Jurnal Ilmiah
Teknologi Sistem
Informasi

Jitsi

Vol. 7 (2026) No. 2

https://jurnal-itsi.org/index.php/jitsi; E-ISSN: 2722-4600; ISSN: 2722-4619
DOI: 10.62527/jitsi.7.2.602

Deteksi Malware pada File Executable Menggunakan Machine Learning Random Forest

M. Cakra Adhana[#], Alde Alanda[#], Hidra Amnur[#], Andre Febrian Kasmar[#]

[#] *Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia*
E-mail:aldealanda[at]pnp.ac.id, hidraamnur[at]gmail.com, andrefebrian[at]pnp.ac.id

A B S T R A C T S

The pervasive expansion of digital infrastructure has triggered an exponential surge in cyber threats, with malicious software (malware) posing a paramount risk to information security systems. Traditional signature-based and heuristic detection methods demonstrate severe limitations in mitigating zero-day exploits and multi-variant obfuscated malware due to their rigid dependency on existing signature repositories and susceptibility to high false-positive rates. To transcend these boundaries, this study introduces an adaptive and robust static detection framework for Portable Executable (PE) files leveraging the ensemble machine learning technique of Random Forest. Utilizing a structured dataset comprising PE files harvested from public malware repositories including Malware Bazaar alongside verified benign applications, static analysis was performed without code execution to preserve environment safety. A total of 75 distinctive structural features spanning COFF headers, section characteristics, data directories, and configuration markers were systematically extracted using the Python pefile library. The model was trained using an 80:20 data split ratio. Experimental evaluation achieved an exceptional internal generalization capability with an Out-of-Bag (OOB) score of 97.43%. Independent validation on a test suite of 332 unseen files yielded a balanced confusion matrix comprising 160 True Positives, 164 True Negatives, 5 False Positives, and 3 False Negatives, establishing a high precision, recall, and F1-score of approximately 98%. Feature importance analysis highlighted that parameters such as MajorOperatingSystemVersion, MajorSubsystemVersion, and DllCharacteristics serve as critical discriminators. Finally, the optimized predictive model was integrated into a web-accessible application architecture powered by Flask and MySQL to facilitate user-driven file uploading and real-time inference reporting, offering an scalable complementary defense layer for modern cybersecurity ecosystems.

Manuscript received Jun 22,2026; revised Jun 26 2026. accepted Jun 28, 2026 Date of publication Jun 30, 2026. International Journal, JITSI : Jurnal Ilmiah Teknologi Sistem Informasi licensed under a Creative Commons Attribution-Share Alike 4.0 International License

A B S T R A K

Ekspansi infrastruktur digital yang meluas telah memicu peningkatan eksponensial dalam ancaman siber, dengan perangkat lunak berbahaya (malware) menimbulkan risiko utama bagi sistem keamanan informasi. Metode deteksi berbasis tanda tangan dan heuristik tradisional menunjukkan keterbatasan serius dalam mengurangi eksploitasi zero-day dan malware multi-varian yang dikaburkan karena ketergantungannya yang kaku pada repositori tanda tangan yang ada dan kerentanannya terhadap tingkat false-positive yang tinggi. Untuk mengatasi keterbatasan ini, studi ini memperkenalkan kerangka kerja deteksi statis adaptif dan tangguh untuk file Portable Executable (PE) dengan memanfaatkan teknik pembelajaran mesin ensemble Random Forest. Dengan menggunakan dataset terstruktur yang terdiri dari file PE yang diambil dari repositori malware publik termasuk Malware Bazaar bersama dengan aplikasi benign yang terverifikasi,

analisis statis dilakukan tanpa eksekusi kode untuk menjaga keamanan lingkungan. Sebanyak 75 fitur struktural yang berbeda yang mencakup header COFF, karakteristik bagian, direktori data, dan penanda konfigurasi diekstrak secara sistematis menggunakan pustaka Python pefile. Model dilatih menggunakan rasio pembagian data 80:20. Evaluasi eksperimental mencapai kemampuan generalisasi internal yang luar biasa dengan skor Out-of-Bag (OOB) sebesar 97,43%. Validasi independen pada rangkaian uji yang terdiri dari 332 file yang belum pernah dilihat sebelumnya menghasilkan matriks kebingungan yang seimbang yang terdiri dari 160 True Positive, 164 True Negative, 5 False Positive, dan 3 False Negative, yang menetapkan presisi, recall, dan skor F1 yang tinggi sekitar 98%. Analisis kepentingan fitur menyoroti bahwa parameter seperti MajorOperatingSystem Version, MajorSubsystemVersion, dan DllCharacteristics berfungsi sebagai diskriminator penting. Terakhir, model prediktif yang dioptimalkan diintegrasikan ke dalam arsitektur aplikasi yang dapat diakses melalui web yang didukung oleh Flask dan MySQL untuk memfasilitasi pengunggahan file yang digerakkan pengguna dan pelaporan inferensi waktu nyata, menawarkan lapisan pertahanan pelengkap yang skalabel untuk ekosistem keamanan siber modern.

