KLASIFIKASI MALWARE MENGGUNAKAN TEKNIK MACHINE LEARNING

Oleh

EVAN VALDIS TJAHJADI

T3119066

SKRIPSI

Untuk memenuhi salah satu syarat ujian

Guna memperoleh gelar Sarjana

PROGRAM SARJANA

TEKNIK INFORMATIKA

UNIVERSITAS ICHSAN GORONTALO

GORONTALO

2023

PERSETUJUAN SKRIPSI

KLASIFIKASI MALWARE MENGGUNAKAN TEKNIK MACHINE LEARNING

Oleh

EVAN VALDIS TJAHJADI
T3119066

SKRIPSI

Untuk memenuhi salah satu syarat ujian
guna memperoleh gelar Sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing
Gorontalo,9 Mei 2023

Pembimbing I	Pembimbing II
Budy Santoso, S.Kom.,M.Eng	**Serwin, M.Kom**
NIDN.0908048403	**NIDN.0918078802**

PENGESAHAN SKRIPSI

KLASIFIKASI MALWARE MENGGUNAKAN TEKNIK MACHINE LEARNING

Oleh

EVAN VALDIS TJAHJADI

T3119066

Diperkasa oleh Panitia Ujian Strata Satu (S1)

Universitas Ichsan Gorontalo

1. Ketua Penguji
   Sudirman Melangi, S.Kom.,M.Kom .....
2. Anggota
   Sunarto Taliki, S.Kom.,M.Kom .....
3. Anggota
   Warid Yunus, S.Kom.,M.Kom .....
4. Anggota
   Budy Santoso, S.Kom., M.Eng .....
5.. Anggota
   Serwin, S.Kom.,M.Kom .....

Mengetahui

Dekan Fakultas Ilmu Komputer

Irvan Abraham Salihi, M.Kom

NIDN. 0928028101

Ketua Program Studi

Sudirman S Panna, M.Kom

NIDN. 0924038205

PERNYATAAN SKRIPSI

Dengan ini saya menyatakan bahwa:

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorotralo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Tim Pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah dipublikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pada dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo

Gorontalo,9 Mei 2023

Yang membuat pernyataan

Evan Valdis Tjahjadi

ABSTRAK

EVAN VALDIS TJAHJADI. T3119066. KLASIFIKASI MALWARE MENGGUNAKAN TEKNIK MACHINE LEARNING

Jaringan komputer yang terhubung dengan internet dapat mengakses informasi dariseluruh dunia dengan sangat mudah. Namun, koneksi antara jaringan dan Internet justru meningkatkan potensi kegagalan sistem. Salah satu metode yang bisa digunakan pada machine learning merupakan metode algoritma random forest. Random forest merupakan salah satu metode pada machine learning yang digunakan untuk memecahkan masalah klarifikasi, Berdasarkan permasalahan tersebut perlu dilakukan klasifikasi malware yang datanya diambil dari dataset malware agar dapat memudahkan dalam mempelajari dan membedakan jenis malware. Proses terdiri dari pengumpulan dataset, pre proseccing, melatih machine learning dan melakukan pengujian performa model atau kinerja. Penelitian ini bertujuan untuk mengetahui performa atau kinerja Machine Learning menggunakan algoritma random forest untuk klasifikasi malwarerandom forest. Pada proses ini pra pemrosesan data dilakukan dengan menginstal beberapa library python. Pandasadalah library python open source yang biasanya digunakan untuk kebutuhan data analisis. Model dilatih pada dataset dengan berbagai fitur dan hasilnya menunjukanakurasi yang tinggi sebesar 99%, yaitu berupa proses menganalisis sekumpulan data untuk meringkas karakteristik utamanya agar pengguna lebih memahami dataset yang akan digunakan. Model random forest memberikan hasil yang sangat baik tanpa preprocessing pada data. Hasilnya bagus meskipun datanya tidak seimbang. Tidak perlu menggunakan teknik apapun untuk menyeimbangkannya. Penskalaan/skaling tidak perlu dilakukan, model andom forest adalah model partisirekursif yang bergantung pada partisi data karena ia bekerja pada pemisahan nilai fitur dan tidak melkukan perhitungan di dalamnya. Hasil penelitian menunjukkan bahwa model memiliki presisi 0,99.

Kata kunci: klasifikasi malware, machine learning, metode Random Forest

ABSTRACT

EVAN VALDIS TJAHJADI. T3119066. MALWARE CLASSIFICATION USING MACHINE LEARNING TECHNIQUES

Computer networks connected to the Internet can access information from all over the world very easily. However, the connection between the network and the Internet increases the potential for system failure. One of the methods that can be used in machine learning is the random forest algorithm method. Random forest is one of the methods in machine learning that is used to solve clarification problems. Based on the problems, it is necessary to classify malware where data is taken from malware datasets to make it easier to learn and distinguish the types of malware. The process consists of collecting datasets, pre-processing, training machine learning, and testing model performance. This study aims to find out the performance of Machine Learning using a random forest algorithm for malware random forest classification. In this process, pre-processing of data is done by installing several Python libraries. Pandas is an open-source Python library that is usually used for data analysis needs. The model is trained on a dataset with various features and the results show a high accuracy of 99%. The random forest model provides excellent results without preprocessing the data. The results are good even if the data is not balanced. There is no need to use any technique to balance it. Scaling is not necessary. The random forest model is a recursive partitioning model that depends on data partitioning as it works on splitting the feature values and does not perform any calculations in it. The results indicate that the model has a precision of 0.99.

Keywords: malware classification, machine learning, Random Forest method

ABSTRAK

EVAN VALDIS TJAHJADI. T3119066. KLASIFIKASI MALWARE MENGGUNAKAN TEKNIK MACHINE LEARNING

Jaringan komputer yang terhubung dengan internet dapat mengakses informasi dariseluruh dunia dengan sangat mudah. Namun, koneksi antara jaringan dan Internet justru meningkatkan potensi kegagalan sistem. Salah satu metode yang bisa digunakan pada machine learning merupakan metode algoritma random forest. Random forest merupakan salah satu metode pada machine learning yang digunakan untuk memecahkan masalah klarifikasi, Berdasarkan permasalahan tersebut perlu dilakukan klasifikasi malware yang datanya diambil dari dataset malware agar dapat memudahkan dalam mempelajari dan membedakan jenis malware. Proses terdiri dari pengumpulan dataset, pre proseccing, melatih machine learning dan melakukan pengujian performa model atau kinerja. Penelitian ini bertujuan untuk mengetahui performa atau kinerja Machine Learning menggunakan algoritma random forest untuk klasifikasi malwarerandom forest. Pada proses ini pra pemrosesan data dilakukan dengan menginstal beberapa library python. Pandasadalah library python open source yang biasanya digunakan untuk kebutuhan data analisis. Model dilatih pada dataset dengan berbagai fitur dan hasilnya menunjukanakurasi yang tinggi sebesar 99%, yaitu berupa proses menganalisis sekumpulan data untuk meringkas karakteristik utamanya agar pengguna lebih memahami dataset yang akan digunakan. Model random forest memberikan hasil yang sangat baik tanpa preprocessing pada data. Hasilnya bagus meskipun datanya tidak seimbang. Tidak perlu menggunakan teknik apapun untuk menyeimbangkannya. Penskalaan/skaling tidak perlu dilakukan, model andom forest adalah model partisirekursif yang bergantung pada partisi data karena ia bekerja pada pemisahan nilai fitur dan tidak melkukan perhitungan di dalamnya. Hasil penelitian menunjukkan bahwa model memiliki presisi 0,99.

Kata kunci: klasifikasi malware, machine learning, metode Random Forest

KATA PENGANTAR

Puji Tuhan, penulis dapat menyelesaikan skripsi ini dengan judul: **"Klasifikasi Malware menggunakan Teknik Machine Learning"**, untuk memenuhi salah satu syarat penyusunan Skripsi Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo. Penulis Menyadari sepenuhnya bahwa usulan penelitian ini tidak mungkin terwujud tanpa bantuan dan dorongan dari berbagai pihak, baik bantuan moril maupun materil. Untuk itu dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan banyak terima kasih dan penghargaan yang setinggi-tingginya kepada :

1. Ibu Dr. Juriko Abdussamad, M.Si selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar Latjokke, M.Si selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan Abraham Salihi, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Ichsan Gorontalo
5. Ibu Irma Surya Kumala Idris, M.Kom selaku Wakil Dekan II Bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo
6. Bapak Sudirman S Panna, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Budy Santoso, S.Kom.,M.Eng, selaku Pembimbing I;
8. Bapak Serwin, M.Kom, selaku Pembimbing II;
9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang telah mendidik dan mengajarkan berbagai disiplin ilmu kepada penulis;

10. Kedua Orang Tua saya yang tercinta, atas segala kasih sayang, jerih payah dan doa restunya dalam membesarkan dan mendidik penulis;
11. Rekan-rekan seperjuangan yang telah banyak memberikan bantuan dan dukungan moril yang sangat besar kepada penulis;
12. Kepada semua pihak yang ikut membantu dalam penyelesaian proposal/skripsi yang tak sempat penulis sebutkan satu-persatu.

Semoga Tuhan Yang Maha Esa melimpah kan balasan atas jasa-jasa mereka kepada kami. Penulis menyadari sepenuhnya bahwa apa yang telah dicapai ini masih jauh dari kesempurnaan dan masih banyak terdapat kekurangan. Oleh karena itu, penulis sangat mengharapkan adanya kritik dan saran yang konstruktif, Akhirnya penulis berharap semoga hasil yang telah dicapai ini dapat bermanfaat bagi kita semua, Aamiin,

Gorontalo, 9 Mei 2023

Penulis

DAFTAR ISI

BAB IDAFTAR GAMBAR

BAB IIDAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang

Jaringan komputer yang terhubung dengan internet dapat mengakses informasi dari seluruh dunia dengan sangat mudah. Namun, koneksi antara jaringan dan Internet justru meningkatkan potensi kegagalan sistem. Komputer menjadi mudah diakses dan ada risiko intrusi oleh orang yang ingin mengakses komputer Anda. Ini mengancam atau menyerang sistem komputer. Ini sangat berbahaya untuk sistem komputer perusahaan yang berisi data sensitif dan hanya dapat diakses oleh orang tertentu.

Ancaman virus dan malware yang dapat merusak komputer, server atau jaringan komputer harus diantisipasi. Internet digunakan sebagai media sosialisasi, dimana perilaku memiliki dampak yang besar. Indonesia sendiri merupakan salah satu negara dengan jumlah serangan malware tertinggi di Asia Pasifik [1].

Serangan terhadap keamanan sistem informasi *(security attacks)* sering terjadi. Kejahatan komputer di dunia maya *(cybercrime)* dilakukan oleh kelompok dengan tujuan menembus keamanan sistem untuk menemukan, memperoleh, memodifikasi atau bahkan menghapus yang sudah ada. Informasi di sistem sangat membutuhkannya. Ada berbagai jenis serangan yang dapat dilakukan penyerang, termasuk pemadaman terjadi ketika informasi yang dikirimkan melalui jaringan dapat dihancurkan, terputus di tengah jalan, dan tidak dapat mencapai tujuannya. Serangan ini bertujuan untuk mendapatkan informasi [2].

Malware ini menyandang nama lengkap malware. Malware adalah istilah umum untuk setiap program atau perangkat lunak yang dirancang untuk menyusup atau merusak sistem komputer. Ada juga, beberapa pengguna internet tidak terbiasa dengan istilah malware. Secara rutin menyebut virus

sebagai virus, dan media banyak menggunakan istilah itu, tetapi itu tidak sepenuhnya akurat. Program diklasifikasikan sebagai berbahaya karena tujuan pembuatannya, bukan karena properti khusus yang dimilikinya. Malware mencakup virus, worm, trojan horse, sebagian besar rootkit, spyware, adware, dan program berbahaya lainnya yang dapat membahayakan komputer. "Malware" adalah program komputer yang dibuat untuk maksud dan tujuan tertentu penciptanya, mencari kerentanan dalam program itu. Malware biasanya dibuat untuk memperkenalkan atau merusak perangkat lunak atau sistem operasi [3].

Malware berkembang pesat, dengan banyak jenis keluarga malware baru bermunculan. Salah satu teknik yang dapat digunakan untuk mendeteksi jenis malware adalah klasifikasi malware menggunakan machine learning. Klasifikasi adalah teknik pembelajaran mesin yang digunakan untuk mengidentifikasi atau memprediksi kategori data baru. Salah satu algoritma klasifikasi machine learning adalah algoritma random forest [4].

Machine Learning adalah teknik untuk memproses data dan memecahkan masalah. Pembelajaran mesin menggunakan algoritme yang secara iteratif belajar dari data untuk menjelaskan atau memprediksi hasil operasi yang dilakukan. Penggunaan machine learning diterapkan dengan cara mengolah data dalam model yang memiliki kegunaan tertentu, mis. B. tingkat akurasi prediksi yang tinggi. Model pembelajaran mesin dapat bersifat prediktif, untuk memprediksi kejadian di masa depan, atau deskriptif, untuk mengekstrak informasi dari data. Pembelajaran mesin yang diawasi digunakan untuk mendeteksi situs web berbahaya karena diharapkan dapat memprediksi situasi di masa depan. Kinerja yang dihasilkan dalam mendeteksi berbahaya [5].

Permasalahan dari malware lebih berbahaya bagi instansi pemerintah dan organisasi dibandingkan untuk pengguna pribadi. Serangan malware sering kali merusak melalui email, hasil download sebuah program di internet, program - program yang sudah terinfeksi malware berbahaya. Beragam tujuan yang dilakukan oleh pelaku untuk melakukan aktivitas berbahaya yang dapat

merugikan orang lain seperti penyadapan, mendapatkan hak akses komputer tanpa sepengetahuan dan izin pemiliknya, memanipulasi transaksi bank untuk mendapatkan keuntungan, pencarian data pribadi, kerugian finansial dan merusak reputasi organisasi [4].

Berbagai kategori pendekatan telah di ususlkan sudah termasuk signature based, yang memerlukan aturan yang dibuat secara manual untuk menyimpulkan data yang relevan untuk deteksi, dan machine learning yang secara otomatis menyimpulkan alasan tentang data malware dan benignware agar sesuai dengan parameter model deteksi. Namun dalam beberapa tahun terakhir machine learning telah mencapai deteksi tingkat tinggi dengan tingkat *low false positive rates* tanpa membebani generasi manusia yang dibutuhkan oleh metode manual. Karena jumlah malware yang dihasilkan setiap hari meningkat, kebutuhan akan cara yang lebih otomatis dan cerdas untuk memepelajari, beradaptasi dan mendeteksi malware menjadi semakin penting. Banyak solusi yang ditawarkan oleh perusahaan dan pengguna komputer lainnya untuk mencegah serangan malware yang berbahaya. Yang terbaru ialah kapabilitas machine learning untuk menangkal dan melawan malware secara real time. *Machine Learning* merupakan teknik yang digunakan untuk memproses data untuk memecahkan masalah. *Machine Learning* menggunakan algoritme yang secara iteratif belajar dari data untuk menjelaskan atau memprediksi hasil operasi yang dilakukan. Penggunaan machine learning diterapkan dengan mengolah data dalam model yang memiliki kegunaan tertentu, seperti tingkat akurasi prediksi yang tinggi. Model pembelajaran mesin dapat bersifat prediktif, untuk memprediksi kejadian di masa depan, atau deskriptif, untuk mengekstrak informasi dari data. *Machine Learning* yang diawasi digunakan untuk mendeteksi perangkat lunak berbahaya karena diharapkan dapat memprediksi situasi di masa depan. Kinerja deteksi malware yang dicapai dapat diperoleh dari evaluasi pembelajaran mesin berdasarkan *confusion matrix* [5].

