Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

**Jurnal Comasie**

ISSN (Online) 2715-6265

# DETEKSI SERANGAN MALWARE MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE*

**Hery Sanjaya Simbolon[1], Andi Maslan[2]**

[1]Program Studi Teknik Informatika, Universitas Putera Batam
[2]Program Studi Teknik Informatika, Universitas Putera Batam
*email*: pb210210069@upbatam.ac.id

## ABSTRACT

*The rapid development of information technology has increased the potential for system security threats, including malware attacks. Malware is malicious software that can damage, disrupt, or steal data without the user's knowledge. This study aims to build a classification model to detect malware attacks based on the behavior of operating system processes using the Support Vector Machine (SVM) method. The dataset consists of 100,000 entries with 33 attributes related to process activities, such as CPU usage, memory, and context switches. The data is split into training and testing sets, followed by exploratory data analysis (EDA), data cleaning, feature selection based on correlation, and training of the model using a linear kernel SVM. Evaluation is carried out using a confusion matrix and metrics such as accuracy, precision, recall, and F1-score. The results show that the SVM model performed very well, achieving 99.57% accuracy, 99.76% precision, 99.38% recall, and a 99.57% F1-score. The model successfully distinguished between malware and normal processes with minimal misclassification. These findings demonstrate that SVM is effective in detecting malware based on system process behavior and has the potential to support the development of automated security systems capable of real-time threat detection.*

***Keywords:*** *classification, cyrber, malware, SVM, Security*

## PENDAHULUAN

Di era Industri 4.0, teknologi berkembang pesat, khususnya di bidang Network Security atau keamaan jaringan. Teknologi ini digunakan untuk mengamankan suatu jaringan di perusahaan atau organisasi. Salah satu teknologi Menurut informasi Dari TechCrunch laporan Gertner. yang sering digunakan masyarakat adalah android. Pada Tahun 2017, Gertner melaporkan bahwa penjualan ponsel pintar global meningkat sebesar 9 persen pada kuartal pertama, dengan 380 juta perangkat seluler terjual, dengan Android memimpin pasar dengan 84,1 persen. Menurut laporan Kementerian Komunikasi dan Informatika, jumlah pengguna aktif smartphone di Indonesia melebihi 200 juta orang pada tahun 2024 dan StatCounter melaporkan bahwa jumlah pengguna Android di Indonesia sebesar 91% pada tahun 2018 , sehingga dapat diasumsikan jumlah pengguna Android di Indonesia sekitar 91 juta orang( laporan serangan malware kaspersky). Jumlah

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

pengguna Android meningkat pesat sehingga sistem operasi Android menjadi sasaran serangan malware, sistem operasi yang diserang malware rusak, dan bahkan dengan niat yang lebih jahat, informasi penting dapat dicuri dari sistem dengan bantuan malware. . Malware atau malware seperti yang biasa dikenal adalah perangkat lunak yang diprogram untuk menyusup ke sistem operasi dan dapat merusak sistem bahkan mencuri data penting dari perangkat korban. Menuurt Lembaga Stnadar Keamanan Menurut konsep Privilege Escalation dalam keamanan siber Ada banyak jenis malware dan mereka memiliki cara kerja yang berbeda, misalnya serangan malware melalui aplikasi jahat yang menggunakan hak istimewa secara ilegal tanpa izin pengguna dan sistem operasi. Karena kerentanan terhadap serangan malware dan kerentanan yang memengaruhi pengguna Android, diperlukan analisis malware lebih lanjut. Pada Tahun 2019, perusahaan keamanan siber Kaspersky melaporkan 556.486 deteksi malware, dan pada Tahun 2020, Kaspersky juga melaporkan penurunan serangan malware di Indonesia sebesar 31,89% menjadi 378.973. Hal ini menjadikan Indonesia sebagai negara dengan jumlah ancaman malware yang terdeteksi terbanyak di Asia Tenggara dan tertinggi keempat di dunia Analisis statis dilakukan dengan mengekstraksi kode sumber malware dan kemudian memeriksa dan memahami perilaku berbahaya melalui kode sumber, data, dan binari, sehingga proses analisis statis tidak memerlukan malware untuk dijalankan. Analisis statis digunakan untuk mengekstrak fitur dan mengidentifikasi setiap aplikasi menggunakan ApkTool 2.0.3 . Berdasarkan data serangan malware dan jumlah pengguna Android yang besar, hal ini menjadi masalah, sehingga penelitian ini menerapkan sistem machine learning untuk menentukan metode klasifikasi yang menjamin akurasi tinggi dalam deteksi dini serangan malware. Beberapa penelitian sebelumnya menggunakan beberapa metode machine learning seperti Naive Bayes, Decision Tree, KNN dan metode lainnya seperti Lopez et al yang melakukan penelitian menggunakan beberapa metode machine learning dengan akurasi hingga 95%.Menurut (Wanli Sitorus et al., 2021) Melaporkan bahwa tahun 2018 pengguna aktif smartphone di Indonesia lebih dari 100 juta orang serta pada tahun yang sama data dari statcounter pengguna android di Indonesia sebanyak 90,85%. Dengan berkembangnya metode pembelajaran mesin untuk klasifikasi, penelitian ini mengadopsi sistem pembelajaran mesin untuk menentukan metode pembelajaran mesin yang memberikan kinerja matriks dalam deteksi dini serangan malware yang lebih baik dari penelitian . Penelitian ini menggunakan metode klasifikasi Support Vector Machine (SVM). Hasil penelitian ini diharapkan dapat membantu pengembang dan pengguna untuk mengunduh aplikasi. Aplikasi yang ingin dipublikasikan oleh individu dan perusahaan dapat menentukan apakah suatu aplikasi tidak aman atau tidak. Sehingga aplikasi lebih terlindungi dari serangan malware dan pengguna memiliki unduhan yang aman.

