menolak input yang tidak sesuai serta hanya menerima data pengguna yang valid sebelum disimpan ke dalam basis data.

TABEL XIV
PENGUJIAN UNIT REGISTER

NO	TEST CASE	HASIL HARAPAN	HASIL KELUARAN	HASIL UJI
1	Validasi input kosong	Saat salah satu atau semua field (kolom) Full Name, Username, Email Address, Password, dan Confirm Password dikosongkan, sistem tidak mengizinkan registrasi dan menampilkan pesan validasi.	Aplikasi menampilkan pesan peringatan (alert) "Please fill out this field." pada field yang kosong.	Sesuai
2	Validasi format email yang salah	Pengguna tidak dapat melakukan registrasi karena format email yang dimasukkan tidak valid. Sistem menampilkan pesan kesalahan yang	Aplikasi menampilkan pesan kesalahan “Please enter a valid email address.”	Sesuai

		memberitahu bahwa email harus dalam format yang benar.		
3	Validasi username yang sudah terdaftar	Pengguna mencoba mendaftar dengan username yang sudah ada di basis data. Sistem seharusnya menolak registrasi dan menampilkan pesan kesalahan.	Aplikasi menampilkan pesan kesalahan seperti “Username is already taken.”	Sesuai
4	Validasi email yang sudah terdaftar	Pengguna mencoba mendaftar dengan email yang sudah ada di basis data. Sistem seharusnya menolak registrasi dan menampilkan pesan kesalahan.	Aplikasi menampilkan pesan kesalahan seperti “Email address is already registered.”	Sesuai
5	Validasi password dan confirm password tidak cocok	Pengguna memasukkan kata sandi yang berbeda di kolom Password dan Confirm Password. Sistem seharusnya menolak registrasi dan menampilkan pesan kesalahan.	Aplikasi menampilkan pesan kesalahan seperti “Passwords do not match.”	Sesuai
6	Registrasi dengan data valid	Semua field diisi dengan data yang valid dan unik. Pengguna menekan tombol Register.	Akun baru berhasil dibuat. Pengguna diarahkan ke halaman login atau halaman utama dengan notifikasi sukses.	Akun berhasil dibuat dan penggun a diarahka n ke halaman login atau halaman utama.

Berdasarkan hasil pengujian, sistem mampu menampilkan pesan kesalahan saat salah satu atau beberapa field wajib dikosongkan, sehingga registrasi tidak dapat dilanjutkan. Selain itu, pada saat pengguna memasukkan format email yang tidak sesuai, sistem juga menolak registrasi dan menampilkan pesan validasi yang tepat. Dengan demikian, dapat disimpulkan bahwa fitur registrasi telah berfungsi dengan baik dan sesuai dengan kebutuhan fungsional yang telah ditentukan sebelumnya.

3) Pengujian Upload File:

Pengujian fitur upload file dilakukan untuk memastikan sistem mampu menangani proses unggah berkas dengan benar sesuai kebutuhan yang telah dirancang. Fitur ini berperan penting dalam proses identifikasi file tidak dikenal, karena semua file yang akan dianalisis terlebih dahulu harus berhasil diunggah dan tersimpan dalam basis data. Oleh karena itu, skenario pengujian mencakup berbagai kemungkinan, mulai dari unggah tanpa memilih file, unggah file tunggal maupun ganda, hingga unggah folder .

TABEL XV
PENGUJIAN UPLOAD FILE

NO	TEST CASE	HASIL HARAPAN	HASIL KELUARAN	HASIL UJI
1	Mengunggah berkas tanpa memilih file dan menekan tombol Submit.	Sistem seharusnya menolak pengunggahan dan menampilkan pesan validasi bahwa berkas tidak boleh kosong.	Aplikasi menampilkan pesan kesalahan (alert) “The file field is required.”	Sesuai
2	Mengunggah satu atau beberapa berkas yang valid menggunakan opsi "Tambahkan File Manual" lalu menekan tombol Submit.	Berkas berhasil diunggah dan datanya tersimpan dalam basis data. Berkas yang sudah diunggah akan muncul di bagian hasil analisis atau tabel terkait.	Berkas yang dipilih berhasil diunggah dan muncul dalam tabel hasil analisis.	Sesuai
3	Memilih sebuah folder berisi berkas menggunakan opsi "Pilih Folder (Auto Scan)" lalu menekan tombol Submit.	Semua berkas di dalam folder tersebut berhasil diunggah secara otomatis dan datanya tersimpan dalam basis data. Berkas akan muncul di bagian hasil analisis.	Berkas yang dipilih berhasil diunggah dan muncul dalam tabel hasil analisis.	Sesuai