Salah satu metode yang bisa digunakan pada machine learning merupakan metode algoritma random forest. Random forest merupakan salah

satu metode pada machine learning yang digunakan untuk memecahkan masalah klarifikasi. Metode ini merupakan metode pohon gabungan yang berasal dari metode *classification and regression tree* (CART) dan didasarkan pada teknik pohon keputusan (decision tree) berdasarkan teknologi pohon keputusan dapat mengatasi masalah non linier. Random Forest merupakan algoritma pembelajaran mesin untuk mengklasifikasikan sejumlah besar data. Random Forest dapat mengklasifikasikan data dengan atribut yang tidak lengkap dan berguna untuk mengklasifikasikan data sampel yang besar. Proses klasifikasi hutan acak membagi (split) data sampel yang ada menjadi pohon keputusan acak. Ketika sebuah pohon terbentuk, itu berisi akar, simpul internal (cabang), dan daun (hasil kelas). Kemudian, Random Forest digunakan untuk mendapatkan pohon terbaik, sehingga menghasilkan akurasi tertinggi dalam klasifikasi data [6].

Berdasarkan permasalahan tersebut perlu dilakukan klasifikasi malware yang datanya diambil dari dataset malware agar dapat memudahkan dalam mempelajari dan membedakan jenis malware. Proses terdiri dari pengumpulan dataset, pre proseccing, melatih machine learning dan melakukan pengujian performa model atau kinerja. Data yang di ambil merupakan data public atau dataset malware dan di proses menggunakan machine learning dengan algoritma random forest. Parameter pengujian yang diukur berupa kecepatan dan akurasi dari machine learning dengan algoritma random forest dalam melakukan klasifikasi malware

Dengan demikian, berdasarkan kajian di atas maka peneliti bermaksud untuk melakukan penelitian ini diharapkan dapat memberikan sebuah solusi karena ingin mengetahui deteksi malware dengan menggunakan teknik machine learning

Berdasarkan berbagai pemaparan di atas, dapat disimpulkan peneliti tertarik untuk mengangkat sebuah penelitian dengan judul: "**Klasifikasi Malware menggunakan teknik Machine Learning".** Penelitian ini diharapkan membantu pengguna komputer dalam mengatasi Malware.

1.2 Identifikasi Masalah

Berdasarkan latar belakang di atas maka dapat di identifikasi masalah

1. keamanan jaringan menjadi perhatian saat ini sehingga perlu dilakukan penelitian – penelitian selanjutnya.
2. Klasifikasi malware dapat dilakukan dengan menggunakan teknik Machine Learning dengan metode algoritma random forest

1.3 Rumusan Masalah

Bagaimana performa atau kinerja Machine Learning menggunakan algoritma random forest untuk klasifikasi malware?

1.4 Tujuan Penelitian

Untuk mengetahui performa atau kinerja Machine Learning menggunakan algoritma random forest untuk klasifikasi malware

1.5. Manfaat Penelitian

1.5.1. Manfaat Teoritis

Memberikan masukkan bagi perkembangan ilmu pengetahuan tentang teknologi, khususnya dalam bidang ilmu computer, yaitu berupa penerapan pendekatan machine learning menggunakan algoritma random forest terhadap keamanan jaringan computer.

1.5.2. Manfaat Praktis

Memberikan pemikiran, karya, bahan pertimbangan atau solusi dalam mengatasi malware dengan pendekatan machine learning menggunakan algoritma random forest kepada pengguna computer agar lebih memperketat keamanan jaringan komputer.

BAB II
LANDASAN TEORI

2.1 Tinjauan Studi

NO	PENELITI	JUDUL	HASIL
1	Togu Novriansyah Turnip, Chatrine Febriyanti Manurung, Yogi Septian Lubis 2023 [1].	Klasifikasi Malware Android Aplikasi Menggunakan Random Forest Berdasarkan Fitur Statik	"Pada penelitian ini, mampu mengklasifikasikan malware ke dalam 13 kelas jenis malware dengan menerapkan algoritma Random Forest dengan perolehan akurasi model terbaik mencapai 92,26%."
2	Raden Budiarto Hadiprakoso, Wahyu Rendra Aditya, Febriora Nevia Pramitha, 2022 [5].	Analisis statis Deteksi Malware android menggunakan algoritma superviset machine learning	": Hasil yang ditampilkan membuktikan bahwa model kami memberikan pencapaian akurasi yang tinggi yakni 96,94% dengan menggunakan algoritma SVM. Hasil pengujian kurva ROC juga menunjukkan bahwa model memiliki luas AUC di kisaran 95%. Untuk pengembangan penelitian lebih lanjut, algoritma deep learning dapat diselidiki untuk meningkatkan kemampuan deteksi malware pada platform Android."

NO	PENELITI	JUDUL	HASIL
3.	Yitshak Wanli Sitorus, Parman Sukarno, Satria Mandala, 2021 [7].	Analisis deteksi malware android menggunakan metode support vector machine dan random forest	"Penelitian ini dilakukan berdasarkan tingginya serangan malware yang terjadi di Indonesia. Pada tahun 2019, terdeteksi 556.486 malware android di Indonesia, menjadikan Indonesia menduduki posisi pertama serangan malware terdeteksi terbanyak se-Asia Tenggara dan posisi keempat secara global. Penelitian ini melakukan deteksi malware dengan menggunakan pendekatan machine learning dalam proses klasifikasi."

2.2. Tinjauan Pustaka

2.2.1. Keamanan Jaringan

Keamanan jaringan merupakan suatu cara pengamanan jaringan agar dapat terhindar dari berbagai ancaman yang berasal dari jaringan luar dan bertujuan untuk merusak atau mencuri data. Oleh karena itu, Anda harus mengambil tindakan pencegahan untuk menghadapi ancaman ini. Pertahanan dapat diimplementasikan melalui firewall, deteksi melalui IDS (Intrusion Detection System) dan kombinasi keduanya melalui IPS (Intrusion Prevention System).

Jaringan komputer merupakan kumpulan sejumlah besar komputer otonom yang saling berhubungan. Secara umum jaringan komputer dapat digambarkan sebagai kumpulan komputer yang dihubungkan oleh media perantara. Media

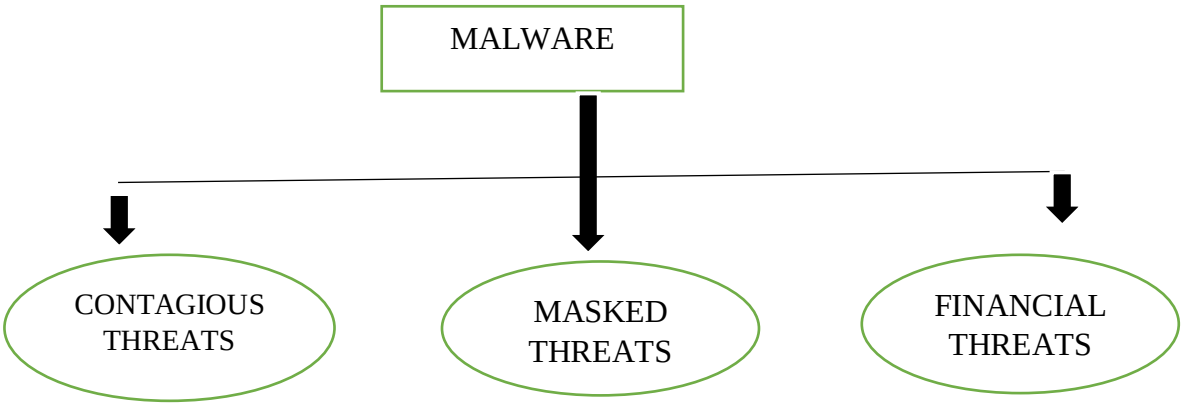
perantara dapat berupa kabel ataunirkabel. Informasi berupa data mengalir dari satu komputer ke komputer lainnya, dan setiap komputer yang terhubung dapat bertukar data atau berbagi perangkat keras [7].

2.2.2. Definisi Malware

Malware merupakan perangkat lunak berbahaya yang diprogram untuk merusak atau mengakses sistem komputer tanpa sepengetahuan pemilik sistem. Virus, worm, Trojan horse, keyloggers, spyware, dan ransomware merupakan contoh malware yang paling umum digunakan. Istilah seperti "worm", "virus", dan "Trojan horse" digunakan untuk mengkategorikan malware yang menunjukkan perilaku jahat serupa [8].

2.2.3. Klasifikasi Malware

Malware dapat dibagi menjadi beberapa kelas dan kategori. Secara umum, mereka dikategorikan berdasarkan proses dan respons berdasarkan desain dan evolusi malware. Gambar 2.2. menunjukkan berbagai jenis malware [8].



Gambar 2.1. klasifikasi Malware [6]

a) *Contagious Threats* (Ancaman yang Menular)

Virus dan worm merupakan kategori ancaman menular. Berdasarkan karakteristik dan mekanisme sistem penginfeksi malware yang ditunjukkan pada Tabel 2.3.

2.2.4. Analisis Malware

Analisis malware merupakan dasar untuk mengatur informasi, Informasi ini dapat dikembangkan sebagai tanda tangan untuk mendeteksi infeksi malware. Tujuan akhir dari analisis malware merupakan untuk menjelaskan dengan tepat bagaimana malware bekerja, menurut Adenansi & Novarina, 2017. Ada tiga teknik yang dapat dilakukan: analisis statis, dinamis, dan hybrid. Uraian lebih jelas mengenai teknik pendeteksian malware merupakan sebagai berikut:

a) Analisis statis

Analisis statis merupakan proses melakukan analisis perangkat lunak tanpa melakukannya. Analisis statis menggunakan alat rekayasa balik untuk menguraikan aplikasi dan merekonstruksi kode sumber dan algoritme yang dibuat oleh aplikasi. Analisis statis dapat dilakukan oleh penganalisis, *debugger,* dan *disassembler*. Berbagai teknik analisis statis antara lain [9].

1. Metode deteksi tanda tangan

Teknik ini juga dikenal sebagai pencocokan pola atau string atau masking atau sidik jari. Tanda tangan merupakan sekumpulan program yang dimasukkan ke dalam aplikasi oleh pembuat malware yang secara unik mengidentifikasi bagian tertentu dari malware. Deteksi malware di kode Anda. Detektor Malware mencari tanda tangan yang telah ditentukan sebelumnya dalam kode Anda [9].

2. Teknologi deteksi heuristik

Teknik ini juga dikenal sebagai teknik proaktif. Teknik ini mirip dengan teknik berdasarkan tanda tangan tertentu dalam kode, di mana pendeteksi malware kini mencari perintah atau petunjuk yang tidak ada dalam program aplikasi.

Akibatnya, varian malware baru yang belum ditemukan dapat dengan mudah dideteksi di sini. Berbagai teknik analisis heuristik antara lain [9].

a) Analisis heuristik berbasis file

Analisis heuristik berbasis file juga dikenal sebagai analisis file. File dengan teknik ini dianalisis secara detail untuk konten, tujuan, penanganan file, dll dan dianggap berbahaya jika file tersebut berisi perintah untuk menghapus atau merusak file lain [8].

b) Analisis heuristik berbasis bobot

Analisis heuristik berbasis bobot merupakan teknik lama. Setiap aplikasi diberi bobot sesuai dengan potensi risikonya. Jika nilai tertimbang melebihi ambang terdeteksi, aplikasi berisi kode bahaya [9]..

c) Analisis heuristik berbasis aturan

Analisis di sini melakukan ekstraksi aturan yang mengidentifikasi aplikasi. Aturan-aturan ini diperiksa terhadap aturan yang ditetapkan sebelumnya. Jika tidak ada aturan yang cocok, aplikasi tersebut mengandung malware [9].

d) Analisis tanda tangan secara umum;

Malware subspesies berarti malware berperilaku berbeda tetapi termasuk dalam keluarga yang sama dengan "kembar identik". Teknik ini menggunakan definisi antivirus yang telah ditentukan sebelumnya untuk menemukan varian malware baru analisis Statis [9].

Analisis statistiknya cepat dan andal serta menangkap struktur kode program yang diperiksa. Jika analisis statis dapat menghitung perilaku mekanisme keamanan masa depan kerugian Analisis Statis [9].

Analisis statis tidak dapat menganalisis malware yang tidak dikenal. Kode sumber untuk banyak aplikasi tidak tersedia. Melakukan analisis statis

membutuhkan peneliti untuk memiliki pemahaman yang mendalam tentang fungsionalitas sistem operasi [9].

e) Analisis Dinamis

Proses menganalisis perilaku atau tindakan yang dilakukan oleh aplikasi saat sedang berjalan disebut analisis dinamis. Analisis dinamis dapat dilakukan dengan memantau pemanggilan fungsi, melacak informasi, melakukan analisis parameter fungsi, dan melacak pernyataan. Mesin virtual atau kotak pasir biasanya digunakan untuk analisis ini. Aplikasi yang mencurigakan biasanya berjalan di lingkungan virtual. Jika suatu aplikasi berperilaku tidak normal, itu diklasifikasikan sebagai berbahaya. Perangkat lunak pemblokiran perilaku untuk memblokir perilaku berbahaya dalam program dari serangan malware analisis dinamis Deteksi malware tak dikenal dengan mudah hanya dengan melakukan analisis perilaku aplikasi Anda. kekurangan analisis dinamik [9].

Penguraian ini membutuhkan waktu selama aplikasi Anda berjalan, sehingga mungkin tidak cepat atau aman. Analisis ini tidak berlaku untuk aplikasi yang menunjukkan perubahan perilaku yang berbeda di bawah kondisi pemicu yang berbeda. Dengan kata lain, gagal mendeteksi malware multipath [9].

f) Analisis hibrid

Teknik ini menggabungkan analisis statis dan dinamis. Langkah selanjutnya merupakan memeriksa terlebih dahulu keberadaan tanda tangan malware di dalam kode, lalu memantau perilaku kode tersebut. Teknik ini menggabungkan keunggulan dari kedua teknik di atas [9].

2.2.5. Definisi Machine Learning

Machine Learning, juga dikenal sebagai pembelajaran mesin, merupakan ilmu komputer yang bekerja tanpa diprogram secara eksplisit. Banyak peneliti berpikir tentang bagaimana membuat kemajuan menuju AI pada skala manusia. Pembelajaran mesin merupakan kecerdasan buatan yang mempelajari cara

membuat data. Pembelajaran mesin biasa disingkat ML. Hal ini diperlukan untuk menerapkan teknik yang cepat dan ampuh untuk menemukan masalah baru.

Menurut definisi, pembelajaran mesin merupakan ilmu atau penelitian yang mempelajari algoritme dan model statistik yang digunakan oleh sistem komputer untuk melakukan tugas tertentu tanpa instruksi eksplisit. Pembelajaran mesin didasarkan pada pola dan inferensi. Untuk memperoleh pola dan kesimpulan ini, algoritme pembelajaran mesin menghasilkan model matematika berdasarkan data sampel, yang sering disebut sebagai "data pelatihan". Penggunaan teknologi ini terkait dengan pembelajaran mesin dan AI. Mesin ini mengesahkan algoritme atau program yang berjalan di komputer. Jadi, jika Anda ingin mempelajari Machine Learning, pastikan Anda terus berinteraksi dengan data Anda. Semua pengetahuan pembelajaran mesin harus melibatkan data [9].

Machine learning merupakan bidang keilmuan yang mempelajari cara membuat program yang dapat menghasilkan pengetahuan baru dari pengetahuan yang sudah ada (disebut sebagai pengalaman atau data), di luar pengetahuan yang “diprogram” langsung ke dalam program.

Istilah yang lebih umum merupakan bagaimana membuat komputer yang dapat belajar dari lingkungannya. Kembangkan "ilmu". Contoh paling sederhana mungkin merupakan prediksi kata di ponsel atau pengenalan wajah di Facebook. Hal ini dimungkinkan karena program di balik kedua hal tersebut telah mengumpulkan pengetahuan dari data yang ada, kebanyakan berupa model matematika.

