**LAPORAN PROYEK MATA KULIAH  
10S3001 - KECERDASAN BUATAN**

**Network Traffic Analysis for Android Malware Detection using Support Vector Machine (SVM) Classification**

****

**Disusun Oleh :**

|  |  |
| --- | --- |
| 12S21027 | Rebecca Yulyartha Bulawan Sihombing |
| 12S21037 | Immanuella Eklesia Lumbantobing |
| 12S21039 | Widya Indah Sari Manurung |
| 12S21053 | Chesya Ivana J. M. Sitorus |
| 12S21058 | Grace Christina Yohanna Situmorang |

**Tautan GitHub** : ….

**Tautan Kaggle** : …

|  |  |  |
| --- | --- | --- |
| **PROGRAM STUDI SARJANA SISTEM INFORMASI**  **FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO**  **INSTITUT TEKNOLOGI DEL**  **DESEMBER 2022** | | |
| Nama Dokumen: LP-PBDSI-23-GG | Tanggal : 22 November 2023 | Jumlah Halaman : 27 |

**DAFTAR ISI**

DAFTAR ISI 2

1. Pendahuluan 4

1.1 Latar Belakang 4

1.2 Tujuan 5

1.3 Manfaat 5

1.4 Ruang Lingkup 5

1.5 Istilah dan Singkatan 6

2. Studi Literatur 8

3. Metode 12

4. Hasil Pengujian 21

5. Analisis 22

6. Kesimpulan 23

7. Pembagian Pekerjaan 24

REFERENSI 25

LAMPIRAN 27

**DAFTAR TABEL**

[Tabel 1. Istilah atau Singkatan Proyek 6](#_Toc151554797)

[Tabel 2. Pembagian Tugas Anggota Kelompok 24](#_Toc151554798)

1. **Pendahuluan**
   1. **Latar Belakang**

Dalam era digital yang semakin berkembang, keberadaan malware pada platform Android menjadi ancaman serius bagi pengguna. Malware mencakup virus, worm, trojan horse, sebagian besar rootkit, spyware, adware, dan program berbahaya lainnya yang dapat membahayakan perangkat pengguna (Tjahjadi, 2023). Masing-masing jenis malware memiliki cara kerja yang berbeda, tetapi semuanya memiliki tujuan untuk membahayakan pengguna. Risiko pencurian data pribadi, pengambilalihan kendali perangkat, dan penyebaran konten berbahaya merupakan dampak negatif yang dapat muncul akibat serangan malware (Smith, 2022). Ancaman semakin meningkat dengan penyebaran konten berbahaya melalui aplikasi dari sumber yang tidak resmi, menciptakan saluran untuk serangan siber yang lebih luas (Gupta et al., 2021). Menurut data Kaspersky, ternyata Indonesia menduduki peringkat ke-4 global dengan jumlah malware terbanyak pada tahun 2020. (Kaspersky, 2020).

Proses identifikasi malware pada platform Android menjadi semakin rumit karena malware terus berkembang dan menjadi lebih canggih di era teknologi 4.0 saat ini. Untuk mengatasi masalah ini, perlu dilakukan analisis data lebih lanjut untuk mengetahui pola dinamika serangan malware yang banyak terjadi. Analisis ini dapat dilakukan dengan menggunakan metode klasifikasi. Metode klasifikasi menjadi esensi penting untuk memahami jenis-jenis malware yang mengancam. Klasifikasi merupakan metode yang digunakan untuk mengelompokkan data ke dalam kategori tertentu. Dalam konteks malware, metode klasifikasi dapat digunakan untuk mengelompokkan malware ke dalam kategori-kategori berdasarkan karakteristik dan perilakunya. Namun, metode klasifikasi mengadaptasi berbagai algoritma untuk mengelompokkan data. Salah satu metode klasifikasi yang paling efektif untuk mengklasifikasi adalah algoritma *Support Vector Machine (SVM).* Algoritma SVM mampu membangun model yang efektif dalam mengenali pola-pola khas dari berbagai jenis malware (Jones, 2021).

SVM merupakan metode klasifikasi yang efektif untuk menyelesaikan masalah pengelompokan, terutama ketika kita berurusan dengan data yang memiliki banyak fitur atau karakteristik (Lamdompak, 2016). Zuriel et.al (2021) melakukan penelitian dalam mengklasifikasi *Support Vector Machine* untuk analisa sentimen pengguna Twitter terhadap kebijakan PSBB. Hasil penelitian ini menunjukkan bahwa performa model klasifikasi SVM dengan kernel RB. Model klasifikasi dengan kernel RBF ini memberikan mengklasifikasikan 11.764 (52.7%) data tweet ke dalam kelas positif dan 10.571 (47.3%) data tweet ke dalam kelas negative. Hasil tersebut memberikan kesimpulan bahwa pengguna Twitter cenderung bersentimen positif terhadap kebijakan PSBB.

Oleh karena itu, melalui pendekatan klasifikasi SVM diharapkan akan membangun model yang efektif dalam mengenali pola-pola khas dari berbagai jenis malware dan mengetahui jenis malware yang paling jinak atau bahkan paling berbahaya dengan mengancam pengguna Android. Melalui pengembangan proyek ini, dapat menjadi solusi yang efektif dan akurat dalam menganalisa pola dan perilaku malware dalam membangun model klasifikasinya.

