Klasifikasi Serangan Malware Menggunakan Machine Learning

Dian Ristey Ajeng Pramono
Fakultas Informatika
Direkotorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
dianristeyajeng@student.telkomuniversity.ac.id

Wahyu Adi Prabowo
Fakultas Informatika
Direkotorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
wahyup@telkomuniversity.ac.id

Muhammad Fajar Sidiq
Fakultas Informatika
Direkotorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
mfsidiq@telkomuniversity.ac.id

***Abstrak* — Serangan malware merupakan ancaman signifikan dalam era digitalisasi dengan lebih dari 6,5 miliar serangan tercatat pada tahun 2024. Penelitian ini bertujuan mengevaluasi dan membandingkan performa tiga algoritma machine learning yaitu Random Forest, Support Vector Machine (SVM), dan Naïve Bayes dalam mengklasifikasi serangan malware. Dataset pe-files-malwares dari Kaggle dengan 19.611 sampel dan 79 fitur digunakan sebagai basis eksperimen. Metodologi penelitian meliputi exploratory data analysis, preprocessing dengan normalisasi StandardScaler, seleksi fitur menggunakan SelectKBest, penanganan ketidakseimbangan kelas dengan SMOTE, dan pembagian data dengan rasio 80:20. Evaluasi model menggunakan confusion matrix dengan metrik accuracy, precision, recall, dan F1-score. Hasil penelitian menunjukkan Random Forest memberikan performa terbaik dengan akurasi 98,8%, precision 98,7%, recall 99,7%, dan F1-score 99,2%. SVM mencapai akurasi 96,7% dengan F1-score 97,7%, sedangkan Naïve Bayes memperoleh akurasi 88,2% dengan F1-score 91,7%. Random Forest terbukti paling efektif dalam mendeteksi malware dengan tingkat false negative terendah, menjadikannya solusi optimal untuk implementasi sistem keamanan siber.**

***Kata kunci*— Malware, Klasifikasi, Cybersecurity, Algorithm, Detection**

I. PENDAHULUAN

Peradaban manusia secara bertahap mengalami modernisasi yang membawa dampak signifikan pada berbagai aspek kehidupan, terutama dalam era teknologi dan informasi yang memanfaatkan internet sebagai salah satu implementasi kemajuan tersebut [1]. Seiring dengan perkembangan internet, keamanan siber menjadi semakin rentan terhadap berbagai jenis serangan, dengan malware menjadi salah satu ancaman yang paling signifikan [2]. Malware merupakan istilah umum yang merujuk pada setiap program atau perangkat lunak yang dibuat untuk menyusup atau merusak sistem komputer dan sistem operasi, termasuk virus, worm, trojan horse, sebagian besar rootkit, spyware, adware, dan program-program berbahaya lainnya [3].

Ancaman malware telah menunjukkan tren peningkatan yang mengkhawatirkan dalam beberapa tahun terakhir. Berdasarkan data dari explodingtopics.com/blog/cybersecurity-stats, pada tahun 2024 tercatat lebih dari 6,5 miliar serangan malware di seluruh dunia, mengalami peningkatan sekitar 8% dibandingkan tahun sebelumnya. Dalam rentang 2020 hingga 2024, jumlah serangan malware terus berkisar miliaran, dengan puncak tertinggi terjadi pada tahun 2018 yaitu mencapai 10,5 miliar serangan. Grafik jumlah serangan malware pada periode 2015-2034 menunjukkan konsistensi ancaman ini sebagai masalah serius dalam dunia digital.

Gambar 1. Grafik jumlah serangan malware pada periode 2015-2034

Menghadapi kompleksitas ancaman tersebut, diperlukan metode deteksi malware yang efektif dengan menerapkan algoritma machine learning yang mampu mendeteksi pola baru dan mengenali karakteristik serangan malware. Machine Learning merupakan implementasi dari Artificial Intelligence (AI) yang bertujuan mengembangkan sistem yang dapat belajar secara mandiri tanpa pemrograman ulang berulang, menggunakan data pelatihan dalam proses pembelajarannya sebelum menghasilkan output yang diinginkan [4].

Penelitian terdahulu telah menunjukkan efektivitas berbagai model machine learning dalam deteksi malware. Random Forest dipilih karena kemampuannya mengurangi overfitting melalui kombinasi beberapa pohon keputusan yang stabil dan akurat, dengan penelitian Evan Valdis Tjahjadi, Budy Santoso, dan Serwin (2023) menunjukkan tingkat akurasi mencapai 99% [5]. Support Vector Machine

(SVM) efektif dalam membedakan kelas secara optimal pada data berdimensi tinggi dengan membentuk hyperplane maksimum, unggul dalam menangani model nonlinier kompleks dengan ketahanan terhadap overfitting [6]. Sementara Naïve Bayes dipilih karena efisiensi komputasionalnya sebagai metode klasifikasi probabilistik sederhana yang mengestimasi probabilitas melalui frekuensi dan kombinasi nilai dataset [7].

Meskipun berbagai penelitian telah mengeksplorasi penggunaan individual model-model tersebut, masih terdapat kebutuhan untuk melakukan evaluasi komprehensif perbandingan kinerja ketiga model dalam konteks deteksi malware yang sama. Permasalahan utama yang dihadapi adalah bagaimana menentukan model machine learning yang paling efektif dalam mengklasifikasi malware berdasarkan metrik evaluasi yang komprehensif.

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja tiga model machine learning yaitu Random Forest, Support Vector Machine (SVM), dan Naïve Bayes dalam mengklasifikasi serangan malware menggunakan dataset pe-files-malwares. Evaluasi dilakukan menggunakan metrik confusion matrix yang meliputi accuracy, precision, recall, dan F1-score untuk menentukan model yang memberikan performa terbaik. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan metode deteksi malware yang lebih efektif dan menjadi referensi dalam pemilihan model klasifikasi malware yang optimal berdasarkan evaluasi metrik yang komprehensif [8], [9].

II. KAJIAN TEORI

A. Analisis Dinamis dalam Deteksi Malware

Analisis dinamis merupakan pendekatan deteksi malware yang mengamati perilaku program berbahaya saat dieksekusi dalam lingkungan terkontrol. Penelitian Ramadhan et al. (2020) menunjukkan bahwa analisis dinamis menggunakan metode Naïve Bayes mampu mencapai akurasi deteksi malware sebesar 85% [10]. Metode ini menganalisis perilaku malware saat dijalankan dan mencari pola mencurigakan dalam aktivitasnya. Meskipun akurasinya lebih rendah dibandingkan analisis statis, analisis dinamis memberikan informasi berharga tentang perilaku real-time malware. Pendekatan ini unggul mengidentifikasi malware yang menggunakan teknik obfuskasi atau packing. Data perilaku dari analisis dinamis dapat digunakan sebagai fitur untuk melatih algoritma klasifikasi machine learning, memberikan dimensi tambahan dalam proses deteksi malware.

