JETI
Jurnal Elektronika dan Teknologi Informasi
Vol. 5, No. 2, September 2024
e-ISSN: 2721-9380

Penerapan Algoritma Random Forest dalam Deteksi dan Klasifikasi Ransomware

Mulia Mahendra Alvanof[1]*, Bustami[2], Rozzi Kesuma Dinata[3]
[1,2,3] Program Studi Teknik Informatika,Fakultas Teknik, Universitas Malikussaleh

*Coresponding Email: mulia.170170026@mhs.unimal.ac.id

ABSTRAK

Ransomware merupakan jenis *malware* yang menghalangi akses ke sistem komputer atau data hingga tebusan dibayar oleh korban. Serangan *ransomware* biasanya terjadi karena *file* berbahaya yang diunduh dan dipasang tanpa sadar oleh korban ke sistem komputernya. Mengingat ancaman dan potensi kerugian yang ditimbulkan, metode deteksi dan klasifikasi *ransomware* terus dikembangkan salah satunya dengan memanfaatkan *algoritma machine learning Random Forest*. *Random Forest* dipilih karena kelebihannya dalam menangani dataset besar, waktu pelatihan yang singkat, akurasi prediksi yang tinggi serta kemampuannya mengurangi risiko *overfitting*. Menggunakan 1380 sampel *ransomware* pada *dataset* yang memiliki 54 fitur, 10 fitur terbaik dipilih melalui seleksi fitur dimana model *Random Forest* yang dibangun berhasil memprediksi *file ransomware* dengan akurasi 98.79%.

Kata Kunci: *Ransomware*, *Machine Learning*, Seleksi Fitur, *Random Forest*

ABSTRACT

Ransomware is a type of malware that blocks access to computer systems or data until a ransom is paid by the victim. Ransomware attacks typically occur due to malicious files that are unknowingly downloaded and installed by the victim onto their computer system. Given the threats and potential losses posed, methods for detecting and classifying ransomware continue to be developed, one of which utilizes the Random Forest machine learning algorithm. Random Forest is chosen for its advantages in handling large datasets, short training time, high prediction accuracy, and its ability to reduce the risk of overfitting. Using 1380 ransomware samples from a dataset with 54 features, 10 best features were selected through Feature Selection where the built Random Forest model successfully predicted ransomware files with an accuracy of 98.79%.

Keywords: *Ransomware, Machine Learning, Feature Selection, Random Forest*

1. PENDAHULUAN

Seiring dengan berkembangnya kecerdasan buatan, machine learning juga menjadi semakin penting dan diterapkan dalam berbagai aspek kehidupan. Machine learning memungkinkan mesin belajar dari data dan pengalaman tanpa diprogram secara eksplisit, dengan algoritma yang terus meningkatkan kinerja seiring data yang diterima [1].

Salah satu penerapan machine learning adalah dalam klasifikasi, klasifikasi adalah proses penggolongan atau pengelompokan fungsi yang akan menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk menghasilkan perkiraan kelas dari suatu objek dengan label yang belum diketahui atau pembagian sesuatu berdasarkan kelas-kelas nya [2]. Contoh penerapannya adalah dalam klasifikasi ransomware.

Ransomware adalah malware yang menghalangi akses ke sistem komputer atau data hingga tebusan dibayarkan oleh korban. File dienkripsi dan akses hanya bisa dipulihkan setelah pembayaran, seringkali dalam mata uang kripto [3]. Serangan ini biasanya terjadi melalui tautan atau file berbahaya yang diunduh tanpa sadar oleh korban.

Melihat besarnya ancaman ransomware, penelitian untuk melakukan deteksi dan klasifikasi ransomware terus dikembangkan. Sejalan dengan hal tersebut para kriminal juga terus mengembangkan jenis ransomware

baru yang lebih kompleks [4]. Deteksi dan klasifikasi ransomware dapat dilakukan dengan analisis statis dengan memeriksa karakteristik sampel tanpa menjalankan file atau dinamis dengan memantau perilaku ransomware selama file berjalan [5]. Antivirus modern umumnya menggunakan analisis statis berbasis signature, namun metode ini dinilai kurang efektif terhadap ransomware baru dan dapat dieksploitasi oleh kriminal. Analisis dinamis lebih mahal dan memakan waktu [6].