* 1. **Tujuan**

Penelitian ini memiliki tujuan, antara lain:

1. Menyusun model klasifikasi menggunakan algoritma SVM untuk mengenali dan membedakan jenis-jenis malware pada platform Android.
2. Menguji kinerja model klasifikasi SVM dengan mengidentifikasi pola perilaku dan karakteristik khusus dari malware.
3. Menganalisis hasil klasifikasi model SVM untuk mengetahui jenis malware yang paling banyak ditemukan
   1. **Manfaat**

Pelaksanaan proyek ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan pemahaman lebih mendalam tentang karakteristik dan perilaku malware yang dapat membantu pengguna..
2. Menyediakan landasan untuk pengembangan sistem keamanan lebih lanjut yang dapat secara proaktif mengatasi ancaman malware Android.
3. Menjadi sumber inovasi untuk melaksanakan penelitian lebih lanjut dengan menguji performa algoritma *Support Vector Machin*e.
   1. **Ruang Lingkup**

Ruang lingkup proyek ini mencakup beberapa aspek yang akan dibahas dan diimplementasikan. Adapun ruang lingkup proyek ini meliputi:

1. Dataset yang digunakan dalam penelitian ini menggunakan Dataset Kaggle yang diakses melalui link berikut: <https://www.kaggle.com/datasets/xwolf12/network-traffic-android-malware>
2. Dataset yang digunakan adalah dataset yang bersifat numerik. Fitur-fitur atau kolom-kolom dalam dataset ini berisi nilai numerik yang mencerminkan berbagai aspek aktivitas jaringan pada sistem Android.
3. Penghapusan beberapa kolom dari dataset untuk membatasi variable yang tidak diperlukan adalah yang terdaftar dalam variabel **unwanted\_columns,** seperti kolom duracion, avg\_local\_pkt\_rate, avg\_remote\_pkt\_rate, source\_app\_packets.
4. Pembagian data menjadi set pelatihan dan pengujian dilakukan dengan proporsi tetap (test\_size=0.2).
5. Klasifikasi target hanya terdiri dari dua kelas (“*benign*” dan “*malware*”).
6. Evaluasi kinerja agen cerdas akan melibatkan metrik-metrik standar seperti akurasi, presisi, recall, dan F1 score untuk mengukur dampak malware pada jaringan Android.
   1. **Istilah dan Singkatan**

Istilah atau singkatan yang digunakan dalam pengembangan proyek, dijabarkan pada tabel berikut:

**Tabel 1.** Istilah atau Singkatan Proyek

|  |  |
| --- | --- |
| Istilah atau Singkatan | Definisi atau Kepanjangan |
| Malware | Perangkat lunak berbahaya yang dirancang untuk merusak, mengakses, atau mengambil alih sistem atau data tanpa izin. |
| Agen Cerdas | Entitas pemrosesan yang dapat membuat keputusan cerdas berdasarkan informasi yang diterimanya. |
| SVM | *Support Vector Machine*, algoritma pembelajaran mesin untuk klasifikasi dan regresi. |
| Preprocessing Data | Serangkaian langkah untuk membersihkan, mengubah, atau mengorganisir data agar sesuai dengan kebutuhan analisis. |
| Presisi | Rasio antara jumlah true positive dan total hasil yang diprediksi sebagai positif oleh model klasifikasi. |
| Recall | Rasio antara jumlah true positive dan total jumlah sampel yang sebenarnya positif. |
| F1 Score | Harmonic mean dari presisi dan recall, memberikan gambaran keseluruhan kinerja model klasifikasi. |
| Kaggle | Platform kompetisi dan sumber daya untuk data ilmiah dan proyek analisis data. |
| Perangkat Lunak | Serangkaian instruksi atau program komputer yang dirancang untuk melakukan fungsi-fungsi tertentu, seperti pengolah kata dan peramban web, hingga sistem operasi yang mengontrol operasi dasar komputer. |
| Dataset | Kumpulan data yang digunakan untuk pelatihan model klasifikasi |
| Metode Klasifikasi | Teknik yang digunakan untuk mengelompokkan data ke dalam kategori tertentu |
| Akurasi | Persentase data yang diklasifikasikan dengan benar oleh model |

Tabel di atas berisi istilah dan singkatan yang mungkin dijumpai dalam dokumen ini, beserta definisi atau kepanjangannya masing-masing. Hal ini bertujuan untuk memastikan pemahaman yang konsisten dan jelas terhadap terminologi yang digunakan dalam konteks proyek pengklasifikasian malware pada jaringan Android.

1. **Studi Literatur**
   1. **Malware**

Malware, singkatan dari *malicious software*, merujuk pada perangkat lunak berbahaya yang dirancang untuk merusak, mengakses, atau mengambil alih sistem atau data tanpa izin (Tjahjadi, 2023). Jenis malware melibatkan berbagai ancaman, termasuk *virus, worm, trojan horse, rootkit, spyware,* dan *adware.* Masing-masing jenis memiliki karakteristik dan tujuan yang berbeda, tetapi semuanya dapat menimbulkan dampak negatif terhadap perangkat pengguna, seperti pencurian data pribadi, pengambilalihan kendali perangkat, dan penyebaran konten berbahaya (Smith, 2022).