B. Analisis Statis untuk Klasifikasi Malware

Analisis statis menganalisis kode dan struktur file malware tanpa menjalankan program, menghindari risiko infeksi sistem. Penelitian Ramadhan et al. (2020) menunjukkan metode Naïve Bayes mencapai akurasi deteksi 93%, lebih efektif dibandingkan analisis dinamis [10]. Keunggulan utamanya adalah kemampuan mengekstraksi fitur struktural dari file executable dengan aman. Octaviani (2022) mengembangkan pendekatan serupa untuk deteksi PDF malware dengan mengekstraksi metadata sebagai fitur klasifikasi. Menggunakan Support Vector Machine dengan kernel RBF, penelitian tersebut mencapai presisi 88,67%, recall 87%, F-1 Score 86,67%, dan akurasi 87%, membuktikan efektivitas metadata sebagai sumber fitur klasifikasi malware [11].

C. Confusion Matrix sebagai Metrik Evaluasi

Confusion matrix adalah instrumen evaluasi fundamental dalam sistem klasifikasi machine learning yang menyajikan representasi tabular untuk menggambarkan kinerja algoritma klasifikasi dengan membandingkan kelas aktual dan prediksi. Dalam klasifikasi biner malware, matriks 2x2 ini menggambarkan empat kemungkinan hasil: True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN) [12]. Metrik turunannya meliputi akurasi, presisi, recall, dan F1-Score. Penelitian Yogasware et al. (2021) menggunakan metrik ini untuk mengevaluasi performa K-Nearest Neighbor dalam klasifikasi malware family, menghasilkan recall 65% dan precision 83% [13].

D. Cyberattack dan Ancaman Keamanan Siber

Cyberattack atau serangan siber merupakan tindakan yang menggunakan perangkat, jaringan komputer, atau kode komputer untuk melakukan serangan yang merusak secara sengaja melalui teknologi komputer dan internet [14]. Dalam konteks keamanan siber modern, malware menjadi salah satu vektor utama serangan siber yang dapat menyebabkan kerugian signifikan bagi individu, organisasi, maupun infrastruktur kritis.

Evolusi cyberattack menunjukkan tren peningkatan kompleksitas dan sofistikasi, membutuhkan pendekatan deteksi yang adaptif dan otomatis. Machine learning menawarkan solusi yang menjanjikan untuk mengidentifikasi pola serangan baru yang belum pernah ditemui sebelumnya, mengatasi keterbatasan sistem deteksi berbasis signature tradisional.

E. Dataset dan Preprocessing Data

Dataset merupakan sekumpulan data terstruktur yang menjadi fondasi untuk melatih dan mengevaluasi algoritma klasifikasi dalam penelitian machine learning deteksi malware. Penelitian ini menggunakan dataset pe-files-malwares dari Kaggle berkapasitas 6,554 MB, berisi klasifikasi file malware (malicious) atau benign [15], [16]. Preprocessing data melibatkan normalisasi Min Max Scaling untuk mengubah rentang nilai menjadi 0-1, dan Random Under-Sampling mengatasi ketidakseimbangan kelas. Rafrastara et al. (2023) menerapkan Principal Component Analysis (PCA) untuk reduksi fitur, berhasil meningkatkan performa Random Forest menjadi 98,7% dengan mengurangi fitur dari 1084 menjadi 32 [17].

F. Google Colab sebagai Platform Penelitian

Google Colab menyediakan lingkungan pengembangan terpadu (IDE) untuk pemrograman Python dengan komputasi server Google yang dilengkapi perangkat keras berkualitas tinggi. Platform ini telah menyediakan pustaka esensial seperti Keras, TensorFlow, NumPy, Pandas, dan Matplotlib untuk visualisasi data. Keunggulan Google Colab meliputi integrasi dengan Google Drive, opsi prosesor CPU/GPU/TPU, dan kapasitas RAM yang memadai untuk penelitian machine learning [18].

Dalam konteks penelitian deteksi malware, Google Colab memfasilitasi eksperimen dengan berbagai algoritma machine learning tanpa memerlukan investasi infrastruktur komputasi yang mahal, memungkinkan peneliti untuk fokus pada pengembangan model dan analisis hasil.

G. Machine Learning dalam Deteksi Malware

Machine Learning adalah teknologi kecerdasan buatan yang memungkinkan komputer memperoleh pengetahuan dari data untuk mengembangkan model prediktif. Dalam deteksi malware, machine learning mempelajari pola dan karakteristik sampel malware yang diketahui untuk mengidentifikasi sampel baru melalui analisis statis, dinamis, atau hybrid [19]. Tjahjadi et al. (2023) membuktikan efektivitas Random Forest dalam klasifikasi malware dengan akurasi 99%, menunjukkan algoritma ensemble memberikan performa superior [5]. Sitorus et al. (2021) membandingkan Support Vector Machine dan Random Forest untuk deteksi malware Android, dengan Random Forest mengungguli SVM mencapai akurasi 98,99% [20].

H. Naïve Bayes sebagai Metode Klasifikasi Probabilistik

Naïve Bayes merupakan metode klasifikasi probabilistik yang beroperasi dengan menghitung probabilitas untuk sekumpulan fitur dalam dataset. Metode ini menentukan kelas klasifikasi dengan membandingkan nilai posterior dari berbagai kelas yang mungkin, dengan kelas yang memiliki nilai posterior tertinggi dipilih sebagai hasil klasifikasi. Rumus dasar Naïve Bayes adalah $P(Ci|X) = P(X|Ci)P(Ci)/P(X)$, di mana $P(Ci|X)$ adalah probabilitas kelas Ci dengan bukti X [21], [22], [23].

Implementasi Naïve Bayes dalam deteksi malware menunjukkan hasil yang menjanjikan, khususnya dalam analisis statis dengan akurasi mencapai 93%. Keunggulan metode ini terletak pada kesederhanaan komputasi dan kemampuan menangani dataset dengan dimensi tinggi secara efisien.

I. Python sebagai Bahasa Pemrograman

Python, yang dikembangkan oleh Guido Van Rossum dan diluncurkan pada tahun 1991, telah berkembang menjadi salah satu bahasa pemrograman paling populer dalam machine learning dan deep learning. Fleksibilitas Python memungkinkan penggunaan dalam berbagai bidang penelitian, didukung oleh ekosistem pustaka yang kaya seperti Scikit-learn untuk implementasi algoritma machine learning, NumPy untuk komputasi numerik, dan Matplotlib untuk visualisasi data [24], [25], [26], [27].

Dalam penelitian deteksi malware, Python menyediakan infrastruktur yang komprehensif untuk pengembangan sistem klasifikasi, mulai dari preprocessing data hingga evaluasi model, memungkinkan peneliti untuk fokus pada aspek metodologis tanpa terhambat oleh kompleksitas implementasi teknis.