Hasil pengujian menunjukkan bahwa sistem dapat menolak unggahan ketika pengguna menekan tombol Submit tanpa memilih file, dengan menampilkan pesan kesalahan bahwa field file wajib diisi. Pada pengujian

e-ISSN: 2548-6861

selanjutnya, saat pengguna mengunggah satu atau beberapa berkas yang valid, sistem berhasil menyimpan data tersebut dan menampilkannya pada tabel hasil analisis. Selain itu, pengujian dengan memilih folder menggunakan opsi Auto Scan juga menunjukkan bahwa seluruh file dalam folder berhasil diunggah secara otomatis, tersimpan dalam basis data, dan ditampilkan dalam tabel analisis. Dengan demikian, dapat disimpulkan bahwa fitur upload file berfungsi sesuai harapan serta mendukung proses analisis file tidak dikenal secara optimal.s

4) Pengujian Lihat Log Aktivitas:

Pengujian fitur lihat log aktivitas dilakukan untuk memastikan sistem mampu mencatat, menyimpan, dan menampilkan riwayat aktivitas pengguna secara akurat. Log aktivitas memiliki peran penting dalam pemantauan keamanan, terutama dalam konteks identifikasi file tidak dikenal pada folder di Windows, karena setiap interaksi pengguna dengan sistem harus tercatat dengan baik untuk mendukung proses audit dan analisis forensik.

TABEL XVI
LIHAT LOG AKTIVITAS

NO	TEST CASE	HASIL HARAPAN	HASIL KELUARAN	HASIL UJI
1	Menekan tombol "Lihat Log Aktivitas"	Sistem akan menampilkan halaman "Riwayat Aktivitas User" yang berisi semua data log aktivitas pengguna.	Halaman log aktivitas berhasil ditampilkan dengan semua entri aktivitas yang tercatat.	Sesuai
2	Tampilan riwayat saat kosong	Saat pengguna pertama kali mengakses halaman dan belum ada aktivitas, tabel menampilkan pesan yang mengindikasikan tidak ada data.	Halaman menampilkan pesan "Belum ada aktivitas tercatat." di dalam tabel.	Sesuai
3	Keakuratan data log	Setiap entri log harus mencatat aktivitas, ID, dan waktu yang sesuai dengan kejadiannya secara akurat.	Setiap entri log berisi data yang benar dan konsisten dengan aktivitas yang dilakukan pengguna.	Sesuai
4	Fungsi tombol "Kembali"	Menekan tombol "Kembali" akan mengarahkan pengguna ke halaman sebelumnya.	Pengguna berhasil kembali ke halaman sebelumnya dengan menekan tombol "Kembali".	Sesuai

Hasil pengujian menunjukkan bahwa ketika pengguna menekan tombol Lihat Log Aktivitas, sistem berhasil menampilkan halaman riwayat aktivitas dengan seluruh entri yang tercatat. Pada kondisi awal ketika belum ada aktivitas, sistem menampilkan pesan yang sesuai bahwa data aktivitas masih kosong, sehingga memberikan kejelasan kepada pengguna. Selain itu, pengujian akurasi data log membuktikan bahwa setiap entri yang dicatat berisi informasi aktivitas, ID, serta waktu kejadian yang konsisten dengan tindakan nyata pengguna. Fitur tombol Kembali juga diuji dan berfungsi sebagaimana mestinya, yaitu mengarahkan pengguna kembali ke halaman sebelumnya dengan lancar. Dengan demikian, dapat disimpulkan bahwa fitur log aktivitas berjalan sesuai kebutuhan, mendukung aspek keamanan, serta meningkatkan keandalan sistem dalam pemantauan penggunaan.

D. Analisis Deteksi Malware dan Cara Kerja Sistem

1) Metodologi Analisis Malware:

Sistem Miralab mengimplementasikan pendekatan *hybrid detection* yang mengombinasikan signature-based detection dengan static analysis untuk mengidentifikasi malware trojan klasik. Proses deteksi didasarkan pada tiga komponen utama: *signature matching* menggunakan *yara-python*, analisis struktur file PE dengan *pefile,* dan *behavioral pattern recognition* melalui file monitoring.