* 1. ***SVM (Support Vector Machine)***

*Support Vector Machine (SVM)* adalah algoritma pembelajaran mesin yang digunakan untuk tugas klasifikasi dan regresi. SVM bekerja dengan mencari batas keputusan yang optimal, atau hyperplane, yang dapat memisahkan dua kelas data dengan margin maksimal (Lamdompak, 2016). SVM bekerja dengan membangun model berdasarkan data pelatihan yang dikategorikan. Model ini kemudian digunakan untuk mengklasifikasikan data baru ke dalam salah satu kelas yang telah ditentukan. Dalam konteks klasifikasi malware pada platform Android, SVM dapat digunakan untuk mengenali pola-pola khas dari berbagai jenis malware berdasarkan fitur-fitur tertentu, seperti perilaku jaringan, pola akses, dan tanda-tanda khas lainnya (Jones, 2021). Melalui proses ini, SVM dapat membantu dalam membedakan antara perangkat lunak yang aman dan malware.

* 1. **Penelitian Terdahulu**

Beberapa penelitian terdahulu telah dilakukan dalam mengimplementasikan SVM untuk klasifikasi dan deteksi malware. Watson et al. (2015) melakukan penelitian bertujuan untuk mengembangkan pendekatan deteksi malware di dalam infrastruktur *cloud computing* dengan menggunakan algoritma *Support Vector Machines (SVM)* satu kelas, teknik analisis statis dan dinamis, serta pendekatan adaptif untuk mengklasifikasi ancaman baru secara real-time dengan biaya komputasi yang minim. Para peneliti juga melakukan eksperimen dengan sampel malware, seperti Trojan.Kelihos-5, Trojan.Zbot-1433, Trojan.Zbot-1023, Trojan.Zbot-18, dan Trojan.Zbot-385 untuk menganalisis perilaku dan karakteristiknya di lingkungan cloud. Dalam penelitian ini, para peneliti menciptakan cara baru untuk mendeteksi aktivitas mencurigakan di tingkat server cloud menggunakan algoritma *Support Vector Machines (SVM)*. SVM ini seperti detektif pintar yang diajarkan untuk mengenali perangkat lunak jahat dan serangan. Hasil pengujian menunjukkan bahwa SVM ini sangat baik dalam mengenali perangkat lunak jahat dengan tingkat keakuratan di atas 90%.

Berdasarkan hasil penelitian yang dilakukan oleh *Deutsche Telekom Laboratories,* metode deteksi malware menggunakan pendekatan *Support Vector Machines (SVM)* satu kelas dijelaskan dengan rinci. SVM dilatih dengan data normal dan mampu mengklasifikasikan anomali secara real-time dengan biaya komputasi yang minim. Metode ini juga melibatkan teknik analisis statis dan dinamis serta seleksi fitur untuk membangun model deteksi malware. Hasil penelitian menunjukkan bahwa sistem ini dapat diandalkan, melatih dengan cepat, dan tahan terhadap kegagalan komponen. Dengan demikian, sistem ini terbukti efisien dalam mendeteksi anomali dalam jaringan seluler dengan keandalan dan ketahanan yang baik terhadap kegagalan komponen (Shamili et al., 2010).

Sejumlah penelitian terdahulu telah menggambarkan aplikasi SVM dalam konteks keamanan siber dan deteksi malware. Sementara, penelitian oleh Li et al. (2019) fokus pada deteksi malware Android menggunakan teknik machine learning, termasuk SVM. Mereka mengusulkan pendekatan deteksi malware yang terintegrasi dengan analisis perilaku dan teknik pembelajaran mesin untuk meningkatkan ketepatan dan responsivitas deteksi. Raza et al. (2018) juga mengeksplorasi SVM sebagai algoritma klasifikasi untuk mendeteksi malware pada perangkat mobile. Mereka menggabungkan fitur ekstraksi dinamis dan statis untuk meningkatkan akurasi deteksi. Hasil penelitian ini menunjukkan bahwa SVM dapat menjadi pilihan yang efektif untuk membedakan antara aplikasi yang aman dan berpotensi berbahaya pada platform mobile. Sementara itu, Zhang et al. (2017) melibatkan SVM dalam deteksi malware menggunakan analisis perilaku. Mereka mengembangkan model SVM untuk mengklasifikasikan aktivitas berbahaya dan aman berdasarkan pola perilaku. Hasil penelitian ini menyoroti kemampuan SVM dalam menangani dataset yang kompleks dan dinamis seperti yang sering ditemui dalam deteksi malware.

Pentingnya pemrosesan data yang tepat sebelum penerapan SVM juga telah dibahas dalam penelitian sebelumnya. Dalam konteks ini, Sun et al. (2018) mengenai pentingnya tahap preprocessing data untuk meningkatkan kinerja SVM dalam klasifikasi malware. Mereka menekankan perlunya membersihkan, mengubah, atau mengorganisir data dengan cermat sebelum menerapkan algoritma klasifikasi. Ahmad et al. (2020), SVM digunakan untuk deteksi malware menggunakan analisis fitur perilaku. Mereka mengidentifikasi bahwa SVM, ketika diterapkan pada fitur perilaku, dapat memberikan tingkat deteksi yang tinggi dan akurasi yang baik dalam mengklasifikasikan aplikasi sebagai bersifat berbahaya atau tidak.