Tujuan dari penelitian ini adalah untuk membuat model klasifikasi yang dapat mengidentifikasi serangan malware berdasarkan perilaku proses sistem operasi saat menggunakan metode Support Vector Machine (SVM). Dataset yang digunakan memiliki 100.000 entri data yang memiliki 33 atribut yang

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

menunjukkan aktivitas proses seperti penggunaan CPU, memori, dan konteks pergeseran. Pembagian data menjadi data latih dan data uji, analisis data exploratory (EDA) untuk memahami karakteristik data, preprocessing data untuk membersihkan dan menstandarisasi atribut, seleksi fitur berdasarkan korelasi untuk mengurangi kompleksitas model, dan pengembangan dan pelatihan model klasifikasi menggunakan SVM dengan kernel linear. Menggunakan confusion matrix dan metrik evaluasi seperti akurasi, precision, recall, dan skor F1, model dievaluasi. Hasil pengujian menunjukkan bahwa model SVM yang dibangun memiliki performa yang sangat baik dengan akurasi sebesar 99,57%, ketepatan sebesar 99,76%, recall sebesar 99,38%, dan skor F1 sebesar 99,57%. Model ini juga memiliki kemampuan untuk membedakan proses yang merupakan malware dari proses normal dengan jumlah kesalahan klasifikasi yang sangat kecil. Hasilnya menunjukkan bahwa SVM dapat melakukan deteksi malware berbasis perilaku proses sistem dengan cukup baik. Penelitian ini dapat berkontribusi pada pengembangan sistem keamanan otomatis yang dapat mendeteksi ancaman secara real-time dan membantu memperkuat pertahanan sistem terhadap serangan siber

## Kajian Teori

### 2.1 Keamanan Jaringan

Menurut (Sinaga, Irmayani, and Hasibuan 2024) Dengan globalisasi dan perkembangan teknologi informasi yang cepat saat ini, keamanan jaringan menjadi masalah yang semakin penting. Semakin banyak organisasi dan individu yang bergantung pada infrastruktur jaringan untuk kegiatan sehari-hari dan operasional bisnis, membuat keamanan informasi menjadi sangat penting. Dengan berbagai jenis serangan yang terus berkembang, seperti malware, ransomware, dan serangan phishing, ancaman keamanan jaringan semakin sering dan semakin kompleks. Akibatnya, untuk mengatasi masalah keamanan ini, diperlukan pendekatan yang lebih canggih dan responsif.

### *2.2 Malware*

Menurut (Wanli Sitorus, Sukarno, and Mandala 2021) *Malicious* Software atau sering dikenal sebagai malware merupakan sebuah software yang diprogram agar dapat menyusup ke sebuah sistem operasi yang dapat merusak cara kerja sistem dan bahkan digunakan untuk mencuri data-data penting pada perangkat korban Selain itu Menurut (Putra and Komputer 2024) Malware adalah perangkat lunak berbahaya yang dirancang untuk merusak, mengganggu, atau mendapatkan akses tidak sah ke sistem komputer. Untuk itu maka diperlukan analisis untuk mendeteksi malware karena tujuan akhir dari analisis malware adalah menggambarkan secara tepat cara kerja sebuah malware. Deteksi SVM ( *Support Vector Machine*) terhadap serangann malware menggunakan algoritma pembelajaran mesin yang membedakan perangkat lunak berbahaya (malware) dan perilaku perangkat lunak yang sah. Malware ada dalam berbagai bentuk *script, code, activecontent,* dan perangkat lunak.

### 2.3 Algoritma *Machine Learning*

Menurut (Mahesh 2020) (Mahesh 2020) Pembelajaran mesin adalah bidang yang menyelidiki bagaimana

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

komputer dapat belajar tanpa diprogram secara eksplisit. Program catur Arthur Samuel terkenal. Pembelajaran mesin (ML) membantu mesin belajar menangani data dengan lebih baik. Kadang-kadang kita tidak dapat menafsirkan informasi ekstrak dari data setelah melihatnya.

### *2.3.1 Support Vector Machine*

Menurut (Tjahjadi and Santoso 2023) *Support Vector Machine* (SVM) adalah algoritma pembelajaran mesin klasifikasi yang berbasis pengawasan (*Supervised Learning*). Algoritma ini menggunakan dua garis vector (*Hyperplane*) dengan margin yang terbesar. Hyperplane adalah garis yang digunakan untuk memisahkan data dari berbagai kelompok. Margin adalah jarak antara hyperplane dengan titik data terdekat dari masing-masing kelompok. SVM mampu menyelesaikan masalah klasifikasi baik yang bersifat linear maupun non-linear.*Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik. Pertama kali dipresentasikan pada lokakarya tahunan pada tahun 1992.