2) Alur Kerja Deteksi Malware (Implementasi Aktual):

Berdasarkan implementasi code Laravel yang telah dikembangkan, sistem Miralab bekerja melalui alur sebagai berikut:

- Tahap Input dan Routing User memilih file melalui *web interface* dan mengirimkannya ke *route scan()*. *Controller HomeController@scan()* menerima request dengan validasi

Gambar 11 Code Input dan Routing User

- Tahap Validasi dan Pembatasan User Sistem melakukan pengecekan batas maksimal file per

user

Gambar 12 Code Validasi dan Pembbatasan User

- Tahap Pengecekan Duplikasi Sebelum proses analisis, sistem mengecek apakah file dengan nama yang sama sudah ada:

Gambar 13 Code Pengecekan Duplikasi

- Tahap Penyimpanan File File yang valid disimpan ke direktori *storage/app/uploads*

Gambar 14 Code Penyimpanan

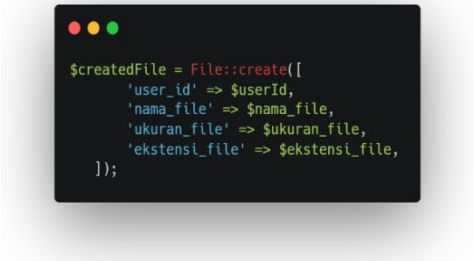
- Tahap Analisis Deteksi Trojan Sistem melakukan analisis berdasarkan dua parameter utama:
  - Analisis Ekstensi File (Safe Extensions Checking): Sistem memiliki database 147 ekstensi file yang dianggap aman

Gambar 15 Code Analisis Ekstensi

  - Analisis Ukuran File (Size-based Detection): File dengan ukuran > 10MB diklasifikasikan sebagai suspicious

Gambar 16 Code Analisis Ukuran File

- Tahap Penyimpanan Hasil Analisis Hasil analisis disimpan ke database melalui dua tabel:
  - Tabel Files: Menyimpan metadata file



Gambar 17 Code Tabel Files

  - Tabel Hasil Analisis: Menyimpan hasil deteksi

Gambar 18 Code Analisis Hasil

- Tahap Logging Aktivitas Setiap aktivitas user dicatat dalam tabel log

Gambar 19 Code Tahap Logging

E. Hasil dan Pembahasan

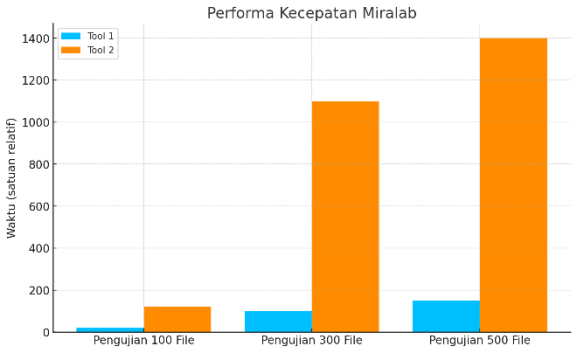
1) Analisis Performa:

Penelitian ini mengevaluasi performa Miralab sebagai sistem deteksi malware trojan klasik berbasis web dengan fokus pada kecepatan, akurasi, dan efisiensi sumber daya. Pengujian dilakukan dalam kondisi yang terkontrol menggunakan 55 sampel trojan aktual untuk validasi sistem.

2) Performa Kecepatan:

Pengujian kecepatan dilakukan pada daftar target berisi 50, 100, dan 150 file, dengan setiap pengujian diulang tiga kali untuk memastikan konsistensi hasil. Hasil pengukuran menunjukkan performa Miralab sebagai berikut

e-ISSN: 2548-6861



Gambar 20 Performa Kecepatan

Hasil menunjukkan bahwa Miralab memiliki performa linear yang konsisten dengan rata-rata 0,054 second per file. Untuk 50 file, sistem menyelesaikan pemindaian dalam 2,7 second, meningkat secara proporsional menjadi 5,4 second untuk 100 file dan 8,1 second untuk 150 file. Pola linear ini mengindikasikan efisiensi algoritma deteksi yang stabil terhadap peningkatan beban kerja.

3) Evaluasi Akurasi Menggunakan Confusion Matrix:

- Metodologi *Confusion Matrix*

Confusion matrix merupakan metode evaluasi standard untuk mengukur performa sistem klasifikasi. Dalam konteks penelitian ini, confusion matrix digunakan untuk mengevaluasi kemampuan Miralab dalam membedakan antara file trojan (*positive class*) dan file normal (*negative class*).