Untuk mengatasi batasan tersebut, metode machine learning diterapkan sebagai solusi. Pada penelitiannya [7] berhasil menerapkan algoritma machine learning untuk klasifikasi ransomware berdasarkan fitur yang diekstraksi dari file. Dalam penelitian ini, penulis memilih algoritma random forest untuk melakukan deteksi dan klasifikasi file ransomware, algoritma ini dipilih karena kemampuannya menangani dataset besar, waktu pelatihan singkat, hasil prediksi yang akurat, dan mengurangi risiko overfitting sehingga dianggap menjadi algoritma yang paling tepat digunakan dalam penelitian ini, selain itu dalam penelitian ini juga dilakukan seleksi fitur dengan tujuan untuk mengetahui fitur mana yang dapat secara efektif dapat mewakili data, meminimalkan dampak dari noise atau fitur yang kurang relevan serta mencapai hasil prediksi yang akurat.

2. KAJIAN TEORITIS

2.1 Feature Selection

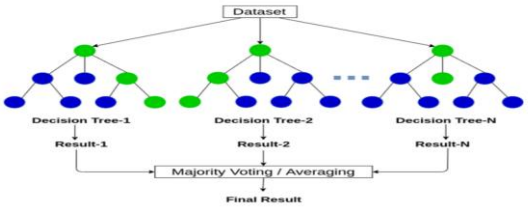
Ransomware adalah jenis malware berbahaya yang mengenkripsi file pada komputer terinfeksi dan membuat komputer tersebut tidak dapat diakses hingga korban membayar tebusan. Para penjahat siber telah memanfaatkan ransomware dan menjadikannya model bisnis yang sangat menguntungkan. Ransomware semakin berkembang seiring waktu, menargetkan tidak hanya pengguna pribadi namun juga perusahaan dalam skala yang lebih besar [5].

2.2 Feature Selection

Feature selection adalah teknik dalam machine learning untuk mengidentifikasi dan memilih fitur-fitur yang paling relevan untuk tugas tertentu. Dengan feature selection, fitur yang tidak relevan atau berlebihan dapat dikesampingkan, dengan begitu tidak hanya data menjadi lebih sederhana tetapi juga dapat meningkatkan kinerja algoritma, mengurangi noise, menghindari overfitting dan meningkatkan stabilitas model [8].

2.3 Random Forest

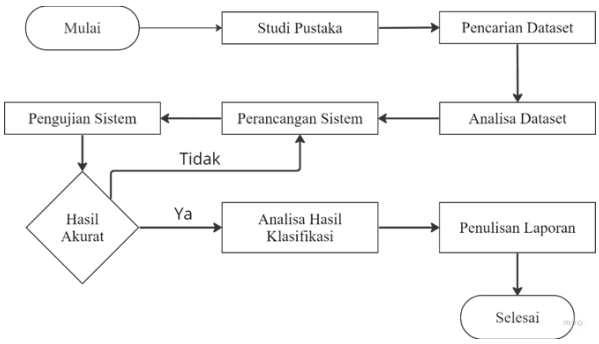
Random Forest merupakan salah satu algoritma machine learning yang paling populer, pada dasarnya random forest adalah kumpulan dari decision tree yang dikelompokan untuk menghasilkan model yang lebih akurat, karena itulah disebut 'forest' yang merupakan sekumpulan decision ‘tree’. Random forest akan membangun sejumlah tree menggunakan data sampel di mana tree yang dibangun pada saat proses training tidak akan bergantung pada tree sebelumnya, kemudian pengambilan keputusan akan diambil berdasarkan voting terbanyak. Dua konsep yang menjadi dasar dari random forest adalah membangun gabungan dari tree melalui bagging (bootstrap aggregating) dengan penggantian dan penyeleksian fitur secara acak untuk tiap tree yang dibangun. Random forest mempunyai dua parameter utama yaitu parameter m yang merupakan presentasi dari jumlah tree yang akan dipakai dan parameter k yang merupakan representasi dari banyaknya fitur maksimal yang dipertimbangkan ketika proses percabangan pada tree [9].