J. Random Forest sebagai Metode Ensemble

Random Forest merupakan metode machine learning efektif untuk klasifikasi dataset besar, terdiri dari pohon keputusan independen yang dibentuk melalui subset data acak menggunakan bootstrap. Parameter kunci meliputi n_estimator (jumlah pohon), max_depth (kedalaman maksimum), max_features (fitur maksimal per split), dan bootstrap untuk mencegah overfitting [28], [29], [30]. Arsitektur menggunakan majority voting dengan rumus $f(x) = \text{argmax} \Sigma\ 1(Ci = k)$ untuk menentukan hasil klasifikasi. Penelitian menunjukkan konsistensi Random Forest memberikan performa tinggi dalam deteksi malware, dengan multiple studies melaporkan akurasi di atas 98%, menjadikannya algoritma robust untuk aplikasi keamanan siber.

K. Support Vector Machine dan Optimasi Hyperplane

Support Vector Machine (SVM) merupakan metode klasifikasi yang berfokus pada pengelompokkan data menggunakan hyperplane dalam ruang berdimensi tinggi dengan margin maksimal. Perkembangan signifikan SVM terjadi pada tahun 1992 ketika Boser, Vapnik, dan Guyon memperkenalkan kernel trick yang memungkinkan SVM menangani data non-linear secara efisien. Anwar et al. (2022) menunjukkan bahwa SVM dengan kernel RBF memberikan performa superior dibandingkan kernel Polynomial dalam deteksi PDF malware [31].

Implementasi SVM dalam deteksi malware melibatkan optimasi fungsi Lagrange untuk menemukan hyperplane optimal yang memisahkan kelas malware dan benign dengan margin maksimum, memberikan solusi yang robust terhadap noise dan overfitting dalam dataset keamanan siber.

III. METODE

A. Rancangan Penelitian

Penelitian ini menggunakan pendekatan eksperimental dengan metode perbandingan algoritma machine learning untuk klasifikasi serangan malware. Rancangan penelitian dirancang untuk membandingkan efektivitas tiga algoritma machine learning: Random Forest (RF), Support Vector Machine (SVM), dan Naïve Bayes (NB) dalam mendeteksi malware melalui analisis file Portable Executable (PE).

B. Prosedur Penelitian

Penelitian ini menggunakan metode sistematis dalam klasifikasi serangan malware dengan machine learning. Dimulai dengan studi literatur mendalam terhadap jurnal internasional dan penelitian terdahulu mengenai Random Forest, Support Vector Machine, dan Naïve Bayes dalam cybersecurity. Dataset PE (Portable Executable) dari Kaggle.com dengan 79 fitur dan 5.554 KB dianalisis melalui Exploratory Data Analysis (EDA). Preprocessing meliputi normalisasi data, feature selection, dan penanganan imbalanced data menggunakan SMOTE dengan train-test split 80:20. Modeling mengimplementasikan tiga algoritma dengan hyperparameter tuning. Evaluasi menggunakan confusion matrix dengan metrik accuracy, precision, recall, dan F1-score untuk mengukur performa klasifikasi malware dan benign files.

C. Waktu Penelitian

Penelitian dilaksanakan selama 6 bulan, dimulai dari Januari hingga Juni 2024, dengan alokasi waktu sebagai berikut: studi literatur (1 bulan), data collection dan EDA (1 bulan), preprocessing (1 bulan), modeling (2 bulan), dan evaluation serta dokumentasi (1 bulan).

D. Sumber Data

TABEL 1 SPESIFIKASI DATASET

Parameter	Keterangan
Sumber	Kaggle.com
Nama Dataset	PE-files-malwares
Ukuran	5.554 KB
Jumlah Fitur	79
Kelas Target	Malware, Benign

E. Perangkat Penelitian

TABEL 2 SPESIFIKASI PERANGKAT KERAS DAN LUNAK

Komponen	Spesifikasi
Device	*Lenovo IdeaPad S145*
Processor	*Intel Core i5-8265U @ 1.60GHz*
RAM	*8 GB*
Sistem Operasi	*Windows 11 Pro*

Bahasa Pemrograman	*Python*
Platform Development	*Google Colab, Visual Studio Code*

F. Metode Analisis

Analisis performa dilakukan menggunakan persamaan evaluasi standar machine learning:

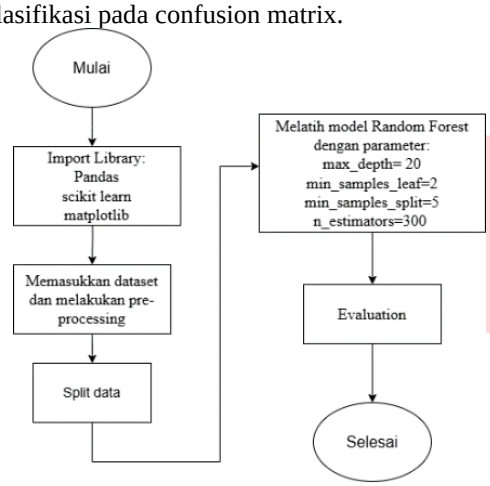
Accuracy = (TP + TN) / (TP + TN + FP + FN) (1)

Precision = TP / (TP + FP) (2)

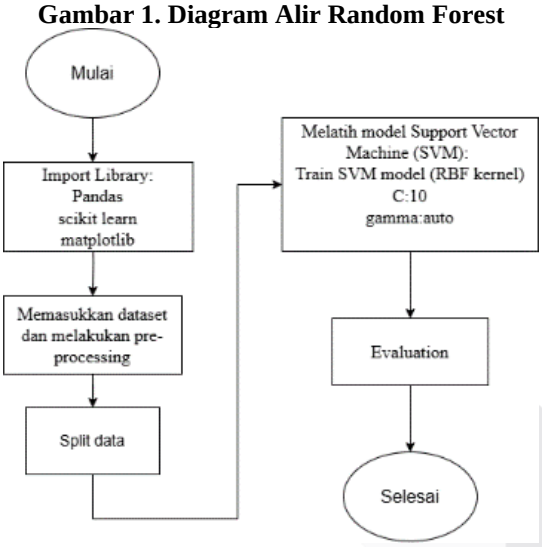
Recall = TP / (TP + FN) (3)

F1-Score = 2 × (Precision × Recall) / (Precision + Recall) (4)

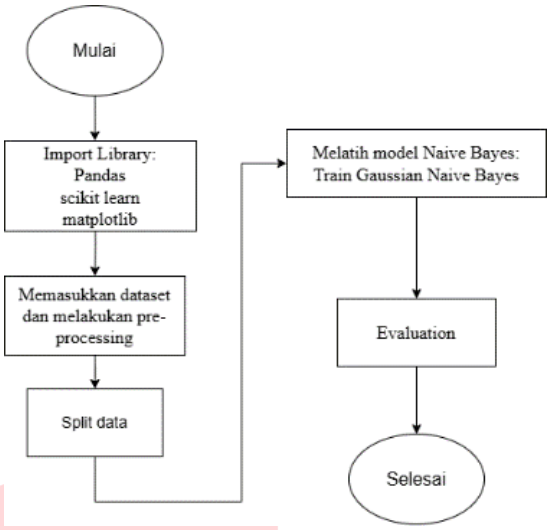
Dimana TP (True Positive), TN (True Negative), FP (False Positive), dan FN (False Negative) merepresentasikan hasil klasifikasi pada confusion matrix.