Ini banyak berkaitan dengan algoritma yang mengekstraksi informasi dari data yang berbeda dan mengenali pola dalam data (pengenalan pola), dan terkait erat dengan statistik. Tetapi pada umumnya, segala sesuatu yang melibatkan proses memperoleh pengetahuan dari data termasuk dalam cabang ilmiah pembelajaran mesin [10].

Machine Learning merupakan salah satu bidang ilmu komputer yang paling disalah pahami. Terlepas dari konotasi negatif istilah “ML” dalam bahasa Indonesia, machine learning juga merupakan bidang dengan cakupan aplikasi yang sangat luas dalam berbagai disiplin ilmu. Sebagian besar bidang yang terkait

dengan "Komputasi Cerdas" membutuhkan pengetahuan pemrograman dan pembelajaran mesin memainkan peran yang sangat penting pada tahap ini. Jika AI berfokus pada pembuatan komputer cerdas, pembelajaran mesin digunakan untuk menghidupkan kecerdasan tersebut. Faktanya, sebagian besar pengetahuan yang kami pelajari di Al dan DM merupakan pembelajaran mesin itu sendiri, yang menyebabkan begitu banyak ambiguitas di bidang ini antara ketiga hal ini [11].

Machine Learning dapat didefinisikan sebagai:

Metode perhitungan empiris untuk meningkatkan kinerja atau membuat prediksi yang akurat. Pengertian pengalaman di sini mengacu pada informasi sebelumnya yang sudah tersedia dan dapat dijadikan sebagai data pembelajar. Skenario pembelajaran mesin meliputi:

1. Superviset Learning

Menggunakan skenario pembelajaran yang diawasi, belajar bersama masukkan data pelatihan berlabel. kemudian dibuat prediksi dari data berlabel.

2. Unsuperviset Learning

Menggunakan skenario pembelajaran tanpa pengawasan dengan Learn masukkan data pelatihan yang tidak berlabel. Kemudian mencobanya mengelompokkan data berdasarkan ciri-ciri yang ditemukan.

3. Reinfortcement Learning

Dalam skenario pembelajaran penguatan, fase pelatihan dan pengujian saling eksklusif. campuran. Secara aktif mengumpulkan informasi tentang peserta didik dengan berinteraksi dengan lingkungan untuk mendapatkan jawaban untuk setiap tindakan pembelajar [11].

2.2.6. Proses-Proses Machine Learning

Membangun mesin pembelajaran mesin yang sukses memerlukan tiga komponen dasar: algoritme matematika, prosesor komputer, dan data. Ketiganya harus tersedia secara bersamaan karena tanpa prosesor komputer tidak mungkin menerapkan algoritma matematika. Hal yang sama terjadi ketika tidak ada data

yang tersedia, sehingga algoritme matematika tidak mungkin menemukan apa pun.

Ketiga komponen ini masing-masing mendorong perkembangan komponen lainnya. Saat ini, teknologi prosesor komputer semakin maju dari hari ke hari dan kekuatan pemrosesan data meningkat dari hari ke hari. Hal ini memungkinkan pemrosesan data dalam jumlah yang semakin besar dan memfasilitasi penelitian dan pengembangan algoritme baru dan proses matematika dengan kompleksitas yang semakin meningkat. Data besar juga mendorong hal-hal seperti perkembangan teknologi prosesor.

Sebuah mesin dikatakan "belajar" jika secara bertahap dapat meningkatkan kualitas keluarannya berdasarkan data yang diberikan. Ini mirip dengan cara orang belajar, menggunakan pengalaman masa lalu untuk memperbaiki cara mereka bekerja sehingga orang lebih responsif ketika menghadapi situasi serupa di masa depan.

Ada banyak algoritme pembelajaran mesin yang ditulis untuk tujuan berbeda, tetapi semuanya mengikuti prinsip yang sama: meniru (atau mencoba meniru) cara manusia belajar. Secara umum, ada tiga langkah penting dalam proses pembelajaran.

1. Pengumpulan Data. Misalnya, pengukuran data transaksi, sensor, catatan, huruf, angka, gambar, suara, dll. Secara komputasional catatan merupakan sekelompok data terkait yang dapat dimanipulasi komputer sebagai satu unit.
2. Abstraksi, proses mengubah data menjadi mode yang lebih umum (definisi model dibahas di bawah).
3. Generalisasi, proses penggunaan model abstrak sebagai dasar pengambilan keputusan atau kesimpulan [12].

2.2.7. Random Forest

Random Forest merupakan metode bagging yang, ketika membangun pohon selama pelatihan, menghasilkan pohon dalam jumlah besar dari data sampel secara independen dari pohon sebelumnya dan membuat keputusan berdasarkan suara terbanyak**.**

Dua konsep yang menjadi dasar Random Forests merupakan membuat ansambel pohon dengan permutasi dan mengantongi dengan pemilihan fitur acak untuk setiap pohon yang dibuat. Pertama, setiap sampel yang diambil dari dataset pohon pelatihan dapat digunakan kembali untuk pohon pelatihan lainnya. Fitur-fitur yang digunakan selama pelatihan dari setiap pohon kemudian merupakan subset dari fitur-fitur yang dimiliki oleh dataset.

Klasifikasi berbasis ensamble bekerja paling baik ketika pelajar dasar memiliki korelasi yang rendah. Ensamble harus membangun pembelajar dasar yang lemah. Ini karena pembelajar yang kuat cenderung berkorelasi kuat dan biasanya juga menyebabkan overfitting random forest, di sisi lain, meminimalkan korelasi dan mempertahankan kekuatan klasifikasi dengan mengacak proses pelatihan. Sejumlah fungsi acak. Ekstrak semua fitur yang ada di setiap pohon pelatihan dan gunakan fitur yang dipilih untuk mendapatkan cabang pohon terbaik. Berbeda dengan proses pohon latih dari pohon keputusan biasa, proses pohon latih yang merupakan bagian dari random forest tidak menggunakan proses pruning tetapi tetap bercabang sampai ukuran batas daun tercapai.

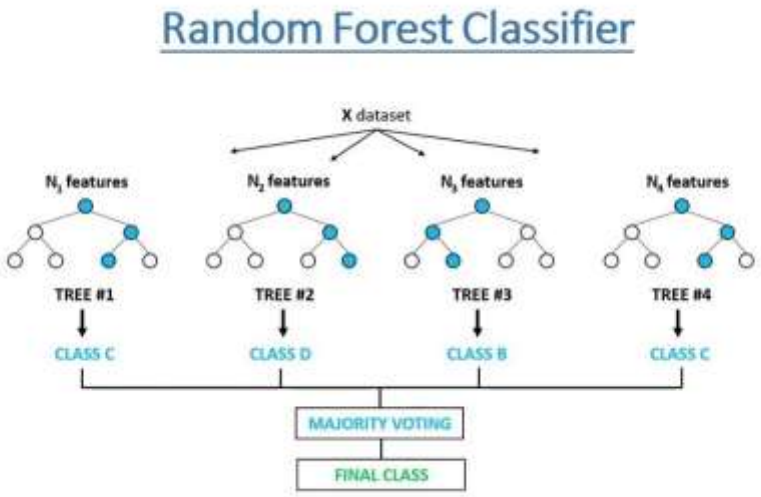
Algoritma Random Forest dapat digunakan untuk klasifikasi data besar. Meskipun pruning dan pruning variabel seperti pohon keputusan tidak ditemukan pada algoritma random forest, keuntungan dari random forest merupakan klasifikasi dan prediksi kelas dapat menggabungkan banyak pohon dan hanya menggunakan satu pohon saja. Ada tiga metode random forest, random forest umum random forest bersyarat, dan menggunakan random forest untuk meningkatkan hasil logistik. random forest merupakan salah satu metode klasifikasi baru dibandingkan dengan metode bagging dan boosting. Metode random forest bertujuan untuk meningkatkan prediksi dari metode bagging. Implementasi random forest telah digunakan dalam beberapa penelitian. Saat menerapkan metode random forest dan metode pohon keputusan pada 20 variabel dalam kumpulan data, hasil menunjukkan bahwa metode random forest secara keseluruhan lebih unggul daripada metode pohon keputusan untuk data dengan jumlah pengamatan yang besar. pengamatan. Metode random forest memiliki

karakteristik yang sama dengan metode pohon keputusan. Adapun kelebihan dari metode Random Forest sebagai berikut:

1. Hasil akurasi bagus
2. Relatif kuat terhadap outliers dan noise
3. Lebih cepat dibandingkan bagging dan boosting
4. Sifaytnya yang sederhana dan mudah dipararelkan

2.2.8. Klasifikasi Rndom Forest

Random Forest dapat mengklasifikasikan data dengan atribut yang tidak lengkap dan berguna untuk mengklasifikasikan data sampel yang besar. Proses klasifikasi hutan acak membagi (split) data sampel yang ada menjadi pohon keputusan acak. Ketika sebuah pohon terbentuk, ia memiliki akar (root), node interior (cabang), dan daun (hasil kelas). Kemudian, Random Forest digunakan untuk mendapatkan random forest, sehingga menghasilkan akurasi tertinggi dalam klasifikasi data. Klasifikasi berbasis ensamble bekerja paling baik ketika pelajar dasar memiliki korelasi yang rendah. Ensamble harus membangun pembelajar dasar yang lemah. Ini karena pembelajar yang kuat cenderung berkorelasi kuat dan biasanya juga menyebabkan overfitting random forest, di sisi lain, meminimalkan korelasi dan mempertahankan kekuatan klasifikasi dengan mengacak proses pelatihan. Sejumlah fungsi acak. Ekstrak semua fitur yang ada di setiap pohon pelatihan dan gunakan fitur yang dipilih untuk mendapatkan cabang pohon terbaik. Berbeda dengan proses pohon latih dari pohon keputusan biasa, proses pohon latih yang merupakan bagian dari random forest tidak menggunakan proses pruning tetapi tetap bercabang sampai ukuran batas daun tercapai [13].



Gambar 2.2. Skema Algoritma Random Forest (13)

Metode *Random Forest* menghasilkan satu set pohon acak. Kelas yang dihasilkan berasal dari proses klasifikasi yang dipilih dari kelas yang paling banyak (modus) yang dihasilkan oleh pohon keputusan yang ada (13). Algoritma atau prosedur dalam membangun *Random Forest* pada gugus data yang terdiri dari *n* amatan dan terdiri atas *p* peubah penjelas (*predictor*), berikut dibawah merupakan tahapan penyusunan dan pendugaan menggunakan *Random Forest.*

a) Tahapan Bootstrapping

Untuk mulai membangun *Random Forest* langkah pertama yang dilakukan adalah pengambilan sampel secara acak berukuran *n* dari kumpulan data asli dengan pengembalian.

Gambar 2.3. Contoh pengambilan Sampel dengan Metode Bootstrapping (13)

b) Tahapan Random *Feature Selection*

Pada tahapan ini pohon dibangun hingga mencapai ukuran maksimum (tanpa pemangkasan). Pada proses pemilah pemilih variabel prediktor *m* dipilih secara acak, dimana *m*<<*p*, kemudian pemilah terbaik dipilih berdasarkan *m* prediktor. Berikut dibawah ini merupakan contoh dalam membangun *Decision Tree*.

Dalam menentukan pohon keputusan langkah awal yang dilakukan yaitu menghitung nilai *entropy* sebagai penentu tingkat ketidakmurnian atribut dan nilai *information gain*. Menghitung nilai *entropy* dapat menggunakan rumus seperti persamaan (3.1) untuk satu atribut, persamaan (3.2) untuk dua atribut menggunakan tabel frekuensi, dan menentukan nilai *information gain* menggunakan persamaan (3.3):

2.2.9. Confusion Matrix

Confusion matrix merupakan sebuah metode yang dapat digunakan untuk mengukur kinerja dari sebuah metode klasifikasi. Pada dasarnya, *Confusion Matrix* berisi informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi sebagaimana mestinya. Saat mengukur kinerja menggunakan *Confusion matrix*, ada empat istilah yang menjelaskan hasil dari proses klasifikasi. Keempat suku tersebut merupakan *true positive* (TP), *true*

negative (TN), *false positive* (FP), dan *false negative* (FN). Nilai True Negative (TN) merupakan jumlah data negatif yang dikenali dengan benar. Positif palsu (FP), di sisi lain, mengenali data negatif sebagai data positif. *Confusion matrix* merupakan metode yang digunakan untuk mengukur atau melakukan perhitungan akurasi untuk konsep data mining. Sebuah *Confusion matrix* terdiri dari kumpulan data uji yang diprediksi benar atau salah oleh model klasifikasi.

Untuk menggambarkan akurasi klasifikasi secara efisien, kami mempresentasikan hasil yang diperoleh dengan menggunakan confusion matrix. Setiap kolom matriks mewakili instance kelas yang diprediksi dan setiap baris mewakili instance kelas aktual (atau sebaliknya). Elemen diagonal mewakili jumlah titik di mana kelas yang diprediksi cocok dengan kelas sebenarnya, sedangkan elemen off-diagonal merupakan elemen yang salah diklasifikasikan oleh pengklasifikasi. Nilai diagonal yang lebih tinggi dalam *Confusion matrix* menunjukkan prediksi yang lebih baik. Di bawah ini merupakan confusion matrix untuk mengklasifikasikan menjadi dua kelas.

	Predicted Positive	Predicteed Negative
Actual Positive Intances	Number of True Positive Instance (TP)	Number of False Negatives instances (FN)
Acual Negative instances	Number of False Positive Instance (FP)	Number of True Negatives instances (TN)

Tabel 2.2 *confusion matrix*

Berdasarkan gambar confusion matrix diatas :

1. True Positives (TP) merupakan jumlah data positif yang diklasifikasikan sebagai nilai positif.
2. Positif palsu (FP) merupakan jumlah data 30egative yang diklasifikasikan sebagai nilai positif.
3. Negatif palsu (FN) merupakan jumlah data positif yang diklasifikasikan sebagai nilai positif.
4. True negative (TN) merupakan jumlah data positif yang diklasifikasikan sebagai nilai 30egative.

Nilai yang dihasilkan melalui metode confusion matrix adalah berupa evaluasi sebagai berikut:

A. Akurasi

Persentase volume data yang diklasifikasikan atau diprediksi dengan benar oleh algoritme

Rumus: Akurasi = (TP + TN) / (TP + TN + FP + FN)

B. Presisi

Nilai akurasi metode yang digunakan untuk klasifikasi. Nilai ini menunjukkan berapa banyak data yang dapat ditempatkan di kelas yang benar untuk beberapa pengujian.