Gambar 1. *Random Forest*

3. METODOLOGI PENELITIAN

Skema sistem merupakan struktur dan mekanisme untuk menghubungkan sekumpulan unsur maupun elemen yang berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan [10]. Berikut adalah skema alur dari penelitian ini



Gambar 2. Skema Alur Penelitian

3.1 Pengumpulan Data

Dalam penelitian ini penulis menggunakan data yang berasal dari *dataset* yang tersedia pada penelitian [11], *dataset* ini dipilih karena tidak banyak *dataset* ransomware lain yang tersedia secara publik dengan jumlah sampel yang memadai untuk melatih dan menguji model *machine learning* secara efektif, selain itu fitur *ransomware* pada *dataset* ini juga telah terdefinisi dengan baik. *Dataset* ini memiliki total 138.047 sampel dengan 56 fitur, pada penelitian ini akan digunakan sebanyak 1% dari total sampel yang tersedia pada *dataset* yaitu 1.380 sampel.

3.2 Pengolahan Data

Karena tidak ada nilai yang hilang maupun duplikat dalam dataset, tahap *preprocessing* hanya melibatkan penghapusan fitur 'name' dan 'md5' karena keduanya tidak memberikan informasi penting untuk penelitian. Fitur lainnya sudah bersifat numerik, sehingga tidak memerlukan *encoding* tambahan. *Dataset* kemudian dibagi menjadi 2, sebagai data *training* dan data *testing*. Data *training* digunakan untuk melatih metode *machine learning* agar dapat melakukan klasifikasi dengan baik dan akurat, sementara data *testing* digunakan untuk menguji akurasi metode yang telah dilatih. Dalam penelitian ini, *dataset* dibagi dengan perbandingan 70:30, dimana 70% digunakan sebagai data *training* dan 30% sebagai data *testing*.

3.3 Seleksi Fitur

Pada tahap ini dilakukan pemilihan fitur dengan tujuan untuk mengetahui fitur mana yang dapat secara efektif mewakili data, meminimalkan dampak dari *noise* atau fitur yang kurang relevan serta mencapai hasil prediksi yang akurat

3.4 Pembangunan Model

Model *machine learning* dilatih untuk melakukan prediksi berdasarkan input data dengan memahami pola dan hubungan antar fitur input dan label output. Dalam penelitian ini, n_estimators=10, criterion='gini', dan random_state=42 digunakan sebagai hyperparameter untuk melatih model algoritma random forest menggunakan pustaka scikit-learn dengan fitur yang telah dipilih sebelumnya. Setelah tahap training selesai, model akan diuji menggunakan data testing untuk mengevaluasi akurasi dan kinerjanya dalam melakukan prediksi pada data yang belum pernah dilihat sebelumnya.

3.5 Pengembangan Sistem

Pada tahap ini akan dilakukan pengembangan aplikasi berbasis web menggunakan Streamlit yang akan digunakan untuk mendeteksi file yang dicurigai sebagai ransomware. Aplikasi akan mengekstraksi fitur dari file yang diunggah, kemudian menggunakan nilai dari tiap fitur file tersebut, prediksi akan dilakukan menggunakan model yang telah dilatih sebelumnya untuk kemudian menentukan apakah file tersebut merupakan ransomware atau tidak.

4. HASIL DAN PEMBAHASAN

4.1 Hasil Dataset

Dataset yang digunakan pada penelitian ini merupakan *dataset* publik yang telah sebelumnya digunakan pada penelitian [11], *dataset* ini dipilih karena selain jumlah sampel yang telah memadai untuk digunakan dalam melatih dan menguji model *machine learning*, fitur *ransomware* pada dataset ini juga telah terdefinisi dengan baik. Pada *dataset* ini terdapat 138.047 sampel dengan 57 kolom (56 fitur dan 1 label) yang mana 70.07% adalah *ransomware* dan 29.93% sisanya adalah *benign*, dari keseluruhan sampel pada dataset, 1.380 sampel akan dipilih secara acak untuk digunakan pada penelitian ini.

```
Report > check_dataset.py > check_dataset
    def check_dataset(dataset_path):
        # Periksa data duplikat dengan 'md5'
        unique_md5_count = df['md5'].nunique()
        total_md5_count = df['md5'].shape[0]
        print("\nJumlah 'md5' unik: ", unique_md5_count)
        print("Jumlah 'md5' total: ", total_md5_count)

Jumlah 'md5' unik:  138047
Jumlah 'md5' total:  138047
PS C:\xampp\htdocs\DeteksiRansomware>
```