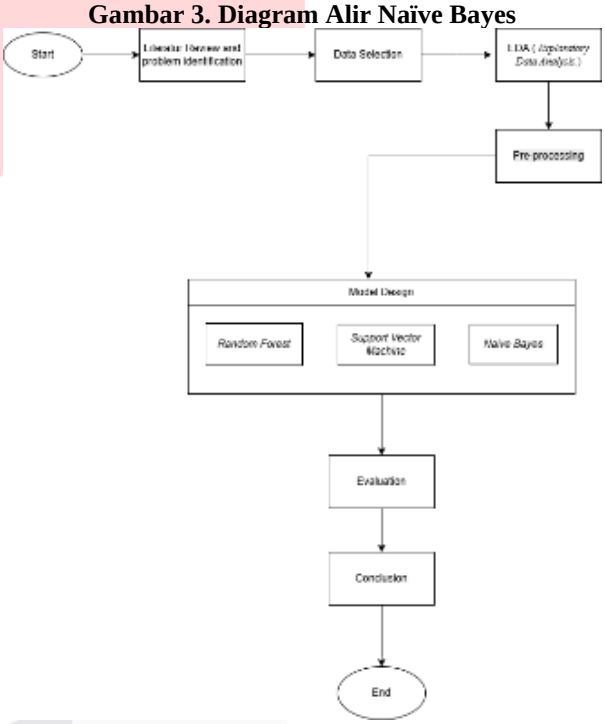
Gambar 1. Diagram Alir Random Forest



Gambar 2. Diagram Alir Support Vector Machine (Svm)



Gambar 3. Diagram Alir Naïve Bayes



Gambar 4. Diagram Blok Keseluruhan Penelitian

G. Validasi Model

Validasi dilakukan menggunakan cross-validation untuk memastikan robustness model dan menghindari overfitting. Setiap algoritma dievaluasi pada dataset test yang sama untuk memastikan komparabilitas hasil. Perbandingan performa akan difokuskan pada nilai F1-score sebagai metrik utama mengingat karakteristik dataset yang berpotensi imbalanced. Metodologi ini dirancang untuk menghasilkan analisis komparatif yang objektif dan dapat direproduksi, sehingga memberikan kontribusi signifikan dalam pengembangan sistem deteksi malware berbasis machine learning.

IV. HASIL DAN PEMBAHASAN

HASIL

A. Karakteristik Dataset dan Eksplorasi Data

Penelitian ini menggunakan dataset PE (Portable Executable) files yang diperoleh dari platform Kaggle.com dengan URL https://www.kaggle.com/datasets/amauricio/pe-files-malwares. Dataset tersebut memiliki karakteristik berupa 79

fitur dengan kapasitas data sebesar 5.554 KB, yang mencakup informasi komprehensif tentang file benign dan file berbahaya (malicious) yang diekstraksi dari struktur PE files.

```
(19611, 79)
```

Gambar 1. Jumlah Data dan Features

Berdasarkan hasil eksplorasi data awal, ditemukan bahwa dataset terdiri dari 19.611 sampel dengan 79 fitur yang mencakup berbagai aspek teknis dari PE files. Fitur-fitur tersebut meliputi informasi header PE, karakteristik section, nilai entropi, dan berbagai atribut teknis lainnya yang relevan untuk proses klasifikasi malware. Struktur fitur dalam dataset mencakup tipe data integer (Int64), float (Float64), dan object, dengan distribusi yang dominan berupa data numerik yang sesuai untuk proses pembelajaran mesin.

Tabel 1. Distribusi Tipe Data dalam Dataset

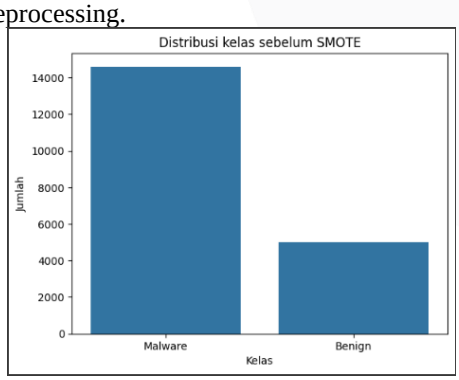
Tipe Data	Jumlah Fitur	Persentase
Int64	77	97.5%
Float64	1	1.3%
Object	1	1.3%

```
# Missing Value
df.isnull().sum()# Missing Value
```

SizeOfHeapReserve	0
SizeOfHeapCommit	0
LoaderFlags	0
NumberOfRvaAndSizes	0
Malware	0
SuspiciousImportFunctions	0
SuspiciousNameSection	0
SectionsLength	0
SectionMinEntropy	0
SectionMaxEntropy	0

Gambar 2. Hasil Pemeriksaan Missing Value pada Dataset

Analisis kualitas data menunjukkan bahwa dataset tidak memiliki nilai yang hilang (missing values), sebagaimana dikonfirmasi melalui pemeriksaan menggunakan fungsi isnull() dari library pandas. Kondisi ini mengindikasikan kualitas data yang baik dan mengurangi kompleksitas dalam tahap preprocessing.



Gambar 3. Distribusi Kelas sebelum SMOTE

```
Malware    14599
Benign      5012
```

Gambar 4. Distribusi Jumlah Kelas Sebelum Melakukan SMOTE

Evaluasi distribusi kelas menunjukkan adanya ketidakseimbangan yang signifikan antara kelas Benign dan Malware. Sebelum penerapan teknik balancing, distribusi menunjukkan 5.012 sampel untuk kelas Benign (25.5%) dan 14.599 sampel untuk kelas Malware (74.5%). Ketidakseimbangan ini berpotensi mempengaruhi performa model pembelajaran mesin, sehingga memerlukan penanganan khusus dalam tahap preprocessing.

B. Preprocessing dan Transformasi Data

Tahap preprocessing dimulai dengan eliminasi fitur yang tidak relevan untuk proses klasifikasi. Tiga fitur dihapus dari dataset, yaitu 'Name' (identifier unik), 'Malware' (target label asli), dan 'Malware_label' (versi label yang telah dimapping untuk visualisasi). Penghapusan fitur-fitur ini bertujuan untuk menghindari kebocoran informasi (data leakage) dan memfokuskan model pada fitur-fitur yang benar-benar prediktif.

```
[[ 0.   -0.04 -0.05 ... -0.07 -0.02 -0.04]
 [ 0.   -0.04 -0.05 ... -0.04 -0.02 -0.02]
 [ 0.   -0.04 -0.05 ... -0.08 -0.02 -0.04]
 ...
 [ 0.   -0.18 -0.05 ... -0.07 -0.02 -0.04]
 [ 0.   -0.04 -0.05 ... -0.07 -0.02 -0.04]
 [ 0.   -0.04 -0.05 ...  0.   -0.02 -0.04]]
```

Gambar 5. Hasil data setelah proses normalisasi

Proses normalisasi data dilakukan menggunakan StandardScaler untuk menyesuaikan skala fitur dengan mengubah setiap fitur agar memiliki rata-rata 0 dan standar deviasi 1. Transformasi ini penting untuk algoritma yang sensitif terhadap skala data seperti Support Vector Machine (SVM), serta membantu mengurangi bias akibat perbedaan skala antar fitur. Seleksi fitur dilakukan menggunakan metode SelectKBest dengan fungsi skor f_classif (ANOVA F-value) untuk mengidentifikasi 10 fitur terbaik berdasarkan relevansi terhadap variabel target. Fitur-fitur terpilih meliputi Machine, TimeDateStamp, SizeOfOptionalHeader, Magic, SectionAlignment, MajorSubsystemVersion, Subsystem, SizeOfStackReserve, SuspiciousImportFunctions, dan SectionMaxChar.