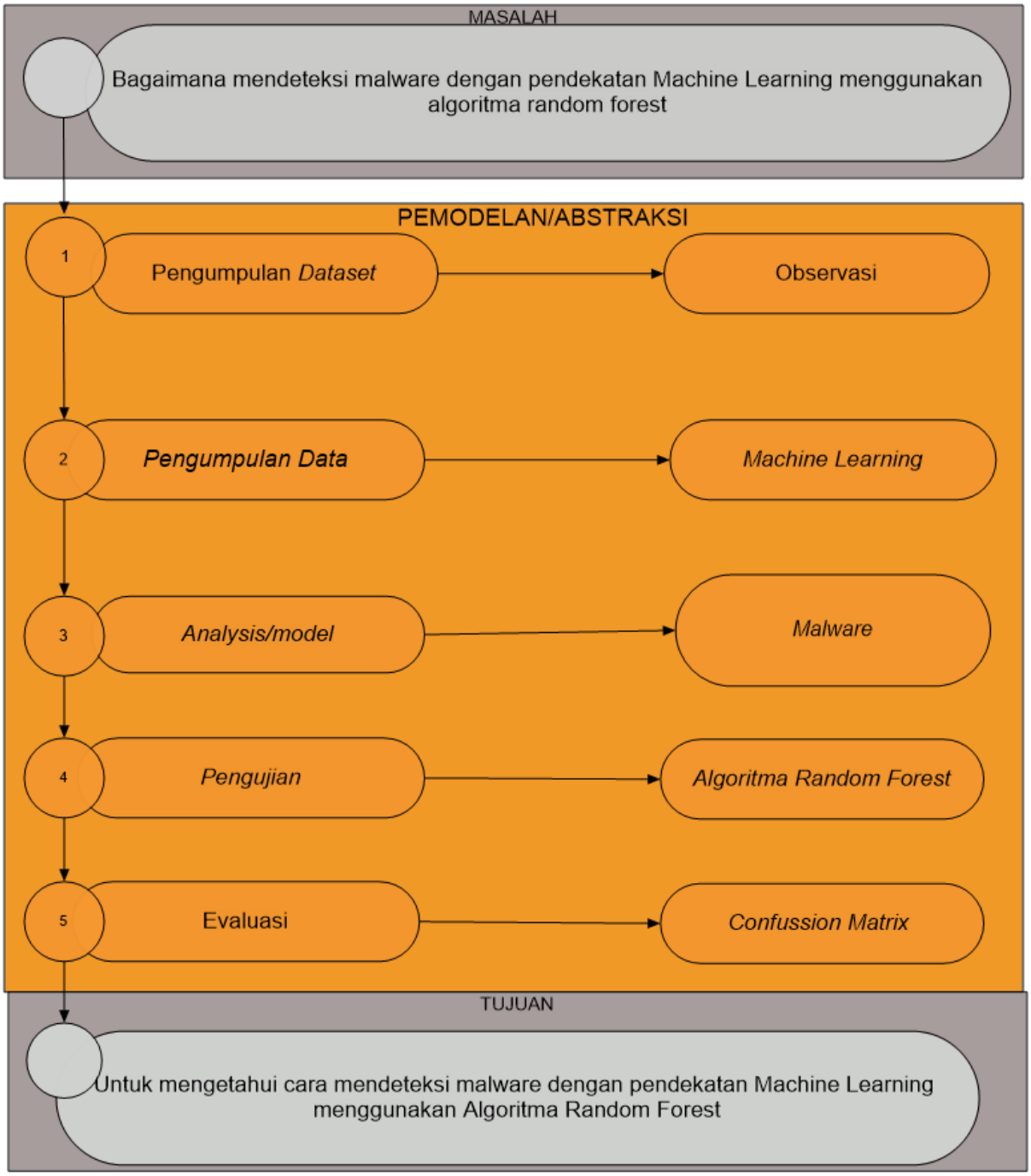
Rumus: Akurasi = TP / (TP + FP)

C. Recall

Berapa persen dari data yang merupakan nilai yang dapat digunakan untuk mengukur hasil yang diklasifikasikan dengan benar, dan rumus untuk menghitung penarikan kembali adalah:

Ingat = TP / (TP + FN) [14].

2.3.Kerangka Pemikiran

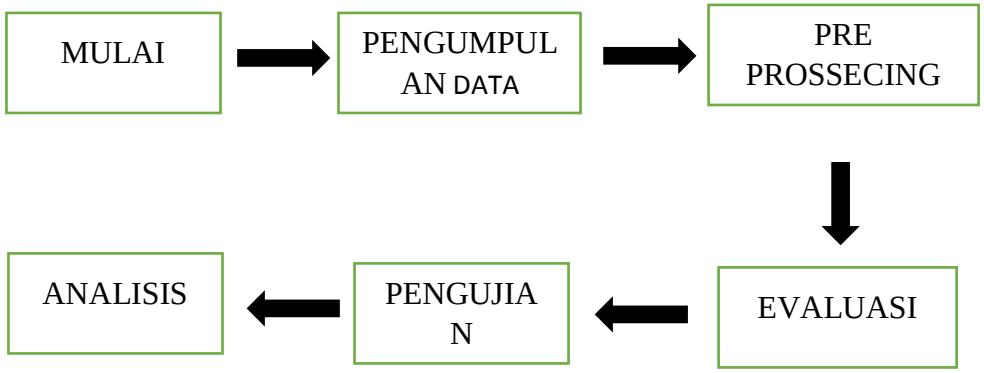


Gambar 2.4. Kerangka Pemikiran

BAB III
METODE PENELITIAN

3.1 Objek dan Metode Penelitian
3.2.1 Objek

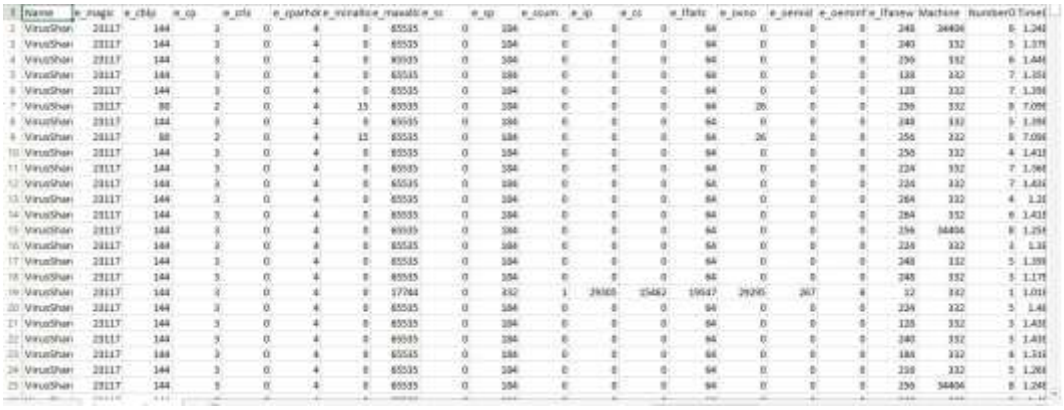
Pada Penelitian ini Objek yang dilakukan penelitian ini ialah mendeteksi Malware menggunakan Machine Learning dengan metode Algoritma Random Forest, maka penelitian ini merupakan penelitian kualitatif. Maka jenis dari penelitian ini merupakan penelitian model uji coba metode – eksperimental.



Gambar 3.1 [10]

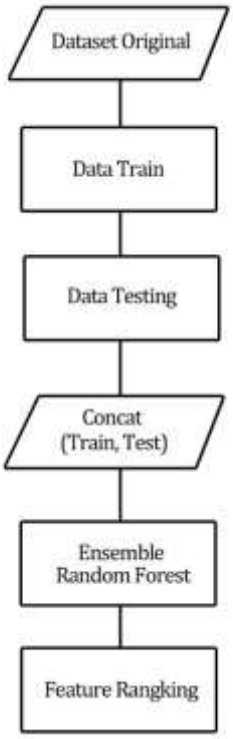
3.2 Metode Penelitian
3.2.1 Dataset yang digunakan

Dalam memulai sebuah penelitian tentunya kita memerlukan sebuah data sebagai pondasi masalah untuk diolah agar mencapai tujuan penelitian yang sukses. dataset yang diambil dalam penelitian ini diperoleh dari internet yaitu pada website https://www.kaggle.com/datasets/amauricio/pe-files-malwares. Terdiri dari 19611 baris dan 79 kolom



Gambar 3.2. Sumber data: dataset malware

3.3 Pemodelan



Gambar 3.3 Flowchart Pemodelan Data

3.4. Machine Learning

Cara kerja machine learning dalam praktik bergantung pada teknik dan metode pembelajaran apa yang Anda gunakan dengan ML. Namun pada dasarnya, prinsip kerja machine learning merupakan sama: pengumpulan data, eksplorasi

data, pemilihan model atau metode, pemberian pelatihan untuk model yang dipilih, dan evaluasi hasil ML.

3.5 Pra Pengolahan Data

Preprocessing adalah bagian selanjutnya dari sebuah kegiatan pengumpulan dataset yang diperoleh dalam bentuk spreadsheet berformat excel maka dari itu belum dapat langsung diolah atau di gunakan karena dataset masih dalam bentuk mentah tidak teratur bahkan hanya di pisahkan tanda koma. data yang diambil merupakan DATASET MALWARE yang bersumber dari kaggle sehingga dijadikan sebagai dataset dengan melalui proses tahapan pengolahan data.

3.5.1 Analysis/Model Malware

Menyajikan taksonomi tentang bagaimana pembelajaran mesin digunakan untuk analisis malware. Identifikasi tiga dimensi utama yang memudahkan pengorganisasian pekerjaan yang diteliti. Yang pertama mencirikan tujuan akhir dari analisis, seperti deteksi malware. Dimensi kedua menjelaskan fitur yang mendasari analisis dalam hal bagaimana mereka diekstraksi, misalnya dengan analisis dinamis, dan fitur apa yang diperhitungkan, misalnya register CPU. Terakhir, dimensi ketiga menentukan jenis algoritme pembelajaran mesin (mis. pembelajaran terawasi) yang digunakan untuk analisis.

3.5.2 Pengujian Algoritma Random Forest

Algoritma Random Forest (RF) merupakan perluasan dari metode klasifikasi dan pohon regresi (CART) dengan menerapkan bootstrap aggregation (bagging) dan metode pemilihan fitur secara acak (Breiman 2001). Algoritma RF merupakan algoritma yang baik untuk mengklasifikasikan data dalam jumlah besar, dan algoritma RF tidak memiliki pemangkasan atau pemangkasan variabel seperti algoritma pohon keputusan. Metode RF menggabungkan banyak pohon (tree) untuk membuat kelas klasifikasi dan prediksi, berbeda dengan pohon yang hanya terdiri dari satu pohon. Dalam RF, pembangunan pohon dilakukan dengan melatih sampel data. Sampling-by-penggantian merupakan metode yang digunakan untuk sampel data. Pemilihan variabel yang digunakan untuk pemisahan merupakan acak. Setelah semua pohon terbentuk, dilakukan

klasifikasi. Keputusan klasifikasi dalam RF ini didasarkan pada voting dari masing-masing pohon, dengan voting terbanyak menjadi pemenangnya.

3.5.3 Evaluasi Conffusion Matrix

Untuk mengevaluasi hasil algoritma klasifikasi, teknik evaluasi model algoritma klasifikasi yang digunakan dalam penelitian ini merupakan fusion matrix. Matriks konfusi merupakan cara paling sederhana untuk menilai hasil kinerja algoritma pengklasifikasi dengan membandingkan contoh positif yang diklasifikasikan sebagai benar atau salah dengan contoh negatif yang diklasifikasikan sebagai benar atau salah. Ada banyak pandangan berbeda tentang matriks kebingungan, dan matriks kebingungan memainkan peran mendasar dalam mengevaluasi kinerja algoritma klasifikasi. Dalam matriks kebingungan, baris mewakili label yang benar dan kolom mewakili prediksi algoritma pengklasifikasi.

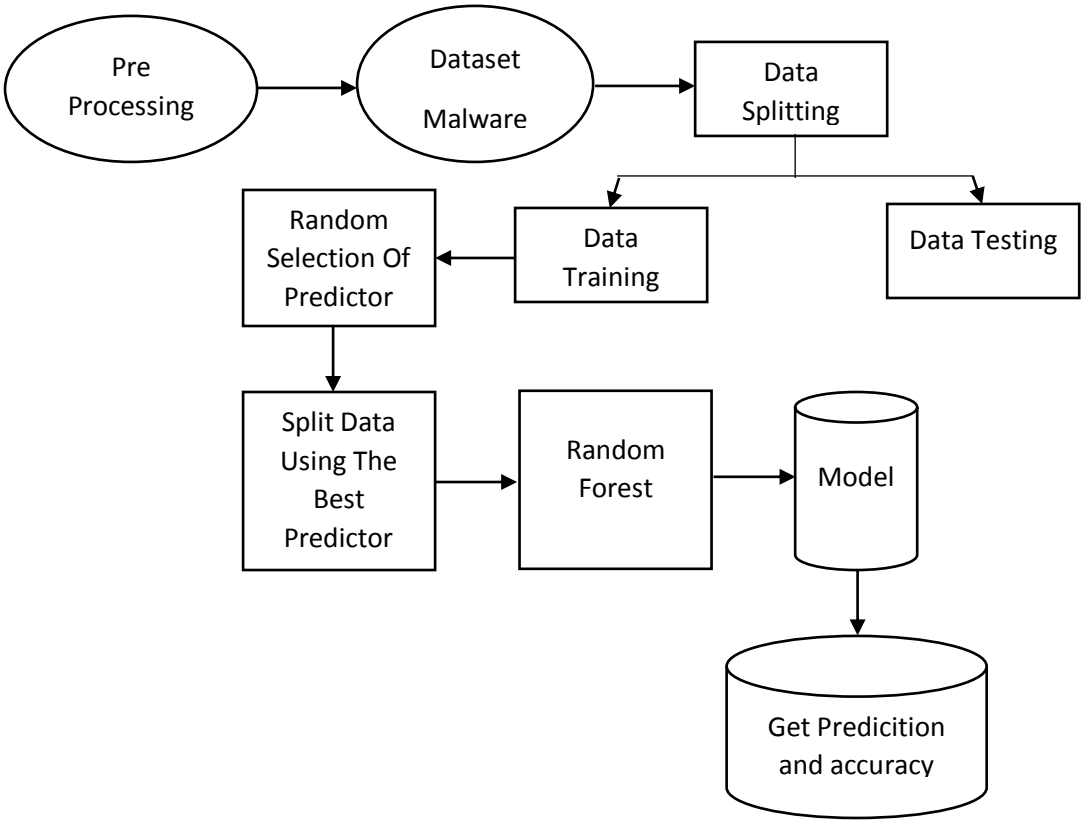
3.5.4 Pengujian

Pada tahap pengujian akan dilakukan pengumpulan data terlebih dahulu, lalu mengumpulkan alat dan bahan yang akan digunakan. Kemudian mendeteksi malware pada laptop yang sudah disiapkan lalu melakukan pengujian malware menggunakan teknik machine learning dengan metode algoritma random forest, sehingga dapat mengetahui apakah teknik ini dapat berguna untuk mendeteksi malware yang ada pada setiap laptop dan bisa mengatasi terjadinya malware pada laptop.

BAB IV
HASIL PENELITIAN

4,1.Hasil Pengumpulan Data

Hasil pengumpulan data ini penulis menggunakan data public https://www.kaggle.com/datasets/amauricio/pe-files-malwares kemudian dataset tersebut akan diproses dan di klasifikasi berdasarkan pemodelan yang telah diolah oleh peneliti. Pengumpulan data berasal dari perangkat/alat dari hasil penelitian orang lain dataset *public*, dataset *public* adalah data yang sudah ada yang digunakan para peneliti sebelumnya. Malware yang menyusup di browser. Dalam satu baris ada satu serangan yang terjadi, dan setiap serangan malware pasti akan meninggalkan jejak yang disebut *loc*.