### *2.3.2 Metode K-Nearest Neighbors*

Algoritma *K-Nearest Neighbor* (K- NN) adalah algoritma klasifikasi yang bergantung pada kedekatan suatu data dengan data lain. Dengan data berdimensi q, jarak antara dua data dapat dihitung. Nilai jarak ini digunakan untuk menentukan seberapa dekat atau mirip data uji dengan data latih. Nilai K pada K-NN menunjukkanK-data uji terdekat. Algoritma klasifikasi K-Nearest Neighbor digunakan untuk membuat sistem deteksi malware yang efektif dan akurat. (Halim and Anraeni 2021)

### 2.3.3 Metode *Random Forest*

Random Forest adalah pengembangan dari metode *Decision Tree* yang menggunakan beberapa *Decision Tree*. Setiap Decision Tree telah dilatih dengan sampel dan setiap atribut didistribusikan ke pohon yang dipilih dari subset atribut yang acak. memiliki beberapa kelebihan, termasuk kemampuan untuk meningkatkan akurasi dalam kasus data yang hilang dan untuk menahan keluaran, serta kemampuan untuk menyimpan data dengan efisien. mempunyai proses seleksi fitur yang dapat mengambil fitur terbaik untuk meningkatkan kinerja model klasifikasi. (Rayuwati, Husna Gemasih, and Irma Nizar 2022)

### 2.3.4 Metode *Naive Bayes*

memberikan penjelasan *Naive Bayes* untuk setiap kelas keputusan, yang menghitung probabilitas dengan asumsi bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek. Algoritma ini menganggap bahwa atribut obyek adalah independen. Jumlah frekuensi dari "master" tabel keputusan adalah jumlah kemungkinan yang terlibat dalam pembuatan perkiraan akhir.Jika dibandingkan dengan model klasifikasi lainnya, *Naive Bayes Classifier* memiliki kinerja yang sangat baik. (Rayuwati et al. 2022)

### 2.3.5 Metode Neural Networks

Jaringan Saraf Tiruan (JST) atau *Artificia Neural Network* (ANN) adalah prosesor terdistribusi besar-besaran secara paralel yang terdiri dari unit proses

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

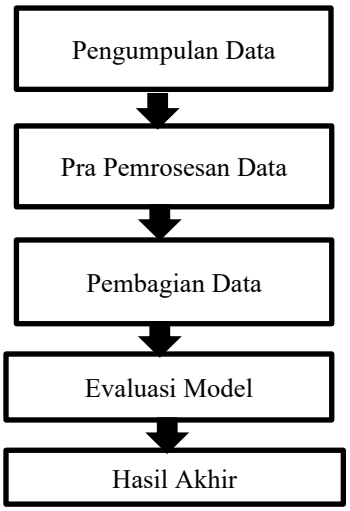
sederhana. ANN dapat menyimpan pengetahuan dalam bentuk pengalaman dan dapat digunakan untuk proses lain. ANN juga dapat digambarkan sebagai jaringan yang mirip dengan otak manusia yang diprogram untuk melakukan tugas tertentu.sederhana. ANN dapat menyimpan pengetahuan dalam bentuk pengalaman dan dapat digunakan untuk proses lain. ANN juga dapat digambarkan sebagai jaringan yang mirip dengan otak manusia yang diprogram untuk melakukan tugas tertentu. ANN dapat digunakan untuk memodelkan hubungan yang kompleks antara input dan output, yang kemudian dapat digunakan untuk menemukan pola dalam data.Biasanya, jaringan ini dijalankan dengan komponen elektronik atau disimulasikan melalui aplikasi komputer. (Rayuwati et al. 2022)

### 2.3.6 Metode Decision Tree

*Decision Tree* termasuk dalam kategori machine learning dan merupakan bagian dari supervised learning. *Decision tree* juga dikenal sebagai CART (Classification and Regression Tree), dan metode ini adalah kombinasi dari dua jenis pohon: *classification tree* dan *regression tree.* (Bintoro, Trisnawan, and Data 2023). Decision Tree biasanya menggunakan strategi pencarian secara top-down. Dalam proses mengklasifikasi data yang tidak diketahui, nilai atribut diuji dengan melacak jalur dari node akar hingga node akhir, yang dikenal sebagai daun. Setelah itu, nilai atribut akan diprediksi kelas yang dimiliki oleh data baru. (Bintoro et al. 2023)

## METODE PENELITIAN

Perencanaan penelitian adalah perencanaan penelitian yang tujuannya untuk memberikan pedoman dalam melakukan proses penelitian. Perencanaan penelitian menyediakan kerangka kerja dan alur kerja untuk seluruh proses penelitian. Berikut adalah desain penelitian dari penelitian Ini




**Gambar 1.** Metode Peneltian
Sumber: Data Penelitian, 2025

Berikut penjelasan untuk desain Penelitian dari penelitian ini:

1. **Pengumpulan Data**

Data dikumpulkan dari kumpulan sampel malware dan non-malware yang berasal dari Sumber tertentu Yaitu Kanggle. Data ini digunakan untuk melakukan pelatihan dan pengujian model.

2. **Pra Pemoresesan Data**

Pada titik ini, data dibersihkan dan disusun. Misalnya, hapus data duplikat atau tidak relevan; normalisasi nilai; menangani data kosong atau tidak lengkap; dan encoding variabel kategori jika diperlukan.

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

3. **Pembagian Data**

Data set yang telah diproses terdiri dari dua bagian yaitu data latih dan data uji

4. **Evaluasi Model**

Akurasi,precision,recall,dan f1 score ini adalah metrik yang digunakan untuk mengevaluasi tes

**5. Hasil Akhir**

Evaluasi menghasilkan kesimpulan tentang seberapa baik model SVM mendeteksi malware. Keputusan ini membantu menentukan efektivitas metode yang digunakan.

### 3.2 Metode Pengumpulan Data

Dalam penelitian ini, metode pengumpulan data dilakukan melalui Bebrapa metode, yaitu Survei, observasi exprimen, dan studi pustaka untuk mendukung pemahaman terhadap kebutuhan serta tantangan dalam sistem deteksi malware.