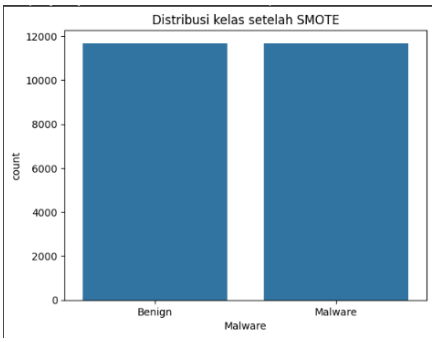
Tabel 2. Fitur Terpilih Berdasarkan Skor Relevansi

No	Nama Fitur	Kategori
1	Machine	PE Header
2	TimeDateStamp	PE Header
3	SizeOfOptionalHeader	PE Header
4	Magic	PE Header
5	SectionAlignment	PE Optional Header
6	MajorSubsystemVersion	PE Optional Header
7	Subsystem	PE Optional Header
8	SizeOfStackReserve	PE Optional Header
9	SuspiciousImportFunctions	Security Feature
10	SectionMaxChar	Section Characteristic

```
X_train, X_test, y_train, y_test = train_test_split(X_selected_df, y, test_size=0.2, random_state=42, stratify=y)
```

Gambar 6. Data Split

Pembagian data dilakukan menggunakan train_test_split dengan rasio 80:20 untuk data latih dan data uji, dengan parameter stratify untuk mempertahankan proporsi kelas yang sama pada kedua subset. Parameter random_state=42 digunakan untuk memastikan konsistensi hasil eksperimen.



Gambar 7. Distribusi Kelas setelah SMOTE

```
0    11679
1    11679
```

Gambar 8. Distribusi Jumlah Kelas Setelah Melakukan SMOTE

Penanganan ketidakseimbangan kelas dilakukan menggunakan Synthetic Minority Over-sampling Technique (SMOTE) yang diterapkan hanya pada data latih untuk menghindari data leakage. Hasil penerapan SMOTE menunjukkan keseimbangan sempurna dengan 11.679 sampel untuk masing-masing kelas Benign dan Malware pada data latih.

Tabel 3. Distribusi Data

Kelas	Sebelum Preprocessing	Sesudah Preprocessing
Benign	5.012 (25.5%)	11.679 (50%)
Malware	14.599 (74.5%)	11.679 (50%)

C. Implementasi dan Optimasi Model

1. Random Forest

```
GridSearchCV
GridSearchCV(cv=3, estimator=RandomForestClassifier(random_state=42), n_jobs=-1,
             param_grid={'max_depth': [None, 10, 20, 30],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [100, 200, 300]},
             scoring='accuracy', verbose=2)
best_estimator_: RandomForestClassifier
RandomForestClassifier(max_depth=20, min_samples_leaf=2, min_samples_split=5,
                       n_estimators=300, random_state=42)
RandomForestClassifier
RandomForestClassifier(max_depth=20, min_samples_leaf=2, min_samples_split=5,
                       n_estimators=300, random_state=42)
```

Gambar 9. Visualisasi Model Random Forest

Implementasi Random Forest menggunakan teknik GridSearchCV untuk pencarian hyperparameter optimal dengan parameter yang dievaluasi meliputi n_estimators (100, 200, 300), max_depth (10, 20, None), min_samples_split (2, 5, 10), dan min_samples_leaf (1, 2, 4). Proses optimasi menggunakan 3-fold cross-validation untuk memperoleh evaluasi yang robust dan meminimalkan risiko overfitting.

2. Support Vector Machine (SVM)

```
GridSearchCV
GridSearchCV(cv=3, estimator=SVC(class_weight='balanced'), n_jobs=-1,
             param_grid={'C': [0.1, 1, 10],
                         'gamma': ['scale', 'auto', 0.01, 0.001, 0.0001,
                                   1e-05]},
             verbose=2)
best_estimator_: SVC
SVC(C=10, class_weight='balanced', gamma='auto')
SVC
SVC(C=10, class_weight='balanced', gamma='auto')
```

Gambar 10. Visualisasi Model Support Vector Machine (SVM)

Model SVM dengan kernel RBF dioptimasi menggunakan GridSearchCV dengan parameter C (0.1, 1, 10, 100) untuk mengontrol trade-off antara kesalahan pelatihan dan margin keputusan, serta gamma (0.001, 0.01, 0.1, 1) untuk menentukan pengaruh sampel individual. Parameter class_weight='balanced' digunakan untuk menangani ketidakseimbangan kelas secara otomatis.

3. Naïve Bayes

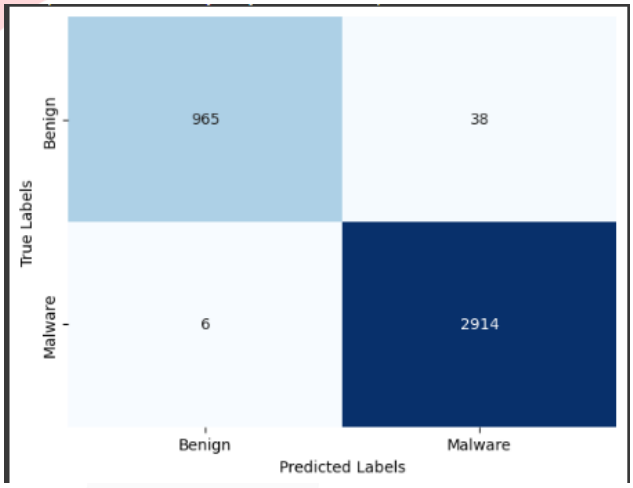
```
GaussianNB
GaussianNB()
```

Gambar 11. Visualisasi Model Naïve Bayes

Model Naïve Bayes menggunakan GaussianNB() dengan asumsi distribusi normal untuk fitur kontinu. Model ini dipilih karena kesederhanaannya dan efektivitas pada dataset dengan fitur yang relatif independen.