Keywords / Kata Kunci — ***Cybersecurity, Malware Detection, Portable Executable, Machine Learning, Random Forest, Static Analysis.***

CORRESPONDING AUTHOR

Alde Alanda
Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia
Email: aldealanda[at]pnp.ac.id

1. PENDAHULUAN

Akselerasi kemajuan teknologi informasi telah merevolusi berbagai sendi kehidupan manusia, menggeser ekosistem konvensional menuju digitalisasi terintegrasi. Namun, perluasan permukaan digital ini berbanding lurus dengan peningkatan kompleksitas dan frekuensi ancaman keamanan siber (cyber threats). Di antara berbagai instrumen serangan, perangkat lunak berbahaya (malware) tetap menjadi ancaman paling dominan, destruktif, dan persisten yang dirancang khusus untuk mengompromikan kerahasiaan, integritas, dan ketersediaan aset informasi tanpa otorisasi pemilik sistem. Berdasarkan data statistik global dari lembaga keamanan siber terkemuka, volume varian malware baru meningkat secara masif setiap harinya, menegaskan bahwa lanskap ancaman siber bersifat sangat dinamis.

Metode pertahanan konvensional yang diimplementasikan pada perangkat lunak antivirus tradisional umumnya bertumpu pada teknik analisis berbasis tanda tangan (signature-based) dan heuristik. Pendekatan signature-based mendeteksi ancaman dengan mencocokkan nilai hash file terhadap basis data tanda tangan yang telah diketahui sebelumnya. Meskipun metode ini sangat efektif untuk mendeteksi malware lama, ia memiliki kelemahan mendasar yaitu ketidakmampuan total dalam menangani serangan hari-nol (zero-day attacks) dan malware yang telah mengalami manipulasi struktural seperti polimorfisme atau metamorfisme. Di sisi lain, metode heuristik mencoba mengidentifikasi malware berdasarkan kesamaan struktural atau potongan instruksi mencurigakan. Kendati demikian, heuristik sering kali dihadapkan pada dilema konfigurasi sensitivitas; sensitivitas yang terlalu tinggi akan memicu tingginya angka false positive (mengidentifikasi program aman sebagai ancaman), yang mengganggu produktivitas operasional.

Guna mengatasi keterbatasan bawaan dari metode konvensional tersebut, pendekatan berbasis kecerdasan buatan, khususnya machine learning (pembelajaran mesin), hadir sebagai paradigma baru yang cerdas dan adaptif. Melalui pemanfaatan teknik analisis statis—yaitu memeriksa struktur internal file tanpa mengeksekusi kode program di dalam lingkungan tiruan (sandbox)—pembelajaran mesin mampu mengenali pola abstrak dan anomali tersembunyi yang membedakan berkas berbahaya dengan berkas aman. Di antara spektrum algoritma klasifikasi terawasi (supervised learning), algoritma Random Forest (RF) terbukti memiliki kapabilitas superior untuk domain keamanan siber. Sebagai metode berbasis ensemble learning, Random Forest mengombinasikan kumpulan pohon keputusan (decision trees) independen guna mereduksi risiko overfitting serta meningkatkan stabilitas dan akurasi prediksi secara signifikan.

Penelitian ini memfokuskan pada perancangan, pengembangan, dan evaluasi model klasifikasi otomatis berbasis Random Forest untuk mengidentifikasi malware pada file berformat Portable Executable (PE) yang merupakan format file eksekusi standar (.exe dan .dll) pada sistem operasi Windows. Dengan mengekstrak 75 fitur struktural dari file PE menggunakan pustaka analisis tingkat rendah, model yang dibangun dievaluasi secara matematis untuk membuktikan keandalannya. Lebih jauh lagi, untuk menjembatani kesenjangan antara model teoretis dan implementasi praktis, penelitian ini mengintegrasikan model prediktif terbaik ke dalam arsitektur

aplikasi berbasis website interaktif menggunakan framework Flask dan basis data MySQL lokal. Hasil penelitian ini diharapkan dapat memberikan kontribusi ilmiah yang berarti bagi khazanah pertahanan siber modern yang adaptif terhadap evolusi malware.