Gambar 3. Periksa *Dataset*

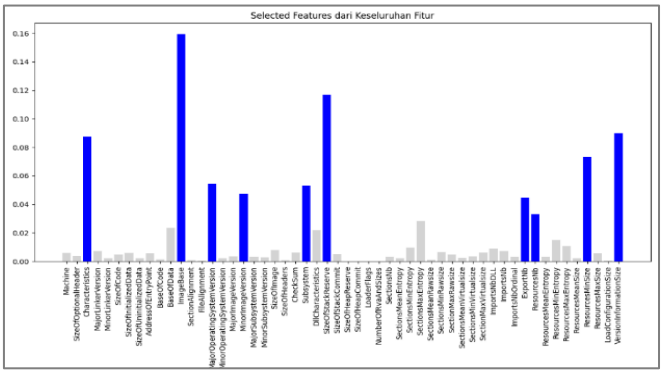
Semua kolom pada *dataset* kecuali ‘Name’ dan ‘md5’ adalah kolom yang berisi data numerik. Menggunakan data pada kolom ‘md5’ dapat diperiksa apakah terdapat data yang duplikat karena tidak akan ada *file* dengan nilai md5 yang sama. Dengan fungsi ‘nunique’ diatas diketahui bahwa terdapat 138.047 data dengan nilai yang berbeda pada kolom ‘md5’, sedangkan dengan fungsi ‘shape’ diketahui bahwa ada 138.047 data pada kolom ‘md5’, dengan begitu dapat dipastikan bahwa tidak ada sampel yang duplikat pada *dataset.*

4.2 Hasil Prepocessing

Telah diketahui bahwa tidak ada sampel duplikat maupun nilai yang hilang pada dataset. Karena fitur 'Name' dan 'md5' tidak memiliki informasi penting yang dapat mempengaruhi penelitian, kedua fitur tersebut akan dihilangkan. Dataset kemudian dibagi menjadi dua bagian, yaitu data training dan data testing. Pada penelitian ini, fungsi `Sklearn train_test_split` digunakan untuk membagi data dengan perbandingan 70:30, dimana 70% dari data, yang berjumlah 966 sampel, akan digunakan sebagai data training, dan 414 sampel, yang merupakan 30% sisanya, akan digunakan sebagai data testing.

4.3 Hasil Feature Selection

Algoritma random forest pada pustaka sklearn.features_selection memiliki atribut ‘feature_importances’ yang dapat digunakan untuk melakukan feature selection metode embedded. Menggunakan fungsi get_support() dan SelectFromModel, dipilihlah 10 fitur terbaik sebagai berikut.



Gambar 4. *Selected Features* dan *Importance* dari 54 Fitur

Semakin tinggi nilai '*importance'* maka semakin penting fitur tersebut, dengan *SelectFromModel* dipilih 10 fitur terbaik dengan pengaruh yang paling signifikan.

Tabel 1. *Selected Feature*

Fitur	Importance
ResourcesNb	0.0330
ExportNb	0.0447
MinorImageVersion	0.0474
Subsystem	0.0532
MajorOperatingSystemVersion	0.0544
ResourcesMinSize	0.0731
Characteristics	0.0875
VersionInformationSize	0.0900
SizeOfStackReserve	0.1168
ImageBase	0.1593

4.4 Hasil *Model Training*

Fitur-fitur yang telah dipilih selanjutnya akan digunakan untuk membentuk data *training* dan data *testing* baru yang hanya memuat fitur yang dipilih tersebut. Pada penelitian ini akan digunakan *hyperparameter* berupa *n_estimators=10*, *criterion='gini'* dan *random_state=42* untuk melatih model *random forest* menggunakan pustaka *scikit-learn* dengan fitur yang telah dipilih sebelumnya. Subset dibuat berdasarkan jumlah *tree* yang akan dibuat, berikut adalah *dataset* yang akan digunakan untuk membuat 10 *subset* berisi sampel acak melalui *bagging* pada *dataset*.