D. Evaluasi Performa Model

1. Random Forest



Gambar 12. Confusion Matrix model Random Forest

	precision	recall	f1-score	support
Benign	0.99	0.96	0.98	1003
Malware	0.99	1.00	0.99	2920
accuracy			0.99	3923
macro avg	0.99	0.98	0.99	3923
weighted avg	0.99	0.99	0.99	3923

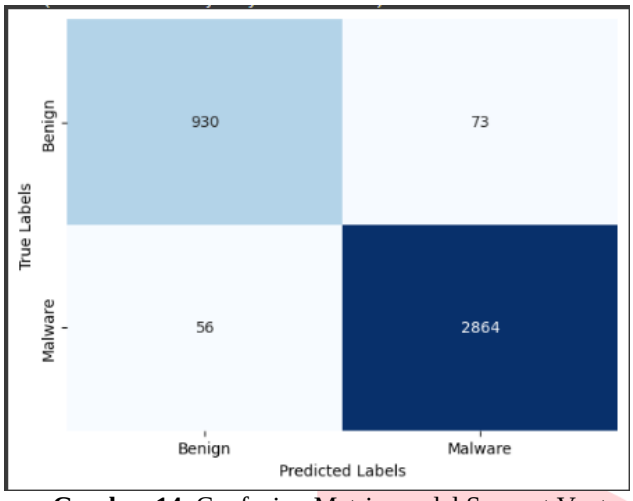
Gambar 13. Classification Report model Random Forest

Evaluasi model Random Forest menunjukkan performa yang sangat baik dengan akurasi 98.8%. Analisis confusion matrix (Gambar 12) menghasilkan nilai True Positive (TP) = 2.914, False Positive (FP) = 38, False Negative (FN) = 6, dan True Negative (TN) = 965. Perhitungan metrik evaluasi menunjukkan precision 98.7%, recall 99.7%, dan F1-score 99.2%. Classification report model Random Forest.

Tabel 4. Hasil Evaluasi Random Forest

Metrik	Nilai Manual	Nilai Sistem
Accuracy	0.988	0.989
Precision	0.987	0.988
Recall	0.997	1.000
F1-Score	0.992	0.994

2. Support Vector Machine (SVM)



Gambar 14. Confusion Matrix model Support Vector Machine (SVM)

	precision	recall	f1-score	support
Benign	0.94	0.93	0.94	1003
Malware	0.98	0.98	0.98	2920
accuracy			0.97	3923
macro avg	0.96	0.95	0.96	3923
weighted avg	0.97	0.97	0.97	3923

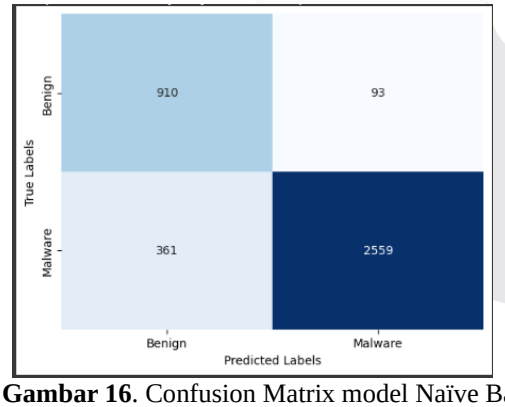
Gambar 15. Classification Report model Support Vector Machine (SVM)

Model SVM menunjukkan performa yang baik dengan akurasi 96.7%. Confusion matrix (Gambar 14) menghasilkan TP = 2.864, FP = 73, FN = 56, dan TN = 930. Metrik evaluasi menunjukkan precision 97.5%, recall 98.0%, dan F1-score 97.7%. Classification report model SVM.

Tabel 5. Hasil Evaluasi Support Vector Machine

Metrik	Nilai Manual	Nilai Sistem
Accuracy	0.967	0.967
Precision	0.975	0.975
Recall	0.980	0.980
F1-Score	0.977	0.978

3. Naïve Bayes



Gambar 16. Confusion Matrix model Naïve Bayes

	precision	recall	f1-score	support
Benign	0.72	0.91	0.80	1003
Malware	0.96	0.88	0.92	2920
accuracy			0.88	3923
macro avg	0.84	0.89	0.86	3923
weighted avg	0.90	0.88	0.89	3923

Gambar 17. Classification Report model Naïve Bayes

Model Naïve Bayes menunjukkan performa yang lebih rendah dibandingkan kedua model lainnya dengan akurasi 88.2%. Confusion matrix (Gambar 16) menghasilkan TP = 2.559, FP = 93, FN = 361, dan TN = 910. Metrik evaluasi menunjukkan precision 96.4%, recall 87.6%, dan F1-score 91.7%. Classification report model Naïve Bayes.

Tabel 6. Hasil Evaluasi Naïve Bayes

Metrik	Nilai Manual	Nilai Sistem
Accuracy	0.882	0.882
Precision	0.964	0.965
Recall	0.876	0.876
F1-Score	0.917	0.918

Perbandingan performa ketiga model menunjukkan bahwa Random Forest memberikan hasil terbaik dengan akurasi tertinggi (98.8%), diikuti oleh SVM (96.7%), dan Naïve Bayes (88.2%). Superioritas Random Forest dapat dikaitkan dengan kemampuannya menangani fitur yang kompleks dan interaksi non-linear antar fitur melalui ensemble dari multiple decision trees.

Tabel 7. Perbandingan Performa Model

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	98.8%	98.7%	99.7%	99.2%
SVM	96.7%	97.5%	98.0%	97.7%
Naïve Bayes	88.2%	96.4%	87.6%	91.7%

Berdasarkan pembahasan jurnal "Klasifikasi Serangan Malware Menggunakan Machine Learning", analisis confusion matrix menunjukkan Random Forest memiliki false negative terendah (6 kasus), krusial untuk keamanan karena kegagalan deteksi malware berisiko tinggi. SVM menunjukkan performa seimbang dengan false positive dan false negative rendah, sedangkan Naïve Bayes memiliki false negative tinggi (361 kasus). Seleksi fitur SelectKBest efektif mengidentifikasi fitur diskriminatif seperti Machine, TimeDateStamp, dan SuspiciousImportFunctions. Penerapan SMOTE berhasil mengatasi ketidakseimbangan kelas dan meningkatkan performa model, terutama dalam deteksi kelas minoritas (Benign), menghasilkan recall tinggi pada semua model untuk identifikasi kedua kelas.

PEMBAHASAN

A. Performa Model Random Forest dalam Klasifikasi Malware

Model Random Forest menunjukkan performa superior dengan akurasi 0.99, precision 0.99, recall 1.00, dan F1-score 0.99 dalam klasifikasi malware [32], [33]. Keunggulan ini disebabkan karakteristik ensemble yang terdiri dari multiple decision trees dengan subset acak data dan fitur [34]. Mekanisme ensemble memungkinkan model menangkap pola kompleks sambil mengurangi overfitting. Confusion matrix menghasilkan F1-score tinggi, menunjukkan keseimbangan optimal antar kelas. Model berhasil mengklasifikasikan data malicious dan benign secara akurat, aspek kritis dalam sistem deteksi malware [35], [36].