### 3.3 Metode Deteksi Malware Menggunakan Algoritma SVM

Berikut beberapa Metode Deteksi Malware Menggunakan Algoritma Support Vector Machine yang digunakan dalam penelitian ini:

**1. Pengumpulan Dataset**

Dalam penelitian ini dataset malware diambil dari sumber yaitu Kanggle. Dataset Yang digunakandalam proses pre-processing 100.000 data sampel untuk menemukan malware dalam penelitian ini. Jumlah fitur yg terdapt ada 33 fitur

**2. Data Processing**

Dalam deteksi serangan malware, dataset yang didapatkan dilakukan preprocessing yang dimulai dengan pembersihan data, yang menghilangkan duplikat data dan menangani nilai yang tidak ada. Selanjutnya, fitur dinamis dan statis, seperti ukuran file, entropi, panggilan API, dan izin, diekstraksi. Encoding mengubah semua fitur menjadi format numerik dan kemudian dinormalisasi untuk memastikan skala yang seragam. Untuk menjaga proporsi kelas, metode sampling stratified digunakan untuk membagi dataset menjadi dua bagian: 80% data latih dan 20% data uji.

***3. Explanatory Data Analysis***

Sebelum memasuki tahap pemodelan, analisis data eksplisit (EDA) dilakukan untuk mendapatkan pemahaman tentang karakteristik data. Pada titik ini, analisis distribusi data, identifikasi pola, pencarian anomali, dan analisis hubungan antar fitur dilakukan. EDA memastikan data yang digunakan tepat dan berkualitas tinggi. Untuk menunjukkan persebaran data, visualisasi seperti histogram, boxplot, dan scatterplot digunakan. Selain itu, dilakukan analisis korelasi fitur untuk menentukan fitur mana yang memiliki korelasi kuat dengan label malware.

***4. Feature Extraction***

Tujuan ekstraksi fitur adalah untuk mengekstrak informasi penting dari data mentah yang terkait dengan aktivitas malware. Studi ini mengekstraksi fitur dinamis seperti jumlah dan jenis panggilan API dan izin yang diminta oleh aplikasi serta fitur statis seperti ukuran file dan entropi. Teknik encoding mengubah fitur menjadi representasi numerik, sehingga dapat digunakan dalam proses pelatihan model pembelajaran Mesin.

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

### *5. Data Split*

Untuk mendukung proses pelatihan dan evaluasi model deteksi malware, dataset malware yang terdiri dari 100.000 sampel akan dibagi menjadi tiga bagian utama dalam penelitian ini. Set pelatihan akan mencakup Klasifikasi 80.000 sampel, atau 80% dari dataset secara keseluruhan. Bagian ini akan digunakan untuk melatih model untuk mengidentifikasi pola malware. Kedua, set validasi terdiri dari 20.000 bentuk testing sampel, atau sekitar 20 persen dari dataset secara keseluruhan, dan berfungsi untuk memvalidasi kinerja model selama proses pelatihan dan membantu dalam penyesuaian parameter model untuk meningkatkan akurasi.

### 6. Evaluation Model

Data uji digunakan untuk mengevaluasi model yang telah dibangun dalam mendeteksi serangan malware. Untuk memahami kesalahan klasifikasi yang terjadi, analisis confusion matrix digunakan. Metrik evaluasi yang digunakan termasuk akurasi, ketepatan, recall, skor F1, dan ROC-AUC. Model SVM yang dihasilkan dapat mendeteksi malware dengan tingkat keakuratan yang tinggi dan generalisasi yang baik, menurut evaluasi ini.Berikut adalah ukuran penilaian yang digunakan dalam peneltian ini.

$$\text{Accuracy} = \frac{T^{p+T_n}}{Tp+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{TP+FP}$$

$$\textit{Recall} = \frac{tp}{TP+FN}$$

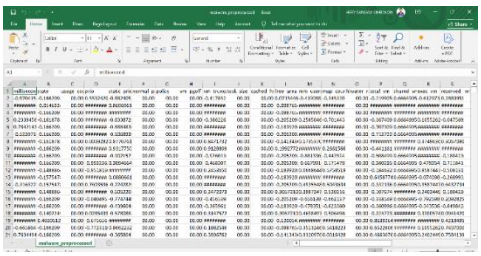
$$\text{F1 – Score} = \frac{2 \times Precission \times Recal}{Precission + recal}$$

## 4.1 HASIL DAN PEMBAHASAN

Bab ini membahas hasil dari proses pelatihan dan pengujian model klasifikasi untuk mengidentifikasi serangan malware menggunakan algoritma Support Vector Machine (SVM). Berdasarkan dataset sistem operasi yang tersedia, penelitian ini mencoba menentukan apakah suatu proses adalah malware atau benign (proses normal). Dataset ini memiliki 100.000 entri yang memiliki 35 atribut yang mewakili berbagai parameter sistem, termasuk penggunaan CPU, waktu proses, aktivitas memori, dan switch konteks, antara lain.