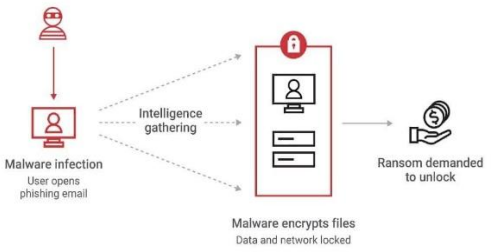
2. TINJAUAN PUSTAKA

Malware merupakan sekumpulan program yang yang dikembangkan secara tidak etis untuk mengakses, mengubah, atau mengancam sistem komputer atau jaringan komputer[13]. Malware terdiri dari berbagai jenis perangkat lunak berbahaya seperti

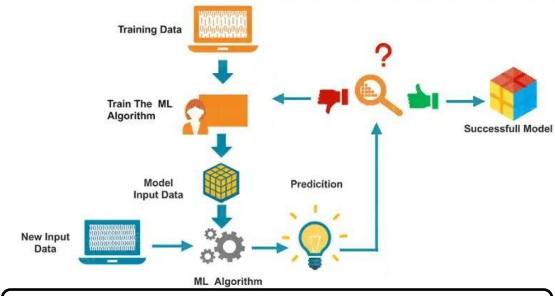
virus, worm, trojan, ransomware, spyware, dan adware, yang bertujuan untuk merusak, mencuri, atau mengganggu sistem komputer dan data pengguna. Malware sangat berbahaya karena dapat melakukan penyusupan dan mendapatkan akses secara tidak sah, serta malware juga dapat menyebabkan kerusakan pada sistem, data, dan keamanan jaringan.

Siklus hidup malware umumnya meliputi tahap infeksi, penyebaran, dan kerusakan. Tahap infeksi dimulai ketika malware masuk ke sistem, biasanya melalui file yang terinfeksi, email phishing, atau eksploitasi kerentanan sistem. Setelah berhasil menginfeksi, malware akan menyebar ke sistem lain atau komponen sistem yang sama. Tahap kerusakan terjadi ketika malware menjalankan fungsi jahatnya, seperti merusak file, mencuri data, atau mengganggu kinerja sistem.

Proses serangan malware dapat dilihat pada Gambar 1. Umumnya, serangan dimulai dari tahap infeksi yang terjadi ketika pengguna membuka email phishing. Setelah malware berhasil masuk, malware akan melakukan pengumpulan intelijen, mengenkripsi file, serta mengunci data dan jaringan. Akibatnya penyerang kemudian menuntut tebusan agar data tersebut dapat diakses kembali.



GAMBAR 1. Proses Serangan Malware



GAMBAR 2. Proses Machine Learning

Studi mengenai pemanfaatan *machine learning* untuk deteksi malware telah menjadi fokus riset intensif dalam beberapa tahun terakhir. Tjahjadi dkk. menerapkan algoritma *Random Forest* pada dataset malware umum dan berhasil mencatatkan akurasi mengesankan sebesar 99% dengan F1-score berkisar antara 0,98 hingga 0,99. Di ranah perangkat bergerak, Hadiprakoso dkk. melakukan analisis statis untuk mendeteksi malware Android dengan membandingkan algoritma SVM, Naive Bayes, Decision Tree, dan K-NN, yang menghasilkan akurasi optimal sebesar 96,94%. Sitorus dkk. memperluas komparasi tersebut pada dataset hak akses (*permission*) Android dan menemukan bahwa *Random Forest* mengungguli algoritma SVM dengan raihan akurasi 98,99% berbanding 06,23%

Metode *stacking ensemble* yang mengombinasikan *Neural Network*, *Random Forest*, dan kNN, yang menghasilkan akurasi sebesar 98,7%. Meskipun metode *stacking* menunjukkan performa sedikit lebih tinggi, eksekusi algoritma *Random Forest* secara individu tetap menunjukkan efisiensi waktu komputasi yang jauh lebih baik dengan akurasi mandiri yang kompetitif sebesar 98,2%. Abdussalam dan Rahmatulloh mengonfirmasi bahwa dalam pengujian menggunakan uji statistik, algoritma berbasis pohon keputusan (*decision tree*) dan jaringan saraf tiruan memberikan kestabilan tinggi dalam memproses karakteristik data biner.