B. Analisis Model Support Vector Machine (SVM)

SVM dengan kernel RBF menunjukkan performa kompetitif dengan akurasi 0.97, precision 0.98, recall 0.98, dan F1-score 0.98, meski di bawah Random Forest. Model ini efektif membedakan malware dan file benign. Optimalisasi bergantung pada parameter tuning C (regularisasi) dan gamma yang menentukan toleransi kesalahan klasifikasi dan pengaruh data point terhadap boundary decision. Confusion matrix menunjukkan kesalahan klasifikasi minimal. Mengingat dataset tidak seimbang dengan proporsi malware lebih tinggi, F1-score menjadi metrik evaluasi lebih relevan dibanding akurasi karena mempertimbangkan keseimbangan

deteksi malware (recall) dan ketepatan menghindari false positive (precision) [37], [38].

C. Evaluasi Model Naive Bayes

Naive Bayes menunjukkan performa acceptable dengan akurasi 0.88, precision tinggi 0.96 mengindikasikan prediksi malware akurat dengan false positive rendah. Recall 0.88 menunjukkan kemampuan mendeteksi sebagian besar malware meski beberapa kasus tidak teridentifikasi. F1-score memperlihatkan keseimbangan baik antara precision dan recall, mengindikasikan stabilitas performa model. Karakteristik Naive Bayes sebagai model ringan dan efisien cocok untuk sistem dengan keterbatasan komputasi [39]. Random Forest terkonfirmasi sebagai pilihan optimal untuk klasifikasi malware. Penelitian masa depan dapat mengeksplorasi deep learning approaches, hybrid models, optimalisasi feature engineering, dan pengembangan sistem real-time detection untuk keamanan siber yang lebih robust [40].

V. KESIMPULAN

Penelitian komparatif tiga algoritma machine learning untuk klasifikasi malware menunjukkan Random Forest sebagai yang terbaik dengan akurasi 98,8% dan recall 99,7%. SVM RBF mencapai akurasi 96,7%, sedangkan Naïve Bayes 88,2% dengan precision tinggi 96,4%. Preprocessing menggunakan SelectKBest dan SMOTE meningkatkan performa signifikan. Random Forest direkomendasikan sebagai solusi optimal untuk sistem deteksi malware berbasis machine learning.

REFERENSI

[1] B. R. Sanjaya *et al.*, "Pengembangan Cyber Security dalam Menghadapi Cyber Warfare di Indonesia," *J. Adv. Res. Def. Secur. Stud.*, vol. 1, no. 1, pp. 19–34, 2022.

[2] E. Tansen and D. W. Nurdiarto, "Analisis dan Deteksi Malware dengan Metode Hybrid Analysis Menggunakan Framework MOBSF," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 191–201, 2020, doi: 10.36294/jurti.v4i2.1338.

[3] T. A. Cahyanto, V. Wahanggara, and D. Ramadana, "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *J. Sist. Teknol. Inf. Indones.*, vol. 2, no. 1, pp. 19–30, 2017.

[4] C. Chazar and B. Erawan, "Machine Learning Diagnosis Kanker Payudara Menggunakan Algoritma Support Vector Machine," *Inf. (Jurnal Inform. dan Sist. Informasi)*, vol. 12, no. 1, pp. 67–80, 2020, doi: 10.37424/informasi.v12i1.48.

[5] E. V. Tjahjadi and B. Santoso, "Klasifikasi Malware Menggunakan Teknik Machine Learning," *J. Ilm. Ilmu Komput.*, vol. 2, no. 1, pp. 60–70, 2023.

[6] K. Abdul Khalim, U. Hayati, and A. Bahtiar, "Perbandingan Prediksi Penyakit Hipertensi Menggunakan Metode Random Forest Dan Naïve Bayes," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 7, no. 1, pp. 498–504, 2023, doi: 10.36040/jati.v7i1.6376.

[7] F. Dharma, S. Shabrina, A. Noviana, M. Tahir, N. Hendrastuty, and W. Wahyono, "Prediction of Indonesian Inflation Rate Using Regression Model Based on Genetic Algorithms," *J. Online Inform.*, vol. 5, no. 1, pp. 45–52, 2020, doi: 10.15575/join.v5i1.532.

[8] T. A. Aziz, Z. Sari, C. Sri, and K. Aditiya, "Klasifikasi Malware android dengan menggunakan metode XGBoost Algoritma," *Repositor*, vol. 7, no. 1, pp. 103–110, 2025.

[9] A. Maslan, A. A. Fajrin, and A. D. Putri, "Klasifikasi dan Deteksi Malware Menggunakan Variasi Model Algoritma Machine learning," *J. Edukasi dan Penelit. Inform.*, vol. 11, no. 1, pp. 129–140, 2025.

[10] B. Ramadhan, Y. Purwanto, and M. Ruriawan, *Forensic Malware Identification Using Naive Bayes Method*. 2020. doi: 10.1109/ICITSI50517.2020.9264959.

[11] S. Devella, Y. Yohannes, and F. N. Rahmawati, "Implementasi Random Forest Untuk Klasifikasi Motif Songket Palembang Berdasarkan SIFT," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 2, pp. 310–320, 2020, doi: 10.35957/jatisi.v7i2.289.

[12] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf. Sci. (Ny).*, vol. 507, pp. 772–794, 2020, doi: https://doi.org/10.1016/j.ins.2019.06.064.

[13] A. R. Yogaswara, "Klasifikasi Malware Family menggunakan Metode k-Nearest Neighbor (k-NN)," *J. Repos.*, vol. 3, no. 3, pp. 305–314, 2021, doi: 10.22219/repositor.v2i3.1313.

[14] M. A. Suharto and Maria Novita Apriyani, "Konsep Cyber Attack, Cyber Crime, Dan Cyber Warfare Dalam Aspek Hukum Internasional," *Risal. Huk.*, vol. 17, pp. 98–107, 2021, doi: 10.30872/risalah.v17i2.705.

[15] A. Nurhopipah and C. Magnolia, "Perbandingan Metode Resampling Pada Imbalanced Dataset Untuk Klasifikasi Komentar Program Mbkm," *J. Publ. Ilmu Komput. dan Multimed.*, vol. 2, no. 1, pp. 9–22, 2023, doi: 10.55606/jupikom.v2i1.862.

[16] S. Salsabilla, P. Putri, V. Suryani, and E. M. Jadied, "A nalisis Performansi Exponential Mechanism Pada Dataset Student ' s Alcohol Consumptions Dalam Memenuhi ? -Differential Privacy," vol. 8, no. 5, pp. 9994–10007, 2021.

[17] F. A. Rafrastaraa, R. A. Pramunendar, D. P. Prabowo, E. Kartikadarma, and U. Sudibyo, "Optimasi Algoritma Random Forest menggunakan Principal Component Analysis untuk Deteksi Malware," *J. Teknol. Dan Sist. Inf. Bisnis*, vol. 5, no. 3, pp. 217–223, 2023, doi: 10.47233/jteksis.v5i3.854.