### 1. Hasil Pengumpulan Dataset

Pertama, penelitian ini mengumpulkan dan menyiapkan dataset yang terdiri dari 100.000 entri data dengan 33 Atribut



**Gambar 2.** Hasil Dataset Malware
Sumber: Data Penelitian, 2025

### 2. Hasil Processing

Data yang semula memiliki beberapa kolom bertipe objek telah berhasil dikonversi menjadi numerik, dibersihkan dari nilai yang tidak valid, dan seluruh fitur telah distandarisasi agar siap digunakan dalam proses klasifikasi. Fitur tersebut adalah sebagai berikut:

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

**Jurnal Comasie**

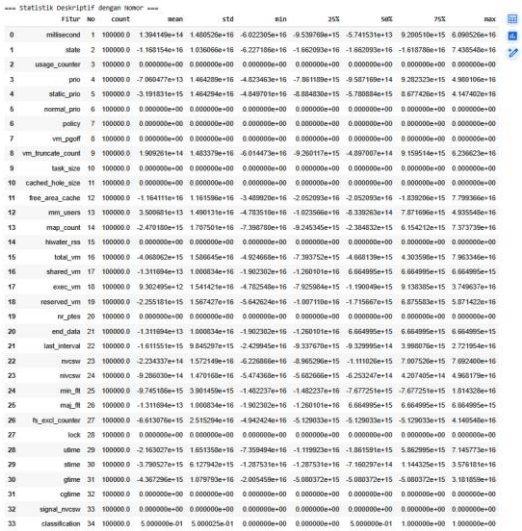
ISSN (Online) 2715-6265

| No | Nama Kolom | Jumlah Data (Non-Null) | Tipe Data |
|---|---|---|---|
| 0 | millisecond | 100000 | float64 |
| 1 | state | 100000 | int64 |
| 2 | usage_counter | 100000 | float64 |
| 3 | prio | 100000 | float64 |
| 4 | static_prio | 100000 | float64 |
| 5 | normal_prio | 100000 | float64 |
| 6 | policy | 100000 | float64 |
| 7 | vm_pgoff | 100000 | float64 |
| 8 | vm_truncate_count | 100000 | float64 |
| 9 | task_size | 100000 | float64 |
| 10 | cached_hole_size | 100000 | float64 |
| 11 | free_area_cache | 100000 | float64 |
| 12 | mm_users | 100000 | float64 |
| 13 | map_count | 100000 | float64 |
| 14 | hiwater_rss | 100000 | float64 |
| 15 | total_vm | 100000 | float64 |
| 16 | shared_vm | 100000 | float64 |
| 17 | exec_vm | 100000 | float64 |
| 18 | reserved_vm | 100000 | float64 |
| 19 | nr_ptes | 100000 | float64 |
| 20 | end_data | 100000 | float64 |
| 21 | last_interval | 100000 | float64 |
| 22 | nvcsw | 100000 | float64 |
| 23 | nivcsw | 100000 | float64 |
| 24 | min_flt | 100000 | float64 |
| 25 | maj_flt | 100000 | float64 |
| 26 | fs_excl_counter | 100000 | float64 |
| 27 | lock | 100000 | float64 |
| 28 | utime | 100000 | float64 |
| 29 | stime | 100000 | float64 |
| 30 | gtime | 100000 | float64 |
| 31 | cgtime | 100000 | float64 |
| 32 | signal_nvcsw | 100000 | float64 |
| 33 | classification | 100000 | int64 |

**Gambar 3**. Hasil Processing
Sumber: Data Penelitian, 2025

### *3.* Hasil *Explanatory Data Analysis*

Ada variasi skala nilai yang cukup besar, seperti yang ditunjukkan oleh statistik deskriptif untuk setiap fitur. Selain itu, ada nilai-nilai ekstrem (*outlier*) pada beberapa fitur, seperti utime, stime, dan gtime. Oleh karena itu, normalisasi data adalah langkah penting untuk memastikan bahwa semua fitur memiliki skala yang sama. Selain itu, seperti yang ditunjukkan oleh analisis korelasi, beberapa fitur memiliki korelasi yang cukup kuat terhadap label target klasifikasi. Ini dapat menjadi tanda penting dari proses klasifikasi. Sebaliknya, ditemukan korelasi tinggi antara fitur, yang mungkin menjadi alasan untuk mempertimbangkan untuk melakukan seleksi fitur untuk mengurangi redundansi. Hasil EDA ini membantu menentukan fitur yang relevan dan menyiapkan data untuk tahap pemodelan