Gambar 4.1. Metode Klasifikasi Random Forest

```
<class 'pandas.core.frame.DataFrame'>
Dataset yang digunakan pada contoh ini adalah dataset malware,
tahapan-tahapan yang akan dilakukan adalah analisis deskriptif dan
penanganan data , pembagian data latih dan data uji, pemodelan
dengan Random Forest serta evaluasi model.

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19611 entries, 0 to 19610
Data columns (total 79 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Name                            19611 non-null  object
 1   e_magic                         19611 non-null  int64
 2   e_cblp                          19611 non-null  int64
 3   e_cp                            19611 non-null  int64
 4   e_crlc                          19611 non-null  int64
 5   e_cparhdr                       19611 non-null  int64
 6   e_minalloc                      19611 non-null  int64
 7   e_maxalloc                      19611 non-null  int64
 8   e_ss                            19611 non-null  int64
 9   e_sp                            19611 non-null  int64
 10  e_csum                          19611 non-null  int64
 11  e_ip                            19611 non-null  int64
 12  e_cs                            19611 non-null  int64
 13  e_lfarlc                        19611 non-null  int64
 14  e_ovno                          19611 non-null  int64
 15  e_oemid                         19611 non-null  int64
 16  e_oeminfo                       19611 non-null  int64
 17  e_lfanew                        19611 non-null  int64
 18  Machine                         19611 non-null  int64
 19  NumberOfSections                19611 non-null  int64
 20  TimeDateStamp                   19611 non-null  int64
 21  PointerToSymbolTable            19611 non-null  int64
 22  NumberOfSymbols                 19611 non-null  int64
 23  SizeOfOptionalHeader            19611 non-null  int64
 24  Characteristics                 19611 non-null  int64
 25  Magic                           19611 non-null  int64
 26  MajorLinkerVersion              19611 non-null  int64
 27  MinorLinkerVersion              19611 non-null  int64
 28  SizeOfCode                      19611 non-null  int64
 29  SizeOfInitializedData           19611 non-null  int64
 30  SizeOfUninitializedData         19611 non-null  int64
 31  AddressOfEntryPoint             19611 non-null  int64
 32  BaseOfCode                      19611 non-null  int64
 33  ImageBase                       19611 non-null  int64
 34  SectionAlignment                19611 non-null  int64
 35  FileAlignment                   19611 non-null  int64
 36  MajorOperatingSystemVersion     19611 non-null  int64
 37  MinorOperatingSystemVersion     19611 non-null  int64
 38  MajorImageVersion               19611 non-null  int64
```

```
 39  MinorImageVersion             19611 non-null  int64
 40  MajorSubsystemVersion         19611 non-null  int64
 41  MinorSubsystemVersion         19611 non-null  int64
 42  SizeOfHeaders                 19611 non-null  int64
 43  CheckSum                      19611 non-null  int64
 44  SizeOfImage                   19611 non-null  int64
 45  Subsystem                     19611 non-null  int64
 46  DllCharacteristics            19611 non-null  int64
 47  SizeOfStackReserve            19611 non-null  int64
 48  SizeOfStackCommit             19611 non-null  int64
 49  SizeOfHeapReserve             19611 non-null  int64
 50  SizeOfHeapCommit              19611 non-null  int64
 51  LoaderFlags                   19611 non-null  int64
 52  NumberOfRvaAndSizes           19611 non-null  int64
 53  Malware                       19611 non-null  int64
 54  SuspiciousImportFunctions     19611 non-null  int64
 55  SuspiciousNameSection         19611 non-null  int64
 56  SectionsLength                19611 non-null  int64
 57  SectionMinEntropy             19611 non-null  float64
 58  SectionMaxEntropy             19611 non-null  int64
 59  SectionMinRawsize             19611 non-null  int64
 60  SectionMaxRawsize             19611 non-null  int64
 61  SectionMinVirtualsize         19611 non-null  int64
 62  SectionMaxVirtualsize         19611 non-null  int64
 63  SectionMaxPhysical            19611 non-null  int64
 64  SectionMinPhysical            19611 non-null  int64
 65  SectionMaxVirtual             19611 non-null  int64
 66  SectionMinVirtual             19611 non-null  int64
 67  SectionMaxPointerData         19611 non-null  int64
 68  SectionMinPointerData         19611 non-null  int64
 69  SectionMaxChar                19611 non-null  int64
 70  SectionMainChar               19611 non-null  int64
 71  DirectoryEntryImport          19611 non-null  int64
 72  DirectoryEntryImportSize      19611 non-null  int64
 73  DirectoryEntryExport          19611 non-null  int64
 74  ImageDirectoryEntryExport     19611 non-null  int64
 75  ImageDirectoryEntryImport     19611 non-null  int64
 76  ImageDirectoryEntryResource   19611 non-null  int64
 77  ImageDirectoryEntryException  19611 non-null  int64
 78  ImageDirectoryEntrySecurity   19611 non-null  int64
dtypes: float64(1), int64(77), object(1)
memory usage: 11.8+ MB
```

Dataset malware terdiri dari 19611 baris. Terdapat 79 kolom, dimana seluruhnya bertipe numerik, Berikut penjelasan variabel nama kolom.

Name: Nama file PE.

e_magic: Tanda identifikasi yang menunjukkan tipe file PE.

e_cblp: Jumlah byte pada blok terakhir dari header file PE yang tidak digunakan

e_cp: Jumlah paragraf dalam file PE.

e_crlc: Jumlah entri dalam tabel penyesuaian relokasi.

e_cparhdr: Ukuran header file PE dalam paragraf.

e_minalloc: Jumlah blok paragraf yang dialokasikan pada waktu runtime minimum.

e_maxalloc: Jumlah blok paragraf yang dialokasikan pada waktu runtime maksimum.

e_ss: Nilai stack segment.

e_sp: Pointer stack.

e_csum: Checksum file PE.

e_ip: Pointer instruction.

e_cs: Nilai code segment.

e_lfarlc: Offset relatif dari tabel penyesuaian relokasi.

e_ovno: Nomor overlay jika ada.

e_oemid: ID produsen OEM.

e_oeminfo: Informasi tambahan dari produsen OEM.

e_lfanew: Offset ke header PE baru.

Machine: Tipe arsitektur mesin target.

NumberOfSections: Jumlah bagian (sections) dalam file PE.

TimeDateStamp: Tanggal dan waktu kompilasi file PE.

PointerToSymbolTable: Pointer ke tabel simbol.

NumberOfSymbols: Jumlah simbol dalam tabel simbol.

SizeOfOptionalHeader: Ukuran header opsional file PE.

Characteristics: Karakteristik file PE.

Magic: Nilai ajaib yang menunjukkan tipe file PE.

MajorLinkerVersion: Versi mayor linker yang digunakan.

MinorLinkerVersion: Versi minor linker yang digunakan.

SizeOfCode: Ukuran kode (section .text) dalam file PE.

SizeOfInitializedData: Ukuran data terinisialisasi dalam file PE.

SizeOfUninitializedData: Ukuran data tidak terinisialisasi dalam file PE.

AddressOfEntryPoint: Alamat entri point eksekusi.

BaseOfCode: Alamat awal kode dalam ruang memori.

ImageBase: Alamat dasar gambar dalam ruang memori.

SectionAlignment: Pembatasan sejajar untuk bagian (sections).

FileAlignment: Pembatasan sejajar untuk file.

MajorOperatingSystemVersion: Versi mayor sistem operasi target.

MinorOperatingSystemVersion: Versi minor sistem operasi target.

MajorImageVersion: Versi mayor gambar.

MinorImageVersion: Versi minor gambar.

MajorSubsystemVersion: Versi mayor subsistem.

MinorSubsystemVersion: Versi minor subsistem.

SizeOfHeaders: Ukuran header dalam file PE.

CheckSum: Checksum file PE.

SizeOfImage: Ukuran gambar dalam memori.

Subsystem: Subsistem yang diperlukan untuk mengeksekusi file PE.

DllCharacteristics: Karakteristik DLL file PE.

SizeOfStackReserve: Ukuran memori yang direserve untuk stack.

SizeOfStackCommit: Ukuran memori untuk stack yang di-commit saat proses runtime.

SizeOfHeapReserve: Ukuran memori yang direserve untuk heap.

SizeOfHeapCommit: Ukuran memori yang di-commit untuk heap saat proses runtime.

LoaderFlags: Flag yang mengontrol perilaku loader.

NumberOfRvaAndSizes: Jumlah entri dalam direktori Data Directories.

Malware: Label yang menunjukkan apakah file PE tersebut tergolong sebagai malware (1) atau bukan (0).

SuspiciousImportFunctions: Jumlah fungsi impor yang mencurigakan.

SuspiciousNameSection: Jumlah bagian (sections) dengan nama yang mencurigakan.

SectionsLength: Panjang (ukuran) dari setiap bagian (sections) dalam file PE.

SectionMinEntropy: Entropi minimum dari setiap bagian (sections) dalam file PE.

SectionMaxEntropy: Entropi maksimum dari setiap bagian (sections) dalam file PE.

SectionMinRawsize: Ukuran raw data minimum dari setiap bagian (sections) dalam file PE.

SectionMaxRawsize: Ukuran raw data maksimum dari setiap bagian (sections) dalam file PE.

SectionMinVirtualsize: Ukuran virtual minimum dari setiap bagian (sections) dalam file PE.

SectionMaxVirtualsize: Ukuran virtual maksimum dari setiap bagian (sections) dalam file PE.

SectionMaxPhysical: Alamat fisik maksimum dari setiap bagian (sections) dalam file PE.

SectionMinPhysical: Alamat fisik minimum dari setiap bagian (sections) dalam file PE.

SectionMaxVirtual: Alamat virtual maksimum dari setiap bagian (sections) dalam file PE.

SectionMinVirtual: Alamat virtual minimum dari setiap bagian (sections) dalam file PE.

SectionMaxPointerData: Jumlah pointer data maksimum dalam setiap bagian (sections) dalam file PE.

SectionMinPointerData: Jumlah pointer data minimum dalam setiap bagian (sections) dalam file PE.

SectionMaxChar: Karakteristik maksimum dari setiap bagian (sections) dalam file PE.

SectionMainChar: Karakteristik utama dari setiap bagian (sections) dalam file PE.

DirectoryEntryImport: Jumlah entri dalam direktori Entry Import.

DirectoryEntryImportSize: Ukuran dari direktori Entry Import.

DirectoryEntryExport: Jumlah entri dalam direktori Entry Export.

ImageDirectoryEntryExport: Jumlah entri dalam direktori Image Entry Export.

ImageDirectoryEntryImport: Jumlah entri dalam direktori Image Entry Import.

ImageDirectoryEntryResource: Jumlah entri dalam direktori Image Entry Resource.

ImageDirectoryEntryException: Jumlah entri dalam direktori Image Entry Exception.

ImageDirectoryEntrySecurity: Jumlah entri dalam direktori Image Entry Security.

Inilah penjelasan untuk setiap variabel yang terdapat dalam dataset "dataset_malware.csv". Variabel-variabel ini mencakup berbagai aspek dan fitur dari file PE yang digunakan untuk analisis dan deteksi malware.

Nilai 1 menunjukkan malware dan nilai 0 menunjukkan benign, tidak terdapat nilai *null* atau *missing* pada dataset. Namun demikian, tetap perlu dilakukan pemeriksaan lebih lanjut untuk melihat kondisi dan sebaran data setiap *feature.*

4.2. Pra Pemrosesan Data

4.2.1 Instal Library Python

Pada proses ini pra pemrosesan data dilakukan dengan menginstal beberapa *library python*. *Pandas* adalah *library python open source* yang biasanya digunakan untuk kebutuhan data analisis. *Pandas* membuat *python* supaya dapat bekerja dengan data yang berbentuk tabular seperti spreadsheet dengan cara pemuatan data yang cepat, manipulasi data, mengggabungkan data, serta ada berbagai fungsi yang lain. Untuk memanggil *library pandas, library numpy* juga ikut dipanggil. Selanjutnya import *pickle* untuk implementasi protocol *biner* untuk *serializing* dan de-*serializing* dari struktur objek pada *python*. Selanjutnya *import seaborn*

untuk melakukan visualisasi data. Selanjutnya *import matplotlib.pyplot* as plt untuk kumpulan fungsi yang membuat beberapa perubahan pada gambar, membuat area plot pada gambar, menambah label di plot dan lainnya.

```
import pandas as pd
import numpy as np
import pickle
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
confusion_matrix
import os
import warnings
warnings.simplefilter('ignore')
```

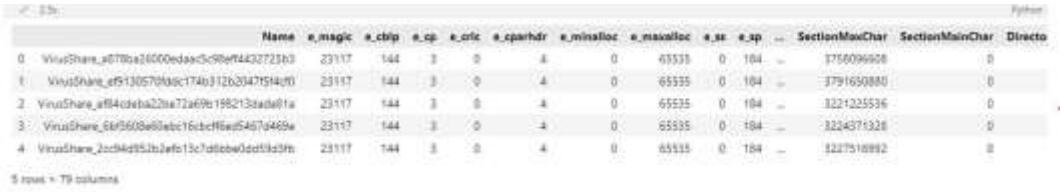
Selanjutnya from *sklearn.ensemble* import *RandomForestClassifier* untuk memperoleh model Random forest berdasarkan data latih, langkah selanjutnya adalah mengevaluasi model menggunakan data uji. Penggunakan data uji atau data yang belum pernah dilihat model saat proses *training* akan memberikan evaluasi yang lebih fair dang menghindari everfitting. Selanjutnya *from sklearn.model_selection import train_test_split* dapat digunakan untuk pembagian data. Pembagian dilakukan secara acak dan proporsional *(stratified random sampling)* . selanjutnya from sklearn.metrics import *classification_report*, *confusion_matrix* untuk evaluasi kinerja dalam pembelajaran mesin yang digunakan untuk menunjukan presisi, daya ingat, *skor F1*, dan model klasifikasi. Import os untuk membuat dan menghapus direktori (folder), mengambil isinya, mengubah dan mengidentifikasi direktori. *Import warnings* fungsi ini menulis pesan peringatan.

4.2.2. Eksplorasi Data

Eksplorasi Data adalah bagian penting dariproses data *science*, yaitu berupa proses menganalisis sekumpulan data untuk meringkas karakteristik utamanya agar pengguna lebih memahami dataset yang akan digunakan.

Head() digunakan untuk menampilkan data awal atau data teratas pada dataframe. Default-nya jika kita tidak memberikan argument di dalam tanda kurung (), data yang akan ditampilkan adalah 5 baris data teratas

```
data = pd.read_csv("dataset_malwares.csv")
data.head()
```



	Name	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp	...	SectionMaxChar	SectionMainChar	Directo
0	VirusShare_a07f0ca26000edaac5c98ef44327231b3	23117	144	3	0	4	0	65535	0	184	...	3756096608	0	
1	VirusShare_ef5130570fddc174b312b2047f5f4cf0	23117	144	3	0	4	0	65535	0	184	...	3791650880	0	
2	VirusShare_af84cdeba22ba72a69b19821dada801a	23117	144	3	0	4	0	65535	0	184	...	3221225536	0	
3	VirusShare_6bf3608e80ebc16cbcff6ed5467d469e	23117	144	3	0	4	0	65535	0	184	...	3224371328	0	
4	VirusShare_2cc94d952b2efb15c7d6bbe0dd5fd3fb	23117	144	3	0	4	0	65535	0	184	...	3227516992	0	

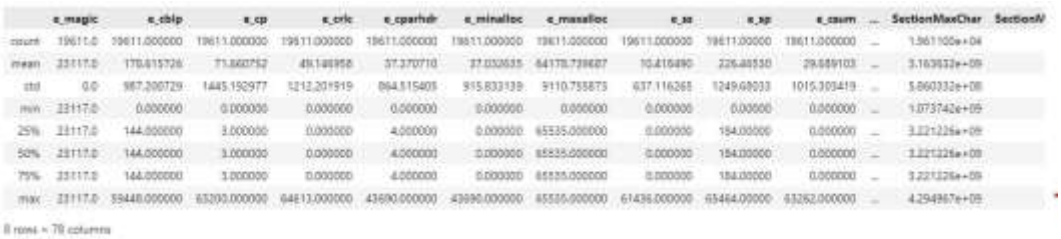
5 rows × 79 columns

Gambar 4.2 Hasil Eksplorasi Data

- Data Describe

  Mengembalikan deskripsi data dalam DataFrame. Jika DataFrame berisi data numerik, deskripsi berisi informasi ini untuk setiap kolom.

```
data.describe()
```



	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp	e_csum	...	SectionMaxChar	SectionM
count	19611.0	19611.000000	19611.000000	19611.000000	19611.000000	19611.000000	19611.000000	19611.000000	19611.000000	19611.000000	...	1.961100e+04	
mean	23117.0	170.615726	71.660752	49.146958	37.370710	37.032655	64178.739687	10.416490	226.46530	29.689103	...	3.163932e+09	
std	0.0	967.200729	1445.152977	1212.201919	864.515405	915.833139	9110.755875	637.116265	1249.68033	1015.303419	...	5.860332e+08	
min	23117.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.00000	0.000000	...	1.073742e+09	
25%	23117.0	144.000000	3.000000	0.000000	4.000000	0.000000	65535.000000	0.000000	184.00000	0.000000	...	3.221226e+09	
50%	23117.0	144.000000	3.000000	0.000000	4.000000	0.000000	65535.000000	0.000000	184.00000	0.000000	...	3.221226e+09	
75%	23117.0	144.000000	3.000000	0.000000	4.000000	0.000000	65535.000000	0.000000	184.00000	0.000000	...	3.221226e+09	
max	23117.0	59448.000000	65200.000000	64613.000000	43690.000000	43690.000000	65535.000000	61436.000000	65464.00000	65262.000000	...	4.294967e+09	

8 rows × 79 columns

Gambar 4.3 Hasil Data Describe

```
data.loc[:,["Name","Machine","TimeDateStamp","Malware"]]
```

	Name	Machine	TimeDateStamp	Malware
0	VirusShare_a878ba26000edaac5c98eff4432723b3	34404	1236512358	1
1	VirusShare_ef9130570fddc174b312b2047f5f4cf0	332	1365109591	1
2	VirusShare_ef84cdeba22be72a69b198213dada81a	332	1438777028	1
3	VirusShare_6bf3608e60ebc16cbcff6ed5467d469e	332	1354629311	1
4	VirusShare_2cc94d952b2efb13c7d6bbe0dd59d3fb	332	1386631250	1
...	...	...	...	...
19606	clip.exe	332	1377143713	0
19607	VNC-Server-6.2.0-Windows.exe	332	1501777476	0
19608	Microsoft.GroupPolicy.Management.ni.dll	332	1377135839	0
19609	cryptuiwizard.dll	332	1377141725	0
19610	winhttp.dll	332	1377139145	0

19611 rows × 4 columns

Gambar 4.4 Hasil data loc

Data.loc merupakan salah satu cara yang efektif untuk memilih baris dan kolom pada dataframe sesuai dengan nama index baris atau kolom. Seleksi kolom yang kita gunakan terdiri dari “Nama”,”Machine”,”TimeDateStamp”,”Malware”.