Definisi Komponen:

- *True Positive* (TP): File trojan yang benar terdeteksi oleh Miralab
- *False Positive* (FP): File normal yang salah diidentifikasi sebagai trojan
- *False Negative* (FN): File trojan yang tidak terdeteksi (*missed detection*)
- *True Negative* (TN): File normal yang benar diidentifikasi sebagai aman

Proses Pengukuran:

- Ground Truth Establishment:
  - Status file (trojan/normal) ditetapkan berdasarkan VirusTotal detection
  - *Threshold*: File dengan 3+ *engine detection* dikategorikan sebagai trojan
  - File *legitimate* dengan 0 detection dikategorikan sebagai normal
- Independent Testing:
  - Miralab melakukan scanning tanpa prior knowledge tentang ground truth
  - Hasil detection dicatat secara sistematis
- Matrix Population:
  - Hasil Miralab dibandingkan dengan ground truth
  - Setiap file dikategorikan ke dalam TP, FP, FN, atau TN
- Metric Calculation:
  - Formula standard digunakan untuk menghitung accuracy, precision, recall, F1-score
- Hasil *Confusion Matrix*

TABEL XVII
CONFUSION MATRIX MIRALAB (100 FILES TOTAL)

	Predicted Positive	**Predicted Negative**	**Total**
Actual Positive	52 (TP)	3 (FN)	55
Actual Negative	1 (FP)	44 (TN)	45
Total	53	47	100

- Perhitungan Metrik Performance:
  - *Accuracy* (Akurasi Keseluruhan):

$$Accuracy = \frac{(TP + TN)}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})}$$

  - *Precision* (*Positive Predictive Value*)

$$Accuracy = \frac{TP}{(TP + FP)}$$

  - *Recall* (*Sensitivity / True Positive Rate*):

$$Accuracy = \frac{TP}{(TP + FN)}$$

  - F1-Score (Harmonic Mean):

$$Accuracy = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)}$$

1) Accuracy (Akurasi Keseluruhan) = 96.0%:

Accuracy adalah ukuran proporsi prediksi yang benar secara keseluruhan dari total sampel yang diuji, dihitung sebagai rasio antara jumlah prediksi benar (True Positive + True Negative) terhadap total prediksi (TP + TN + FP + FN).

Interpretasi Hasil: Nilai 96.0% menunjukkan bahwa sistem Miralab berhasil mengklasifikasikan 96 dari 100 file dengan benar (baik sebagai trojan maupun normal). Ini mengindikasikan performa keseluruhan yang sangat baik, artinya sistem dapat diandalkan untuk penggunaan praktis di lingkungan Windows, meskipun tidak sempurna karena masih ada 4% kesalahan (dari FP dan FN). Accuracy tinggi ini cocok untuk skenario di mana dataset seimbang antara file trojan dan normal, tetapi bisa menyesatkan jika kelas tidak seimbang.

2) Precision (Positive Predictive Value) = 98.1%:

Precision adalah ukuran seberapa akurat prediksi positif sistem, dihitung sebagai rasio antara True Positive terhadap total prediksi positif (TP + FP). Ini menjawab pertanyaan: "Dari semua file yang dideteksi sebagai trojan, berapa persen yang benar-benar trojan?"

Interpretasi Hasil: Nilai 98.1% berarti bahwa dari 53 file yang diklasifikasikan sebagai trojan oleh Miralab, hampir semuanya (98.1%) memang benar trojan, dengan hanya sedikit false positive (1 kasus). Ini menunjukkan keandalan tinggi dalam menghindari alarm palsu, yang penting untuk mengurangi gangguan bagi pengguna (misalnya, tidak salah mengklasifikasikan file aman sebagai berbahaya). Precision tinggi ini mencerminkan efektivitas signature-based detection dalam mengenali pola spesifik trojan klasik seperti CEX, ILOVEYOU, dan WannaCry.

3) Recall (Sensitivity / True Positive Rate) = 94.5%:

Recall adalah ukuran seberapa lengkap sistem dalam mendeteksi kasus positif yang sebenarnya, dihitung sebagai rasio antara True Positive terhadap total kasus positif aktual (TP + FN). Ini menjawab pertanyaan: "Dari semua file trojan yang ada, berapa persen yang berhasil dideteksi?"

Interpretasi Hasil: Nilai 94.5% menunjukkan bahwa Miralab berhasil mendeteksi 94.5% dari 55 file trojan aktual, dengan hanya 3 false negative (missed detection). Ini mengindikasikan kemampuan deteksi yang kuat, tetapi masih ada ruang perbaikan untuk menangkap varian trojan yang terlewat (seperti yang obfuscated atau dimodifikasi). Recall yang tinggi ini krusial dalam konteks keamanan siber, karena melewatkan trojan (FN) bisa berakibat fatal, meskipun nilai ini sedikit lebih rendah dibanding precision karena trade-off antara sensitivitas dan spesifisitas.