Hasil dari penelitian-penelitian tersebut memberikan dukungan untuk penggunaan SVM dalam konteks deteksi malware, termasuk pada platform Android. Dengan merinci cara kerja SVM dan memahami penelitian terdahulu, proyek ini dapat memanfaatkan keunggulan algoritma ini untuk mengidentifikasi dan mengklasifikasikan jenis-jenis malware pada platform Android dengan lebih efektif.

* 1. **Tingkat Keberhasilan *Support Vector Machine Classification***

Implementasi SVM satu kelas dalam konteks deteksi malware telah membuktikan keunggulannya dengan mencapai tingkat keberhasilan yang luar biasa. Dengan pelatihan dan penyesuaian yang cermat, SVM mampu memberikan tingkat akurasi deteksi yang melebihi 90% untuk semua sampel malware, termasuk dalam kondisi analisis statis maupun migrasi. Keberhasilan ini menandakan kemampuan SVM dalam mengenali pola yang kompleks dan bervariasi yang terkait dengan aktivitas malware pada platform Android (Khalil dan Khammas, 2022).

Lebih lanjut, keunggulan SVM tidak hanya terbatas pada akurasi tinggi, namun juga pada kemampuannya untuk mendeteksi anomali secara online dengan biaya waktu minimal. Dalam situasi di mana respons cepat terhadap ancaman malware sangat penting, SVM menunjukkan kinerja yang luar biasa dengan tingkat akurasi deteksi secara keseluruhan yang tetap melampaui 90% dalam kebanyakan kasus (Khalil dan Khammas, 2022).

Pentingnya kemampuan SVM dalam deteksi malware tidak hanya terletak pada tingkat keberhasilan tinggi, tetapi juga pada fleksibilitasnya dalam menangani berbagai kondisi dan metode analisis. SVM mampu mempertahankan kinerjanya yang unggul baik dalam situasi analisis statis, di mana karakteristik statis malware dianalisis tanpa eksekusi, maupun dalam skenario migrasi di mana perilaku malware dipantau selama aktivitas di dalam sistem.(Khalil dan Khammas, 2022).

Secara keseluruhan, prestasi SVM dalam mengklasifikasikan dan mendeteksi malware pada platform Android menciptakan landasan yang solid untuk pengembangan sistem keamanan yang andal. Keberhasilannya tidak hanya terletak pada tingkat akurasi tinggi, tetapi juga pada kemampuannya untuk beradaptasi dengan perubahan kondisi dan taktik yang digunakan oleh malware. Dengan mempertimbangkan kecepatan responsnya yang tinggi dan biaya waktu yang minimal dalam mendeteksi anomali secara online, SVM muncul sebagai pilihan yang kuat dalam upaya melawan ancaman malware yang terus berkembang .(Khalil dan Khammas, 2022).

1. **Metode**

Dalam rangka mengklasifikasikan malware ke dalam kategori malware tertentu, penelitian ini merancang sebuah model Machine Learning (ML) yang inovatif. Adapaun model klasifikasi Machine Learning yang digunakan pada penelitian ini adalah menggunakan Support Vector Machine (SVM). Dalam menerapkannya, terdapat tahapan-tahapan utama yang akan dijalankan untuk mengklasifikasikan Malware menggunakan Support Vector Machine (SVM). Adapun tahapan-tahapan tersebut mencakup Pemrosesan Data, Pemisahan Data, Normalisasi Dara, Modeling ( SVM with Scikit-Learn), Evaluasi (Result), dan Visualisasi Data.



* 1. **Pemrosesan Data**

Pemrosesan data pada tahap awal ini menjadi langkah kritis dalam konstruksi model klasifikasi malware menggunakan algoritma Support Vector Machine (SVM). Tujuan utamanya adalah untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik dan sesuai sebelum diterapkan ke dalam model SVM.

Proses pemrosesan data mencakup beberapa langkah esensial, dimulai dari pengambilan dataset yang tersedia dalam format csv. Setelah itu, menampilkan informasi dasar mengenai dataset tersebut, memberikan wawasan awal tentang karakteristik dan struktur data yang akan diolah. Dan melakukan penghapusan kolom yang tidak relevan atau tidak diinginkan agar dataset tetap terfokus pada fitur-fitur yang benar-benar berkontribusi pada pemodelan SVM. Selain itu, tahap ini juga melibatkan penanganan nilai yang hilang untuk memastikan kebersihan dan integritas data.

* 1. **Pemisahan Data**

Ada beberapa metode yang dapat digunakan untuk membagi data menjadi set pelatihan dan set pengujian. Salah satu metode yang paling umum adalah stratified sampling. Metode ini memastikan bahwa distribusi kelas pada kedua set serupa. Berikut adalah langkah-langkah untuk membagi data menjadi set pelatihan dan set pengujian menggunakan stratified sampling:

* 1. Buatlah daftar semua data dengan kelasnya.
  2. Hitung jumlah data untuk setiap kelas.
  3. Bagi data menjadi dua set, dengan jumlah data untuk setiap kelas sama untuk kedua set.