Name	ResourcesNb	ExportNb	Minor Image Version	Subsystem	Major Operating System Version	Resources MinSize	Characteristics	Version InformationSize	SizeOf Stack Reserve	ImageBase	legitimate
Malrope.exe	10	0	0	2	5	16	258	10	1048576	4194304	0
Drixed.exe	13	0	0	2	5	104	258	17	1048576	4194304	0
admin.dll	1	4	0	2	4	1260	8462	16	1048576	1538588672	1
Nabuc.exe	8	0	0	2	4	48	783	0	2097152	4194304	0
Qazxsw.exe	6	0	0	2	5	48	258	15	1048576	4194304	0
Hafnium.exe	11	0	0	2	5	132	258	0	1048576	4194304	0
Trickbot.exe	0	0	0	2	5	0	259	0	1048576	4194304	0
ieproxy.dll	1	5	1	3	6	992	8450	17	262144	147062784	1
Nerba.exe	5	0	0	2	5	48	258	15	1048576	4194304	0
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
urlmon.dll	115	298	3	2	6	20	8450	17	262144	440401920	1
Ursnif.exe	6	0	0	2	5	48	258	15	1048576	4194304	0
wsbcmdlet.dll	1	0	0	3	4	928	8450	17	1048576	1211695104	1
Zeus.exe	6	0	0	2	5	48	258	15	1048576	4194304	0
ADAnalyzer.exe	5	0	0	3	4	34	258	17	1048576	4194304	1
Nymaim.exe	5	0	0	2	5	48	258	15	1048576	4194304	0

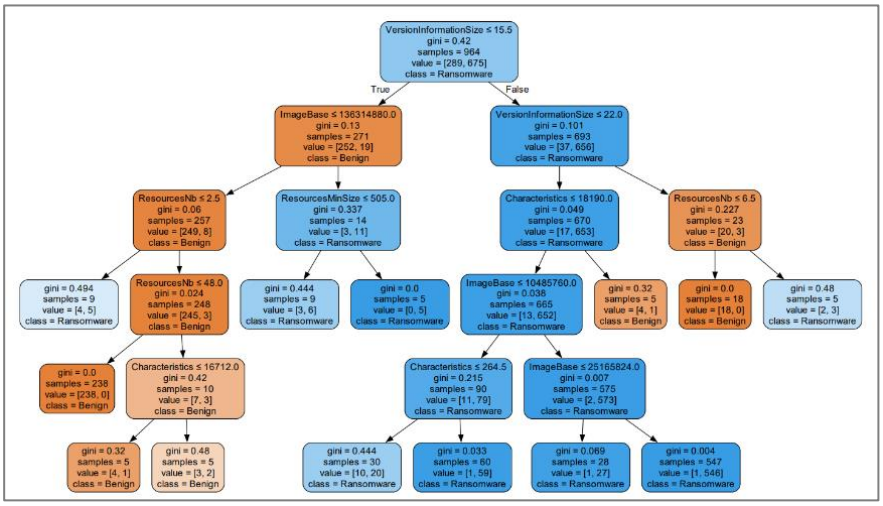
Gambar 5. *Dataset*

Dengan Gini Index akan ditentukan *root* terbaik dari fitur pada setiap *subset.*

Fitur	Split pada	Gini Kiri	Gini Kanan	Gini Total
ResourcesNb	5	0,052487	0,489454	0,270971
ExportNb	1	0,494524	0,017698	0,272928
MinorImageVersion	1	0,487901	0,012711	0,256714
Subsystem	3	0,499986	0,015305	0,304404
MajorOperatingSystemVersion	6	0,498874	0,009569	0,287722
ResourcesMinSize	64	0,449848	0,084866	0,23669
Characteristics	3106	0,399964	0,107673	0,214098
VersionInformationSize	16	0,13039	0,101081	0,10932
SizeOfStackReserve	1048576	0,011928	0,472912	0,233812
ImageBase	16777216	0,359021	0,016721	0,148457