[18] Trias Handayanto and Rahmadya Raya, "Prediksi KelasJamak dengan Deep Learning Berbasis GraphicsProcessing Units," *J. Kaji. Ilm.*, vol. 20, no. 1, pp. 1410–9794, 2020, [Online]. Available: http://ejurnal.ubharajaya.ac.id/index.php/JKI

[19] Y. I. Kurniawan, "Perbandingan Algoritma Naive Bayes dan C.45 dalam Klasifikasi Data Mining," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 4, pp. 455–464, 2018, doi: 10.25126/jtiik.201854803.

[20] Y. W. Sitorus, P. Sukarno, and S. Mandala, "Analisis Deteksi Malware Android menggunakan metode

Support Vector Machine & Random Forest," *eProceedings Eng.*, vol. 8, no. 6, 2021.

[21] H. M. Siregar, "Bulletin of Information Technology (BIT) Implementasi Metode Naive Bayes Pada Perancangan Aplikasi Sistem Pakar Diagnosa Bronkiektasis," *Bull. Inf. Technol.*, vol. 1, no. 3, pp. 112–121, 2020.

[22] M. R. S. Alfarizi, M. Z. Al-farish, M. Taufiqurrahman, G. Ardiansah, and M. Elgar, "Penggunaan Python Sebagai Bahasa Pemrograman untuk Machine Learning dan Deep Learning," *Karya Ilm. Mhs. Bertauhid (KARIMAH TAUHID)*, vol. 2, no. 1, pp. 1–6, 2023.

[23] R. Rosaly and A. Prasetyo, "Flowchart Beserta Fungsi dan Simbol-Simbol," *J. Chem. Inf. Model.*, vol. 2, no. 3, pp. 5–7, 2020.

[24] A. Puspita and A. D. Kalifia, "Analisi Depresi Selama Pandemi Covid-19 Menggunakan Algoritma Random Forest," vol. 2, no. 1, pp. 336–338, 2024.

[25] D. Diana, R. E. Indrajit, and E. Dazki, "Komparasi Algoritma Naïve Bayes, Logistic Regression Dan Support Vector Machine pada Klasifikasi File Application Package Kit Android Malware," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 11, no. 1, p. 109, 2022, doi: 10.35889/jutisi.v11i1.815.

[26] X. Xiong, S. Hu, D. Sun, S. Hao, H. Li, and G. Lin, "Detection of false data injection attack in power information physical system based on SVM–GAB algorithm," *Energy Reports*, vol. 8, pp. 1156–1164, 2022, doi: 10.1016/j.egyr.2022.02.290.

[27] E. Susilowati, M. K. Sabariah, and A. A. Gozali, "Implementasi Metode Support Vector Machine untuk Melakukan Klasifikasi Kemacetan Lalu Lintas Pada Twitter," *E-Proceeding Eng.*, vol. 2, no. 1, pp. 1478–1484, 2015.

[28] R. I. Nurachim, "Pemilihan Model Prediksi Indeks Harga Saham Yang Dikembangkan Berdasarkan Algoritma Support Vector Machine(Svm) Atau Multilayer Perceptron(Mlp) Studi Kasus : Saham Pt Telekomunikasi Indonesia Tbk," *J. Teknol. Inform. dan Komput.*, vol. 5, no. 1, pp. 29–35, 2019, doi: 10.37012/jtik.v5i1.243.

[29] L. Siliayani, Iqbal Agis Junizar, Uyu Nuraeni, Edi Tohidi, and Irfan Ali, "Penerapan Algoritma Naive Bayes Untuk Mengetahui Kepuasan Mahasiswa Terhadap Layanan Administrasi Keuangan," *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 4, no. 3, pp. 72–79, 2020, doi: 10.32485/kopertip.v4i3.122.

[30] K. Liu, S. Xu, G. Xu, M. Zhang, D. Sun, and H. Liu, "A Review of Android Malware Detection Approaches Based on Machine Learning," *IEEE Access*, vol. 8, pp. 124579–124607, 2020, doi: 10.1109/ACCESS.2020.3006143.

[31] S. Anwar, T. Suprapti, G. Dwilestari, and I. Ali, "Pengelompokkan Hasil Belajar Siswa dengan Metode Clustering K-Means," *JURSISTEKNI (Jurnal Sist. Inf. dan Teknol. Informasi)*, vol. 4, no. 2, pp. 60–72, 2022.

[32] M. A. Abubakar, M. Muliadi, A. Farmadi, R. Herteno, and R. Ramadhani, "Random Forest Dengan Random Search Terhadap Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung," *J. Inform.*, vol. 10, no. 1, pp. 13–18, 2023, doi: 10.31294/inf.v10i1.14531.

[33] M. Arief, P. H. Trisnawan, and M. Data, "Implementasi Sistem Deteksi Serangan Slowloris pada Arsitektur Jaringan Software-Defined Network Menggunakan Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, pp. 2548–964, 2024, [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/13399

[34] H. Nalatissifa, W. Gata, S. Diantika, and K. Nisa, "Perbandingan Kinerja Algoritma Klasifikasi Naive Bayes, Support Vector Machine (SVM), dan Random Forest untuk Prediksi Ketidakhadiran di Tempat Kerja," *J. Inform. Univ. Pamulang*, vol. 5, no. 4, p. 578, 2021, doi: 10.32493/informatika.v5i4.7575.

[35] V. Z. Kamila and E. Subastian, "KNN vs Naive Bayes Untuk Deteksi Dini Putus Kuliah Pada Profil Akademik Mahasiswa," *J. Rekayasa Teknol. Inf.*, vol. 3, no. 2, p. 116, 2019, doi: 10.30872/jurti.v3i2.3097.

[36] D. A. Nasution, H. H. Khotimah, and N. Chamidah, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 78, 2019, doi: 10.24114/cess.v4i1.11458.

[37] N. C. Nugraha, H. Hikmayanti, J. Indra, and A. R. Juwita, "Implementasi Metode Resampling Dalam Menangani Data Imbalance Pada Klasifikasi Multiclass Penyakit Thyroid," *Build. Informatics, Technol. Sci.*, vol. 6, no. 2, pp. 890–900, 2024, doi: 10.47065/bits.v6i2.5652.

[38] Y. Galahartlambang, T. Khotiah, and Jumain, "Visualisasi Data Dari Dataset COVID-19 Menggunakan Pemrograman Python," *J. Ilm. Intech Inf. Technol. J. UMUS*, vol. 3, no. 01, pp. 58–60, 2021, [Online]. Available: https://jurnal.umus.ac.id/index.php/intech/article/view/417

[39] I. M. M. Matin, M. Agustin, B. Sugiarto, and A. N. Asri, "Deteksi Malware Menggunakan Machine Learning Dengan Metode Ensemble," *Pros. Sains Nas. dan Teknol.*, vol. 13, no. 1, pp. 265–270, 2023, doi: 10.36499/psnst.v13i1.9224.

[40] V. Kirankumar, S. Ramasubbareddy, G. Kannayaram, and K. Nikhil Kumar, "Classification of Heart Disease Using Support Vector Machine," *J. Comput. Theor. Nanosci.*, vol. 16, pp. 2623–2627, May 2019, doi: 10.1166/jctn.2019.7941.