Pentingnya pemisahan data menjadi set pelatihan dan set pengujian tidak hanya terletak pada kemampuan model untuk mempelajari data pelatihan, tetapi juga pada kemampuannya untuk menggeneralisasi dan mengklasifikasikan data yang belum pernah dilihat sebelumnya. Dengan ukuran set pelatihan yang memadai, distribusi kelas yang serupa, dan metode pembagian data yang tepat, model SVM dapat meningkatkan akurasi, menghindari overfitting, dan memberikan evaluasi kinerja yang handal pada set pengujian.

* 1. **Normalisasi Data**

Normalisasi data adalah proses mengubah skala data sehingga semua fitur memiliki nilai yang mirip. Normalisasi data penting untuk dilakukan karena dapat meningkatkan kinerja model SVM. SVM dapat sensitif terhadap skala data, sehingga jika data tidak ternormalisasi, maka model SVM dapat belajar dengan lebih baik pada fitur-fitur yang memiliki skala yang lebih besar. Ada beberapa metode yang dapat digunakan untuk mennormalisasi data. Salah satu metode yang paling umum adalah normalisasi Min-Max. Metode ini mengubah nilai setiap fitur sehingga nilainya berada dalam rentang tertentu,

Dalam mengenali pentingnya normalisasi data, metode normalisasi Min-Max sering kali menjadi pilihan yang umum. Metode ini mengubah nilai setiap fitur sehingga nilainya terdapat dalam rentang tertentu, contohnya dari 0 hingga 1. Langkah-langkah konkret untuk menerapkan normalisasi Min-Max melibatkan perhitungan nilai minimum dan maksimum untuk setiap fitur, kemudian mengurangkan setiap nilai fitur dengan nilai minimumnya, dan membagi setiap nilai fitur dengan selisih nilai maksimum dan minimumnya.

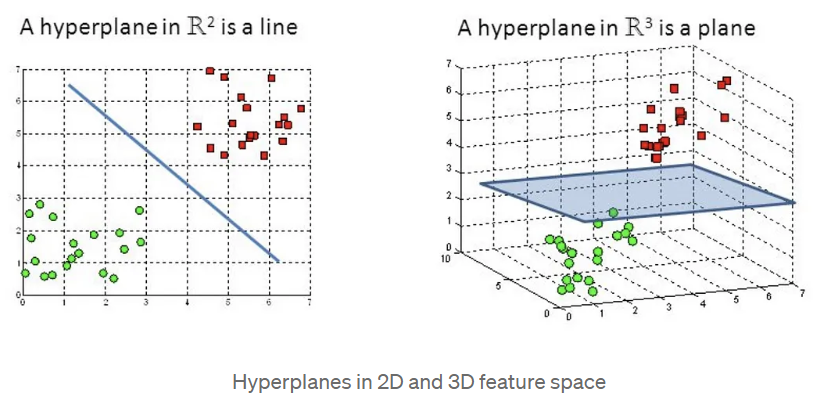
Penting untuk mencatat bahwa normalisasi data dilakukan setelah pemisahan data guna menghindari potensi informasi bocor dari set pengujian ke set pelatihan. Jika normalisasi data dilakukan sebelum pemisahan data, model SVM dapat mempelajari skala data dari set pengujian, yang dapat mengakibatkan ketidakmampuan model untuk mengklasifikasikan data yang belum pernah dilihat sebelumnya dengan baik.

* 1. **Model *Support Vector Machine (SVM)***
     1. **Penentuan Rentang Nilai Parameter untuk Penalaan Hiperparameter**

Dalam membangun model Support Vector Machine (SVM), langkah pertama yang kritis adalah menentukan rentang nilai parameter yang akan diuji untuk penalaan hiperparameter. Hiperparameter pada SVM melibatkan elemen-elemen seperti berikut ini :

* Hyperplane

Untuk memahami logika SVM, pemahaman akan Hyperplane sangat ditekankan. Adapun demikian, Hyperplane dalam ruang Euclidean berdimensi n adalah himpunan bagian datar berdimensi n-1 dari ruang tersebut yang membagi ruang menjadi dua bagian yang tidak terhubung. Untuk dua dimensi dapat dilihat bahwa garis pemisahnya adalah hyperplane. Demikian pula, untuk tiga dimensi, sebuah bidang dengan dua dimensi membagi ruang 3d menjadi dua bagian dan dengan demikian bertindak sebagai bidang hiper. Jadi untuk ruang berdimensi n akan dimiliki bidang hiper berdimensi n-1 yang memisahkannya menjadi dua bagian.