Gambar 6. *Gini Index Subset 1*

Tree dibangun dengan fitur terbaik sebagai *root*. *Tree* yang dibangun kemudian akan digunakan untuk melakukan prediksi dengan menggabungkan hasil prediksi dari semua *tree*, adapun hasil akhir ditentukan berdasarkan *voting* terbanyak dari semua *tree*.



Gambar 7. *Tree Subset 1*

4.4 Hasil Model Testing

Data testing dengan fitur yang telah dipilih sebelumnya akan digunakan untuk menguji model *random forest* yang telah dilatih.

Name	Resou rcesNb	Expo rtNb	Minor Image Versio n	Sub system	Major Operating System Version	Resources MinSize	Charac teristics	Version Information Size	SizeOf Stack Reserve	ImageBase	legiti mate	Predi ction
KBBASH.DLL	1	1	1	1	6	932	8450	16	262144	1610547200	1	1
cgmrgmr.dll	34	0	0	2	5	12	33166	18	1048576	4194304	0	0
dc0d2a8063.dll	14	0	0	2	1	44	33167	15	1048576	4194304	0	0
EPB04A.DLL	5	0	0	2	5	48	258	15	1048576	4194304	0	0
vsvvsvc.exe	6	0	0	2	5	48	258	15	1048576	4194304	0	0
inetcomm.dll	3	116	1	2	6	727	8226	16	262144	8793795526	1	1
LInkEd.dll	13	0	0	2	5	132	259	14	1048576	4194304	0	0
aclayers.dll	73	2	1	3	5	484	8462	16	262144	1901658112	1	1
msoert2.dll	3	208	1	2	6	208	8226	17	262144	8793702072	1	1
IntWANLD.dll	271	0	0	2	5	20	8226	16	1048576	6442450944	1	1
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
deffa1478a9.dll	0	0	0	3	4	0	783	0	2097152	4194304	0	1
AcroBroker.exe	7	0	0	2	4	52	258	16	1048576	4194304	1	0
ssvagent.exe	2	0	0	2	4	675	271	17	1048576	4194304	1	0
ea115506ce.dll	5	4	0	2	5	44	8450	14	1048576	1627389952	0	1
Allsafe.exe	5	0	0	2	5	48	258	15	1048576	4194304	0	1

Gambar 8. Hasil *Gini Index Subset 1*

4.5 Evaluasi Model

Setelah model diuji dengan data *testing* evaluasi dilakukan untuk melihat kinerja model, tabel 4.24 menunjukan bahwa dari 414 sampel *ransomware* yang digunakan sebagai *test*, model *random forest* yang dilatih berhasil melakukan prediksi dengan benar sebanyak 409 sampel dan salah melakukan prediksi sebanyak 5 sampel.

e-ISSN: 2721-9380

Tabel 1. *Confussion Matrix*

Confusion matrix		Aktual	
		Positive	Negative
Prediksi	Positive	297	3
	Negative	2	112

Dari tabel diatas dapat kita ketahui metrik-metrik sebagai berikut:

$$Precision = \frac{TP}{TP + FP} = \frac{297}{297 + 3} = 0.99$$

$$Recall = \frac{TP}{TP + FN} = \frac{297}{297 + 2} = 0.99$$

$$F1 - score = \frac{2 \times (precision \times recall)}{precision + recall} = \frac{2 \times (0.99 \times 0.99)}{0.99 + 0.99} = 0.99$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{297 + 112}{297 + 112 + 3 + 2} = \frac{409}{414} = 98.79\%$$

Dari data diatas dapat disimpulkan bahwa model *random forest* yang dilatih dengan terlebih dahulu melakukan seleksi fitur menghasilkan model dengan kinerja yang sangat baik dan efektif dalam klasifikasi *ransomware*.