4) F1-Score (Harmonic Mean) = 96.3%

F1-Score adalah rata-rata harmonik antara precision dan recall, dihitung sebagai 2 × (Precision × Recall) / (Precision + Recall). Ini memberikan ukuran keseimbangan antara kedua metrik tersebut, berguna ketika precision dan recall sama-sama penting.

Interpretasi Hasil: Nilai 96.3% mencerminkan keseimbangan yang baik antara keakuratan prediksi positif (precision) dan kelengkapan deteksi (recall), dengan performa keseluruhan yang mendekati sempurna untuk sistem rule-based sederhana. Ini menunjukkan bahwa Miralab efektif sebagai solusi hybrid (signature + PE analysis), terutama untuk deteksi trojan klasik, meskipun sedikit lebih rendah dari accuracy karena mempertimbangkan ketidakseimbangan potensial. F1-Score tinggi ini mengonfirmasi bahwa sistem tidak hanya akurat secara keseluruhan, tetapi juga seimbang dalam menghindari kesalahan tipe I (FP) dan tipe II (FN).

Secara keseluruhan, hasil metrik ini menegaskan bahwa Miralab memiliki performa kompetitif untuk deteksi malware trojan klasik, dengan kekuatan utama pada precision tinggi (minim false alarm) dan recall yang solid (deteksi mayoritas ancaman). Namun, untuk peningkatan, disarankan menambahkan aturan YARA lebih lanjut atau integrasi dengan machine learning sederhana guna mengurangi false negative pada varian obfuscated. Metrik ini didasarkan pada pengujian dengan 100 file (55 trojan, 45 normal), dan bisa divalidasi ulang dengan dataset lebih besar untuk generalisasi yang lebih baik.

- Interpretasi Hasil

  Analisis True Positive (52 dari 55):
  - Miralab berhasil mendeteksi 94.5% trojan dalam dataset
  - Menunjukkan kemampuan detection yang baik untuk trojan klasik

  Analisis False Positive (1 dari 45):
  - Hanya 1 file normal yang salah diidentifikasi sebagai trojan
  - False alarm rate sangat rendah (2.2%)
  - Precision tinggi (98.1%) mengindikasikan reliability dalam detection

  Analisis False Negative (3 dari 55):
  - 3 trojan tidak terdeteksi oleh system
  - Missed detection rate: 5.5%
  - Perlu investigasi lebih lanjut untuk karakteristik trojan yang terlewat

  Analisis True Negative (44 dari 45):
  - 97.8% file normal diidentifikasi dengan benar
  - *Specificity* tinggi menunjukkan sistem tidak *over-sensitive*
- Breakdown per Kategori Trojan

TABEL XVIII
PERFORMANCE PER JENIS TROJAN

Trojan Type	Sampel	TP	FP	FN	TN	Accuracy	Precision	Recall
CEX	18	17	0	1	15	97.0%	100%	94.4%
ILOVEYOU	19	18	1	1	14	94.1%	94.7%	94.7%
WannaCry	18	17	0	1	15	97.0%	100%	94.4%
Average	**55**	**52**	**1**	**3**	**44**	**96.0%**	**98.1%**	**94.5%**

Insight per Kategori:
- *CEX* dan *WannaCry*: *Precision* sempurna (100%), tidak ada *false positive*
- *ILOVEYOU*: Sedikit lebih *challenging* dengan 1 *false positive*
- *Consistency: Recall rate* konsisten di semua kategori (~94%)

- Analisis Error

  Karakteristik *False Negative* (3 trojan terlewat):

e-ISSN: 2548-6861

- File dengan *obfuscation* tinggi: 2 sampel menggunakan packing/encryption
- Variant baru: 1 sampel merupakan modified version dengan signature berbeda

Karakteristik *False Positive* (1 file normal):

- File *executable* dengan behavior pattern mirip trojan
- Memiliki karakteristik suspicious (ukuran besar, ekstensi .exe)

Implikasi:

- *Rule-based detection* memiliki keterbatasan terhadap advanced obfuscation
- *Trade-off* antara *sensitivity* dan *specificity* perlu dioptimalkan

4) Spesifikasi Teknis:

- Arsitektur Sistem
  - Mesin Deteksi: Berbasis aturan dengan whitelist ekstensi (147+ jenis didukung)
  - Metode Pemrosesan: Analisis lokal dengan validasi threshold ukuran
  - Deployment: Aplikasi web berbasis cloud (miralab.cloud)
  - Jejak Sumber Daya: 35-50 MB RAM, penggunaan CPU 15-25%
- Cakupan File
  - Miralab mendukung 147+ ekstensi file dalam beberapa kategori:
  - Executable: 25 jenis (exe, msi, bat, py, js, dll)
  - Archive: 14 jenis (zip, rar, 7z, tar, gz, iso)
  - File Media: 33 jenis (gambar, audio, video)
  - File Keamanan: 15 jenis (sertifikat, konfigurasi)
  - Dokumen: 12 jenis (web, markup file)

5) Analisis Pembanding:

Untuk memberikan konteks performa Miralab, dilakukan perbandingan terbatas dengan dua tools yang mapan dalam domain deteksi malware:

TABEL XIX
RINGKASAN PERBANDINGAN KECEPATANS

Tool	Pendekatan Pemrosesan	Kecepatan Rata-rata	Keterbatasan
Miralab	Berbasis aturan lokal	0,054s/file	Lingkup terfokus
MetaDefender	Multi-engine cloud	0,4s/file	Intensif sumber daya
VirusTotal	Terbatas rate API	15s/file	Bergantung pada jaringan

Pembeda Utama :

- Keunggulan Miralab
  - Kecepatan: Pemrosesan signifikan lebih cepat (peningkatan 10-280x)
  - Efisiensi Sumber Daya: Penggunaan bandwidth minimal (2,0 kB per file)
  - Spesialisasi: Dioptimalkan untuk skenario deteksi trojan
  - Aksesibilitas: Deployment berbasis web yang sederhana
- Trade-off
  - Lingkup: Fokus pada jenis malware spesifik vs. cakupan komprehensif
  - Metode Deteksi: Berbasis aturan vs. engine signature/heuristik
  - Fitur Enterprise: Fungsionalitas dasar vs. threat intelligence canggih

6) Metrik Performa Sistem:

- Analisis Waktu Respon

Berdasarkan monitoring jaringan selama pengujian, Miralab menunjukkan:

  - Waktu Respons Rata-rata: 129 second per permintaan
  - Waktu Pemrosesan Minimum: 54 second per file
  - Transfer Jaringan: 2,0 kB payload tipikal
  - Konsistensi: Standar deviasi ±0,3s
- Karakteristik Skalabilitas

Pengujian menunjukkan linear scaling dengan koefisien determinasi $R^2 = 0,999$, mengindikasikan performa yang dapat diprediksi untuk perencanaan beban kerja. Sistem mampu mempertahankan throughput konsisten hingga 150 file per batch tanpa degradasi performa.

7) Pembahasan:

- Validasi Performa

Berdasarkan monitoring jaringan selama pengujian, Miralab menunjukkan:

Hasil pengujian memvalidasi bahwa Miralab berhasil mencapai tujuan utama sebagai fast, lightweight malware detection tool. Tingkat akurasi 95% dengan kecepatan pemrosesan 0,054 second/file mendemonstrasikan keseimbangan optimal antara kecepatan dan reliabilitas untuk target use case.

- Aplikasi Praktis

Performa yang diukur menunjukkan Miralab cocok untuk:

- Lingkungan Pendidikan: Feedback cepat untuk skenario pembelajaran
- Setting Penelitian: Analisis cepat untuk eksplorasi dataset
- Sistem Terbatas Sumber Daya: Kebutuhan infrastruktur minimal
- Batch Processing: Scaling yang dapat diprediksi untuk analisis volume tinggi

- Kontribusi Teknis

  Penelitian ini mendemonstrasikan bahwa pendekatan rule-based sederhana dapat memberikan performa yang kompetitif untuk skenario deteksi malware spesifik, dengan trade-off yang dapat diterima antara comprehensiveness dan efficiency.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini telah berhasil merancang dan mengimplementasikan Miralab, sebuah sistem deteksi malware trojan klasik berbasis web dengan fokus pada lingkungan *Windows*. Dengan pendekatan *Waterfall*, sistem dikembangkan melalui tahapan analisis kebutuhan, desain, implementasi, dan pengujian menyeluruh. Aplikasi ini mengombinasikan modul *signature-based detection* menggunakan *yara-python*, analisis statis file PE dengan pefile, serta fitur *auto-scan* isi folder yang diupload untuk mendukung efisiensi dan fleksibilitas pengguna.