=== Statistik Deskriptif dengan Nomor ===

| | Fitur | No | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | millisecond | 1 | 100000.0 | 1.394149e+14 | 1.480526e+16 | -6.022305e+16 | -9.539769e+15 | -5.741531e+15 | 9.200510e+15 | 6.090526e+16 |
| 1 | state | 2 | 100000.0 | -1.168154e+16 | 1.036066e+16 | -6.227186e+16 | -1.662093e+16 | -1.662093e+16 | -1.618786e+16 | 7.438548e+16 |
| 2 | usage_counter | 3 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 3 | prio | 4 | 100000.0 | -7.060477e+13 | 1.464289e+16 | -4.823463e+16 | -7.861189e+15 | -9.587169e+14 | 9.282323e+15 | 4.980106e+16 |
| 4 | static_prio | 5 | 100000.0 | -3.191831e+15 | 1.464294e+16 | -4.849701e+16 | -8.884830e+15 | -5.780884e+15 | 8.677426e+15 | 4.147402e+16 |
| 5 | normal_prio | 6 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 6 | policy | 7 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 7 | vm_pgoff | 8 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 8 | vm_truncate_count | 9 | 100000.0 | 1.909261e+14 | 1.483379e+16 | -6.014473e+16 | -9.260117e+15 | -4.897007e+14 | 9.159514e+15 | 6.236623e+16 |
| 9 | task_size | 10 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 10 | cached_hole_size | 11 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 11 | free_area_cache | 12 | 100000.0 | -1.164111e+16 | 1.161596e+16 | -3.489920e+16 | -2.052093e+16 | -2.052093e+16 | -1.839206e+15 | 7.799366e+16 |
| 12 | mm_users | 13 | 100000.0 | 3.500681e+13 | 1.490131e+16 | -4.783510e+16 | -1.033586e+16 | -8.339263e+14 | 7.871696e+15 | 4.935548e+16 |
| 13 | map_count | 14 | 100000.0 | -2.470180e+15 | 1.707501e+16 | -7.398780e+16 | -9.245345e+15 | -2.384832e+15 | 6.154212e+15 | 7.373739e+16 |
| 14 | hiwater_rss | 15 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 15 | total_vm | 16 | 100000.0 | -4.068062e+15 | 1.586645e+16 | -4.924668e+16 | -7.393752e+15 | -4.668139e+15 | 4.303596e+15 | 7.963346e+16 |
| 16 | shared_vm | 17 | 100000.0 | -1.311694e+13 | 1.000834e+16 | -1.902302e+16 | -1.260101e+16 | 6.664995e+15 | 6.664995e+15 | 6.664995e+15 |
| 17 | exec_vm | 18 | 100000.0 | 9.302495e+12 | 1.541421e+16 | -4.782548e+16 | -7.925984e+15 | -1.190049e+15 | 9.138385e+15 | 3.749637e+16 |
| 18 | reserved_vm | 19 | 100000.0 | -2.255181e+15 | 1.567427e+16 | -5.642624e+16 | -1.007110e+16 | -1.715667e+15 | 6.875583e+15 | 5.871422e+16 |
| 19 | nr_ptes | 20 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 20 | end_data | 21 | 100000.0 | -1.311694e+13 | 1.000834e+16 | -1.902302e+16 | -1.260101e+16 | 6.664995e+15 | 6.664995e+15 | 6.664995e+15 |
| 21 | last_interval | 22 | 100000.0 | -1.611551e+15 | 9.845297e+15 | -2.429945e+16 | -9.337670e+15 | -9.329995e+14 | 3.998076e+15 | 2.721954e+16 |
| 22 | nvcsw | 23 | 100000.0 | -2.234337e+14 | 1.572149e+16 | -6.226866e+16 | -8.965296e+15 | -1.111026e+15 | 7.007526e+15 | 7.092400e+16 |
| 23 | nivcsw | 24 | 100000.0 | -9.286030e+14 | 1.470168e+16 | -5.474368e+16 | -5.682666e+15 | -6.253247e+14 | 4.207405e+14 | 4.968179e+16 |
| 24 | min_flt | 25 | 100000.0 | -9.745186e+15 | 3.901459e+15 | -1.482237e+16 | -1.482237e+16 | -7.677251e+15 | -7.677251e+15 | 1.814328e+16 |
| 25 | maj_flt | 26 | 100000.0 | -1.311694e+13 | 1.000834e+16 | -1.902302e+16 | -1.260101e+16 | 6.664995e+15 | 6.664995e+15 | 6.664995e+15 |
| 26 | fs_excl_counter | 27 | 100000.0 | -6.613076e+15 | 2.515294e+16 | -4.942424e+16 | -5.129033e+15 | -5.129033e+15 | -5.129033e+15 | 4.140548e+16 |
| 27 | lock | 28 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 28 | utime | 29 | 100000.0 | -2.163027e+15 | 1.651358e+16 | -7.359494e+16 | -1.119923e+16 | -1.861591e+15 | 5.862995e+15 | 7.145773e+16 |
| 29 | stime | 30 | 100000.0 | -3.790527e+15 | 6.127942e+15 | -1.287531e+16 | -1.287531e+16 | -7.160297e+14 | 1.144325e+15 | 3.576181e+16 |
| 30 | gtime | 31 | 100000.0 | -4.367296e+15 | 1.079793e+16 | -2.065459e+16 | -5.080372e+15 | -5.080372e+15 | -5.080372e+15 | 3.181859e+16 |
| 31 | cgtime | 32 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 32 | signal_nvcsw | 33 | 100000.0 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| 33 | classification | 34 | 100000.0 | 5.000000e-01 | 5.000025e-01 | 0.000000e+00 | 0.000000e+00 | 5.000000e-01 | 1.000000e+00 | 1.000000e+00 |

**Gambar 4.** Hasil Statistik Deskriptif
Sumber: Data Penelitian,2025

### *4.* Hasil *Feature Extraction*

Ekstraksi fitur dilakukan untuk memilih dan menyaring fitur-fitur yang paling penting bagi tujuan klasifikasi. Tujuan dari proses ini adalah untuk meningkatkan efisiensi model, mengurangi kompleksitas data, dan mengurangi kemungkinan terjadinya overfitting selama pelatihan.

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

**Jurnal Comasie**

ISSN (Online) 2715-6265

```
=== Tabel Fitur Terpilih Berdasarkan Korelasi > 0.1 ===
   No              Fitur  Korelasi
0   1    free_area_cache  0.235423
1   2            min_flt  0.146533
2   3          map_count -0.105953
3   4           total_vm -0.109715
4   5              nvcsw -0.119344
5   6              utime -0.173988
6   7  vm_truncate_count -0.251970
7   8          shared_vm -0.324839
8   9           end_data -0.324839
9  10            maj_flt -0.324839
```

**Gambar 5.** Hasil Fitur terpilih Berdasarkan Korelasi
Sumber: Data Penelitian, 2025

5. **Hasil *Data Split***

Data dibagi menjadi data latih dan data uji untuk keperluan pelatihan dan evaluasi model.