4.6 Implementasi Sistem

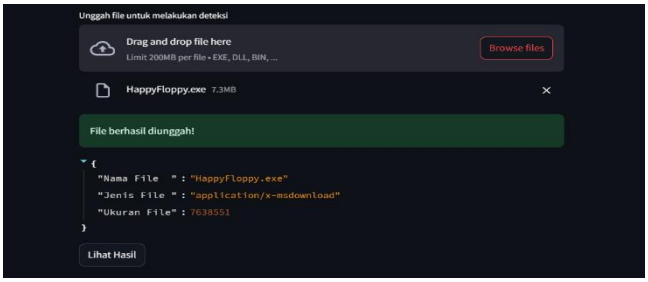
Dalam penelitian ini akan dikembangkan aplikasi berbasis web menggunakan bahasa pemrograman Python dengan framework Streamlit. Aplikasi akan mengekstraksi fitur dari file yang diunggah kemudian fitur dari file tersebut akan digunakan untuk melakukan prediksi menggunakan model yang telah dilatih sebelumnya, setelah itu akan ditentukan apakah file tersebut merupakan ransomware atau tidak. Sistem yang akan dikembangkan akan memiliki 2 halaman, yaitu halaman utama untuk mengunggah file dan halaman hasil untuk melihat hasil deteksi.

A. Halaman Utama

Pada halaman utama ini pengguna dapat mengunggah file yang ingin mereka periksa, dengan mengklik *browse files* maka manajer berkas akan muncul dimana pengguna dapat mencari dan memilih *file* yang ingin mereka periksa, selain itu pengguna juga dapat mengunggah file dengan drag dan drop file ke halaman ini.

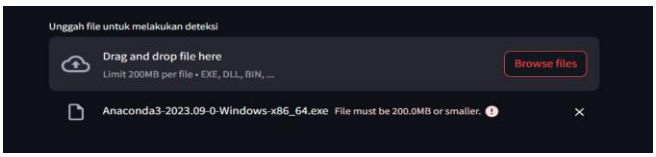
Gambar 9. Halaman Utama

Setelah memilih *file* yang akan diperiksa akan ditampilkan teks yang menyatakan bahwa "File berhasil diunggah!" jika *file* memang berhasil diunggah, selain itu akan ditampilkan juga nama, jenis, dan ukuran *file*



Gambar 10. Unggah Berhasil

Jika terdapat masalah saat mengunggah file, misalnya pengguna mengunggah file dengan ukuran yang melebihi 200MB, maka akan ditampilkan pesan agar pengguna mengunggah *file* dengan ukuran ≤ 200MB. Pengguna dapat kembali mengklik *browse files* untuk mengunggah *file* baru.



Gambar 11. Unggah Gagal

B. Halaman Hasil

Setelah berhasil mengunggah *file*, pengguna dapat melihat hasil prediksi dengan mengklik pada tombol "Lihat Hasil".

Gambar 12. Hasil Deteksi *Legitimate*

Jika ternyata *file* yang diunggah bukan merupakan *ransomware*, maka akan ditampilkan hasil bahwa *file* terdeteksi sebagai *Legitimate*, sebaliknya jika *file* terdeteksi sebagai *ransomware* maka akan ditampilkan hasil bahwa *file* adalah *ransomware*.

Gambar 13. Hasil Deteksi *Ransomware*

5. SIMPULAN

Berdasarkan tahapan penelitian yang dilakukan, algoritma Random Forest berhasil diterapkan untuk mendeteksi dan mengklasifikasi file ransomware dengan tingkat akurasi yang sangat tinggi, yaitu 98.79%. Proses ini dimulai dengan pembagian dataset ransomware menjadi data pelatihan dan pengujian dengan rasio 70:30, tanpa memerlukan banyak langkah praproses karena fitur dalam dataset sudah terdefinisi dengan baik. Untuk meningkatkan efisiensi dan kinerja model, seleksi fitur dilakukan menggunakan pustaka Scikit-learn, yang menghasilkan pemilihan 10 fitur terbaik dari total 54 fitur, berdasarkan signifikansi terhadap prediksi ransomware. Model dilatih menggunakan 966 sampel ransomware dan diuji dengan 414 sampel, menunjukkan bahwa pendekatan ini efektif dalam mengidentifikasi ransomware secara akurat dan dapat diandalkan untuk aplikasi keamanan siber.