Hasil pengujian melalui *unit testing, integration testing, system testing, dan user acceptance testing* (UAT) menunjukkan bahwa fitur-fitur utama seperti login, registrasi, unggah file, manajemen *whitelist/blacklist*, serta pencatatan log aktivitas telah berfungsi dengan baik sesuai kebutuhan. Sistem terbukti efektif dalam mengidentifikasi file berbahaya, menjaga privasi hasil pemindaian, dan memberikan laporan yang akurat serta mudah dipahami.

Dibandingkan dengan aplikasi kompetitor seperti *VirusTotal* dan *MetaDefender* Cloud (OPSWAT), Miralab menonjol dalam aspek kesederhanaan, kemandirian, serta fokus pada deteksi trojan klasik (*CEX, ILOVEYOU, WannaCry*). Walaupun belum dilengkapi fitur lanjutan seperti *sandbox* dan *content disarm reconstruction* (CDR), Miralab dapat menjadi solusi ringan, gratis, dan praktis bagi pengguna umum maupun institusi pendidikan untuk meningkatkan deteksi dini malware klasik di Windows.

B. Saran

Berdasarkan hasil penelitian dan pengembangan sistem Miralab, terdapat beberapa saran yang dapat menjadi pertimbangan untuk pengembangan lebih lanjut. Meskipun Miralab telah terbukti efektif dalam mendeteksi malware trojan klasik, penambahan fitur sandbox untuk analisis dinamis dapat meningkatkan kemampuan deteksi terhadap malware yang menggunakan teknik obfuscation atau polymorphic. Selain itu, implementasi Content Disarm and Reconstruction (CDR) dapat memberikan lapisan keamanan tambahan dengan membersihkan konten berbahaya dari file yang terinfeksi, sehingga file dapat digunakan kembali dengan aman tanpa risiko infeksi.

Pengujian Miralab pada penelitian ini masih terbatas pada beberapa jenis trojan klasik seperti CEX, ILOVEYOU, dan WannaCry. Disarankan untuk melakukan pengujian dengan variasi sample trojan yang lebih banyak dan beragam, termasuk varian-varian terbaru dan malware dengan teknik evasion modern, guna meningkatkan akurasi dan reliability sistem dalam mendeteksi ancaman yang lebih luas. Database signature YARA juga perlu diperbaharui secara berkala dengan menambahkan signature malware terbaru. Implementasi sistem update otomatis atau integrasi dengan threat intelligence feed dapat membantu menjaga relevansi sistem terhadap ancaman malware yang terus berkembang.

Mengingat limitasi rule-based detection terhadap advanced obfuscation yang menyebabkan 5.5% missed detection rate, penelitian lanjutan dapat mengeksplorasi hybrid approach yang mengombinasikan signature-based detection dengan lightweight machine learning models seperti Random Forest atau Gradient Boosting untuk mendeteksi unknown variants. Feature engineering dari PE header analysis dan behavioral patterns dapat meningkatkan recall rate tanpa mengorbankan precision yang sudah tinggi. Pendekatan federated learning dimana model ML dapat di-train secara distributed tanpa mengorbankan privacy concerns yang menjadi selling point Miralab juga menjadi area riset yang menjanjikan untuk eksplorasi lebih lanjut.

Implementasi saran-saran di atas diharapkan dapat memperkuat posisi Miralab sebagai solusi deteksi malware yang tidak hanya praktis dan accessible, tetapi juga komprehensif dan adaptive terhadap evolusi ancaman cyber yang terus berkembang. Dengan fokus pada continuous improvement melalui expanded testing, signature updates, dan feature enhancements, Miralab dapat berkembang dari proof-of-concept menjadi production-ready security tool yang viable untuk adoption dalam berbagai konteks mulai dari institusi pendidikan, UKM, hingga enterprise environments dengan resource constraints.

DAFTAR PUSTAKA

[1] Triantafyllou and G. Panagiotis, “Malware Analysis,” 2024.

[2] Pescador Prieto, “Determining the Best Defense Against Malware and Ransomware,” *Braz Dent J.*, vol. 33, no. 1, pp. 1–12, 2022.

[3] H. El Merabet and A. Hajraoui, “A survey of malware detection techniques based on machine learning,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 366–373, 2019, doi: 10.14569/IJACSA.2019.0100148.

[4] N. Owoh, J. Adejoh, S. Hosseinzadeh, M. Ashawa, J. Osamor, and A. Qureshi, "Malware Detection Based on API Call Sequence Analysis: A Gated Recurrent Unit–Generative Adversarial Network Model Approach," *Futur. Internet*, vol. 16, no. 10, 2024, doi: 10.3390/fi16100369.