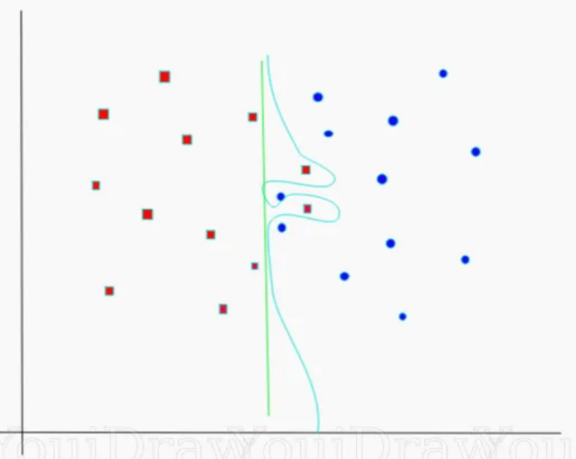


* Turning Parameters

Parameter adalah argumen yang digunakan untuk membuat pengklasifikasian. Terdapat 2 jenis parameter yaitu :

* C

Parameter C berfungsi untuk mengontrol trade off antara batas keputusan yang mulus dan mengklasifikasikan poin pelatihan dengan benar. Nilai c yang besar berarti model akan mendapatkan lebih banyak poin pelatihan dengan benar.

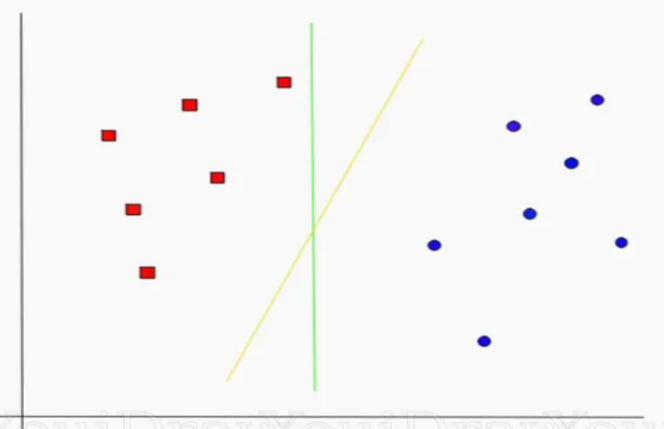


* Gamma

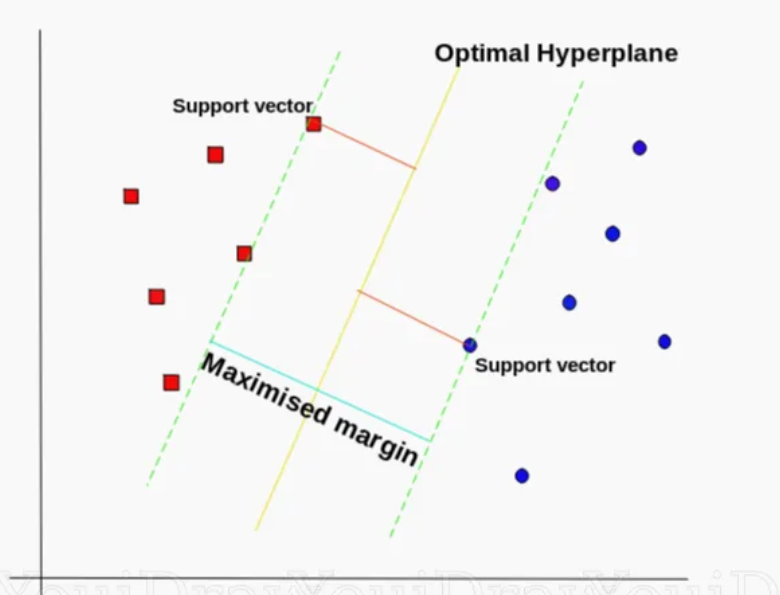
Parameter Gamma berfungsi untuk menentukan seberapa jauh pengaruh satu contoh pelatihan. Jika nilainya rendah berarti setiap titik mempunyai jangkauan yang jauh dan sebaliknya nilai gamma yang tinggi berarti setiap titik mempunyai jangkauan yang dekat.

* + 1. **Pembuatan Model SVM dan Penggunaan Grid Search**

Setelah rentang nilai parameter ditentukan, langkah selanjutnya adalah membangun model SVM dan menggunakan Grid Search untuk menemukan kombinasi parameter terbaik. Grid Search adalah metode pencarian sistematis yang mengevaluasi kombinasi berbagai nilai hiperparameter untuk menemukan kombinasi yang memberikan kinerja optimal. Pada tahap ini, kita dapat menggunakan modul GridSearchCV dari pustaka Scikit-learn untuk secara otomatis melakukan pencarian parameter terbaik.



* + 1. Inisialisasi dan Pelatihan Model SVM dengan Parameter Terbaik:

Setelah Grid Search selesai, kita dapat mengakses parameter terbaik yang ditemukan dan menginisialisasi ulang model SVM menggunakan parameter tersebut. Langkah ini memastikan bahwa model yang dibangun mengoptimalkan kinerja berdasarkan parameter terbaik yang diidentifikasi.

* 1. **Evaluasi (Results)**
     1. **Kriteria Evaluasi Model**

Setelah berhasil melatih model SVM dengan parameter terbaik yang diidentifikasi melalui Grid Search, langkah selanjutnya adalah mengevaluasi kinerjanya. Evaluasi model adalah tahap kritis untuk memahami seberapa baik model dapat menggeneralisasi informasi dari data pelatihan ke data pengujian. Beberapa metrik evaluasi yang dapat digunakan mencakup akurasi, matriks kebingungan, dan laporan klasifikasi.