DAFTAR PUSTAKA

[1] V., Rameshbabu., C., Vijayakumaran., P., B., EDWIN, PRABHAKAR. (2023). Machine Learning. Character Lab tips, doi: 10.53776/tips-gratitude-machine-learning.
[2] Dinata, R., Akbar, H., & Hasdyna, N. (2020). Algoritma K-Nearest Neighbor dengan Euclidean Distance dan Manhattan Distance untuk Klasifikasi Transportasi Bus. ILKOM Jurnal Ilmiah, 12(2), 104-111. doi:https://doi.org/10.33096/ilkom.v12i2.539.104-111.
[3] Noorbehbahani, Fakhroddin & Rasouli, Farzaneh & Saberi, Mohammad. (2019). Analysis of Machine learning Techniques for Ransomware Detection. 128-133. 10.1109/ISCISC48546.2019.8985139.
[4] Adamu, Umaru & Awan, Irfan. (2019). Ransomware Prediction Using Supervised Learning Algorithms. 57-63. 10.1109/FiCloud.2019.00016.
[5] Dinata, R. K., Fajriana, F., Zulfa, Z., & Hasdyna, N. (2020). Klasifikasi Sekolah Menengah Pertama/Sederajat Wilayah Bireuen Menggunakan Algoritma K-Nearest Neighbors Berbasis Web. CESS (Journal of Computer Engineering, System and Science), 5(1), 33-37.
[6] Bahaa, Yamany., Mahmoud, Said, Elsayed., Anca, Delia, Jurcut., Nashwa, Abdelbaki., Marianne, A., Azer. (2022). A New Scheme for Ransomware Classification and Clustering Using Static Features. Electronics, doi: 10.3390/electronics11203307.
[7] Manabu, Hirano., Ryotaro, Kobayashi. (2022). Machine learning-based Ransomware Detection Using Low-level Memory Access Patterns Obtained From Live-forensic Hypervisor. doi: 10.1109/CSR54599.2022.9850340.
[8] Masum, Mohammad & Hossain Faruk, Md Jobair & Shahriar, Hossain & Qian, Kai & Lo, Dan & Adnan, Muhaiminul. (2022). Ransomware Classification and Detection With Machine learning Algorithms. 10.1109/CCWC54503.2022.9720869.
[9] Dinata, R. K., Hasdyna, N., & Alif, M. (2021). Applied of Information Gain Algorithm for Culinary Recommendation System in Lhokseumawe. Journal Of Informatics And Telecommunication Engineering, 5(1), 45-52.
[10] Gustavo, Sosa-Cabrera., M., Garc'ia-Torres., Christian, E., Schaerer. (2023). Feature Selection: A perspective on inter-attribute cooperation. arXiv.org, doi: 10.48550/arXiv.2306.16559.
[11] [11] Muliadi, Muliadi., Andi, Farmadi., Rudy, Herteno., Rahmat, Ramadhani. (2023). Random forest Dengan Random Search Terhadap Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung. Jurnal Informatika, doi: 10.31294/inf.v10i1.14531
[12] Rizal, R., Bustami, B., & Azzahra, D. (2019). Pendeteksi Tajwid Idgham Mutajanisain Pada Citra Al-Qur'an Menggunakan Fuzzy Associative Memory (FAM). TECHSI-Jurnal Teknik Informatika, 11(3), 395-407.
[13] Dinata, R. K., Hasdyna, N., Retno, S., & Nurfahmi, M. (2021). K-means algorithm for clustering system of plant seeds specialization areas in east Aceh. ILKOM Jurnal Ilmiah, 13(3), 235-243.
[14] M. Mathur, "Ransomware (malware) detection using Machine learning," GitHub, https://github.com/muditmathur2020/RansomwareDetection/blob/master/Ransomware.csv. [Diakses: Maret, 2024].