Data Kolom

```
data.columns
```

```
Index(['Name', 'e_magic', 'e_cblp', 'e_cp', 'e_crlc', 'e_cparhdr',
       'e_minalloc', 'e_maxalloc', 'e_ss', 'e_sp', 'e_csum', 'e_ip', 'e_cs',
       'e_lfarlc', 'e_ovno', 'e_oemid', 'e_oeminfo', 'e_lfanew', 'Machine',
       'NumberOfSections', 'TimeDateStamp', 'PointerToSymbolTable',
       'NumberOfSymbols', 'SizeOfOptionalHeader', 'Characteristics', 'Magic',
       'MajorLinkerVersion', 'MinorLinkerVersion', 'SizeOfCode',
       'SizeOfInitializedData', 'SizeOfUninitializedData',
       'AddressOfEntryPoint', 'BaseOfCode', 'ImageBase', 'SectionAlignment',
       'FileAlignment', 'MajorOperatingSystemVersion',
       'MinorOperatingSystemVersion', 'MajorImageVersion', 'MinorImageVersion',
       'MajorSubsystemVersion', 'MinorSubsystemVersion', 'SizeOfHeaders',
       'CheckSum', 'SizeOfImage', 'Subsystem', 'DllCharacteristics',
       'SizeOfStackReserve', 'SizeOfStackCommit', 'SizeOfHeapReserve',
       'SizeOfHeapCommit', 'LoaderFlags', 'NumberOfRvaAndSizes', 'Malware',
       'SuspiciousImportFunctions', 'SuspiciousNameSection', 'SectionsLength',
       'SectionMinEntropy', 'SectionMaxEntropy', 'SectionMinRawsize',
       'SectionMaxRawsize', 'SectionMinVirtualsize', 'SectionMaxVirtualsize',
       'SectionMaxPhysical', 'SectionMinPhysical', 'SectionMaxVirtual',
       'SectionMinVirtual', 'SectionMaxPointerData', 'SectionMinPointerData',
       'SectionMaxChar', 'SectionMainChar', 'DirectoryEntryImport',
       'DirectoryEntryImportSize', 'DirectoryEntryExport',
       'ImageDirectoryEntryExport', 'ImageDirectoryEntryImport',
       'ImageDirectoryEntryResource', 'ImageDirectoryEntryException',
       'ImageDirectoryEntrySecurity'],
      dtype='object')
```

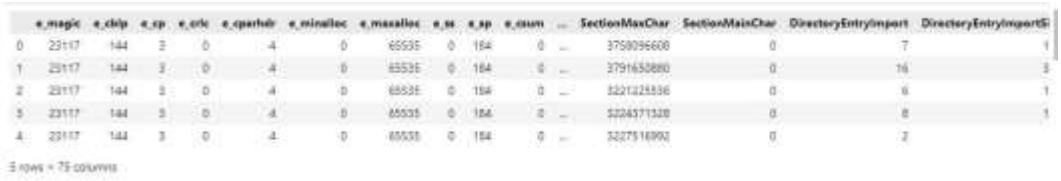
Gambar 4.5 Hasil Data Kolom

Berikut seluruh daftar nama kolom 79 data column yang ada pada dataset malware ini. Data.columns digunakan untuk menampilkan nama nama kolom pada dataframe.

Dropped_data.head()

Berikut variable yang di hapus terdiri dari ‘Nama’, ‘Machine’, ‘TimeDateStamp’, ‘Malware’. Dropped_data.head() berfungsi untuk menghapus variabel yang tidak bermakna

```
dropped_data = data.drop([‘Name’, ‘Machine’,
‘TimeDateStamp’,’Malware’], axis=1)
dropped_data.head()
```



	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minalloc	e_maxalloc	e_ss	e_sp	e_csum	...	SectionMaxChar	SectionMainChar	DirectoryEntryImport	DirectoryEntryImportSi
0	23117	144	3	0	4	0	65535	0	184	0	...	3758096608	0	7	1
1	23117	144	3	0	4	0	65535	0	184	0	...	3791650880	0	16	3
2	23117	144	3	0	4	0	65535	0	184	0	...	3221225536	0	9	1
3	23117	144	3	0	4	0	65535	0	184	0	...	3224371328	0	8	1
4	23117	144	3	0	4	0	65535	0	184	0	...	3227516992	0	2	

5 rows × 79 columns

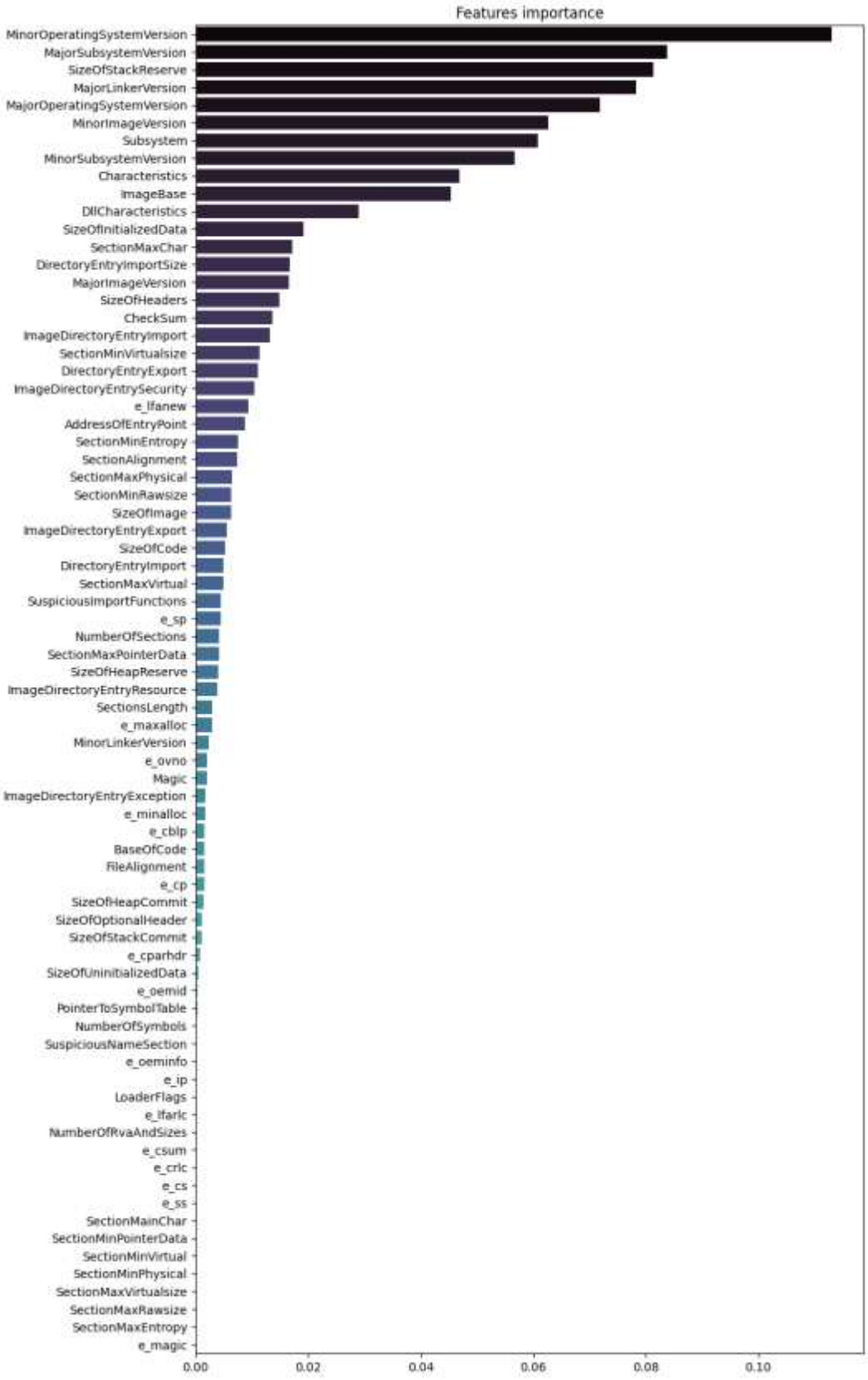
Gambar 4.6 Hasil Dropped data (Menghapus Data yang Tidak Bermakna)

4.2.3. Feature Importance

```
importance = rfc.feature_importances_
importance_dict = {dropped_data.columns.values[i]: importance[i] for
i in range (len(importance))}
sorted_dict = {k: v for k, v in sorted(importance_dict.items(),
key=lambda item: item[1])}
plt.figure(figsize=(10, 20))
sns.barplot(y=list(sorted_dict.keys())[::-1],
x=list(sorted_dict.values())[::-1], palette='mako')
plt.title('Features importance')
```

```
Text(0.5, 1.0, 'Features importance')
```

Hasil dari tahap Feature Importance merupakan sekumpulan fitur beserta ukuran tingkat kepentingannya. Setelah pentingnya fitur ditentukan, fitur dapat dipilih dengan tepat. Berikut merupakan deretan hasil feature importance dari yang terpenting sampai terendah.



Gambar 4.7. Hasil Feature Importance

Dengan informasi tersebut, kita dapat memahami dengan lebih baik bagaimana fitur-fitur pada model *Random Forest Classifier* berpengaruh dalam memprediksi hasilnya. Plot tersebut dapat membantu dalam mengevaluasi kualitas model dan memberikan insight pada proses feature selection atau reduksi dimensi yang mungkin diperlukan paca model tersebut. Dari plot diatas kita dapat melihat bahwa fitur *minoroperatingsystemversion* yang paling membantu model kita untuk membedakan malware dan bukan malware.

4.2.4 Data Splitting

Data splitting merupakan aspek penting dari data *science*, terutama untuk membuat model berbasis data. Teknik ini membantu memastikan model data yang dibuat sudah akurat dan model dapat digunakan pada proses lanjutan, misalnya *machine learning*. pada dasarnya data *splitting* dibagi menjadi dua bagian yaitu data *training* dan data *testing*. Data *training* atau data latih digunakan untuk melatih dan mengembangkan model. Kumpulan data training biasanya digunakan untuk mengestimasikan parameter yang berbeda atau untuk membandingkan kinerja model yang berbeda. Sedangkan Data *Testing* atau data uji digunakan setelah proses *training* selesai. Data *training* dan *testing* dibandingkan untuk memeriksa apakah model akhir yang digunakan bekerja dengan benar.

```
X_train, X_test, y_train, y_test =
train_test_split(dropped_data, data['Malware'],
test_size=0.2, random_state=0)

print(f'Jumlah Fitur sebanyak {X_train.shape[1]}')
```

Jumlah Fitur sebanyak 75

Langkah pertama import model nya from sklearn, model_selection import train_test_split. Pada train_test_split input X dan Y. Lalu tentukan test size = 0.2 artinya ukuran testing nya adalah 20% sedangkan 80% di alokasikan sebaga train test. Kemudian random state nya pilih 0 yang artinya random state number dan ini digunakan untuk menjamin agar eksperimen ini bisa menghasilkan nilai eksperimen ini bisa menghasilkan nilai yang konsisten. Berikut pemanggilan train test split ini akan membutuhkan 4 variabel maka dari itu disiapkan x_train x_test y_train y_test.

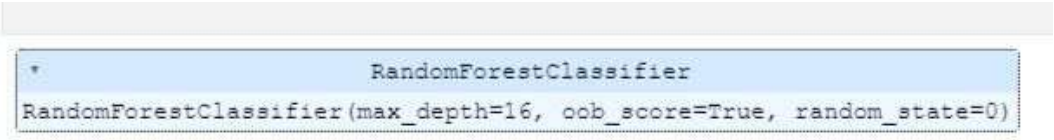
BAB V

PEMBAHASAN

5.1 Pembahasan Model

Pada analisis ini dibangun model pembelajaran mesin menggunakan *random forest clasifir* untuk mengklasifikasikan malware. Model dilatih pada dataset dengan berbagai fitur dan hasilnya menunjukan akurasi yang tinggi sebesar 99%. Laporan *confusion matrix* dan klasifikasi juga menunjukkan presisi tinggi, daya ingat, dan *score f1*. Fitur importance fitur menunjukkan pentingnya setiap fitur dalam klasifikasi malware. Model terakhir yang dilatih disimpan menggunakan perpustakaan acar. Analisis ini menyoroti potensi penggunaan algoritme pembeljaran mesin dalam klasifikasi malware

```
rfc = RandomForestClassifier(
    n_estimators=100 ,
    random_state=0,
    oob_score = True,
    max_depth = 16)
rfc.fit(X_train, y_train)
```



Gambar 5.1. Hasil Random Forest Classifier

Untuk membangun model *Random Forest* kita akan menggunakan fungsi *RandomForestClassifier* dari modul *sklearn.ensemble*. Beberapa parameter penting pada fungsi tersebut adalah sebagai berikut:

- N_estimators : Atur jumlah pohon yang akan digunakan (default 100)
- Random state : Atur status acak untuk memastikan reproduktivitas (default 0)
- Oob_score : enable_out_of_bag (oob) score (default true)
- Max_depth : Atur kedalaman maksimum pohon (default 16)
- Rfc.fit(X_train, y_train) : Sesuaikan classifier dengan data training

5.2 Klasifikasi Menggunakan Random Forest

Selanjutnya fungsi *classification_report* digunakan untuk menghasilkan evaluasi komprehensif dari kinerja model pada data uji. Variabel y_test dan y_pred masing-masing adalah nilai target aktual dan nilai prediksi. Parameter target_names digunakan untuk menentukkan label kelas untuk variabel target. Dalam kasus ini, target variabel memiliki dua kelas: “Bukan Malware” dan “Malware”.