1 to 2 of 2 entries Filter

| index | Jenis Data | Ukuran Data | Fitur | Distribusi Label 0 | Distribusi Label 1 |
|---|---|---|---|---|---|
| 0 | X_train | 80000 | 33 | 50% | 50% |
| 1 | X_test | 20000 | 33 | 50% | 50% |

Show 25 per page

**Gambar 6.** Hasil Data Split
Sumber: Data Penelitian, 2025

**6. Hasil Build Model Klasifikasi**

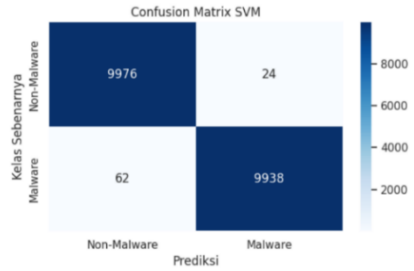
Pada titik ini, model klasifikasi dibangun menggunakan algoritma Support Vector Classifier (SVC) yang memiliki kernel linear. Data latihan yang diperoleh dari proses pembagian dataset sebelumnya digunakan untuk melatih model.

| index | Komponen | Nilai/Deskripsi |
|---|---|---|
| 0 | Algoritma | Support Vector Classifier (SVC) |
| 1 | Library | sklearn.svm.SVC |
| 2 | Parameter Utama | random_state=42 |
| 3 | Data latih (X_train) | 80000 baris = 33 fitur |
| 4 | Data target (y_train) | 80000 label |
| 5 | Tahapan | 1. Instansiasi model, 2. Pelatihan model |
| 6 | Status | Model berhasil dilatih |

**Gambar 7**. Hasil Build Model Klasifikasi
Sumber: Data Penelitian, 2025

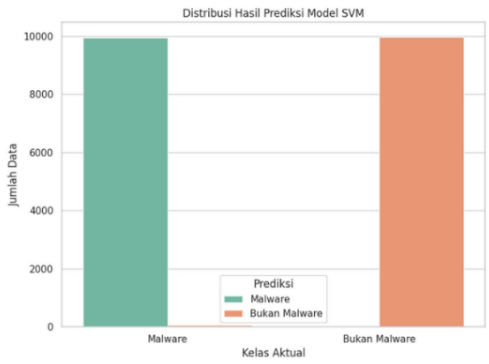
7. **Hasil Evaluation Model**

Untuk melakukan evaluasi model, confusion matrix digunakan. Hasil pengujian model SVM dengan confusion matrix adalah Sebagai berikut.




**Gambar 8.** *Confusion Matrix SVM*
Sumber**:** Data Penelitian, 2025

Gambar di bawah ini menunjukka distribusi label asli dan hasil prediksi. Ditunjukkan oleh dominasi prediksi yang sesuai dengan labe; aktual, model SVM mampu mengklasifikasikan data malware dan bukan malware dengan sangat baik. Dengan akurasi 99,57%, model ini memiliki jumlah kesalahan prediksi yang sangat kecil.




**Gambar 9.** Distribusi Hasil prediksi Model SVM
Sumber: Data Penelitian, 2025

Tabel di bawah ini menunjukkan hasil evaluasi model klasifikasi menggunakan metrik akurasi, ketepatan, recall, dan skor

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

Jurnal Comasie

ISSN (Online) 2715-6265

F1. Metrik-metrik ini menunjukkan kemampuan model untuk mengklasifikasikan data dengan tepat, terutama dalam membedakan antara kelas normal dan kelas malware.

**Tabel 1.** Tabel Evaluasi Model

| Metrik | Nilai |
|---|---|
| Akurasi | 0.9957 |
| Precision | 0.9976 |
| Recall | 0.9938 |
| F1-Score | 0.9957 |

Sumber: Data Penelitian, 2025

Hasil perhitungan confusion matrix berikut menunjukkan bagaimana model berhasil membedakan data normal dan malware berdasarkan prediksi dan label aktual.

Accuracy = $\frac{Tp+Tn}{Tp+TN+FP+FN} = \frac{9.938+9.976}{9.938+9.976+24+62}$ = $\frac{19.914}{20.000}$ = 0,9957 = 99,57%

Precision = $\frac{TP}{TP+FP} = \frac{9.938}{9.938+24} = \frac{9.938}{9.962}$ = 0,9976= 99,76%

Recall = $\frac{TP}{TP+FN} = \frac{9.938}{9.938+62} = \frac{9.938}{10.000}$ = 0,9938 = 99,38 %

F1-Score = $\frac{2 \times 0,9976 \times 0,9938}{0,9976+0,9938} = \frac{1,9828}{1,9914}$ = 0,9957

Hasil pengujian menunjukkan bahwa algoritma SVM memiliki kinerja yang sangat baik dengan tingkat akurasi 99,57%. Selain itu, nilai *Precission* 99,76%, *recall* 99,38%, dan skor F1 99,57% menunjukkan bahwa model mampu membedakan secara sangat tepat antara data serangan malware dan non-malware. Dari 20.000 data yang diuji, hanya ada 86 kesalahan klasifikasi.

## SIMPULAN

1. Menurut fitur aktivitas sistem seperti penggunaan CPU, memori, dan *switching* konteks, algoritma *Support Vector Machine* (SVM) berhasil mengklasifikasikan proses sistem operasi menjadi dua kelas: malware dan non-malware.
2. Dengan *accuracy* 99,57%, *Precision* 99,76%, *recall* 99,38%, dan skor F1 99,57%, model yang dibangun menggunakan SVM dengan kernel linear menunjukkan tingkat keakuratan dan konsistensi model dalam mengenali serangan malware.
3. Proses *preprocessing* data seperti normalisasi, konversi tipe data numerik, dan seleksi fitur berbasis korelasi meningkatkan
4. Untuk memastikan bahwa proses pelatihan dan evaluasi model berjalan secara adil dan representatif, distribusi data uji dan latihan yang seimbang (masing- masing 50% untuk setiap kelas diperlukan
5. Secara keseluruhan, metode klasifikasi berbasis SVM sangat cocok untuk sistem pendeteksi dini malware,
6. Terutama yang bergantung pada perilaku proses sistem operasi. Namun, untuk menjaga keakuratan, model harus diperbarui secara berkala dengan data terbaru seiring dengan perkembangan metode serangan malware.