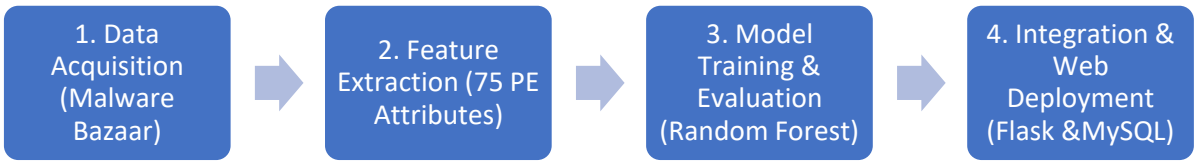
Format berkas *Portable Executable* (PE) merupakan struktur standar yang digunakan oleh sistem operasi keluarga Windows NT untuk berkas-berkas eksekusi, objek kode, dan *Dynamic Link Libraries* (DLL). Struktur internal PE terdiri dari beberapa komponen krusial, meliputi *DOS Header*, *PE Header* (yang membawahi *COFF File Header* dan *Optional Header*), *Data Directories*, serta *Section Table* (seperti `.text` untuk kode instruksi, `.data` untuk variabel global, dan `.rsrc` untuk sumber daya eksternal). Analisis statis bekerja dengan membedah komponen struktural ini tanpa melakukan eksekusi instruksi biner. Keuntungan utama analisis statis adalah aspek keamanan yang absolut karena kode berbahaya tidak diberi kesempatan untuk berjalan di memori, serta kecepatan ekstraksi informasi yang relatif instan.

Random Forest (RF) merupakan algoritma ensemble berbasis Decision Tree yang digunakan untuk klasifikasi maupun regresi. RF bekerja dengan membangun sejumlah decision tree secara acak, lalu menggabungkan hasil prediksi tiap pohon melalui voting untuk menghasilkan keputusan akhir. Keunggulannya terletak pada kemampuannya mengurangi overfitting dan memberikan performa yang stabil di berbagai jenis dataset.

Flask merupakan sebuah web framework berbasis Python yang tergolong sebagai micro framework. Disebut demikian karena Flask tidak mewajibkan penggunaan pustaka atau alat tertentu serta tidak memiliki database abstraction layer bawaan. Meskipun demikian, Flask mendukung berbagai ekstensi yang dapat menambahkan beragam fungsi aplikasi, seperti object-relational mapper (ORM), validasi formulir, pengelolaan unggahan, hingga autentikasi. Fleksibilitas ini menjadikan Flask salah satu framework ringan yang banyak digunakan untuk pengembangan aplikasi web karena mudah diimplementasikan, mendukung unit testing, serta memiliki dokumentasi yang lengkap dan komunitas yang aktif

3. METODOLOGI PENELITIAN

Arsitektur penelitian dan tahapan rekayasa perangkat lunak dalam riset ini mengadopsi model sekuensial linear (Waterfall) yang dimodifikasi untuk pipeline *machine learning*. Alur metodologi dipilah ke dalam empat fase utama berikut:



GAMBAR 3. Metodologi Penelitian

1. Akuisisi Dataset (Data Acquisition)
   Sampel data dalam penelitian ini dihimpun dari repositori publik malware terverifikasi, yaitu Malware Bazaar, yang menyediakan sampel biner file executableWindows (.exe) yang representatif terhadap varian malware modern di dunia nyata. Sebagai data pembanding seimbang (ground-truth), dikumpulkan berkas-berkas aman (benign) yang diekstrak dari instalasi bersih sistem operasi Windows dan aplikasi perkantoran resmi.
2. Ekstraksi Fitur Statis (Static Feature Extraction)
   Setiap berkas biner mentah diproses menggunakan skrip Python yang memanfaatkan pustaka parsing tingkat rendah pefile. Skrip ini mengurai byte biner untuk mengekstrak informasi struktural dari header internal berkas PE tanpa mengeksekusinya. Sebanyak 75 fitur statis berhasil diekstrak dari setiap file, yang mencakup parameter ukuran memori, versi subsistem, karakteristik DLL, entropi section, dan direktori data keamanan. Hasil ekstraksi dari seluruh sampel dikompilasi menjadi satu berkas terstruktur berformat Comma-Separated Values (.csv) untuk masuk ke tahap prapemrosesan data.
3. Pelatihan dan Pembagian Data (Model Training Configuration)
   Dataset terstruktur kemudian dibagi secara acak menggunakan teknik stratified sampling ke dalam dua subset dengan rasio konvensional 80:20. Sebanyak 80% data dialokasikan sebagai data latih (training set) yang berfungsi memberikan paparan pola fitur yang komprehensif kepada algoritma. Sisa 20% data dialokasikan secara ketat sebagai data uji (testing set) independen, yang bertindak sebagai instrumen evaluasi kemampuan generalisasi model terhadap data baru yang belum pernah dilihat selama fase pelatihan. Pelatihan model Random Forest dilakukan menggunakan pustaka komputasi ilmiah scikit-learn pada platform Python
4. Desain Sistem Aplikasi Web
   Model dengan akurasi terbaik diekspor ke dalam format biner ter-serialisasi (.pkl) menggunakan pustaka pickle. Untuk implementasi di dunia nyata, dibangun sebuah arsitektur aplikasi berbasis web interaktif:
   - Back-end Server. Dikembangkan menggunakan framework mikro Flask (Python) yang bertindak sebagai mesin inferensi (inference engine). Saat pengguna mengunggah file .exe baru, Flask menerima file tersebut, menghasilkan penamaan unik berbasis UUID v4 untuk menghindari tabrakan data, mengekstrak fitur statis file secara on-the-fly, dan melemparkan vektor fitur tersebut ke model Random Forest untuk diprediksi.

   - Database Layer. Menggunakan RDBMS MySQL untuk menyimpan riwayat audit pemindaian (scan history). Skema basis data terdiri dari entitas user (menyimpan kredensial terenkripsi hash Bcrypt) yang berelasi one-to-many terhadap entitas scan_history (mencatat nama file, hasil klasifikasi, probabilitas kecurigaan, dan stempel waktu)

4. HASIL DAN PEMBAHASAN

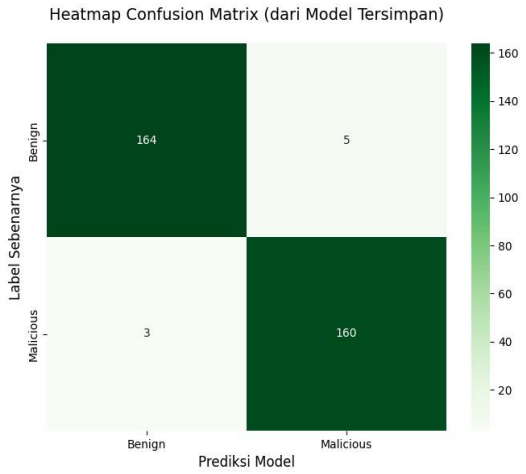
4.1. Analisis Performa Internal Menggunakan Out-of-Bag (OOB) Score

Dalam algoritma berbasis *bagging* seperti *Random Forest*, validasi internal dapat diestimasi secara akurat tanpa memerlukan validasi silang (*cross-validation*) terpisah melalui metrik *Out-of-Bag* (OOB) Score. Selama proses

penumbuhan pohon keputusan, sekitar sepertiga data latihan tidak terpilih ke dalam sampel *bootstrap* untuk pohon tertentu. Data yang tersisih inilah (data OOB) yang dijadikan instrumen pengujian internal untuk pohon tersebut. Hasil komputasi pada fase pelatihan riset ini menorehkan nilai OOB Score sebesar **97,43%**. Pencapaian angka ini menjadi indikator teoretis awal bahwa model memiliki ketahanan generalisasi internal yang sangat kokoh dan risiko *overfitting* yang minimal

4.2. Analisis Matriks Kebingungan (Confusion Matrix Evaluation)

Confusion matrix digunakan untuk merangkum performa klasifikasi secara lebih rinci. Visualisasi dalam bentuk heatmap confusion matrix pada data uji ditunjukkan pada Gambar 4. Untuk melakukan validasi empiris yang objektif, model yang telah dilatih diuji menggunakan data uji eksternal yang sepenuhnya terpisah, dengan total volume pengujian sebanyak 332 file berkas PE. Performa klasifikasi biner dirangkum ke dalam tabel Confusion Matrix yang memetakan kecocokan antara prediksi model dengan nilai kebenaran mutlak (ground truth) data uji. Detail sebaran hasil pengujian 332 berkas tersebut dipaparkan sebagai berikut:



GAMBAR 4. Confusion Matrix

- True Negative (TN): Sebanyak 164 file benign (aman) berhasil diidentifikasi secara tepat oleh model sebagai file non-malware.
- True Positive (TP): Sebanyak 160 file malware (berbahaya) berhasil dideteksi dengan benar sebagai kategori malicious.
- False Positive (FP): Terjadi galat di mana 5 file benign salah diklasifikasikan sebagai ancaman berbahaya.
- False Negative (FN): Terjadi celah keamanan di mana 3 file malware gagal terdeteksi dan lolos sebagai file aman.

Rendahnya nilai FP (5 kasus) dan FN (3 kasus) di antara ratusan sampel data uji menegaskan keandalan taktis dari model Random Forest dalam meminimalkan tingkat kesalahan klasifikasi pada skenario dunia nyata.

4.3. Laporan Klasifikasi (Classification Report)

Laporan klasifikasi pada data uji memberikan informasi rinci mengenai performa model dalam mengklasifikasikan dua kelas, yaitu Benign (Aman) dan Malicious. Nilai precision, recall, dan f1-score dihitung untuk masing-masing kelas, seperti ditunjukkan pada Tabel 1.

Berdasarkan Tabel 2 terlihat bahwa nilai precision dan recall pada kedua kelas hampir seimbang, dengan f1-score sebesar 0,98. Hal ini menunjukkan bahwa model mampu mendeteksi file benign maupun file malware secara konsisten dengan tingkat kesalahan yang sangat rendah. Selain itu, metrik rata-rata dan akurasi keseluruhan model juga dihitung untuk memberikan gambaran performa secara umum. Hasilnya dapat dilihat pada Tabel 2.

TABEL 1. Laporan Klasifikasi Peforma Model

Kelas	Precision	Recall	Support
0,98%	0,97%	0,98%	169
0,97%	0,98%	0,98%	163

TABEL 2. Metrik Rata-Rata dan Akurasi

Metrik	Nilai
Accuracy	0,98%
Macro Avg	0,98%
Weighted Avg	0,98%

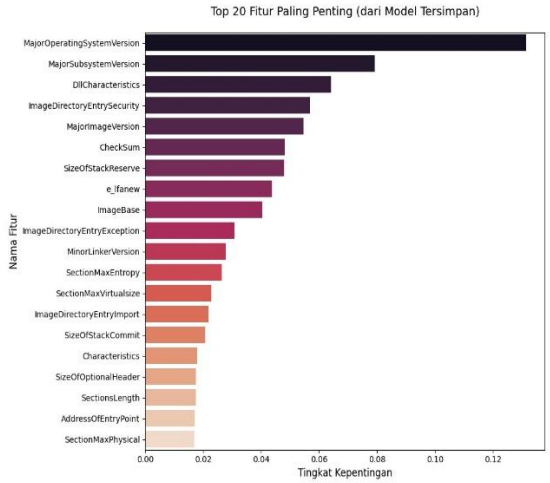
4.4. Analisis Tingkat Kepentingan Fitur (Feature Importance Analysis)

Salah satu keunggulan utama dari penggunaan Random Forest dibandingkan model black-box seperti Deep Learning adalah aspek interpretabilitas model (explainability). Algoritma mampu menghitung kontribusi relatif dari setiap fitur dalam mereduksi ketidakmurnian (impurity) di seluruh pohon keputusan. Berdasarkan hasil ekstraksi signifikansi fitur, ditemukan lima fitur teratas yang memiliki pengaruh paling dominan dalam proses pengambilan keputusan klasifikasi:

- MajorOperatingSystemVersion.
  Menandakan versi mayor dari sistem operasi yang dibutuhkan untuk mengeksekusi file PE. Berkas malware sering kali memanipulasi nilai ini agar kompatibel dengan arsitektur OS lama guna mengeksploitasi kerentanan tertentu.
- MajorSubsystemVersion.
  Menunjukkan versi subsistem utama. Penyimpangan nilai struktural pada field ini

menjadi anomali kuat pembeda file benign.