Hasil prediksi *RandomForestClassifier* untuk membuat prediksi guna menguji keakuratan model

```
y_pred = rfc.predict(X_test)

print(classification_report(y_test, y_pred, target_names=['Benign',
'Malware']))
```

	Precision	Recall	F1-Score	Support
Benign	0,99%	96%	97%	1004
Malware	0,99%	100%	99%	2919
Accuracy			99%	3923
Marco avg	0,99%	98%	98%	3923
Weighted avg	99%	99%	99%	3923

Tabel 5.1 Hasil Akurasi Random Forest

. Ini menampilkan skor presisi, recall, f1. Pada eksperimen yang telah dilakukan terdapat beberapa factor yang menjadi indikator berjalannya tahap pembelajaran dan klasifikasi untuk mendapatkan akurasi model, balancing terhadap dataset dan visualisasi dataset malware. Kemudian ini hasil *accuracy* nya 99%, sebagai bentuk evaluasi terhadap model *random forest.*

Precision merupakan jumlah prediksi positif yang benar dibagi dengan jumlah total prediksi positif. *Recall* merupakan jumlah prediksi positif yang benar dibagi dengan jumlah total observasi positif yang sebenarnya. *Skor f1* merupakan rata-rata harmonic dari presisi dan recall, dengan nilai yang lebih tinggi menunjukkan keseimbangan yang lebih baik Dan juga dari persamaannya, kita tahu **semakin False Positive (FP), membuat precision semakin besar.** Sedangkan untuk Recall, **semakin kecil False Negative (FN) membuat recall semakin besar.** Sedangkan F1-Score mengambarkan perbandingan rata – rata *precision* dan *recall* yang dibobotkan.

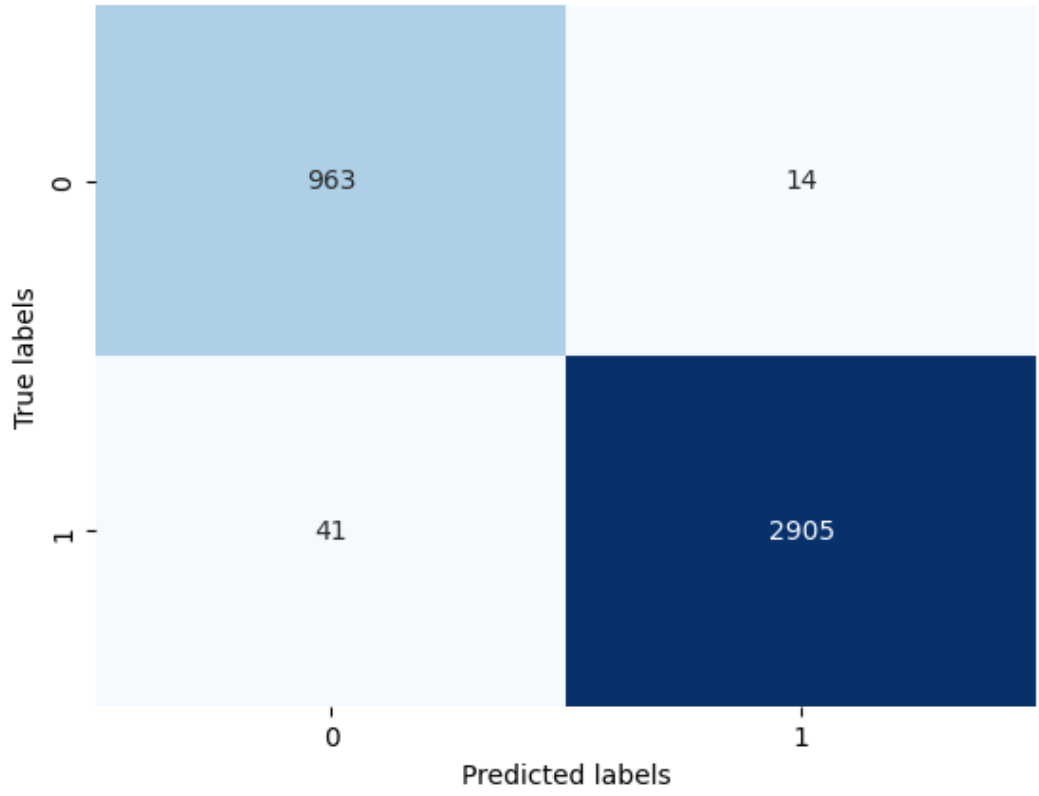
5.3 Evaluasi Model

Confusion Matrix merupakan alat untuk meringkas kinerja algoritma klasifikasi. *Confusion matriks* akan memeberikan kita gambaran yang jelas tentang kinerja model klasifikasi dan jenis kesalahan yang dihasilkan oleh model tersebut. Ini memberi kita ringkasan prediksi yang

benar dan salah yang dikelompokkan berdasarkan masing-masing kategori.Dengan menggunakan *confusion matrix* heatmap tersebut, kita dapat memahami seberapa baik model *Machine Learning* dapat melakukan prediksi pada dataset yang diberikan. Dengan melihat jumlah *True Positive* dan *True Negative* yang besar, kita dapat menyimpulkan bahwa model tersebut mampu melakukan prediksi dengan baik. Sebaliknya, jika terdapat banyak *False Positive* dan *False Negative*, hal ini menunjukkan bahwa model tersebut perlu ditingkatkan performanya. Ringkasan disajikan dalam bentuk table.

```
Ax=sns.heatmap(confusion_matrix(y_pred, y_test), annot=True,
fmt=”d”, cmap=plt.cm.Blues, cbar=False)
ax.set_xlabel(‘Predicted labels’)
  ax.set_ylabel(‘True labels’)
```

Confusion Matriks



Gambar 5.3 Confusion Matrix

5.4 Visualisasi Fitur

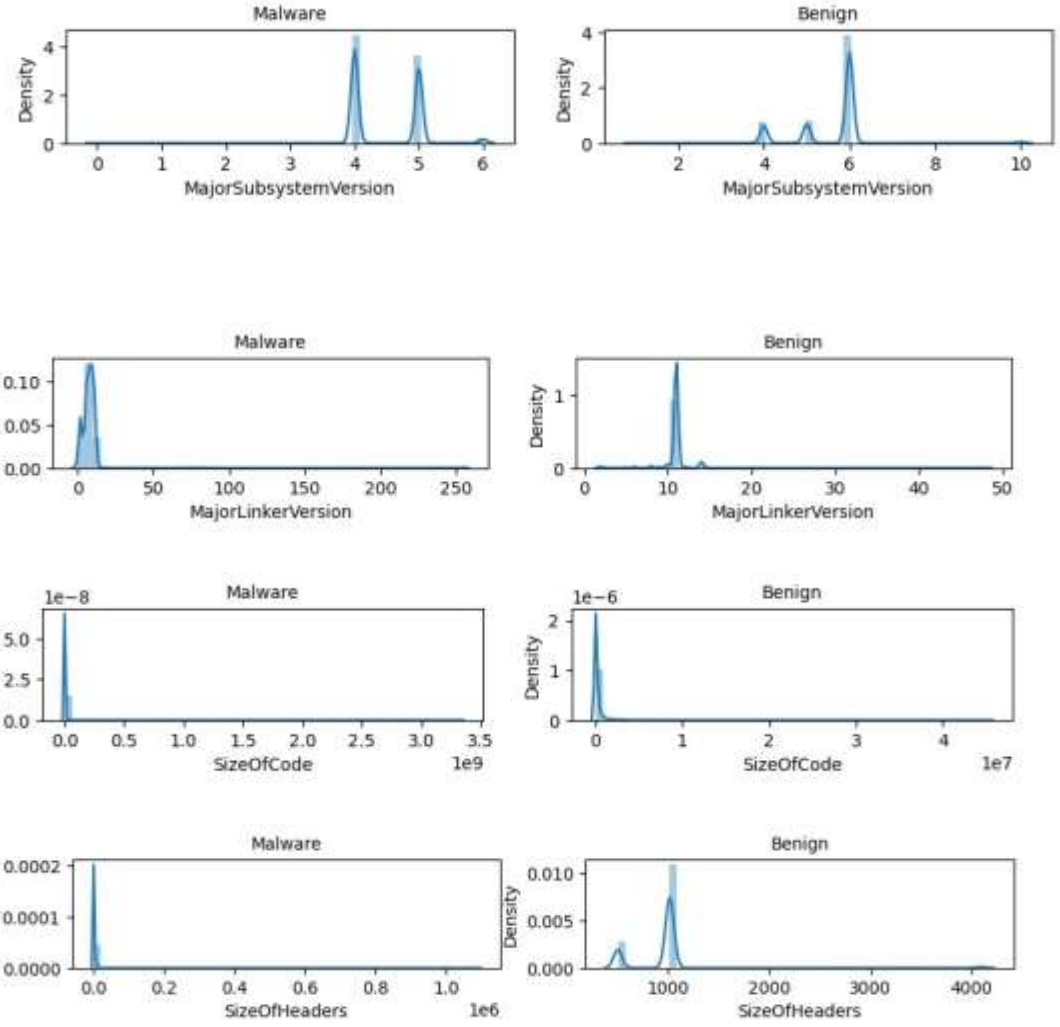
Terdiri dari 11 kolom yang akan di prediksi dan di simpan dalam variable features. Kemudian plt.figure digunakan untuk membuat kanvas plot kosong, *figsize* berfungsi mengatur ukuran gambar, subplot digunakan untuk membuat multiple plot yang memiliki *weight* dan *height* yang sama. Distplot merepresentasikan distribusi data secara *univariate*.

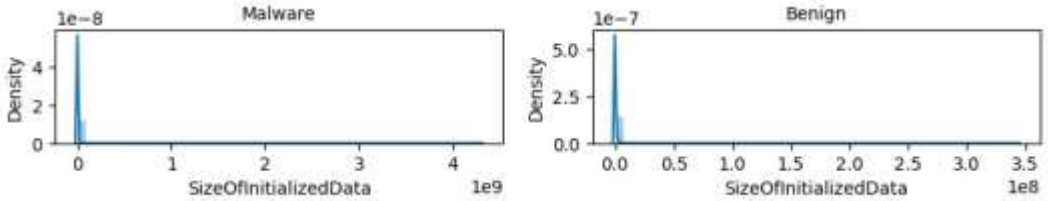
Berdasarkan grafik yang diperoleh dapat diketahui karakteristik data dari setiap hasil klasifikasi, dimana terdapat perbedaan file*size* untuk setiap Malware dan *Benign*/Bukan Malware.

```
features = ['MajorSubsystemVersion', 'MajorLinkerVersion',
'SizeOfCode', 'SizeOfImage', 'SizeOfHeaders',
'SizeOfInitializedData', 'SizeOfUninitializedData',
'SizeOfStackReserve', 'SizeOfHeapReserve', 'NumberOfSymbols',
'SectionMaxChar']
```

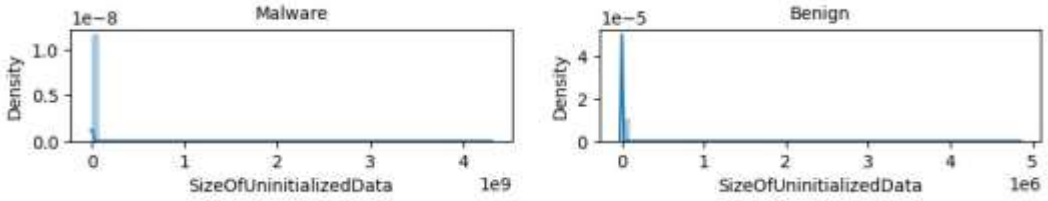
```
i=1

for feature in features:
    plt. figure(figsize=(10, 15))
    ax1 = plt.subplot(len(features), 2, i)
    sns.displot(data[data['Malware']==1][feature], ax=ax1,
kde_kws={'bw': 0.1})
    ax1.set_title(f'Malware', fontsize=10)
    ax2 = plt.subplot(len(features), 2, i+1)
    sns.distplot(data[data['Malware']==0][feature], ax=ax2,
kde_kws={'bw':0.1})
    ax2.set_title(f'Benign', fontsize=10)
    i= i+2
```

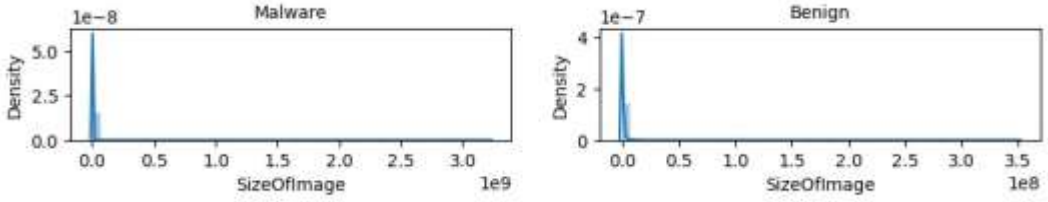



Malware
1e−8
Density
SizeOfInitializedData
1e9

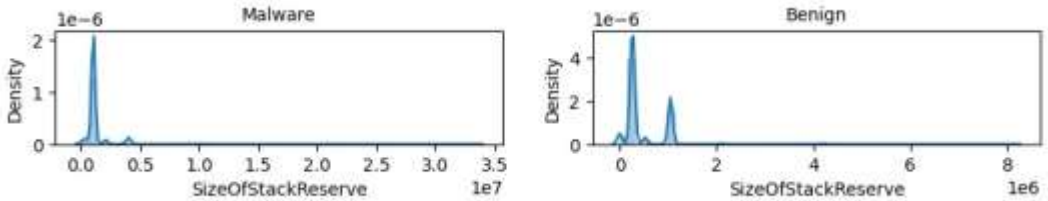
Benign
1e−7
Density
SizeOfInitializedData
1e8


Malware
1e−8
Density
SizeOfUninitializedData
1e9

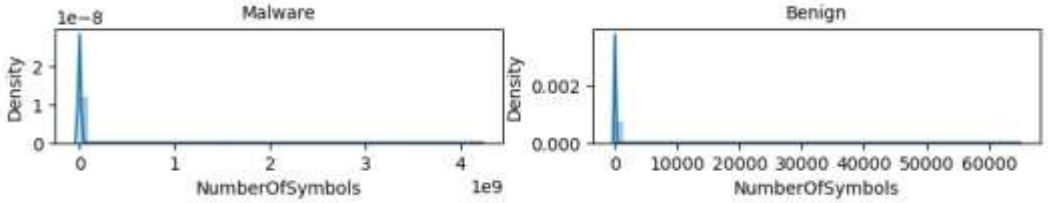
Benign
1e−5
Density
SizeOfUninitializedData
1e6


Malware
1e−8
Density
SizeOfImage
1e9

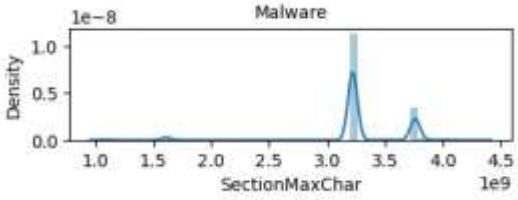
Benign
1e−7
Density
SizeOfImage
1e8

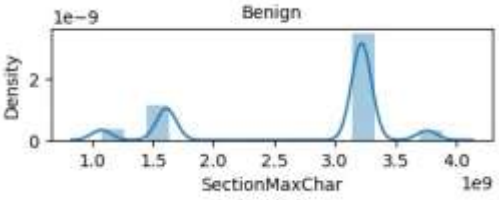

Malware
1e−6
Density
SizeOfStackReserve
1e7

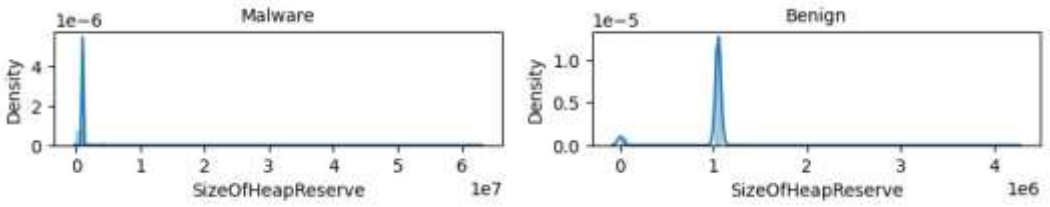
Benign
1e−6
Density
SizeOfStackReserve
1e6


Malware
1e−8
Density
NumberOfSymbols
1e9

Benign
Density
NumberOfSymbols


Malware
1e−8
Density
SectionMaxChar
1e9


Benign
1e−9
Density
SectionMaxChar
1e9



Gambar 5.4 Hasil Visualisasi Fitur

Setiap grafik menampilkan histogram distribusi nilai-nilai fitur pada dua kelompok file *executable*: Malware dan Benign. Distribusi fitur yang signifikan antara Malware dan Benign dapat dilihat dari perbedaan bentuk kurva pada setiap histogram. Pada beberapa fitur, perbedaan distribusi antara kedua kelompok file *executable* sangat signifikan, seperti pada fitur *SizeOfUninitializedData* dan *SizeOfHeapReserve*. Sementara pada beberapa fitur lainnya, perbedaan distribusi antara kedua kelompok tidak terlalu jelas, seperti pada fitur *NumberOfSymbols*.