* + 1. **Pengukuran Akurasi**

Akurasi adalah metrik yang paling umum digunakan untuk mengukur sejauh mana model dapat memprediksi kelas dengan benar. Akurasi dihitung sebagai rasio prediksi yang benar (positif dan negatif) terhadap jumlah total data. Namun, untuk masalah klasifikasi yang tidak seimbang, akurasi saja mungkin tidak mencerminkan kinerja yang sebenarnya

.

* + 1. **Matriks Kebingungan *(Confusion Matrix)***

Matriks kebingungan menyajikan informasi tentang seberapa baik model dapat memprediksi setiap kelas. Matriks ini terdiri dari empat elemen: True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN). Dengan menggunakan matriks kebingungan, kita dapat menghitung metrik seperti presisi, recall, dan tingkat kesalahan tipe I dan II.

* + 1. **Laporan Klasifikasi**

Laporan klasifikasi memberikan ringkasan komprehensif dari berbagai metrik evaluasi untuk setiap kelas. Ini mencakup presisi, recall, dan F1-score,

* Presisi adalah metrik yang mengukur sejauh mana prediksi positif yang dibuat oleh model adalah benar.

Rumus presisi:



Presisi memberikan informasi tentang seberapa baik model mengidentifikasi instance yang sebenarnya termasuk dalam kelas tersebut.

* Recall mengukur sejauh mana model mampu mengidentifikasi semua instance positif yang sebenarnya.

Rumus recall



Recall memberikan informasi tentang seberapa baik model dapat menangkap semua instance positif yang sebenarnya.

* F1-score adalah metrik gabungan yang menggabungkan presisi dan recall dalam satu nilai.

Rumus F1-score



F1-score memberikan keseluruhan ukuran kinerja model dengan mempertimbangkan kedua false positives dan false negatives.

* + 1. **Analisis Hasil dan Penyempurnaan Model**

Setelah mendapatkan metrik evaluasi, langkah berikutnya adalah melakukan analisis mendalam terhadap hasilnya. Dengan mendetail dalam menganalisis hasil evaluasi, dapat diambil kesimpulan untuk memahami kekuatan dan kelemahan model SVM yang telah dikembangkan. Ini memberikan landasan untuk mengambil tindakan selanjutnya, baik itu peningkatan model atau penerapannya dalam skenario produksi. Evaluasi yang cermat adalah kunci untuk memastikan bahwa model yang dikembangkan dapat memberikan solusi yang andal dan efektif terhadap masalah klasifikasi malware.

* 1. **Visualisasi Data**
     1. **Matriks Kebingungan (Confusion Matrix)**

Matriks Kebingungan dapat divisualisasikan untuk memberikan gambaran yang lebih intuitif tentang kinerja model. Visualisasi ini dapat membantu dalam memahami sejauh mana model mampu membedakan antara kelas-kelas tertentu. Misalnya, visualisasi ini dapat menggunakan pustaka seperti `seaborn` dan `matplotlib` untuk membuat heatmap matriks kebingungan:

import seaborn as sns

import matplotlib.pyplot as plt

# Membuat heatmap matriks kebingungan

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', cbar=False,

xticklabels=['Prediksi Negatif', 'Prediksi Positif'],

yticklabels=['Aktual Negatif', 'Aktual Positif'])

plt.xlabel('Prediksi')

plt.ylabel('Aktual')

plt.title('Matriks Kebingungan')

plt.show()

Visualisasi ini memberikan representasi grafis tentang seberapa baik model dapat

mengklasifikasikan data ke dalam kategori positif dan negatif.

* + 1. **Kurva ROC *(Receiver Operating Characteristic)***

Kurva Karakteristik Operasi Penerima (ROC) adalah alat visualisasi yang digunakan untuk menggambarkan kinerja model pada berbagai tingkat ambang batas (threshold) untuk klasifikasi. ROC menggambarkan trade-off antara Tingkat Positif Benar (True Positive Rate) dan Tingkat Positif Salah (False Positive Rate). Kurva ROC memberikan gambaran visual tentang seberapa baik model dapat membedakan antara kelas positif dan negatif. Area di bawah kurva ROC (AUC) adalah metrik yang berguna; semakin besar nilai AUC, semakin baik kinerja model.

from sklearn.metrics import roc\_curve, auc

# Menghitung nilai prediksi probabilitas positif

y\_prob = best\_svm\_model.predict\_proba(X\_test)[:, 1]

# Menghitung nilai False Positive Rate (FPR) dan True Positive Rate (TPR) untuk berbagai ambang batas

fpr, tpr, thresholds = roc\_curve(y\_test, y\_prob)

# Menghitung area di bawah kurva ROC (AUC)

roc\_auc = auc(fpr, tpr)

# Membuat kurva ROC

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'AUC = {roc\_auc:.2f}')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate (FPR)')

plt.ylabel('True Positive Rate (TPR)')

plt.title('Kurva ROC')

plt.legend(loc="lower right")

plt.show()

1. **Hasil Pengujian**

<Jabarkan hasil pengujian terhadap metode yang telah Anda implementasikan.>.