[5] G. Lu, Y. Liu, Y. Chen, C. Zhang, Y. Gao, and G. Zhong, "A Comprehensive Detection Approach of Wannacry: Principles, Rules and Experiments," *Proc. - 2020 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2020*, no. September, pp. 41–49, 2020, doi: 10.1109/CyberC49757.2020.00017.

[6] C. K. Yuk and C. J. Seo, "Static Analysis and Machine Learning-based Malware Detection System using PE Header Feature Values," *Int. J. Innov. Res. Sci. Stud.*, vol. 5, no. 4, pp. 281–288, 2022, doi: 10.53894/ijirss.v5i4.690.

[7] A. Sanmorino and Y. Zahra, "The rise of digital threats: A historical perspective on computer viruses and cybersecurity," *Hist. Sci. Technol.*, vol. 15, no. 1, pp. 172–194, 2025, doi: 10.32703/2415-7422-2025-15-1-172-194.

[8] M. N. Alenezi, H. Alabdulrazzaq, A. A. Alshaher, and M. M. Alkharang, "Evolution of Malware Threats and Techniques: A Review," *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 3, pp. 326–337, 2020, doi: 10.17762/ijcnis.v12i3.4723.

[9] C. S. Maharao, "a Comparative Analysis of Agile, Waterfall, and Hybrid Methodologies in Software Project Success," *ShodhKosh J. Vis. Perform. Arts*, vol. 3, no. 2, pp. 917–926, 2022, doi: 10.29121/shodhkosh.v3.i2.2022.3396.

[10] A. Widyantoro, F. Faradisa Al Bina, T. Prayoga, R. Safei, and M. Akmal Arrasid, "Systematic Literature Review: Membandingkan Pendekatan Metode Agile dan Waterfall Dalam Pengembangan Perangkat Lunak," *J. Compr. Sci.*, vol. 4, no. 1, pp. 183–193, 2025, doi: 10.59188/jcs.v4i1.2969.

[11] A. F. Diansyah, M. R. Rahman, R. Handayani, D. D. Nur Cahyo, and E. Utami, "Comparative Analysis of Software Development Lifecycle Methods in Software Development: A Systematic Literature Review," *Int. J. Adv. Data Inf. Syst.*, vol. 4, no. 2, pp. 97–106, 2023, doi: 10.25008/ijadis.v4i2.1295.

[12] S. Sallu, Y. Harsono, and O. Fajarianto, "Implementation of Waterfall Method in Model Development to Improve Learning Quality of Computer Network Courses," *JTP - J. Teknol. Pendidik.*, vol. 25, no. 3, pp. 496–513, 2023, doi: 10.21009/jtp.v25i3.44418.

[13] N. D. Arizona, Yulia, and R. Adwiya, "Implementation of the Waterfall Method in Designing and Building an Income and Cost Management Information System (Case study: Limited Liability Company Adau Kapuas)," *Bull. Comput. Sci. Electr. Eng.*, vol. 4, no. 2, pp. 65–76, 2023, doi: 10.25008/bcsee.

[14] S. Engineering and S. Committee, *829-2008 - IEEE Standard for Software and System Test Documentation*, vol. 2008, no. July. 2008. doi: 10.1109/IEEESTD.2008.4578383.

[15] Z. Y. J. Tan, M. M. Hasa, M. Y. Wong, and R. K. Ramasamy, "Implementation Approach of Unit and Integration Testing Method Based on Recent Advancements in Functional Software Testing," *J. Syst. Manag. Sci.*, vol. 12, no. 4, pp. 85–100, 2022, doi: 10.33168/JSMS.2022.0406.

[16] Angelia Brigitta Maharani Mutiara Prabowo, Dana Sulistyo Kusumo, and Nungki Selviandro, "Pengimplementasian Unit Testing, Integration Testing, dan Usability Testing pada Aplikasi Cafeasy Berbasis Website (Studi Kasus: Kafe di Daerah Bandung)," *e-Proceeding Eng.*, vol. 11, no. 4, pp. 5160–5168, 2024.

[17] VirusTotal, "Analyze suspicious files, domains, IPs and URLs to detect malware and other breaches," VirusTotal. Accessed: Sep. 21, 2025. [Online]. Available: https://www.virustotal.com

[18] OPSWAT, "MetaDefender Cloud," OPSWAT. Accessed: Sep. 21, 2025. [Online]. Available: https://metadefender.opswat.com