## DAFTAR PUSTAKA

Bintoro, R. F. A., P. H. Trisnawan, and M. Data. 2023. “Deteksi Bot Network (BOTNET) Menggunakan Metode Decision Tree Dari Dataset CTU.” *… Teknologi Informasi Dan Ilmu …* 7(6):2921–30.

Halim, Andi Ainun Dzariah, and Siska

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

**Jurnal Comasie**

ISSN (Online) 2715-6265

Anraeni. 2021. "Analisis Klasifikasi Dataset Citra Penyakit Pneumonia Menggunakan Metode K-Nearest Neighbor (KNN)." *Indonesian Journal of Data and Science* 2(1):01–12. doi: 10.33096/ijodas.v2i1.23.

Mahesh, Batta. 2020. "Machine Learning Algorithms - A Review." *International Journal of Science and Research (IJSR)* 9(1):381–86. doi: 10.21275/art20203995.

Putra, Rizki Ramadhan, and Ilmu Komputer. 2024. "ANALISIS DATA MINING UNTUK DETEKSI MALWARE PADA." 1(6):1–17.

Rayuwati, Husna Gemasih, and Irma Nizar. 2022. "IMPLEMENTASI AlGORITMA NAIVE BAYES UNTUK MEMPREDIKSI TINGKAT PENYEBARAN COVID." *Jural Riset Rumpun Ilmu Teknik* 1(1):38–46. doi: 10.55606/jurritek.v1i1.127.

Sinaga, Novica Handayani, Deci Irmayani, and Mila Nirmala Sari Hasibuan. 2024. "Mengoptimalkan Keamanan Jaringan: Memanfaatkan Kecerdasan Buatan Untuk Meningkatkan Deteksi Dan Respon Ancaman." *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI* 7 Nomor 2(September):364–69.

Tjahjadi, Evan Valdis, and Budy Santoso. 2023. "Klasifikasi Malware Menggunakan Teknik Machine Learning." *Jurnal Ilmiah Ilmu Komputer* 2(1):60–70.

Wanli Sitorus, Yitshak, Parman Sukarno, and Satria Mandala. 2021. "Analisis Deteksi Malware Android Menggunakan Metode Support Vector Machine & Random Forest." *E-Proceeding of Engineering* 8(6):12500–518.

Bintoro, R. F. A., P. H. Trisnawan, and M. Data. 2023. "Deteksi Bot Network (BOTNET) Menggunakan Metode Decision Tree Dari Dataset CTU." … *Teknologi Informasi Dan Ilmu …* 7(6):2921–30.

Halim, Andi Ainun Dzariah, and Siska Anraeni. 2021. "Analisis Klasifikasi Dataset Citra Penyakit Pneumonia Menggunakan Metode K-Nearest Neighbor (KNN)." *Indonesian Journal of Data and Science* 2(1):01–12. doi: 10.33096/ijodas.v2i1.23.

Mahesh, Batta. 2020. "Machine Learning Algorithms - A Review." *International Journal of Science and Research (IJSR)* 9(1):381–86. doi: 10.21275/art20203995.

Putra, Rizki Ramadhan, and Ilmu Komputer. 2024. "ANALISIS DATA MINING UNTUK DETEKSI MALWARE PADA." 1(6):1–17.

Rayuwati, Husna Gemasih, and Irma Nizar. 2022. "IMPLEMENTASI AlGORITMA NAIVE BAYES UNTUK MEMPREDIKSI TINGKAT PENYEBARAN COVID." *Jural Riset Rumpun Ilmu Teknik* 1(1):38–46. doi: 10.55606/jurritek.v1i1.127.

Sinaga, Novica Handayani, Deci Irmayani, and Mila Nirmala Sari Hasibuan. 2024. "Mengoptimalkan Keamanan Jaringan: Memanfaatkan Kecerdasan Buatan Untuk Meningkatkan Deteksi Dan Respon Ancaman." *Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI* 7 Nomor 2(September):364–69.

Tjahjadi, Evan Valdis, and Budy Santoso. 2023. "Klasifikasi Malware Menggunakan Teknik Machine Learning." *Jurnal Ilmiah Ilmu Komputer* 2(1):60–70.

Terbit *online* pada laman web jurnal : http://ejournal.upbatam.ac.id/index.php/comasiejournal

**Jurnal Comasie**

ISSN (Online) 2715-6265

Wanli Sitorus, Yitshak, Parman Sukarno, and Satria Mandala. 2021. "Analisis Deteksi Malware Android Menggunakan Metode Support Vector Machine & Random Forest." *E-Proceeding of Engineering* 8(6):12500–518.


| | |
|---|---|
|  | Hery Sanjaya Simbolon, merupakan mahasiswa Prodi Teknik Informatika Universitas Putera Batam |
|  | Andi Maslan, S.T,M.SI,P.HD. merupakan Dosen Prodi Teknik Informatika Universitas Putera Batam, yang aktif dan *expert* di bidang Teknik Informatika |