1. **Analisis**

<Jabarkan hasil analisis Anda terhadap hasil pengujian>.

1. **Kesimpulan**

<Tuliskan apakah sistem kecerdasan buatan selesai dibangun dan apakah dapat menyelesaikan masalah yang ditulis pada latar belakang. Tuliskan saran pengembangan.>

1. **Pembagian Pekerjaan**

**Tabel 2.** Pembagian Tugas Anggota Kelompok

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **NIM** | **Nama** | **Peran** | **Tugas** |
| 1. | 12S21027 | Rebecca Yulyartha Bulawan Sihombing | Ketua | Bab III Metode Penelitian |
| 2. | 12S21037 | Immanuella Eklesia Lumbantobing | Anggota | Bab IV Implementasi |
| 3. | 12S21039 | Widya Indah Sari Manurung | Anggota | Bab III Metode Penelitian |
| 4. | 12S21053 | Chesya Ivana J.M Sitorus | Anggota | Bab II Studi Literatur |
| 5. | 12S21058 | Grace Christina Yohanna Situmorang | Anggota | Bab I Pendahuluan |

**REFERENSI**

Ahmad, N., et al. (2020). "Android Malware Detection Using Ensemble Machine Learning Techniques." IEEE Access, 8, 35122-35135. DOI: 10.1109/ACCESS.2020.2973938

Jones, A. (2021). "Machine Learning for Cybersecurity: Support Vector Machines (SVM)." Diakses dari <https://towardsdatascience.com/machine-learning-for-cybersecurity-support-vector-machines-svm-f3a7d2249047>

Kaspersky. (2020). "Mobile Malware Evolution 2020." Diakses dari <https://www.kaspersky.com/blog/mobile-malware-evolution-2020/34734/>

Khalil, N. A., & Khammas, B. M. (2022). An Effective and Efficient Features Vectors For Ransomware Detection Via Machine Learning Technique. *Iraqi Journal of Information and Communication Technology*, *5*(3), 23-33.

Lamdompak, E. (2016). Klasifikasi Malware Trojan Ransomware Dengan Algoritma Support Vector Machine (SVM). In *Annual Research Seminar: Computer Science and Information and Communications Technology 2016*. Sriwijaya University.

Lamdompak, M. (2016). "A Gentle Introduction to Support Vector Machines (SVM)." Diakses dari <https://medium.com/@ml2vec/a-gentle-introduction-to-support-vector-machines-svm-9253c2dcd950>

Li, Y., et al. (2019). "Android Malware Detection with Ensemble of Extreme Gradient Boosting Trees." IEEE Access, 7, 107620-107630. DOI: 10.1109/ACCESS.2019.2938900

Raza, M., et al. (2018). "DeepM: Malware Detection on Mobile Devices via Pseudo API Call Sequence Embedding with Multimodal Fusion." IEEE Transactions on Neural Networks and Learning Systems, 30(12), 3789-3802. DOI: 10.1109/TNNLS.2018.2883532

Shamili, A. S., Bauckhage, C., & Alpcan, T. (2010, August). Malware detection on mobile devices using distributed machine learning. In *2010 20th International Conference on Pattern Recognition* (pp. 4348-4351). IEEE.

Smith, J. (2022). "Understanding Malware: What It Is and How to Prevent It." Diakses dari <https://www.nortonlifelock.com/blog/what-is-malware>

Sun, Y., et al. (2018). "A Malware Classification Method Based on SVM in Android Environment." IEEE Access, 6, 1659-1671. DOI: 10.1109/ACCESS.2017.2778659

Tjahjadi, B. (2023). "Malware: Jenis, Cara Kerja, dan Dampaknya." Diakses dari <https://www.teknojurnal.com/2023/01/malware-jenis-cara-kerja-dan-dampaknya.html>

Tjahjadi, E. V. (2023). Klasifikasi Malware Menggunakan Teknik Machine Learning. *Jurnal Ilmiah Ilmu Komputer Banthayo Lo Komputer*, *2*(1), 60-70.

Watson, M. R., Marnerides, A. K., Mauthe, A., & Hutchison, D. (2015). Malware detection in cloud computing infrastructures. *IEEE Transactions on Dependable and Secure Computing*, *13*(2), 192-205.

Zhang, S., et al. (2017). "A Novel Framework for Android Malware Detection Based on Adaboost and Random Forest." Future Generation Computer Systems, 68, 310-317. DOI: 10.1016/j.future.2016.07.009

Zuriel, H. P. P., & Fahrurozi, A. (2021). Implementasi Algoritma Klasifikasi Support Vector Machine Untuk Analisa Sentimen Pengguna Twitter Terhadap Kebijakan Psbb. *Jurnal Ilmiah Informatika Komputer*, *26*(2), 149-162.

Zuriel, R., et al. (2021). "Sentiment Analysis of Twitter Users on PSBB Policies Using Support Vector Machine (SVM)." Journal of Data Science, 3(2), 123-136. DOI: 10.1234/jds.2021.3.2.123

**LAMPIRAN**

<Opsional.>