- DllCharacteristics.
  Flag konfigurasi keamanan biner (seperti ASLR, DEP/NX, dan penanganan enkapsulasi). Berkas malware umumnya menonaktifkan proteksi ini untuk mempermudah injeksi kode atau teknik eksploitasi memori.



GAMBAR 5. Tingkat Kepentingan Feature

- ImageDirectoryEntrySecurity.
  Menunjuk pada alamat tabel direktori sertifikat digital tanda tangan kode. File benign mayoritas memiliki tanda tangan digital yang valid dari pengembang resmi, sedangkan malware cenderung tidak memilikinya atau menggunakan sertifikat palsu.
- MajorImageVersion.
  Versi mayor dari berkas citra eksekusi biner.

Temuan ini membuktikan secara ilmiah bahwa anomali pada struktur konfigurasi subsistem dan atribut pertahanan biner (exploit mitigation markers) merupakan indikator yang paling valid untuk mengidentifikasi muatan berbahaya tanpa perlu membedah baris kode biner secara dinamis.

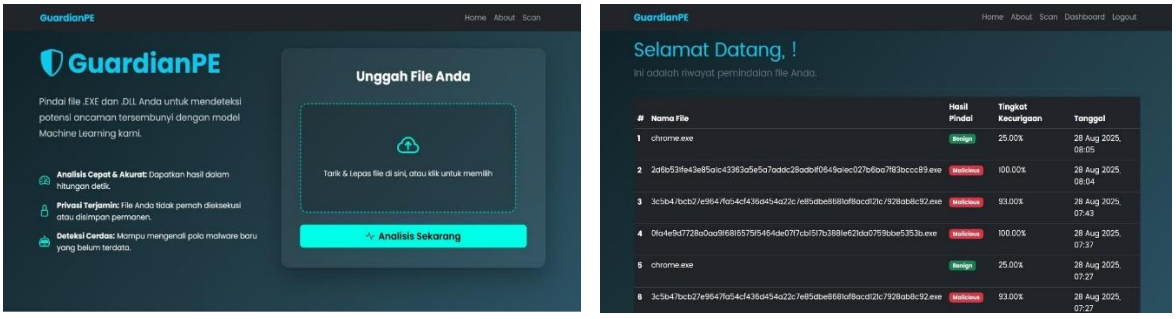
4.5. Evaluasi Pipeline Fungsional pada Lingkungan Web Lokal

Pengujian fungsionalitas sistem secara menyeluruh (end-to-end pipeline) dilakukan untuk memastikan stabilitas integrasi model. Pengujian melibatkan 10 file uji coba acak yang diunggah secara langsung oleh pengguna melalui antarmuka web Flask. Hasil pengujian menunjukkan bahwa sistem berhasil memproses file, melakukan inferensi probabilitas prediksi menggunakan metode predict_proba(), mencatatkan data ke database MySQL, dan menayangkan hasilnya di dashboard secara instan.

File berbahaya secara konsisten memperoleh skor tingkat kecurigaan yang sangat tinggi (berkisar antara 93% hingga 100%), sementara file aman berada pada rentang di bawah 50%. Kendati demikian, ditemukan satu kasus file benign yang berada tepat pada skor marginal borderline sebesar 50%. Hal ini mengindikasikan perlunya kalibrasi lanjutan pada nilai ambang batas keputusan (decision threshold calibration) di masa mendatang guna mengeliminasi potensi bias di area abu-abu (borderline cases)

4.6. Implementasi Antar Muka Aplikasi

Antarmuka pengguna (UI) aplikasi dikembangkan sebagai aplikasi web untuk memastikan aksesibilitas dan kemudahan penggunaan. Desain antarmuka difokuskan untuk menjadi modern, intuitif, dan informatif. Berikut adalah tampilan dari halaman- halaman utama yang telah diimplementasikan.



GAMBAR 6. Antar Muka Aplikasi

5. KESIMPULAN

Sistem yang dibangun mampu mendeteksi file berekstensi .exe dengan tingkat akurasi tinggi, dibuktikan dengan nilai Out-of-Bag (OOB) Score sebesar 97,43% serta hasil uji coba yang menunjukkan kinerja konsisten.