Dengan demikian, hasil dari kode program ini dapat membantu dalam melakukan klasifikasi file *executable* sebagai Malware atau Benign berdasarkan nilai-nilai fitur yang diambil dari file tersebut. Hasil visualisasi tersebut dapat digunakan untuk melatih model *Machine Learning* yang dapat memprediksi apakah suatu file *executable* termasuk dalam kelompok Malware atau Benign.

Setiap grafik akan menunjukkan distribusi nilai-nilai fitur dari dua kelompok file *executable*, yaitu Malware dan Benign. Kelompok Malware ditandai dengan histogram berwarna biru, sedangkan kelompok Benign ditandai dengan histogram berwarna orange. Setiap grafik akan ditampilkan dalam subplot yang berbeda, sehingga pengguna dapat membandingkan distribusi fitur antara kelompok Malware dan Benign secara berdampingan.

Selain itu, pengguna juga dapat melihat perbedaan distribusi fitur antara Malware dan Benign dari bentuk kurva pada histogram yang ditampilkan. Kurva yang memiliki puncak dan/atau bentuk yang berbeda antara kedua kelompok dapat menunjukkan perbedaan distribusi fitur yang signifikan antara Malware dan Benign. Oleh karena itu, output dari kode program tersebut akan membantu

pengguna dalam menganalisis dan memahami data file executable yang telah diolah.

BAB VI

PENUTUP

6.1 Kesimpulan

Model random forest memberikan hasil yang sangat baik tanpa *preprocessing* pada data. Hasilnya bagus meskipun datanya tidak seimbang. Tidak perlu menggunakan teknik apapun untuk menyeimbangkannya. Penskalaan/skaling tidak perlu dilakukan, model *random forest* adalah model partisi *rekursif* yang bergantung pada partisi data karena ia bekerja pada pemisahan nilai fitur dan tidak melakukan perhitungan di dalamnya. Hasil penelitian menunjukkan bahwa model memiliki presisi 0,99 untuk kelas ”Bukan Malware” dan “Malware”, *recall* 0,96 untuk “Bukan Malware” dan 1,00 untuk “Malware”, serta *f1-score* 0,98 untuk “Bukan Malware” dan 0,99 untuk “Malware”. Akurasi modelnya adalah 0,99 yang cukup baik rata-rata tertimbang dari *presisi,recall, dan f1-score* juga 0,99.

analisis ini dibangun model pembelajaran mesin menggunakan *random forest clasifir* untuk mengklasifikasikan *malware*. Model dilatih pada dataset dengan berbagai fitur dan hasilnya menunjukan akurasi yang tinggi sebesar 99%. Laporan *confusionmatrix* dan klasifikasi juga menunjukkan presisi tinggi, daya ingat, dan *score f1*. Plot kepentinga fitur menunjukkan pentingnya setiap fitur dalam klasifikasi *malware*. Model terakhir yang dilatih disimpan menggunakan perpustakaan acar. Analisis ini menyoroti potensi penggunaan algoritme pembeljaran mesin dalam klasifikasi malware.

6.2 Saran

Setelah penelitian ini selesai, peneliti memberikan beberapa saran untuk dilakukan pengembangan penelitian selanjutnya:

1. Pada penelitian selanjutnya dapat mempertimbangkan dataset apa yang akan di klasifikasikan atau di olah, juga dapat menambahkan metode atau algoritma yang lain sebagai pembanding.
2. Pada penelitian selanjutnya juga dapat mengembangkan penelitian serupa dengan melakukan pembangunan sistem.

DAFTAR PUSTAKA

[1] Togu Novriansyah Turnip, Chatrine Febriyanti Manurung, Yogi Septian Lubis 2023 "Klasifikasi Malware Android Aplikasi Menggunakan Random Forest Berdasarkan Fitur Statik," *Jurnal Teknik Informatika dan Sistem Informasi, Vol. 10, No.1, Maret 2023*.

[2] Edward Tansen, Deris Wahyu Nurdianto "Program studi teknik komputer, Universitas amikom yogyakarta", *Jurnal Teknologi Informasi Vol.4, No.2, Desember 2020.*

[3] Triawan Adi Cahyono, Victor Wahanggara, Darmawan Ramadana "Analisis dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis dan Malware Analisis Statis," *Jurnal Sistem & Teknologi Informasi Indonesia, Vol. 2, No. 1, Februari 2017*

[4] Zalavadiya & Priyanka, Klasifikasi Malware: http://repository.uin-suska.ac.id/16332/8/7.%20BAB%20II%20LANDASAN%20TEORI.pdf.

[5] Raden Budiarto Hadiprakoso, Wahyu Rendra Aditya, Febriora Novia Pramitha," Analisis Statis Deteksi Malware Android Menggunakan Algoritma Supervised Machine Learning,"*CyberSecurity dan forensik Digital, Vol. 5, No. 1, Mei 2022.*

[6] Henny Wahyu Sulistyo, Hardian Oktavianto "Analisis klasifikasi Kanker Payudara Menggunakan Algoritma Random Forest,"*Jurnal Informatika Vol.8 No.2 Februari 2020*

[7] Yitsak Wanli Sitorus, Parman Sukarno, Satria Mandala, "Analisis Deteksi Malware Android menggunakan Metode Support Vector Machine & Random Forest,"*Jurnal e-Proceeding of Engineering, Vol. 8, No. 6,, Desember 2021*.

[8] Devi Rizky Septani, Nur Widiyasono, Husni Mubarok, "Investigasi Serangan Malware Njrat Pada PC," *Jurnal Edukasi dan Penelitian*

Informatika (JEPIN), Vol. 2, No. 2, 2016.

[9] Robi Aziz Zuama, Syaifur Rahmatullah, Yuri Yulian "Analisis Performa Algoritma Machine Learning pada Prediksi Penyakit Cerebrovascular Accidents," *Jurnal Media Informatika Budidarma, Vol. 6, No. 1, Januari 2022*

[10] Fikri Bahriar, Nur Widyasono, Aldy Putra Aldya, "Memory Volatile Forensik Untuk Deteksi Malware Menggunakan Algoritma Machine Learning,"*Jurnal Teknik Informatika dengan Sistem Informasi, Vol. 4, No. 2, Agustus 2018*.

[11] Syafrial Fachri Pane, Jenly Ramdan, "Pemodelan Machine Learning : Analisis Sentimen Masyarakat Terhadap Kebijakan PPKM Menggunakan Data Twitter,"*Jurnal Sistem Cerdas (2022), Vol. 5, No. 1 eISSN : 2622-8254.*

[12] Fajar Mu Alim, Rahmi Hidayati, "Implementasi Metode Random Forest Untuk Penjurusan Siswa Di Madrasah Aliyah Negeri Sintang,"*Jurnal Jupiter, Vol. 14 No. 1 Bulan April, Tahun 2022.*

[13] http://library.binus.ac.id/eColls/eThesisdoc/Bab2/2010-1-00247-IF%20BAB%202.pdf

[14] Karsito, Santi Susanti,"Klasifikasi Kelayakan Peserta Pengajuan Kredit Rumah Dengan Algoritma Naïve Bayes Di Perumahan Azzura Residencia,"*Jurnal Teknologi Pelita Bangsa, Vol. 9, No. 3, Maret 2019 ISSN : 2407-3903*

LAMPIRAN

Name	e_magic	e_cblp	e_cp	e_crlc	e_cparhdr	e_minallo	e_maxallo	e_ss	e_sp	e_csum	e_ip	e_cs	e_lfarlc	e_ovno	e_oemid
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	17744	0	332	1	29305	15462	19547	29295	267
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0

VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	29822	0	8192	0	45211	0	22281	0	60447	0	10469	0	53249	17289	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	

VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	64	1	0	2	0	65535	0	184	0	0	0	10	0	46080	52489
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	50688	6928
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0

VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	280
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	208
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	208
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	176
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	208
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	288
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	272
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	64
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240

VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	272
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	328
VirusShar	23117	144	3	0	4	0	65535	0	184	0	1011	217	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	200
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	272
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	176
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	176
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	200
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	200
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	208

VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	272
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	296
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	200
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	64
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	240
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	224
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	280
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	128
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	184
VirusShar	23117	64	1	0	2	0	65535	0	184	0	0	0	10	0	46080	52489	64
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	272
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	248
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	264
VirusShar	23117	80	2	0	4	15	65535	0	184	0	0	0	64	26	0	0	256
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	216
VirusShar	23117	144	3	0	4	0	65535	0	184	0	0	0	64	0	0	0	232

Kode Program

```
import pandas as pd
import numpy as np
import pickle
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report,
confusion_matrix
import os
import warnings
warnings.simplefilter('ignore')


data = pd.read_csv("dataset_malwares.csv")
```

```
        data.head()
        data.describe()

data.columns

      dropped_data = data.drop(['Name', 'Machine',
      'TimeDateStamp','Malware'], axis=1)
      dropped_data.head()

importance = rfc.feature_importances_
importance_dict = {dropped_data.columns.values[i]: importance[i] for
i in range (len(importance))}
sorted_dict = {k: v for k, v in sorted(importance_dict.items(),
key=lambda item: item[1])}
plt.figure(figsize=(10, 20))
sns.barplot(y=list(sorted_dict.keys())[::-1],
x=list(sorted_dict.values())[::-1], palette='mako')
plt.title('Features importance')

            X_train, X_test, y_train, y_test =
            train_test_split(dropped_data, data['Malware'],
            test_size=0.2, random_state=0)
    rfc = RandomForestClassifier(
        n_estimators=100 ,
        random_state=0,
        oob_score = True,
        max_depth = 16)
    rfc.fit(X_train, y_train)


    y_pred = rfc.predict(X_test)


        print(classification_report(y_test, y_pred, target_names=['Benign',
        'Malware']))
            Ax=sns.heatmap(confusion_matrix(y_pred, y_test), annot=True,
            fmt="d", cmap=plt.cm.Blues, cbar=False)
```

ax.set_xlabel(‘Predicted labels’)

ax.set_ylabel(‘True labels’)

```
i=1

for feature in features:
    plt. figure(figsize=(10, 15))
    ax1 = plt.subplot(len(features), 2, i)
    sns.displot(data[data['Malware']==1][feature], ax=ax1,
kde_kws={'bw': 0.1})
    ax1.set_title(f'Malware', fontsize=10)
    ax2 = plt.subplot(len(features), 2, i+1)
    sns.distplot(data[data['Malware']==0][feature], ax=ax2,
kde_kws={'bw':0.1})
    ax2.set_title(f'Benign', fontsize=10)
    i= i+2
```

BIODATA MAHASISWA EVAN
UNIVERSITAS ICHSAN GORONTALO
JURUSAN TEKNIK INFORMATIKA

Nama	: Evan Valdis Tjahjadi
Tempat/tgl lahir	: Gorontalo, 14 Februari 2001
Jenis kelamin	: Pria
Alamat	:jln. Hos cokroaminoto
RT/RW	: 002/003
Kelurahan	: LIMBA U I
Kecamatan	: Kota Selatan
Agama	: Kristen Protestan
Status perkawinan	: Belum Menikah
Pekerjaan	: Mahasiswa
Kewerganegaraan	: WNI
Riwayat pendidikan	:
SD	: SDN 46 Kota Gorontalo
SMP	: SMPN 2 Kota Gorontalo
SMA	: SMAN 1 Kota Gorontalo
Perguruan Tinggi	: UNIVERSITAS ICHSAN GORONTALO

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
UPT. PERPUSTAKAAN FAKULTAS
SK. MENDIKNAS RI NO. 84/D/O/2001
Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 011/Perpustakaan-Fikom/V/2023

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota	:	Evan Valdis Tjahjadi
No. Induk	:	T3119066
No. Anggota	:	M202339

Terhitung mulai hari, tanggal : Selasa, 09 Mei 2023, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.

Gorontalo, 09 Mei 2023
Mengetahui,
Kepala Perpustakaan

Apriyanto Alhamad, M.Kom
NIDN : 0924048601

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI
No. **146/FIKOM-UIG/R/V/2023**

Yang bertanda tangan di bawah ini :

Nama	: Irvan Abraham Salihi, M.Kom
NIDN	: 0928028101
Jabatan	: Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa	:	Evan Valdis Tjahjadi
NIM	:	T3119066
Program Studi	:	Teknik Informatika (S1)
Fakultas	:	Fakultas Ilmu Komputer
Judul Skripsi	:	Klasifikasi Malware Menggunakan Machine Learning

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **21%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujiankan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Gorontalo, 11 Mei 2023
Mengetahui
Dekan,

Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

Tim Verifikasi,

Zulfrianto Y. Lamasigi, M.Kom
NIDN. 0914089101

Terlampir :
Hasil Pengecekan Turnitin

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER
SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001
Jl. Achmad Nadjamuddin No. 17 Telp (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT KETERANGAN PENELITIAN

Nomor : 14/FIKOM-UIG/SKP/I/2023

Yang bertanda tangan dibawah ini :

N a m a	: Irvan Abraham Salihi, M.Kom
N I D N	: 0928028101
Jabatan	: Dekan Fakultas Ilmu Komputer

Dengan ini Menerangkan bahwa :

N a m a Mahasiswa	: Evan Valdis Tjahjadi
N I M	: T3119066
Program Studi	: Teknik Informatika

Bahwa yang bersangkutan benar-benar telah melakukan penelitian tentang **"Klasifikasi Malware Menggunakan Teknik Machine Learning (studi kasus Prodi Teknik Informatika Fakultas Ilmu Komputer Universitas Ichsan Gorontalo)** " Guna untuk meyelesaikan Studi pada Program Studi Teknik Informatika Fakultas Ilmu Komputer, dan bersangkutan telah menyelesaikan penelitian Tersebut pada **TGL 09 Februari Januari 2023** sesuai dengan waktu yang telah di tentukan.

Demikian Surat Keterangan ini dibuat dan digunakan untuk seperlunya.

Gorontalo, 09 Januari 2023

Dekan

Irvan Abraham Salihi, M.Kom

NIDN. 0928028101

turnitin

Similarity Report ID: oid:25211:35141666

PAPER NAME

SKRIPSI_T3119066_EVANVALDISTJAHJADI.docx

AUTHOR

T3119066-Evan Valdis Tjahjadi evantjahjadi4@gmail.com

WORD COUNT

6691 Words

CHARACTER COUNT

45956 Characters

PAGE COUNT

46 Pages

FILE SIZE

1.3MB

SUBMISSION DATE

May 10, 2023 12:04 PM GMT+8

REPORT DATE

May 10, 2023 12:04 PM GMT+8

21% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 20% Internet database
- Crossref database
- 0% Submitted Works database
- 5% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Quoted material
- Small Matches (Less then 10 words)