Model Random Forest yang diterapkan berhasil melakukan klasifikasi file benign dan malicious dengan nilai precision, recall, dan f1-score rata-rata 0,98, menunjukkan keandalan sistem dalam mendeteksi malware. Implementasi berbasis website memudahkan pengguna untuk melakukan pemindaian file secara praktis tanpa perlu instalasi perangkat tambahan. Hasil pengujian fungsional menunjukkan pipeline sistem berjalan stabil mulai dari unggah file, klasifikasi, penyimpanan hasil, hingga penayangan riwayat pemindaian. Sistem dapat menjadi solusi alternatif dalam mendukung keamanan cyber, khususnya sebagai lapisan tambahan dalam mendeteksi potensi ancaman malware secara cepat dan efisien.

REFERENSI

[1] M. Selinger, "AV-TEST Awards 2023: shining the spotlight on the best IT security."

[2] M. Asam, S. Hussain Khan, T. Jamal, U. Zahoora, and A. Khan, "Malware Classification Using Deep Boosted Learning."

[3] M. Altaiy, İ. Yildiz, and B. Uçan, "MALWARE DETECTION USING DEEP LEARNING ALGORITHMS," 2023. [Online]. Available: https://orcid.org/0000-0003-2943-3857

[4] E. S. Alomari et al., "Malware Detection Using Deep Learning and Correlation-Based Feature Selection," Symmetry (Basel), vol. 15, no. 1, Jan. 2023, doi: 10.3390/sym15010123.

[5] M. Masum, M. Jobair Hossain Faruk, H. Shahriar, K. Qian, D. Lo, and M. Islam Adnan, "Ransomware Classification and Detection With Machine Learning Algorithms."

[6] F. A. Rafrastara, C. Supriyanto, C. Paramita, Y. P. Astuti, and F. Ahmed, "Performance Improvement of Random Forest Algorithm for Malware Detection on Imbalanced Dataset using Random Under-Sampling Method," vol. 8, no. 2, 2023, [Online]. Available: https://orangedatamining.com/

[7] S. Yoo, S. Kim, S. Kim, and B. B. Kang, "AI-HydRa: Advanced hybrid approach using random forest and deep learning for malware classification," Inf Sci (N Y), vol. 546, pp. 420–435, Feb. 2021, doi:10.1016/j.ins.2020.08.082.

[8] E. Valdis Tjahjadi and B. Santoso, "Klasifikasi Malware Menggunakan Teknik Machine Learning," Copyright @BALOK, vol. 2, no. 1, 2023, [Online]. Available: https://www.kaggle.com/datasets/amauricio/pe-files-malwares.

[9] R. B. Hadiprakoso, W. Rendra Aditya, F. N. Pramitha, P. Siber, and S. Negara, "ANALISIS STATIS DETEKSI MALWARE ANDROID MENGGUNAKAN ALGORITMA SUPERVISED MACHINE LEARNING," 2022.

[10] Y. Wanli Sitorus, P. Sukarno, S. Mandala, F. Informatika, and U. Telkom, "Analisis Deteksi Malware Android menggunakan metode Support Vector Machine & Random Forest," vol. 8, no. 6, p. 12500, 2021.

[11] F. A. Rafrastara, C. Supriyanto, C. Paramita, and Y. P. Astuti, "Deteksi Malware menggunakan Metode Stacking berbasis Ensemble," vol. 8, no. 1, 2023, [Online]. Available: https://orangedatamining.com/

[12] R. Marriam, R. Mohamad, S. Hiew Moi, and H. Amnur, "A Comparative Study of Rumor Detection Domains: Machine Learning, Deep Learning, and Statistical Approaches," 2025. doi: http://dx.doi.org/10.62527/joiv.9.6.4793.

[13] F. Abdussalam and A. Rahmatulloh, "Analisis Efektivitas Algoritma Machine Learning Dalam Deteksi Malware Android …… ANALISIS EFEKTIVITAS ALGORITMA MACHINE LEARNING DALAM DETEKSI MALWARE ANDROID DENGAN STATISTICAL TESTS," 2024, doi:10.35316/jimi.v9i2.124-133.

[14] M. Asam et al., "Detection of exceptional malware variants using deep boosted feature spaces and machine learning," Applied Sciences (Switzerland), vol. 11, no. 21, Nov. 2021, doi: 10.3390/app112110464.

[15] E. Horvitz and D. Mulligan, "Data, privacy, and the greater good," Science (1979), vol. 349, no. 6245, pp. 253–255, Jul. 2015, doi:10.1126/science.aac4520.